Convergence of the Discrete-Time Nonlinear Model Predictive Control with Successive Time-Varying Linearization along Predicted Trajectories

P. Orlowski
West Pomeranian University of Technology Szczecin, Department of Control and Measurements, Sikorskiego 37, 70-313 Szczecin, Poland, phone: +48 914495409, e-mail: orzel@zut.edu.pl

crossref http://dx.doi.org/10.5755/j01.eee.113.7.607

Introduction

Significant attention has been given in the last decade to nonlinear model predictive control (NMPC). Currently there are some successful methods [1–4], and applications in nonlinear systems also with finite time horizon and reviews concerning NMPC methods [5–7].

Choose the cost function, signal constraints, the reference trajectory and the initial control trajectory $\hat{u}_0$.

Transform the non-linear model given in general form

$$x(k+1) = f(x(k), u(k), k)$$

into the time-varying state dependent form

$$x(k+1) = A(x(k), u(k), k)x(k) + B(x(k), u(k), k)u(k)$$

Increase iteration number $j = j + 1$

Calculate new control $\hat{u}_{(i)}$

Check stopping condition

$$\|\hat{u}_{(i)} - \hat{u}_{(i-1)}\| \leq \epsilon$$

Satisfied ?

Yes

Optimal control $\hat{u}_{opt} = \hat{u}_{(i)}$ found

No

Fig. 1. Algorithm of the time-varying linearization along predicted trajectory

The nonlinear system described by the discrete-time nonlinear state space model can be rearranged into the so-called state and control dependent linear form [8, 9]. The non-linear behaviour of the system is included in the state and control dependent matrices. If the trajectory prediction for the system may be obtained within the algorithm then one can pretend that the future behaviour is known during the prediction horizon [3]. Such a system can be treated as a linear time-varying (LTV) one. Most often the algorithm has the following common steps shown in fig. 1 [2, 10]. The control can be computed using arbitrary method for LTV systems. Also the technique presented in [3, 4] uses similar idea to [2], but with a different model representation and an optimisation technique.

The main aim of this paper is to analyse convergence of the NMPC successive model linearization method along predicted state and input trajectories. Particularly stopping and necessary convergence condition are discussed.

The algorithm from Fig. 1 refer only to one time step computation. Usually it is employed with receding horizon, where the algorithm must be repeated for successive time steps $k_0 = k_0 + 1$.

Model description

General discrete-time (DT), time-varying nonlinear model is assumed in the following form

$$x(k+1) = f(x(k), u(k), k).$$

The non-linear system can be transformed into following discrete-time, time-varying state-dependent form

$$x(k+1) = A(x(k), u(k), k)x(k) + B(x(k), u(k), k)u(k),$$

where state and input dependent matrices are calculated for given initial condition $x_0$ and control trajectory $u(k)$ at each time instant.

Then, using the past trajectory, matrices $A(k) = A(x(k), u(k), k)$, $B(k) = B(x(k), u(k), k)$ may be calculated for the subsequent points of the trajectory and the nonlinear system (1) is approximated by the LTV model with matrices $A(k)$, $B(k)$. 

DT-LTV system is given in the state space form
\[ x(k+1) = A(k)x(k) + B(k)u(k), \]
where \( k = k_0, k_0 + 1, \ldots, k_0 + N - 1 \), \( A(k) \in \mathbb{R}^{n \times n} \), \( B(k) \in \mathbb{R}^{n \times m} \) and \( N \) is the prediction horizon.

It can be equivalently defined using evolution operators or, in the considered finite horizon case, also by following block matrix operators \( \hat{L}, \hat{B} \):
\[
\hat{L} = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
\phi_{k+1} & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
\phi_{k+N-1} & \cdots & \phi_{k+N-1} & 1
\end{bmatrix},
\]
\[
\hat{B} = \begin{bmatrix}
B(k_0) & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & B(k+N-1)
\end{bmatrix},
\]
where \( \phi_k = A(k)A(k-1)\ldots A(i) \). For vectors \( \mathbf{x}, \mathbf{u} \) we use the following block vector notation, i.e.
\[
\mathbf{x} = \begin{bmatrix}
x^T(k_0 + 1) \\
\vdots \\
x^T(k_0 + N)
\end{bmatrix}.
\]

It follows that the mathematical model can be rewritten in the final form as
\[
\mathbf{x} = \hat{L}\hat{B}\mathbf{u} + \hat{N}\mathbf{x}_0.
\]

We assume that at each time instant the system can be analyzed as starting from time sample equal to zero with a current initial condition \( \mathbf{x}_0 = x(k_0) \) up to \( N \) steps into the future (prediction horizon).

The operator \( \hat{L}\hat{B} \) is a compact and Hilbert-Schmidt one from \( L_2 \) into \( L_2 \) and boundedly maps signals \( u(k) \in L = L_2 \{k_0, k_0 + N - 1\} \) into signals \( x \in X \).

For simulation purposes we employ cost function in following form
\[
J = (\mathbf{x} - \mathbf{x}_{\text{ref}})^T \mathbf{P} (\mathbf{x} - \mathbf{x}_{\text{ref}}) + \mathbf{u}^T \mathbf{Q} \mathbf{u},
\]
where \( \mathbf{P} \in \mathbb{R}^{(nN) \times (nN)}, \mathbf{Q} \in \mathbb{R}^{(MN) \times (MN)} \) are diagonal weighting operators, constructed with weighting matrices \( P(k) \in \mathbb{R}^{n \times n}, k = 1, 2, \ldots, N \), \( Q(k) \in \mathbb{R}^{m \times m}, k = 0, 1, \ldots, N - 1 \).

**Convergence of the algorithm**

**Definition 1.** The algorithm from Fig. 1 is convergent if there exists a limiting control sequence \( \mathbf{u}_{\text{got}} \) such that for any arbitrarily small positive number \( \varepsilon > 0 \), there is a large integer \( I \) such that for all \( i \geq I \), \( \| \mathbf{u}_{i+1} - \mathbf{u}_i \| \leq \varepsilon \). The algorithm that is not convergent is said to be divergent.

The algorithm converges both for local or global optimal solutions. Divergent algorithm cannot satisfy a stopping condition usually given by following absolute tolerance condition:
\[
\| \mathbf{u}_{i+1} - \mathbf{u}_{i} \| \leq \varepsilon
\]
for arbitrarily small \( \varepsilon \).

**Definition 2.** Let the state trajectory deviation norm from the reference trajectory for any given time horizon be given by \( \| \mathbf{x}(i+1) - \mathbf{x}_{\text{ref}} \| \) for the \( i \)-th iteration of the algorithm and \( \| \mathbf{x}_{i+1} - \mathbf{x}_{\text{ref}} \| \) for the next \( i+1 \) iteration. The state error rate is denoted by R and defined as the following relation
\[
R(i) := \frac{\| \mathbf{x}(i+1) - \mathbf{x}_{\text{ref}} \|}{\| \mathbf{x}(i) - \mathbf{x}_{\text{ref}} \|}
\]
or for \( \mathbf{x}_{\text{ref}} = 0 \), \( R_0(i) := \frac{\| \mathbf{x}(i+1) \|}{\| \mathbf{x}(i) \|} \).

**Theorem 1.** The state trajectory in the consecutive iteration of the algorithm from Fig. 1, calculated for any nonlinear system transformed into the state-dependent LTV form under the assumption \( \| \mathbf{L}(i)\hat{A} \| < 1 \) can be calculated from the following equation
\[
\mathbf{x}_{i+1} = (\mathbf{I} - \mathbf{L}(i)\hat{A})(i)\mathbf{x}(i) + \mathbf{L}(i)(\hat{B}(i)\mathbf{u}(i) - \hat{B}(i)\mathbf{u}(i)),
\]
where
\[
\hat{A} = \mathbf{A}_{i+(1)} - \hat{A}(i)(\hat{A}(i)\mathbf{x}(i) + \mathbf{B}(i)\mathbf{u}(i) + \hat{A}(i))
\]
denotes the difference system operator and \( \mathbf{x}(i), \mathbf{u}(i) \).

\( \hat{B}(i), \mathbf{l}(i) \) denotes respectively: the state, the input trajectory, the input operator and the system operator in the \( i \)-th iteration of the algorithm, \( \mathbf{x}_{\text{ref}} \) denotes the reference trajectory.

**Proof.** The proof follows from similar results for perturbed systems [10] with the difference that the term \( \hat{A}(i) \) represents deviation of the linearized time-varying system matrix. The deviation corresponds to corrections of the state and input trajectories which are applied in the consecutive iteration of the NMPC algorithm. Previously, this methodology was used for uncertain systems for which \( \hat{A}(i) \) represented model uncertainty. The system in the \( i+1 \) iteration can be treated as the perturbed system from the \( i \)-th iteration, and the system in the \( i+1 \)-th can be written in following form
\[
\mathbf{x}_{i+1} = \mathbf{x}(i) + \mathbf{L}(i)\hat{A}(i)\mathbf{x}(i) + \mathbf{L}(i)(\hat{B}(i)\mathbf{u}(i) - \hat{B}(i)\mathbf{u}(i)),
\]
or equivalently
\[
(\mathbf{I} - \mathbf{L}(i)\hat{A}(i))\mathbf{x}(i+1) = \mathbf{x}(i) + \mathbf{L}(i)(\hat{B}(i)\mathbf{u}(i) - \hat{B}(i)\mathbf{u}(i)).
\]

To derive the trajectory \( \mathbf{x}_{i+1} \), the term \( \mathbf{I} - \mathbf{L}(i)\hat{A}(i) \) has to be invertible. Under the sufficient condition that \( \| \mathbf{L}(i)\hat{A}(i) \| < 1 \), the term \( \mathbf{I} - \mathbf{L}(i)\hat{A}(i) \) becomes invertible. Calculating the left side inverse of above equation leads to eq. (9).

**Corollary 1.** The state error rate of the algorithm from Fig. 1 can be evaluated from following expression
\[
R(i) = \frac{\| \mathbf{x}(i+1) - \mathbf{x}_{\text{ref}} \|}{\| \mathbf{x}(i) - \mathbf{x}_{\text{ref}} \|}.
\]

**Corollary 2.** The state error rate of the algorithm for \( \mathbf{x}_{\text{ref}} = 0 \) can be evaluated from expression:
The algorithm is convergent to the optimal control trajectory \( \hat{u}_{opt} \) if and only if the following limit exists
\[
\lim_{i \to \infty} \hat{u}(i) = \hat{u}_{opt} \quad \text{or equivalently}
\]
\[
\lim_{i \to \infty} \hat{u}(i) - \hat{u}_{opt} = 0
\]
where \( \hat{u}(i) \) is the input trajectory in the \( i \)-th iteration of the algorithm.

Let us define iterative control differences vector field in following way
\[
V_{gfs0}(\hat{u}(i)) = \hat{u}(i+1) - \hat{u}(i).
\]
It means that \( \hat{u}_{opt} \) must be stationary point of the field \( V_{gfs0} \). Taking account eqs. (16) and (21) it may be written
\[
\lim_{i \to \infty} \left( V_{gfs0}(\hat{u}(i)) - \hat{u}(i) \right) = 0.
\]

The algorithm is monotonically convergent if \( R \leq 1 \). The closer the coefficient to zero, the faster the rate of convergence. However, when approaching the optimal solution we get \( R \to 1 \). For linear systems we have \( R = 1 \) since the solution is calculated in the first iteration of the algorithm. For values \( R > 1 \) the algorithm can be divergent. In some cases \( R \) can oscillate above and below 1. In such case the algorithm is convergent if for the consecutive iterations \( |R(i) - 1| \) is decreasing function for \( i \geq I \), where \( I \) is a large finite integer. The convergence to the optimal solution \( J \to J_{opt} \) is always connected with approaching zero by nonlinearity differences \( \Delta A(i) \).

What could be done to ensure that \( R \) is near 1? There are two conditions which have to be satisfied. The operator product \( \hat{L} \Lambda_{A} \) should approach zero or equivalently \( \| L \Lambda_{A} \| \to 0 \). From the definition of operator \( \hat{L} \), it follows that \( \| L \| \geq 1 \). On the other hand the norm of \( \Lambda_{A} \) can be arbitrary small. Its actual value depends on the nonlinearity of the system i.e. the nonlinearity degree and the method which decomposes nonlinearity into matrices \( A \) and \( B \) (step 2 of the algorithm from Fig. 1). For linear systems the matrix \( A \) does not depend on the input and state and hence the norm is equal to zero. The assumption \( A = 0 \) results in \( \| \Lambda_{A} \| = 0 \). However, it also increases the second difference \( J_{opt}(i) \left( B(i+1) \hat{u}_{opt}(i+1) - B(i) \hat{u}_{opt}(i) \right) \), especially, for small values of input \( \hat{a} \). In most cases it results in the divergence of the algorithm, similarly as does the assumption \( B = 0 \). Thus the difference operator norms
The cost function is assumed in following form

\[
J = \mathbf{A}^T \hat{\mathbf{L}}(i) \mathbf{A}^{opt}(k) + \mathbf{B}^T(i) \hat{\mathbf{L}}(i) \mathbf{B}^{opt}(k),
\]

so that the norms of difference operators of the system

\[
\begin{bmatrix}
\hat{\mathbf{A}}_i(i) \\
\hat{\mathbf{L}}(i) \\
\hat{\mathbf{B}}(i)
\end{bmatrix}
\] and

\[
\begin{bmatrix}
\mathbf{A}^{opt}(k) \\
\mathbf{B}^{opt}(k)
\end{bmatrix}
\]

are possibly small.

**Numerical example – convergence necessary condition**

It is assumed that the control is calculated iteratively using cost function (8) with \[\mathbf{x}_{ref} = \mathbf{0}\], from the formula

\[
\hat{\mathbf{u}}(i+1) = -\left(\hat{\mathbf{L}}(i) \hat{\mathbf{B}}(i) \mathbf{P}(i+1) \hat{\mathbf{B}}(i) + \hat{\mathbf{Q}}\right)^{-1} \hat{\mathbf{L}}(i) \hat{\mathbf{B}}(i) \mathbf{P}(i+1) \hat{\mathbf{N}}(i)x_0. \tag{24}
\]

The system can be transformed into the state space dependent form

\[
x_{k+1} = A(x_k) x_k + B(u_k).
\]

The cost function is assumed in following form

\[
J = \mathbf{A}^T \hat{\mathbf{L}}(i) \mathbf{A}(x_k) \mathbf{x}_k + \mathbf{B}^T(i) \hat{\mathbf{L}}(i) \mathbf{B}(u_k).
\]

The matrices \(A, B\) and the system operators for the time horizon \(N=2\) are as follows:

\[
A^{opt}(k) = \begin{bmatrix}
x_0 & 0 \\
k & 0
\end{bmatrix},
\]

\[
B^{opt}(k) = \begin{bmatrix}
0 & 0 \\
\frac{1}{2} & 0
\end{bmatrix},
\]

\[
\hat{\mathbf{B}}(i) = \begin{bmatrix}
\frac{1}{2} & 0 \\
0 & 0
\end{bmatrix},
\]

\[
\hat{\mathbf{L}}(i) = \begin{bmatrix}
1 & 0 \\
x_1 & 1
\end{bmatrix},
\]

\[
\hat{\mathbf{N}}(i) = \begin{bmatrix}
x_0 \\
0
\end{bmatrix}.
\]

The new control \(\hat{\mathbf{u}}(2)\) have following form

\[
\hat{\mathbf{u}}(2) = \begin{bmatrix}
-\frac{3}{2} \\
0
\end{bmatrix}^T = \hat{\mathbf{u}}(1) = \hat{\mathbf{u}}_{opt}.
\]

**Conclusions**

Methods proposed in the paper concerns the transformation method from a general nonlinear form into the state space dependent form. The suitability of the chosen transformation method follows from the necessary condition for convergence, what can be deduced from theorem 2 and also from theorem 1, concerning the uniform convergence.

From a practical point of view, the chosen method is suitable if: assumption of theorem 2 is satisfied – the method is not divergent and nonlinearities are decomposed into two additive terms – state and input dependent matrices of the state space dependent form so as to the norms of difference operators of the system

\[
\begin{bmatrix}
\hat{\mathbf{A}}_i(i) \\
\hat{\mathbf{L}}(i) \\
\hat{\mathbf{B}}(i)
\end{bmatrix}
\] and

\[
\begin{bmatrix}
\mathbf{A}^{opt}(k) \\
\mathbf{B}^{opt}(k)
\end{bmatrix}
\]

are possibly small.

**References**

Model predictive control techniques for nonlinear systems very often take advantage of nonlinear model linearization. The model can be linearized once or repeatedly. In the paper the second type of method is considered: successive model linearization along predicted state and input trajectories. The nonlinear behaviour is represented by recurrent set of linear time-varying models. Solution of such optimal non-linear model predictive control problem is mostly obtained in an iterative way where the most important step is the successive system linearization along predicted trajectory. The main aim of the paper is to analyse convergence of the considered NMPC method, discuss problems concerning necessary condition for the convergence and prove proposed solutions. Ill. 1, bibl. 11 (in English; abstracts in English and Lithuanian).


Netiesinėse sistemos nusėjamosios kontrolės modelis turi daugiau pranašumų, palyginti su netiesinio modelio tiesiškumu. Toks modelis gali būti „tiesinamas“ vieną kartą arba nuolat. Analizuojama pastaroji modelio „tiesinimo“ situacija. II. 1, bibl. 11 (anglų kalba; santraukos anglų ir lietuvių k.).