

# E

## Explicit Model Predictive Control



Alberto Bemporad  
IMT Institute for Advanced Studies Lucca,  
Lucca, Italy

### Abstract

Model predictive control (MPC) has been used in the process industries for more than 40 years because of its ability to control multivariable systems in an optimized way under constraints on input and output variables. Traditionally, MPC requires the solution of a quadratic program (QP) online to compute the control action, sometimes restricting its applicability to slow processes. Explicit MPC completely removes the need for online solvers by precomputing the control law off-line, so that online operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller is equivalently expressed as a lookup table of linear gains, a form that is extremely easy to code, requires only basic arithmetic operations, and requires a maximum number of iterations that can be exactly computed a priori.

### Keywords

Constrained control; Embedded optimization;  
Model predictive control; Multiparametric  
programming; Quadratic programming

### Introduction

Model predictive control (MPC) is a well-known methodology for synthesizing feedback control laws that optimize closed-loop performance subject to prespecified operating constraints on inputs, states, and outputs (Mayne and Rawlings 2009; Borrelli et al. 2017). In MPC, the control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant. Each optimization yields a sequence of optimal control moves, but only the first move is applied to the process: At the next time step, the computation is repeated over a shifted time horizon by taking the most recently available state information as the new initial condition of the new optimal control problem. For this reason, MPC is also called “receding horizon control.” In most practical applications, MPC is based on a linear discrete-time time-invariant model of the controlled system and quadratic penalties on tracking errors and actuation efforts; in such a formulation, the optimal control problem can be recast as a quadratic programming (QP) problem, whose linear term of the cost function and right-hand side of the constraints depend on a vector of parameters that may change from one step to another (such as the current state and reference signals). To enable the implementation of MPC in real industrial products, a QP solution method must be embedded in the control hardware. The method must be fast enough to provide a solution within short sampling intervals and require

simple hardware, limited memory to store the data defining the optimization problem and the code implementing the algorithm itself, a simple program code, and good worst-case estimates of the execution time to meet real-time system requirements.

Several *online* solution algorithms have been studied for embedding quadratic optimization in control hardware, such as active-set methods (Ricker 1985; Bemporad 2016), interior-point methods (Wang and Boyd 2010), fast gradient-projection methods (Patrinos and Bemporad 2014), and the alternating direction method of multipliers (Banjac et al. 2017). Explicit MPC takes a different approach to meet the above requirements, where multiparametric quadratic programming is proposed to pre-solve the QP *off-line*, therefore converting the MPC law into a continuous and piecewise-affine function of the parameter vector (Bemporad et al. 2002b; Bemporad 2015). We review the main ideas of explicit MPC in the next section, referring the reader to Alessio and Bemporad (2009) for a more complete survey paper on explicit MPC.

## Model Predictive Control Problem

Consider the following finite-time optimal control problem formulation for MPC:

$$V^*(p) = \min_z \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \quad (1a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k \quad (1b)$$

$$C_x x_k + C_u u_k \leq c \quad (1c)$$

$$k = 0, \dots, N-1$$

$$C_N x_N \leq c_N \quad (1d)$$

$$x_0 = x \quad (1e)$$

where  $N$  is the prediction horizon;  $x \in \mathbb{R}^m$  is the current state vector of the controlled system;  $u_k \in \mathbb{R}^{n_u}$  is the vector of manipulated variables at prediction time  $k$ ,  $k = 0, \dots, N-1$ ;  $z \triangleq [u'_0 \dots u'_{N-1}]' \in \mathbb{R}^n$ ,  $n \triangleq n_u N$ , is the vector of decision variables to be optimized;

$$\ell(x, u) = \frac{1}{2} x' Q x + u' R u \quad (2a)$$

$$\ell_N(x) = \frac{1}{2} x' P x \quad (2b)$$

are the stage cost and terminal cost, respectively;  $Q$ ,  $P$  are symmetric and positive semidefinite matrices; and  $R$  is a symmetric and positive definite matrix.

Let  $n_c \in \mathbb{N}$  be the number of constraints imposed at prediction time  $k = 0, \dots, N-1$ , namely,  $C_x \in \mathbb{R}^{n_c \times m}$ ,  $C_u \in \mathbb{R}^{n_c \times n_u}$ , and  $c \in \mathbb{R}^{n_c}$ , and let  $n_N$  be the number of terminal constraints, namely,  $C_N \in \mathbb{R}^{n_N \times m}$  and  $c_N \in \mathbb{R}^{n_N}$ . The total number  $q$  of linear inequality constraints imposed in the MPC problem formulation (1) is  $q = Nn_c + n_N$ .

By eliminating the states  $x_k = A^k x + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$  from problem (1), the optimal control problem (1) can be expressed as the convex quadratic program (QP):

$$V^*(x) \triangleq \min_z \frac{1}{2} z' H z + x' F' z + \frac{1}{2} x' Y x \quad (3a)$$

$$\text{s.t. } G z \leq W + S x \quad (3b)$$

where  $H = H' \in \mathbb{R}^{n \times n}$  is the Hessian matrix;  $F \in \mathbb{R}^{n \times m}$  defines the linear term of the cost function;  $Y \in \mathbb{R}^{m \times m}$  has no influence on the optimizer, as it only affects the optimal value of (3a); and the matrices  $G \in \mathbb{R}^{q \times n}$ ,  $S \in \mathbb{R}^{q \times m}$ ,  $W \in \mathbb{R}^q$  define in a compact form the constraints imposed in (1). Because of the assumptions made on the weight matrices  $Q$ ,  $R$ , and  $P$ , matrix  $H$  is positive definite, and matrix  $\begin{bmatrix} H & F' \\ F & Y \end{bmatrix}$  is positive semidefinite.

The MPC control law is defined by setting

$$u(x) = [I \ 0 \ \dots \ 0]z(x) \quad (4)$$

where  $z(x)$  is the optimizer of the QP problem (3) for the current value of  $x$  and  $I$  is the identity matrix of dimension  $n_u \times n_u$ .

## Multiparametric Solution

Rather than using a numerical QP solver online to compute the optimizer  $z(x)$  of (3) for each given current state vector  $x$ , the basic idea of explicit MPC is to pre-solve the QP off-line for the entire set of states  $x$  (or for a convex polyhedral subset  $X \subseteq \mathbb{R}^m$  of interest) to get the optimizer function  $z$ , and therefore the MPC control law  $u$ , explicitly as a function of  $x$ .

The main tool to get such an explicit solution is *multiparametric quadratic programming* (mpQP). For mpQP problems of the form (3), Bemporad et al. (2002b) proved that the optimizer function  $z^* : X_f \mapsto \mathbb{R}^n$  is piecewise affine and continuous over the set  $X_f$  of parameters  $x$  for which the problem is feasible ( $X_f$  is a polyhedral set, possibly  $X_f = X$ ) and that the value function  $V^* : X_f \mapsto \mathbb{R}$  associating with every  $x \in X_f$  the corresponding optimal value of (3) is continuous, convex, and piecewise quadratic.

An immediate corollary is that the explicit version of the MPC control law  $u$  in (4), being the first  $n_u$  components of vector  $z(x)$ , is also a continuous and piecewise-affine state-feedback

law defined over a partition of the set  $X_f$  of states into  $M$  polyhedral cells:

$$u(x) = \begin{cases} F_1x + g_1 & \text{if } H_1x \leq K_1 \\ \vdots & \vdots \\ F_Mx + g_M & \text{if } H_Mx \leq K_M \end{cases} \quad (5)$$

E

An example of such a partition is depicted in Fig. 1. The explicit representation (5) has mapped the MPC law (4) into a lookup table of linear gains, meaning that for each given  $x$ , the values computed by solving the QP (3) online and those obtained by evaluating (5) are exactly the same.

## Multiparametric QP Algorithms

A few algorithms have been proposed in the literature to solve the mpQP problem (3). All of them construct the solution by exploiting the Karush-Kuhn-Tucker (KKT) conditions for optimality:

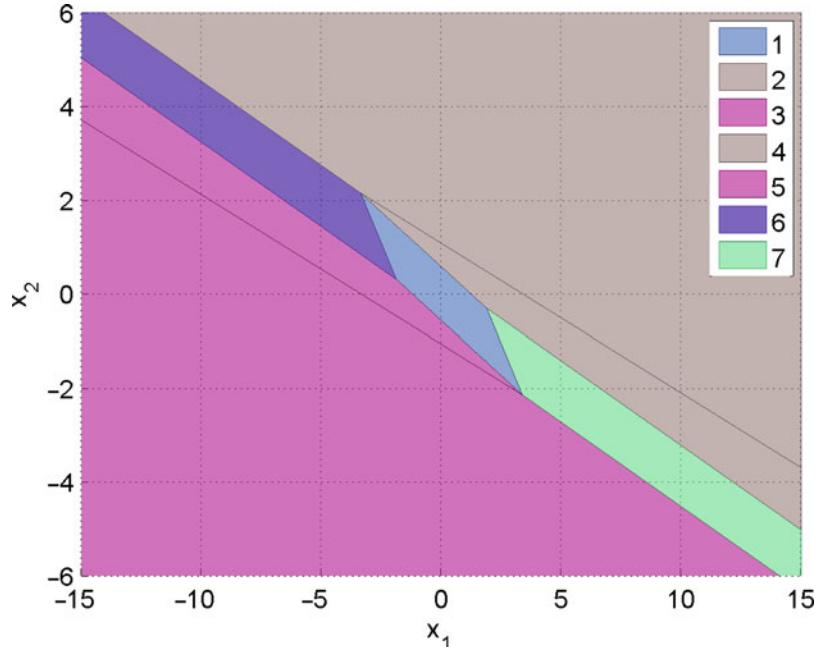
$$Hz + Fx + G'\lambda = 0 \quad (6a)$$

$$\lambda_i(G^i z - W^i - S^i x) = 0, \forall i = 1, \dots, q \quad (6b)$$

$$Gz \leq W + Sx \quad (6c)$$

$$\lambda \geq 0 \quad (6d)$$

**Explicit Model Predictive Control, Fig. 1** Explicit MPC solution for the double integrator example



where  $\lambda \in \mathbb{R}^q$  is the vector of Lagrange multipliers. For the strictly convex QP (3), conditions (6) are necessary and sufficient to characterize optimality.

An mpQP algorithm starts by fixing an arbitrary starting parameter vector  $x_0 \in \mathbb{R}^m$  (e.g., the origin  $x_0 = 0$ ), solving the QP (3) to get the optimal solution  $z(x_0)$ , and identifying the subset

$$\tilde{G}z(x) = \tilde{S}x + \tilde{W} \quad (7a)$$

of all constraints (6c) that are active at  $z(x_0)$  and the remaining inactive constraints:

$$\hat{G}z(x) \leq \hat{S}x + \hat{W} \quad (7b)$$

Correspondingly, in view of the complementarity condition (6b), the vector of Lagrange multipliers is split into two subvectors:

$$\tilde{\lambda}(x) \geq 0 \quad (8a)$$

$$\hat{\lambda}(x) = 0 \quad (8b)$$

We assume for simplicity that the rows of  $\tilde{G}$  are linearly independent. From (6a), we have the relation

$$z(x) = -H^{-1}(Fx + \tilde{G}'\tilde{\lambda}(x)) \quad (9)$$

that, when substituted into (7a), provides

$$\tilde{\lambda}(x) = -\tilde{M}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x) \quad (10)$$

where  $\tilde{M} = \tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}$  and, by substitution in (9),

$$z(x) = H^{-1}(\tilde{M}\tilde{W} + \tilde{M}(\tilde{S} + \tilde{G}H^{-1}F)x - Fx) \quad (11)$$

The solution  $z(x)$  provided by (11) is the correct one for all vectors  $x$  such that the chosen combination of active constraints remains optimal. Such all vectors  $x$  are identified by imposing constraints (7b) and (8a) on  $z(x)$  and  $\tilde{\lambda}(x)$ , respectively, that leads to constructing the polyhedral set (“critical region”):

$$CR_0 = \{x \in \mathbb{R}^n : \tilde{\lambda}(x) \geq 0, \hat{G}z(x) \leq \hat{W} + \hat{S}x\} \quad (12)$$

Different mpQP solvers were proposed to cover the rest  $X \setminus CR_0$  of the parameter set with other critical regions corresponding to new combinations of active constraints. The most efficient methods exploit the so-called “facet-to-facet” property of the multiparametric solution (Spjøtvold et al. 2006) to identify neighboring regions as in Tøndel et al. (2003a), Baotić (2002), and Bemporad (2015). Alternative methods were proposed in Jones and Morari (2006), based on looking at (6) as a multiparametric linear complementarity problem, in Patrinos and Sarimveis (2010), which provides algorithms for determining all neighboring regions even in the case the facet-to-facet property does not hold, and in Gupta et al. (2011) based on the implicit enumeration of active constraint combinations.

All methods handle the case of *degeneracy*, which may happen for some combinations of active constraints that are linearly dependent, that is, the associated matrix  $\tilde{G}$  has no full row rank (in this case,  $\tilde{\lambda}(x)$  may not be uniquely defined).

## Extensions

The explicit approach described earlier can be extended to the following MPC setting:

$$\begin{aligned} \min_{z, \epsilon} \sum_{k=0}^{N-1} & \frac{1}{2}(y_k - \mathbf{r}_k)'Q_y(y_k - \mathbf{r}_k) + \frac{1}{2}\Delta u_k' R_\Delta \Delta u_k \\ & + (u_k - \mathbf{u}_k^r)' R(u_k - \mathbf{u}_k^r)' + \rho_\epsilon \epsilon^2 \end{aligned} \quad (13a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + B_v \mathbf{v}_k \quad (13b)$$

$$y_k = Cx_k + D_u u_k + D_v \mathbf{v}_k \quad (13c)$$

$$u_k = u_{k-1} + \Delta u_k, \quad k = 0, \dots, N-1 \quad (13d)$$

$$\Delta u_k = 0, \quad k = N_u, \dots, N-1 \quad (13e)$$

$$\mathbf{u}_{\min}^k \leq u_k \leq \mathbf{u}_{\max}^k, \quad k = 0, \dots, N_u-1 \quad (13f)$$

$$\Delta \mathbf{u}_{\min}^k \leq \Delta u_k \leq \Delta \mathbf{u}_{\max}^k, \quad k = 0, \dots, N_u-1 \quad (13g)$$

$$\begin{aligned} \mathbf{y}_{\min}^k - \epsilon V_{\min} & \leq y_k \leq \mathbf{y}_{\max}^k + \epsilon V_{\max} \\ k & = 0, \dots, N_c-1 \end{aligned} \quad (13h)$$

where  $R_\Delta$  is a symmetric and positive definite matrix; matrices  $Q_y$  and  $R$  are symmetric and positive semidefinite;  $\mathbf{v}_k$  is a vector of measured disturbances;  $y_k$  is the output vector;  $\mathbf{r}_k$  its corresponding reference to be tracked;  $\Delta u_k$  is the vector of input increments;  $\mathbf{u}_k^r$  is the input reference;  $\mathbf{u}_{\min}^k$ ,  $\mathbf{u}_{\max}^k$ ,  $\Delta \mathbf{u}_{\min}^k$ ,  $\Delta \mathbf{u}_{\max}^k$ ,  $y_{\min}^k$ ,  $y_{\max}^k$  are bounds; and  $N$ ,  $N_u$ ,  $N_c$  are, respectively, the prediction, control, and constraint horizons. The extra variable  $\epsilon$  is introduced to soften output constraints, penalized by the (usually large) weight  $\rho_\epsilon$  in the cost function (13a).

Everything marked in boldface in (13), together with the command input  $u_{-1}$  applied at the previous sampling step and the current state  $x$ , can be treated as a parameter with respect to which to solve the mpQP problem and obtain the explicit form of the MPC controller. For example, for a tracking problem with no anticipative action ( $\mathbf{r}_k \equiv r_0$ ,  $\forall k = 0, \dots, N - 1$ ), no measured disturbance, and fixed upper and lower bounds, the explicit solution is a continuous piecewise-affine function of the parameter vector  $\begin{bmatrix} x \\ r_0 \\ u_{-1} \end{bmatrix}$ . Note that prediction models and/or weight matrices in (13) cannot be treated as parameters to maintain the mpQP formulation (3).

### Linear MPC Based on Convex Piecewise-Affine Costs

A similar setting can be repeated for MPC problems based on linear prediction models and convex piecewise-affine costs, such as 1- and  $\infty$ -norms. In this case, the MPC problem is mapped into a multiparametric linear programming (mpLP) problem, whose solution is again continuous and piecewise-affine with respect to the vector of parameters. For details, see Bemporad et al. (2002a).

### Robust MPC

Explicit solutions to min-max MPC problems that provide robustness with respect to additive and/or multiplicative unknown-but-bounded uncertainty were proposed in Bemporad et al. (2003), based on a combination of mpLP and dynamic programming. Again the solution is piecewise affine with respect to the state vector.

### Hybrid MPC

An MPC formulation based on convex piecewise-affine costs (like 1- or  $\infty$ -norms) and linear hybrid dynamics expressed in mixed-logical dynamical (MLD) form can be solved explicitly by treating the optimization problem associated with MPC as a multiparametric mixed integer linear programming (mpMILP) problem. The solution is still piecewise affine but may be discontinuous, due to the presence of binary variables (Bemporad et al. 2000). A better approach based on dynamic programming combined with mpLP (or mpQP) was proposed in Borrelli et al. (2005) for hybrid systems in piecewise-affine (PWA) dynamical form and linear (or quadratic) costs. For explicit hybrid MPC based on quadratic costs, the issue of possible partition overlaps is solved in Fuchs et al. (2015).

### Applicability of Explicit MPC

#### Complexity of the Solution

The complexity of the solution is given by the number  $M$  of regions that form the explicit solution (5), dictating the amount of memory to store the parametric solution ( $F_i$ ,  $G_i$ ,  $H_i$ ,  $K_i$ ,  $i = 1, \dots, M$ ) and the worst-case execution time required to compute  $F_i x + G_i$  once the problem of identifying the index  $i$  of the region  $\{x : H_i x \leq K_i\}$  containing the current state  $x$  is solved (which usually takes most of the time). The latter is called the “point location problem,” and a few methods have been proposed to solve the problem more efficiently than searching linearly through the list of regions (see, e.g., the tree-based approach of Tøndel et al. 2003b).

An upper bound to  $M$  is  $2^q$ , which is the number of all possible combinations of active constraints. In practice,  $M$  is much smaller than  $2^q$ , as most combinations are never active at optimality for any of the vectors  $x$  (e.g., lower and upper limits on an actuation signal cannot be active at the same time, unless they coincide). Moreover, regions in which the first  $n_u$  component of the multiparametric solution  $z(x)$  is the same can be joined together, provided that

their union is a convex set (an optimal merging algorithm was proposed by Geyer et al. (2008) to get a minimal number  $M$  of partitions). Nonetheless, the complexity of the explicit MPC law typically grows exponentially with the number  $q$  of constraints. The number  $m$  of parameters is less critical and mainly affects the number of elements to be stored in memory (i.e., the number of columns of matrices  $F_i$ ,  $H_i$ ). The number  $n$  of free variables also affects the number  $M$  of regions, mainly because they are usually upper and lower bounded.

### Computer-Aided Tools

The Model Predictive Control Toolbox (Bemporad et al. 2014) offers functions for designing explicit MPC controllers in MATLAB since 2014, implementing the algorithm described (Bemporad 2015). Other tools exist such as the Hybrid Toolbox (Bemporad 2003) and the Multi-Parametric Toolbox (Kvasnica et al. 2006).

### Summary and Future Directions

Explicit MPC is a powerful tool to convert an MPC design into an equivalent control law that can be implemented as a lookup table of linear gains. Whether explicit MPC is preferable to online QP depends on available CPU time, data memory, program memory, and other practical considerations. In order to answer this question, Cimini and Bemporad (2017) analyzed the behavior of a state-of-the-art dual active set method for QP in a parametric way, finding out that the number of iterations to solve the QP online is a piecewise constant function of the parameter vector. They precisely quantify how many flops the online QP solver takes in the worst case, so that the corresponding CPU time, as well as memory footprint, can be exactly compared with explicit MPC. Although suboptimal explicit MPC methods have been proposed for reducing the complexity of the control law, still the multiparametric approach remains convenient for relatively small problems (say one or two command inputs, short control and constraint horizons, up to ten states). For larger-size problems

and/or time-varying prediction models, online QP solution methods tailored to embedded MPC are usually preferable.

### Cross-References

- ▶ [Model-Predictive Control in Practice](#)
- ▶ [Nominal Model-Predictive Control](#)
- ▶ [Optimization Algorithms for Model Predictive Control](#)

### Recommended Reading

For getting started in explicit MPC, we recommend reading the paper by Bemporad et al. (2002b) and the survey paper by Alessio and Bemporad (2009). Hands-on experience using one of the MATLAB tools listed above is also useful for fully appreciating the potentials and limitations of explicit MPC. For understanding how to program a good multiparametric QP solver, the reader is recommended to take one of the approaches described in Tøndel et al. (2003a), Spjøtvold et al. (2006), Bemporad (2015), Patrinos and Sarimveis (2010), or Jones and Morari (2006).

### Bibliography

- Alessio A, Bemporad A (2009) A survey on explicit model predictive control. In: Magni L, Raimondo DM, Allgower F (eds) Nonlinear model predictive control: towards new challenging applications. Lecture notes in control and information sciences, vol 384. Springer, Berlin/Heidelberg, pp 345–369
- Banjac G, Stellato B, Moehle N, Goulart P, Bemporad A, Boyd S (2017) Embedded code generation using the OSQP solver. IEEE Conference on Decision and Control, Melbourne, pp 1906–1911
- Baočić M (2002) An efficient algorithm for multiparametric quadratic programming. Technical Report AUT02-05, Automatic Control Institute, ETH, Zurich
- Bemporad A (2003) Hybrid toolbox – user’s guide. <http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox>
- Bemporad A (2015) A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares. IEEE Trans Autom Control 60(11):2892–2903

- Bemporad A (2016) A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Trans Autom Control* 61(4):1111–1116
- Bemporad A, Borrelli F, Morari M (2000) Piecewise linear optimal controllers for hybrid systems. In: Proceedings of American Control Conference, Chicago, pp 1190–1194
- Bemporad A, Borrelli F, Morari M (2002a) Model predictive control based on linear programming – the explicit solution. *IEEE Trans Autom Control* 47(12):1974–1985
- Bemporad A, Morari M, Dua V, Pistikopoulos E (2002b) The explicit linear quadratic regulator for constrained systems. *Automatica* 38(1):3–20
- Bemporad A, Borrelli F, Morari M (2003) Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans Autom Control* 48(9):1600–1606
- Bemporad A, Morari M, Ricker N (2014) Model predictive control toolbox for MATLAB – User's guide. The Mathworks, Inc. <http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/>
- Borrelli F, Baotić M, Bemporad A, Morari M (2005) Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* 41(10):1709–1721
- Borrelli F, Bemporad A, Morari M (2017) Predictive control for linear and hybrid systems. Cambridge University Press, Cambridge/New York
- Cimini G, Bemporad A (2017) Exact complexity certification of active-set methods for quadratic programming. *IEEE Trans Autom Control* 62(12):6094–6109
- Fuchs A, Axehill D, Morari M (2015) Lifted evaluation of mp-MIQP solutions. *IEEE Trans Autom Control* 60(12):3328–3331
- Geyer T, Torrisi F, Morari M (2008) Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica* 44:1728–1740
- Gupta A, Bhartiya S, Nataraj PSV (2011) A novel approach to multiparametric quadratic programming. *Automatica* 47(9):2112–2117
- Jones C, Morari M (2006) Multiparametric linear complementarity problems. In: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, pp 5687–5692
- Kvasnica M, Grieder P, Baotić M (2006) Multi parametric toolbox (MPT). <http://control.ee.ethz.ch/~mpt/>
- Mayne D, Rawlings J (2009) Model predictive control: theory and design. Nob Hill Publishing, LCC, Madison
- Patrinos P, Bemporad A (2014) An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans Autom Control* 59(1):18–33
- Patrinos P, Sarimveis H (2010) A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings. *Automatica* 46(9):1405–1418
- Ricker N (1985) Use of quadratic programming for constrained internal model control. *Ind Eng Chem Process Des Dev* 24(4):925–936
- Spjøtvold J, Kerrigan E, Jones C, Tøndel P, Johansen TA (2006) On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica* 42(12):2209–2214
- Tøndel P, Johansen TA, Bemporad A (2003a) An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39(3):489–497
- Tøndel P, Johansen TA, Bemporad A (2003b) Evaluation of piecewise affine control via binary search tree. *Automatica* 39(5):945–950
- Wang Y, Boyd S (2010) Fast model predictive control using online optimization. *IEEE Trans Control Syst Technol* 18(2):267–278