Scientia Iranica, Vol. 12, No. 1, pp 43–54

SCIENTIA
IRANICA

Analysis of the Convergence and Closed Loop Stability in EDMC

M. Haeri* and H. Zadehmorshed Beik[1]

In this paper, the convergence and stability conditions of extended DMC in the control of nonlinear SISO and MIMO systems are investigated. The formulations are based on the ordinary DMC in which, with successive linearization of the nonlinear model and new interpretation of disturbance, the nonlinear extension is deduced. In addition, new convergence and stability criteria are derived for SISO and MIMO systems. These criteria include convergence and stability in the case of longer control ($M > 1$) and prediction ($P > 1$) horizons, as well as the finite and infinite sampling time. Finally, the simulation results for a MIMO (3×3) model, based on a power unit nonlinear plant, are presented.

INTRODUCTION

Model Predictive Control (MPC) and its industrial applications have become more and more popular during the last few decades [1-3]. MPC refers to a family of controllers that share three basic schemes [4]. In all model predictive controllers, an explicit model is employed to predict the future outputs of the process. This is why they are also categorized in model based control approaches. The control signal is determined, based on an optimization algorithm to optimize an objective function. This property enables the controllers to handle the constraints explicitly during the control design stage. Finally, the calculated control signals are applied, based on the receding horizon strategy, which is why they are also called receding horizon controllers.

So far, almost all types of model structure have been employed in MPC. Linear MPC refers to those controllers that use a linear model structure of the process. In the absence of constraints, use of linear models results in a closed form solution to the optimization problem. However, in constrained linear MPC, a quadratic programming is usually performed. In nonlinear MPC controllers, on the other hand, a nonlinear model of the process is employed. In this case, the optimization problem does not generally have a closed form solution. Usually, nonlinear programming is implemented to obtain results. Nonlinear programming is run, based on either a sequential [5,6] or simultaneous [7,8] strategy. In either case, high computational power is required [9]. This deficiency confines application of nonlinear MPC on processes that possess slow dynamics. In order to reduce computational requirements, linear approximations have been employed in some designs of nonlinear model predictive controllers. Multi Model adaptive Predictive Control (MMPC) [10], Quadratic Dynamic Matrix Control (QDMC and NLQDMC) [11,12], Extended Dynamic Matrix Control (EDMC) [13,14] and Universal Dynamic Matrix Control (UDMC) [15] have been introduced from this point of view. In the first three approaches, an approximated linear model is used to predict the effects of future control moves at each sample time. Free response, however, is computed by integrating the nonlinear model equations. While the linear models are determined off line in MMPC, in QDMC and EDMC they are obtained based on the Jacobian and the perturbation methods at each sample time. The main difference between QDMC and EDMC is the way of computation and consideration of the model/process mismatch and external disturbance. In UDMC, linear approximation is implemented only in the calculation of the Jacobians during the updating state of the optimization algorithm.

There is plenty of research work on the stability

*. *Corresponding Author, Department of Electrical Engineering, Sharif University of Technology, P.O. Box 11365 9363, Tehran, I.R. Iran.*
1. *Department of Electrical Engineering, Sharif University of Technology, P.O. Box 11365 9363, Tehran, I.R. Iran.*

consideration of linear [16-21] and nonlinear [22-25] MPC controllers. The backbone of ideas presented in the literature relies on the infiniteness of the prediction horizon and, therefore, use of the established stability property of the LQ control theory. Because of impracticality, in most cases, the infiniteness of the horizon is replaced by the stability constraints (including terminal constraints) in some improved schemes. Both hard [16,21] and soft constraints [26,27] were considered. By choosing the cost function in MPC as the candidate of the Lyapunov function and showing that it is a non-increasing function, the stability of the closed loop system is proved. The stability is derived for the nominal condition in most cases and is based on the state space model representation of the process, in which states are assumed to be measurable directly or determined through indirect measurements. It is also assumed that the implemented optimization algorithms are converged to the correct solution of the problem.

Due to their conceptual simplicity and lower computational requirements, predictive controllers, such as QDMC and EDMC, provide realistic advanced control strategies for industrial applications. It is shown in [12] that by tolerating a negligible loss in the closed loop performance, QDMC was six times faster than a nonlinear MPC run based on the sequential method. Unfortunately, there is not an analytical and explicit stability proof for QDMC or NLQDMC. Nevertheless, stable closed loop behavior has been illustrated in computer simulations, as well as actual implementation of the controllers, even for open loop integrating, as well as unstable processes [9,11,12]. To get a stable closed loop system, the state space representation of the process, along with a state estimator, has been employed in NLQDMC [12]. On the other hand, EDMC has proved to be stable, provided that some conditions on the control parameters and the steady state gain of the process are satisfied [13]. Proof of the convergence in the internal loop, as well as closed loop stability, was derived based on the contraction mapping theorem. Input/output representations of the process and the controller are used in the formulation of the problem. Imposing unrealistic conditions, such as infinite sample time and restricting assumptions, such as unity control and prediction horizons, make the mentioned proof unacceptable in practice. In this paper, the authors try to extend the applicability of the proof by relaxing the above mentioned restricting conditions. In this way, a nonlinear MPC (or time varying linear MPC) has been provided with less computational requirements, as well as acceptable and realistic stability conditions.

The paper is organized as follows. First, the DMC and its nonlinear extension are reviewed and nonlinear vector equations and some powerful solutions are presented. Then, the convergence and closed loop stability conditions for a SISO process case with $M > 1$, for finite and infinite sampling time, are proposed. After that, the EDMC formulation for the MIMO system is derived and the related convergence theorem and the stability criteria are presented, respectively. Finally, the simulation results for a nonlinear model (3×3) of a power unit plant are illustrated. Conclusions are given in the last section.

DMC FORMULATION AND ITS NONLINEAR EXTENSION FOR SISO SYSTEMS

Since EDMC formulation is based on ordinary DMC, a brief description of both controllers is presented in this section.

DMC

Considering a SISO system step response model, the output could be determined using the following discrete convolution:

$$y^{\mathrm{lin}}(k) = \sum_{i=1}^{N} a_i \Delta u(k-i) + a_N u(k-N-1) + d(k), \tag{1}$$

where $u(k)$ and $\Delta u(k)$ are the input and its variation i.e. $u(k) - u(k-1)$ in sample time k, respectively. a_i is the step response coefficient at sample time i. N is the number of sample time at which step response reaches its steady state. The following disturbance represents any difference between the measured output and the one predicted by the above model:

$$d(k) = y^{meas}(k) - \sum_{i=1}^{N} a_i \Delta u(k-i) - a_N u(k-N-1). \tag{2}$$

The future predictions of the process output are given in the following matrix-vector relation:

$$\begin{bmatrix} y^{\mathrm{lin}}(k+1) \\ y^{\mathrm{lin}}(k+2) \\ \vdots \\ y^{\mathrm{lin}}(k+P) \end{bmatrix}$$

$$= \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ \vdots & & & & \\ a_M & a_{M-1} & & \cdots & a_1 \\ \vdots & & & & \\ a_p & a_{p-1} & & \cdots & a_{p-M+1} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+M-1) \end{bmatrix} +$$

$$\begin{bmatrix} a_2 & a_3 & \cdots & & a_N \\ a_3 & a_4 & \cdots & a_N & 0 \\ \vdots & & & & \\ a_{p+1} & & \cdots & a_N & \cdots & 0 \end{bmatrix} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-N+1) \end{bmatrix}$$

$$+ \begin{bmatrix} a_N u(k-N) \\ a_N u(k-N+1) \\ \vdots \\ a_N u(k-N+P-1) \end{bmatrix} + \begin{bmatrix} d(k+1) \\ d(k+2) \\ \vdots \\ d(k+P) \end{bmatrix}. \quad (3)$$

This can be written, equivalently, in the following vector form:

$$\mathbf{y}^{\text{lin}} = \mathbf{A}\boldsymbol{\Delta}\mathbf{u} + \mathbf{y}^{\text{past}} + \mathbf{d}, \quad (4)$$

P and M are the prediction and moving (control) horizons, respectively, $\mathbf{y}^{\text{past}}$ denotes the effects of the past inputs on the predicted outputs. $\mathbf{A}$ is a Toeplitz matrix, consisting of step response coefficients and is called the dynamic matrix of the process. Since future estimates of the mismatches are not available, it is customary to assume $d(k+i) = d(k)$ for $i = 1, 2, \cdots, P$. The control moves, $\boldsymbol{\Delta}\mathbf{u}$, are determined according to the solution of the following optimization problem.

$$\min_{\Delta u} \sum_{i=1}^{P} \left[y^d(k+i) - y^{\text{lin}}(k+i)\right]^2 + \sum_{j=1}^{M} \lambda^2 \left[\Delta u(k+M-j)\right]^2, \quad (5)$$

y^d is the desired output trajectory and λ is the weighting coefficient on the input variations. Using least square estimation, the solution of the problem is given by:

$$\boldsymbol{\Delta}\mathbf{u} = \left(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I}\right)^{-1} \mathbf{A}^T \left(\mathbf{y}^d - \mathbf{y}^{\text{past}} - \mathbf{d}\right). \quad (6)$$

Usually, the first component of $\boldsymbol{\Delta}\mathbf{u}$, i.e. $\Delta u(k)$, will be applied to the process ($u(k) = u(k-1) + \Delta u(k)$) and the same procedure will be performed in future sampling intervals.

Extended DMC

To extend the application of DMC (which is originally based on a linear model of the process) to nonlinear systems, it is required to employ an approximated linear model for a nonlinear process in each sample interval. This is done via linearization of the process's nonlinear model or by determination of the process response for a step perturbation. Also, a new interpretation of disturbance is introduced, due to the nonlinear character of the process. This is explained in more detail in the next section.

Nonlinear Disturbance

In this extension, a new interpretation of $\mathbf{d}$ is exploited [13]. In other words, $\mathbf{d}$ is split into two parts, the unknown parts, $\mathbf{d}^{\text{ext}}$, which are treated as in ordinary DMC and the known parts, $\mathbf{d}^{\text{nl}}$, which represent the difference between approximated linear and nonlinear models of the process.

$$\mathbf{d} = \mathbf{d}^{\text{ext}} + \mathbf{d}^{\text{nl}}. \quad (7)$$

To consider this partitioning, the predicted outputs of the process are written as:

$$\mathbf{y}^{\text{el}} = \mathbf{A}\boldsymbol{\Delta}\mathbf{u} + \mathbf{y}^{\text{past}} + \mathbf{d}^{\text{ext}} + \mathbf{d}^{\text{nl}}, \quad (8)$$

$d^{\text{ext}}(k+i)$ is assumed constant over the prediction horizon ($i = 1, 2, \cdots, P$) and $d^{\text{nl}}(k+i)$ varies during the horizon. Solving Problem 5 results in the following relation for input variations:

$$\boldsymbol{\Delta}\mathbf{u}(\mathbf{d}^{\text{nl}}) = \left(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I}\right)^{-1}\mathbf{A}^T\left(\mathbf{y}^{\text{sp}} - \mathbf{y}^{\text{past}} - \mathbf{d}^{\text{ext}} - \mathbf{d}^{\text{nl}}\right), \quad (9)$$

$\mathbf{d}^{\text{nl}}$ is determined, in order to have the same predicted outputs from the nonlinear, $\mathbf{y}^{\text{nl}}$, and linear, $\mathbf{y}^{el}$, models,

$$\mathbf{y}^{\text{nl}}(\mathbf{d}^{\text{nl}}) = \mathbf{y}^{\text{el}} = \mathbf{A}\boldsymbol{\Delta}\mathbf{u}(\mathbf{d}^{\text{nl}}) + \mathbf{y}^{\text{past}} + \mathbf{d}^{\text{ext}} + \mathbf{d}^{\text{nl}}. \quad (10)$$

To solve the above problem, equation 10 is reformulated in the following equation, which is a root finding problem:

$$\mathbf{f}(\mathbf{d}^{\text{nl}}) = \mathbf{y}^{\text{nl}}(\mathbf{d}^{\text{nl}}) - \mathbf{A}\boldsymbol{\Delta}\mathbf{u}(\mathbf{d}^{\text{nl}}) - \mathbf{y}^{\text{past}} - \mathbf{d}^{\text{nl}} - \mathbf{d}^{\text{ext}} = \mathbf{0}. \quad (11)$$

There are P nonlinear equations in P unknowns, $d^{\text{nl}}(k+1), \cdots, d^{\text{nl}}(k+P)$. Nonlinearity of the equations arises from the nonlinear relation that exists between $\mathbf{y}^{\text{nl}}$ and $\mathbf{u}$ (and, therefore, $\boldsymbol{\Delta}\mathbf{u}$). In the following subsection, some well established methods are summarized that can be applied to find the solution to the problem. One of the simplest methods is successive substitution via a fixed-point algorithm,

$$\mathbf{d}^{\text{nl}}_{k+1} = \mathbf{f}_1(\mathbf{d}^{\text{nl}}_k) = \mathbf{d}^{\text{nl}}_k + \beta(\mathbf{y}^{\text{nl}}_k - \mathbf{y}^{\text{el}}_k) = \mathbf{d}^{\text{nl}}_k + \beta\mathbf{f}(\mathbf{d}^{\text{nl}}_k). \quad (12)$$

A block diagram of EDMC in Figure 1 shows the internal iteration and external DMC closed loop after convergence.

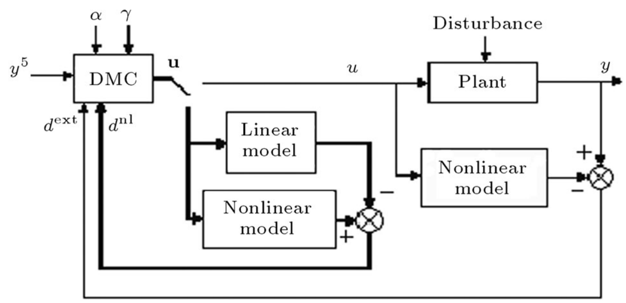

Figure 1. Block diagram of EDMC algorithm.

Nonlinear Vector Equation Solution

Almost all existing approaches to finding a solution to a nonlinear vector function, such as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, rely on iterative methods. The Newton method [28] is perhaps one of the most popular and is given by the following relation:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - [\mathbf{f}'(\mathbf{x}^k)]^{-1}\mathbf{f}(\mathbf{x}^k). \tag{13}$$

This method benefits from convergence rate 2 around the optimal solution. However, it requires a Jacobian matrix that is not easily available in most applications, like Problem 11. A simplified version of the Newton method is obtained by substituting $\mathbf{f}'(\mathbf{x})$ with a fixed matrix, $\mathbf{C}$. Sometimes, the initial value of the Jacobian matrix, $\mathbf{f}'(\mathbf{x}^0)$, is used in this regard:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{C}^{-1}\mathbf{f}(\mathbf{x}^k) = \mathbf{x}^k - \mathbf{f}'(\mathbf{x}^0)^{-1}\mathbf{f}(\mathbf{x}^k). \tag{14}$$

The fixed-point iteration method is obtained by replacing $-\mathbf{C}^{-1}$ with $\beta\mathbf{I}$, in which β is a small positive coefficient:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \beta\mathbf{f}(\mathbf{x}^k), \quad \beta \in (0,1). \tag{15}$$

This method requires the least computational effort and benefits from a convergence rate of one. Iteration methods given in Equations 12 and 14 have been used in [13] and some good results have been obtained for certain conditions. In Quasi Newton (QN) methods, some approximation of the Jacobian matrix is employed. In Broyden's method [29], the following steps are performed in each iteration:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k,$$

$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k),$$

$$\mathbf{w}_k\mathbf{s}_k = -\mathbf{f}(\mathbf{x}_k),$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{(\mathbf{y}_k - \mathbf{w}_k\mathbf{s}_k)\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{s}_k}, \tag{16}$$

$\mathbf{w}_k$ is an approximate for $\mathbf{f}'(\mathbf{x}_k)$ and initialized by $\mathbf{w}_0 = \mathbf{f}'(\mathbf{x}_0)$. This method converges to the solution at an approximate rate of 1.6, which means super linear convergence. Some other methods of the QN family, such as Greenstadt, Barnes and Thomas, can be found in [30].

The two following subsections describe the contribution made by the present work. This includes an extension of the work in [13] to higher M and MIMO systems.

Convergence for SISO Systems with $M > 1$

As in [13], convergence of the fixed-point iterations can be proved via the contraction-mapping theorem. It is shown that for a globally asymptotically stable open loop system, iterations in Equations 9 and 12 will converge if the sampling time and weighting factor are large enough and the relaxation factor, β, is small enough. Since in the above-mentioned conditions, assumption $M = 1$ is not used, the statement is also valid for $M > 1$. With some mathematical manipulation of the results given in [13], it is shown that the following condition on the weighting factor (λ) is required to get the convergence:

$$\lambda^2 >> 2MP \max_i \{|a_i\overline{a_i}|\}, \tag{17}$$

in which a_i and $\overline{a_i}$ are the steady state gain of the linear and nonlinear models at the ith iteration. Since M is allowed to be higher than one, more performance improvement can be expected from the controller.

Stability for SISO Systems with $M > 1$

The operator theory and contraction mapping [31] can be used to show the closed loop stability of the system for $M > 1$ and infinite/finite sampling time (T) as follows.

Infinite Sampling Time

In this section, it is required to extend the stability criteria of an SISO system for a $M > 1$ case. Regarding stability, it is assumed that the iterative computation of $\mathbf{d}^{\text{nl}}$ in sample time k has been converged and the goal is to derive a relation between present (u_k) and previous (u_{k-1}) inputs.

$$\begin{aligned} u_k &= u_{k-1} + \Delta u_k \\ &= u_{k-1} + \mathbf{e}_1^T \left[(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1} \right. \\ &\qquad \left. \mathbf{A}^T(\mathbf{y}_{k+1}^{\text{sp}} - \mathbf{y}_{k+1}^{\text{past}} - \mathbf{d}_{k+1}^{\text{nl}}) \right], \end{aligned} \tag{18}$$

$\mathbf{e}_i$ is a $M \times 1$ vector with all elements zero, except the ith element, which is one. Note that for a nominal system, $\mathbf{d}^{\text{ext}} = \mathbf{0}$. To complete and simplify Equation 18, the converged value of $\mathbf{d}_{k+1}^{\text{nl}}$ should be determined. When the convergence has been reached, the following equality is satisfied:

$$\mathbf{y}^{\mathrm{nl}} = \mathbf{y}^{el}, \quad \mathbf{y}^{\mathrm{nl}} = [y^{\mathrm{nl}}(k+1)\cdots y^{\mathrm{nl}}(k+P)]^T,$$

and:

$$\mathbf{y}^{el} = [y^{el}(k+1)\cdots y^{el}(k+P)]^T. \tag{19}$$

The output of the extended linear model, $\mathbf{y}_k^{\mathrm{el}}$, is obtained as follows:

$$\begin{aligned}\mathbf{y}_{k+1}^{\mathrm{el}} &= \mathbf{y}_{k+1}^{\mathrm{past}} + \mathbf{A}\boldsymbol{\Delta}\mathbf{u}_k + \mathbf{d}_{k+1}^{\mathrm{nl}}\\ &= \mathbf{y}_{k+1}^{\mathrm{past}} + \mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\\ &\quad \mathbf{A}^T(\mathbf{y}_{k+1}^{\mathrm{sp}} - \mathbf{y}_{k+1}^{\mathrm{past}} - \mathbf{d}_{k+1}^{\mathrm{nl}}) + \mathbf{d}_{k+1}^{\mathrm{nl}}.\end{aligned} \tag{20}$$

Let one define:

$$\mathbf{A}_0 = \mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T.$$

Using Equations 19 and 20 and the above definition, one can derive the following relation:

$$(\mathbf{I} - \mathbf{A}_0)\mathbf{d}_{k+1}^{\mathrm{nl}} = \mathbf{y}_{k+1}^{\mathrm{nl}} - (\mathbf{I} - \mathbf{A}_0)\mathbf{y}_{k+1}^{\mathrm{past}} - \mathbf{A}_0\mathbf{y}_{k+1}^{\mathrm{sp}},$$

or:

$$\mathbf{d}_{k+1}^{\mathrm{nl}} = (\mathbf{I}-\mathbf{A}_0)^{-1}\mathbf{y}_{k+1}^{\mathrm{nl}} - \mathbf{y}_{k+1}^{\mathrm{past}} - (\mathbf{I}-\mathbf{A}_0)^{-1}\mathbf{A}_0\mathbf{y}_{k+1}^{\mathrm{sp}}. \tag{21}$$

Substituting $\mathbf{d}_k^{\mathrm{nl}}$ from Equation 21 in Equation 18, one obtains:

$$\begin{aligned}u_k &= u_{k-1} + \Delta u_k\\ &= u_{k-1} + \mathbf{e}_1^T\big[(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\\ &\quad \mathbf{A}^T(\mathbf{y}_{k+1}^{\mathrm{sp}} - \mathbf{y}_{k+1}^{\mathrm{past}} - (\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{y}_{k+1}^{\mathrm{nl}} + \mathbf{y}_{k+1}^{\mathrm{past}}\\ &\quad + (\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{A}_0\mathbf{y}_{k+1}^{\mathrm{sp}})\big].\end{aligned} \tag{22}$$

Using the following two matrix relationships for $\mathbf{A}_0$:

$$(\mathbf{I} - \mathbf{A}_0)^{-1} = \mathbf{I} + \frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T,$$

$$(\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{A}_0 = \frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T. \tag{23}$$

Equation 22 is simplified as:

$$\begin{aligned}u_k &= u_{k-1} + \mathbf{e}_1^T(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\\ &\quad \mathbf{A}^T(\mathbf{I} + \frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T)(\mathbf{y}_{k+1}^{\mathrm{sp}} - \mathbf{y}_{k+1}^{\mathrm{nl}})\\ &= u_{k-1} + \frac{1}{\lambda^2}\mathbf{e}_1^T\mathbf{A}^T(\mathbf{y}_{k+1}^{\mathrm{sp}} - \mathbf{y}_{k+1}^{\mathrm{nl}}).\end{aligned} \tag{24}$$

When considering the infinite sampling time assumption ($T \to \infty$), one can use the following definitions:

$$\mathbf{y}_{k+1}^{\mathrm{nl}} = y^{\mathrm{nl}}(k+1)\mathbf{1}_P, \qquad \mathbf{y}_{k+1}^{\mathrm{sp}} = y^{\mathrm{sp}}\mathbf{1}_P,$$

$$\mathbf{A} = a\mathbf{1}_L, \qquad e_1^T\mathbf{A}^T\mathbf{1}_P = aP, \tag{25}$$

where:

$$\mathbf{1}_P = \begin{bmatrix}1\\1\\ \vdots\\1\end{bmatrix}_{P\times 1}, \quad \mathbf{1}_L = \begin{bmatrix}1 & 0 & 0 & \cdots & 0\\ 1 & 1 & 0 & \cdots & 0\\ \vdots & & & & \\ 1 & 1 & 1 & \cdots & 1\\ \vdots & & & & \\ 1 & 1 & 1 & \cdots & 1\end{bmatrix}_{P\times M}. \tag{26}$$

Therefore, Equation 24 is further simplified as:

$$u_k = u_{k-1} + \frac{aP}{\lambda^2}(y^{\mathrm{sp}} - y^{\mathrm{nl}}(k+1)). \tag{27}$$

This equation is similar to the one derived in [13] but assumption $M = 1$ is not used in this case. Therefore, results given in [13] for the stability of the closed loop system are also applicable to $M > 1$. This results in the following theorem.

Nominal Stability Theorem

Suppose that the system to be controlled is globally asymptotically stable for all feasible inputs and, furthermore, suppose that the following is valid:

1. The steady state gain of the system does not change sign;
2. The weight on the change of the input is larger than zero;
3. The sampling time is long enough ($T \to \infty$);
4. The set point is constant in the prediction horizon.

Then, the closed loop system is guaranteed to be nominally stable.

Finite Sampling Time

In this section, one considers relaxing assumption 3 in the above stability theory. According to the previous section, this condition is employed in two cases, first in definitions given in Equation 25 and, second, in deriving the Jacobian of the input in Equation 27. In the following lines, analysis of the stability is continued without considering assumption 3. Recall the relation for u_k in Equation 24.

$$u_k = u_{k-1} + \frac{1}{\lambda^2}\mathbf{e}_1^T\mathbf{A}^T(\mathbf{y}_{k+1}^{\mathrm{sp}} - \mathbf{y}_{k+1}^{\mathrm{nl}}). \tag{28}$$

It is assumed that the internal iteration in EDMC has been converged. This relation is expanded using elements of $\mathbf{e}_1, \mathbf{A}, \mathbf{y}^{\mathrm{sp}}_{k+1}$ and $\mathbf{y}^{\mathrm{nl}}_{k+1}$:

$$u_k = u_{k-1} + \frac{1}{\lambda^2}\sum_{i=1}^{P} a_i(y^{\mathrm{sp}}(k+i) - y^{\mathrm{nl}}(k+i)). \quad (29)$$

To derive the Jacobian of $\frac{\partial u_k}{\partial u_{k-1}}$, derivatives of the future outputs, with respect to u_{k-1}, are required.

$$\begin{aligned}\frac{\partial u_k}{\partial u_{k-1}} &= 1 - \frac{1}{\lambda^2}\sum_{i=1}^{P} a_i \frac{\partial y^{\mathrm{nl}}(k+i)}{\partial u_{k-1}}\\ &= 1 - \frac{1}{\lambda^2}\Big(a_1\frac{\partial y^{\mathrm{nl}}(k+1)}{\partial u_{k-1}} + a_2\frac{\partial y^{\mathrm{nl}}(k+2)}{\partial u_{k-1}}\\ &\quad + \cdots + a_p\frac{\partial y^{\mathrm{nl}}(k+P)}{\partial u_{k-1}}\Big).\end{aligned} \quad (30)$$

To solve the problem, the linear approximations of $y^{\mathrm{nl}}(k+i)$ are used in the computation:

$$\begin{aligned}y(k+1) &= a_1u(k) + (a_2-a_1)u(k-1) + (a_3-a_2)u(k-2)\\ &\quad + \cdots + (a_N - a_{N-1})u(k-N+1),\\ y(k+2) &= a_1u(k+1) + (a_2-a_1)u(k) + (a_3-a_2)u(k-1)\\ &\quad + \cdots + (a_N - a_{N-1})u(k-N+2),\\ &\vdots\\ y(k+M) &= a_1u(k+M-1) + (a_2-a_1)u(k+M-2)\\ &\quad + \cdots + (a_{M+1}-a_M)u(k-1)\\ &\quad + \cdots + (a_N - a_{N-1})u(k-N+M),\\ y(k+M+1) &= a_2u(k+M-1) + (a_3-a_2)u(k+M-2)\\ &\quad + \cdots + (a_{M+2}-a_{M+1})u(k-1)\\ &\quad + \cdots + (a_N - a_{N-1})u(k-N+M+1),\\ &\vdots\\ y(k+P+1) &= a_{P-M+1}u(k+M-1) +\\ &\quad (a_{P-M+2} - a_{P-M+1})u(k+M-2)\\ &\quad + \cdots + (a_{P+1}-a_P)u(k-1) + \cdots\\ &\quad + (a_N - a_{N-1})u(k-N+P).\end{aligned} \quad (31)$$

Let the following definitions be used:

$$\mathbf{A} = [\underline{\mathbf{a}}_1\ \underline{\mathbf{a}}_2 \cdots \underline{\mathbf{a}}_M],$$

and:

$$z_1 = \frac{\partial u_k}{\partial u_{k-1}},\ z_2 = \frac{\partial u_{k+1}}{\partial u_{k-1}},\ \cdots,\ z_M = \frac{\partial u_{k+M-1}}{\partial u_{k-1}}. \quad (32)$$

Therefore, z_i are determined as follows:

$$\begin{aligned}&z_1 = 1 - \frac{1}{\lambda^2}\underline{\mathbf{a}}_1^T l, \quad z_2 = z_1 - \frac{1}{\lambda^2}\underline{\mathbf{a}}_2^T\mathbf{l}, \quad \cdots,\\ &z_M = z_{M-1} - \frac{1}{\lambda^2}\underline{\mathbf{a}}_M^T\mathbf{l},\end{aligned} \quad (33)$$

$\mathbf{l}$ is a column vector given as:

$$\mathbf{l} = \begin{bmatrix} a_1z_1 + a_2 - a_1\\ a_1z_2 + (a_2-a_1)z_1 + a_3 - a_2\\ \vdots\\ a_1z_M + (a_2-a_1)z_{M-1} + \cdots\\ \quad + (a_M - a_{M-1})z_1 + a_{M+1} - a_M\\ a_2z_M + (a_3-a_2)z_{M-1} + \cdots\\ \quad + (a_{M+1}-a_M)z_1 + a_{M+2} - a_{M+1}\\ \vdots\\ a_{P-M+1}z_M + (a_{P-M+2} - a_{p-M+1})z_{M-1}\\ \quad + \cdots + (a_P - a_{P-1})z_1 + a_{P+1} - a_P\end{bmatrix}. \quad (34)$$

Substitute for z_i in Equation 34 from Equations 33 and, with some manipulation, $\mathbf{l}$ is reformulated as follows:

$$\mathbf{l} = \begin{bmatrix} a_2 - \frac{1}{\lambda^2}a_1\underline{\mathbf{a}}_1^T\mathbf{l}\\ a_3 - \frac{1}{\lambda^2}(a_2\underline{\mathbf{a}}_1^T + a_1\underline{\mathbf{a}}_2^T)\mathbf{l}\\ a_{p+1} - \frac{1}{\lambda^2}(a_P\underline{\mathbf{a}}_1^T + \cdots + a_{P+M-1}\underline{\mathbf{a}}_M^T)\mathbf{l}\end{bmatrix}. \quad (35)$$

Or, in simple form, it can be written as:

$$\mathbf{l} = \hat{\mathbf{b}} - \frac{1}{\lambda^2}\hat{\mathbf{B}}\mathbf{l}, \quad (36)$$

where:

$$\hat{\mathbf{b}} = \begin{bmatrix} a_2\\ a_3\\ \vdots\\ a_{P+1}\end{bmatrix},$$

$$\hat{\mathbf{B}} = \begin{bmatrix} a_1\underline{\mathbf{a}}_1^T\\ a_2\underline{\mathbf{a}}_1^T + a_1\underline{\mathbf{a}}_2^T\\ a_M\underline{\mathbf{a}}_1^T + \cdots + a_1\underline{\mathbf{a}}_M^T\\ a_{M+1}\underline{\mathbf{a}}_1^T + \cdots + a_2\underline{\mathbf{a}}_M^T\\ \vdots\\ a_P\underline{\mathbf{a}}_1^T + \cdots + a_{P+M-1}\underline{\mathbf{a}}_M^T\end{bmatrix}. \quad (37)$$

Equation 36 can be solved for $\mathbf{l}$:

$$\mathbf{l} = (\mathbf{I} + \frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T)^{-1}\hat{\mathbf{b}}. \quad (38)$$

Using the definition of $\mathbf{A}_0$ in the previous section, since $(\mathbf{I} + \frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{I} - \mathbf{A}_0$, therefore:

$$\mathbf{l} = (\mathbf{I} - \mathbf{A}_0)\hat{\mathbf{b}} = \hat{\mathbf{b}} - \mathbf{A}_0\hat{\mathbf{b}}. \quad (39)$$

From the first row of Equation 34, z_1 is determined as follows:

$$\begin{aligned} z_1 &= \frac{1}{a_1}(l_1 + a_1 - a_2) \\ &= \frac{1}{a_1}(\mathbf{e}_1^T\mathbf{l} + a_1 - a_2) \\ &= \frac{1}{a_1}(\mathbf{e}_1^T\hat{\mathbf{b}} - \mathbf{e}_1^T\mathbf{A}_0\hat{\mathbf{b}} + a_1 - a_2) \\ &= \frac{1}{a_1}(a_1 - \mathbf{e}_1^T\mathbf{A}_0\hat{\mathbf{b}}). \end{aligned} \quad (40)$$

Based on the contraction mapping theorem, closed loop stability requires $z_1 < 1$. In other words:

$$\frac{1}{a_1}\mathbf{e}_1^T\mathbf{A}_0\hat{\mathbf{b}} > 0. \quad (41)$$

Some Special Cases for Stability

1. $P = M = 1$,

$$\mathbf{A}_0 = a_1(a_1^2 + \lambda^2)^{-1}a_1, \quad \hat{\mathbf{b}} = a_2,$$

$$\frac{1}{a_1}\mathbf{e}_1^T\mathbf{A}_0\hat{\mathbf{b}} = \frac{1}{a_1}\frac{a_1^2}{a_1^2+\lambda^2}a_2 = \frac{a_1a_2}{a_1^2+\lambda^2} > 0$$

$$\Rightarrow a_1a_2 > 0. \quad (42)$$

2. $M = 1, P > 1$,

$$\mathbf{A} = [a_1a_2\cdots a_P]^T,$$

$$\frac{1}{a_1}\mathbf{e}_1^T\mathbf{A}_0\hat{\mathbf{b}} = \frac{\mathbf{A}^T\hat{\mathbf{b}}}{\mathbf{A}\mathbf{A}^T + \lambda^2} > 0$$

$$\Rightarrow \mathbf{A}^T\hat{\mathbf{b}} > 0. \quad (43)$$

Or, equivalently,

$$a_1a_2 + a_2a_3 + \cdots + a_Pa_{P+1} = \sum_{i=1}^{P} a_ia_{i+1} > 0. \quad (44)$$

3. $M = 2, P > 2$

$$\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I} = \begin{bmatrix} \sum_{i=1}^{p} a_i^2 + \lambda^2 & \sum_{i=1}^{p-1} a_ia_{i+1} \\ \sum_{i=1}^{p-1} a_ia_{i+1} & \sum_{i=1}^{p-1} a_i^2 + \lambda^2 \end{bmatrix}, \quad (45)$$

$$\frac{1}{a_1}\mathbf{e}_1^T\mathbf{A}_0\hat{\mathbf{b}} =$$

$$\frac{\left[\sum_{i=1}^{p-1} a_i^2 + \lambda^2 \quad -\sum_{i=1}^{p-1} a_ia_{i+1}\right]}{\det(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{l})} \begin{bmatrix} \sum\limits_{i=1}^{p} a_ia_{i+1} \\ \sum\limits_{i=1}^{p-1} a_ia_{i+2} \end{bmatrix}$$

$$= \frac{\left(\sum\limits_{i=1}^{p-1} a_i^2 + \lambda^2\right)\left(\sum\limits_{i=1}^{p} a_ia_{i+1}\right) - \left(\sum\limits_{i=1}^{p-1} a_ia_{i+1}\right)\left(\sum\limits_{i=1}^{p-1} a_ia_{i+2}\right)}{\left(\sum\limits_{i=1}^{p} a_i^2 + \lambda^2\right)\left(\sum\limits_{i=1}^{p-1} a_i^2 + \lambda^2\right) - \left(\sum\limits_{i=1}^{p-1} a_ia_{i+1}\right)^2} > 0. \quad (46)$$

DMC FORMULATION FOR LINEAR MIMO SYSTEMS

For the sake of simplicity, formulations are given for 2×2 systems. The same line of calculations is used to determine formulas for $n \times n$ systems. Outputs of a stable linear time invariant 2×2 system can be represented, using its step response model, by the following equations:

$$\begin{aligned} y_1(k) &= \sum_{i=1}^{N} a_i\Delta u_1(k-i) + a_Nu_1(k-N-1) \\ &\quad + \sum_{i=1}^{N} b_i\Delta u_2(k-i) + b_Nu_2(k-N-1) + d_1(k), \\ y_2(k) &= \sum_{i=1}^{N} c_i\Delta u_1(k-i) + c_Nu_1(k-N-1) \\ &\quad + \sum_{i=1}^{N} d_i\Delta u_2(k-i) + d_Nu_2(k-N-1) + d_2(k), \end{aligned} \quad (47)$$

where $u_i(k)$ and $\Delta u_i(k)$ are the ith input and it's variation in sample time k. a_i, b_i, c_i and d_i are the step response coefficients at sample time i. N is the sample time at which all the step responses reach their steady state. $\mathbf{d}_i$ stands for any differences between the system output and the one predicted by the step response model. These errors account for model/system mismatches and external disturbances. Future predictions of the system outputs for prediction horizon P, based on control horizon M, are given in the following matrix-vector relation:

$$\begin{bmatrix} y_1(k+1) \\ \vdots \\ y_1(k+P) \\ y_2(k+1) \\ \vdots \\ y_2(k+P) \end{bmatrix} = \begin{bmatrix} a_1 0 & \cdots & 0 & b_1 0 & \cdots & 0 \\ a_2 a_1 0 & \cdots & 0 & b_2 b_1 0 & \cdots & 0 \\ \vdots & & & & & \\ a_P a_{P-1} & \cdots & a_{P-M+1} & b_P b_{P-1} & \cdots & b_{P-M+1} \\ c_1 0 & \cdots & 0 & d_1 0 & \cdots & 0 \\ c_2 c_1 0 & \cdots & 0 & d_2 d_1 0 & \cdots & 0 \\ \vdots & & & & & \\ c_P c_{P-1} & \cdots & c_{P-M+1} & d_P d_{P-1} & \cdots & d_{P-M+1} \end{bmatrix} \begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \\ \vdots \\ \Delta u_1(k+M-1) \\ \Delta u_2(k) \\ \Delta u_2(k+1) \\ \vdots \\ \Delta u_2(k+M-1) \end{bmatrix}$$

$$+ \begin{bmatrix} a_2 & a_3 & \cdots & & a_{N+1} & b_2 & b_3 & \cdots & & b_{N+1} \\ a_3 & a_4 & \cdots & a_{N+1} & 0 & b_3 & b_4 & \cdots & b_{N+1} & 0 \\ \vdots & & & & & & & & & \\ a_{P+1} & & \cdots & a_{N+1} & \cdots & 0 & b_{P+1} & \cdots & b_{N+1} \cdots & 0 \\ c_2 & c_3 & \cdots & & c_{N+1} & d_2 & d_3 & \cdots & & d_{N+1} \\ c_3 & c_4 & \cdots & c_{N+1} & 0 & d_3 & d_4 & \cdots & d_{N+1} & 0 \\ \vdots & & & & & & & & & \\ c_{P+1} & & \cdots & c_{N+1} & \cdots & 0 & d_{P+1} & \cdots & d_{N+1} \cdots & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_1(k-2) \\ \vdots \\ \Delta u_1(k-N+1) \\ \Delta u_2(k-1) \\ \Delta u_2(k-2) \\ \vdots \\ \Delta u_2(k-N+1) \end{bmatrix}$$

$$+ \begin{bmatrix} a_{N+1}u_1(k-N) + b_{N+1}u_2(k-N) \\ a_{N+1}u_1(k-N+1) + b_{N+1}u_2(k-N+1) \\ \vdots \\ a_{N+1}u_1(k-N+P-1) + b_{N+1}u_2(k-N+P-1) \\ c_{N+1}u_1(k-N) + d_{N+1}u_2(k-N) \\ c_{N+1}u_1(k-N+1) + d_{N+1}u_2(k-N+1) \\ \vdots \\ c_{N+1}u_1(k-N+P-1) + d_{N+1}u_2(k-N+P-1) \end{bmatrix}$$

$$+ \begin{bmatrix} d_1(k+1) \\ d_1(k+2) \\ \vdots \\ d_1(k+P) \\ d_2(k+1) \\ d_2(k+2) \\ \vdots \\ d_2(k+P) \end{bmatrix}. \tag{48}$$

This can be written, equivalently, in vector form as:

$$\mathbf{y}^{\text{lin}} = \mathbf{A}\boldsymbol{\Delta}\mathbf{u} + \mathbf{y}^{\text{past}} + \mathbf{d}, \tag{49}$$

$$d_j(k+i) = d_j(k), \quad i = 1, 2, \cdots, P, \quad j = 1, 2. \tag{50}$$

The control moves, $\boldsymbol{\Delta}\mathbf{u}$, are determined, according to the solution of the following optimization problem:

$$J = \min_{\Delta u} \left(\mathbf{y}^{\text{sp}} - \mathbf{y}^{\text{lin}}\right)^T \left(\mathbf{y}^{\text{sp}} - \mathbf{y}^{\text{lin}}\right) + \boldsymbol{\Delta}\mathbf{u}^T \boldsymbol{\Lambda}^T \boldsymbol{\Lambda} \boldsymbol{\Delta}\mathbf{u}. \tag{51}$$

$\boldsymbol{\Lambda}$ is the weighting matrix on the control effort. Under unconstraint minimization, the optimal input is determined as:

$$\Delta \mathbf{u} = \left(\mathbf{A}^T \mathbf{A} + \boldsymbol{\Lambda}^T \boldsymbol{\Lambda}\right)^{-1} \mathbf{A}^T \left(\mathbf{y}^{\text{sp}} - \mathbf{y}^{\text{past}} - \mathbf{d}\right). \tag{52}$$

Usually, the first component of each computed input vector variation is applied to the system and the same procedure is performed in the next sampling interval.

Convergence in EDMC for MIMO Systems

Results presented in the previous section are extendable to MIMO systems as well. Iterations in the fixed-point method are shown to be convergent (Theorem 1) based on the contraction-mapping theorem. This requires selection of small β and large $\boldsymbol{\Lambda}$. When Broyden's method (or some other QN family) is employed instead of the fixed-point iteration method, locally super linear convergence is guaranteed because of the intrinsic characteristic of the method [32].

Theorem 1

If the MIMO nonlinear system is globally asymptotically stable for all feasible inputs, then, the iteration (Equation 12) will converge if the sampling time and the weight on the inputs ($\boldsymbol{\Lambda} = \lambda \mathbf{I}$) are chosen large enough and the relaxation factor, β, is selected small enough.

Proof

Regarding the convergence condition in the fixed-point iteration method, the following matrices are defined.

$$\mathbf{A}_{T\to\infty} = \begin{bmatrix} a\mathbf{1}_L & b\mathbf{1}_L \\ c\mathbf{1}_L & d\mathbf{1}_L \end{bmatrix}. \tag{53}$$

So,

$$\mathbf{A} = \overline{\mathbf{A}}\mathbf{L} = \begin{bmatrix} a\mathbf{I} & b\mathbf{I} \\ c\mathbf{I} & d\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{1}_L & 0 \\ 0 & \mathbf{1}_L \end{bmatrix}. \tag{54}$$

The steady state gain matrix of the nonlinear system is defined as follows:

$$\mathbf{B} = \frac{\partial \mathbf{y}^{\text{nl}}}{\partial \boldsymbol{\Delta}\mathbf{u}} = \begin{bmatrix} a'_1 & \cdots & 0 & b'_1 & \cdots & 0 \\ 0 & \cdots & a'_P & 0 & \cdots & b'_P \\ c'_1 & \cdots & 0 & d'_1 & \cdots & 0 \\ 0 & \cdots & c'_P & 0 & \cdots & d'_P \end{bmatrix}_{2P\times 2P}. \tag{55}$$

Based on Equation 12, iterations are convergent if a norm of the gradient matrix, $\mathbf{f}'_1(\mathbf{d}_k^{\text{nl}})$, is less than one. Doing some simplifications, one can derive the following results (details of similar computations for SISO systems are given in [13]):

$$\mathbf{f}_1'(\mathbf{d}^{\rm nl}) = \frac{\partial \mathbf{f}_1}{\partial \mathbf{d}^{\rm nl}} = \mathbf{I} + \beta(\frac{\partial \mathbf{y}^{\rm nl}}{\partial \mathbf{d}^{\rm nl}} - \frac{\partial \mathbf{y}^{\rm el}}{\partial \mathbf{d}^{\rm nl}})$$
$$= \mathbf{I} + \beta(\frac{\partial \mathbf{y}^{\rm nl}}{\partial \boldsymbol{\Delta}\mathbf{u}}\frac{\partial \Delta\mathbf{u}}{\partial \mathbf{d}^{\rm nl}} - \mathbf{A}\frac{\partial \boldsymbol{\Delta}\mathbf{u}}{\partial \mathbf{d}^{\rm nl}} - \mathbf{I})$$
$$= (1-\beta)\mathbf{I} + \beta(\mathbf{B}-\mathbf{A})\frac{\partial \boldsymbol{\Delta}\mathbf{u}}{\partial \mathbf{d}^{\rm nl}}$$
$$= \mathbf{I} - \beta(\mathbf{B}-\mathbf{A})(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^T$$
$$= (1-\beta)\mathbf{I} - \beta(\overline{\mathbf{B}}\mathbf{L} - \overline{\mathbf{A}}\mathbf{L})(\mathbf{L}^T\overline{\mathbf{A}}^T\overline{\mathbf{A}}\mathbf{L} + \lambda^2\mathbf{I})^{-1}\mathbf{L}^T\overline{\mathbf{A}}^T$$
$$= (1-\beta)\mathbf{I} - \beta(\overline{\mathbf{B}}\overline{\mathbf{A}}^{-1} - \mathbf{I})\overline{\mathbf{A}}\mathbf{L}(\mathbf{L}^T\overline{\mathbf{A}}^T\overline{\mathbf{A}}\mathbf{L} + \lambda^2\mathbf{I})^{-1}\mathbf{L}^T\overline{\mathbf{A}}^T. \quad (56)$$

Using the matrix inversion lemma given in Equation 57, this relation is rearranged as in Equation 58,

$$\mathbf{u}(\mathbf{I} + \mathbf{u}^T\mathbf{u})^{-1}\mathbf{u}^T = \mathbf{u}\mathbf{u}^T(\mathbf{I} + \mathbf{u}\mathbf{u}^T)^{-1}, \quad (57)$$

$$\mathbf{f}_1'(\mathbf{d}^{\rm nl}) = (1-\beta)\mathbf{I}$$
$$- \beta(\overline{\mathbf{B}}\overline{\mathbf{A}}^{-1} - \mathbf{I})\overline{\mathbf{A}}\mathbf{L}\mathbf{L}^T\overline{\mathbf{A}}^T(\lambda^2\mathbf{I} + \overline{\mathbf{A}}\mathbf{L}\mathbf{L}^T\overline{\mathbf{A}}^T)^{-1}. \quad (58)$$

Using the following property:

$$\overline{\mathbf{A}}\mathbf{L}\mathbf{L}^T\ \overline{\mathbf{A}}^T = \mathbf{L}\mathbf{L}^T\ \overline{\mathbf{A}}\overline{\mathbf{A}}^T. \quad (59)$$

It is further simplified as:

$$\mathbf{f}_1'(\mathbf{d}^{\rm nl}) = (1-\beta)\mathbf{I}$$
$$- \beta(\overline{\mathbf{B}}\overline{\mathbf{A}}^{-1} - \mathbf{I})\mathbf{L}\mathbf{L}^T\left(\lambda^2(\overline{\mathbf{A}}\overline{\mathbf{A}}^T)^{-1} + \mathbf{L}\mathbf{L}^T\right)^{-1}. \quad (60)$$

The maximum singular value of the matrix is considered as its norm here. Therefore,

$$\boldsymbol{\sigma}^*\left(\mathbf{f}_1'(\mathbf{d}^{\rm nl})\right) \le 1-\beta$$
$$+\beta\sigma^*\left(\mathbf{I} - \overline{\mathbf{B}}\overline{\mathbf{A}}^{-1}\right)\frac{\boldsymbol{\sigma}^{*2}(\mathbf{L})}{\boldsymbol{\sigma}_*(\mathbf{L}\mathbf{L}^T + \lambda^2(\overline{\mathbf{A}}\overline{\mathbf{A}}^T)^{-1})}$$
$$\le 1-\beta+\beta\left(1-\frac{\boldsymbol{\sigma}^*(\overline{\mathbf{B}})}{\boldsymbol{\sigma}_*(\overline{\mathbf{A}})}\right)\frac{\boldsymbol{\sigma}^{*2}(\mathbf{L})}{\boldsymbol{\sigma}_*^2(\mathbf{L}) + \frac{\lambda^2}{\boldsymbol{\sigma}^{*2}(\overline{\mathbf{A}})}}. \quad (61)$$

By selecting a large λ and a small β, the maximum singular value of $\mathbf{f}_1'(\mathbf{d}^{\rm nl})$ would be less than 1 and, therefore, convergence of the fixed-point iterations in Equation 12 will be guaranteed. For some simplification, the following approximation can be used:

$$\boldsymbol{\sigma}^*(\mathbf{L}) = 2\times 2\times MP = 4MP,$$
$$\boldsymbol{\sigma}_*(\mathbf{L}) \in (0.25, 0.5). \quad (62)$$

Stability for MIMO Systems (with Infinite Sampling Time Assumption)

Stability of a closed loop 2×2 MIMO system is guaranteed under above convergence conditions and the positive definiteness of $\mathbf{D}_0 = \begin{bmatrix} a & c \\ b & d \end{bmatrix}\mathbf{G}$, which is stated in Theorem 2.

Theorem 2

Suppose that the nonlinear 2×2 MIMO system to be controlled is globally asymptotically stable for all feasible inputs, the weight on the change of the inputs is larger than zero, the sampling time is long enough and the steady state gain, $\mathbf{G}$, satisfies the following criteria, then the closed loop system is guaranteed to be nominally stable.

$$\mathbf{D}_0 = \begin{bmatrix} a & c \\ b & d \end{bmatrix}\mathbf{G} > \mathbf{0}. \quad (63)$$

Proof

As in [33], closed loop stability can be investigated via computing singular values of the derivative of the nonlinear operator, $\mathbf{N}$, that is defined as:

$$\mathbf{u}(k) = \mathbf{N}\left(\mathbf{u}(k-1)\right). \quad (64)$$

To determine $\mathbf{N}$, one starts from Equation 52. It is assumed that $\mathbf{d}_k^{\rm ext} = \mathbf{0}$ for a nominal system.

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \mathbf{e}_1^T\left(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{I}\right)^{-1}$$
$$\mathbf{A}^T(\mathbf{\Pi}\mathbf{y}_{k+1}^{\rm sp} - \mathbf{\Pi}\mathbf{y}_{k+1}^{\rm past} - \mathbf{\Pi}\mathbf{d}_{k+1}^{\rm nl}), \quad (65)$$

where:

$$\mathbf{e}_1^T = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{2\times 2M},$$

and:

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{1}_P & 0 \\ 0 & \mathbf{1}_P \end{bmatrix}_{2P\times 2}.$$

The convergent value of $\mathbf{d}_k^{\rm nl}$ is calculated in a similar way to that found in the section of Infinite Sampling Time and, then, it is replaced in Equation 65.

$$\mathbf{\Pi}\mathbf{d}_{k+1}^{\rm nl} = (\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{\Pi}\mathbf{y}_{k+1}^{\rm nl} - \mathbf{\Pi}\mathbf{y}_{k+1}^{\rm past}$$
$$- (\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{A}_0\mathbf{\Pi}\mathbf{y}_{k+1}^{\rm sp}, \quad (66)$$

$$\mathbf{y}_{k+1}^{\rm nl} = \left[y_1^{\rm nl}(k+1) \quad y_2^{\rm nl}(k+1)\right],$$
$$\mathbf{y}_{k+1}^{\rm sp} = \left[y_1^{\rm sp}(k+1) \quad y_2^{\rm sp}(k+1)\right]. \quad (67)$$

Equation 66 is simplified as:

$$\mathbf{\Pi d}^{\mathrm{nl}}_{k+1}=(\mathbf{I}+\frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T)\mathbf{\Pi y}^{\mathrm{nl}}_{k+1}-\frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T\mathbf{y}^{\mathrm{sp}}_{k+1}-\mathbf{\Pi y}^{\mathrm{past}}_{k+1}. \quad (68)$$

Substituting $\mathbf{\Pi d}^{\mathrm{nl}}_{k+1}$ into Equation 65 results in the following relation.

$$\mathbf{u}_k=\mathbf{u}_{k-1}+\mathbf{e}_1^T\left(\mathbf{A}^T\mathbf{A}+\lambda^2\mathbf{I}\right)^{-1}\mathbf{A}^T\Big(\mathbf{\Pi y}^{\mathrm{sp}}_{k+1}-\mathbf{\Pi y}^{\mathrm{past}}_{k+1} -\left(\mathbf{I}+\frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T\right)\mathbf{\Pi y}^{\mathrm{nl}}_{k+1}+\frac{1}{\lambda^2}\mathbf{A}\mathbf{A}^T\mathbf{y}^{\mathrm{sp}}_{k+1}+\mathbf{\Pi y}^{\mathrm{past}}_{k+1}\Big). \quad (69)$$

This relation is further simplified as:

$$\mathbf{u}_k=\mathbf{u}_{k-1}+\frac{1}{\lambda^2}\mathbf{e}_1^T\mathbf{A}^T\mathbf{\Pi}\left(\mathbf{y}^{\mathrm{sp}}_{k+1}-\mathbf{y}^{\mathrm{nl}}_{k+1}\right). \quad (70)$$

Relations obtained so far are based on open loop stability and $T\rightarrow\infty$ conditions. It can be shown that:

$$\mathbf{e}_1^T\mathbf{A}^T\mathbf{\Pi}=\begin{bmatrix} a & c\\ b & d\end{bmatrix}P=\hat{\mathbf{A}}P. \quad (71)$$

Then, $\mathbf{u}_k$ is:

$$\mathbf{u}_k=\mathbf{u}_{k-1}-\frac{P}{\lambda^2}\begin{bmatrix} a & c\\ b & d\end{bmatrix}\mathbf{y}^{\mathrm{nl}}_{k+1}+\frac{P}{\lambda^2}\begin{bmatrix} a & c\\ b & d\end{bmatrix}\mathbf{y}^{\mathrm{sp}}_{k+1}$$

$$=\mathbf{u}_{k-1}-\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{G}\mathbf{u}_k+\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{y}^{\mathrm{sp}}_{k+1}, \quad (72)$$

$$\left(\mathbf{I}+\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{G}\right)\mathbf{u}_k=\mathbf{u}_{k-1}+\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{y}^{\mathrm{sp}}_{k+1},$$

or:

$$\mathbf{u}_k=\left(\mathbf{I}+\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{G}\right)^{-1}\mathbf{u}_{k-1}+\left(\mathbf{I}+\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{G}\right)^{-1}\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{y}^{\mathrm{sp}}_{k+1}. \quad (73)$$

Now, one can determine $\mathbf{N}'$ from Equation 73.

$$\mathbf{N}'=\frac{\partial\mathbf{u}_k}{\partial\mathbf{u}_{k-1}}=\left(\mathbf{I}+\frac{P}{\lambda^2}\hat{\mathbf{A}}\mathbf{G}\right)^{-1}. \quad (74)$$

It can be seen that when the matrix:

$$\mathbf{D}_0=\begin{bmatrix} a & c\\ b & d\end{bmatrix}\mathbf{G}, \quad (75)$$

is positive definite, the closed loop system is stable. In the case of SISO systems, $\mathbf{D}_0$ is reduced to ag, which implies that sign changes in the steady state gain could result in the instability of the closed loop system. This is the same result that is given in the nominal stability theorem [13].

SIMULATION RESULT: POWER UNIT NONLINEAR MODEL

A power unit is simulated, along with a nonlinear dynamic model of an 160 MW oil fired drum boiler-turbine-generator unit, intended for overall wide range simulation [34-36]. This model represents a three-input, three-output, third-order nonlinear system. The inputs are the position of the valve actuators that control fuel mass flow rate (u_1 in pu), steam flow rate (u_2 in pu) and water flow rate (u_3 in pu). The outputs are the electric power (P_e in MW), drum steam pressure (P_r in kg/cm^2) and drum water level (L in m). The state variables are electric power, drum steam pressure and fluid (steam-water) density (ρf). The model is given by the following state equations:

$$\frac{dP_r}{dt}=0.9u_1-0.0018u_2P_r^{9.8}-0.15u_3,$$

$$\frac{dP_e}{dt}=(0.073u_2-0.016)P_r^{9.8}-0.1P_e,$$

$$\frac{d\rho f}{dt}=(141u_3-(1.1u_2-0.19)P)/85. \quad (76)$$

Drum water level is calculated using the following algebraic equation:

$$q_e=(0.85u_2-0.14)P_r+45.59u_1-2.51u_3-2.09,$$

$$\alpha_{\mathrm{cs}}=\frac{(1-0.00153\rho f)(0.8P_r-25.6)}{\rho f(1.0394-0.00123P_r)},$$

$$L=0.05(0.13\rho f+100\alpha_{\mathrm{cs}}+0.11q_e-67.97), \quad (77)$$

where α_{cs} is the steam quality and q_e is the evaporation rate (kg/s). The positions of the valve actuators are constrained to lie in the interval [0,1], while their rate of change (pu/s) is limited as follows:

$$-0.007\leq\frac{du_1}{dt}\leq 0.007,$$

$$-2.0\leq\frac{du_2}{dt}\leq 0.02,$$

$$-0.05\leq\frac{du_3}{dt}\leq 0.05. \quad (78)$$

At the load level of 66.65 MW, pressure of 108 kg/cm^2 and fluid density of 428 kg/m^3, the nominal inputs are found to be u_n = [0.34 0.69 0.43]. These values are selected as initial points for inputs and state variables. The responses of the Extended DMC are shown in Figure 2. The graph consists of three outputs and inputs versus time. As indicated, good tracking is obtained for both pressure set point changes in $t=100^{\mathrm{s}}$ and power demand in $t=200^{\mathrm{s}}$. Since no change in drum level set point is required, the controller tries to compensate level deviation. Results justify the usage of an algorithm for computing $\mathbf{d}^{\mathrm{nl}}$ in a MIMO nonlinear system with $M>1$.

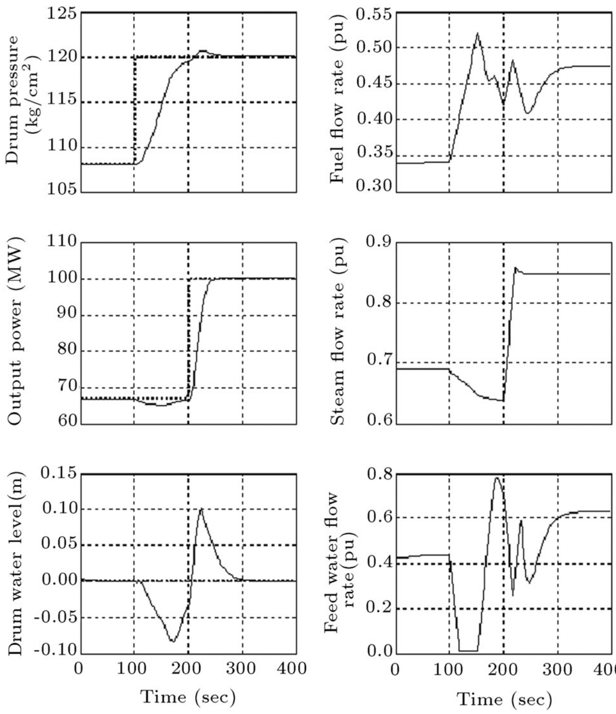

Figure 2. Responses of extended DMC for power unit.

CONCLUSION

In this paper, some extensions on EDMC are introduced. These extensions include application of the existing method on SISO, as well as MIMO systems, with longer control (M) and prediction (P) horizons. Analysis of convergence and closed loop stability, both in finite and infinite sampling time, are presented in detail and summarized in two theorems (Theorems 1 and 2). Simple conditions are obtained for some special cases, both for SISO and MIMO systems. Results given here confirm those in [13], which were obtained for a special case ($M = P = 1$ and $T \to \infty$). Computer simulation results for a power unit plant indicate that application of the method has good performances in set point tracking.

REFERENCES

1. Takatsu, H. and Itoh, T. "Future needs for control theory in industry-reports of the control technology survey in Japanese industry", *IEEE Transactions on Control Systems Technology*, **7**(3), pp 298-305 (1986).

2. Qin, S.J. and Badgwell, J.P. "An overview of nonlinear model predictive control applications", *Proceedings Chemical Process Control-V Assessment and New Directions for Research*, Tahoe City, Ca, USA (1996).

3. Qin, S.J. and Badgwell, J.P. "An overview of industrial model predictive control technology", Kantor, J.C., Garcia, C.E. and Cannahan, B., Eds., *Fifth International Conference on Chemical Process Control*, pp 232-256, AIChE and CACHE (1997).

4. Garcia, C.G., Prett, D.M. and Morari, M. "Model predictive control: Theory and practice - a survey", *Automatica*, **25**(3), pp 335-348 (1989).

5. Wright, G.T. and Edgar, T.F. "Nonlinear model predictive control of a fixed-bed water-gas shift reactor: An experimental study", *Computers and Chemical Engineering*, **18**, pp 83-102 (1994).

6. Kiparissides, C. and Georgiou, A. "Finite element solution of nonlinear optimal control problems with a quadratic performance index", *Computers and Chemical Engineering*, **11**, pp 77-81 (1987).

7. Biegler, L.T. "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation", *Computers and Chemical Engineering*, **8**, pp 243-248 (1984).

8. Renfro, J.G., Morshedi, A.M. and Asbjornsen, O.A. "Simultaneous optimization and solution of systems described by differential/algebraic equations", *Computers and Chemical Engineering*, **11**, pp 503-517 (1987).

9. Nagy, Z. and Agachi, S. "Model predictive control of a PVC batch reactor", *Computers and Chemical Engineering*, **21**(6), pp 571-591 (1997).

10. Yu, C., Roy, R.J., Kaufman, H. and Bequette, B.W. "Multi model adaptive predictive control of mean arterial pressure and cardiac output", *IEEE Transaction on Biomedical Engineering*, **39**(8), pp 765-778 (Aug. 1992).

11. Garcia, C.E. "Quadratic dynamic matrix control of nonlinear processes: An application to a batch reactor process", presented in *AIChE Meeting*, San Francisco, CA, USA (1984).

12. Gattu, G. and Zafiriou, E. "Nonlinear quadratic dynamic matrix control with state estimation", *Ind. Eng. Chem. Res.*, **31**, pp 1096-1104 (1992).

13. Peterson, T., Hernandez, E., Arkun, Y. and Schork, F.J. "A nonlinear DMC algorithm and its application to a semibatch polymerization reactor", *Chemical Engineering Science*, **47**(4), pp 737-742 (1992).

14. Peterson, T.J. "Nonlinear predictive control by an extended DMC", M.Sc. Thesis, Georgia Tech. (spring 1989).

15. Morshedi, A.M. "Universal dynamic matrix control", Morari, M. and McAvoy, T.J., Eds., *Third International Conference on Chemical Process Control*, pp 547-577, CACHE, Elsevier, New York, USA (1986).

16. Clarke, D.W. and Scattolini, R. "Constrained receding horizon predictive control", *IEE Proceedings-D*, **138**(4), pp 347-354 (July 1991).

17. Mosca, E. and Zhang, J. "Stable redesign of predictive control", *Automatica*, **28**(6), pp 1229-1233 (1992).

18. Kouvaritakis, B., Rossiter, J.A. and Chang, A.O.T. "Stable generalized predictive control: An algorithm with guaranteed stability", *IEE Proceedings-D*, **139**(4), pp 349-362 (July 1992).

19. Rawlings, J.B. and Muske, K.R. "The stability of constrained receding horizon control", *IEEE Transactions on Automatic Control*, **38**(10), pp 1512-1516 (Oct. 1993).
20. Rossiter, J.A., Gossner, J.R. and Kouvaritakis, B. "Infinite horizon stable predictive control", *IEEE Transactions on Automatic Control*, **41**(10), pp 1522-1527 (Oct. 1996).
21. De Nicolao, G. and Strada, S. "On the stability of receding horizon LQ control with zero- state terminal constraint", *IEEE Transactions on Automatic Control*, **42**(2), pp 257-260 (Feb. 1997).
22. Allgower, F., Badgwell, T.A., Qin, J.S., Rawlings, J.B. and Wright, S.J. "Nonlinear predictive control and oving horizon estimation-an introductory overview", Frank, P.M., Ed., *Advanced in Control*, Highlights of ECC'99, Springer, pp 391-449 (1999).
23. De Nicolao, G., Magni, L. and Scattolini, R. "Stability and robustness of nonlinear receding horizon control", Zhang, A. and Allgower, F., Eds., *Nonlinear Predictive Control*, Birkhauser Verlag, Boston, pp 3-23 (2000).
24. Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O.M. "Constrained nonlinear model predictive control: Stability and optimality", *Automatica*, **26**(6), pp 789-814 (2000).
25. Imsland, L., Findeisen, R., Bullinger, E., Allgower, F. and Foss, A. "A note on stability, robustness and performance of output feedback nonlinear model predictive control", *Journal of Process Control*, **13**, pp 633-644 (2003).
26. Chisci, L., Lombardi, A. and Mosca, E. "Dual receding horizon control of constrained discrete time systems", *European Journal of Control*, **2**, pp 278-285 (1996).
27. Michalska, H. and Mayne, D.Q. "Robust receding horizon control of constrained nonlinear systems", *IEEE Transactions on Automatic Control*, **38**, PP 1623-1632.
28. Ortega, J.M. and Rheinboldt, W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, USA (1970).
29. Schnabel, R.B. and Dennis, J.E., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Englewood Cliff., N.J. (1996).
30. Lee, J. and Lee, W.K. "Least squares updates for Quasi-Newton methods in process design and simulation", *Computers and Chemical Engineering*, **13**(6), pp 717-723 (1989).
31. Economou, C. "An operator theory approach to nonlinear controller design", Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA (1985).
32. Dennis, J.E. and More, J.J. "A characterization of super linear convergence and its application to Quasi-Newton methods", *Mathematics of Computation*, **28**(126), pp 549-560 (1974).
33. Li, W.C. and Biegler, L.T. "A constrained pseudo Newton control strategy for nonlinear systems", *Computers and Chemical Engineering*, **14**(4/5), pp 451-468 (1990).
34. Dime, R. and Lee, K.Y. "Boiler-turbine control system design using a genetic algorithm", *IEEE Transactions on Energy Conversion*, **10**(4) (Dec. 1995).
35. Ramirez, R.G. and Lee, K.Y. "Power plant coordination-control with wide-range control-loop interaction compensation", *IFAC* (2002).
36. Ramirez, R.G. and Lee, K.Y. "Wide range operation of a power unit via feed forward fuzzy control", *IEEE Transactions on Energy Conversion*, **15**(4) (Dec. 2000).