Efficient Convex Optimization for Linear MPC

Stephen J. Wright

Abstract MPC formulations with linear dynamics and quadratic objectives can be solved efficiently by using a primal-dual interior-point framework, with complexity proportional to the length of the horizon. An alternative, which is more able to exploit the similarity of the problems that are solved at each decision point of linear MPC, is to use an active-set approach, in which the MPC problem is viewed as a convex quadratic program that is parametrized by the initial state x_0. Another alternative is to identify explicitly polyhedral regions of the space occupied by x_0 within which the set of active constraints remains constant, and to pre-calculate solution operators on each of these regions. All these approaches are discussed here.

1 Introduction

In linear model predictive control (linear MPC), the problem to be solved at each decision point has linear dynamics and a quadratic objective. This a classic problem in optimization — quadratic programming (QP) — which is convex when (as is usually true) the quadratic objective is convex. It remains a convex QP even when linear constraints on the states and controls are allowed at each stage, or when only linear functions of the state can be observed. Moreover, this quadratic program has special structure that can be exploited by algorithms that solve it, particularly interior-point algorithms.

In deployment of linear MPC, unless there is an unanticipated upset, the quadratic program to be solved differs only slightly from one decision point to the next. The question arises of whether the solution at one decision point can be used to "warm-start" the algorithm at the next decision point. Interior-point methods can make only limited use of warm-start information. Active-set methods, which treat a subset

Stephen J. Wright
Computer Sciences Department, University of Wisconsin, 1210 W. Dayton Street, Madison, WI 53706 e-mail: swright@cs.wisc.edu

of the inequality constraints (the *active set* or *working set*) as equality constraints at each iteration, are much better in this regard. Although the cost of solving the quadratic program from scratch is typically more expensive with the active-set approach than with interior-point, the cost of updating the solution at a decision point using solution information from the prior point is often minimal.

We start by outlining the most elementary control problem that arises in linear MPC — the LQR formulation — and interpret methods for solving it from both a control and optimization perspective. We then generalize this formulation to allow constraints on states and inputs, and show how interior-point methods and parametrized quadratic programming methods can be used to solve such models efficiently.

2 Formulating and Solving LQR

We consider the following discrete-time finite-horizon LQR problem:

$$\min_{x,u} \frac{1}{2} \sum_{j=0}^{N-1} (x_j^T Q x_j + u_j^T R u_j + 2 x_j^T M u_j) + \frac{1}{2} x_N^T \tilde{Q} x_N \tag{1a}$$

$$\text{subject to} \quad x_{j+1} = A x_j + B u_j, \quad j = 0,1,\ldots,N-1, \quad (x_0 \text{ given}), \tag{1b}$$

where $x_j \in \mathbb{R}^n$, $j = 0,1,\ldots,N$ and $u_j \in \mathbb{R}^m$, $j = 0,1,\ldots,N-1$; and $x = (x_1, x_2, \ldots, x_N)$ and $u = (u_0, u_1, \ldots, u_{N-1})$. This is a convex quadratic program if and only if we have

$$\begin{bmatrix} Q & M \\ M^T & R \end{bmatrix} \succeq 0, \quad \tilde{Q} \succeq 0, \tag{2}$$

where the notation $C \succeq D$ indicates that $C - D$ is positive semidefinite. When the convexity condition holds, the solution of (1) can be found by solving the following optimality conditions (also known at the Karush-Kuhn-Tucker or KKT conditions) for some vector $p = (p_0, p_1, \ldots, p_{N-1})$ with $p_j \in \mathbb{R}^n$:

$$Q x_j + M u_j + A^T p_j - p_{j-1} = 0, \quad j = 1,2,\ldots,N-1, \tag{3a}$$

$$\tilde{Q} x_N - p_{N-1} = 0, \tag{3b}$$

$$R u_j + M^T x_j + B^T p_j = 0, \quad j = 0,1,\ldots,N-1, \tag{3c}$$

$$-x_{j+1} + A x_j + B u_j = 0, \quad j = 0,1,\ldots,N-1, \tag{3d}$$

for some given value of x_0. The costates p_j, $j = 0,1,\ldots,N-1$ can be thought of as Lagrange multipliers for the state equation (1b). Since (3) is simply a system of linear equations, we can obtain the solution using standard techniques of numerical linear algebra. This can be done in a particularly efficient manner, because when the variables and equations are ordered appropriately, the coefficient matrix of this linear system is banded. We order both equations and variables in a *stagewise* manner, to express (3) as follows:

$$\begin{bmatrix} R & B^T & & & & & & & & & & & & \\ B & 0 & -I & & & & & & & & & & & \\ & -I & Q & M & A^T & & & & & & & & & \\ & & M^T & R & B^T & & & & & & & & & \\ & & A & B & 0 & -I & & & & & & & & \\ & & & & -I & Q & M & A^T & & & & & & \\ & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ & & & & & & A & B & 0 & -I & & & & \\ & & & & & & & & -I & Q & M & A^T & & \\ & & & & & & & & & M^T & R & B^T & & \\ & & & & & & & & & A & B & 0 & -I \\ & & & & & & & & & & & -I & \tilde{Q} \end{bmatrix} \begin{bmatrix} u_0 \\ p_0 \\ x_1 \\ u_1 \\ p_1 \\ \vdots \\ x_{N-1} \\ u_{N-1} \\ p_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} -M^T x_0 \\ A x_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{4}$$

This system is square, with dimension $N(2n+m)$ and bandwidth $2n+m-1$. (A matrix C is said to have bandwidth b if $C_{ij} = 0$ whenever $|i-j| > b$.)

Since factorization of a square matrix of size q with bandwidth b requires $O(qb^2)$ operations, the cost of factoring the coefficient matrix in (4) is $O(N(m+n)^3)$, which is linear in the number of stages N. A careful implementation of the banded factorization can exploit the fact that the band is "narrower" in some places than others, and thus attain further savings. If we denote the coefficient matrix in (4) by C, the vector of unknowns by z, and the right-hand side by Ex_0, where E is the matrix $\left[-M \,|\, A^T \,|\, 0 \,|\, \ldots \,|\, 0\right]^T$, an LU factorization of C with row partial pivoting has the form

$$PC = LU, \tag{5}$$

where P is a permutation matrix and L and U are lower- and upper-triangular factors whose bandwidth is a small multiple of $(m+n)$.[1] Using this factorization, we can write the system (4) as

$$PCz = LUz = PEx_0. \tag{6}$$

Thus we can obtain the solution z, for a given value of x_0, as follows:

- Calculate Ex_0;
- Apply permutations to obtain $P(Ex_0)$;
- Solve $Ly = PEx_0$ via forward-substitution to obtain y;
- Solve $Uz = y$ via back-substitution to obtain z.

Note that the LU factorization need not be recomputed for each x_0; only the four steps above need be performed. The two steps involving triangular substitution are the most computationally expensive; these require $O(N(m+n))$ operations.

The system (4) can alternatively be solved by a block-elimination technique that is equivalent to a well known concept in control: the Riccati equation. We describe

[1] The LU factorization does not exploit the fact that the coefficient matrix is symmetric. The LDL^T factorization is commonly used for such matrices, but unfortunately the permutations required in this factorization tend to destroy the band structure, so it is not appropriate here.

this approach below in the more general context of solving the banded linear system that arises at each iteration of an interior-point method.

3 Convex Quadratic Programming

Before introducing constraints into the LQR formulation (1), as happens in MPC subproblems, we introduce convex quadratic programs using general notation, and discuss their optimality conditions and the basic framework of primal-dual interior-point methods. We write the general problem as follows:

$$\min_w \tfrac{1}{2} w^T V w + c^T w \quad \text{subject to } Kw = b, \; Lw \le l, \tag{7}$$

where V is a positive semidefinite matrix. Solutions w of (7) are characterized completely by the following first-order optimality conditions (usually known as Karush-Kuhn-Tucker or KKT conditions): There are vectors λ and τ such that

$$Vw + c + K^T \lambda + L^T \tau = 0, \tag{8a}$$
$$Kw = b, \tag{8b}$$
$$0 \ge Lw - l \perp \tau \ge 0, \tag{8c}$$

where the notation $a \perp b$ means that $a^T b = 0$. Here, λ and τ are the Lagrange multipliers for the constraints $Kw = b$ and $Lw \le l$, respectively.

Primal-dual interior-point methods for (7) are often motivated as path-following methods that follow a so-called *central path* to a solution of (8). To define the central path, we first rewrite (8) equivalently by introducing slack variables s for the inequality constraints:

$$Vw + c + K^T \lambda + L^T \tau = 0, \tag{9a}$$
$$Kw = b, \tag{9b}$$
$$Lw + s = l, \tag{9c}$$
$$0 \le s \perp \tau \ge 0. \tag{9d}$$

The conditions on s and τ together imply that for each component of these vectors (s_i and τ_i) we have that both are nonnegative and at least one of the pair is zero. We can express these conditions by defining the diagonal matrices

$$S := \operatorname{diag}(s_1, s_2, \dots), \quad T := \operatorname{diag}(\tau_1, \tau_2, \dots),$$

and writing $s \ge 0$, $\tau \ge 0$, and $STe = 0$, where $e = (1, 1, \dots)^T$. (Note that STe is the vector whose components are $s_1\tau_1, s_2\tau_2, \dots$.) We can thus rewrite (9) as follows:

$$Vw + c + K^T\lambda + L^T\tau = 0, \tag{10a}$$
$$Kw = b, \tag{10b}$$
$$Lw + s = l, \tag{10c}$$
$$STe = 0, \tag{10d}$$
$$s \geq 0, \ \ \tau \geq 0. \tag{10e}$$

The central-path equations are obtained by replacing the right-hand side of (10d) by μe, for any $\mu > 0$, to obtain

$$Vw + c + K^T\lambda + L^T\tau = 0, \tag{11a}$$
$$Kw = b, \tag{11b}$$
$$Lw + s = l, \tag{11c}$$
$$STe = \mu e, \tag{11d}$$
$$s > 0, \ \ \tau > 0. \tag{11e}$$

It is known that (11) has a unique solution for each $\mu > 0$, provided that the original problem (7) has a solution.

Primal-dual interior-point methods generate iterates $(w^k, \lambda^k, \tau^k, s^k)$, $k = 0, 1, 2, \ldots$, that converge to a solution of (8). Strict positivity is maintained for all components of s^k and τ^k, for all k; that is, $s^k > 0$ and $\tau^k > 0$. At step k, a search direction is generated as a Newton-like step for the square nonlinear system of equations formed by the four equality conditions in (11), for some value of μ that is chosen by a (somewhat elaborate) adaptive scheme. These Newton equations are as follows:

$$\begin{bmatrix} V & K^T & L^T & 0 \\ K & 0 & 0 & 0 \\ L & 0 & 0 & I \\ 0 & 0 & S^k & T^k \end{bmatrix} \begin{bmatrix} \Delta w^k \\ \Delta \lambda^k \\ \Delta \tau^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} r_w^k \\ r_\lambda^k \\ r_\tau^k \\ r_{st}^k \end{bmatrix}, \tag{12a}$$

$$\text{where} \quad \begin{bmatrix} r_w^k \\ r_\lambda^k \\ r_\tau^k \\ r_{st}^k \end{bmatrix} = - \begin{bmatrix} Vw^k + c + K^T\lambda^l + L^T\tau^k \\ Kw^k - b \\ Lw^k + s^k - l \\ S^kT^ke - \mu_k e \end{bmatrix}, \tag{12b}$$

where $S^k = \operatorname{diag}(s_1^k, s_2^k, \ldots)$ and $T^k = \operatorname{diag}(t_1^k, t_2^k, \ldots)$. The step along this direction has the form

$$(w^{k+1}, \lambda^{k+1}, \tau^{k+1}, s^{k+1}) := (w^k, \lambda^k, \tau^k, s^k) + \alpha_k(\Delta w^k, \Delta\lambda^k, \Delta\tau^k, \Delta s^k),$$

where $\alpha_k > 0$ is chosen to maintain strict positivity on the s and τ vectors, that is, $\tau^{k+1} > 0$ and $s^{k+1} > 0$.

Some block elimination is usually applied to the system (12a), rather than factoring the matrix directly. By substituting out for Δs^k, we obtain

$$\begin{bmatrix} V & K^T & L^T \\ K & 0 & 0 \\ L & 0 & -(T^k)^{-1}S^k \end{bmatrix} \begin{bmatrix} \Delta w^k \\ \Delta \lambda^k \\ \Delta \tau^k \end{bmatrix} = \begin{bmatrix} r_w^k \\ r_\lambda^k \\ r_\tau^k - (T^k)^{-1} r_{st}^k \end{bmatrix}. \tag{13}$$

(Note that the matrix $(T^k)^{-1}S^k$ is positive diagonal.) We can further use the third row in (13) to eliminate $\Delta\tau^k$, to obtain the following block 2×2 system

$$\begin{bmatrix} (V + L^T(S^k)^{-1}T^kL) & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} \Delta w^k \\ \Delta \lambda^k \end{bmatrix} = \begin{bmatrix} r_w^k - L^T(S^k)^{-1}T^k(r_\tau^k - (T^k)^{-1}r_{st}^k) \\ r_\lambda^k \end{bmatrix}. \tag{14}$$

In practice, the forms (13) and (14) are most commonly used to obtain the search directions. We see below that the form (14) applied to a constrained version of (1) leads (with appropriate ordering of the variables) to a linear system with the same structure as (4).

4 Linear MPC Formulations and Interior-Point Implementation

We now describe an extension of (1), more common in applications of linear MPC, in which the inputs u_k and states x_k are subject to additional constraints at each stage $k = 1,2,\ldots,N-1$. The final state x_N is often also constrained, with the goal of steering the state into a certain polyhedral set at the end of the time horizon, from which an unconstrained LQR strategy can be pursued from that point forward without fear of violating the stagewise constraints.

We start with the formulation of linear MPC, in its most natural form (involving both states and inputs at each time point) and in a condensed form in which all states beyond the initial state x_0 are eliminated from the problem. We then derive KKT conditions for the original formulation and show how the structure that is present in linear MPC allows primal-dual interior-point methods to be implemented efficiently.

4.1 Linear MPC Formulations

We define the linear MPC problem as follows:

$$\min_{x_1,\ldots,x_N,u_0,\ldots,u_{N-1}} \; \frac{1}{2}\sum_{j=0}^{N-1} (x_j^T Q x_j + u_j^T R u_j + 2x_j^T M u_j) + \frac{1}{2} x_N^T \tilde{Q} x_N \tag{15a}$$

$$\text{subject to} \quad x_{j+1} = Ax_j + Bu_j, \quad j = 0,1,\ldots,N-1, \quad (x_0 \text{ given}); \tag{15b}$$

$$Gu_j + Hx_j \le h, \quad j = 0,1,\ldots,N-1; \tag{15c}$$

$$Fx_N \le f. \tag{15d}$$

The linear constraints (15c),(15d) define polyhedral regions; these could be bounds (upper and/or lower) on individual components of x_j and u_j, more complicated linear constraints that are separable in x_j and u_j, or mixed constraints that involve states and inputs together.

We obtain a condensed form of (15) by using the state constraints (15b) to eliminate $x_1, x_2, \ldots, x_N$. We start by aggregating the variables and constraints in (15) into a representation that disguises the stagewise structure. We define

$$\bar{x} := \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad \bar{u} := \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad \bar{h} := \begin{bmatrix} h \\ h \\ \vdots \\ h \\ f \end{bmatrix},$$

$$\bar{Q} := \begin{bmatrix} Q & & & & \\ & Q & & & \\ & & \ddots & & \\ & & & Q & \\ & & & & \tilde{Q} \end{bmatrix}, \quad \bar{R} := \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix}, \quad \bar{M} := \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{H} := \begin{bmatrix} H & & & & \\ & H & & & \\ & & \ddots & & \\ & & & H & \\ & & & & F \end{bmatrix}, \quad \bar{G} := \begin{bmatrix} G & & & \\ & G & & \\ & & \ddots & \\ & & & G \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and note that the objective (15a) can be written as

$$\frac{1}{2}\left[\bar{x}^T \bar{Q} \bar{x} + \bar{u}^T \bar{R} \bar{u} + 2\bar{x}^T \bar{M} \bar{u}\right], \tag{16}$$

while the constraints (15c) and (15d) can be written

$$\bar{G}\bar{u} + \bar{H}\bar{x} \le \bar{h}. \tag{17}$$

From the state equation (15b), we have

$$\bar{x} = \bar{A}x_0 + \bar{B}\bar{u}, \tag{18}$$

where

$$\bar{B} := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ B & & & & \\ AB & B & & & \\ A^2B & AB & B & & \\ \vdots & \vdots & & \ddots & \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \ldots & B \end{bmatrix}, \quad \bar{A} := \begin{bmatrix} I \\ A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix}. \tag{19}$$

By substituting (18) into (16) and (17), we obtain the following condensed form of (15):

$$\min_{\bar{u}} \tfrac{1}{2}\bar{u}^T(\bar{R}+\bar{B}^T\bar{Q}\bar{B}+\bar{B}^T\bar{M}+\bar{M}^T\bar{B})\bar{u}+x_0^T\bar{A}^T(\bar{Q}\bar{B}+\bar{M})\bar{u} \tag{20a}$$

$$\text{subject to } \left[\bar{G}+\bar{H}\bar{B}\right]\bar{u} \le h-\bar{H}\bar{A}x_0. \tag{20b}$$

We have omitted a quadratic term in x_0 from the objective in (20a), because it is independent of the variable $\bar{u}$ and thus does not affect the solution. That is, the problem defined by (16), (17), (18) is equivalent to the problem defined by (20) in that the solution of (20) is identical to the $\bar{u}$ component of the solution of (16), (17), (18).

4.2 KKT Conditions and Efficient Interior-Point Implementation

We can follow the methodology of Section 3 to write down the KKT optimality conditions. As in Section 2, we use p_j to denote the costates (or Lagrange multipliers for the state equation (15b)). We introduce λ_j, $j=0,1,\dots,N-1$ as Lagrange multipliers for $Gu_j \ge g$, ζ_j, $j=0,1,\dots,N-1$ as multipliers for the constraints $Hx_j \le h$, and β as the vector of Lagrange multipliers for the constraint $Fx_N \le f$. The KKT conditions, following the template (8), as as follows:

$$Qx_j+Mu_j+A^Tp_j-p_{j-1}+H^T\zeta_j=0, \qquad j=1,2,\dots,N-1, \tag{21a}$$

$$\tilde{Q}x_N+F^T\beta-p_{N-1}=0, \tag{21b}$$

$$Ru_j+M^Tx_j+B^Tp_j+G^T\lambda_j=0, \qquad j=0,1,\dots,N-1, \tag{21c}$$

$$-x_{j+1}+Ax_j+Bu_j=0, \qquad j=0,1,\dots,N-1, \tag{21d}$$

$$0 \ge Gu_j+Hx_j-h \perp \lambda_j \ge 0, \quad j=0,1,\dots,N-1, \tag{21e}$$

$$0 \ge Fx_N-f \perp \beta \ge 0. \tag{21f}$$

By introducing slack variables s_j^λ for the constraints $Gu_j+Hx_j \le h$ and s^β for the constraint $Fx_N \le f$, we obtain the following formula (cf. (9)):

$$
\begin{aligned}
Qx_j + Mu_j + A^T p_j - p_{j-1} + H^T \zeta_j &= 0, \qquad && j = 1,2,\ldots,N-1, && \text{(22a)}\\
\tilde{Q} x_N + F^T \beta - p_{N-1} &= 0, && && \text{(22b)}\\
Ru_j + M^T x_j + B^T p_j + G^T \lambda_j &= 0, && j = 0,1,\ldots,N-1, && \text{(22c)}\\
-x_{j+1} + Ax_j + Bu_j &= 0, && j = 0,1,\ldots,N-1, && \text{(22d)}\\
Gu_j + Hx_j + s_j^\lambda &= h, && j = 0,1,\ldots,N-1, && \text{(22e)}\\
Fx_N + s^\beta &= f, && && \text{(22f)}\\
0 \le s_j^\lambda \perp \lambda_j &\ge 0, && j = 0,1,\ldots,N-1, && \text{(22g)}\\
0 \le s^\beta \perp \beta &\ge 0. && && \text{(22h)}
\end{aligned}
$$

By proceeding with the primal-dual interior-point approach, as described in Section 3, we solve the following system of equations to be solved at iteration k, obtained by specializing the form (14) to the structure of the MPC problem:

$$
\begin{bmatrix}
R_0 & B^T \\
B & 0 & -I \\
& -I & Q_1 & M_1 & A^T \\
& & M_1^T & R_1 & B^T \\
& & A & B & 0 & -I \\
& & & & -I & Q_2 & M_2 & A^T \\
& & & & & \ddots & \ddots & \ddots & \ddots & \ddots \\
& & & & & & A & B & 0 & -I \\
& & & & & & & & -I & Q_{N-1} & M_{N-1} & A^T \\
& & & & & & & & & M_{N-1}^T & R_{N-1} & B^T \\
& & & & & & & & & A & B & 0 & -I \\
& & & & & & & & & & & -I & \tilde{Q}_N
\end{bmatrix}
\begin{bmatrix}
\Delta u_0 \\ \Delta p_0 \\ \Delta x_1 \\ \Delta u_1 \\ \Delta p_1 \\ \vdots \\ \Delta x_{N-1} \\ \Delta u_{N-1} \\ \Delta p_{N-1} \\ \Delta x_N
\end{bmatrix}
=
\begin{bmatrix}
\tilde{r}_0^u \\ \tilde{r}_0^p \\ \tilde{r}_1^x \\ \tilde{r}_1^u \\ \tilde{r}_1^p \\ \vdots \\ \tilde{r}_{N-1}^x \\ \tilde{r}_{N-1}^u \\ \tilde{r}_{N-1}^p \\ \tilde{r}_N^x
\end{bmatrix},
\tag{23}
$$

where

$$
\begin{aligned}
\begin{bmatrix} Q_j & M_j \\ M_j^T & R_j \end{bmatrix} &= \begin{bmatrix} Q & M \\ M^T & R \end{bmatrix} + \begin{bmatrix} G^T \\ H^T \end{bmatrix} (S_j^\lambda)^{-1} D_j^\lambda \begin{bmatrix} G \; H \end{bmatrix}, \quad j = 1,2,\ldots,N-1,\\
R_0 &= R + G^T (S_0^\lambda)^{-1} D_0^\lambda G,\\
\tilde{Q}_N &= \tilde{Q} + F^T (S^\beta)^{-1} D^\beta F,
\end{aligned}
$$

and

$$
\begin{aligned}
S_j^\lambda &= \operatorname{diag}((s_j^\lambda)_1, (s_j^\lambda)_2, \ldots), \quad j = 0,1,\ldots,N-1,\\
S^\beta &= \operatorname{diag}((s^\beta)_1, (s^\beta)_2, \ldots),\\
D_j^\lambda &= \operatorname{diag}((\lambda_j)_1, (\lambda_j)_2, \ldots), \quad j = 0,1,\ldots,N-1,\\
D^\beta &= \operatorname{diag}(\beta_1, \beta_2, \ldots).
\end{aligned}
$$

We omit definitions of the terms in the right-hand side of (23); we refer to the general form of Section 3 for information on how to construct this vector. We note that the intial state x_0 appears linearly in $\tilde{r}_0^u$ and $\tilde{r}_0^p$.

This system can be solved using direct LU factorization of the coefficient matrix, as described in Section 2. But we describe here an alternative approach based on block-factorization of the matrix, which is essentially identical to solving a discrete-time, time-varying Riccati equation. We follow the derivation of [11, Section 3.3] to describe this technique. The key step is to use (23) to find matrices $\Pi_N, \Pi_{N-1}, \dots, \Pi_1$ of size $n \times n$ and vectors $\pi_N, \pi_{N-1}, \dots, \pi_1$ such that

$$-p_{k-1} + \Pi_k \Delta x_k = \pi_k, \quad k = N, N-1, \dots, 1. \tag{24}$$

We find a recursive formula, working backwards from N. We see immediately from (4) that (24) is satisfied for $k = N$ by setting

$$\Pi_N = \tilde{Q}_N, \quad \pi_N = \tilde{r}_N^x. \tag{25}$$

Now supposing that the relationship (24) holds for some k, with known values of Π_k and π_k, we obtain formulas for Π_{k-1} and π_{k-1}. By combining (24) with three successive block rows from the system (23), we obtain the system

$$\begin{bmatrix} -I & Q_{k-1} & M_{k-1} & A^T & \\ & M_{k-1}^T & R_{k-1} & B^T & \\ & A & B & 0 & -I \\ & & & -I & \Pi_k \end{bmatrix} \begin{bmatrix} \Delta p_{k-2} \\ \Delta x_{k-1} \\ \Delta u_{k-1} \\ \Delta p_{k-1} \\ \Delta x_k \end{bmatrix} = \begin{bmatrix} \tilde{r}_{k-1}^x \\ \tilde{r}_{k-1}^u \\ \tilde{r}_{k-1}^p \\ \pi_k \end{bmatrix}. \tag{26}$$

By eliminating Δp_{k-1} and Δx_k, we obtain the reduced system

$$\begin{bmatrix} -I & Q_{k-1} + A^T \Pi_k A & A^T \Pi_k B + M_{k-1} \\ 0 & B^T \Pi_k A + M_{k-1}^T & R_{k-1} + B^T \Pi_k B \end{bmatrix} \begin{bmatrix} \Delta p_{k-2} \\ \Delta x_{k-1} \\ \Delta u_{k-1} \end{bmatrix} = \begin{bmatrix} \tilde{r}_{k-1}^x + A^T \Pi_k \tilde{r}_{k-1}^p + A^T \pi_k \\ \tilde{r}_{k-1}^u + B^T \Pi_k \tilde{r}_{k-1}^p + B^T \pi_k \end{bmatrix}. \tag{27}$$

Finally, by eliminating Δu_{k-1}, we obtain

$$-\Delta p_{k-2} + \Pi_{k-1} \Delta x_{k-1} = \pi_{k-1}, \tag{28}$$

where

$$\Pi_{k-1} = Q + A^T \Pi_k A - (A^T \Pi_k B + M)(R + B^T \Pi_k B)^{-1} (B^T \Pi_k A + M^T), \tag{29a}$$

$$\begin{aligned} \pi_{k-1} = {} & \tilde{r}_{k-1}^x + A^T \Pi_k \tilde{r}_{k-1}^p + A^T \pi_k \\ & - (A^T \Pi_k B + M)(R + B^T \Pi_k B)^{-1} (\tilde{r}_{k-1}^u + B^T \Pi_k \tilde{r}_{k-1}^p + B^T \pi_k). \end{aligned} \tag{29b}$$

A recursive argument based on symmetry of all Q_k and R_k reveals that all Π_k, $k = N, N-1, \dots, 1$ are symmetric, and the matrix inversions in (29) can be performed whenever R is positive definite, a sufficient condition that is usually assumed in practice. The formula (29a) along with the initialization (25) is the *discrete-time, time-varying* Riccati equation. The more familiar algebraic Riccati equation is the limit of (29a) obtained by setting $\Pi_k = \Pi_{k-1} = \Pi$, and assuming that all Q_k, R_k and M_k are identically equal to Q, R, and M, respectively. We obtain

$$\Pi = Q + A^T \Pi A - (A^T \Pi B + M)(R + B^T \Pi B)^{-1}(B^T \Pi A + M^T). \tag{30}$$

Having computed Π_k and π_k for $k = N, N-1, \ldots, 1$, we proceed to solve (23) as follows. By combining the first two rows of (23) with the formula (24) for $k = 1$, we obtain

$$\begin{bmatrix} R & B^T & \\ B & 0 & -I \\ & -I & \Pi_1 \end{bmatrix} \begin{bmatrix} \Delta u_0 \\ \Delta p_0 \\ \Delta x_1 \end{bmatrix} = \begin{bmatrix} \tilde{r}_0^u \\ \tilde{r}_0^p \\ \pi_1 \end{bmatrix}. \tag{31}$$

This square system (with $2n+m$ rows and columns) can be solved to find Δu_0, Δp_0, and Δx_1. We now use the second row of (27) along with the equations involving both Δx_k and Δx_{k-1} to obtain the following formulas:

$$\begin{aligned} \Delta u_{k-1} &= (R + B^T \Pi_k B)^{-1} \\ &\qquad \left[\tilde{r}_{k-1}^u + B^T \Pi_k \tilde{r}_{k-1}^p + B^T \pi_k - (B^T \Pi_k A + M_{k-1}^T)\Delta x_{k-1}\right], && \text{(32a)} \\ \Delta x_k &= A \Delta x_{k-1} + B \Delta u_{k-1} - \tilde{r}_{k-1}^p, && \text{(32b)} \end{aligned}$$

which we iterate forward for $k = 2, 3, \ldots, N$ to obtain all inputs $\Delta u_0, \Delta u_1, \ldots, \Delta u_{N-1}$ and all states $\Delta x_1, \Delta x_2, \ldots, \Delta x_N$. The costates $\Delta p_1, \Delta p_2, \ldots, \Delta p_{N-1}$ can be recovered directly by substituting $\Delta x_2, \Delta x_3, \ldots, \Delta x_N$ into (28).

5 Parametrized Convex Quadratic Programming

The linear MPC is actually a convex QP that is parametrized by the current (initial) state x_0. In this section we formulate parametrized QP in general terms, write down optimality conditions, and describe a primal-dual active-set approach for finding its solution for some value of x_0, when the solution is already known for a (usually nearby) value of x_0. This approach leverages “warm-start” information from the current solution, and is often able to find the new solution quickly when the change to x_0 is small, as is often the case in the MPC context, unless an unmodeled upset occurs.

Omitting the equality constraints in (7) (which are not required for our application to linear MPC), we write the parametrized form as follows:

$$\min_w \tfrac{1}{2} w^T V w + c^T w - (J x_0)^T w \quad \text{subject to } \; Lw \le l + E x_0. \tag{33}$$

Note that this formulation corresponds exactly to the condensed form (20) of the linear MPC problem. We assume throughout this section that V is a *positive definite* matrix (so that this QP is strongly convex), L is a matrix of size $m_I \times p$, J and E are matrices, and x_0 is the parameter vector. Following (8), we can write the optimality conditions as follows:

$$Vw + c + L^T \tau = Jx_0, \tag{34a}$$
$$0 \geq Lw - l - Ex_0 \perp \tau \geq 0. \tag{34b}$$

Because of the complementarity conditions (34b), we can identify an *active set* $\mathscr{A} \subset \{1, 2, \ldots, m_I\}$ that indicates which of the m_I inequality constraints are satisfied *at equality*. That is, $i \in \mathscr{A}$ only if $L_{i\cdot} w = (l + Ex_0)_i$, where $L_{i\cdot}$ denotes the ith row of L. It follows from this definition and (34b) that

$$L_{i\cdot} w < (l + Ex_0)_i \text{ for all } i \notin \mathscr{A}; \quad \tau_{\mathscr{A}} \geq 0; \quad \tau_{\mathscr{A}^c} = 0; \tag{35}$$

where $\tau_{\mathscr{A}} = [\tau_i]_{i \in \mathscr{A}}$ and $\tau_{\mathscr{A}^c} = [\tau_i]_{i \notin \mathscr{A}}$. (Note that $\mathscr{A}^c$ denotes the complement of $\mathscr{A}$ in $\{1, 2, \ldots, m_I\}$.) We can substitute these definitions into (34) to obtain

$$Vw + c + L_{\mathscr{A}}^T \tau_{\mathscr{A}} = Jx_0, \tag{36a}$$
$$L_{\mathscr{A}} w = (l + Ex_0)_{\mathscr{A}}, \tag{36b}$$
$$L_{\mathscr{A}^c} w \leq (l + Ex_0)_{\mathscr{A}^c}, \tag{36c}$$
$$\tau_{\mathscr{A}} \geq 0, \tag{36d}$$
$$\tau_{\mathscr{A}^c} = 0. \tag{36e}$$

At this point, we can make several interesting observations. Let us define the set $\mathscr{P}$ of points x_0 such that the feasible region for (33) is nonempty, that is,

$$\mathscr{P} := \{x_0 \,|\, Lw \leq l + Ex_0 \text{ for some } w\}.$$

First, because of the strong convexity of the quadratic objective, a solution of (36) is guaranteed to exist for all $x_0 \in \mathscr{P}$. Second, the set $\mathscr{P}$ is a polyhedron. This follows from the fact that $\mathscr{P}$ is the projection onto x_0 space of the polyhedral set

$$\{(w, x_0) \,|\, Lw \leq l + Ex_0\},$$

and the projection of a polyhedron onto a plane is itself polyhedral. Similar logic indicates that the subset of $\mathscr{P}$ that corresponds to a given active set $\mathscr{A}$ is also polyhedral. By fixing $\mathscr{A}$ in (36), we can note that the set

$$\mathscr{P}_{\mathscr{A}} := \{x_0 \,|\, (w, \tau, x_0) \text{ satisfies (36) for some } w, \tau\} \tag{37}$$

is the projection of the polyhedron defined by (36) onto x_0-space, so is itself polyhedral.

5.1 Enumeration

The last observation above suggests an enumeration approach, first proposed in [1, 2] and developed further by [12, 9] and others. We describe this approach for the case in which the row submatrices $L_{\mathscr{A}}$ of L have full row rank, for all possible active sets $\mathscr{A}$ of interest. In this case, because V is positive definite, the vectors w and $\tau_{\mathscr{A}}$ are uniquely determined by the conditions (36a), (36b), that is,

$$\begin{bmatrix} w \\ \tau_{\mathscr{A}} \end{bmatrix} = \begin{bmatrix} V & L_{\mathscr{A}}^T \\ L_{\mathscr{A}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -c + Jx_0 \\ (l + Ex_0)_{\mathscr{A}} \end{bmatrix} = \begin{bmatrix} z_{w,\mathscr{A}} + Z_{w,\mathscr{A}} x_0 \\ z_{\tau,\mathscr{A}} + Z_{\tau,\mathscr{A}} x_0 \end{bmatrix}, \tag{38}$$

where

$$\begin{bmatrix} z_{w,\mathscr{A}} \\ z_{\tau,\mathscr{A}} \end{bmatrix} := \begin{bmatrix} V & L_{\mathscr{A}}^T \\ L_{\mathscr{A}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -c \\ l_{\mathscr{A}} \end{bmatrix}, \quad \begin{bmatrix} Z_{w,\mathscr{A}} \\ Z_{\tau,\mathscr{A}} \end{bmatrix} := \begin{bmatrix} V & L_{\mathscr{A}}^T \\ L_{\mathscr{A}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} J \\ E_{\mathscr{A}} \end{bmatrix}, \tag{39}$$

where $E_{\mathscr{A}}$ is the row submatrix of E corresponding to $\mathscr{A}$. (Nonsingularity of the matrix that is inverted in (38) follows from positive definiteness of V and full row rank of $L_{\mathscr{A}}$.) We can substitute into (36c), (36d) to check the validity of this solution, that is,

$$L_{\mathscr{A}^c}(z_{w,\mathscr{A}} + Z_{w,\mathscr{A}} x_0) \leq l_{\mathscr{A}^c} + E_{\mathscr{A}^c} x_0, \tag{40a}$$

$$z_{\tau,\mathscr{A}} + Z_{\tau,\mathscr{A}} x_0 \geq 0. \tag{40b}$$

In fact, these inequalities along with the definitions (39) provide another characterization of the set $\mathscr{P}_{\mathscr{A}}$ defined in (37): $\mathscr{P}_{\mathscr{A}}$ is exactly the set of vectors x_0 that satisfies the linear inequalities (40), which we can express as $Y_{\mathscr{A}} x_0 \leq y_{\mathscr{A}}$, where

$$Y_{\mathscr{A}} := \begin{bmatrix} L_{\mathscr{A}^c} Z_{c,\mathscr{A}} - E_{\mathscr{A}^c} \\ -Z_{\tau,\mathscr{A}} \end{bmatrix}, \quad y_{\mathscr{A}} := \begin{bmatrix} l_{\mathscr{A}^c} - L_{\mathscr{A}^c} z_{w,\mathscr{A}} \\ z_{\tau,\mathscr{A}} \end{bmatrix}. \tag{41}$$

An enumeration approach stores the pairs $(Y_{\mathscr{A}}, y_{\mathscr{A}})$ for some or all of the $\mathscr{A}$ for which $\mathscr{P}_{\mathscr{A}}$ is nonempty. Then, when presented with a particular value of the parameter x_0, it identifies the set $\mathscr{A}$ for which $Y_{\mathscr{A}} x_0 \leq y_{\mathscr{A}}$. The solution $(w, \tau_{\mathscr{A}})$ can then be recovered from (38), and we set $\tau_{\mathscr{A}^c} = 0$ to fill the remaining components of the Lagrange multiplier vector.

Enumeration approaches shift much of the work in calculating solutions of (33) offline. The pairs $(Y_{\mathscr{A}}, y_{\mathscr{A}})$ can be pre-computed for all $\mathscr{A}$ for which $\mathscr{P}_{\mathscr{A}}$ is nonempty. The online computation consists of testing the conditions $Y_{\mathscr{A}} x_0 \leq y_{\mathscr{A}}$ for the given x_0. The order of testing can be crucial, as we want to identify the correct $\mathscr{A}$ for this value of x_0 as quickly as possible. (The approach in [9] maintains a table of the most frequently occurring instances of $\mathscr{A}$.) Full enumeration approaches become impractical quickly as the dimensions of the problem (and particularly the number of constraints m_I) increase. Partial enumeration stores only the pairs $(Y_{\mathscr{A}}, y_{\mathscr{A}})$ that have occurred most frequently and / or most recently during plant operation; when an x_0 is encountered that does not fall into any of the polyhedra currently stored, the solution can be computed from scratch, or some suboptimal backup strategy can be deployed. For plants of sufficiently high dimension, this approach too becomes impractical, but these enumeration approaches can be an appealing and practical way to implement linear MPC on small systems.

5.2 Active-Set Strategy

For problems that are too large for complete or partial enumeration to be practical, the conditions (36) can be used as the basis of an active-set strategy for solving (33). Active-set strategies make a series of estimates of the correct active set for (36), changing this estimate in a systematic way by a single index $i \in \{1,2,\dots,m_I\}$ (added or removed) at each iteration. This approach can be highly efficient in the context of linear MPC, when the parameter x_0 does not change greatly from one decision point to the next. If a solution of (33) is known for value of x_0 and we need to know a new solution for a nearby value, say x_0^{new}, it can often be found with just a few changes to the active set, which requires just a few steps of the algorithm. We give just an outline of the approach here; further details can be found in [4] and [5].

Before proceeding, we pay a little attention to the issue of *degeneracy*, which complicates significantly the implementation of active-set approaches. Degeneracy is present when there is ambiguity in the definition of the active set $\mathscr{A}$ for which (36) holds at a certain parameter x_0, or when the active constraint matrix $L_{\mathscr{A}}$ fails to have full row rank. Degeneracy of the former type occurs when there is some index $i \in \{1,2,\dots,m_I\}$ such that both $(Lw-l-Ex_0)_i = 0$ and $\tau_i = 0$ for (w,τ) satisfying (34). Thus, for $\mathscr{A}$ satisfying (34), we may have either $i \in \mathscr{A}$ or $i \notin \mathscr{A}$; there is ambiguity about whether the constraint i is really "active." (Constraints with this property are sometimes called "weakly active.") Degeneracy of the latter type — rank-deficiency of $L_{\mathscr{A}}$ — leads to possible ambiguity in the definition of $\tau_{\mathscr{A}}$. Specifically, there may be multiple vectors $\tau_{\mathscr{A}}$ that satisfy conditions (36a) and (36d), that is,

$$L_{\mathscr{A}}^T \tau_{\mathscr{A}} = -Vw - c + Jx_0, \quad \tau_{\mathscr{A}} \geq 0. \tag{42}$$

For a particular $\mathscr{A}$ and a particular choice of (w,τ) satisfying (36), *both* types of degeneracy may be present. These degeneracies can be resolved by choosing $\tau_{\mathscr{A}}$ to be an extreme point of the polyhedron represented by (42), and removing from $\mathscr{A}$ those elements i for which $\tau_i = 0$.

We note that there is no ambiguity in the value of w for a given parameter x_0; the positive definiteness assumption on V precludes this possibility.

To account for the possibility of degeneracy, active-set algorithms introduce the concept of a *working set*. This is a subset $\mathscr{W}$ of $\{1,2,\dots,m_I\}$ that is an estimate of the (possibly ambiguous) optimal active set $\mathscr{A}$ from (36), but with the additional property that the row constraint submatrix $L_{\mathscr{W}}$ has full rank. Small changes are made to the working set $\mathscr{W}$ at each step of the active-set method, to maintain the full-rank property but ultimately to converge to an optimal active set $\mathscr{A}$ for (36).

Let x_0 be the value of the parameter for which a primal-dual solution of (36) is known, for a certain active set $\mathscr{A}$. By doing some processing of (36), as discussed in the previous paragraph, we can identify a working set $\mathscr{W} \subset \mathscr{A}$ such that (36) holds when we replace $\mathscr{A}$ by $\mathscr{W}$ and, in addition, $L_{\mathscr{W}}$ has full row rank.

Suppose we wish to calculate the solution of (33) for a new value x_0^{new}, and define $\Delta x_0 := x_0^{\text{new}} - x_0$. We can determine the effect of replacing x_0 by x_0^{new} on the primal-dual solution components $(x, \tau_{\mathscr{W}})$ by applying (36a) and (36b) to account for the change in x_0 (similar to what we did in (38)):

$$\begin{bmatrix} \Delta w \\ \Delta \tau_{\mathscr{W}} \end{bmatrix} = \begin{bmatrix} V & L_{\mathscr{W}}^T \\ L_{\mathscr{W}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} J\Delta x_0 \\ (E\Delta x_0)_{\mathscr{W}} \end{bmatrix}, \tag{43}$$

If we can take a full step along this perturbation vector without violating the other conditions (36c) and (36d), we are done! To check these conditions, we need to verify that

$$L_{\mathscr{W}^c}(w + \Delta w) \le (l + E x_0^{\text{new}})_{\mathscr{W}^c}, \quad \tau_{\mathscr{W}} + \Delta \tau_{\mathscr{W}} \ge 0. \tag{44}$$

If these conditions do not hold, it is necessary to make systematic changes to the active set $\mathscr{W}$. To start this process, we find the longest steplength that can be taken along $(\Delta w, \Delta \tau_{\mathscr{W}})$ while maintaining (36c) and (36d). That is, we seek the largest value of α in the range $(0,1]$ such that

$$L_{\mathscr{W}^c}(w + \alpha \Delta w) \le (l + E x_0 + \alpha E \Delta x_0)_{\mathscr{W}^c}, \quad \tau_{\mathscr{W}} + \alpha \Delta \tau_{\mathscr{W}} \ge 0. \tag{45}$$

We call this value $\alpha_{\max}$; it can be calculated explicitly from the following formulas:

$$i_P := \arg\min_{i \in \mathscr{W}^c} \frac{(l + E x_0)_i - L_{i\cdot} w}{L_{i\cdot} \Delta w - (E \Delta x_0)_i}, \quad \alpha_{\max,P} := \frac{(l + E x_0)_{i_P} - L_{i_P\cdot} w}{L_{i_P\cdot} \Delta w - (E \Delta x_0)_{i_P}}, \tag{46a}$$

$$i_D := \arg\min_{i \in \mathscr{W}} -\frac{\tau_i}{\Delta \tau_i}, \quad \alpha_{\max,D} := -\frac{\tau_{i_D}}{\Delta \tau_{i_D}}, \tag{46b}$$

$$\alpha_{\max} := \min(1, \alpha_{\max,P}, \alpha_{\max,D}). \tag{46c}$$

The constraint that "blocks" α at a value less than 1 — either i_P or i_D from (46) — motivates a change to the working set $\mathscr{W}$. If one of the Lagrange multipliers τ_i for $i \in \mathscr{W}$ goes to zero first, we remove this index from $\mathscr{W}$, allowing the corresponding constraint to move away from its constraint boundary at the next iteration. Alternatively, if one of the constraints i becomes active, we add this index to the working set $\mathscr{W}$ for the next iteration. We may need to do some postprocessing of the working set in the latter case, possibly removing some other element of the working set to maintain full rank of $L_{\mathscr{W}}$. In both of these cases, we update the values of x_0, w, and τ to reflect the step just taken, recalibrate the parameter perturbation vector Δx_0, and repeat the process. If there are no blocking constraints, and a full step can be taken, then we have recovered the solution and the active-set algorithm declares success and stops.

The active-set procedure is summarized as Algorithm 1. An example of the polyhedral decomposition of parameter space is shown in Figure 1. In this example, four steps of the active-set method (and three changes to the working set) are required to move from x_0 to the new parameter x_0^{new}.

Algorithm 1 Online Active Set Approach

Given current parameter x_0, new parameter x_0^{new}, current primal-dual solution (w,τ) and working set $\mathscr{W}$ obtained from the active set $\mathscr{A}$ satisfying (36) as described in the text;
Set $\alpha_{\max} = 0$;
while $\alpha_{\max} < 1$ **do**
  Set $\Delta x_0 := x_0^{\text{new}} - x_0$;
  Solve (43) for Δw and $\Delta \tau_{\mathscr{W}}$, and set $\Delta \tau_{\mathscr{W}^c} = 0$;
  Determine maximum steplength $\alpha_{\max}$ from (46);
  Set $\tilde{x_0} \leftarrow x_0 + \alpha_{\max} \Delta x_0$, $\tilde{w} \leftarrow w + \alpha_{\max} \Delta w$, $\tilde{\tau} \leftarrow \tau + \alpha_{\max} \Delta \tau$;
  if $\alpha_{\max} = 1$ **then**
    Set $w^{\text{new}} \leftarrow \tilde{w}$, $\tau^{\text{new}} \leftarrow \tilde{\tau}$, $\mathscr{A} \leftarrow \mathscr{W}$ and STOP;
  else if $\alpha_{\max} = \alpha_{\max,D}$ **then**
    Remove dual blocking constraint i_D from working set: $\mathscr{W} \leftarrow \mathscr{W} \setminus \{i_D\}$;
  else if $\alpha_{\max} = \alpha_{\max,P}$ **then**
    Add primal blocking constraint i_P to the working set: $\mathscr{W} \leftarrow \mathscr{W} \cup \{i_P\}$, possibly removing some other element of $\mathscr{A}$ if necessary to maintain full rank of $L_{\mathscr{W}}$;
  end if
  Set $x_0 \leftarrow \tilde{x}_0$, $w \leftarrow \tilde{w}$, $\tau \leftarrow \tilde{\tau}$;
end while

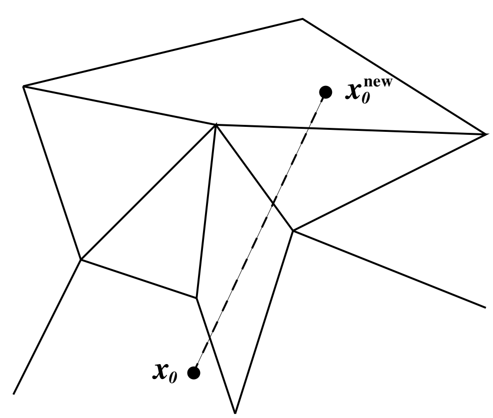

Fig. 1 Polyhedral decomposition of parameter space, showing path from x_0 to x_0^{new}

6 Software

Software packages are available online that facilitate efficient implementations of two of the approaches discussed in this chapter. The object-oriented code OOQP for structured convex quadratic programming [8, 7] can be customized to linear MPC problems; its $C++$ data structures and linear algebra modules can be tailored to problems of the form (15) and to solving systems of the form (23). The modeling framework YALMIP supports MPC; its web site shows several examples for setting up models and invoking underlying QP software (such as OOQP and general commercial solvers such as Gurobi).

The qpOASES solver [5, 6], which implements the approach described in Section 5.2, is available in an efficient and well maintained implementation.

References

1. Bemporad, A., Borrelli, F., Morari, M.: Model predictive control based on linear programming — the explicit solution. IEEE Transactions on Automatic Control **47**(12), 1974–1985 (2002)
2. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica **38**(3–20) (2002)
3. Ferreau, H.J.: An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. Ph.D. thesis, Ruprecht-Karls-Universit at Heidelberg Fakult at fu r Mathematik und Informatik (2006)
4. Ferreau, H.J., Bock, H.G., Diehl, M.: An online active set strategy to overcome the limitations of explicit MPC. International Journal of Robust and Nonlinear Control **18**(8), 816–830 (2008)
5. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. Mathematical Programming Computation **6**(4), 327–363 (2014)
6. Ferreau, H.J., Potschka, A., Kirches, C.: qpOASES webpage. http://www.qpOASES.org/ (2007–2017)
7. Gertz, E.M., Wright, S.J.: OOQP. http://www.cs.wisc.edu/∼swright/ooqp/
8. Gertz, E.M., Wright, S.J.: Object-oriented software for quadratic programming. ACM Transations on Mathematical Software **29**, 58–81 (2003)
9. Pannocchia, G., Rawlings, J.B., Wright, S.J.: Fast, large-scale model predictive control by partial enumeration. Automatica **43**, 852–860 (2007)
10. Pannocchia, G., Wright, S.J., Rawlings, J.B.: Partial enumeration MPC: Robust stability results and application to an unstable CSTR. Journal of Process Control **21**, 1459–1466 (2011)
11. Rao, C.V., Wright, S.J., Rawlings, J.B.: Application of interior-point methods to model predictive control. Journal of Optimization Theory and Applications **99**, 723–757 (1998)
12. Tondel, P., Johansen, T.A., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. Automatica **39**(5), 945–950 (2003)
13. Wright, S.J.: Applying new optimization algorithms to model predictive control. In: J.C. Kantor (ed.) Chemical Process Control-V, *AIChE Symposium Series*, vol. 93, pp. 147–155. CACHE Publications (1997)
14. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM Publications, Philadelphia (1997)