Tuning of methods for offset free MPC based on ARX model representations

Huusom, Jakob Kjøbsted; Poulsen, Niels Kjølstad; Jørgensen, Sten Bay; Jørgensen, John Bagterp

Published in:
Proceedings of the American Control Conference

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Huusom, J. K., Poulsen, N. K., Jørgensen, S. B., & Jørgensen, J. B. (2010). Tuning of methods for offset free MPC based on ARX model representations. In *Proceedings of the American Control Conference* (pp. 2255-2360). IEEE.

2010 American Control Conference
Marriott Waterfront, Baltimore, MD, USA
June 30-July 02, 2010

Tuning of Methods for Offset Free MPC based on ARX Model Representations

Jakob Kjøbsted Huusom, Niels Kjølstad Poulsen, Sten Bay Jørgensen and John Bagterp Jørgensen

Abstract— **In this paper we investigate model predictive control (MPC) based on ARX models. ARX models can be identified from data using convex optimization technologies and is linear in the system parameters. Compared to other model parameterizations this feature is an advantage in embedded applications for robust and automatic system identification. Standard MPC is not able to reject a sustained, unmeasured, non zero mean disturbance and will therefore not provide offset free tracking. Offset free tracking can be guaranteed for this type of disturbances if Δ variables are used or if the state space is extended with a disturbance model state. The relation between the base case and the two extended methods are illustrated which provides good understanding and a platform for discussing tuning for good closed loop performance.**

I. INTRODUCTION

Model Predictive Control is a state of the art control technology which utilizes a model of the system in order to predict the process output over some future horizon and solve a quadratic optimization problem with the control signal as decision variables. Inequality constraints can be formulated for both manipulated variables and the process outputs. The first of the controls are implemented. After retrieving the next process output the problem is solved again for the next control etc. Early achievements and industrial implementations in Model Prediction Control include IDCOM [1] and Dynamic Matrix Control [2]. These early algorithms were based on step or impulse response models. More general linear input-output models structure, typical ARMAX, ARIMAX or CARIMA, were used by [3] in Generalized Predictive Control. ARMAX models can be identified using standard tools from time series analysis and systems identification. However, for MIMO systems it is difficult to select a structure for the ARMAX model. Furthermore, identification of the parameters in ARMAX models constitutes a non-linear non-convex optimization problem. If the input-output model is simplified to an ARX model, the optimization problem becomes a convex optimization problem and the MIMO system can be handled as easily as SISO systems. An interest in MPC implementations based on state space models were created by the seminal paper [4]. The state space approach provide a unified framework for discussion of the various predictive control algorithms and is well suited for stability analysis [5]. Direct identification of the parameters in a state space model is a non-linear non-convex parameter estimation problem. Therefore, model predictive control based on state space models is most useful as an implementation paradigm for the other linear model classes. This paper will use the following linear, discrete time, single input/single output ARX model

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \varepsilon(t) \tag{1a}$$

where $\varepsilon(t) \in \mathcal{N}_{iid}(0, \sigma^2)$ and A and B are polynomials of order n in the backwards shift operator q^{-1}.

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_n q^{-n} \tag{1b}$$
$$B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2} + \cdots + b_n q^{-n} \tag{1c}$$

This paper presents an introduction to MPC based on ARX models and discuses closed loop performance of the controller in case of unmeasured step disturbances. In order to reject such types of disturbances the basic ARX-MPC formulation needs to be expanded and different approaches to achieve offset free closed loop performance are investigated and discussed with respect to tuning parameters. The paper is organized as follows: An introduction is given to the basic ARX-MPC formulation and implementation in Section II. In order to reject unmeasured, sustained, non zero mean disturbances the ARX-MPC is expanded to the ΔARX description in Section III. Section IV presents the disturbance modelling approach. These formulations are discussed with respect to closed loop performance based on a set of simulation studies in Section V before the final conclusions are drawn.

A. A numerical test case

In the following a series of closed loop simulations with different MPC control implementations will be performed and compared in terms of performance on a numerical example. The example will use the same ARX-model as the true system and for the model in the MPC-controller. The model is

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \varepsilon(t) \tag{2a}$$

where $\varepsilon(t) \in \mathcal{N}_{iid}(0, \sigma^2)$ and

$$A(q^{-1}) = 1 - 2.4q^{-1} + 2.05q^{-2} - 0.63q^{-3} \tag{2b}$$
$$B(q^{-1}) = 0.5q^{-1} \tag{2c}$$
$$\sigma = 0.1 \tag{2d}$$

This model has a pole in 0.9 and a set of complex poles in $0.75 \pm 0.37i$.

J. K. Huusom and S. B. Jørgensen are with the Department of Chemical and Biochemical Engineering, Technical University of Denmark, Building 227. DK - 2800 Lyngby, Denmark jkh@kt.dtu.dk, sbj@kt.dtu.dk

N. K. Poulsen and J. B. Jørgensen are with the Department of Informatics and Mathematical Modelling, Technical University of Denmark, Building 321. DK - 2800 Lyngby, Denmark nkp@imm.dtu.dk, jbj@imm.dtu.dk

978-1-4244-7427-1/10/$26.00 ©2010 AACC

II. BASIC ARX-MPC

The ARX model (1) may be realized as a stationary state space model in innovation form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + K\varepsilon_k && \text{(3a)}\\ y_k &= Cx_k + \varepsilon_k && \text{(3b)} \end{aligned}$$

with the matrices (A,B,K,C) in observer canonical form

$$A = \left[\begin{array}{c|cccc} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -a_{n-1} & 0 & 0 & \cdots & 1 \\ \hline -a_n & 0 & 0 & \cdots & 0 \end{array}\right] \; B = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \; K = \begin{bmatrix} -a_1 \\ \vdots \\ -a_n \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

The optimal predictions in the stationary state space model in innovation form (3) is based on computation of the innovations

$$\varepsilon_k = y_k - \hat{y}_{k|k-1} \qquad (4)$$

using the measurement y_k at time k and the one-step-ahead prediction, $\hat{y}_{k|k-1} = C\hat{x}_{k|k-1}$. The one-step-ahead prediction of the states and outputs are

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k-1} + Bu_{k|k} + K\varepsilon_k && \text{(5a)}\\ \hat{y}_{k+1|k} &= C\hat{x}_{k+1|k} && \text{(5b)} \end{aligned}$$

and similarly the $(j+1)$-step-ahead $(j \geq 1)$ predictions are

$$\begin{aligned} \hat{x}_{k+1+j|k} &= A\hat{x}_{k+j|k} + Bu_{k+j|k} && j = 1, \ldots, N-1 && \text{(6a)}\\ \hat{y}_{k+1+j|k} &= C\hat{x}_{k+1+j|k} && j = 1, \ldots, N-1 && \text{(6b)} \end{aligned}$$

The ℓ_2-based constrained predictive controller use an objective function of the form

$$\phi = \frac{1}{2}\sum_{j=0}^{N-1} (\hat{y}_{k+1+j|k} - r_{k+1+j})^2 + \rho\Delta u_{k+j|k}^2 \qquad (7)$$

which obviously depends on the control variables, hence the optimal control problem is

$$\begin{aligned} \min_{\{u_{k+j|k}\}_{j=0}^{N-1}} \quad & \phi = \phi(\{u_{k+j|k}\}_{j=0}^{N-1}) && && \text{(8a)}\\ s.t. \quad & (5), (6) && && \text{(8b)}\\ & u_{\min} \leq u_{k+j|k} \leq u_{\max} && j \in \mathcal{N} && \text{(8c)}\\ & \Delta u_{\min} \leq \Delta u_{k+j|k} \leq \Delta u_{\max} && j \in \mathcal{N} && \text{(8d)} \end{aligned}$$

with $\Delta u_{k+j|k} = u_{k+j|k} - u_{k+j-1|k}$ $(j \in \mathcal{N})$, $u_{k-1|k} = \hat{u}_{k-1|k-1}$, and $\mathcal{N} = \{0, 1, \ldots, N-1\}$. The optimal solution is denoted $\{\hat{u}_{k+j|k}\}_{j=0}^{N-1}$. Only the first part of the solution, $\hat{u}_{k|k}$, is implemented on the process and the computations are repeated as new measurements arrive.

A. MPC as a Convex Quadratic Program

In this subsection we convert the constrained optimal control problem (8) to a standard convex quadratic program. To simplify the notation consider a horizon of $N = 4$. Define the vectors Y, R, U, and ΔU as

$$Y = \begin{bmatrix} \hat{y}_{k+1|k} \\ \hat{y}_{k+2|k} \\ \hat{y}_{k+3|k} \\ \hat{y}_{k+4|k} \end{bmatrix} \qquad R = \begin{bmatrix} r_{k+1} \\ r_{k+2} \\ r_{k+3} \\ r_{k+4} \end{bmatrix}$$
$$U = \begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ u_{k+2|k} \\ u_{k+3|k} \end{bmatrix} \qquad \Delta U = \begin{bmatrix} \Delta u_{k|k} \\ \Delta u_{k+1|k} \\ \Delta u_{k+2|k} \\ \Delta u_{k+3|k} \end{bmatrix}$$

With this notation the constraints $u_{\min} \leq u_{k+j|k} \leq u_{\max}$ for $j = 0, 1, 2, 3$ may be denoted as

$$U_{\min} \leq U \leq U_{\max} \qquad (9)$$

with

$$U_{\min} = \begin{bmatrix} u_{\min} \\ u_{\min} \\ u_{\min} \\ u_{\min} \end{bmatrix} \qquad U_{\max} = \begin{bmatrix} u_{\max} \\ u_{\max} \\ u_{\max} \\ u_{\max} \end{bmatrix}$$

ΔU may be expressed as

$$\Delta U = \Psi U - I_0 \hat{u}_{k-1|k-1} \qquad (10)$$

with

$$\Psi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \qquad I_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

such that the constraints $\Delta u_{\min} \leq \Delta u_{k+j|k} \leq \Delta u_{\max}$ for $j = 0, 1, 2, 3$ may be expressed as

$$b_l \leq \Psi U \leq b_u \qquad (11)$$

using

$$\begin{aligned} b_l &= \Delta U_{\min} + I_0 \hat{u}_{k-1|k-1} && \text{(12)}\\ b_u &= \Delta U_{\max} + I_0 \hat{u}_{k-1|k-1} && \text{(13)} \end{aligned}$$

and

$$\Delta U_{\min} = \begin{bmatrix} \Delta u_{\min} \\ \Delta u_{\min} \\ \Delta u_{\min} \\ \Delta u_{\min} \end{bmatrix} \qquad \Delta U_{\max} = \begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \Delta u_{\max} \\ \Delta u_{\max} \end{bmatrix} \qquad (14)$$

Using (5) and (6), the output, Y, may be expressed as

$$Y = b + \Gamma U \qquad (15)$$

in which b is defined as

$$b = \Phi_x \hat{x}_{k|k-1} + \Phi_\varepsilon \varepsilon_k \qquad (16)$$

and

$$\Gamma = \begin{bmatrix} H_1 & 0 & 0 & 0 \\ H_2 & H_1 & 0 & 0 \\ H_3 & H_2 & H_1 & 0 \\ H_4 & H_3 & H_2 & H_1 \end{bmatrix} \; \Phi_x = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ CA^4 \end{bmatrix} \; \Phi_\varepsilon = \begin{bmatrix} CK \\ CAK \\ CA^2K \\ CA^3K \end{bmatrix}$$

The impulse response parameters, H_i, are defined as

$$H_i = CA^{i-1}B \quad i = 1, 2, \ldots \tag{17}$$

Using these relations for Y and ΔU, the objective function may be expressed as

$$\begin{aligned}\phi &= \frac{1}{2}\|Y - R\|_2^2 + \frac{\rho}{2}\|\Delta U\|_2^2 \\ &= \frac{1}{2}\|b + \Gamma U - R\|_2^2 + \frac{\rho}{2}\|\Psi U - I_0\hat{u}_{k-1|k-1}\|_2^2 \\ &= \frac{1}{2}U'HU + g_k'U + \varrho_k \end{aligned} \tag{18a}$$

in which

$$H = \Gamma'\Gamma + \rho\Psi'\Psi \tag{18b}$$

$$g_k = \Gamma'(b - R) - \rho\Psi' I_0 \hat{u}_{k-1|k-1} \tag{18c}$$

$$\varrho_k = \frac{1}{2}\|b - R\|_2^2 + \frac{\rho}{2}\|I_0\hat{u}_{k-1|k-1}\|_2^2 \tag{18d}$$

Consequently, the constrained optimal control problem (8) may be expressed as the convex quadratic program

$$\min_U \quad \phi = \frac{1}{2}U'HU + g_k'U + \varrho_k \tag{19a}$$

$$s.t. \quad U_{\min} \leq U \leq U_{\max} \tag{19b}$$

$$b_l \leq \Psi U \leq b_u \tag{19c}$$

The computations in the predictive controller consists of an off-line part involving computation of the state space model $\{A, B, K, C\}$, the matrices $\{H, \Phi_x, \Phi_\varepsilon, \Gamma, \Psi\}$, and the vectors $\{U_{\min}, U_{\max}, \Delta U_{\min}, \Delta U_{\max}\}$. With these matrices available, it is straightforward to do the on-line computation, $u_k = \mu(\hat{x}_{k|k-1}, u_{k-1}, y_k, \{r_{k+j}\}_{j=1}^N)$. This computation is listed in Algorithm 1. Note that the optimization is independent of ϱ_k.

Algorithm 1 Basic ARX-MPC

```
Require: $\hat{x}_{k|k-1}$, $\hat{u}_{k-1|k-1}$, $y_k$, $\{r_{k+1+j}\}_{j=0}^{N-1}$
  Compute: $\varepsilon_k = y_k - C\hat{x}_{k|k-1}$
  Compute: $b = \Phi_x\hat{x}_{k|k-1} + \Phi_\varepsilon\varepsilon_k$
  Compute: $c = b - R$
  Compute: $g_k = \Gamma'c - \Psi'(I_0(\rho\hat{u}_{k-1|k-1}))$
  Compute: $b_l = \Delta U_{\min} + I_0\hat{u}_{k-1|k-1}$
  Compute: $b_u = \Delta U_{\max} + I_0\hat{u}_{k-1|k-1}$
  Solve
        $\min_U$     $\psi = \frac{1}{2}U'HU + g_k'U$
        $s.t.$       $U_{\min} \leq U \leq U_{\max}$
                     $b_l \leq \Psi U \leq b_u$
  for $U$.
  $\hat{u}_{k|k} = U(1)$.
  $\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + B\hat{u}_{k|k} + K\varepsilon_k$
  Return: $\hat{u}_{k|k}$ and $\hat{x}_{k+1|k}$
```

III. ΔARX-MPC

In this section a reformulation of the ARX-MPC problem is given which provides offset free tracking also when the system is subject to a sustained unmeasured disturbance with a non zero mean. For this type of disturbance it is convenient to expand the system description in (1) to a description with two different sources of external noise. One term $\varepsilon(t)$ is still zero mean Gaussian distributed random noise, but $d(t)$ is a non zero constant.

$$A(q^{-1})y(t) = B(q^{-1})u(t) + d(t) + \varepsilon(t) \tag{20a}$$

where $\varepsilon(t) \in \mathcal{N}_{iid}(0, \sigma^2)$ and

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_nq^{-n} \tag{20b}$$

$$B(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \cdots + b_nq^{-n} \tag{20c}$$

By assuming the process noise term to be integrated white noise, the non zero mean contribution can be removed.

$$\begin{aligned}\frac{1}{1 - q^{-1}}e(t) &= d(t) + \varepsilon(t) \Rightarrow \\ e(t) &= (1 - q^{-1})\varepsilon(t) = \varepsilon_k - \varepsilon_{k-1}\end{aligned}$$

since $d(t)$ is constant in time. Modelling the noise as integrated white noise in the ARX-model (20) gives

$$(1-q^{-1})A(q^{-1})y(t) = (1-q^{-1})B(q^{-1})u(t) + (1-q^{-1})\varepsilon(t)$$

Hence we can write this extended model as

$$\bar{A}(q^{-1})y(t) = \bar{B}(q^{-1})u(t) + e(t) \tag{21a}$$

where

$$\bar{A}(q^{-1}) = (1 - q^{-1})A(q^{-1}) \tag{21b}$$

$$\bar{B}(q^{-1}) = (1 - q^{-1})B(q^{-1}) \tag{21c}$$

$$e(t) = (1 - q^{-1})\varepsilon(t) \tag{21d}$$

This is not a standard ARX model since the ARX model is assumed to be driven by white noise [6]. However, the constant disturbance disappears in this ΔARX model structure. In fact the effect of a step in the disturbance $d(t)$ will for the ΔARX-MPC correspond to a impulse disturbance in $d(t)$ for the basic ARX-MPC. This property is due to the fact that Δ is an integrator on the signals in the original ARX-model and the introduction of an integrator gives the offset free tracking performance.

Implementing of the ΔARX-MPC is very similar to the standard case. Given a model of the system on the form (20), hence we know the coefficients in the $A(q^{-1})$ and $B(q^{-1})$ polynomials, the system is brought to the ΔARX form by multiplying Δ on the polynomials to achieve $\bar{A}(q^{-1})$ and $\bar{B}(q^{-1})$. The MPC implementations is then constructed from a state space transformation of the model (21) instead of (1) despite the fact that $e(t)$ is not white. This implementation gives the offset eliminating effect in the controller. The drawback is that the observer filter, which is designed as if the noise was white, is no longer optimal.

IV. DISTURBANCE MODELLING

In presence of unmeasured disturbances the classical approach to achieve offset free tracking performance for a model predictive control implementation, is to include disturbance states in the process model. This method was originally presented in [7] and a thorough presentation of disturbance models for linear model predictive control is given in [8] and [9] with conditions for detectability of the augmented systems. Given a general system description on state space form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B_d d_k + Gw_k \\ y_k &= Cx_k + C_d d_k + v_k \end{aligned} \tag{22}$$

It is assumed that the disturbance evolves as

$$d_{k+1} = d_k + \xi_k \tag{23}$$

where the noise in the system is given by the following Gaussian distribution

$$\begin{bmatrix} w_k \\ \xi_k \\ v_k \end{bmatrix} = \mathcal{N}_{iid}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & 0 & S \\ 0 & Q_\xi & 0 \\ S^T & 0 & R \end{bmatrix}\right) \tag{24}$$

The augmented system description becomes

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_k \\ \xi_k \end{bmatrix} \\ y_k &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + v_k \end{aligned} \tag{25}$$

The general idea is to use a state estimator with the augmented system in the model predictive controller. The prediction equations are

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_{k|k} + B_d \hat{d}_{k|k} \tag{26}$$

$$\hat{d}_{k+1|k} = \hat{d}_{k|k} \tag{27}$$

and the stationary Kalman filter are

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{d}_{k|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{d}_{k|k-1} \end{bmatrix} + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (y_k - C\hat{x}_{k|k-1} - C_d \hat{d}_{k|k-1}) \tag{28}$$

By an appropriate design of the gains $\{L_x, L_d\}$ the state estimator can estimate the unmeasured disturbance and render the controller capable of offset free tracking. If $\{B_d = 0, C_d = 1\}$ in (25) the model is referred to as the output disturbance model and in the opposite case where $\{B_d \neq 0, C_d = 0\}$ one achieve the state disturbance model. A special case of the state disturbance model is when $B_d = B$ and an input disturbance in modeled. For the ARX model structure in (20) the correct disturbance model is when $\{B_d = G = K, C_d = 1, Q = R = S = \sigma^2\}$. In [10] it is shown that any choice of disturbance model can give the same closed loop performance despite the nature of the disturbance. The requirement is that the disturbance covariance, used in calculation of the estimator gain, is estimated from the autocovariance of plant data. This result removes the focus from modelling the disturbance to estimation of the observer gain, which is a significant simplification.

An MPC implementation using the augmented system in (25) and the state estimator (28) will provide offset free tracking when the system is subjected to an unmeasured step disturbance. The level of the disturbance is estimated by the state estimator which allow the MPC to use this information in the predictions.

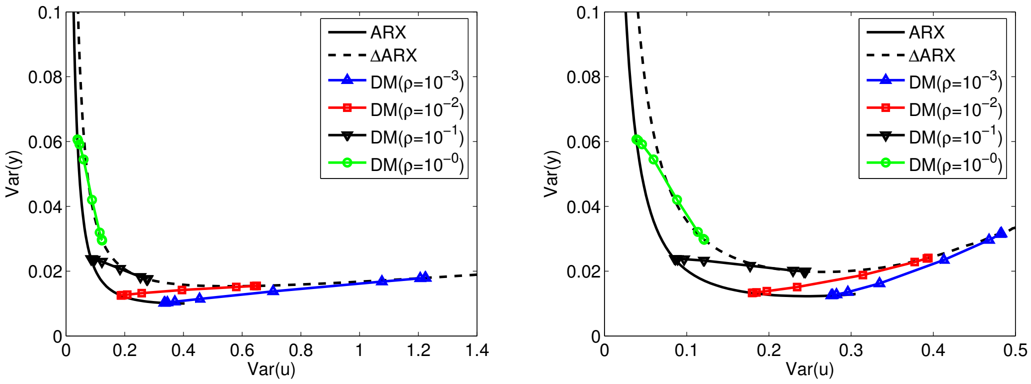
V. DISCUSSION

It is of interest to see how the presented MPC implementations perform, both with respect to rejection of zero mean random noise and for the case where a step disturbance enters the system. It is clear already from the given presentation that the methods differ in the degree of freedom the user needs to specify. For the basic ARX-MPC and the ΔARX-MPC, the only free parameter is the weight ρ in the cost function (7) which gives the relative penalty on the control move compared to the tracking error. When disturbance modelling is used one additional tuning parameter comes into play. That is the choice of the variance Q_ξ which will affect the gain L_d in the observer. First a formal relation between these three methods will be established. This will provide insight to the user with respect to tuning and which method to use. Secondly a set of closed loop simulations will show the performance of these methods.

A. Establishing a Formal Connection

It is clear that ARX-MPC and ΔARX-MPC differ since the Δ variable introduces a deliberate model mismatch which is necessary in order to reject non zero mean disturbances. As a consequence the variance of the noise $e(t)$ in the model (21) is twice that of the model (1) which is used in the basic ARX-MPC, hence the variance of the output from the ARX-MPC will always be lower than from the ΔARX-MPC. This is the price to be paid for offset free tracking.

The offset free tracking is achieved by different means when using disturbance modelling. The true dynamics of the system is not changed but the system is extended with an additional disturbance state and the observer corrects its value according to the prediction error. The design of the optimal observer given full process knowledge leaves Q_ξ as a free tuning variable. The free variable gives a tradeoff between fast estimation of an unknown step disturbance and the sensitivity to random noise. For $Q_\xi \to 0$, the optimal gain L_d is zero, the disturbance state is not updated. In this extreme the augmented system with the disturbance model becomes equivalent to the original systems and the performance is equal to the basic ARX-MPC. In the other extreme where Q_ξ is chosen very large to give very fast estimation of the step disturbance, the optimal gain L_d approaches 1. As a consequence the variance of the disturbance state is equal to that of the prediction error. The variance of the process output receives twice the variance of the prediction error additional to the state variance just as in the ΔARX-MPC implementation. Hence by looking at the extremes for the tuning of the disturbance modelling method we can see that for an invariant ρ the tuning of Q_ξ spans the range between ARXMPC and ΔARX-MPC. This is illustrated in Figure



(a) Unconstrained

(b) Constrained

Fig. 1. Pareto plot of the input and output variance of closed loop simulations with MPC implementations based on ARX, ΔARX and disturbance models. Different values of ρ in the performance cost function and for the variance $Q_\xi \in [10^{-7}; 10^2]$ in the disturbance model as indicated by the square, triangular and round markers on the curves. Both results from constrained and unconstrained MPC are given.

1 which present a series of closed loop simulations of all three methods varying ρ and Q_ξ. The performance is ploted in terms of the variance on the input and output in a Pareto plot. All simulations in the figure are based on the same seed of zero mean random noise and 10000 samples. From this figure it is easy to see the performances degradation rendered by the ΔARX model compared to the true model. This is expected since the ΔARX model is introduced in order to handle non zero mean disturbances which is not the scenario on the figure. Another unfortunate property of the ΔARX implementation is that for very low values of ρ, the output variance is increasing. This is in contrast to when the true model is used and the rationale behind tuning ρ in the cost function. Furthermore the four series of simulations with the disturbance model MPC implementation truly show that for extreme values of Q_ξ the line hit the curves for ARX-MPC and ΔARX-MPC for the corresponding value of ρ.

With respect to tuning of ρ for the different methods, Figure 1 is very useful. A good choice of the tuning parameter would render the variance of both the input and the output reasonable small. It is clear for the figure that a value which gave a point on the curves close to the origins is where they bend. For the unconstrained case $\rho = 0.01$ and $\rho = 0.1$ for the ARX-MPC and the ΔARX-MPC respectively seems to give the right trade off while reasonably values are $\rho = 0.1$ and $\rho = 1$ for the constrained case.

B. Closed loop performance

In the following the three MPC implementations will be tested on a fixed scenario. The total simulation horizon is 250 samples. Between time 50 and 100 a step is introduced in the reference and between 150 and 200 an unmeasured step disturbance is acting on the system. The input will be constrained between $u \in [-1; 1]$ but the control move is left unconstrained. In all simulations the value of ρ is 0.1 and for the disturbance modelling, DM-MPC, the variance $Q_\xi = 10^{-3}$. The closed loop response is shown on Figure 2. The random noise sequence used on all runs is kept the same in order to compare performance. The performance will be quantified by the following function which reflects the MPC performance cost but for the entire simulation horizon.

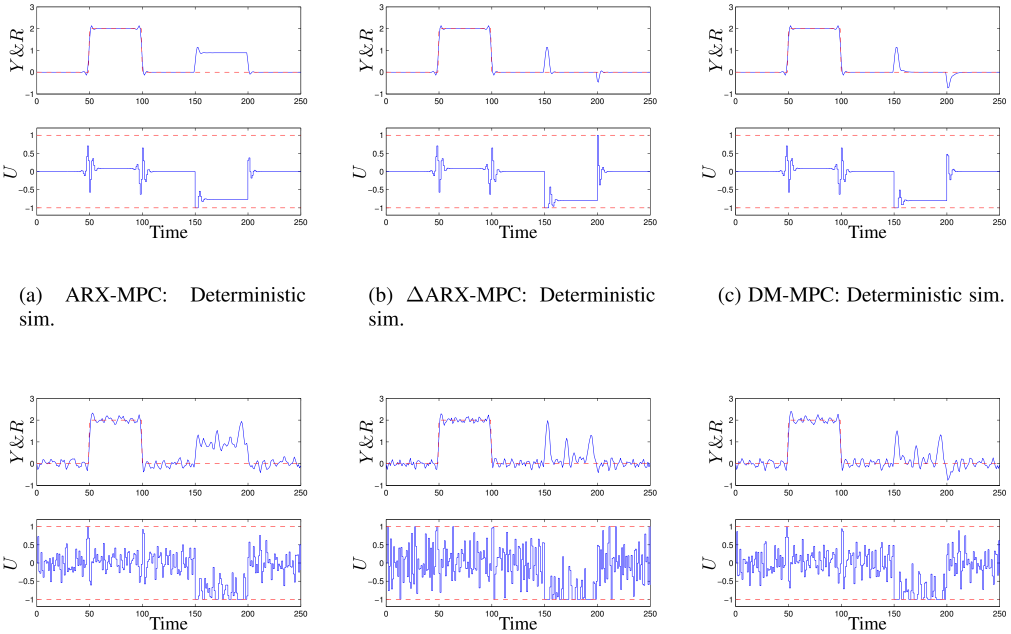
$$\bar{\phi} = \frac{1}{2(t_f - t_0)} \sum_{t=t_0}^{t_f} (y_t - r_t)^2 + \rho \Delta u_t^2 \qquad (29)$$

The results from the closed loop simulations are given in Table I. The closed loop simulations on Figure 2 show that the basic ARX-MPC is incapable of providing offset free tracking when the unmeasured step disturbance enters the system while the two other methods do reject this disturbance. From the figures in Table I we seen that in the deterministic case the ΔARX-MPC is superior while the disturbance modelling approach perform best in the presence of process noise. These results are natural since the tuning of Q_ξ gives the tradeoff between fast disturbance estimation and sensitivity to process noise. Hence for the deterministic simulation no noise is present and the best performance is achieved for $Q_\xi \to \infty$ which is the same as the ΔARX-MPC. When random noise is present in the system which is the case for all true systems the optimal performance is achieved by balancing the convergence of the disturbance estimate and its sensitivity to noise.

TABLE I

PERFORMANCE COST, $\bar{\phi}$, FOR THE CLOSED LOOP SIMULATIONS IN FIGURE 2 WITH ALL THREE MPC IMPLEMENTATIONS.

Method	Deterministic sim.	Noisy sim.
ARX-MPC	0.0895	0.1344
ΔARX-MPC	0.0195	0.0740
DM-MPC	0.0217	0.0589



(a) ARX-MPC: Deterministic sim.

(b) ΔARX-MPC: Deterministic sim.

(c) DM-MPC: Deterministic sim.

(d) ARX-MPC: Noisy sim.

(e) ΔARX-MPC: Noisy sim.

(f) DM-MPC: Noisy sim.

Fig. 2. Closed loop simulation of all three MPC implementations with $\rho = 0.1$ and $Q_\xi = 10^{-3}$. Two steps are induced in the reference signal and an unmeasured step disturbance of 0.4 is added between time 150 and 200.

VI. CONCLUSIONS

This paper presents three MPC implementations based on ARX model representations. The base case where the controller uses the true system for predictions and two descriptions, ΔARX-MPC and disturbance model MPC, which both provide offset free tracking in case of unmeasured non zero mean disturbances. It is shown that a the disturbance modelling approach has the ARX-MPC and the ΔARX-MPC as its extremes when tuning the disturbance state variance Q_ξ. Hence this extra tuning parameter renders a better closed loop performance since it explicitly balances the speed of convergence for the disturbance state and the sensitivity to noise in this estimate. The other free tuning parameter ρ in the MPC cost functions balances input versus output variance is common for all three implementations but it is seen by analyzing a set of Pareto plots that its tuning is dependent on the specific MPC implementation. This lead to the following main conclusion:

For systems with unmeasured non zero mean disturbances the disturbance model implementation in the MPC offers the best closed loop performance. When tuning this implementation, start by tuning the state variance Q_ξ to get the right balance between estimation power versus noise sensitivity. Information on the variance of the process noise and the expected size and frequency of these unknown step disturbances will be useful in this respect. Secondly, tune the weight in the cost function ρ to achieve a reasonable balance between input and output variance.

VII. ACKNOWLEDGMENTS

The first author gratefully acknowledge the Danish Council for Independent Research, Technology and Production Sciences (FTP) for funding through grand no. 274-08-0059

REFERENCES

[1] J. Richalet, A. Rault, J. L. Testud, and J. Papon, “Model predictive heuristic control: Application to industrial processes,” *Automatica*, vol. 14, no. 5, pp. 413 – 428, 1978.

[2] C. Cutler and B. Ramaker, “Dynamic matrix control - a computer control algorithm,” in *Proceedings of the Joint Automatic Control Conference*, 1980.

[3] D. Clarke, C. Mohtadi, and P. S. Tuffs, “Generalized predictive control - part i. the basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137 – 148, 1987.

[4] K. R. Muske and J. B. Rawlings, “Model predictive control with linear models,” *AIChE Journal*, vol. 39, no. 2, pp. 262 – 287, 1993.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[6] G. E. P. Box and G. M. Jenkins, *Time Series Analysis forecasting and control*. Holden-Day, 1970.

[7] E. J. Davison and H. W. Smith, “Pole assignment in linear time-invariant multivariable systems with consant disturbances,” *Automatica*, vol. 7, no. 4, pp. 489 – 498, 1971.

[8] K. R. Muske and T. A. Badgwell, “Disturbance modeling for offset-free linear model predictive control,” *Journal of Process Control*, vol. 12, no. 5, pp. 617 – 632, 2002.

[9] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE journal*, vol. 49, no. 2, pp. 426 – 437, 2003.

[10] M. R. Rajamani, J. B. Rawlings, and S. J. Qin, “Achiving state estimation equivalence for misassigned disturbances in offset-free model predictive control,” *AIChE Journal*, vol. 55, no. 2, pp. 396 – 407, 2009.