State-Space Models in Model Predictive Control

Michael Lundh and Mats Molander

ABB Automation Products AB
721 59 Västerås

Abstract:

This paper presents a new multivariable model predictive controller from ABB. Some of its more important features are that is has three degrees of freedom (independent tuning of responses to set-point changes, to measurable disturbances and to non-measurable disturbances/model-mismatch) and that it is based on discrete time state-space models obtained from a truly multivariable identification procedure. Model predictive controllers based on such models can, compared to model predictive controllers based on input-output models, offer substantial improvement in rejection of certain types of disturbances. An example will demonstrate this.

1 Introduction

Model predictive control, MPC, is a widely used industrial technique for advanced multivariable control. An overview is given in [QB 1996]. For processes with strong interaction between different signals MPC can offer substantial performance improvement compared with traditional single-input single-output control strategies. Model predictive control has been used for several decades, and has been accepted as an important tool in many process industry applications.

The purpose of this paper is twofold. It presents a new multivariable model predictive controller product, 3*d*MPC, from ABB Automation Products AB. Further we will show some benefits of using multivariable state space models in model predictive control.

The paper is organized as follows: Section 2 gives an overview of the product 3*d*MPC, section 3 describes briefly the underlying mathematics, section 4 focuses on identification, section 5 gives an example of one of the benefits of using state space models. Finally some conclusions are given in section 6.

2 Product Overview

This section gives an overview of 3*d*MPC, a new multivariable model predictive controller product from ABB Automation Products AB.

The controller 3*d*MPC uses a combination of feedback and feed-forward. The controller determines the manipulated outputs, based on actual measurements of process variables, and feed forward signals. The process variables can be assigned set-points that are the target for the feedback control law or they could just be used in the state-observer to improve the state estimate. The feed forward signals are measurable disturbances acting on the process that can be used for feed forward.

At each sample the manipulated outputs are obtained from a sequence of constrained optimizations. The loss functions in these problems penalizes a weighted sum of squared control-errors and moves in the manipulated outputs. The constraints in the optimization comes from limits on the manipulated outputs and from limits on the process variables.

The 3*d*MPC product contains both on-line and off-line components. The on-line components consist of algorithms for MPC, a data-logging function, and a function for the generation of excitation signals for identification. The on-line components also include graphical user interfaces for operator interaction and start and stop of the on-line algorithms. The off-line engineering tools for configuration, data pre-processing, modeling, tuning, and analysis are available as a MATLAB toolbox.

The 3*d*MPC product is intended to run on a PC under Windows NT. It is connected to an underlying instrumentation system through an OPC server. The graphical interface can run on the same PC or on another PC connected through a network.

2.1 Controller Functionality

This section lists some of the features of 3*d*MPC to give a glimpse of the functionality.

The controller works with *three degrees of freedom* (3d). This has motivated its name. Three degrees of freedom means that the controller can be configured to have different dynamic responses to set-point changes, to changes in measurable disturbances, and to other disturbances and to model mismatch. The two first functions are implemented as feed forward from set-points and from measurable disturbances. The third function is accomplished through feedback. Three optimization problems are solved with appropriate control error formulations.

After the dynamic optimizations has been performed, a *static optimization* algorithm is employed to drive the manipulated outputs towards desired values. This algorithm will only have effect in cases where there is ambiguity in how manipulated outputs are chosen in order to reach the desired targets for the process variables.

The inputs and outputs of the controller can operate in independent *operational modes*. This means that some signals in the controller can work in automatic mode while other signals can work in manual mode. The mode for a signal is determined either by the operator or by external inputs.

A main feature of model predictive controllers is the ability to *handle constraints*, not only for manipulated outputs but also for process variables. In 3*d*MPC, constraints ranked with priorities, can be defined for process variables, for deviations, and for manipulated outputs. These constraints can be violated under exceptional operating conditions. They are then violated according to their priorities. Hard constraints can be defined for manipulated outputs and their speed, and can never be violated.

The 3*d*MPC is based on a *discrete time state-space model*. The state vector is estimated using an observer. The parameters of the model are determined using truly multivariable identification methods provided by the modeling tools of the 3*d*MPC product.

Using state space models is not a goal per se, instead the reason for using state space models is that it provides some nice features for the model predictive controller:

- Uniform treatment of stable, integrating, and unstable processes.
- Allows feed-forward from non-measurable disturbances using extra measurement signals.

The controller is able to *handle non-linear processes* in two ways: Static non-linearities can be defined for each signal to compensate for known characteristics. The controller can also use an external input signal for parameter scheduling. One of four pre-defined sets of controller parameters is then selected. Such a set contains e.g a complete process model, weights on control-errors and moves, and constraint definitions. This high level controller parameter scheduling allows complete changes of the controller. The controller parameters can also be modified on-line by the operator.

The operator interface allows a user to have full insight in and control of 3*d*MPC through a number of displays. These are created automatically from the configuration.

2.2 Off-line Engineering Tools

The engineering tools are used to define and analyze a multivariable control system. The tools are available as a MATLAB toolbox with an elaborate graphical user interface. The most important tools are described below.

The *configuration tool* is used to create and modify the structure of the control system: Which signals should be used as manipulated outputs, as process variables, and as feed-forward variables. Properties associated with these signals are also defined using the configuration tool.

The *pre-processing tool* provides functions for signal processing to generate data sequences suitable for the process identification procedure in the modeling tool. The functionality include linearization functions to compensate for know non-linear elements, linear filtering, removal of means and trends, data editing to remove outliers and to split and merge data sequences, and finally there is also a function to resample data.

The *modeling tool* is used for building models of the process from data. There are modeling tools for process identification and for merging of models obtained from different process identifications. The tool also contains elaborate analysis functions to assist the user. The main part of the modeling tools are the parametric identification tool that is used for building state-space models from data sequences. The used two step state-of-the art algorithm for parametric identification combines the ease of use with the ability to use data sequences from both open and closed loop identification experiments. More details are found in Section 4.

The *tuning tool* is used to define the tuning parameters for the controller, e.g weights on control errors and moves, constraint definitions, and observer design parameters. The tool also contains elaborate analysis functions to assist the user.

3 MPC based on State-Space Models

A state space based model predictive controller, like 3*d*MPC, is described in this section. The controller design is based on a model of the open loop process.

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_d d(k) + w(k) \\ z(k) &= Cx(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{1}$$

where *y(k)* and *z(k)* are vectors with measured and noise free process variables, *u(k)* is the vector with manipulated outputs, and *d(k)* is the vector with measurable disturbances. The noise vectors *w(k)* and v*(k)* are assumed to be white noise sequences. It is assumed that the model (1) is stabilizable and detectable.

Integrators are introduced by using an extended state space model that uses the differentiated state vector $\Delta x(k) = x(k) - x(k-1)$ and the controlled outputs *z(k)* of (1). This gives

$$\begin{aligned} \begin{bmatrix} \Delta x(k+1) \\ z(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B_u \\ CB_u \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ CB_d \end{bmatrix} \Delta d(k) + \begin{bmatrix} I \\ C \end{bmatrix} \Delta w(k) \\ z(k) &= \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ z(k) \end{bmatrix} \\ y(k) &= z(k) + v(k) \end{aligned}$$

which in short notation can be written as

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k) + \bar{B}_u \Delta u(k) + \bar{B}_d \Delta d(k) + \Delta \bar{w}(k) \\ z(k) &= \bar{C}\bar{x}(k) \\ y(k) &= z(k) + v(k) \end{aligned} \tag{2}$$

The state vector is estimated using a state observer. It is based on the model (2). The observer is given by

$$\begin{aligned} \varepsilon(k) &= y(k) - \bar{C}\hat{x}(k|k-1) \\ \hat{x}(k+1|k) &= \bar{A}\hat{x}(k|k-1) + \bar{B}_u \Delta u(k) + \bar{B}_d \Delta d(k) + K\varepsilon(k) \end{aligned} \tag{3}$$

The observer (3) provides the one step ahead prediction of the extended state vector. Further predictions are obtained by repeated use of (2) with the assumption that $\Delta u(k) = 0, k > m$, $\Delta d(k) = 0, k > 1$, and $\varepsilon(k) = 0, k > 1$. Multiplication with $\bar{C}$ provides prediction of *z*, based on estimated state, actual measurements, and future manipulated output moves. The output vector is predicted *p* samples ahead (prediction horizon) and control actions are considered for *m* future samples, $m \le p$ (control horizon). To simplify the notation, introduce

$$U(k) = \begin{bmatrix} u(k) \\ : \\ u(k+m-1) \end{bmatrix}, \qquad Z(k) = \begin{bmatrix} z(k) \\ : \\ z(k+p-1) \end{bmatrix} \tag{4}$$

that collects manipulated outputs over the control horizon and process variables over the prediction horizon. Then the predicted process variables over the prediction horizon are

$$Z(k+1|k) = \begin{bmatrix} \bar{C}\bar{A} \\ \bar{C}\bar{A}^2 \\ : \\ \bar{C}\bar{A}^p \end{bmatrix} \hat{x}(k|k-1) + \begin{bmatrix} \bar{C}\bar{B}_u & 0 & \cdots & 0 \\ \bar{C}\bar{A}\bar{B}_u & \bar{C}\bar{B}_u & \ddots & 0 \\ : & & \ddots & \\ \bar{C}\bar{A}^{p-1}\bar{B}_u & \cdots & & \bar{C}\bar{A}^{(p-m)}\bar{B}_u \end{bmatrix} \Delta U(k) + \begin{bmatrix} \bar{C}\bar{B}_d \\ \bar{C}\bar{A}\bar{B}_d \\ : \\ \bar{C}\bar{A}^{(p-1)}\bar{B}_d \end{bmatrix} \Delta d(k) + \begin{bmatrix} \bar{C}K \\ \bar{C}\bar{A}K \\ : \\ \bar{C}\bar{A}^{(p-1)}K \end{bmatrix} \varepsilon(k) \tag{5}$$

which in short notation can be written as

$$Z(k+1|k) = S^x \hat{x}(k|k-1) + S^u \Delta U(k) + S^d \Delta d(k) + S^e \varepsilon(k) \tag{6}$$

The presence $\varepsilon(k)$ in (6) shows that the feedback is based on the most recent measurement of *y(k)*.

The control error over the prediction horizon is the difference between predictions and the trajectory of future set-points, i.e

$$E(k+1) = Z(k+1|k) - R_f(k+1)$$

The three degrees of freedom design is obtained by splitting the error function in three different parts,

$$E(k+1|k) = E_{sp}(k+1|k) + E_{ff}(k+1|k) + E_{fb}(k+1|k)$$

each with its own state vector definition. The first part is the set-point error, the second part is the feed forward error, and the third part is the remaining error. Three consecutive optimization problems are then solved for the three error functions to provide $\Delta U_{sp}(k)$, $\Delta U_{ff}(k)$, and $\Delta U_{fb}(k)$. These sum up to $\Delta U(k)$, the increments of the manipulated outputs.

Each optimization problem is of the form:

$$\text{Minimize } \left\| \begin{matrix} \Gamma E(k+1) \\ \Lambda \Delta U(k) \end{matrix} \right\|_2^2$$

with respect to constraints on predicted process variables and to constraints on manipulated outputs.

4 Identification

One main benefit with the observer based state space MPC implementation is that, in addition to known process inputs, all available process outputs are used when predicting one of the outputs, thus improving the prediction quality. Specifically this means that added secondary measurements can be used to help in the predictions of the controlled variables (See Section 5). For this to be possible, however, the model must include the necessary relationships. If, as is common in the MPC community, the model is constructed by merging single output models together, each measurement can only be used to predict itself.

Thus, the whole multivariable system must be identified at once, which complicates matters. The methodology used in the 3*d*MPC tools is a two-step procedure where first a preliminary state space model is identified using a subspace method. This model is then refined by a prediction error method. The procedure combines the relative simplicity of use for the subspace method with the theoretical advantages of the prediction error method (most important, the ability to produce unbiased results in closed loop).

4.1 Subspace Identification

Subspace identification comprises a whole family of algorithms. The algorithm used in 3*d*MPC has been developed by Peter Van Overschee and is described in detail in [VODM 96] (in the reference referred to as a robust algorithm for combined deterministic-stochastic identification).

Here, a very short sketch of the ideas behind the method is given, mostly to introduce some concepts used later. For a detailed description, see [VODM 96].

The purpose is to identify the matrices of the model (1) (here we simplify the input relationships by introducing the matrix $B = \begin{bmatrix} B_u & B_d \end{bmatrix}$ and including the measurable disturbances d in the input vector u). The first step is to set up the multi-step prediction equations

$$y_f(t) = L_1 y_p(t) + L_2 u_p(t) + L_3 u_f(t)$$
$$y_f(t) = \begin{bmatrix} y(t|t-1) \\ : \\ y(t+\gamma-1|t-1) \end{bmatrix} \quad y_p(t) = \begin{bmatrix} y(t-\beta) \\ : \\ y(t-1) \end{bmatrix} \quad u_p(t) = \begin{bmatrix} u(t-\beta) \\ : \\ u(t-1) \end{bmatrix} \quad u_f(t) = \begin{bmatrix} u(t) \\ : \\ u(t+\gamma-1) \end{bmatrix} \tag{7}$$

where β and γ are user supplied parameters. This predictor can easily be found through ordinary least-squares. To get the connection to the state space model, it is now assumed that the best estimate of the state vector $\hat{x}(t)$ based on the signal vectors $y_p(t)$ and $u_p(t)$ is available. The prediction of $y_f(t)$ can then alternatively be written

$$y_f(t) = \Gamma_\gamma \hat{x}(t) + H_\gamma u_f(t) \tag{8}$$

The extended observability matrix Γ_γ and H_γ containing Markov parameters, are easily constructed by iterating (1). By comparing (7) and (8), the following relation can be seen

$$\Gamma_\gamma \begin{bmatrix} \hat{x}(t_1) & \hat{x}(t_2) & \dots \end{bmatrix} = L_1 \begin{bmatrix} y_p(t_1) & y_p(t_2) & \dots \end{bmatrix} + L_2 \begin{bmatrix} u_p(t_1) & u_p(t_2) & \dots \end{bmatrix} \tag{9}$$

The left hand side is known, and if the state dimension is known[1], singular value decomposition can be used to find a column space of that dimension that can act as Γ_γ for some state representation. With Γ_γ known, the state sequence can in principle be extracted through (pseudo-)inversion of (9) and with the state sequence known, the system matrices appear linearly in (1) and can be solved for through least-squares.

To help the user in the choice of state dimension and of the parameters β and γ, the identification tool can perform scans over intervals of the parameters. All produced models are automatically evaluated by calculating a measure of the prediction errors over a prediction horizon. The result is presented in a list ranking the models and from where the user can select models for a more thorough evaluation.

4.2 Prediction Error Identification

In the prediction error method, each candidate model is used to calculate the optimal predictor (given the candidate is the true model) and by using this predictor, the prediction errors are calculated by looping through the data. The model giving the smallest value of a norm of the prediction errors is chosen.

Our use of the method can be summarized as**:**

Find A, B, C, K minimizing

$$\det\left(\sum_{t=1}^{N} e(t)e^{\mathrm{T}}(t)\right) \tag{10}$$

with

$$\begin{aligned} e(t) &= y(t) - C\hat{x}(t) \\ \hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + Ke(t) \end{aligned} \tag{11}$$

The search is started from the model found by the subspace method. Large residuals e are damped to decrease the sensitivity to outliers. The elements of B, C and K are all treated as independent parameters while for A a special tri-diagonal parameterization described in [McKHe 96] is used.

4.3 Identification using Merged Data

The 3*d*MPC tools contain functions to merge data collected at different occasions. The identification tools then uses information about the merging to perform the identification in a proper way.

For subspace identification, when solving the least squares problem to determine the predictor (7), equations containing values from more than one data set are removed. For the prediction error method, the loss function (10) is divided into partial sums, each sum involving only one data set, and with the state estimates of (11) re-initialized at the start of each set. Theoretical justifications for these ways of treating merged data can be found in [Lju 99].

When the results of different identification experiments are to be merged, the situation can occur that not all signals are covered by all the experiments. This situation is handled in 3*d*MPC by automatically filling in the missed parts of the data sets with zeros.

5 Example

To demonstrate the possibility to use secondary measurements to improve the predictions of the controlled variables, the process of Figure 1 is studied.

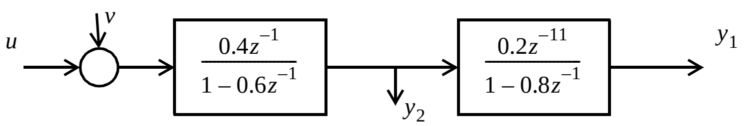

Figure 1 Process example.

The input u is the manipulated output and v is an non-measurable disturbance. The output y_1 is the controlled variable and y_2 is used only to improve the predictions. Simulated data were used to create two models, one by directly identifying a model with u as input and y_1 and y_2 as outputs (MO model) and the other by merging two single output models (SO model). The latter approach is a standard procedure when commissioning MPC controllers. The step-responses of the models are shown in Figure 2.

1.In principle, the singular values could be used to determine the rank (equalling the state dimension), but in practice, due to noisy measurements and the fact that in reality a low-dimensional approximation is desired, there is no simple rule to set a threshold for the values.

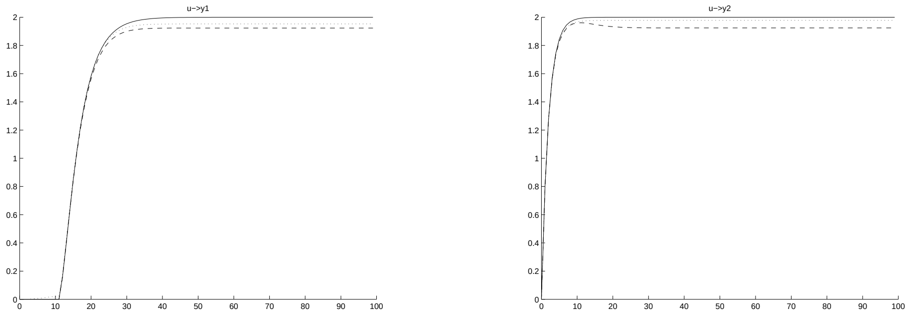

Figure 2 Step-responses of the model in Figure 1 (solid lines) together with the responses of the MO model (dashed lines) and of the SO model (dotted lines).

Two controllers with identical design parameters, having y_1 as the controlled variable were designed based on the two identified models. The result of a simulation is shown in Figure 3.

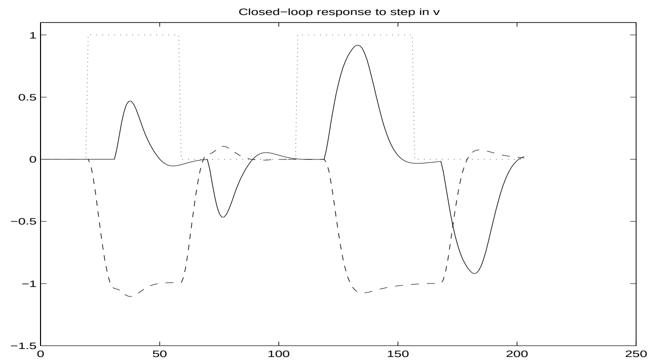

Figure 3 Closed-loop responses to steps in *v* (dotted line). Solid line shows y_1 and dashed line *u*. The first response is for the controller based on the MO model and the second is for the controller based on the SO model.

6 Conclusions

Although its long and widespread use, the MPC products have not adopted to modern state space theory, and hence not been able to benefit fully from what state space models can offer. The new product 3*d*MPC from ABB Automation Products changes this since it relies on state-space models obtained from a truly multivariable identification procedure. It further allows for independent tuning of responses to set-point changes. to measurable disturbances and to non-measurable disturbances/model-mismatch. The first two loops are purely feed-forward and can often be tuned tightly while still retaining robustness through a more cautious tuning of the feed-back loop.

7 References

[Lju 99] Ljung, L. *System Identification-Theory for the User, second edition.* Prentice Hall 1999.

[McKHe 96] McKelvey, T., Helmersson, A. *State-Space Parametrizations of Multivariable Linear Systems using Tridiagonal Matrix Forms.* CDC Kobe Japan 1996.

[QB 1996] Qin, S.J. and T. Badgwell, *An Overview of Industrial Model Predictive Control Technology*, Chemical Process Control - V, Tahoe, California., 1996

[VODM 96] Van Overschee P., DeMoor B. *Subspace Identification for Linear Systems: Theory-Implementation-Applications.* Kluwer Academic Publishers 1996.