

Complexity Reduction in Explicit MPC through Model Reduction^{*}

Svein Hovland^{*} Jan Tommy Gravdahl^{*}

^{*} *Department of Engineering Cybernetics, Norwegian University of
Science and Technology, Trondheim, Norway (Tel: +4773594376,
e-mail: {svein.hovland,tommy.gravdahl}@itk.ntnu.no).*

Abstract: In this paper we propose to use model reduction techniques to make explicit model predictive control possible and more attractive for a larger number of applications and for longer control horizons. The main drawback of explicit model predictive control is the large increase in controller complexity as the problem size increases. For this reason, the procedure is limited to applications with low-order models, a small number of constraints and/or short control horizons. The proposed use of model reduction techniques is demonstrated for several applications, among others for control of fuel cell breathing. In all applications, a significant reduction in controller complexity is achieved.

1. INTRODUCTION

Model predictive control (MPC) is a control strategy that has been widely adopted in the industrial process control community and implemented successfully in many applications. The greatest strength of MPC is the intuitive way in which constraints can be incorporated in a multivariable control problem formulation. Fast implementation of model predictive control based e.g. on online optimization in real-time systems has been considered among others by Bleris and Kothare [2005]. However, the traditional MPC strategy demands a great amount of online computation, limiting the use of this kind of controllers to processes with relatively slow dynamics, since an optimization problem (often a constrained quadratic program) is solved at each sampling time.

It has recently been shown that a great deal of the computational effort in traditional MPC can be done offline. In Bemporad et al. [2002], the authors proposed solving multiparametric quadratic programs (mpQPs) that are used to obtain explicit solutions to the MPC problem, such that the control input can be computed by evaluating a piecewise affine function of the system state. Thus, the explicit model predictive controller (eMPC) accomplishes online MPC functionality without solving an optimization problem at each time step. This has several advantages: Firstly, the online computational time can be reduced to the microsecond-millisecond range, and secondly, MPC functionality is achieved with low complexity, easily verifiable real-time code. Further, execution is deterministic, and there is no need for floating point arithmetics (no recursive numerical computations). All these advantages justify the employment of eMPC in embedded and safety-critical systems.

The main drawback of eMPC is the large increase in both offline and online complexity as the size of the

system models grow larger and the control horizon and the number of constraints are increased. For this reason, the procedure is limited to models of relatively low order, typically with less than 10 states. This has motivated the use of complexity reduction techniques, such as input parametrization, as discussed in Tøndel and Johansen [2002].

The main contribution of this paper is the joint use of eMPC and rigorous model reduction techniques with upper bounds on the approximation error, thereby reducing the complexity of eMPC. This makes the control scheme attractive for a number of systems that would otherwise be excluded due to the high complexity of the resulting controllers.

Throughout the paper, positive (semi-) definiteness of matrices is indicated by $\succ 0$ ($\succeq 0$), $\mathcal{G}(s)$ denotes the transfer function of a linear time-invariant system, while $\|\cdot\|_\infty$ and $\|\cdot\|_2$ denote the \mathcal{H}_∞ norm of a system and the \mathcal{L}_2 norm of a signal, respectively. The identity matrix is denoted by I_n , where subscript n denotes the matrix dimension.

The paper is organized as follows: In Section 2 we briefly introduce model reduction by balanced truncation. Section 3 describes model predictive control, and in particular the explicit solution in 3.3. In Section 4 we discuss issues regarding reduced-order MPC, before we demonstrate the proposed methodology in examples in Section 5. Concluding remarks are made in Section 6.

2. MODEL REDUCTION BY BALANCED TRUNCATION

The goal of model reduction is to derive a model of low order that approximates the input-output behavior of the original model in an appropriate manner. In this work we use the balanced truncation introduced by Moore [1981] to get models of reduced order, although other model reduction techniques such as optimal Hankel model reduction or LQG balanced truncation could also be used.

^{*} This work was supported by the Research Council of Norway, through the strategic university program Computational Methods in Nonlinear Motion Control.

Balanced truncation is known to preserve stability; given that the original model is asymptotically stable, balanced truncation produces asymptotically stable reduced models. Controllability and observability are also preserved in the model reduction process [Moore, 1981].

We consider stable, linear time invariant (LTI) systems $\mathcal{G}(s)$ of the standard form

$$\dot{x} = Ax + Bu \quad (1a)$$

$$y = Cx, \quad (1b)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ denote the state, input and output, respectively, and the matrices A , B and C are of appropriate dimensions. Loosely speaking, balanced truncation is done by truncating states that give the least contribution to the input-output behavior. This motivates considering the controllable and observable subspaces of the state space. The controllable subspace contains the set of states that can be reached with zero initial state and a given input $u(t)$, while the observable subspace comprises those states that, as initial conditions, can produce a non-zero output $y(t)$ without external input. The controllability and observability grammians \mathcal{P} and \mathcal{Q} are $n \times n$ matrices whose eigenvectors span the controllable and observable subspaces, respectively. The grammians can be found by solving the Lyapunov equations

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \quad (2)$$

$$A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0. \quad (3)$$

A system is said to be *balanced* when the states that are excited most by input are at the same time the states that produce the most output energy. In such a realization, the grammians are both equal to a diagonal matrix, say Σ , with the elements σ_i on the diagonal in descending order,

$$\mathcal{P} = \mathcal{Q} = \Sigma. \quad (4)$$

The diagonal elements σ_i are called the system's Hankel singular values. A balancing transformation T that brings the system (1) on this form can be found as follows [Obinata and Anderson, 2001]: Assuming that (1) is a minimal realization (controllable and observable), \mathcal{P} and \mathcal{Q} can be decomposed as

$$\mathcal{P} = U_c S_c U_c^T \quad (5)$$

$$\mathcal{Q} = U_o S_o U_o^T, \quad (6)$$

where U_c and U_o are orthogonal transformations and S_c and S_o are diagonal matrices. The matrix

$$H = S_o^{1/2} U_o^T U_c S_c^{1/2} \quad (7)$$

is constructed, and its singular value decomposition is found as

$$H = U_H S_H V_H^T. \quad (8)$$

The balancing transformation is then given by

$$T = U_o S_o^{-1/2} U_H S_H^{1/2}, \quad (9)$$

and the balanced realization is given by

$$\bar{A} = T^{-1}AT, \quad \bar{B} = T^{-1}B, \quad \bar{C} = CT. \quad (10)$$

Finally, the balanced system matrices can be partitioned conformally as

$$\bar{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \bar{C} = [C_1 \ C_2], \quad (11)$$

and a truncated r th order model $\mathcal{G}_r(s)$ can be found as

$$\dot{x}_r = A_r x_r + B_r u \quad (12a)$$

$$y_r = C_r x_r, \quad (12b)$$

where $x_r \in \mathbb{R}^r$, $y \in \mathbb{R}^p$ and $A_r \triangleq A_{11} \in \mathbb{R}^{r \times r}$, $B_r \triangleq B_1 \in \mathbb{R}^{r \times m}$ and $C_r \triangleq C_1 \in \mathbb{R}^{p \times r}$ correspond to the states with large Hankel singular values (σ_i , $i = 1, \dots, r$). Provided that the truncated σ_i s are small, the corresponding truncated states are minimally excited by input, and at the same time contribute only slightly to the output. The error introduced by balanced truncation is upper bounded by

$$\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_\infty \leq 2 \sum_{k=r+1}^n \sigma_k. \quad (13)$$

This means that the error is equal to twice the sum of the truncated Hankel singular values. The error can also be represented in terms of a time-domain output error,

$$\|y(t) - y_r(t)\|_2 \leq 2 \sum_{k=r+1}^n \sigma_k \|u(t)\|_2. \quad (14)$$

3. MPC AND MULTIPARAMETRIC QUADRATIC PROGRAMMING

A brief outline of the standard MPC formulation will be given before we address the explicit solution.

3.1 A standard MPC formulation

The plant under consideration is modeled by the discretized state space model

$$x_{k+1} = A_d x_k + B_d u_k, \quad (15a)$$

$$y_k = C_d x_k, \quad (15b)$$

where $k \in \mathbb{Z}$, and $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $y_k \in \mathbb{R}^p$ denote the state, input and output, respectively, at step k . The matrices A_d , B_d and C_d are of appropriate dimensions. For the regulator problem, the model predictive controller solves at time step k the optimization problem

$$\min_{U_k} \left\{ \mathcal{J}(U_k, x_k) = x_{k+\mathcal{N}|k}^T P x_{k+\mathcal{N}|k} + \sum_{i=0}^{\mathcal{N}-1} \left(x_{k+i|k}^T Q x_{k+i|k} + u_{k+i}^T R u_{k+i} \right) \right\} \quad (16a)$$

subject to:

$$u_{\min} \leq u_{k+i} \leq u_{\max}, \quad i = 0, \dots, \mathcal{M} - 1 \quad (16b)$$

$$y_{\min} \leq y_{k+i} \leq y_{\max}, \quad i = 1, \dots, \mathcal{N} \quad (16c)$$

$$u_{k+1} = K x_{k+i|k}, \quad \mathcal{M} \leq i \leq \mathcal{N} - 1 \quad (16d)$$

$$x_{k|k} = x_k \quad (16e)$$

$$x_{k+i+1|k} = A_d x_{k+i|k} + B_d u_{k+i}, \quad i \geq 0 \quad (16f)$$

$$y_{k+i|k} = C_d x_{k+i|k}, \quad k \geq 0, \quad (16g)$$

where P , Q and R are weighting matrices of appropriate dimensions. P and Q penalize deviation from zero of the states x_{k+i} at the end of the prediction horizon and over the entire horizon, respectively, and R penalizes use of control action u . In this work, the final cost matrix P and gain K are calculated from the algebraic Riccati equation, under the assumption that the constraints are not active for $k \geq \mathcal{N}$. The sequence $U_k = [u_k^T \ u_{k+1}^T \ \dots \ u_{k+\mathcal{M}-1}^T]^T$ contains the future control inputs that yield the best predicted output with respect to the performance criterion on the prediction horizon \mathcal{N} , while \mathcal{M} denotes the control horizon. Throughout this paper, we let $\mathcal{M} = \mathcal{N}$, for

simplicity. Once the set U_k has been found, the first control input u_k is applied to the process, before the whole optimization is repeated at the next time step. The optimization problem is then slightly different, having been updated by a new process measurement, a new starting point and an additional time slice at the end of the horizon. It is assumed that the weighting matrices in (16a) are such that $P \succeq 0$, $R \succ 0$ and $Q \succeq 0$, and that (A_d, B_d) form a stabilizable pair.

3.2 Computing the control input

Sensitivity analysis is a technique used to describe how the solution to a mathematical program changes with small changes in the problem parameters. Closely related is parametric programming, where the solution is found explicitly for a range of parameter values. Mathematical programs that contain more than a single parameter are commonly referred to as multiparametric programs [Tøndel, 2003].

It is well established that implementing a model predictive controller requires solving a quadratic program in U_k at each time step [Maciejowski, 2001]. With some manipulations, the problem in (16) can be written

$$\min_{U_k} \left\{ \frac{1}{2} U_k^T H U_k + x_k^T F U_k \right\} \quad (17a)$$

$$\text{subject to: } G U_k \leq W + E x_k, \quad (17b)$$

where the matrices H , F , G , W and E are functions of the weighting matrices P , Q , R and the bounds u_{\min} , u_{\max} , y_{\min} and y_{\max} . Further, since $H \succ 0$ the problem is strictly convex, and the Karush-Kuhn-Tucker conditions (KKT) are sufficient conditions for optimality, giving a unique solution U_k [Nocedal and Wright, 1999]. The problem (17) can be viewed as a multiparametric quadratic program (mpQP) in U_k , where x_k is a vector of parameters.

3.3 MPC via multiparametric quadratic programming

By defining

$$z \triangleq U_k + H^{-1} F^T x_k, \quad (18)$$

the problem in (17) can be transformed into

$$\min_z \left\{ \frac{1}{2} z^T H z \right\} \quad (19a)$$

$$\text{subject to: } G z \leq W + S x_k, \quad (19b)$$

which is a mpQP in z , parameterized by x_k . The matrix S is found as $S = E + G H^{-1} F^T$. Note that when the control horizon is increased, the number of elements in U_k increases, and consequently the number of free variables in the problem (19) increases. Also, a higher number of states leads to more parameters in the problem. By considering the KKT conditions for (19), the solution z^* can be shown to remain optimal in a neighborhood of x_k where the active set remains optimal. The region in which this active set remains optimal can be shown to be a polyhedron in the parameter space (that is, the state space) [Bemporad et al., 2002]. Then, (19) can be solved offline for the state space area of interest. Computing the control input at a time step k then becomes a straightforward task: Given the system state x_k , the optimal control inputs U_k are obtained through an affine mapping,

$$U_k = K_i x_k + k_i, \quad (20)$$

where the subscript i denotes the i th affine function. K_i and k_i are constant within each polyhedral region in the parameter space. The online effort is thus reduced from solving a potentially large optimization problem at each time step for traditional MPC, to evaluating a piecewise affine function of the current state. By implementing the piecewise affine function as a binary search tree, the online computational time is logarithmic in the number of polyhedral in the state space [Tøndel et al., 2003]. The online memory and processing requirements increase with the number of regions in the partition. This is therefore used in the following as a measure of complexity of the explicit model predictive controller, and a reduction in the number of regions is considered to be a reduction in controller complexity.

4. REDUCED-ORDER MPC

Reduced-order models will be used to design output-feedback controllers for the systems. The eMPC control input is computed based on the reduced state vector $x_r(k)$ at every iteration, and x_r must therefore be estimated by an observer, based on measurements from the plant (or the output of the original model). When we are dealing with output constraints, it is particularly important that the output of the reduced-order model is a good estimate of the plant output, in order to satisfy the output constraints for the plant. The observer(s) should therefore account for the approximation error in the reduced model.

A basic linear observer such as the Luenberger observer, does not account explicitly for uncertainties, that are amplified by the observer gain matrices. Consequently, the state estimate may not be accurate enough in the presence of model perturbation. We therefore follow common practice in the literature [Astrid et al., 2002, Muske and Rawlings, 1993], and use a Kalman filter, which is known to have desirable properties for systems with noise in outputs and state equations. The Kalman filter is here defined in terms of the discretized reduced model with added noise,

$$x_r(k+1) = A_r x_r(k) + B_r u(k) + \Gamma w(k) \quad (21a)$$

$$y_r(k) = C_r x_r(k) + v(k), \quad (21b)$$

where $v(k)$ and $w(k)$ are assumed to be zero mean white noise processes with covariance matrices $R_k = R_k^T \succ 0$ and $Q_k = Q_k^T \succ 0$, respectively, and where Γ defines the mapping between w and the different states. In this setup, the noise processes are expected to account for uncertainty in the state equations through $\Gamma w(k)$, and the uncertainty in the output through $v(k)$.

4.1 Stability, Feasibility and Constraint Fulfillment

A number of questions regarding robust stability, feasibility and robust constraint fulfillment arises when the reduced model is used to control the high-order model.

Since the explicit MPC solution is equivalent to the standard MPC solution, many methods for robust stability analysis techniques developed for standard MPC [Bemporad and Morari, 1999] can be used to conclude stability for the reduced-order eMPC in the presence of the uncertainty introduced through the model reduction process. Some

recent results on MPC stability in the presence of model uncertainty have been developed [Heath et al., 2005b, Heath and Wills, 2005, Heath et al., 2005a]. Also, tests for robust MPC stability of input-constrained systems with unstructured uncertainty have recently been established by Løvaas et al. [2007]. The applicability of these methods to establish robustness in the context of MPC with reduced-order models remains a challenging open research question, and it is outside the scope of the current paper. Instead we use the nominal model (the reduced model) for controller design, and address certain robustness issues during the design stage. While we do not explicitly analyze the robustness of the reduced model predictive controller, good performance is achieved by *ad hoc* tuning based on exhaustive simulations for ranges of operating conditions. In many cases this approach leads to better performance than using robust MPC techniques. Choosing the right robust MPC technique is an art, and much experience is necessary to make it work [Bemporad and Morari, 1999].

Given the uncertainty introduced through the model reduction process, one cannot guarantee that feasibility of the underlying optimization problem is maintained and that the constraints on the states/outputs are fulfilled. This problem can be handled through the use of soft constraints. Constraints on the states/outputs often represent operational desirables rather than fundamental operational constraints. In addition, from a practical point of view it does not make sense to use tight state constraints because of the presence of noise, disturbances and numerical errors. Relaxing the state constraints in effect removes the feasibility problem, at least for stable systems [Bemporad and Morari, 1999]. *Exact* penalty functions can be used to allow constraint violation only when absolutely necessary [Kerrigan and Maciejowski, 2000].

5. EXAMPLES

The proposed procedure will be demonstrated using 6 different random systems to illustrate the great potential for complexity reduction, and two specific examples to show performance when using reduced-order eMPC.

5.1 Example 1

Without considering approximation quality and closed-loop performance, 6 different random systems of order $n = 6$, with two inputs and two outputs have been considered. For all six systems, the inputs and outputs are constrained such that

$$|u_i| \leq 1, \quad i = 1, 2 \quad (22)$$

$$|y_i| \leq 1, \quad i = 1, 2 \quad (23)$$

and the control horizon is fixed at $\mathcal{M} = 4$. The resulting controller complexity is tabulated in Table 1.

The table shows that eMPC for the original system is very demanding, with $O(10^5)$ polyhedral in the state space partition. But by truncating only one state, the controller complexity is reduced to a manageable level, as the number of regions is reduced by two orders of magnitude.

5.2 Example 2

For a random stable LTI system of order $n = 15$, the input is constrained such that $|u| \leq 5$ and the output

System/r	3	4	5	6
1	603	1447	1487	117573
2	625	1549	1589	122675
3	519	1095	1145	109656
4	539	1125	1136	95896
5	537	1033	1755	116438
6	513	1461	2145	109711

Table 1. Example 1: Controller complexity (in terms of number of regions in the state) for 6 random systems with two inputs and two outputs, with upper and lower bounds on inputs and outputs.

is constrained such that $|y| \leq 1$. Figure 1 compares the complexity of the eMPC solution for different model orders r and different control horizons \mathcal{M} for this example. For all r and \mathcal{M} , we set $Q = 1 \times 10^3 \cdot C_r^T C_r$ and $R = 1 \times 10^{-3}$. The

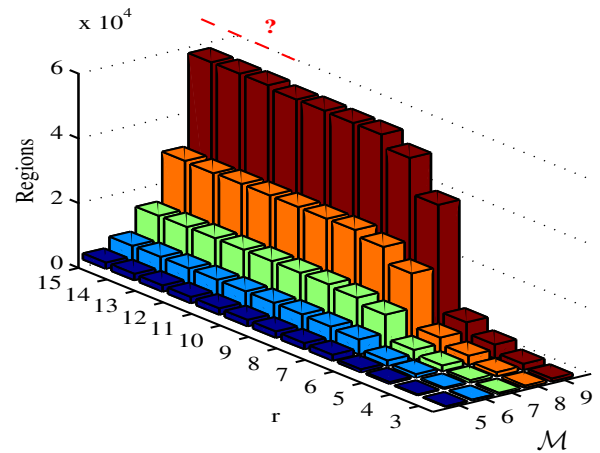


Fig. 1. Example 2: Complexity in terms of number of regions in the eMPC solution, for different model orders r and different control horizons \mathcal{M} . For $r = 13, 14$ and 15 , no solutions have been found with control horizon $\mathcal{M} = 9$, indicated by the dotted line and the question mark. The system order should be reduced to $r = 7$ or even $r = 6$ to obtain a significant reduction in complexity.

figure illustrates that the controller complexity increases by over an order of magnitude as we include more states in the reduced model and increase the control horizon \mathcal{M} . For $r = 3$, the number of regions ranges from 155 for $\mathcal{M} = 5$ to 1287 for $\mathcal{M} = 10$. For the original 15th order model, we are unable to compute the state space partition for $\mathcal{M} > 8$, due to the formidable computational requirement. The state space partition comprises 27442 regions for $\mathcal{M} = 8$. For $r = 12$, the number of regions in the state space partition is 55139 for $\mathcal{M} = 9$.

The model reduction error bound (13) is shown in Table 2, and illustrates the trade-off that must be made between controller complexity and quality of the reduced model, and consequently the quality of the resulting controller.

From Figure 1 it can be seen that by reducing the number of states down to 6, the controller complexity remains relatively low for the control horizons considered. We therefore generate our explicit model predictive controller using 6 states in the reduced model. For $r = 6$, the error bound is $\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_\infty \leq 6.7 \times 10^{-3}$. Still, the

r	Error bound
3	1.4×10^{-1}
4	7.4×10^{-2}
5	3.3×10^{-2}
6	6.7×10^{-3}
7	3.1×10^{-3}
8	1.5×10^{-4}
9	2.0×10^{-6}
10	3.5×10^{-7}
11	2.7×10^{-8}
12	4.5×10^{-10}
13	5.5×10^{-14}
14	4.3×10^{-17}

Table 2. Bound on model reduction error for Example 2.

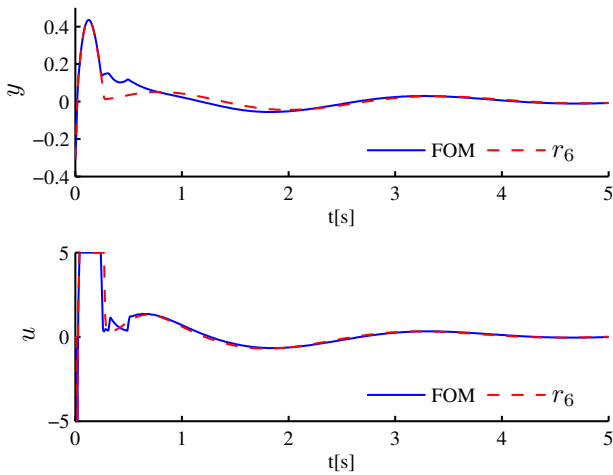


Fig. 2. Top: Output y of Example 2 with eMPC based on full order model (FOM) with $\mathcal{M} = 8$ and reduced-order model with $r = 6$ and $\mathcal{M} = 9$. The output is constrained between ± 1 . Bottom: Control input, constrained between ± 5 .

eMPC controller based on the 6th order reduced model is sufficient for control, as illustrated in Figure 2, where it can be seen that both the input and the output are kept within their bounds, when the plant is initialized with a representative non-zero state vector. The horizon length is $\mathcal{M} = 9$ and the explicit MPC solution based on the reduced-order model consists of 7625 polyhedral. The figure shows the performance of eMPC based on the full-order model, with a control horizon $\mathcal{M} = 8$, for which the controller consists of 27442 regions.

Although the error bound merely establishes a bound on the error between the two transfer functions in open loop, it does not guarantee performance, degree of sub-optimality and constraint satisfaction for the closed loop system, as discussed in Section 4.1. It is nevertheless an indication that a great reduction in complexity might be achieved without compromising the performance.

5.3 Example 3

This example is a linearized model for control of fuel cell breathing, as described in Pukrushpan et al. [2004].

r/\mathcal{M}	1	2	3	4	5
3	7	19	41	69	105
4	7	51	237	740	1813
5	7	55	333	1472	5020
6	7	55	331	1575	6068
7	7	57	393	2186	9964
8	7	61	445	2695	14999

Table 3. Controller complexity for Example 3. $r = 8$ corresponds to no model truncation ($r = n$).

r	Error bound
3	1.6×10^{-3}
4	1.3×10^{-4}
5	4.9×10^{-5}
6	4.4×10^{-6}
7	2.6×10^{-7}

Table 4. Bound on model reduction error for Example 3.

The model is a stable LTI with one input (compressor voltage), two outputs (system net power and oxygen excess ratio) and 8 states. Focusing on the methodology presented above, we use a slightly simplified version of the model in Pukrushpan et al. [2004], where we ignore disturbances and assume that the performance variables are measured. We discretize the model with sampling time $T_s = 1$ ms, and derive reduced-order models with $r = 3$ to $r = 7$ states. For these reduced models, we solve the eMPC offline problem for eMPC horizons 1-5, with bounds on the input and outputs:

$$|u| \leq 5, \quad |y_1| \leq 0.03, \quad |y_2| \leq 0.2. \quad (24)$$

We set the weight matrices to be $Q = 1000 \times C_r^T C_r$ and $R = 1$, where C_r is the output matrix corresponding to the reduced model used for controller design. The complexity of different eMPC controllers for this example is shown in Table 3, while the model reduction error bound (13) is shown in Table 4. It can be seen from Table 3 that the complexity of the controller increases rapidly for the original model ($r = 8$), while the increase is less pronounced for $r = 3$ and $r = 4$. The tables also show that by truncating 4 states, the controller complexity is reduced by an order of magnitude for $\mathcal{M} = 5$, at the cost of introducing an approximation error $\|\mathcal{G}(s) - \mathcal{G}_r(s)\|_\infty \leq 1.3 \times 10^{-4}$. If we reduce the number of states down to $r = 3$, the number of regions in the state space partition is reduced by over two orders of magnitude compared to the original model, for $\mathcal{M} = 5$. By truncating only one state, the controller complexity is reduced by 34% for $\mathcal{M} = 5$.

The simulation in Figure 3 shows the difference in closed loop behavior when using the full-order model with 8 states, and reduced-order models with 3 and 7 states.

In this simulation, the eMPC horizon is $\mathcal{M} = \mathcal{N} = 5$, which gives 105 regions in the controller for $r = 3$, 9964 regions for $r = 7$ and 14999 regions for the full-order model with 8 states. Moreover, it can be seen that both outputs remain within their bounds. The sub-optimality of the reduced-order controllers is clearly illustrated in the plot.

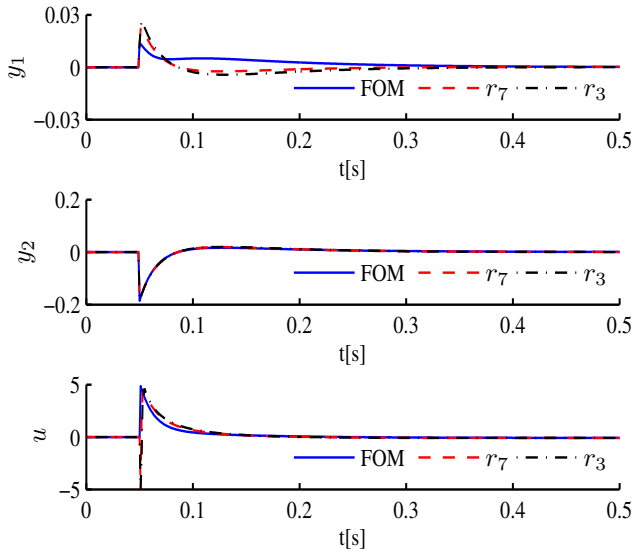


Fig. 3. Example 3: Closed-loop response to a disturbance at $t = 0.05$ s. The figure compares the performance for the full-order model (FOM), and reduced models with $r = 3$ (r_3) and $r = 7$ (r_7), all with $M = 5$.

6. CONCLUSIONS

It has been demonstrated that the performance of eMPC based on reduced-order models is of comparable quality to that of eMPC for the original systems. It is possible to use longer control horizons, while at the same time keeping the controller complexity low, at the cost of some controller sub-optimality. The degree of complexity reduction depends on the application, but is shown to be significant in all our examples. For input-constrained and soft-constrained systems, the approach is especially attractive, since the requirements to satisfy the output constraints need not be met. However, further work focuses on developing guarantees for satisfaction of output constraints.

ACKNOWLEDGEMENTS

The authors thank Petter Tøndel for providing mpQP software.

REFERENCES

P. Astrid, L. Huisman, S. Weiland, and A. C. P. M. Backx. Reduction and predictive control design for a computational fluid dynamics model. In *Proc. 41st IEEE Conf. on Decision and Control*, volume 3, pages 3378–3383, Las Vegas, NV, 2002.

A. Bemporad and M. Morari. Robust Model Predictive control: A survey. In A. Garulli, A. Tesi, and A. Vicino, editors, *Robustness in Identification and Control*, number 245 in Lecture Notes in Control and Information Sciences, pages 207–226. Springer, 1999.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

LG Bleris and MV Kothare. Real-time implementation of model predictive control. In *American Control Conference*, pages 4166–4171, 2005.

W.P. Heath, G. Li, A.G. Wills, and B. Lennox. IQC analysis of linear constrained MPC. In *IEEE sponsored Colloquium on Predictive Control*, University of Sheffield, 2005a.

W.P. Heath and A.G. Wills. The inherent robustness of constrained linear model predictive control. In *16th IFAC World Congress*, Prague, 2005.

W.P. Heath, A.G. Wills, and J.A.G. Akkermans. A sufficient stability condition for optimizing controllers with saturating actuators. *International Journal for Robust and Nonlinear Control*, 15(12):515–529, 2005b.

E. C. Kerrigan and J.M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *Proc. UKACC International Conference (Control 2000)*, 2000.

C. Løvaas, M. M. Seron, and G. C. Goodwin. Robust Model Predictive Control of Input-Constrained Stable Systems with Unstructured Uncertainties. In *Proc. European Control Conference*, 2007.

J. M. Maciejowski. *Predictive Control with Constraints*. Pearson Education, 2001.

Bruce C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. on Automatic Control*, 26(1):17–32, 1981.

Kenneth R. Muske and James B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2): 262–287, 1993. doi: 10.1002/aic.690390208.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

Goro Obinata and Brian D.O. Anderson. *Model Reduction for Control System Design*. Applied Mechanics Reviews. Springer, 2001.

J.T. Pukrushpan, A.G. Stefanopoulou, and H. Peng. Control of Fuel Cell Breathing: Initial Results on the Oxygen Starvation Problem. *IEEE Control Systems Magazine*, 24(2):30–46, 2004.

P. Tøndel. *Constrained Optimal Control via Multiparametric Quadratic Programming*. PhD thesis, Department of Engineering Cybernetics, Norw. University of Sci. and Techn., 2003.

P. Tøndel, T. A. Johansen, and A. Bemporad. Computation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, 2003.

P. Tøndel and T.A. Johansen. Complexity Reduction in Explicit Linear Model Predictive Control. In *XV IFAC World Congress*, 2002.