

# NONLINEAR INTO STATE AND INPUT DEPENDENT FORM MODEL DECOMPOSITION

## *Applications to Discrete-time Model Predictive Control with Successive Time-varying Linearization along Predicted Trajectories*

Przemyslaw Orłowski

*Institute of Control Engineering, West Pomeranian University of Technology, Szczecin, Poland*

**Keywords:** Non-linear systems, Successive linearization, Predictive control, Optimal control, Discrete time systems.

**Abstract:** Linearization techniques are well known tools that can transform nonlinear models into linear models. In the paper we employ a successive model linearization along predicted state and input trajectories resulting in linear time-varying model. The nonlinear behaviour is represented in each time sample by recurrent set of linear time-varying models. Solution of the optimal non-linear model predictive control problem is obtained in an iterative way where the most important step is the linearization along predicted trajectory. The main aim of this paper is to analyse how the nonlinear system should be transformed into linear one to ensure possibly fast solution of the model predictive control problem based on the successive linearization method.

## 1 INTRODUCTION

Model predictive control (MPC) is attractive control strategy, which have 3 common properties (Camacho et. al. 2004): explicit use of a model to predict the output at future time instants, calculation of a control trajectory minimizing an objective function and receding horizon (moving horizon) strategy. MPC issues for linear systems including stability are well known (Camacho et. al., 2004), (Morari et. al. 1999), (Tatjewski, 2007), (Mayne et. al., 2000) also (Qin et. al. 2003), (Magni et. al. 1999), including fast algorithms (Blachuta, 1999) and discrete-time system with delays (Kowalczuk et. al. 2005). Many real systems are inherently nonlinear. Due to higher product quality specifications, some important environmental and economical reasons linear models are often inadequate to describe the system properties. Computing the optimal control trajectory directly for nonlinear model is difficult, non-convex optimization problem. Generally there is no guarantee that the computed solution is global optimal solution. Moreover it is difficult to prove global stability of the system using directly the nonlinear model for control synthesis. In practise some transformations and simplifications are applied to the nonlinear model in order to prove stability,

and also to take advantages of theory for linear systems.

Among some existing approaches in nonlinear model predictive control in the paper we consider successive model linearization along predicted state and input trajectories with recurrent linear time-varying (LTV) model. A large class of these methods uses a common algorithm, i.e. (Kouvartiakis et. al., 1999) employ an optimal control trajectory calculated at the previous time instant of the control algorithm for NMPC. (Lee et. al., 2002) use a similar methodology and employ a linearization at points of the seed trajectory for the discrete-time model of the system. Also the technique presented in (Dutka et. al., 2004), (Ordys et. al., 2001), (Mracek et. al., 1998), (Grimble et. al., 2001), (Dutka et. al., 2003) uses similar idea to (Kouvartiakis et. al., 1999), (Lee et. al., 2002) but with a different model representation and an optimisation technique. Similar approach for the construction of an explicit nonlinear control law approximating nonlinear constrained finite-time optimal control using approximate mapping of a general nonlinear system into a set of piecewise affine systems is presented in (Ulbig et. al., 2007). The main aim of this paper is to analyse how to linearize (decompose) nonlinear system into linear one for using with the successive model linearization method along predicted state and input trajectories.

The main difficulty is to find proper transformation method, which ensure fast computation of stable and optimal solution for nonlinear control problem.

## 2 SYSTEM DESCRIPTION

Let us assume general discrete-time, time-varying nonlinear model in the following form:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k) \quad (1)$$

The nonlinear system can be transformed into following discrete-time, time-varying state-dependent form:

$$\mathbf{x}(k+1) = \mathbf{A}(\mathbf{x}(k), \mathbf{u}(k), k)\mathbf{x}(k) + \mathbf{B}(\mathbf{x}(k), \mathbf{u}(k), k)\mathbf{u}(k) \quad (2)$$

where  $\mathbf{A}(\mathbf{x}(k), \mathbf{u}(k), k)$ ,  $\mathbf{B}(\mathbf{x}(k), \mathbf{u}(k), k)$  are state and input dependent matrices calculated for given initial condition  $\mathbf{x}_0$  and control trajectory  $\mathbf{u}(k)$  at each time instant.

Then, using the past input and state trajectories, 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 of the trajectories and the nonlinear system (1) is approximated by the LTV model with matrices  $\mathbf{A}(k)$ ,  $\mathbf{B}(k)$ . Discrete-time LTV system is given in the state space form:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (3)$$

where

$\mathbf{A}(k) \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B}(k) \in \mathbb{R}^{n \times m}$ ,  $k = k_0, \dots, k_0 + N - 1$  and  $N$  is the prediction horizon.

Linear time-varying discrete-time system can be equivalently defined using evolution operators or in the finite horizon case, also by following block matrix operators  $\hat{\mathbf{L}}, \hat{\mathbf{N}}, \hat{\mathbf{B}}$ :

$$\hat{\mathbf{L}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \phi_{k_0+1}^{k_0+1} & \mathbf{I} & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{I} & \mathbf{0} \\ \phi_{k_0+1}^{k_0+N-1} & \cdots & \phi_{k_0+1}^{k_0+N-1} & \mathbf{I} \end{bmatrix}, \hat{\mathbf{N}} = \begin{bmatrix} \phi_{k_0}^{k_0} \\ \vdots \\ \phi_{k_0}^{k_0+N-1} \end{bmatrix} \quad (4)$$

$$\hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B}(k_0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}(k_0 + N - 1) \end{bmatrix} \quad (5)$$

where  $\phi_i^k = \mathbf{A}(k)\mathbf{A}(k-1)\dots\mathbf{A}(i)$ . For state and input trajectories  $\hat{\mathbf{x}}, \hat{\mathbf{u}}$  we use the following block column vector notation, i.e.

$$\hat{\mathbf{x}} = [\mathbf{x}^T(k_0+1) \cdots \mathbf{x}^T(k_0+N)]^T \quad (6)$$

$$\hat{\mathbf{u}} = [\mathbf{u}^T(k_0) \cdots \mathbf{u}^T(k_0+N-1)]^T \quad (7)$$

It follows that the mathematical model can be rewritten in the final form as

$$\hat{\mathbf{x}} = \hat{\mathbf{L}}\hat{\mathbf{B}}\hat{\mathbf{u}} + \hat{\mathbf{N}}\mathbf{x}_0 \quad (8)$$

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  $\mathbf{x}_0 = \mathbf{x}(k_0)$  up to  $N$  steps into the future (prediction horizon).

The operator  $\hat{\mathbf{L}}\hat{\mathbf{B}}$  is a compact and Hilbert-Schmidt one from  $l_2$  into  $l_2$  and boundedly maps signals  $\mathbf{u}(k) \in \mathcal{L} = l_2[k_0, k_0 + N - 1]$  into signals  $x \in \mathcal{X}$ .

For simulation purposes we employ cost function in the following form:

$$J = (\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref})^T \hat{\mathbf{P}}(\hat{\mathbf{x}} - \hat{\mathbf{x}}_{ref}) + \hat{\mathbf{u}}^T \hat{\mathbf{Q}}\hat{\mathbf{u}} \quad (9)$$

where  $\hat{\mathbf{P}} \in \mathbb{R}^{(nN) \times (nN)}$ ,  $\hat{\mathbf{Q}} \in \mathbb{R}^{(mN) \times (mN)}$  are weighting operators, constructed with weighting matrices  $\mathbf{P}(k) \in \mathbb{R}^{n \times n}$ ,  $k = 1 \dots N$ ,  $\mathbf{Q}(k) \in \mathbb{R}^{m \times m}$ ,  $k = 0 \dots N - 1$ , respectively usually given in following block matrix form:

$$\hat{\mathbf{P}} = \begin{bmatrix} \mathbf{P}(1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}(N) \end{bmatrix}, \hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}(N-1) \end{bmatrix}$$

Usually weighting matrices are time-invariant with the exception of  $\mathbf{P}(N)$  which represents the terminal cost. Equivalently the cost function can be rewritten in the following form:

$$J = \sum_{k=1}^N \begin{pmatrix} \mathbf{x}(k+k_0) \\ -\mathbf{x}_{ref}(k+k_0) \end{pmatrix}^T \mathbf{P}(k) \begin{pmatrix} \mathbf{x}(k+k_0) \\ -\mathbf{x}_{ref}(k+k_0) \end{pmatrix} + \sum_{k=0}^{N-1} \mathbf{u}^T(k+k_0)\mathbf{Q}(k)\mathbf{u}(k+k_0) \quad (10)$$

where the term

$$\begin{pmatrix} \mathbf{x}(N+k_0) \\ -\mathbf{x}_{ref}(N+k_0) \end{pmatrix}^T \mathbf{P}(N) \begin{pmatrix} \mathbf{x}(N+k_0) \\ -\mathbf{x}_{ref}(N+k_0) \end{pmatrix}$$

for  $k=N$  in the first sum of (10) is the terminal cost.

## 3 PROBLEM DESCRIPTION

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 (Mracek et. al., 1998), (Huang et. al., 1996). 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 (Dutka et. al., 2004). Such a system can be treated as a linear time-varying (LTV) one. Most often the algorithm, shown on fig. 1 has common steps (Kouvartiakis et. al., 1999), (Orlowski, 2005).

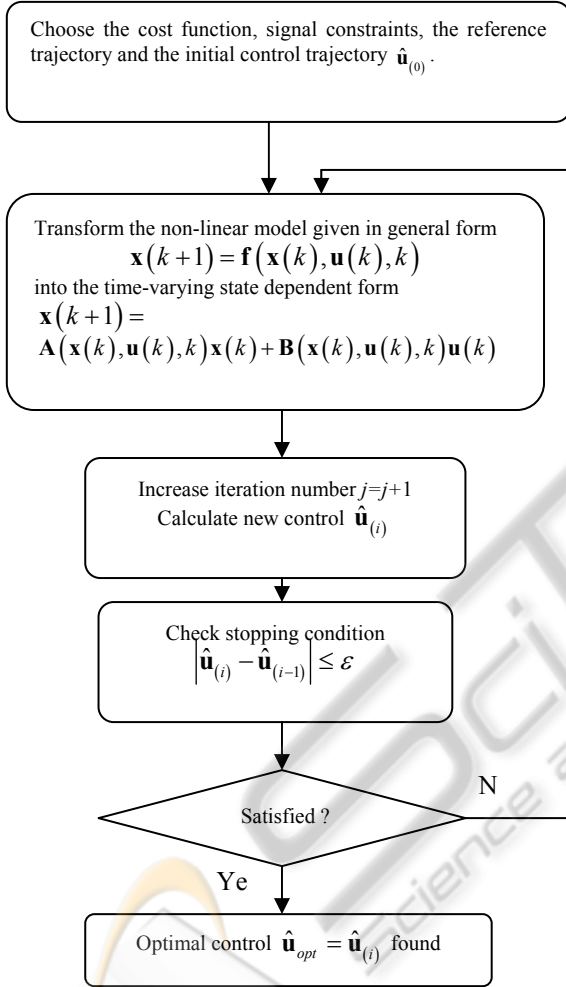


Figure1: Algorithm of the time-varying linearization along predicted trajectory.

In general there no restrictions to the cost function. For simulation purposes we employ cost function given by eq. (9). However in practise the method can be also used with different frequently used in MPC cost functions and stabilizing conditions, e.g.: terminal cost function, terminal equality constraint, terminal constraint set. It is only required to define an MPC problem for the LTV system.

The second important problem is choosing initial control trajectory. The simplest choice could be step control signal with amplitude from normal operating range for the control. Another possibility is to use at the beginning a few initial control trajectories and choose the one which results in the smallest cost function. The trajectory is required *only for linearization purposes* and *only in the first iteration* of the algorithm *for the first time step*. For the consecutive time steps on receding horizon it may be assumed from previous control predictions.

**Definition 1.** The algorithm from fig. 1 is convergent if there exists a limiting control sequence  $\hat{\mathbf{u}}_{opt}$  such that for any arbitrarily small positive number  $\varepsilon > 0$ , there is a large integer  $I$  such that for all  $i \geq I$ ,  $|\hat{\mathbf{u}}_{(i)} - \hat{\mathbf{u}}_{opt}| \leq \varepsilon$ . 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:

$$|\hat{\mathbf{u}}_{(i)} - \hat{\mathbf{u}}_{(i-1)}| \leq \varepsilon \quad (11)$$

for arbitrarily small  $\varepsilon$ .

The control can be computed using arbitrary method for LTV systems, including algorithms with signal constraints. The algorithm from fig. 1 refer only to one time step computation. Usually it is employed with receding horizon. The algorithm must be repeated for successive time steps  $k_0 = k_0 + 1$ .

## 4 NONLINEAR SYSTEM DECOMPOSITION

To transform of the non-linear model (1) into the time-varying state dependent form given by eq. (2) one needs to decompose nonlinear function  $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k)$  into 2 factors corresponding to state and input matrices such that:  $\mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k)$ .

For example, let us assume nonlinear function:

$$f(x(k), u(k)) = x(k) \sin(x(k)) + u(k) \arctan(u(k))$$

Transformation into state and input dependent form can be easily done by simple expansion terms dependent on state and input only, i.e.:

$$\mathbf{A}(k) = \sin(x(k)), \quad \mathbf{B}(k) = \arctan(u(k))$$

More difficult problem is decomposition of a system consisting coupled input-state terms. Assume for example function  $f(x(k), u(k)) = x(k)u(k)$ . One

of possible decompositions is to divide the function into following 2 additive terms:

$$f(x(k), u(k)) = \alpha x(k)u(k) + (1-\alpha)x(k)u(k)$$

where:

$$\mathbf{A}(k) = \alpha u(k), \quad \mathbf{B}(k) = (1-\alpha)x(k)$$

In general we propose following method which allow to decompose arbitrary nonlinear function  $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k)$  into series of  $M$  additive components. Using the simplified notation  $\mathbf{f}_i = \mathbf{f}_i(\mathbf{x}(k), \mathbf{u}(k), k)$  for a fixed input trajectory and initial conditions we have

$$\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k) = \sum_{i=1}^M \mathbf{f}_i(\mathbf{x}(k), \mathbf{u}(k), k) = \sum_{i=1}^M \mathbf{f}_i \quad (12)$$

Every system (1) can be decomposed into the state dependent form (2). In general, this decomposition takes the following form:

$$\mathbf{x}(k+1) = \sum_{i=1}^M \mathbf{f}_i = \sum_{j=1}^n \left( \sum_{i=1}^M \alpha_{i,j} \mathbf{f}_i \right) + \sum_{j=1}^m \left( \sum_{i=1}^M \beta_{i,j} \mathbf{f}_i \right) \quad (13)$$

$$\mathbf{x}(k+1) = \sum_{i=1}^M \alpha_{i,1} \mathbf{f}_i + \dots + \sum_{i=1}^M \alpha_{i,n} \mathbf{f}_i + \sum_{i=1}^M \beta_{i,1} \mathbf{f}_i + \dots + \sum_{i=1}^M \beta_{i,m} \mathbf{f}_i \quad (14)$$

What can be arranged into following vector-matrix state and input dependent form:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{a}_1 x_1 + \dots + \mathbf{a}_n x_n + \mathbf{b}_1 u_1 + \dots + \mathbf{b}_m u_m \\ &= [\mathbf{a}_1 \dots \mathbf{a}_n] [x_1 \dots x_n]^T + [\mathbf{b}_1 \dots \mathbf{b}_m] [u_1 \dots u_m]^T \end{aligned} \quad (15)$$

$$= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

where

$$\mathbf{a}_j x_j = \sum_{i=1}^M \alpha_{i,j} \mathbf{f}_i, \quad (16)$$

$$\mathbf{b}_j u_j = \sum_{i=1}^M \beta_{i,j} \mathbf{f}_i, \quad (17)$$

$$\forall_i \sum_{j=1}^n \alpha_{i,j} + \sum_{j=1}^m \beta_{i,j} = 1 \quad (18)$$

The component column vectors of matrices  $\mathbf{A}(k)$  and  $\mathbf{B}(k)$  can be determined under assumption that the

following limits  $\forall_j \lim_{x_j \rightarrow 0} \frac{\mathbf{a}_j x_j}{x_j}, \forall_j \lim_{u_j \rightarrow 0} \frac{\mathbf{b}_j u_j}{u_j}$  exist and

are finite. These vectors are given by expressions

$$\mathbf{a}_j = \begin{cases} \frac{\mathbf{a}_j x_j}{x_j} & x_j \neq 0 \\ \lim_{x_j \rightarrow 0} \frac{\mathbf{a}_j x_j}{x_j} & x_j = 0 \end{cases} \quad (19)$$

$$\mathbf{b}_j = \begin{cases} \frac{\mathbf{b}_j u_j}{u_j} & u_j \neq 0 \\ \lim_{u_j \rightarrow 0} \frac{\mathbf{b}_j u_j}{u_j} & u_j = 0 \end{cases} \quad (20)$$

where

$\mathbf{A}(k) = [\mathbf{a}_1 \dots \mathbf{a}_n], \mathbf{B}(k) = [\mathbf{b}_1 \dots \mathbf{b}_m]$ ,  $n$  - order,  $m$  - number of inputs,  $\mathbf{a}_j, \mathbf{b}_j$  - column vectors with  $n$  rows

Let us assume that function  $\mathbf{f}(\mathbf{x}, \mathbf{u}, k)$  can be decomposed into the following four additive terms:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, k) = \mathbf{f}_1(\mathbf{x}, k) + \mathbf{f}_2(\mathbf{x}, \mathbf{u}, k) + \mathbf{f}_3(\mathbf{u}, k) + \mathbf{f}_4(k) \quad (21)$$

The vector functions  $\mathbf{f}$  must be continuous and the following limits calculated in respect to all coordinates of  $\mathbf{f}$  and  $\mathbf{x}/\mathbf{u}$  must be finite:

$$\lim_{\mathbf{x} \rightarrow 0} \frac{\mathbf{f}_1}{\mathbf{x}}, \lim_{\mathbf{u} \rightarrow 0} \frac{\mathbf{f}_3}{\mathbf{u}} \quad (22)$$

and either

$$\lim_{\mathbf{x} \rightarrow 0} \frac{\mathbf{f}_2}{\mathbf{x}}, \text{ or/and } \lim_{\mathbf{u} \rightarrow 0} \frac{\mathbf{f}_2}{\mathbf{u}} \quad (23)$$

where

$$\mathbf{f}_1 = \begin{bmatrix} f_{1[1]} \\ \vdots \\ f_{1[R]} \end{bmatrix}, \quad \lim_{\mathbf{x} \rightarrow 0} \frac{\mathbf{f}_1}{\mathbf{x}} \sim \begin{bmatrix} \lim_{x_1 \rightarrow 0} \frac{f_{1[1]}}{x_1} & \dots & \lim_{x_R \rightarrow 0} \frac{f_{1[1]}}{x_R} \\ \vdots & \ddots & \vdots \\ \lim_{x_1 \rightarrow 0} \frac{f_{1[R]}}{x_1} & \dots & \lim_{x_R \rightarrow 0} \frac{f_{1[R]}}{x_R} \end{bmatrix}$$

And the limit is finite if and only if all elements in above matrix are finite.

Norms of matrices  $\mathbf{A}$ ,  $\mathbf{B}$  should approach neither zero nor infinity. The best performance is achieved if the norms of matrices  $\mathbf{A}$ ,  $\mathbf{B}$  have similar order of magnitudes.

Although the convergence of the algorithm from fig. 1 for a given decomposition cannot be proved for general nonlinear systems stability for linearized ones follows directly from the applied computation method for control. The conversion from a nonlinear into LTV system can be successfully applied to all systems for which the optimal nonlinear control lies in the neighbourhood of the optimal control for the linearized LTV system.

## 5 NUMERICAL EXAMPLE

In the example algorithm from fig. 1 is combined with formula (24), where  $\mathbf{x}_0$  is current initial

condition  $\mathbf{x}_0 = \mathbf{x}(k_0)$  and  $\hat{\mathbf{P}}, \hat{\mathbf{Q}}$  are weighting matrices. Control is calculated iteratively using cost function (9) with  $\hat{\mathbf{x}}_{ref} = \mathbf{0}$ , from following formula:

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

We assume following model for the nonlinear system:

$$x_{k+1} = x_k^2 + 0.5x_k^2 u_k + u_k^3 \quad (25)$$

The initial control trajectory is equal to  $\hat{\mathbf{u}}_{(0)} = -0.5[1,1,1]$ , the absolute tolerance, defined by (11)  $\varepsilon = 0.001$  and the weighting matrices are unitary  $\hat{\mathbf{P}} = \hat{\mathbf{I}}_p, \hat{\mathbf{Q}} = \hat{\mathbf{I}}_q$ . The system (25) can be decomposed into two following state and input dependent parts:

$$x_{k+1} = \underbrace{(x_k + \alpha x_k u_k)}_{A(x_k, u_k)} \cdot x_k + \underbrace{((0.5 - \alpha)x_k^2 + u_k^2)}_{B(x_k, u_k)} \cdot u_k \quad (26)$$

The decomposition is dependent on parameter  $\alpha$ . Equation (26) is equivalent to (25) for arbitrary values of  $\alpha$ , although convergence of the algorithm from fig. 1 is analysed for  $\alpha \in [-0.5, 0.5]$ .

Figure 2 shows number of iterations  $\eta$  required to converge to optimal control solution for given initial state  $x_0 \in (0, 8]$  and decomposition parameter  $\alpha \in [-0.5, 0.5]$ . To improve readability of the figure 2 it is also assumed that  $\eta \leq 100$ . Value  $\eta = 100$  corresponds to a divergent solutions or solutions with that require more than 100 iterations. It may be concluded from fig. 2 that convergence of the algorithm from fig. 1 is dependent both on the initial state and the decomposition. Usually it is required for the algorithm to be convergent and possibly fast for all initial conditions from given range. To ensure fast convergence (the minimal number of iterations) for e.g.  $x_0 = 8$  parameter  $\alpha$  should be chosen in the range  $\alpha \in [-0.5, 0]$ , whereas for  $x_0 = 1.4$  the smallest number of iterations is for  $\alpha \in [-3, -1.5]$ . For  $x_0 < 1$  the algorithm is fast convergent for all  $\alpha$ .

It should be underlined that the convergence/divergence is a property of: the system, the initial condition, the decomposition and the initial control trajectory. First of all it is assumed that the system is controllable and observable and the state is reachable from arbitrary initial state  $x_0$ . Although changes in each of three above factors may be effective to achieve convergence of the algorithm, the easiest way to improve the method or fasten the algorithm is to change the decomposition. Convergence of the algorithm is strongly connected with the conditional number  $r_{cond}$  of the inverse of

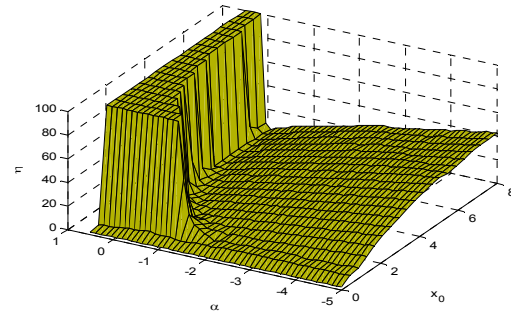


Figure 2: Number of iterations  $\eta$  required to converge optimal control solution for given initial state  $x_0$  and the decomposition parameter  $\alpha$  for unitary weighting operators without terminal cost and time horizon  $N=3$ .

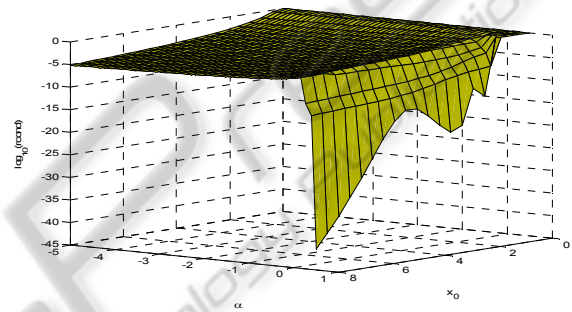


Figure 3: Logarithm base 10 of reciprocal condition number estimate vs. initial state  $x_0$  and the decomposition parameter  $\alpha$  for unitary weighting operators without terminal cost and time horizon  $N=3$ .

matrix  $\left(\left(\hat{\mathbf{L}}_{(i)} \hat{\mathbf{B}}_{(i)}\right)^T \hat{\mathbf{P}} \hat{\mathbf{L}}_{(i)} \hat{\mathbf{B}}_{(i)} + \hat{\mathbf{Q}}\right)$ . Logarithm base 10 of the conditional number is shown in figure 3.

## 6 CONCLUSIONS

The paper discuss selected problems concerned to successive model linearization along predicted state and input trajectories with linear time varying model.

The paper mainly focus on the transformation method from a general nonlinear form into the state space dependent form. We formulate the problem and introduce the generalised form of the algorithm. Nonlinearities are decomposed into two additive terms – state and input dependent matrices of the state space dependent form and then model predictive control can be calculated using methods for linear systems.

An important consequence of the chosen decomposition is reachability of the optimal solution

and required computation time – number of iterations. In many cases the number of iterations can be cut down. The optimal decomposition, for which the algorithm is convergent with minimal number of iterations depends on the initial condition – for receding horizon problems the initial condition is the current state in each time sample. The selection of the decomposition parameters  $\alpha$ ,  $\beta$  should be always connected with current value of the state to ensure suitable value of conditional number corresponding to the inverse of matrix in formula (24).

## ACKNOWLEDGEMENTS

This work was supported by the Ministry of Science and Higher Education in Poland under the grant NN N514 298535.

## REFERENCES

- Błachuta, M. J. 1999. On Fast State-Space Algorithms for Predictive Control. *Int. J. Appl. Math. Comput. Sci.*, Vol. 9, No. 1, 149-160.
- Camacho EF., Bordons C. 2004. *Model Predictive Control*. Springer.
- Chen H., Allgower F. 1998. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1218.
- De Nicolao G., Magni L., Scattolini R.. 2000. Stability and robustness of nonlinear receding horizon control. In F. Allgower and A. Zheng, editors, *Nonlinear Predictive Control*, pp. 3–23. Birkhauser.
- Dutka A. S., Ordys A. W., Grimble M. J. 2003. Nonlinear Predictive Control of 2 dof helicopter model, *IEEE CDC proceedings*.
- Dutka, A., Ordys A. 2004. The Optimal Non-linear Generalised Predictive Control by the Time-Varying Approximation, *Proc. of 10<sup>th</sup> IEEE Int. Conf. MMAR*. Miedzydroje. Poland. pp. 299-304.
- Fontes F. A. 2000. A general framework to design stabilizing nonlinear model predictive controllers. *Syst. Contr. Lett.*, 42(2):127–143.
- Grimble M. J., Ordys A. W. 2001. Non-linear Predictive Control for Manufacturing and Robotic Applications, *Proc. of 7<sup>th</sup> IEEE Int. Conf. MMAR*. Miedzydroje.
- Huang Y., Lu W.M. 1996. Nonlinear Optimal Control: Alternatives to Hamilton-Jacobi Equation”, *Proc. of the 35th IEEE Conference on Decision and Control*, pp. 3942-3947.
- Kouvaritakis B., Cannon M., Rossiter J. A. 1999. Non-linear model based predictive control, *Int. J. Control*, Vol. 72, No. 10, pp. 919-928.
- Kowalczyk Z., Suchomski P. 2005. Discrete-Time Predictive Control With Overparameterized Delay-Plant Models And An Identified Cancellation Order. *Int. J. Appl. Math. Comput. Sci.*, Vol. 15, No. 1, 5–34.
- Lee Y. I., Kouvaritakis B., Cannon M. 2002. Constrained receding horizon predictive control for nonlinear systems, *Automatica*, Vol. 38, No. 12, pp. 2093-2102.
- Magni, L., De Nicolao, G., Scattolini, R. 1999. Some Issues in the Design of Predictive Controllers. *Int. J. Appl. Math. Comput. Sci.*, Vol. 9, No. 1, 9-24.
- Mayne D.Q., Rawlings J.B., Rao C.V., Scokaert P.O.M. 2000. Constrained model predictive control: stability and optimality. *Automatica*, 26(6):789–814.
- Morari M. and Lee J. 1999. Model predictive control: Past, present and future. *Comput. Chem. Eng.*, Vol. 23, No. 4/5, pp. 667–682.
- Morari M., de Oliveira Kothare S. 2000. Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Aut. Contr.*, 45(6):1053–1071.
- Mracek C. P., Cloutier J. R. 1998. Control Designs for the nonlinear benchmark problem via the State-Dependent Riccati Equation method, *International Journal of Robust and Nonlinear Control*, 8, pp. 401-433.
- Ordys A. W., Grimble M. J. 2001. Predictive control design for systems with the state dependent nonlinearities, *SIAM Conference on Control and its Applications*, San Diego, California.
- Orlowski, P. 2005. Convergence of the optimal non-linear GPC method with iterative state-dependent, linear time-varying approximation, *Proc. of Int. Workshop on Assessment and Future Directions of NMPC*, Freudenstadt-Lauterbad, Germany, pp. 491-497.
- Primbs J., Nevistic V., Doyle J. 1999. Nonlinear optimal control: A control Lyapunov function and receding horizon perspective. *Asian Journal of Control*, 1(1):14–24.
- Qin S. J. and Badgwell T. 2003. A survey of industrial model predictive control technology. *Contr. Eng. Pract.*, Vol. 11, No. 7, pp. 733–764.
- Scokaert P.O.M., Mayne D.Q., Rawlings J.B. 1999. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Automat. Contr.*, 44(3):648–654.
- Tatjewski P. 2007. *Advanced Control of Industrial Processes, Structures and Algorithms*. London: Springer.
- Ulbis A., Oлару S., Dumur D., Boucher P. 2007. Explicit solutions for nonlinear model predictive control : a linear mapping approach, *European Control Conference ECC 2007*, Kos, Greece.