Convergence of the optimal non-linear GPC method with iterative state-dependent, linear time-varying approximation

Przemyslaw Orlowski*

* *Institute of Control Engineering, Technical University of Szczecin, Poland,* `orzel@ps.pl`

Keywords: non-linear systems, predictive control, optimal control, discrete time systems

Solution of the optimal non-linear GPC method is mostly obtained in an iterative way with 3 following steps: initialisation, transformation from non-linear system given in general form into time-varying state-dependent form, and checking whether the convergence condition is satisfied. The main aim of this paper is to analyse how the transformation method from nonlinear model into time varying state-space dependent form have effect on the convergence of the algorithm. We try to answer following questions: If chosen transformation method is suitable ? If the convergence to the optimal solution is guaranteed ? If the number of iterations can be cut down ?

1 Introduction

Significant attention has been given recently to nonlinear predictive control (NLPC) methods. A large class of these methods uses common 3 steps algorithm. For example: Kouvartiakis et. al. [1] employ optimal control trajectory calculated in the previous time instant of the control algorithm for the NLPC. Lee et. al. [2] uses similar methodology and employing linearization at points of the seed trajectory for discrete-time model of the system. Also technique presented in [3], [4], [5], [6] uses similar idea as [1], [2], but with the different model representation and optimisation technique. The nonlinear system described by the discrete-time nonlinear state space model can be rearranged into so-called state and control dependent linear form [7], [8]. Non-linear behaviour of the system is included in the state and control dependent matrices. If the future trajectory prediction for the system may be obtained within the algorithm then one can pretend that future behaviour of the system is known during the prediction horizon [3]. Such system can be treated as a linear time-varying (LTV) one. Most often the algorithm has three following common steps [1], [3]:

1. Choose the initial control trajectory, cost functional (e.g. weighting matrices) and possibly the reference trajectory.
2. Transform the model of a non-linear system given in following general form

   $$\mathbf{x}(k+1)=\mathbf{f}\left(\mathbf{x}(k),\mathbf{u}(k),k\right) \tag{1}$$

   into the time-varying state-dependent form given by

   $$\mathbf{x}(k+1)=\mathbf{A}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)\cdot\mathbf{x}(k)+\mathbf{B}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)\cdot\mathbf{u}(k) \tag{2}$$

3. Calculate new control and check if the convergence condition is satisfied. If not go to step 2, otherwise return the final optimal control.

Our considerations are concerned around transformation method from step 2. It is obvious that there exists infinite number of possible transformations of eq. (1) into eq. (2). We try to answer if chosen transformation is suitable and convergent.

2 Model description

It is well known that every discrete LTV system can be described by evolutionary operators.

$$\mathbf{x}(k) = (\mathbf{N}\mathbf{x}_0)(k) + (\mathbf{L}(\mathbf{B}\mathbf{u}))(k) \tag{3}$$

Using the past trajectory, the matrices $\mathbf{A}(k) = \mathbf{A}(\mathbf{x}(k), \mathbf{u}(k), k)$, $\mathbf{B}(k) = \mathbf{B}(\mathbf{x}(k), \mathbf{u}(k), k)$ may be calculated for the subsequent points on the trajectory and the nonlinear system (1) is approximated by LTV model with matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$. Also if the system is defined on finite time horizon it is possible to define operators **N, L, B** in following block matrix form.

$$\hat{\mathbf{L}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \phi_1^1 & \mathbf{I} & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{I} & \mathbf{0} \\ \phi_1^{N-1} & \cdots & \phi_{N-1}^{N-1} & \mathbf{I} \end{bmatrix} \qquad \hat{\mathbf{N}} = \begin{bmatrix} \phi_0^0 \\ \vdots \\ \phi_0^{N-1} \end{bmatrix} \qquad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}(N-1) \end{bmatrix} \tag{4}$$

where $\phi_i^k = \mathbf{A}(k)\cdot\mathbf{A}(k-1)\cdot\ldots\cdot\mathbf{A}(i)$. Vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{u}}$ have following block vector notation i.e.

$$\hat{\mathbf{x}} = \left[\mathbf{x}^T(1)\cdots\mathbf{x}^T(N)\right]^T, \quad \hat{\mathbf{u}} = \left[\mathbf{u}(0)\cdots\mathbf{u}(N-1)\right] \tag{5}$$

Thus the state equation can be rewritten in following form

$$\hat{\mathbf{x}} = \hat{\mathbf{L}}\hat{\mathbf{B}}\cdot\hat{\mathbf{u}} + \hat{\mathbf{N}}\cdot\mathbf{x}_0 \tag{6}$$

The operator $\hat{\mathbf{L}}\hat{\mathbf{B}}$ is a compact and Hilbert-Schmidt operator from l_2 into l_2 and actually maps boundedly signals $\mathbf{u}(k) \in \mathsf{U} = l_2[0, N]$ into signals $x \in \mathsf{X}$.

3 Convergence of the algorithm

Definition 1. Let the cost functional for arbitrary *i*-th iteration of the algorithm from section 1 is denoted by $J(i) = (\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref})^T \hat{\mathbf{P}}(\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref}) + \hat{\mathbf{u}}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}$. Let the same functional for the next *i*+1 iteration of the same algorithm is denoted by $J(i+1) = (\hat{\mathbf{x}}_\mathbf{n} - \hat{\mathbf{x}}_{ref})^T \hat{\mathbf{P}}(\hat{\mathbf{x}}_\mathbf{n} - \hat{\mathbf{x}}_{ref}) + \hat{\mathbf{u}}_\mathbf{n}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}_\mathbf{n}$. Coefficient of functional convergence is denoted by R_J and defined by following ratio:

$$R_J(i) := \frac{J(i+1)}{J(i)} = \frac{(\hat{\mathbf{x}}_\mathbf{n} - \hat{\mathbf{x}}_{ref})^T \hat{\mathbf{P}}(\hat{\mathbf{x}}_\mathbf{n} - \hat{\mathbf{x}}_{ref}) + \hat{\mathbf{u}}_\mathbf{n}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}_\mathbf{n}}{(\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref})^T \hat{\mathbf{P}}(\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref}) + \hat{\mathbf{u}}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}} \tag{7}$$

Definition 2. Let the state trajectory deviation norm from reference trajectory for any given time horizon is given by $\left\|\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref}\right\|$ for the *i*-th iteration of the algorithm and $\left\|\hat{\mathbf{x}}_\mathbf{n} - \hat{\mathbf{x}}_{ref}\right\|$ for the next *i*+1 iteration. Coefficient of state error convergence is denoted by R_c and defined by following relationship:

$$R_c(i) := \frac{\left\|\hat{\mathbf{x}}_\mathbf{n} - \hat{\mathbf{x}}_{ref}\right\|}{\left\|\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref}\right\|} \quad \text{or in particular case for } \hat{\mathbf{x}}_{ref} = \mathbf{0}, \; R_c(i) := \frac{\left\|\hat{\mathbf{x}}_\mathbf{n}\right\|}{\left\|\hat{\mathbf{x}}\right\|} \tag{8}$$

Theorem 1. State trajectory in the consecutive iteration of the algorithm from section 1, calculated for any nonlinear system transformed into state-dependent LTV form under assumption $\left\|\hat{\mathbf{L}}\hat{\Delta}_\mathbf{A}\right\| < 1$ can be calculated from following equation:

$$\hat{\mathbf{x}}_\mathbf{n} = \left(\mathbf{I} - \hat{\mathbf{L}}\hat{\Delta}_\mathbf{A}\right)^{-1}\left(\hat{\mathbf{x}} + \hat{\mathbf{L}}\left(\hat{\mathbf{B}}_\mathbf{n}\hat{\mathbf{u}}_\mathbf{n} - \hat{\mathbf{B}}\hat{\mathbf{u}}\right)\right) \tag{9}$$

where: $\hat{\Delta}_{\mathbf{A}}\left(\mathbf{x}_{\mathbf{n}}(k),\mathbf{u}_{\mathbf{n}}(k),k\right)=\hat{\mathbf{A}}_{\mathbf{n}}\left(\mathbf{x}_{\mathbf{n}}(k),\mathbf{u}_{\mathbf{n}}(k),k\right)-\hat{\mathbf{A}}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)$ - difference system operator.

$\hat{\mathbf{x}},\hat{\mathbf{u}}$, $\hat{\mathbf{B}},\hat{\mathbf{L}}$ - state, input trajectory, input operator and system operator in *i*-th iteration of the algorithm,

$\hat{\mathbf{x}}_{\mathbf{n}},\hat{\mathbf{u}}_{\mathbf{n}}$, $\hat{\mathbf{x}}_{ref}$, $\hat{\mathbf{B}}_{\mathbf{n}}$ - state, input, reference trajectory and input operator in *i*+1-th iteration of the algorithm.

Proof follows from similar results proved for perturbed systems. The system in *i*+1 iteration can be treated as perturbed system from *i*-th iteration, thus the system in *i*+1-th can be written in following form

$$\hat{\mathbf{x}}_{\mathbf{n}}=\hat{\mathbf{x}}+\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\hat{\mathbf{x}}_{\mathbf{n}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)\text{ or equivalently }\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)\hat{\mathbf{x}}_{\mathbf{n}}=\hat{\mathbf{x}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)$$

To derive $\hat{\mathbf{x}}_{\mathbf{n}}$ trajectory, the term $\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)$ must be invertible. Assuming that $\left\|\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right\|<1$, the term $\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)$ is invertible. Calculating left side inverse of above equation follows to eq. (9). □

Corollary 1. Functional convergence coefficient of the algorithm from section 1 can be evaluated from following equation:

$$R_J(i)=\frac{\left(\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)^{-1}\left(\hat{\mathbf{x}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)\right)-\hat{\mathbf{x}}_{ref}\right)^T\hat{\mathbf{P}}\left(\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)^{-1}\left(\hat{\mathbf{x}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)\right)-\hat{\mathbf{x}}_{ref}\right)+\hat{\mathbf{u}}_{\mathbf{n}}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}_{\mathbf{n}}}{\left(\hat{\mathbf{x}}-\hat{\mathbf{x}}_{ref}\right)^T\hat{\mathbf{P}}\left(\hat{\mathbf{x}}-\hat{\mathbf{x}}_{ref}\right)+\hat{\mathbf{u}}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}}\tag{10}$$

Corollary 2. State error convergence coefficient of the algorithm from section 1 can be evaluated from following equation:

$$R_c(i)=\frac{\left\|\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)^{-1}\left(\hat{\mathbf{x}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)\right)-\hat{\mathbf{x}}_{ref}\right\|}{\left\|\hat{\mathbf{x}}-\hat{\mathbf{x}}_{ref}\right\|}\tag{11}$$

Corollary 3. State error convergence coefficient of the algorithm from section 1 for $\hat{\mathbf{x}}_{ref}=\mathbf{0}$ can be evaluated from following equation:

$$R_c(i)=\frac{\left\|\hat{\mathbf{x}}_{\mathbf{n}}\right\|}{\left\|\hat{\mathbf{x}}\right\|}=\frac{\left\|\left(\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right)^{-1}\left(\hat{\mathbf{x}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)\right)\right\|}{\left\|\hat{\mathbf{x}}\right\|}\le\frac{\left\|\hat{\mathbf{x}}+\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}}-\hat{\mathbf{B}}\hat{\mathbf{u}}\right)\right\|}{\left\|\mathbf{I}-\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right\|\left\|\hat{\mathbf{x}}\right\|}\tag{12}$$

Proofs follow directly from triangle inequality for norms.

Theorem 2. If the optimal control $\hat{\mathbf{u}}_{opt}=\hat{\mathbf{g}}\left(\hat{\mathbf{A}}\left(\mathbf{x}_0,\hat{\mathbf{u}}_{\mathbf{n}}\right),\hat{\mathbf{B}}\left(\mathbf{x}_0,\hat{\mathbf{u}}_{\mathbf{n}}\right)\right)$ is calculated for a some $\hat{\mathbf{u}}_{\mathbf{n}},\mathbf{x}_0$, then the same optimal control follows from the condition:

$$\hat{\mathbf{u}}_{opt}=\hat{\mathbf{g}}\left(\hat{\mathbf{A}}\left(\mathbf{x}_0,\hat{\mathbf{u}}_{opt}\right),\hat{\mathbf{B}}\left(\mathbf{x}_0,\hat{\mathbf{u}}_{opt}\right)\right)\tag{13}$$

In other words, the algorithm is convergent if the optimal control $\hat{\mathbf{u}}_{opt}$ follows directly from time varying system matrices $\hat{\mathbf{A}}_{opt},\hat{\mathbf{B}}_{opt}$ linearized at $\hat{\mathbf{u}}_{opt}$.

Proof. Let us assume that exists such $\hat{\mathbf{u}}_{opt}$, that

$$\mathbf{x}_{opt}(k+1)=\mathbf{A}\left(\mathbf{x}_{opt}(k),\mathbf{u}_{opt}(k),k\right)\cdot\mathbf{x}_{opt}(k)+\mathbf{B}\left(\mathbf{x}_{opt}(k),\mathbf{u}_{opt}(k),k\right)\cdot\mathbf{u}_{opt}(k)\tag{14}$$

$$\hat{\mathbf{A}}_{opt}=\hat{\mathbf{A}}\left(\mathbf{x}_0,\hat{\mathbf{u}}_{opt}\right)\triangleq\mathbf{A}\left(\mathbf{x}_{opt}(k),\mathbf{u}_{opt}(k),k\right),\ \hat{\mathbf{B}}_{opt}=\hat{\mathbf{B}}\left(\mathbf{x}_0,\hat{\mathbf{u}}_{opt}\right)\triangleq\mathbf{B}\left(\mathbf{x}_{opt}(k),\mathbf{u}_{opt}(k),k\right),\ k=0\ldots N-1\tag{15}$$

$$J_{opt}=\left(\hat{\mathbf{x}}_{opt}-\hat{\mathbf{x}}_{ref}\right)^T\hat{\mathbf{P}}\left(\hat{\mathbf{x}}_{opt}-\hat{\mathbf{x}}_{ref}\right)+\hat{\mathbf{u}}_{opt}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}_{opt}=\min\tag{16}$$

New control estimated from step 3 of the algorithm is evaluated from linearized time varying system matrices

$$\hat{\mathbf{u}}_{\mathbf{n}} = \hat{\mathbf{g}}\left(\hat{\mathbf{A}}_{opt}, \hat{\mathbf{B}}_{opt}\right) \tag{17}$$

Let us assume that $\hat{\mathbf{u}}_{\mathbf{n}} \neq \hat{\mathbf{u}}_{opt}$ then $\hat{\mathbf{A}}\left(\mathbf{x}_0, \hat{\mathbf{u}}_{\mathbf{n}}\right) \neq \hat{\mathbf{A}}_{opt}$, $\hat{\mathbf{B}}\left(\mathbf{x}_0, \hat{\mathbf{u}}_{\mathbf{n}}\right) \neq \hat{\mathbf{B}}_{opt}$ and $J_{\mathbf{n}} > J_{opt}$. What is divergent from the optimal control. Thus the equation (13) is necessary condition for convergence of the algorithm from section 1. □

Functional convergence coefficient R_J can be easily estimated numerically on the basis of known current, previous, reference state and input trajectories. Error norm convergence coefficient R_c can be evaluated using only current $\hat{\mathbf{x}}_{\mathbf{n}}$, previous $\hat{\mathbf{x}}$ and reference $\hat{\mathbf{x}}_{ref}$ state trajectories. From the numerical point of view calculating of the coefficients R_c and R_J is quite easy task. Nevertheless from analytical point of view coefficient R_c is easier to analyse. In practise, determining the convergence of the algorithm is strongly connected either with R_c or R_J factor. In respect to similar properties of R_c and R_J we will call further both the coefficients by R.

The algorithm is monotonically convergent if $R \leq 1$. The closer the coefficient to zero, the faster the absolute convergence. However when approaching to the optimal solution $R \to 1$. For linear systems $R = 1$ because the solution is calculated from the first iteration of the algorithm. For value $R > 1$ the algorithm can be divergent. In some cases R_c can oscillate above and below 1. In such case the algorithm is convergent if for consecutive iterations proceed $R \to 1$. Convergence to the optimal solution $J \to J_{opt}$ is always connected with approach zero of the nonlinearity differences $\hat{\Delta}_{\mathbf{A}}$, $\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}} - \hat{\mathbf{B}}\hat{\mathbf{u}}\right)$. Similarly convergence coefficients R_c, R_J approach 1.

Corollary 4. What shall be done to ensure R near 1 ? Two following conditions have to be satisfied. Operators product $\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}$ should approach zero matrix or equivalently $\left\|\hat{\mathbf{L}}\hat{\Delta}_{\mathbf{A}}\right\| \to 0$. Norm of operator $\hat{\mathbf{L}}$ cannot be less then 1, what follows from (4). Norm $\hat{\Delta}_{\mathbf{A}}$ can be arbitrary small. The value is dependent on the degree of nonlinearity of the system and the decomposition into matrices **A** and **B** carried out in step 2 of the algorithm. For the linear system matrix **A** does not depend on input and state, thus the norm is equal to zero. Assumption **A=0** results in $\left\|\hat{\Delta}_{\mathbf{A}}\right\| = 0$. However it also results in arising the difference $\hat{\mathbf{L}}\left(\hat{\mathbf{B}}_{\mathbf{n}}\hat{\mathbf{u}}_{\mathbf{n}} - \hat{\mathbf{B}}\hat{\mathbf{u}}\right)$ especially drastically for small values of input $\hat{\mathbf{u}}$. In most cases it results in divergence of the algorithm, similarly as assumption **B=0**, which results in arising of norm $\left\|\hat{\Delta}_{\mathbf{A}}\right\|$.

4 Additive decomposition

Arbitrary function $\mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k), k\right)$ can be transformed into series of N additive components. The notation can be simplified for fixed input trajectory and initial conditions and $\mathbf{f}_i = \mathbf{f}_i\left(\mathbf{x}(k), \mathbf{u}(k), k\right)$. In particular for indecomposable function N=1.

$$\mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k), k\right) = \sum_{i=1}^{N} \mathbf{f}_i\left(\mathbf{x}(k), \mathbf{u}(k), k\right) = \sum_{i=1}^{N} \mathbf{f}_i \tag{18}$$

Every system (1) can be decomposed into state dependent form (2). In general it takes following form:

$$\begin{gathered}\mathbf{x}(k+1) = \sum_{i=1}^{N} \mathbf{f}_i = \sum_{j=1}^{R}\left(\sum_{i=1}^{N} \alpha_{i,j}\mathbf{f}_i\right) + \sum_{j=1}^{Q}\left(\sum_{i=1}^{N} \beta_{i,j}\mathbf{f}_i\right) = \sum_{i=1}^{N} \alpha_{i,1}\mathbf{f}_i + \ldots + \sum_{i=1}^{N} \alpha_{i,R}\mathbf{f}_i + \sum_{i=1}^{N} \beta_{i,1}\mathbf{f}_i + \ldots + \sum_{i=1}^{N} \beta_{i,Q}\mathbf{f}_i \\ \mathbf{x}(k+1) = \mathbf{a}_1 x_1 + \ldots + \mathbf{a}_R x_R + \mathbf{b}_1 u_1 + \ldots + \mathbf{b}_Q u_Q = \left[\mathbf{a}_1 \cdots\ \mathbf{a}_R\right]\left[x_1 \cdots x_R\right]^T + \left[\mathbf{b}_1 \cdots \mathbf{b}_Q\right]\left[u_1 \cdots u_Q\right]^T = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\end{gathered} \tag{19}$$

where

$$\mathbf{a}_j x_j = \sum_{i=1}^{N} \alpha_{i,j} \mathbf{f}_i, \ \ \mathbf{b}_j u_j = \sum_{i=1}^{N} \beta_{i,j} \mathbf{f}_i, \ \ \forall_i \ \sum_{j=1}^{R} \alpha_{i,j} + \sum_{j=1}^{Q} \beta_{i,j} = 1 \tag{20}$$

Component column vectors of matrices **A**(k) and **B**(k) can be determined under assumption that following limits exists and are finite $\forall_j \lim_{x_j \to 0} \frac{\mathbf{a}_j x_j}{x_j}, \forall_j \lim_{u_j \to 0} \frac{\mathbf{b}_j u_j}{u_j}$. The vectors are given as follows.

$$\mathbf{a}_j = \begin{cases} \dfrac{\mathbf{a}_j x_j}{x_j} & x_j \neq 0 \\ \lim\limits_{x_j \to 0} \dfrac{\mathbf{a}_j x_j}{x_j} & x_j = 0 \end{cases} \qquad \mathbf{b}_j = \begin{cases} \dfrac{\mathbf{b}_j u_j}{u_j} & \mathrm{u}_j \neq 0 \\ \lim\limits_{u_j \to 0} \dfrac{\mathbf{b}_j u_j}{u_j} & \mathrm{u}_j = 0 \end{cases} \tag{21}$$

where

$\mathbf{A}(k) = [\mathbf{a}_1 \cdots \mathbf{a}_R], \mathbf{B}(k) = [\mathbf{b}_1 \cdots \mathbf{b}_Q]$, R – order, Q – number of inputs, $\mathbf{a}_j, \mathbf{b}_j$ - column vectors with R rows

Every nonlinear system can be decomposed into additive form. Although it should be recalled corollary 4. Norms $\left\| \hat{\Delta}_{\mathbf{A}} \right\|$ and $\left\| \hat{\mathbf{L}} \left(\hat{\mathbf{B}}_{\mathbf{n}} \hat{\mathbf{u}}_{\mathbf{n}} - \hat{\mathbf{B}} \hat{\mathbf{u}} \right) \right\|$ should be possibly small.

Let us assume, that function **f**(**x**,**u**,k) can be decomposed into 4 following additive terms:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, k) = \mathbf{f}_1(\mathbf{x}, k) + \mathbf{f}_2(\mathbf{x}, \mathbf{u}, k) + \mathbf{f}_3(\mathbf{u}, k) + \mathbf{f}_4(k) \tag{22}$$

Functions **f** must be continuous and following limits must be finite.:

$$\lim_{x \to 0} \frac{\mathbf{f}_1}{x}, \lim_{u \to 0} \frac{\mathbf{f}_3}{u}, \text{ and either } \lim_{x \to 0} \frac{\mathbf{f}_2}{x}, \text{ or/and } \lim_{u \to 0} \frac{\mathbf{f}_2}{u} \tag{23}$$

Main aim of control is to minimize input and state of the system which follows $\mathbf{x}(k) \to 0, \mathbf{u}(k) \to 0$ for $k \to N$. Decomposition of components **f**$_1$ and **f**$_3$ is intuitive, e.g. **Ax**=**f**$_1$ and **Bu**=**f**$_3$. If the components are assigned in different way often results in fast increasing of norms $\left\| \hat{\Delta}_{\mathbf{A}} \right\|$ i $\left\| \hat{\mathbf{L}} \left(\hat{\mathbf{B}}_{\mathbf{n}} \hat{\mathbf{u}}_{\mathbf{n}} - \hat{\mathbf{B}} \hat{\mathbf{u}} \right) \right\|$. Furthermore limits (23) will be infinite and the algorithm from section 1 would be divergent. Another question is how to assign components **f**$_2$ and **f**$_4$? First of all, the assigning depends on the function $\hat{\mathbf{u}}_{\mathbf{n}} = \hat{\mathbf{g}}(\hat{\mathbf{A}}, \hat{\mathbf{B}})$. If it takes the following form:

$$\hat{\mathbf{u}}_{\mathbf{n}} = -\left(\left(\hat{\mathbf{L}} \hat{\mathbf{B}} \right)^T \hat{\mathbf{P}} \hat{\mathbf{L}} \hat{\mathbf{B}} + \hat{\mathbf{Q}} \right)^{-1} \left(\hat{\mathbf{L}} \hat{\mathbf{B}} \right)^T \hat{\mathbf{P}} \hat{\mathbf{N}} \mathbf{x}_0 \tag{24}$$

significant for the algorithm is conditional number in respect to the inverse of matrix $\left(\left(\hat{\mathbf{L}} \hat{\mathbf{B}} \right)^T \hat{\mathbf{P}} \hat{\mathbf{L}} \hat{\mathbf{B}} + \hat{\mathbf{Q}} \right)$. The number is defined as ratio of maximal to minimal singular value of the matrix. Norms of matrices **A,B** should approach neither zero nor infinity. In practise, the best performance is if the norms of matrices **A,B** have similar order of magnitude.

If the input tends faster to zero then the state of the system $\forall_{k > k_0} |\mathbf{x}(k)| - |\mathbf{u}(k)| > 0$, **f**$_2$ should be assigned to **Bu**, in respect to stronger dependence from **u.** From the same reason **f**$_4$ should be assigned rather to **Ax** then to **Bu**, e.g.

$$\mathbf{Ax} = \mathbf{f}_1 + \mathbf{f}_4, \quad \mathbf{Bu} = \mathbf{f}_2 + \mathbf{f}_3 \tag{25}$$

Minimization of norms $\left\| \hat{\Delta}_{\mathbf{A}} \right\|$ and $\left\| \hat{\mathbf{L}} \left(\hat{\mathbf{B}}_{\mathbf{n}} \hat{\mathbf{u}}_{\mathbf{n}} - \hat{\mathbf{B}} \hat{\mathbf{u}} \right) \right\|$ follows from equations (10), (11), (12). On the other hand the system should be balanced realization or at least norms of matrices **A, B** should be bounded to ensure invertibility in equation (24). When the function **f** does not have components $\mathbf{f_1}$ and $\mathbf{f_4}$, one have to assign a part of component $\mathbf{f_2}$ to **Ax**, to avoid zero matrix **A** and problems with computing inverse in (24).

$$\mathbf{Ax} = a\mathbf{f}_2, \quad \mathbf{Bu} = (1-a)\mathbf{f}_2 + \mathbf{f}_3 \tag{26}$$

Value of coefficient a depends on the form of functions $\mathbf{f_2}$ and $\mathbf{f_3}$. Usually the value does not exceed 0.05-0.1.

5 Numerical examples

In these examples control is calculated by iterative using the algorithm from section 1 and equation (24), where $\mathbf{x}_0$ is initial condition for current prediction, $J = \hat{\mathbf{x}}^T\hat{\mathbf{P}}\hat{\mathbf{x}} + \hat{\mathbf{u}}^T\hat{\mathbf{Q}}\hat{\mathbf{u}}$ is cost function and $\hat{\mathbf{P}} = \hat{\mathbf{I}}_P, \hat{\mathbf{Q}} = \hat{\mathbf{I}}_Q$ are unitary weight matrices.

Example 1. Necessary condition

Let us think over convergence of the algorithm for following dynamical nonlinear discrete-time system:

$$x_{k+1} = x_k^2 + u_k^3 \quad x_0 = 8 \tag{27}$$

The system can be transformed into state space dependent form.

$$x_{k+1} = \underbrace{x_k}_{A(x_k)} \cdot x_k + \underbrace{u_k^2}_{B(u_k)} \cdot u_k \tag{28}$$

The system is so simple that the optimal input sequence can be match out of hand. Control in equation (27) is in 3rd power and state in 2nd thus optimal control which satisfy $J = \hat{\mathbf{x}}^T\hat{\mathbf{x}} + \hat{\mathbf{u}}^T\hat{\mathbf{u}}$ is following dead-beat control.

$$u_{opt}(k) = \begin{cases} -x_0^{\frac{2}{3}} & k = 0 \\ 0 & k > 0 \end{cases} \tag{29}$$

Matrices A, B and system operators for time horizon N=2 are respectively equal.

$$A_{opt}(k) = \begin{cases} x_0 & k = 0 \\ 0 & k > 0 \end{cases}, \quad B_{opt}(k) = \begin{cases} x_0^{\frac{4}{3}} & k = 0 \\ 0 & k > 0 \end{cases}, \quad \hat{\mathbf{B}} = \begin{bmatrix} x_0^{\frac{4}{3}} & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{L}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{\mathbf{N}} = \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \tag{30}$$

and after calculating the new control $\mathrm{u_n}$ we have:

$$\hat{\mathbf{u}}_\mathbf{n} = -\left(\left(\hat{\mathbf{L}}\hat{\mathbf{B}}\right)^T \hat{\mathbf{L}}\hat{\mathbf{B}} + \hat{\mathbf{I}}\right)^{-1} \left(\hat{\mathbf{L}}\hat{\mathbf{B}}\right)^T \hat{\mathbf{N}}\mathbf{x}_0 = \begin{bmatrix} -\dfrac{x_0^{\frac{10}{3}}}{x_0^{\frac{8}{3}} + 1} & 0 \end{bmatrix}^T \neq \begin{bmatrix} -x_0^{\frac{2}{3}} & 0 \end{bmatrix}^T = \hat{\mathbf{u}}_{opt} \tag{31}$$

In general new control calculated on the basis of optimal control is not optimal control. On the basis of theorem 2 it can be said that the algorithm can be divergent. As well functional J calculated for optimal control u_{opt} is equal to J=16, the same functional calculated for new control $u_\mathbf{n}$ take value J≅16.6. Such results confirm divergence of the algorithm for analysed example.

Example 2. Additive decomposition

Let us modify the system from example 1 by introducing additional feedback term. In such case model of the system can be described following:

$$x_{k+1} = x_k^2 + 0.5x_k u_k + u_k^3 \quad x_0 = 8 \tag{32}$$

First of all let us assume that initial control sequence is equal to $\hat{\mathbf{u}} = -0.7 \cdot [1,1,1,...,1]$ and relative tolerance is less then 0.01. For following decomposition of the system (32)

$$x_{k+1} = \underbrace{(x_k + 0.05u_k)}_{A(x_k,u_k)} \cdot x_k + \underbrace{(0.45x_k + u_k^2)}_{B(x_k,u_k)} \cdot u_k \tag{33}$$

the algorithm is convergent and the solution is determined in 8 steps. On the other hand assuming following decomposition $x_{k+1} = \underbrace{(x_k)}_{A(x_k)} \cdot x_k + \underbrace{(0.5x_k + u_k^2)}_{B(x_k,u_k)} \cdot u_k$ the solution is determined in 26 steps.

What decide if the algorithm is convergent ? After analysis of several examples it has been noticed that conditional number in respect to inverse of matrix $\left(\left(\hat{\mathbf{L}}\hat{\mathbf{B}}\right)^T \hat{\mathbf{L}}\hat{\mathbf{B}} + \hat{\mathbf{I}}\right)$ correlate with convergence of the algorithm. For the first case eq. (33) maximal condition number among iterations is equal to 3.4e38. In the second case (convergence after 26 steps) the condition number is equal to 9.5e162. For divergent configurations maximal condition number is divergent to infinity. Although it is difficult to give an analytical rule how to decompose the nonlinear system into state dependent form. Step 3 of the algorithm from section 1 should be modified so as to minimize condition number in respect to inverse. It can be achieved not only by changes in coefficients of the decomposition into state space dependent form but also by changing initial control sequence of the system. The best initial sequence would be close to the optimal solution, good initial sequence has at least similar order of magnitude as the optimal solution.

Initial state have significant meaning to predictive control of nonlinear system. Considered above example with x_0=8 is quite difficult, because the system naturally goes to be unstable. At least a slight offset of given decomposition follows to divergence of the algorithm. Much more easier is the case with e.g. x_0=2 which is convergent for wide class of configurations. The convergence is assured e.g. for decompositions which satisfy following conditions:

$$x_{k+1} = \underbrace{\left(a_1 x_k + (1-b_1)u_k + (1-b_2)u_k^3 / x_k\right)}_{A(x_k,u_k)} \cdot x_k + \underbrace{\left((1-a_1)x_k^2 / u_k + b_1 x_k + b_2 u_k^2\right)}_{B(x_k,u_k)} \cdot u_k$$
$$a_1 \in \langle 0.8,1 \rangle,\ \mathrm{b}_1 \in \langle 0.48,0.5 \rangle,\ \mathrm{b}_2 \in \langle 0.7,1 \rangle \tag{34}$$

Assuming that that necessary condition from theorem 2 is satisfied the algorithm should be convergent to optimal solution but it could be only local optimum.

6 Conclusion

The main aim of the study is to answer 3 following questions, concerning to transformation from general nonlinear form into state space dependent form.

1) If chosen transformation method is suitable ?

Answer to this question follows from necessary condition for convergence. It can be deduced from theorem 3 and also from theorems 1-2, concerning the uniform convergence.

From practical point of view chosen method is suitable if:

a) Assumptions of theorem 3 are satisfied – the method is not divergent.

b) Condition number in respect to inverse is finite.

c) Nonlinearities are decomposed into matrices of the state space dependent form so as to the difference operators of the system would be as small as possible.

Although condition a) is the most important, it is also very difficult to settle. Condition b) is significant and easy to compute, which generally makes it very useful. The least significance has condition c). The algorithm can be convergent non-uniformly, what often take place especially for strong nonlinear systems.

2) If the convergence to the optimal solution is guaranteed ?

Within a framework of the study the sufficient condition for convergence have not been found. However there were proposed a few conditions (equations (7), (8)) for numerical detecting stability/instability of the algorithm.

3) If the number of iterations can be cut down ?

In many cases can be. Example 2 acknowledge that at least small modification of the decomposition results on the number of iterations. Recalling example 2: convergence for decomposition $A(k) = x_k$ is much more slower than for $A(k) = x_k + 0.05$ (26 vs. 8 steps). Nevertheless it should be mentioned that the convergence does not depend monotonically on the decomposition, the algorithm can be divergent for intermediate decompositions, e.g. for $A(k) = x_k + 0.056$ or $A(k) = x_k + 0.049$ the algorithm is divergent.

Much more difficult is question : how the number of iterations can be cut down? At the moment the optimization can be made rather using try and errors method than any analytical one. Moreover for different initial conditions optimal decomposition would be different, what really take place for each time sample.

References

[1] Kouvaritakis B., Cannon M., Rossiter J. A., “Non-linear model based predictive control”, *Int. J. Control*, Vol. **72**, No. 10, pp. 919-928, (1999).

[2] Lee Y. I., Kouvaritakis B., Cannon M., “Constrained receding horizon predictive control for nonlinear systems”, *Automatica*, Vol. **38**, No. 12, pp.2093-2102, (2002)

[3] Dutka, A., Ordys A., “The Optimal Non-linear Generalised Predictive Control by the Time-Varying Approximation”, *Proc. of 10th IEEE Int. Conf. MMAR.* Międzyzdroje. Poland. pp. 299-304 (2004).

[4] Grimble M. J., , Ordys A. W., “Non-linear Predictive Control for Manufacturing and Robotic Applications”, *Proc. of 7th IEEE Int. Conf. MMAR.* Miedzyzdroje, Poland, (2001).

[5] Ordys A. W., Grimble M. J., “Predictive control design for systems with the state dependent non-linearities”, *SIAM Conference on Control and its Applications*, San Diego, California, (2001).

[6] Dutka A. S., Ordys A. W., Grimble M. J., “Nonlinear Predictive Control of 2 dof helicopter model”, *IEEE CDC proceedings*, (2003).

[7] Mracek C. P., Cloutier J. R., “Control Designs for the nonlinear benchmark problem via the State-Dependent Ricati Equation method”, *International Journal of Robust and Nonlinear Control*, **8**, 401-433, (1998).

[8] Huang Y., Lu W-M., “Nonlinear Optimal Control: Alternatives to Hamilton-Jacobi Equation”, *Proceedings of the 35th IEEE Conference on Decision and Control*, 3942-3947, (1996).