Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Analytical Solution of Input Constrained Reference Tracking Problems by Dynamic Programming

José B. Mare and José A. De Doná

***Abstract*— This paper is concerned with the explicit solution to constrained receding-horizon reference tracking control problems. The goal of this work is, for any scalar reference trajectory, to find the optimal control law for SISO linear systems such that a quadratic cost functional is minimised over a horizon of length N, subject to the satisfaction of input constraints, and under the assumption that the reference is known over the entire horizon. A global solution (i.e., valid in the entire data-space) for this problem, and for arbitrary horizon N, is derived analytically by using dynamic programming. The optimal solution is given by a piece-wise affine function of the data (the initial state of the system and the reference sequence), and the data-space is partitioned into a number of polyhedral regions, inside each of which a unique affine function is applied. From the dynamic programming solution, a clear relationship is exposed between input-constrained reference tracking problems and state estimation problems in the presence of constrained disturbances.**

I. INTRODUCTION

The explicit solution of constrained optimal control problems has attracted considerable attention recently (see, e.g., [5], [7]). This interest is, mainly, due to the fact that these problems constitute the core underlying optimisation problem that is solved, at each sampling time, by model predictive control algorithms (one of the most popular control methodologies used in industry at present). Model predictive control has been traditionally associated with process industries, where the plants are sufficiently slow and, hence, there is plenty of time to perform the on-line optimisations required in the implementation of the control law (see, e.g., [4]). Explicit solutions, on the other hand, provide a characterisation of the optimal solution that is pre-computed off-line, thus making the on-line numerical optimisation for such problems unnecessary. This has, at least, two potential benefits. One benefit is related to implementation aspects, as explicit solutions provide an alternative approach (to traditional numerical on-line optimisation) to implement the control algorithm, which could offer advantages in terms of the computational time required to compute on-line the control action. Another area in which explicit solutions, distinctively, offer benefits is in terms of the *verifiability* of the control scheme, since the complete knowledge of the controller structure allows to perform off-line tests such as closed-loop stability, robustness analysis, sensitivity analysis, etc., using traditional analytical and/or numerical methods. Hence, the complete closed-form characterisation provided by explicit solutions allows for such off-line analysis and a better understanding of the control algorithm (whether implemented using the off-line explicit solution or on-line optimisation methods).

In the present paper we obtain, using dynamic programming, the explicit solution to constrained receding-horizon reference tracking control problems. The result is based on a previous result obtained by the authors in [8] for the case of regulation problems. The new problem, studied here, has more generality than the previous one (as a regulation problem can be thought of as a zero-reference tracking one). Tracking problems, in spite of their importance in control applications, have received significantly less attention in the control literature as compared to regulation problems. And, although both problems share many common features, the more general problem of reference tracking poses some new challenges in, for example, obtaining explicit closed-form solutions. The technique used in [8], based on dynamic programming, is extended here to deal with the case of reference tracking. It is assumed that at each sampling time it is possible to preview the present and N future samples of the reference signal. The approach is quite general, in the sense that the reference trajectory, over the entire horizon N, is allowed to change at each sample instant.

One of the outcomes of the analytical solution to reference tracking problems presented here is that it exposes a clear connection to another optimisation-based problem of current interest; namely, state estimation in the presence of constraints. Drawing a parallel with a recent result reported by the authors in [9], a clear relationship emerges between input-constrained reference tracking problems and state estimation problems in the presence of constrained disturbances. In fact, as explained in this paper, both problems are identical under a suitable change of the system parameters and a different interpretation of the problem data.

II. Reference Tracking with Input Constraints

Consider the discrete-time linear state-space model

$$x_{k+1} = Ax_k + Bu_k, \tag{1a}$$

$$y_k = Cx_k, \tag{1b}$$

$$e_k = y_k^* - Cx_k, \tag{1c}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}$ are the state and control input respectively, $y_k \in \mathbb{R}$ is the system output, $y_k^* \in \mathbb{R}$ is a given output reference trajectory, and e_k is the tracking error. In

This work was supported by the Centre for Complex Dynamic Systems and Control, and The University of Newcastle, Australia

José B. Mare is with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan 2308, NSW, Australia jose.mare@studentmail.newcastle.edu.au

José A. De Doná is with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan 2308, NSW, Australia Jose.Dedona@newcastle.edu.au

0-7803-9568-9/05/$20.00 ©2005 IEEE

(1a) the pair (A, B) is assumed to be stabilisable, and the control input is required to satisfy the constraint $u_k \in \Omega$, where $\Omega \triangleq [\Delta_1, \Delta_2]$, $\Delta_1 < \Delta_2$.

The following notation will be employed. The *control sequence*, for some horizon N, is denoted $\mathbf{u} \triangleq \mathbf{u}_0 \triangleq \{u_0, u_1, \ldots, u_{N-1}\}$. For some initial time $r \in \{0, \ldots, N-1\}$, let $\mathbf{u}_r$ denote the *partial control sequence* $\mathbf{u}_r \triangleq \{u_r, u_{r+1}, \ldots, u_{N-1}\}$. By $\mathbf{u} \in \Omega^N$ ($\mathbf{u}_r \in \Omega^{N-r}$) we denote the case in which each element in the sequence satisfies $u_k \in \Omega$, $k = 0, \ldots, N-1$ ($k = r, \ldots, N-1$).

The solution of (1a) at time $k \geq r$ when the initial state at time r is x_r, and the control sequence is $\mathbf{u}_r$, is denoted $x_k^{\mathbf{u}_r}(x_r, r)$. To simplify notation, the initial time is dropped when it is zero; i.e., $x_k^{\mathbf{u}}(x_0) \triangleq x_k^{\mathbf{u}_0}(x_0, 0)$. The fixed-horizon optimal control problem considered is

$$\begin{aligned}\mathcal{P}_N^{\mathrm{c}} : V_0^{\mathrm{OPT}}&(x_0, \mu_N, y_{N-1}^*, \ldots, y_0^*) \\ &= \min_{\mathbf{u}} V_0(x_0, \mu_N, y_{N-1}^*, \ldots, y_0^*, \mathbf{u}),\end{aligned} \tag{2}$$

subject to the constraint $\mathbf{u} \in \Omega^N$. The cost $V_0(\cdot)$ in (2) is defined by

$$\begin{aligned}V_0(x_0, \mu_N, &y_{N-1}^*, \ldots, y_0^*, \mathbf{u}) \\ &= \sum_{k=0}^{N-1} [(y_k^* - Cx_k)^T Q (y_k^* - Cx_k) + u_k^T R u_k] \\ &\quad + (x_N - \mu_N)^T P_N (x_N - \mu_N) ,\end{aligned} \tag{3}$$

with initial state x_0 and $x_k = x_k^{\mathbf{u}}(x_0)$, and where μ_N is any vector in $\mathbb{R}^n$ satisfying $y_N^* = C\mu_N$. A suitable choice is the minimum-norm solution, given by $\mu_N = C^{\mathrm{T}}(CC^{\mathrm{T}})^{-1} y_N^*$ (see, e.g., [3]). The matrix Q is the tracking error weighting matrix, assumed to be positive semidefinite, R is the control weighting matrix, assumed to be positive definite. (Notice that, in the present context, both Q and R are 1×1 matrices, and the transpose operations in, e.g., expression (3) are not required. However, without loss of rigour, we have used the standard notation from the vector case.) The matrix P_N is the terminal state weighting matrix which is chosen as the positive definite matrix solution of the algebraic Riccati equation

$$P_N = A^{\mathrm{T}} P_N A + C^{\mathrm{T}} Q C - K^{\mathrm{T}} \bar{R} K, \tag{4}$$

where $K \triangleq \bar{R}^{-1} B^{\mathrm{T}} P_N A$, $\bar{R} \triangleq R + B^{\mathrm{T}} P_N B$. This particular choice for the terminal weighting matrix P_N is not essential for the developments of this paper, however it is well known (see, for example, [6]) that, for regulation problems, with this choice of *terminal weight* P_N, and provided that the horizon N is large enough, the resulting *receding-horizon* implementation of the control law gives an asymptotically stable closed-loop system and possesses all the properties of infinite-horizon optimal control. By the receding-horizon implementation it is understood the standard technique (also known as model predictive control) in which the first control action u_0 in the optimal control sequence $\mathbf{u}$ that minimises (2)–(3) is applied to system (1) and, as the state evolves to a new value in the next sampling time, the optimisation process is repeated over a horizon of length N (receding horizon).

III. Dynamic Programming

For $r = 0, \ldots, N-1$, the *partial value function* (or *optimal cost to go*), is defined as

$$\begin{aligned}V_r^{\mathrm{OPT}}(x_r, \mu_N, &y_{N-1}^*, \ldots, y_r^*) \\ &= \min_{\mathbf{u}_r} V_r(x_r, \mu_N, y_{N-1}^*, \ldots, y_r^*, \mathbf{u}_r),\end{aligned} \tag{5}$$

subject to the constraint $\mathbf{u}_r \in \Omega^{N-r}$, where the *partial cost* $V_r(\cdot)$ is defined by

$$\begin{aligned}V_r(x_r, \mu_N, &y_{N-1}^*, \ldots, y_r^*, \mathbf{u}_r) \\ &= \sum_{k=r}^{N-1} [(y_k^* - Cx_k)^T Q (y_k^* - Cx_k) + u_k^T R u_k] \\ &\quad + (x_N - \mu_N)^T P_N (x_N - \mu_N) ,\end{aligned} \tag{6}$$

with $x_k = x_k^{\mathbf{u}_r}(x_r, r)$, $k = r, r+1, \ldots, N$. We refer to $V_r^{\mathrm{OPT}}(\cdot)$ as the partial value function (or, just the value function) 'at time r', meaning that the (partial) value function 'starts at time r'. We also define

$$V_N^{\mathrm{OPT}}(x_N, \mu_N) \triangleq (x_N - \mu_N)^T P_N (x_N - \mu_N). \tag{7}$$

To solve problem $\mathcal{P}_N^{\mathrm{c}}$ defined in (2)–(3), dynamic programming (see [1]) will be used, which is based on the Principle of Optimality: *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.* Applying the principle of optimality to problem $\mathcal{P}_N^{\mathrm{c}}$, we have

$$\begin{aligned}V_N^{\mathrm{OPT}}&(x_N, \mu_N) = (x_N - \mu_N)^T P_N (x_N - \mu_N), \\ V_r^{\mathrm{OPT}}&(x_r, \mu_N, y_{N-1}^*, \ldots, y_r^*) \\ &= \min_{u_r \in \Omega} \Big\{ (y_r^* - Cx_r)^T Q (y_r^* - Cx_r) + u_r^T R u_r \\ &\quad + V_{r+1}^{\mathrm{OPT}}(Ax_r + Bu_r, \mu_N, y_{N-1}^*, \ldots, y_{r+1}^*) \Big\},\end{aligned} \tag{8}$$

for $r = 0, \ldots, N-1$. Thus, by using the principle of optimality, the sequence of optimal costs to go $\{V_0^{\mathrm{OPT}}(\cdot), V_1^{\mathrm{OPT}}(\cdot), \ldots, V_N^{\mathrm{OPT}}(\cdot)\}$ and the sequence of optimal controls $\{u_0^{\mathrm{OPT}}, u_1^{\mathrm{OPT}}, \ldots, u_{N-1}^{\mathrm{OPT}}\}$ are obtained by solving (8) recursively.

IV. Analytical Solution

In the sequel, I denotes the identity matrix of the same size as A, $0_{i \times j}$ denotes the zero matrix with i rows and j columns, and c denotes a generic constant. Let us define

$$\mathcal{Y}_N \triangleq \mu_N, \quad \mathcal{Y}_{N-1} \triangleq \begin{bmatrix} \mu_N \\ y_{N-1}^* \end{bmatrix}, \ldots, \; \mathcal{Y}_0 \triangleq \begin{bmatrix} \mu_N \\ y_{N-1}^* \\ \vdots \\ y_0^* \end{bmatrix}. \tag{9}$$

Let us also define $\alpha_N \triangleq I, \beta_N \triangleq 0_{n \times 1}$ such that $x_N^* \triangleq \alpha_N \mathcal{Y}_N + \beta_N = \mu_N$. The partial value function at time N

is considered first, which from (7) and the above definitions can be written as

$$V_N^{\mathrm{OPT}}(x_N, x_N^*) = (x_N - x_N^*)^{\mathrm{T}} P_N (x_N - x_N^*).$$

Then, the partial value function at time $N-1$ is expressed, using the Principle of Optimality (8), as

$$\begin{aligned} V_{N-1}^{\mathrm{OPT}}(x_{N-1}, x_N^*, y_{N-1}^*) = \min_{u_{N-1}\in\Omega} \Big\{ & u_{N-1}^{\mathrm{T}} R u_{N-1} \\ & + (y_{N-1}^* - C x_{N-1})^{\mathrm{T}} Q (y_{N-1}^* - C x_{N-1}) \\ & + V_N^{\mathrm{OPT}}(A x_{N-1} + B u_{N-1}, x_N^*) \Big\}. \end{aligned} \tag{10}$$

Consider first the minimisation problem (10) in the absence of constraints. Substituting the expression for V_N^{OPT} into V_{N-1}^{OPT}, taking derivatives with respect to u_{N-1} and setting to zero, the expression of the *unconstrained* u_{N-1}^{unc} that minimises V_{N-1}^{OPT} is obtained as:

$$u_{N-1}^{\mathrm{unc}} = \left[B^{\mathrm{T}} P_N B + R\right]^{-1} B^{\mathrm{T}} P_N (x_N^* - A x_{N-1}). \tag{11}$$

Notice that the objective function in (10) is a quadratic function of u_{N-1} whose *unconstrained* minimum is achieved at u_{N-1}^{unc} computed above. From the convexity of the objective function it follows that the *constrained* optimum, u_{N-1}^{OPT}, is given by the point in the allowed interval $\Omega = [\Delta_1, \Delta_2]$ that is closest in distance to the unconstrained optimum u_{N-1}^{unc}. Hence, three different cases arise, depending on whether $u_{N-1}^{\mathrm{unc}} < \Delta_1$, $\Delta_1 \le u_{N-1}^{\mathrm{unc}} \le \Delta_2$, or $u_{N-1}^{\mathrm{unc}} > \Delta_2$. It follows that the optimal *constrained* solution can be written as

$$u_{N-1}^{\mathrm{OPT}} = L_N Z_N (x_N^* - A x_{N-1}) + h_N, \tag{12}$$

where

$$Z_N = \left[B^{\mathrm{T}} P_N B + R\right]^{-1} B^{\mathrm{T}} P_N, \tag{13}$$

$$L_N = \begin{cases} 1 & \text{if } \Delta_1 \le Z_N(x_N^* - A x_{N-1}) \le \Delta_2, \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

$$h_N = \begin{cases} 0 & \text{if} \quad \Delta_1 \le Z_N(x_N^* - A x_{N-1}) \le \Delta_2, \\ \Delta_1 & \text{if} \quad Z_N(x_N^* - A x_{N-1}) < \Delta_1, \\ \Delta_2 & \text{if} \quad Z_N(x_N^* - A x_{N-1}) > \Delta_2. \end{cases} \tag{15}$$

To simplify the notation, in the sequel we make the assumption that matrix A is invertible. Note that this assumption is not very restrictive since any matrix A obtained from the time-discretisation of an underlying continuous-time system will satisfy it.

Substituting (12) into (10) and completing squares, the minimum attained in (10) can be written as

$$V_{N-1}^{\mathrm{OPT}}(x_{N-1}, x_{N-1}^*) = (x_{N-1} - x_{N-1}^*)^{\mathrm{T}} P_{N-1} (x_{N-1} - x_{N-1}^*) + c,$$

where

$$\begin{aligned} x_{N-1}^* = & \left[(I - P_{N-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} \quad P_{N-1}^{-1} C^{\mathrm{T}} Q\right] \begin{bmatrix} x_N^* \\ y_{N-1}^* \end{bmatrix} \\ & - (I - P_{N-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} B h_N, \end{aligned} \tag{16}$$

$$\begin{aligned} P_{N-1} = A^{\mathrm{T}} \big[& (I - L_N B Z_N)^{\mathrm{T}} P_N (I - L_N B Z_N) \\ & + L_N Z_N^{\mathrm{T}} R Z_N \big] A + C^{\mathrm{T}} Q C. \end{aligned} \tag{17}$$

The fact that, from the definitions (14) and (15), $L_N^2 = L_N$ and $L_N h_N = 0 \ \forall\ x_N^*,\ x_{N-1} \in \mathbb{R}^n$ has been used to simplify the expressions. Note that when $L_N = 1$ (that is, when the constraints are inactive, $\Delta_1 \le u_{N-1}^{\mathrm{unc}} \le \Delta_2$), equation (17) reduces to the standard Riccati equation. Substituting $x_N^* \triangleq \alpha_N \mathcal{Y}_N + \beta_N$ into (16), we have

$$\begin{aligned} x_{N-1}^* = & \left[(I - P_{N-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} \alpha_N \quad P_{N-1}^{-1} C^{\mathrm{T}} Q\right] \begin{bmatrix} \mathcal{Y}_N \\ y_{N-1}^* \end{bmatrix} \\ & + (I - P_{N-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} (\beta_N - B h_N) \\ = & \ \alpha_{N-1} \mathcal{Y}_{N-1} + \beta_{N-1}, \end{aligned} \tag{18}$$

where we have defined

$$\alpha_{N-1} \triangleq \left[(I - P_{N-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} \alpha_N \quad P_{N-1}^{-1} C^{\mathrm{T}} Q\right], \tag{19}$$

$$\beta_{N-1} \triangleq (I - P_{N-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} (\beta_N - B h_N). \tag{20}$$

From the previous discussion, if the induction hypothesis

$$V_k^{\mathrm{OPT}}(x_k, x_k^*) = (x_k - x_k^*)^{\mathrm{T}} P_k (x_k - x_k^*) + c,$$

with $x_k^* = \alpha_k \mathcal{Y}_k + \beta_k$, were introduced, following the same steps as before, an equivalent set of equations (10)–(20), with subindex k instead of N, would be obtained, proving by induction the validity of these equations for $k = 1, \ldots, N$.

Based on the matrices α_k of size $n \times (n+N-k)$ obtained by solving the recursive relations given by (19) and (20) starting at $\alpha_N \triangleq I$, $\beta_N \triangleq 0_{n\times 1}$, for $k = N, N-1, ..., 0$ we will define matrices α_k^* of constant size $n \times (N+n)$, for which the first $n+N-k$ columns correspond to the matrix α_k, and the last k columns are zeros:

$$\alpha_k^* \triangleq [\alpha_k \ 0_{n\times k}]. \tag{21}$$

In this way we can rewrite each x_k^* in terms of the full reference vector $\mathcal{Y}_0$ (cf. (9)), i.e., for $k = 0, ..., N$, we have:

$$x_k^* = \alpha_k^* \mathcal{Y}_0 + \beta_k. \tag{22}$$

Notice that the results obtained thus far define the fixed-horizon optimal control sequence $\{u_0^{\mathrm{OPT}}, u_1^{\mathrm{OPT}}, \ldots, u_{N-1}^{\mathrm{OPT}}\}$ implicitly, in terms of the resulting state sequence (cf. equation (12), and recall that this expression is also valid when k replaces N, for $k = 1, ..., N$). However, as we show next, it is straightforward to express the optimal control sequence as a function of the initial state of the system, x_0, and the rest of the data of the problem (i.e., the reference trajectory $y_0^*, y_1^*, \ldots, y_{N-1}^*$, and μ_N).

The elements of the sequence of states driven by the optimal constrained input sequence, denoted by $\mathbf{x}^{\mathrm{OPT}}(x_0, \mu_N, y_{N-1}^*, \ldots, y_0^*)$, will be derived by induction. The initial state in the sequence can be written as

$$x_0^{\mathrm{OPT}} = \phi_0 x_0 + \gamma_0 \mathcal{Y}_0 + \delta_0, \tag{23}$$

where we define $\phi_0 \triangleq I$, $\gamma_0 \triangleq 0_{n\times(N+n)}$, and $\delta_0 \triangleq 0_{n\times 1}$. Assuming $x_{k-1} = \phi_{k-1} x_0 + \gamma_{k-1} \mathcal{Y}_0 + \delta_{k-1}$, from (1) and (12) (with subindex k instead of N in the latter equation), and from (22), it is easy to prove by induction that

$$x_k^{\mathrm{OPT}} = \phi_k x_0 + \gamma_k \mathcal{Y}_0 + \delta_k, \tag{24}$$

with

$$\phi_k = (I - L_k B Z_k) A \phi_{k-1}, \quad (25)$$

$$\gamma_k = (I - L_k B Z_k) A \gamma_{k-1} + L_k B Z_k \alpha_k^*, \quad (26)$$

$$\delta_k = (I - L_k B Z_k) A \delta_{k-1} + B(L_k Z_k \beta_k + h_k), \quad (27)$$

starting at $\phi_0 \triangleq I$, $\gamma_0 \triangleq 0_{n\times(N+n)}$, and $\delta_0 \triangleq 0_{n\times 1}$.

We are now ready to express the optimal control sequence in terms of the initial state of the problem and the reference vector. We do this in the next subsection.

Main Result

The following theorem summarises the derivations presented so far, that provide the solution of problem $\mathcal{P}_N^c$ defined by (2)–(3).

Theorem 1: Consider the linear system (1) and the fixed-horizon optimal control problem $\mathcal{P}_N^c$ defined in (2)–(3). Then, given the initial state x_0 and the full data vector $\mathcal{Y}_0 \triangleq \left[\mu_N^{\mathrm{T}}, y_{N-1}^*, \ldots, y_0^*\right]^{\mathrm{T}}$, the control sequence that minimises $\mathcal{P}_N^c$ is given by

$$u_k^{\mathrm{OPT}} = L_{k+1} Z_{k+1} [(\alpha_{k+1}^* - A\gamma_k)\mathcal{Y}_0 - A\phi_k x_0 + (\beta_{k+1} - A\delta_k)] + h_{k+1}, \quad (28)$$

for $k = 0, ..., N-1$.

In (28), L_k and h_k, $k = 1, ..., N$, are given by

$$L_k = \begin{cases} 1 & \text{if } \Delta_1 \le Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 \\ & \qquad +(\beta_k - A\delta_{k-1})] \le \Delta_2, \\ 0 & \text{otherwise,} \end{cases}$$

$$h_k = \begin{cases} 0 & \text{if } \Delta_1 \le Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 \\ & \qquad +(\beta_k - A\delta_{k-1})] \le \Delta_2, \\ \Delta_1 & \text{if } Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 \\ & \qquad +(\beta_k - A\delta_{k-1})] < \Delta_1, \\ \Delta_2 & \text{if } Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 \\ & \qquad +(\beta_k - A\delta_{k-1})] > \Delta_2; \end{cases}$$

Z_k, $k = 1, ..., N$, are given by

$$Z_k = \left[B^{\mathrm{T}} P_k B + R\right]^{-1} B^{\mathrm{T}} P_k,$$

with P_k, $k = 1, ..., N$, given by the following recursive equation, starting at the terminal weighting matrix P_N,

$$P_{k-1} = A^{\mathrm{T}}\left[(I - L_k B Z_k)^{\mathrm{T}} P_k (I - L_k B Z_k) + L_k Z_k^{\mathrm{T}} R Z_k\right] A + C^{\mathrm{T}} Q C;$$

α_k^* and β_k, $k = 1, ..., N$, starting at $\alpha_N \triangleq I$, $\beta_N \triangleq 0_{n\times 1}$, are given by the recursive equations

$$\alpha_{k-1} = \left[(I - P_{k-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} \alpha_k \quad P_{k-1}^{-1} C^{\mathrm{T}} Q\right],$$

$$\alpha_k^* = [\alpha_k \; 0_{n\times k}],$$

$$\beta_{k-1} = (I - P_{k-1}^{-1} C^{\mathrm{T}} Q C) A^{-1} (\beta_k - B h_k).$$

Finally, ϕ_k, γ_k, and δ_k, $k = 1, ..., N$, are given by the recursive equations

$$\phi_k = (I - L_k B Z_k) A \phi_{k-1},$$

$$\gamma_k = (I - L_k B Z_k) A \gamma_{k-1} + L_k B Z_k \alpha_k^*,$$

$$\delta_k = (I - L_k B Z_k) A \delta_{k-1} + B(L_k Z_k \beta_k + h_k),$$

starting at $\phi_0 \triangleq I$, $\gamma_0 \triangleq 0_{n\times(N+n)}$, and $\delta_0 \triangleq 0_{n\times 1}$.

Remark 2: Note that as each pair $\{L_k, h_k\}$, for $k = 1, \ldots, N$, used in the calculations can take 3 different sets of values corresponding to u_{k-1}^{unc} saturating or not (i.e., $\{1, 0\}$, $\{0, \Delta_1\}$, or $\{0, \Delta_2\}$), there are 3^N possible sequences $\mathbf{u}_0^{\mathrm{OPT}}$. The methodology we are presenting consists in calculating the 3^N possibilities, and determining in which region of the data-space each of these possibilities is valid (see [8]).

From Theorem 1 it is easy to obtain the regions for which each of the 3^N possible sequences $\mathbf{u}_0^{\mathrm{OPT}}$, as discussed in Remark 2, is valid. The result is presented next.

Corollary 3: The region of the data-space $\{\mathcal{Y}_0, x_0\}$ where the *constrained* control sequence $\mathbf{u}_0^{\mathrm{OPT}}$ computed from Theorem 1 for any particular choice of $\{L_k, h_k\}$, for $k = 1, \ldots, N$, is optimal, is given by the intersection of linear inequalities chosen according to one of the following cases, for each $k = 1, \ldots, N$,

$$\Delta_1 \le Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 + (\beta_k - A\delta_{k-1})] \le \Delta_2, \text{ if } \{L_k, h_k\} = \{1, 0\}, \quad (29)$$

or

$$Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 + (\beta_k - A\delta_{k-1})] < \Delta_1, \text{ if } \{L_k, h_k\} = \{0, \Delta_1\}, \quad (30)$$

or

$$Z_k[(\alpha_k^* - A\gamma_{k-1})\mathcal{Y}_0 - A\phi_{k-1}x_0 + (\beta_k - A\delta_{k-1})] > \Delta_2, \text{ if } \{L_k, h_k\} = \{0, \Delta_2\}. \quad (31)$$

Note that the regions, computed from Corollary 3 as the intersection of a number of linear inequalities, constitute a polyhedral partition of the data-space $\{\mathcal{Y}_0, x_0\}$. Although for large horizons N the number of possible regions, 3^N, is potentially very large, in practice a considerable number of these regions are actually empty, arising from infeasible sequences of active/inactive constraints. Thus, the complexity of the resulting partition, and of the controller structure, can be considerably reduced.

Note as well that, inside each polyhedral region, a unique optimal control sequence is valid, given by equation (28) with all the coefficients (Z_{k+1}, α_{k+1}, γ_{k+1}, etc.) computed for the particular combination of $\{L_k, h_k\}$, $k = 1, \ldots, N$, corresponding to that specific region.

Online Receding-horizon Implementation

The first control law in the optimal control sequence (28) can be rewritten, using $\phi_0 = I$, $\gamma_0 = 0_{n\times(N+n)}$, and $\delta_0 = 0_{n\times 1}$, as:

$$u_0^{\mathrm{OPT}} = F_1 \begin{bmatrix} \mathcal{Y}_0 \\ x_0 \end{bmatrix} + G_1, \qquad (32)$$

where $F_1 = [L_1 Z_1 \alpha_1^* \ \ -L_1 Z_1 A]$, and $G_1 = L_1 Z_1 \beta_1 + h_1$.

Thus, the on-line receding-horizon implementation of the controller reduces to: Given the data $\mathcal{Y}_0$ and x_0 (where, using the standard notation in receding-horizon control, the subindex 0 stands for the current time k), find the corresponding region and, via the simple affine function evaluation (32), obtain the optimal *current* control action (i.e., the first element u_0^{OPT} of the optimal fixed-horizon control sequence). The coefficients in equation (32) can be precomputed and stored off-line (for all the possible combinations of $\{L_k, h_k\}$ that result in non-empty regions in the data-space partition) and then the main on-line computational requirement is that of determining the region to which the current data belongs to.

Example

Consider the discrete time linear system given by (1), with matrices $A = [1.6375, -0.6703; 1, 0]$, $B = [0.2500; 0]$, $C = [0.0701, 0.0613]$, corresponding to a sampling period $T_s = 0.2$.
For horizon $N = 5$, Theorem 1 and Corollary 3 are used to implement a receding-horizon strategy aimed at tracking a reference trajectory given by sampling a sinusoidal function, $\sin \omega t$. For the fixed-horizon cost function, the values $Q = 10$, $R = 0.5$ are chosen, P_N is chosen as the positive definite matrix solution of (4), and $\mu_N = C^{\mathrm{T}}(CC^{\mathrm{T}})^{-1} y_N^*$. The control input u_k is constrained to lie in the interval $[-1.5, 1.5]$.

A partition of the data-space is obtained, as explained in Corollary 3. To illustrate the solution for horizon $N = 5$, a projection, corresponding to $y_0^* = \cdots = y_5^* = 0$, of this partition onto the plane $\tilde{x}_0 = T x_0$ is shown in Figure 1. (Note: the state-space coordinates have been transformed by a $\frac{2\pi}{7}$-rotating matrix T so as to display the regions in a clearer way.) The values of the coefficients in the affine control law (32) (for the original coordinates x_0), corresponding to some of these regions, are shown in Table I.

Figure 2 shows the reference trajectory (with frequency $\omega = \pi/12$) and the output of the system obtained with the receding-horizon controller. The initial condition x_0 was arbitrarily chosen as $[4 \ \ -2]^{\mathrm{T}}$. The control signal is plotted in the lower subplot of the figure.

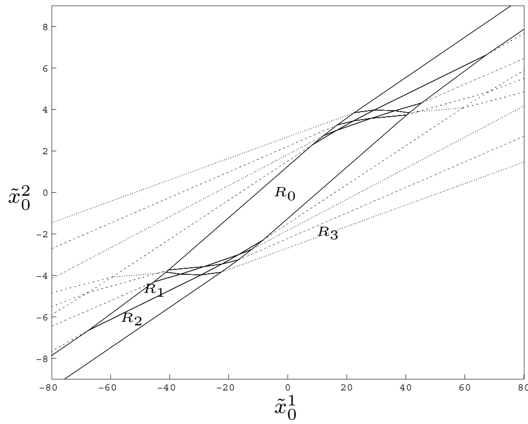

Fig. 1. Projection of the data-space partition onto the plane $\tilde{x}_0 = (\tilde{x}_0^1, \tilde{x}_0^2)$ for horizon $N = 5$ (cut corresponding to $y_0^* = \cdots = y_5^* = 0$).

TABLE I
OPTIMAL CONTROL LAW $u_0^{\mathrm{OPT}} = F_1[\mathcal{Y}_0^{\mathrm{T}}\ x_0^{\mathrm{T}}]^{\mathrm{T}} + G_1$ FOR SOME OF THE REGIONS IN FIGURE 1

R_i	F_1	G_1
R_0	[0.285 -0.082 0.655 0.699 0.606 0.269 0 -1.023 0.627]	0
R_1	[0.546 -0.219 0.654 0.615 0.526 0.247 0 -1.247 0.796]	0.813
R_2	[0.781 -0.321 0.842 0.733 0.535 0.213 0 -1.619 1.052]	1.680
R_3	[0 0 0 0 0 0 0 0 0]	-1.5
⋮	⋮	⋮

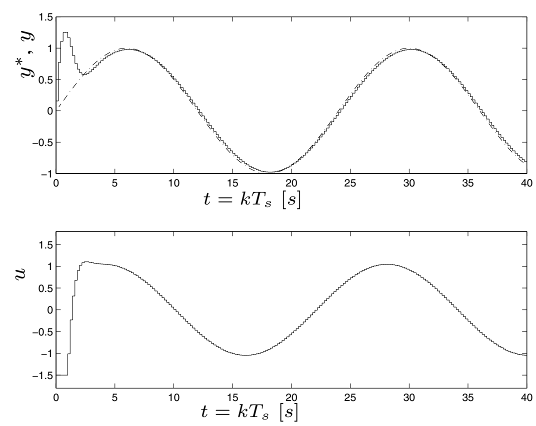

Fig. 2. Upper plot: reference trajectory (dashed-dot line) and system output (stairwise continuous line). Lower plot: constrained control input. Horizon $N = 5$.

V. Relationship to State Estimation with Constrained Disturbances

In this section we will briefly present some related results obtained for a different problem; namely, the problem of estimating the states of a linear system under the assumption that the process disturbance is bounded to satisfy some, known a priori, constraint. The exposition is very brief (for the details, see [9]) as our aim here is to draw some parallels and symmetries between both problems (i.e., reference tracking and state estimation), that become evident from their dynamic programming solutions. In fact, we conclude that, with a suitable change of the system parameters (as shown in table II below), both problems are actually identical and the same solution (as presented in Section IV) applies to both of them.

Consider the discrete-time linear state-space model

$$x_{k+1} = A_e x_k + B_e w_k, \tag{33a}$$
$$y_k = C_e x_k + v_k, \tag{33b}$$

where $x_k \in \mathbb{R}^n$, and w_k, y_k, and $v_k \in \mathbb{R}$. Suppose that x_0, $\{w_k\}$, $\{v_k\}$ are i.i.d. sequences, mutually independent, x_0 and $\{v_k\}$ have Gaussian distributions, and $\{w_k\}$ has a truncated Gaussian distribution (i.e., every element in the sequence $\{w_k\}$ is constrained to lie on some interval Ω_e in $\mathbb{R}$).

Given the observations $\mathbf{y}_N^d = \{y_1^d, \dots y_N^d\}$ and the mean value of x_0, denoted by μ_0^e, the aim is to obtain the *joint a posteriori most probable* (JAPMP) state estimates $\hat{\mathbf{x}}_N = \{\hat{x}_0, \dots \hat{x}_N\}$. That is, based on the knowledge of the *a posteriori* distribution of $\mathbf{x}_N = \{x_0, \dots x_N\}$ given $\mathbf{y}_N = \{y_1, \dots y_N\}$, denoted $p_{\mathbf{x}_N|\mathbf{y}_N}$, and on the observations $\mathbf{y}_N^d$, we want to determine the vector $\hat{\mathbf{x}}_N$ that solves the following optimization problem

$$\hat{\mathbf{x}}_N^{\mathrm{OPT}} \triangleq \arg\max_{\hat{\mathbf{x}}_N} p_{\mathbf{x}_N|\mathbf{y}_N}(\hat{\mathbf{x}}_N|\mathbf{y}_N^d). \tag{34}$$

This problem can be formulated as a quadratic program (see [2] for the details), as follows:

Given the observations $\mathbf{y}_N^d = \{y_1^d, \dots y_N^d\}$ and the mean value of x_0, μ_0^e, solve

$$\mathcal{P}_N^e: \ J_N^{\mathrm{OPT}}(\mu_0^e, \mathbf{y}_N^d) \triangleq \min J_N(\{\hat{x}_k\}, \{\hat{v}_k\}, \{\hat{w}_k\}), \tag{35}$$

subject to:

$$\hat{x}_{k+1} = A_e \hat{x}_k + B_e \hat{w}_k \quad \text{for } k = 0, \dots, N-1, \tag{36}$$
$$\hat{v}_k = y_k^d - C_e \hat{x}_k \quad \text{for } k = 1, \dots, N, \tag{37}$$
$$\hat{w}_k \in \Omega_e \quad \text{for } k = 0, \dots, N-1 \tag{38}$$

where

$$J_N(\{\hat{x}_k\}, \{\hat{v}_k\}, \{\hat{w}_k\}) \triangleq \sum_{k=0}^{N-1} \hat{w}_k^{\mathrm{T}} Q_e^{-1} \hat{w}_k + \sum_{k=1}^{N} \hat{v}_k^{\mathrm{T}} R_e^{-1} \hat{v}_k + (\hat{x}_0 - \mu_0^e)^{\mathrm{T}} P_{e0}^{-1} (\hat{x}_0 - \mu_0^e). \tag{39}$$

Based on recent results concerning the analytical solution of problem $\mathcal{P}_N^e$ defined above (see [9]), we describe below how to obtain the solution of problem $\mathcal{P}_N^e$ using Theorem 1 of Section IV, with a suitable change of system parameters.

Given the state estimation problem $\mathcal{P}_N^e$ with constrained process noise defined in (35)–(39), there exists an associated tracking problem $\mathcal{P}_N^c$ defined by (1) and (2)–(3), with parameters obtained from those of problem $\mathcal{P}_N^e$ according to Table II, such that solving the tracking problem (i.e., using Theorem 1 to obtain the matrices α_k^*, β_k, ϕ_k, γ_k, and δ_k, for $k = 0, \dots, N$) for this new set of parameters, the solution for the original estimation problem is obtained as follows:

1) The reference vector $\mathcal{Y}_0$ in the tracking problem (cf. (9)) is set to $\mathcal{Y}_0 = \left[\mu_0^{e\mathrm{T}}, y_1^d, \dots, y_N^d\right]^{\mathrm{T}}$.
2) The optimal estimate for $\hat{x}_N$ is obtained as
$$\hat{x}_N = \alpha_0^* \mathcal{Y}_0 + \beta_0. \tag{40}$$
3) The rest of the elements in the sequence $\hat{\mathbf{x}}_N = \{\hat{x}_0, \dots \hat{x}_N\}$ are obtained, for $k = 1, \dots, N$, as
$$\hat{x}_{N-k} = \phi_k \hat{x}_N + \gamma_k \mathcal{Y}_0 + \delta_k. \tag{41}$$

As explained in Remark 2, again there are 3^N possible sequences $\hat{\mathbf{x}}_N$, according to which choices of $\{L_k, h_k\}$ have been used in the recursions, for $k = 1, \dots, N$. The regions where a particular sequence $\hat{\mathbf{x}}_N$ is optimal can be determined by means of Corollary 3, using, instead of x_0, the expression for $\hat{x}_N$ given by equation (40).

TABLE II

PARAMETERS TRANSLATION

$\mathcal{P}_N^c$	$\mathcal{P}_N^e$	$\mathcal{P}_N^c$	$\mathcal{P}_N^e$
$A =$	A_e^{-1}	$R =$	Q_e^{-1}
$B =$	$-A_e^{-1} B_e$	$Q =$	R_e^{-1}
$C =$	C_e	$P_N =$	P_{e0}^{-1}
$\mu_N =$	μ_0^e	$\Omega =$	Ω_e

VI. CONCLUSIONS

An analytical solution for the input-constrained reference tracking problem was derived using dynamic programming, that comprises a piece-wise affine control law structure and a partition of the data-space in regions where each affine control law is valid. An example was provided to illustrate the behaviour of the optimal solution for a sinusoidal reference trajectory. Some connections, emerging from the dynamic programming solution, between the constrained reference tracking problem and a related constrained state estimation problem were discussed.

REFERENCES

[1] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, USA, 1957.

[2] G. C. Goodwin, M. M. Seron, and J. A. De Doná. *Constrained Control and Estimation. An Optimisation Approach*, Springer-Verlag, 2005.

[3] B. D. O. Anderson, and J .B. Moore. *Optimal Control: Linear Quadratic Methods* Prentice-Hall, 1989.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Skokaert. Constrained model predictive control: Stability and optimality. *Automatica*, Vol. 36:pp. 789–814, 2000.

[5] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, Vol. 38:pp. 3–20, 2002.

[6] J. Rawlings and K. Muske. Stability of constrained receding horizon control. *IEEE Trans. on Automatic Control*, Vol. 38, No. 10:pp. 1512–1516, 1993.

[7] M. Seron, G. Goodwin, and J. De Doná. Characterisation of receding horizon control for constrained linear systems. *Asian Journal of Control*, Vol. 5:pp. 271–286, 2003.

[8] J. Mare and J. De Doná , Use of Dynamic Programming for the Analytical Solution of Input-Constrained LQR Problems. *5th Asian Control Conference*, pp. 441–447, 2004.

[9] J. Mare and J. De Doná , Dynamic Programming Solution of State Estimation Problems with Constrained Disturbances. *16th IFAC World Congress*, 2005.