Proceedings of the American Control Conference
Philadelphia, Pennsylvania • June 1998

A computationally efficient constrained predictive control law

J.A. Rossiter M. J. Rice J. Schuurmans
Department of Mathematical Sciences
Loughborough University
Leicestershire LE11 3TU, UK
e-mail: J.A.Rossiter@lboro.ac.uk
Tel. (44) 1509 222860

B. Kouvaritakis
Department of Engineering Sciences
Parks Road
Oxford, OX1 3PJ, UK
email: Basil.Kouvaritakis@eng.ox.ac.uk
Tel: (44) 1865 273105

Abstract

A predictive control algorithm is presented that uses a weighted sum of Linear quadaratic (LQ) optimal predictions and 'Mean Level' predictions. Each time step only a single weighting parameter is solved for using a Linear Program problem hence giving a low computational load. The controller achieves LQ-optimality when constraints are inactive, and remains feasible (under mild conditions) if constraints are active.

1 Introduction

Recently, several predictive control algorithms have been developed that reduce to an LQ control law when constraints are inactive, e.g. [6], [4], [3]. The key to the approach is to base the predictions on those that would arise from implementing an LQ feedback control law and then to add some perturbation (degrees of freedom) to the predictions which can be utilised for constraint handling. If the number of degrees of freedom (d.o.f.) is large enough than one can also achieve constrained LQ optimality, but in general for small numbers of d.o.f. LQ optimality is achieved only as constraints become inactive. The differences between the above algorithms are unimportant for the context of this paper, in fact what are more important are their similarities.

- They all must utilise infinite prediction and constraint handling horizons
- They require a quadratic programming (QP) solution with possibly a large number of constraints and d.o.f.

The 'infinite' constraint horizon can usually be reduced to a finite horizon (see [1]) and the infinite prediction horizon can be handled efficiently via Lyapunov equations, hence the main computational effort is the solution of the QP. These types of algorithm will be referred to as Constrained Linear Quadratic Regulator ($CLQR$).

The aim of this paper is to propose a slightly different structure of predictions, still containing the LQ optimal predictions, but expressing the d.o.f. in terms of only one variable. By utilising only one d.o.f., the QP problem reduces to a simple set of inequality checks on one variable which is computationally trivial so long as the dependence on the d.o.f. is simple; here the dependence will be linear. This work forms a development of the prediction equations presented in [3] and [2]. The former of these suggested a particular form for the perturbation term about the LQ optimal, but utilised several d.o.f. while the latter provided a framework for guaranteed 'feasibility' and with one d.o.f., but only in the uncertainty free case (it assumed that a linear interpolation between the previous predicted input and the current unconstrained optimal would contain at least one feasible solution).

To illustrate this, let the constraints be given as

$$\mathbf{g}\alpha - \mathbf{d} \leq 0 \tag{1}$$

where $\mathbf{g}$, $\mathbf{d}$ depend in a simple linear manner on $\mathbf{x}_0$ the current state measurement and α, $0 \leq \alpha \leq 1$ is a scalar d.o.f. If furthermore, the performance index to be minimised can be shown to be of the form

$$J = a + c\alpha^2, \quad b > 0 \tag{2}$$

then the optimal policy is to minimise α subject to (1), which is computationally trivial in that it amounts to checking each constraint equation (represented by the elements of $\mathbf{g}, \mathbf{d}$) only once. By assumption $\mathbf{g} - \mathbf{d} \leq 0$ so $\alpha = 1$ is always a feasible choice hence an α exists such that constraints (1) can be satisified.

In this paper we show that there exists a parameterisation of predictive controller in terms of one d.o.f. which has the above properties and hence could be used for fast sampling rates or where low computational load is critical. We first introduce the model and constraints and then illustrate how stable and convergent system

A prediction is feasible if it satisfies input/state constraints and is stabilising predictions can be computed. Section 3 then details the algorithm and section 4 gives example simulations. Section 5 contains the stability proof and this is followed by the conclusions.

2 The model and prediction equations

For convenience here we will utilise a standard discrete state-space model (where as usual in model predictive control (MPC) an integrator is included in the model hence $\mathbf{u}$ is an incremental input)

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k; \quad \mathbf{y}_k = C\mathbf{x}_k \tag{3}$$

where $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$, $\mathbf{y} \in R^l$. Transfer function models can be used in analagous fashion, e.g. [4].

Let system state, input and output constraints be time invariant and at each sample instant be given as

$$\left.\begin{array}{lll} \underline{u}_i & \le u_i \le & \overline{u}_i, \quad i = 1, ..., m \\ \underline{y}_i & \le y_i \le & \overline{y}_i, \quad i = 1, ..., l \\ \underline{x}_i & \le x_i \le & \overline{x}_i, \quad i = 1, ..., n \end{array}\right\} \tag{4}$$

where u_i, y_i, x_i are the components of the vectors $\mathbf{u}$, $\mathbf{y}$, $\mathbf{x}$. For simplicity of presentation, henceforth we take the setpoint to be $\mathbf{r} = 0$, but this is not a limitation.

2.1 LQ optimal predictions

Given a performance index

$$J = \sum_{k=0}^{\infty} \mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k \tag{5}$$

let the corresponding LQ optimal state feedback minimising (5) be

$$\mathbf{u}_k = -K_L \mathbf{x}_k \tag{6}$$

Implementing this state feedback starting from an initial condition $\mathbf{x}_0$, the state and input predictions ($\hat{\mathbf{x}}$, $\hat{\mathbf{u}}$ can be computed as

$$\hat{\mathbf{x}}_k = \Phi_L^k \mathbf{x}_0; \quad \hat{\mathbf{u}}_k = -K_L \Phi_L^{k-1} \mathbf{x}_0, \quad \Phi_L = A - BK_L \tag{7}$$

2.2 Mean-level predictions

By mean level (ML) in this paper we refer to predictions arising from a very detuned state feedback, hence one that in general will not demand inputs likely to violate constraints; these predictions can therefore be used to ensure feasibility. For instance, if there are only input constraints a suitable performance index could be

$$J = \sum_{k=0}^{\infty} \mathbf{x}_k^T Q_M \mathbf{x}_k + \mathbf{u}_k^T R_M \mathbf{u}_k \tag{8}$$

where $R_M \gg Q_M$. Let the ML state feedback be given as $\mathbf{u} = -K_M \mathbf{x}$, then the ML predictions are

$$\hat{\mathbf{x}}_k = \Phi_M^k \mathbf{x}_0; \quad \hat{\mathbf{u}}_k = -K_M \Phi_M^{k-1} \mathbf{x}_0, \quad \Phi_M = A - BK_M \tag{9}$$

Investigation into other ML solutions forms ongoing research, but we expect alternative costs like

$$J = \sum_{k=0}^{\infty} [\mathbf{x}_k - \mathbf{x}_a]^T Q_M [\mathbf{x}_k - \mathbf{x}_a] + [\mathbf{u}_k - \mathbf{u}_a]^T R_M [\mathbf{u}_k - \mathbf{u}_a] \tag{10}$$

where $\mathbf{x}_a$, $\mathbf{u}_a$ are steady-state values as far from constraints (4) as possible. It is also possible to find a prediction based on an infinity norm, for instance:

$$\left\| \begin{array}{c} \frac{\mathbf{U} - \mathbf{u}_a}{\overline{u} - \underline{u}} \\ \frac{\mathbf{Y} - \mathbf{y}_a}{\overline{y} - \underline{y}} \end{array} \right\|_\infty; \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \end{bmatrix}; \quad \mathbf{Y} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{y}_1 \\ \vdots \end{bmatrix} \tag{11}$$

However, this would add the on-line computational burden of an ∞-norm, though this is of course still far less than a QP.

Remark 2.1 *From here on the assumption is made that the ML predictions are always feasible, that is satisfy constraints. In general, it is straightforward to specify a set (e.g. [1])*

$$\mathbf{C}_a \mathbf{x}_0 - \mathbf{d}_a \tag{12}$$

such that this is true. It is then sufficient to ensure that the one step ahead prediction for $\mathbf{x}$ *lies in this set.*

2.3 Superposition of predictions

For a linear system such as (3), it is possible to use superposition of different stabilising input predictions and still retain a stable output prediction. This can be illustrated as follows. Let the initial state be

$$\mathbf{x}_0 = \mathbf{w}_0 + \mathbf{z}_0 \tag{13}$$

then the input predictions

$$\hat{\mathbf{u}}_k = -K_L \Phi_L^{k-1} \mathbf{w}_0 - K_M \Phi_M^{k-1} \mathbf{z}_0 \tag{14}$$

will give state predictions

$$\hat{\mathbf{x}}_k = \Phi_L^k \mathbf{w}_0 + \Phi_M^k \mathbf{z}_0 \tag{15}$$

which are clearly stable/convergent.

There are a number of ways of separating $\mathbf{x}$ into the sum of $\mathbf{y}$, $\mathbf{z}$; here we list some of the partitions that prove useful in predictive control.

$$\begin{array}{ll} \mathbf{z}_0 = \alpha \mathbf{x}_0 & \mathbf{w}_0 = (1-\alpha)\mathbf{x}_0 \\ \mathbf{z}_0 = \alpha(\mathbf{x}_0 - \Phi_L \mathbf{w}_{-1}) & \mathbf{w}_0 = (1-\alpha)\mathbf{x}_0 + \alpha \Phi_L \mathbf{w}_{-1}; \\ \mathbf{z}_0 = \mathbf{x}_0 - (1-\alpha)\Phi_L \mathbf{w}_{-1} & \mathbf{w}_0 = (1-\alpha)\Phi_L \mathbf{w}_{-1}; \end{array} \tag{16}$$

The top two of these also implicitly assume $0 \le \alpha \le 1$. Each choice affords different advantages. See section 6 for more detailed discussions.

- Choice (16a) allows for $\mathbf{w}_0 = 0$ and hence by assumption of remark 2.1 there exist a prediction set

(14,15) meeting constraints. However at sample instant $k = 1$ one can only recover the predictions used at sample instant k if $\alpha = 0$ or 1; this makes the proof of convergence difficult for other values of α.

- Choice (16b) includes the predictions used at the previous and allows for $\mathbf{z}_0 = 0$, i.e. the LQ predictions only. However, it does not include the possibility $\mathbf{w}_0 = 0$ and hence the degree of freedom α may be insufficient to guarantee feasibility in the presence of noise/uncertainty.
- Choice (16c) this includes the previous predictions in its class (see footnote) and also the choice $\mathbf{w}_0 = 0$ and hence would be robust to uncertainty/noise. However, the degree of freedom is limited to a direction which may not allow for fast convergence in some cases.

The following discussion will focus only on choices (16a, 16b) due to the potential performance limitations of choice (16c).

2.4 Constraints

Given input and state predictions (14, 15), it is elementary algebra to convert the system constraints (4) into the form

$$\alpha W\mathbf{x}_0 - \mathbf{d} \leq 0; \quad \mathbf{g} = W\mathbf{x}_0 \tag{17}$$

Technically W, $\mathbf{d}$ have an infinite number of rows but the work of [1] can be used to reduce this to a finite number which is often small, e.g. [5]. In general (given that the mix of ML and LQ can vary with time), a guarantee of future feasibility for $\alpha = 1$ can obtained if this constraint is appended to the constraint (12), e.g. $\mathbf{C}_a\hat{\mathbf{x}}_1 - \mathbf{d}_a \leq 0$. Let the total set of constraints be

$$\alpha\mathbf{g}_t - \mathbf{d}_t \leq 0; \tag{18}$$

3 A computationally efficient MPC Algorithm

We ensure that the ML predictions (14,15 with $\alpha = 1$) satisfy constraints. Hence the choice $\alpha = 1$ will always lead to constraint satisfaction and convergence, however the convergence rate would be very slow. Conversely, the choice $\alpha = 0$ gives predictions which move towards LQ optimal state feedback and would be selected if possible. The only question remaining then is how to select α in the cases where $\alpha = 0$ leads to infeasibility.

Theorem 3.1 *The optimal predicitive control policy is to minimise α subject to prediction equations (14, 15) satisfying (17). Of the predicted inputs implement the first and recompute the remainder at the next sampling instant.*

Hence a guarantee of stability for the nominal case

Proof: Assume that the performance one ideally wishes to minimise is (5) and substitute in the predictions of (14, 15). The cost becomes

$$J_\alpha = \mathbf{w}_0^T S_L \mathbf{w}_0 + \mathbf{z}_0^T S_M \mathbf{z}_0 + \mathbf{w}_0^T S_{LM}\mathbf{z}_0 \tag{19}$$

where Lyapunov equations can be used to compute S_L, S_M, S_{LM}. Expanding J as a quadratic in α gives

$$J_\alpha = a + b\alpha + c\alpha^2 \tag{20}$$

where a, b, c can be derived by subtituting from (16) into (19). However, it is known that in the absence of constraints, the optimal choice is $\alpha = 0$ which implies that $b = 0$ and hence

$$J_\alpha = a + c\alpha^2 \tag{21}$$

For example use of (16a) gives $a = \mathbf{x}_0^T S_L \mathbf{x}_0$; $c = \mathbf{x}_0^T[S_M - S_L]\mathbf{x}_0$. Minimising J_α is equivalent to minimising α. ∎

Although experimentation shows that use of predictions (16a) gives good performance, the proof of stability sometimes necessitates conservative choices for α as opposed to the minimising value. Hence, given that the stability of the algorithm implied by Theorem 3.1 with predictions (16b) is immediate (it is well known that if the previous predictions can be utilised again, then J can be shown to be monotonic decreasing), these will be preferred. However, as this choice alone is not robust to model uncertainty, the following algorithm could be proposed.

Algorithm 3.1 *EMPC (Efficient MPC) Using predictions (16b) minimise α subject to (18) and $0 \leq \alpha \leq 1$. If there is no solution, use predictions (16a) and again minimise α subject to the corresponding (18).*

4 Example

In this section, we illustrate the efficiency of the EMPC algorithm using either (16a) or (16b) through a simulation example. We evaluate the performance using J_{run}, given by

$$J_{run} = \sum_{i=0}^{runtime} (\mathbf{y}_i - \mathbf{r}_i)^2 + \lambda\mathbf{u}_i^2 \tag{22}$$

with $\mathbf{r}_i$ the setpoint value at the ith sampling instant.

The model is given by

$$A = \begin{bmatrix} -0.23 & 0.95 & -0.16 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix};$$
$$C = [1 \ -0.08 \ -0.001 \ 0] \tag{23}$$

Furthermore, $Q = R = 1$ and $\underline{u} = -0.15 = -\overline{u}$. As usual in MPC, an integrator has been included in the above model.

The EMPC algortihm is compared to the CLQR algorithm of [6] by way of J_{run} for a closed-loop response to a unit set point change at the 10th sampling instant and computational load at a single sample instant. A value $n_c = 4$ was required for CLQR to avoid infeasibility and in fact this also gave very close to the constrained optimal for $n_c \to \infty$. The runtime costs are given in table 1 and the simulation plots for CLQR and EMPC (For 16b) are in Figures 1,2 respectively. There is a negligible difference between the two algorithms. Figure 2 also shows the value of α (denoted by 'alpha'). At the set-point change, the LQ unconstrained control actions are infeasible and therefore, the algorithm increases α. At the 15th sample instant, the unconstrained LQ control actions are feasible and α has converged back to zero. It is noted that α does not always decrease monotonically. For completeness the tables give the results for predictions (16a) and (16b).

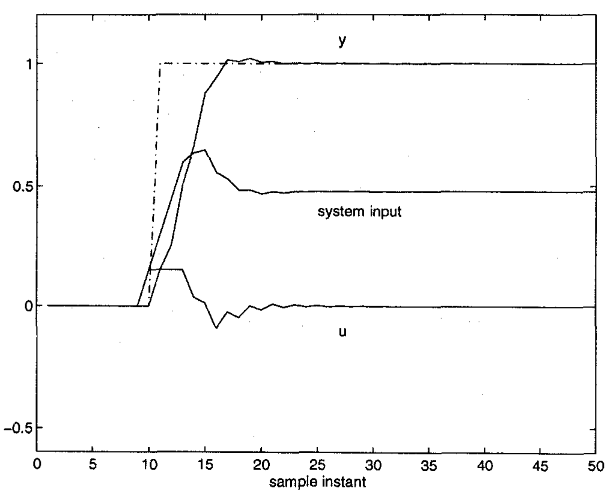

Figure 1. Step response with CLQR.

CLQR $n_c = 4$	EMPC (16a)	EMPC (16b)
1.7553	1.7553	1.7588

Table 1. Runtime costs

The computational load comparison at the 11th sampling instant in terms of flops (measured on MATLAB where no particular care has been taken to programming efficiency) between the QP used by CLQR and the LP used by EMPC is shown in Figure 2. It is clear that EMPC is far quicker.

CLQR $n_c = 4$	EMPC (16a)	EMPC (16b)
17930	497	1067

Table 2. Computational load flops at 11th sample

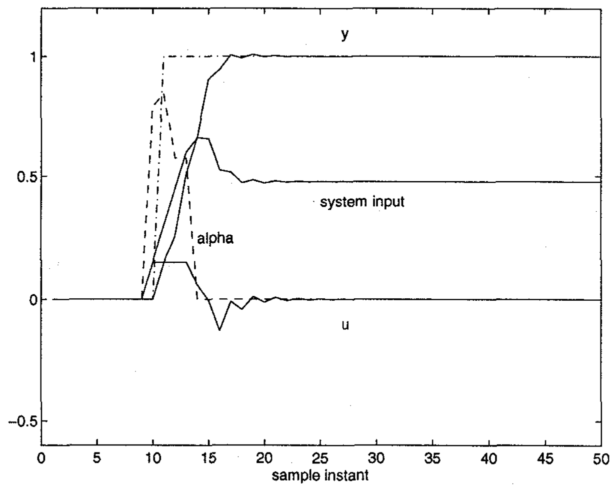

Figure 2. Step response with EMPC.

5 Conclusions

A predictive control algorithm was presented in which the control actions are based on a weighted sum of LQ predictions and ML predictions. The algorithm achieves LQ optimality when no constraints are active. In the presence of constraints, it remains feasible (under mild conditions). Since the algorithm requires the solution of a simple Linear Program problem with only one degree of freedom, it is computationally efficient. Despite this great improvement in computational demand, the results of a simulation study indicate that the performance of the control algorithm is only marginally below the theoretical optimum (that achievable for an infinite number of degrees of freedom) when constraints are active.

References

[1] E.G. Gilbert and K. T. Tan, Linear systems with state and control constriants: the theory and application of maximal output admissable sets, 1991, IEEE Trans AC, 36, 9, pp1008-1020

[2] B. Kouvaritakis, J.A. Rossiter, M. Cannon and C.K.J. Wong, Constrained stable predictive control: an l_2 optimal approach, 5th IEEE Mediterranean Conference on Control and Systems, Cyprus, 1997

[3] B. Kouvaritakis, J.A. Rossiter and M. Cannon, Linear quadratic feasible predictive control, submitted.

[4] B. Kouvaritakis, J.A. Rossiter and M.J.Rice, Predictive LQ control with constraints, to appear CDC 97

[5] J.A. Rossiter, M.J.Rice and B. Kouvaritakis, Handling hard constraints in infinite horizon predictive control, ECC97

[6] P.O.M.Scokaert and J. B. Rawlings, Infinite horizon linear quadratic control with constraints, Proceedings IFAC96, San Fransisco, 1996, Vol. M, pp109-114

6 Appendix A: Proofs of stability/convergence

It is normal to assume that the use of infinite horizons guarantees nominal stability, even in the presence of constraints, because (in the absence of uncertainty) the cost function J can be shown to be a Lyapunov function. However, this is true only because it makes the assumption that the optimum input trajectory at sample instant $t = 0$, say

$$\hat{\mathbf{u}} = [\mathbf{u}_{0|0}, \mathbf{u}_{1|0}, \mathbf{u}_{2|0}, ...]$$

can still be utilised at the next sample instant, not including the first element $\mathbf{u}_{0|0}$ which has been used, that is at $t = 1$ the input trajectory

$$\hat{\mathbf{u}} = [\mathbf{u}_{1|0}, \mathbf{u}_{2|0}, ...]$$

is stabilising and leads to a smaller cost, that is $J_1 < J_0$, because $J_1 \leq J_0 - \mathbf{x}_0^T Q \mathbf{x}_0 - \mathbf{u}_0^T R \mathbf{u}_0$. The extra d.o.f introduced at $t = 1$ is then used to decrease J_1 further still.

The proposed EMPC algorithm with predictions(16b) has the above property in the nominal case, that is in the abscence of noise/disturbances and uncertainty. This is easy to show; for brevity we give state prediction equations only. At sample instant $t = -1$, the predictions utilised were

$$\hat{\mathbf{x}}_k = \Phi_L^k \mathbf{w}_{-1} + \Phi_M^k \mathbf{z}_{-1} \tag{24}$$

At sample instant $t = 0$, substitution of (16b) gives

$$\hat{\mathbf{x}}_k = \Phi_L^{k-1}[\alpha\Phi_L \mathbf{w}_{-1} + (1-\alpha)\mathbf{x}_0] + \Phi_M^{k-1}\alpha[\mathbf{x}_0 - \Phi_L \mathbf{w}_{-1}) \tag{25}$$

But, in the nominal case $\mathbf{x}_0 - \Phi_L \mathbf{w}_{-1} = \Phi_M \mathbf{z}_{-1}$ and hence the choice $\alpha = 1$ makes eqns. (24,25) identical.

The full algorithm allowed for a switch to predictions (16a) in the event that uncertainty caused temporary infeasibility with predictions (16b). At the next sample one could return to the predictions (16b) with the new $\mathbf{w}$, $\mathbf{z}$ and hence achieve convergence and stability.

6.1 Prediction equations (16a)

The disadvantage of using predictions (16b) is a higher likelihood of infeasibility in the event of bounded uncertainty/disturbances because $\mathbf{w} = 0$ is not an option in general. To 'guarantee' feasibility one would have to use prediction equations (16a) (or 16c), however then the stability proof is not straightforward because the prediction class at sample instant $t = 1$ does not in general include the optimum predictions from $t = 0$ (unless $\alpha = 0, 1$). For example, at $t = 0, 1$ the predictions are

$$\begin{aligned} \mathbf{x}_{j|0} &= (1-\alpha_0)\Phi_L^j \mathbf{x}_0 + \alpha_0 \Phi_M^j \mathbf{x}_0 \\ \mathbf{x}_{j|1} &= (1-\alpha_1)\Phi_L^{j-1}\mathbf{x}_{1|0} + \alpha_1 \Phi_M^{j-1}\mathbf{x}_{1|0} \end{aligned} \tag{26}$$

Hence it is easy to see that $\mathbf{x}_{j|1} = \mathbf{x}_{j|0}$ iff

$$[\Phi_L^{j-1} - \Phi_M^{j-1}] Z \mathbf{x}_0 = 0; \;\; Z = (\alpha_0 - 1)\alpha_1 \Phi_L + (1-\alpha_1)\alpha_0 \Phi_M \tag{27}$$

This strange anomaly arises because the predicted input is the mix of two control laws; the interaction of the two laws with the new initial condition at sample instant $k+1$ gives predictions with no obvious correspondence to those computed at sample instant k.

Here, an alternative proof of stability works by establishing an upper bound on the cost-to-go J_i where

$$\overline{J}_0 = \mathbf{x}_0 S_M \mathbf{x}_0; \;\; S_M = \sum_{i=0}^{\infty} [\Phi_M^T]^i [Q_M + K_M^T R_M K_M] \Phi_M^i \tag{28}$$

is the predicted cost (5) using ML predictions. We can force this upper bound to be monotonically decreasing. At sample instant $t = 1$, the actual value of $\mathbf{x}$ is given from (26) so that J_1 is bounded by

$$\begin{aligned} \overline{J}_1 = & \;\mathbf{x}_1 S_M \mathbf{x}_1 \\ = & \;\mathbf{x}_0[(1-\alpha_0)^2 \Phi_L^T S_M \Phi_L + \alpha_0^2 \Phi_M^T S_M \Phi_M \\ & + \alpha_0(1-\alpha_0)\Phi_L^T S_M \Phi_M + \alpha_0(1-\alpha_0)\Phi_M^T S_M \Phi_L]\mathbf{x}_0 \\ = & \;\mathbf{x}_0^T S_\alpha \mathbf{x}_0 \end{aligned} \tag{29}$$

To guarantee convergence it is sufficient to establish that

$$\mathbf{x}_0^T S_M \mathbf{x}_0 > \mathbf{x}_0 S_\alpha \mathbf{x}_0, \quad \forall \mathbf{x}_0, \; \alpha \tag{30}$$

It is obviously true for $\alpha = 1$ as $S_\alpha(1) = S_M - Q_M - K_2^T R_M K_2$. To establish the result for a range of α we look for points where the ellipsoids (in $\mathbf{x}$-space) for $\overline{J}_0$ and $\overline{J}_1$ are tangential. For a given α, $0 < \alpha < 1$ and $\|\mathbf{x}_o\|_2 = c$, c a constant, if $\overline{J}_1 < \overline{J}_0$ at all points $\mathbf{x}_0$ where the tangents are equal, then $\overline{J}_1 < \overline{J}_0$, $\forall \mathbf{x}_0$. The points of tangency can be solved from the generalised eigenvalue/vector problem

$$S_\alpha \mathbf{x} = \lambda S_M \mathbf{x} \tag{31}$$

as the eigenvectors. Each eigenvector $\mathbf{x}_i$ in turn can be substituted for $\mathbf{x}_0$ to compute $H_i(\alpha) = \overline{J}_0/\overline{J}_1$ If $H_i(\alpha) < 1$ for any $i = 1, ..., n$, $0 \leq \alpha < 1$ then there exists an initial condition such that $\overline{J}_1 > \overline{J}_0$.

Remark 6.1 *This test is performed offline. In general there is at least one direction such that $H_i(\alpha) < 1$ for $\alpha \ll 1$. Hence, one could only ensure that the cost-to-go was monotonic by first computing optimum α and then computing*

$$G = \mathbf{x}_0 S_M \mathbf{x}_0 - \mathbf{x}_0 S_\alpha \mathbf{x}_0 \tag{32}$$

If $G > 0$, use α, if $G < 0$ use $\alpha = 1$.

Remark 6.2 *Even without this test, we have yet to find an example where use of (16a) failed to give excellent performance which suggests a tighter stability proof exists.*