Chapter 4
Interpolating Control—Nominal State Feedback Case

4.1 Problem Formulation

Consider the problem of regulating to the origin the following time-invariant linear discrete-time system,

$$x(k+1) = Ax(k) + Bu(k) \tag{4.1}$$

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ are respectively, the measurable state vector and the input vector. The matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. Both $x(k)$ and $u(k)$ are subject to bounded polytopic constraints,

$$\begin{cases} x(k) \in X, \ X = \left\{x \in \mathbb{R}^n : F_x x \le g_x\right\} \\ u(k) \in U, \ U = \left\{u \in \mathbb{R}^m : F_u u \le g_u\right\} \end{cases} \quad \forall k \ge 0 \tag{4.2}$$

where the matrices F_x, F_u and the vectors g_x, g_u are assumed to be constant. The inequalities are taken element-wise. It is assumed that the pair (A, B) is stabilizable, i.e. all uncontrollable states have stable dynamics.

4.2 Interpolating Control via Linear Programming—Implicit Solution

Define a linear controller $K \in \mathbb{R}^{m\times n}$, such that,

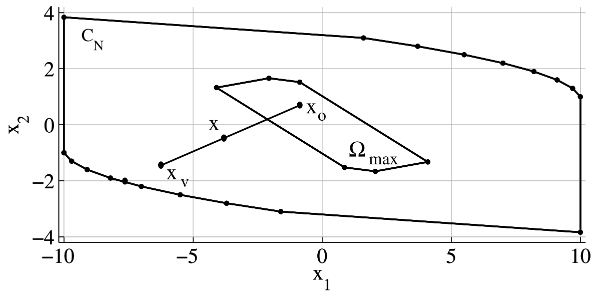
$$u(k) = Kx(k) \tag{4.3}$$

asymptotically stabilizes the system (4.1) with some desired performance specifications. The details of such a synthesis procedure are not reproduced here, but we assume that feasibility is guaranteed. For the controller (4.3) using Procedure 2.1 or Procedure 2.2 the maximal invariant set $\Omega_{\max}$ can be computed as,

$$\Omega_{\max} = \left\{x \in \mathbb{R}^n : F_o x \le g_o\right\} \tag{4.4}$$

H.-N. Nguyen, *Constrained Control of Uncertain, Time-Varying, Discrete-Time Systems*, Lecture Notes in Control and Information Sciences 451,
DOI 10.1007/978-3-319-02827-9_4,

Fig. 4.1 Any state $x(k)$ can be decomposed as a convex combination of $x_v(k) \in C_N$ and $x_o(k) \in \Omega_{\max}$



Furthermore with some given and fixed integer $N > 0$, based on Procedure 2.3 the controlled invariant set C_N can be found as,

$$C_N = \{x \in \mathbb{R}^n : F_N x \le g_N\} \tag{4.5}$$

such that all $x \in C_N$ can be steered into $\Omega_{\max}$ in no more than N steps when a suitable control is applied. As in Sect. 3.4, the set C_N is decomposed as a sequence of simplices $C_N^{(j)}$, each formed by n vertices of C_N and the origin. For all $x(k) \in C_N^{(j)}$, the vertex controller

$$u(k) = K^{(j)} x(k), \tag{4.6}$$

with $K^{(j)}$ given in (3.38) asymptotically stabilizes the system (4.1), while the constraints (4.2) are fulfilled.

The main advantage of the vertex control scheme is the size of the domain of attraction, i.e. the set C_N. Clearly, C_N, that is the feasible domain for vertex control, might be as large as that of any other constrained control scheme. However, a weakness of vertex control is that the full control range is exploited only on the boundary of C_N in the state space, with progressively smaller control action when state approaches the origin. Hence the time to regulate the plant to the origin is often unnecessary long. A way to overcome this shortcoming is to switch to another, more aggressive, local controller, e.g. the controller (4.3), when the state reaches $\Omega_{\max}$. The disadvantage of this solution is that the control action becomes *nonsmooth* [94].

Here a method to overcome the nonsmooth control action [94] will be proposed. For this purpose, any state $x(k) \in C_N$ is decomposed as,

$$x(k) = c(k)x_v(k) + \big(1 - c(k)\big)x_o(k) \tag{4.7}$$

with $x_v \in C_N$, $x_o \in \Omega_{\max}$ and $0 \le c \le 1$. Figure 4.1 illustrates such a decomposition.

Consider the following control law,

$$u(k) = c(k)u_v(k) + \big(1 - c(k)\big)u_o(k) \tag{4.8}$$

where $u_v(k)$ is the vertex control law (4.6) at $x_v(k)$ and $u_o(k) = Kx_o(k)$ is the control law (4.3) in $\Omega_{\max}$.

Theorem 4.1 *For system* (4.1) *and constraints* (4.2), *the control law* (4.7), (4.8) *guarantees recursive feasibility for all initial states* $x(0) \in C_N$.

Proof For recursive feasibility, we have to prove that,

$$\begin{cases} F_u u(k) \le g_u \\ x(k+1) = Ax(k) + Bu(k) \in C_N \end{cases}$$

for all $x(k) \in C_N$. For the input constraints,

$$\begin{aligned} F_u u(k) &= F_u \big\{ c(k) u_v(k) + \big(1 - c(k)\big) u_o(k) \big\} \\ &= c(k) F_u u_v(k) + \big(1 - c(k)\big) F_u u_o(k) \\ &\le c(k) g_u + \big(1 - c(k)\big) g_u = g_u \end{aligned}$$

and for the state constraints,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= A\big\{ c(k) x_v(k) + \big(1 - c(k)\big) x_o(k) \big\} + B\big\{ c(k) u_v(k) + \big(1 - c(k)\big) u_o(k) \big\} \\ &= c(k) \big\{ Ax_v(k) + Bu_v(k) \big\} + \big(1 - c(k)\big) \big\{ Ax_o(k) + Bu_o(k) \big\} \end{aligned}$$

Since $Ax_v(k) + Bu_v(k) \in C_N$ and $Ax_o(k) + Bu_o(k) \in \Omega_{\max} \subseteq C_N$, it follows that $x(k+1) \in C_N$. □

Since the controller (4.3) is designed to give specified unconstrained performance in $\Omega_{\max}$, it might be desirable to have $u(k)$ in (4.8) as close as possible to it also outside $\Omega_{\max}$. This can be achieved by minimizing c,

$$c^* = \min_{x_v, x_o, c} \{c\} \tag{4.9}$$

subject to

$$\begin{cases} F_N x_v \le g_N, \\ F_o x_o \le g_o, \\ c x_v + (1 - c) x_o = x, \\ 0 \le c \le 1 \end{cases}$$

Denote $r_v = c x_v \in \mathbb{R}^n$, $r_o = (1 - c) x_o \in \mathbb{R}^n$. Since $x_v \in C_N$ and $x_o \in \Omega_{\max}$, it follows that $r_v \in c C_N$ and $r_o \in (1 - c)\Omega_{\max}$ or equivalently

$$\begin{cases} F_N r_v \le c g_N \\ F_o r_o \le (1 - c) g_o \end{cases}$$

Hence the nonlinear optimization problem (4.9) is transformed into the following linear programming problem,

$$c^* = \min_{r_v, c}\{c\} \tag{4.10}$$

subject to

$$\begin{cases} F_N r_v \le c g_N, \\ F_o(x - r_v) \le (1 - c) g_o, \\ 0 \le c \le 1 \end{cases}$$

Remark 4.1 If one would like to maximize c, it is obvious that $c = 1$ for all $x \in C_N$. In this case the controller (4.7), (4.8) becomes the vertex controller.

Theorem 4.2 *The control law* (4.7), (4.8), (4.10) *guarantees asymptotic stability for all initial states* $x(0) \in C_N$.

Proof First of all we will prove that all solutions starting in $C_N \setminus \Omega_{\max}$ will reach $\Omega_{\max}$ in *finite time*. For this purpose, consider the following non-negative function,

$$V(x) = c^*(x), \quad \forall x \in C_N \setminus \Omega_{\max} \tag{4.11}$$

$V(x)$ is a candidate Lyapunov function. After solving the LP problem (4.10) and applying (4.7), (4.8), one obtains, for $x(k) \in C_N \setminus \Omega_{\max}$,

$$\begin{cases} x(k) = c^*(k) x_v^*(k) + \big(1 - c^*(k)\big) x_o^*(k) \\ u(k) = c^*(k) u_v(k) + \big(1 - c^*(k)\big) u_o(k) \end{cases}$$

It follows that,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= c^*(k) x_v(k+1) + \big(1 - c^*(k)\big) x_o(k+1) \end{aligned}$$

where

$$\begin{cases} x_v(k+1) = Ax_v^*(k) + Bu_v(k) \in C_N \\ x_o(k+1) = Ax_o^*(k) + Bu_o(k) \in \Omega_{\max} \end{cases}$$

Hence $c^*(k)$ is a feasible solution for the LP problem (4.10) at time $k+1$. By solving (4.10) at time $k+1$, one gets the optimal solution, namely

$$x(k+1) = c^*(k+1) x_v^*(k+1) + \big(1 - c^*(k+1)\big) x_o^*(k+1)$$

where $x_v^*(k+1) \in C_N$ and $x_o^*(k+1) \in \Omega_{\max}$. It follows that $c^*(k+1) \le c^*(k)$ and $V(x)$ is non-increasing.

Using the vertex controller, an interpolation between a point of C_N and the origin is obtained. Conversely using the controller (4.7), (4.8), (4.10) an interpolation is constructed between a point of C_N and a point of $\Omega_{\max}$ which in turn contains the

Algorithm 4.1 Interpolating control—Implicit solution

1. Measure the current state $x(k)$.
2. Solve the LP problem (4.10).
3. Compute u_{rv} in (4.12) by determining to which simplex r_v^* belongs and using (3.38).
4. Implement as input the control signal (4.12).
5. Wait for the next time instant $k := k + 1$.
6. Go to step 1 and repeat.

origin as an interior point. This last property proves that the vertex controller is a feasible choice for the interpolation scheme (4.7), (4.8), (4.10). Hence it follows that,

$$c^*(k) \le \sum_{i=1}^{s} \beta_i^*(k)$$

for any $x(k) \in C_N$, with $\beta_i^*(k)$ obtained in (3.46), Sect. 3.4.

Since the vertex controller is asymptotically stabilizing, the state reaches any bounded set around the origin in finite time. In our case this property will imply that using the controller (4.7), (4.8), (4.10) the state of the closed loop system reaches $\Omega_{\max}$ in *finite time* or equivalently that there exists a finite k such that $c^*(k) = 0$.

The proof is complete by noting that inside $\Omega_{\max}$, the LP problem (4.10) has the trivial solution $c^* = 0$. Hence the controller (4.7), (4.8), (4.10) becomes the local controller (4.3). The feasible stabilizing controller $u(k) = Kx(k)$ is *contractive*, and thus the interpolating controller assures asymptotic stability for all $x \in C_N$. □

The control law (4.7), (4.8), (4.10) obtained by solving on-line the LP problem (4.10) is called Implicit Interpolating Control.

Since $r_v^*(k) = c^*(k)x_v^*(k)$ and $r_o^*(k) = (1 - c^*(k))x_o^*(k)$, it follows that,

$$u(k) = u_{rv}(k) + u_{ro}(k) \tag{4.12}$$

where $u_{rv}(k)$ is the vertex control law at $r_v^*(k)$ and $u_{ro}(k) = Kr_o^*(k)$.

Remark 4.2 Note that at each time instant Algorithm 4.1 requires the solutions of two LP problems, one is (4.10) of dimension $n + 1$, the other is to determine to which simplex r_v^* belongs.

Example 4.1 Consider the following time-invariant linear discrete-time system,

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.3 \end{bmatrix} u(k) \tag{4.13}$$

The constraints are,

$$-10 \le x_1(k) \le 10, \qquad -5 \le x_2(k) \le 5, \qquad -1 \le u(k) \le 1 \tag{4.14}$$

The local controller is chosen as a linear quadratic (LQ) controller with weighting matrices $Q = I$ and $R = 1$, giving the state feedback gain,

$$K = [-0.5609 \quad -0.9758] \tag{4.15}$$

The sets Ω_{max} and C_N with $N = 14$ are shown in Fig. 4.1. Note that $C_{14} = C_{15}$ is the maximal controlled invariant set. Ω_{max} is presented in minimal normalized half-space representation as,

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.1627 & -0.9867 \\ -0.1627 & 0.9867 \\ -0.1159 & -0.9933 \\ 0.1159 & 0.9933 \\ -0.4983 & -0.8670 \\ 0.4983 & 0.8670 \end{bmatrix} x \le \begin{bmatrix} 1.9746 \\ 1.9746 \\ 1.4115 \\ 1.4115 \\ 0.8884 \\ 0.8884 \end{bmatrix} \right\} \tag{4.16}$$

The set of vertices of C_N is given by the matrix $V(C_N)$, together with the corresponding control matrix U_v,

$$V(C_N) = [V_1 \quad -V_1], \qquad U_v = [U_1 \quad -U_1] \tag{4.17}$$

where

$$V_1 = \begin{bmatrix} 10.0000 & 9.7000 & 9.1000 & 8.2000 & 7.0000 & 5.5000 & 3.7000 & 1.6027 & -10.0000 \\ 1.0000 & 1.3000 & 1.6000 & 1.9000 & 2.2000 & 2.5000 & 2.8000 & 3.0996 & 3.8368 \end{bmatrix},$$

$$U_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}$$

The state space partition of vertex control is shown in Fig. 4.2(a). Using the implicit interpolating controller, Fig. 4.2(b) presents state trajectories of the closed loop system for different initial conditions.

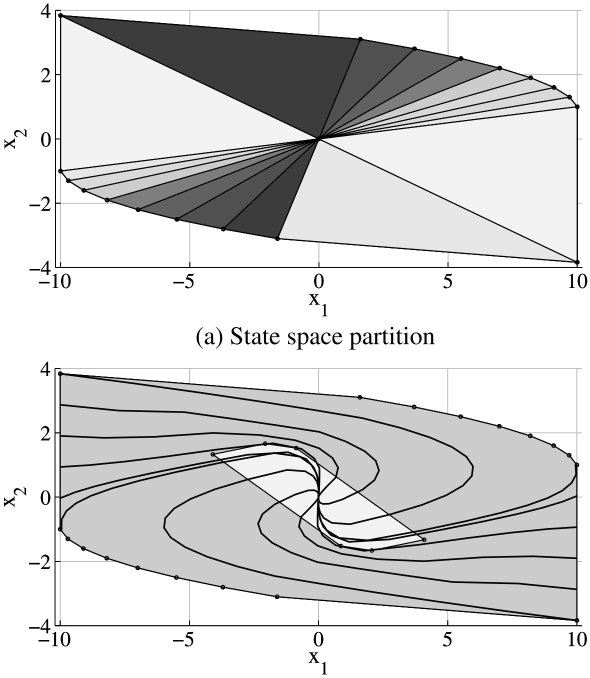
For the initial condition $x(0) = [-2.0000 \; 3.3284]^T$, Fig. 4.3 shows the state and input trajectories for the implicit interpolating controller (solid). As a comparison, we take MPC, based on quadratic programming, where an LQ criterion is optimized, with identity weighting matrices. Hence the set Ω_{max} for the local unconstrained control is identical for the MPC solution and for the implicit interpolating controller. The prediction horizon for the MPC was chosen to be 14 to match the controlled invariant set C_{14} used for the implicit interpolating controller. Figure 4.3 shows the state and input trajectories obtained for the implicit MPC (dashed).

Using the tic/toc function of Matlab 2011b, the computational burdens of interpolating control and MPC were compared. The result is shown in Table 4.1

Table 4.1 Durations [ms] of the on-line computations during one sampling interval for interpolating control and MPC, respectively for Example 4.1

	Computational time
Implicit interpolating control	0.7652
Implicit QP-MPC	4.6743

Fig. 4.2 State space partition of vertex control and state trajectories for Example 4.1



(a) State space partition

(b) State trajectories

As a final analysis element, Fig. 4.4 presents the interpolating coefficient $c^*(k)$. It is interesting to note that $c^*(k) = 0$, $\forall k \geq 15$ indicating that from time instant $k = 15$, the state of the closed loop system is in $\Omega_{\max}$, and consequently is *optimal* in the MPC cost function terms. The monotonic decrease and the positivity confirms the Lyapunov interpretation given in the present section.

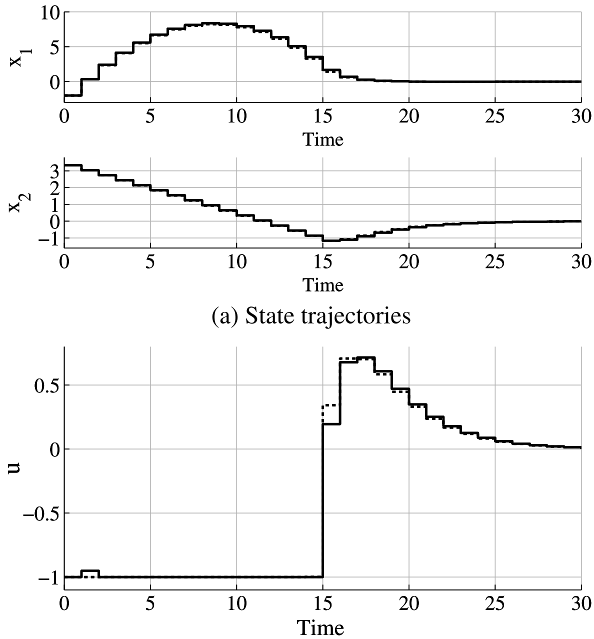
4.3 Interpolating Control via Linear Programming—Explicit Solution

The structural implication of the LP problem (4.10) is investigated in this section.

4.3.1 Geometrical Interpretation

Let $\partial(\cdot)$ denotes the boundary of the corresponding set $(\cdot)$. The following theorem holds

Theorem 4.3 *For all* $x \in C_N \setminus \Omega_{\max}$, *the solution of the LP problem* (4.10) *satisfies* $x_v^* \in \partial C_N$ *and* $x_o^* \in \partial \Omega_{\max}$.



(a) State trajectories

(b) Input trajectories

Fig. 4.3 State and input trajectories for Example 4.1 for implicit interpolating control (*solid*), and for implicit QP-MPC (*dashed*)

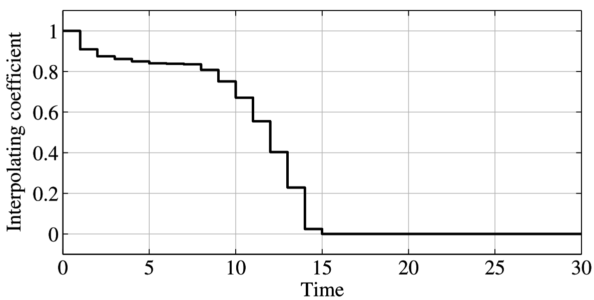


Fig. 4.4 Interpolating coefficient c^* as a function of time example 4.1

Proof Consider $x \in C_N \setminus \Omega_{\max}$, with a particular convex combination

$$x = cx_v + (1 - c)x_o$$

where $x_v \in C_N$ and $x_o \in \Omega_{\max}$. If x_o is strictly inside $\Omega_{\max}$, one can set $\tilde{x}_o = \partial\Omega_{\max} \cap \overline{x, x_o}$, i.e. $\tilde{x}_o$ is the intersection between $\partial\Omega_{\max}$ and the line segment connecting x and x_o, see Fig. 4.5. Apparently, x can be expressed as the convex combination of x_v and $\tilde{x}_o$, i.e.

$$x = \tilde{c}x_v + (1 - \tilde{c})\tilde{x}_o$$

with $\tilde{c} < c$, since x is closer to $\tilde{x}_o$ than to x_o. So (4.10) leads to $\{c^*, x_v^*, x_o^*\}$ with $x_o^* \in \partial\Omega_{\max}$.

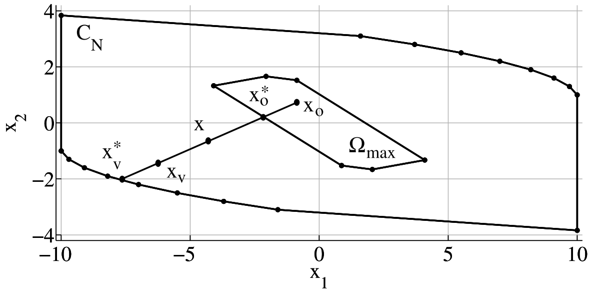


Fig. 4.5 Graphical illustration for the proof of Theorem 4.3

One the other hand, if x_v is strictly inside C_N, one can set $\tilde{x}_v = \partial C_N \cap \overrightarrow{x, x_v}$, i.e. $\tilde{x}_v$ is the intersection between ∂C_N and the ray starting from x through x_v, see Fig. 4.5. Again, x can be written as the convex combination of $\tilde{x}_v$ and x_o, i.e.

$$x = \tilde{c}\tilde{x}_v + (1 - \tilde{c})x_o$$

with $\tilde{c} < c$, since x is further from $\tilde{x}_v$ than from x_v. This leads to the conclusion that for the optimal solution $\{c^*, x_v^*, x_o^*\}$ we have $x_v^* \in \partial P_N$. □

Theorem 4.3 states that for all $x \in C_N \setminus \Omega_{\max}$, the interpolating coefficient c is minimal if and only if x is written as a convex combination of two points, one belonging to C_N and the other to $\partial\Omega_{\max}$. It is obvious that for $x \in \Omega_{\max}$, the LP problem (4.10) has the trivial solution $c^* = 0$ and thus $x_v^* = 0$ and $x_o^* = x$.

Theorem 4.4 *For all $x \in C_N \setminus \Omega_{\max}$, the convex combination $x = cx_v + (1 - c)x_o$ gives the smallest value of c if the ratio $\frac{\|x_v - x\|}{\|x - x_o\|}$ is maximal, where $\|\cdot\|$ denotes the Euclidean vector norm.*

Proof It holds that

$$\begin{aligned} & x = cx_v + (1 - c)x_o \\ \Rightarrow \quad & x_v - x = x_v - cx_v - (1 - c)x_o = (1 - c)(x_v - x_o) \end{aligned}$$

consequently

$$\|x_v - x\| = (1 - c)\|x_v - x_o\| \tag{4.18}$$

Analogously, one obtains

$$\|x - x_o\| = c\|x_v - x_o\| \tag{4.19}$$

Combining (4.18) and (4.19) and the fact that $c \neq 0$ for all $x \in C_N \setminus \Omega_{\max}$, one gets

$$\frac{\|x_v - x\|}{\|x - x_o\|} = \frac{(1 - c)\|x_v - x_o\|}{c\|x_v - x_o\|} = \frac{1}{c} - 1$$

$c > 0$ is minimal if and only if $\frac{1}{c} - 1$ is maximal, or equivalently $\frac{\|x_v - x\|}{\|x - x_o\|}$ is maximal. □

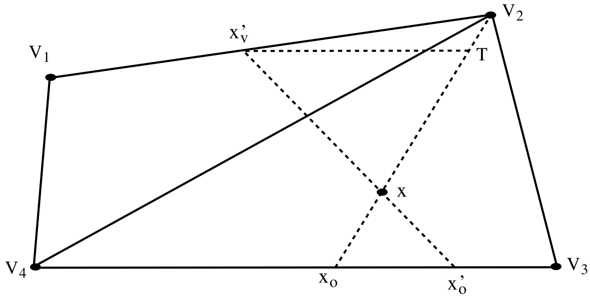


Fig. 4.6 Graphical illustration for the proof of Theorem 4.5

4.3.2 Analysis in $\mathbb{R}^2$

In this subsection an analysis of the optimization problem (4.9) in the $\mathbb{R}^2$ parameter space is presented with reference to Fig. 4.6. The discussion is insightful in what concerns the properties of the partition in the explicit solution. The problem considered here is to decompose the polyhedral X_{1234} such that the explicit solution $c^* = \min\{c\}$ is given in the decomposed cells.

For illustration we will consider four points V_1, V_2, V_3, V_4, and any point $x \in \text{Conv}(V_1, V_2, V_3, V_4)$. This schematic view can be generalized to any pair of faces of C_N and $\Omega_{\max}$. Denote V_{ij} as the interval connecting V_i and V_j for $i, j = 1, \ldots, 4$. The problem is reduced to the expression of a convex combination $x = cx_v + (1-c)x_o$, where $x_v \in V_{12} \subset \partial C_N$ and $x_o \in V_{34} \subset \partial \Omega_{\max}$ providing the minimal value of c.

Without loss of generality, suppose that the distance from V_2 to V_{34} is greater than the distance from V_1 to V_{34}, or equivalently the distance from V_4 to V_{12} is smaller than the distance from V_3 to V_{12}.

Theorem 4.5 *Under the condition that the distance from V_2 to V_{34} is greater than the distance from V_1 to V_{34}, or equivalently the distance from V_4 to V_{12} is smaller than the distance from V_3 to V_{12}, the decomposition of the polytope V_{1234}, $V_{1234} = V_{124} \cup V_{234}$ is the result of the minimization of the interpolating coefficient c.*

Proof Without loss of generality, suppose that $x \in V_{234}$. x can be decomposed as,

$$x = cV_2 + (1-c)x_o \tag{4.20}$$

where $x_o \in V_{34}$, see Fig. 4.6. Another possible decomposition is

$$x = c'x'_v + \left(1-c'\right)x'_o \tag{4.21}$$

where x'_v belongs to V_{34} and x'_o belongs to V_{12}.

Clearly, if the distance from V_2 to V_{34} is greater than the distance from V_1 to V_{34} then the distance from V_2 to V_{34} is greater than the distance from any point in V_{12} to V_{34}. Consequently, there exists the point T in the ray, starting from V_2 through

x such that the distance from T to V_{34} is equal to the distance from x'_v to V_{34}. It follows that the line connecting T and x'_v is parallel to X_{34}, see Fig. 4.6.

Using Basic Proportionality Theorem, one has

$$\frac{\|x - x'_v\|}{\|x - x'_o\|} = \frac{\|x - T\|}{\|x - x_o\|} \tag{4.22}$$

by using Theorem 4.4 and since

$$\frac{\|x - T\|}{\|x - x_o\|} < \frac{\|x - V_2\|}{\|x - x_o\|}$$

it follows that $c < c'$. □

Theorem 4.5 states that the minimal value of the interpolating coefficient c is found with the help of the decomposition of V_{1234} as $V_{1234} = V_{124} \cup V_{234}$.

Remark 4.3 Clearly, if V_{12} is parallel to V_{34}, then any convex combination $x = cx_v + (1 - c)x_o$ gives the same value of c. Hence the partition may not be unique.

Remark 4.4 As a consequence of Theorem 4.5, it is clear that the region $C_N \setminus \Omega_{\max}$ can be subdivided into partitions (cells) as follows,

- For each facet of the set $\Omega_{\max}$, one has to find the furthest point on ∂C_N on the same side of the origin as the facet of $\Omega_{\max}$. A polyhedral cell is obtained as the convex hull of that facet of $\Omega_{\max}$ and the furthest point in C_N. By the bounded polyhedral structure of C_N, the existence of some vertex of C_N as the furthest point is guaranteed.
- On the other hand, for each facet of C_N, one has to find the closest point on $\partial \Omega_{\max}$ on the same side of the origin as the facet of C_N. A polyhedral cell is obtained as the convex hull of that facet of C_N and the closest point in $\Omega_{\max}$. Again by the bounded polyhedral structure of $\Omega_{\max}$, the existence of some vertex $\Omega_{\max}$ as the closest point is guaranteed.

Remark 4.5 Clearly, in $\mathbb{R}^2$, the state space partition according to Remark 4.4 cover the entire set C_N, see e.g. Fig. 4.7. However in $\mathbb{R}^n$, that is not necessarily the case as shown in the following example. Let C_N and $\Omega_{\max}$ be given by the vertex representations, displayed in Fig. 4.8(a),

$$C_N = \text{Conv}\left\{\begin{bmatrix} -4 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ -4 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \\ -4 \end{bmatrix}\right\}$$

$$\Omega_{\max} = \text{Conv}\left\{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \end{bmatrix}, \begin{bmatrix} -0.5 \\ 0.5 \\ 0 \end{bmatrix}, \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix}\right\}$$

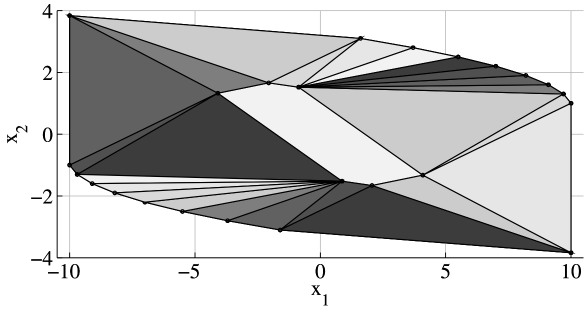
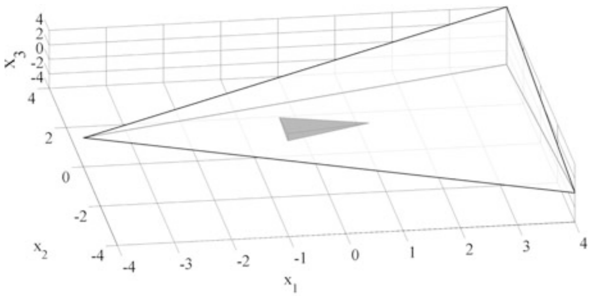
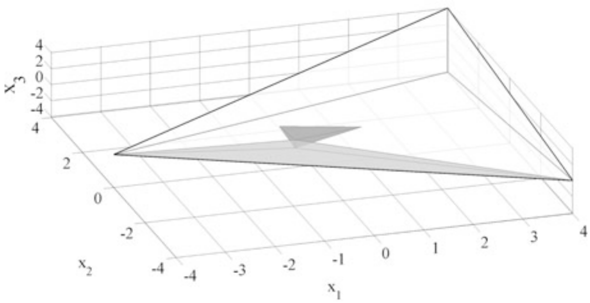


Fig. 4.7 Simplex based decomposition as an explicit solution of the LP problem (4.10)



(a) Ω_{max}(black) and C_N(white)



(b) Two polyhedral partitions

Fig. 4.8 Graphical illustration for Remark 4.5. The partition is obtained by two vertices of the inner set $\Omega_{\max}$ and two vertices of the outer set C_N

By solving the parametric linear programming problem (4.10) with respect to x, the state space partition is obtained [19]. Figure 4.8(b) shows two polyhedral partitions of the state space partition. The black set is $\Omega_{\max}$. The gray set is the convex hull of *two vertices* of $\Omega_{\max}$ and *two vertices* of C_N.

In conclusion, in $\mathbb{R}^n$ for all $x \in C_N \setminus \Omega_{\max}$, the smallest value c will be reached when $C_N \setminus \Omega_{\max}$ is decomposed into polytopes with vertices both on ∂C_N and $\partial \Omega_{\max}$. These polytopes can be further decomposed into simplices, each formed by r vertices of C_N and $n - r + 1$ vertices of $\Omega_{\max}$ where $1 \le r \le n$.

4.3.3 Explicit Solution

Theorem 4.6 *For all $x \in C_N \setminus \Omega_{\max}$, the controller (4.7), (4.8), (4.10) is a piecewise affine state feedback law defined over a partition of $C_N \setminus \Omega_{\max}$ into simplices. The controller gains are obtained by linear interpolation of the control values at the vertices of simplices.*

Proof Suppose that x belongs to a simplex formed by n vertices $\{v_1, v_2, \ldots, v_n\}$ of C_N and the vertex v_o of $\Omega_{\max}$. The other cases of $n+1$vertices distributed in a different manner between C_N and $\Omega_{\max}$ can be treated similarly.

In this case, x can be expressed as,

$$x = \sum_{i=1}^{n} \beta_i v_i + \beta_{n+1} v_o \tag{4.23}$$

where

$$\sum_{i=1}^{n+1} \beta_i = 1, \quad \beta_i \geq 0 \tag{4.24}$$

Given that $n+1$ linearly independent vectors define a non-empty simplex, let the invertible $(n+1) \times (n+1)$ matrix be

$$T_s = \begin{bmatrix} v_1 & v_2 & \ldots & v_n & v_o \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix} \tag{4.25}$$

Using (4.23), (4.24), (4.25), the interpolating coefficients β_i with $i = 1, 2, \ldots, n+1$ are defined uniquely as,

$$\begin{bmatrix} \beta_1 & \beta_2 & \ldots & \beta_n & \beta_{n+1} \end{bmatrix}^T = T_s^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{4.26}$$

On the other hand, from (4.7),

$$x = c x_v + (1-c) x_o,$$

Due to the uniqueness of (4.23), $\beta_{n+1} = 1 - c$ and

$$x_v = \sum_{i=1}^{n} \frac{\beta_i}{c} v_i$$

The Vertex Controller (3.46) gives

$$u_v = \sum_{i=1}^{n} \frac{\beta_i}{c} u_i$$

where u_i are an admissible control value at v_i, $i = 1, 2, \ldots, n$. Therefore

$$u = cu_v + (1-c)u_o = \sum_{i=1}^{n} \beta_i u_i + \beta_{n+1} u_o.$$

with $u_o = Kx_o$. Together with (4.26), one obtains

$$\begin{aligned} u &= \begin{bmatrix} u_1 & u_2 & \ldots & u_n & u_o \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_2 & \ldots & \beta_n & \beta_{n+1} \end{bmatrix}^T \\ &= \begin{bmatrix} u_1 & u_2 & \ldots & u_n & u_o \end{bmatrix} T_s^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix} \\ &= Lx + v \end{aligned}$$

where the matrix $L \in \mathbb{R}^{m \times n}$ and the vector $v \in \mathbb{R}^m$ are defined by,

$$\begin{bmatrix} L & v \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \ldots & u_n & u_o \end{bmatrix} T_s^{-1}$$

Hence for all $x \in C_N \setminus \Omega_{\max}$ the controller (4.7), (4.8), (4.10) is a piecewise affine state feedback law. □

It is interesting to note that the interpolation between the *piecewise linear* Vertex Controller and the *linear* controller in $\Omega_{\max}$ give rise to a *piecewise affine* controller. This is not completely unexpected since (4.10) is a multi-parametric linear program with respect to x.

As in MPC, the number of cells can be reduced by merging those with identical control laws [45].

Remark 4.6 It can be observed that Algorithm 4.2 uses only the information about the state space partition of the explicit solution of the LP problem (4.10). The explicit form of c^*, r_v^* and r_o^* as a piecewise affine function of the state is not used.

Clearly, the simplex-based partition over $C_N \setminus \Omega_{\max}$ in step 2 might be very complex. Also the fact, that for all facets of $\Omega_{\max}$ the local controller is of the form $u = Kx$, is not exploited. In addition, as practice usually shows, for each facet of C_N, the vertex controller is usually constant. In these cases, the complexity of the explicit interpolating controller (4.7), (4.8), (4.10) might be reduced as follows.

Consider the case when the state space partition CR of $C_N \setminus \Omega_{\max}$ is formed by one vertex x_v of C_N and one facet F_o of $\Omega_{\max}$. Note that from Remark 4.4 such a partition always exists as an explicit solution to the LP problem (4.10). For all $x \in CR$ it follows that

$$x = c^* x_v^* + \left(1 - c^*\right) x_o^* = c^* x_v^* + r_o^*$$

with $x_o^* \in F_o$ and $r_o^* = (1 - c^*) x_o^*$.

Algorithm 4.2 Interpolating control—Explicit solution

Input: The sets C_N, $\Omega_{\max}$, the optimal feedback controller $u = Kx$ in $\Omega_{\max}$ and the control values at the vertices of C_N.
Output: The piecewise affine control law over the partitions of C_N.

1. Solve the LP (4.10) by using explicit multi-parametric linear programming. As a result, one obtains the state space partition of C_N.
2. Decompose each polyhedral partition of $C_N \setminus \Omega_{\max}$ in a sequence of simplices, each formed by r vertices of C_N and $n - z + 1$ vertex of $\Omega_{\max}$, where $1 \le z \le n$. The result is a the state space partition over $C_N \setminus \Omega_{\max}$ in the form of simplices CR_i.
3. In each simplex $CR_i \subset C_N \setminus \Omega_{\max}$ the control law is defined as,

$$u(x) = L_i x + v_i \tag{4.27}$$

   where $L_i \in \mathbb{R}^{m\times n}$ and $v_i \in \mathbb{R}^m$ are defined as

$$\begin{bmatrix} L_i & v_i \end{bmatrix} = \begin{bmatrix} u_1^{(i)} & u_2^{(i)} & \dots & u_{n+1}^{(i)} \end{bmatrix} \begin{bmatrix} v_1^{(i)} & v_2^{(i)} & \dots & v_{n+1}^{(i)} \\ 1 & 1 & \dots & 1 \end{bmatrix}^{-1} \tag{4.28}$$

   with $\{v_1^{(i)}, v_2^{(i)}, \dots v_{n+1}^{(i)}\}$ are vertices of CR_i that defines a full-dimensional simplex and $\{u_1^{(i)}, u_2^{(i)}, \dots u_{n+1}^{(i)}\}$ are the corresponding control values at the vertices.

Let $u_v \in \mathbb{R}^m$ be an admissible control value at x_v and denote the explicit solution of c^* and r_o^* to the LP problem (4.10) for all $x \in CR$ as,

$$\begin{cases} c^* = L_c x + v_c \\ r_o^* = L_o x + v_o \end{cases} \tag{4.29}$$

where L_c, v_c and L_o, v_o are matrices of appropriate dimensions. The control value for $x \in CR$ is computed as,

$$u = c^* u_v + \left(1 - c^*\right) K x_o^* = c^* u_v + K r_o^* \tag{4.30}$$

By substituting (4.29) into (4.30), one obtains

$$u = u_v (L_c x + v_c) + K(L_o x + v_o)$$

or, equivalently

$$u = (u_v L_c + K L_o) x + (u_v v_c + K v_o) \tag{4.31}$$

The fact that the control value is a piecewise affine function of state is confirmed. Clearly, the complexity of the explicit solution with the control law (4.31) is lower than the complexity of the explicit solution with the simplex based partition, since

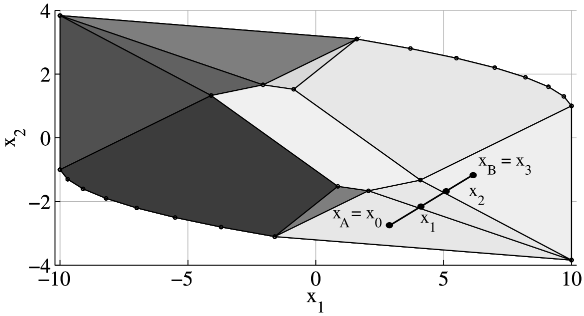


Fig. 4.9 Graphical illustration for the proof of Theorem 4.7

one does not have to divide up the facets of $\Omega_{\max}$ (and facets of C_N, in the case when the vertex control for such facets is constant) into a set of simplices.

4.3.4 Qualitative Analysis

Theorem 4.7 below shows the Lipschitz continuity of the control law based on linear programming (4.7), (4.8), (4.10).

Theorem 4.7 *The explicit interpolating control law* (4.7), (4.8), (4.10) *obtained by using Algorithm* 4.2 *is continuous and Lipschitz continuous with Lipschitz constant* $M = \max_i \|L_i\|$, *where* i *ranges over the set of indices of partitions and* $\|L_i\|$ *is defined in* (4.28).

Proof The explicit interpolating controller might be discontinuous only on the boundary of polyhedral cells CR_i. Suppose that x belongs to the intersection of s cells CR_j, $j = 1, 2, \dots, s$.

For CR_j, as in (4.23), the state x can be expressed as,

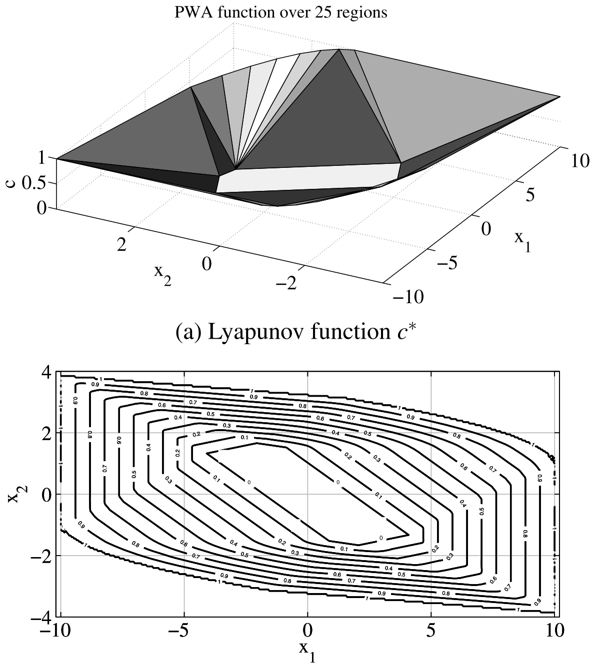
$$x = \beta_1^{(j)} v_1^{(j)} + \beta_2^{(j)} v_2^{(j)} + \cdots + \beta_{n+1}^{(j)} v_{n+1}^{(j)}$$

where $\sum_{i=1}^{n+1} \beta_i^{(j)} = 1$, $0 \le \beta_i^{(j)} \le 1$ and $v_i^{(j)}$, $i = 1, 2, \dots, n+1$ are the vertices of CR_j, $j = 1, 2, \dots, s$. It is clear that the only nonzero entries of the interpolating coefficients $\{\beta_1^{(j)}, \dots, \beta_{n+1}^{(j)}\}$ are those corresponding to the vertices that belong to the intersection. Therefore

$$u = \beta_1^{(j)} u_1^{(j)} + \cdots + \beta_{n+1}^{(j)} u_{n+1}^{(j)}$$

is equal for all $j = 1, 2, \dots, s$.

For the Lipschitz continuity property, for any two points x_A and x_B in C_N, there exist $r+1$ points $x_0, x_1, \dots, x_r$ that lie on the line segment, connecting x_A and x_B, and such that $x_A = x_0$, $x_B = x_r$ and $(x_{i-1}, x_i) = \overline{x_A, x_B} \cap \partial CR_i$, i.e. (x_{i-1}, x_i) is the intersection between the line connecting x_A, x_B and the boundary of some critical



(a) Lyapunov function c^*

(b) Lyapunov level curves

Fig. 4.10 Lyapunov function and Lyapunov level curves for the interpolating controller for Example 4.2

region CR_i, see Fig. 4.9. Due to the continuity property, proved above, of the control law (4.27), one has,

$$
\begin{aligned}
&\big\|(L_A x_A + v_A) - (L_B x_B + v_B)\big\| \\
&\quad = \big\|(L_0 x_0 + v_0) - (L_0 x_1 + v_0) + (L_1 x_1 + v_1) - \cdots - (L_r x_r + v_r)\big\| \\
&\quad = \|L_0 x_0 - L_0 x_1 + L_1 x_1 - \cdots - L_r x_r\| \\
&\quad \le \sum_{i=1}^{r} \big\|L_{i-1}(x_i - x_{i-1})\big\| \le \sum_{k=1}^{r} \|L_{i-1}\| \big\|(x_i - x_{i-1})\big\| \\
&\quad \le \max_k \big\{\|L_{i-1}\|\big\} \sum_{i=1}^{r} \big\|(x_i - x_{i-1})\big\| = M\|x_A - x_B\|
\end{aligned}
$$

where the last equality holds, since the points x_i with $k = 0, 1, \ldots, r$ are aligned. □

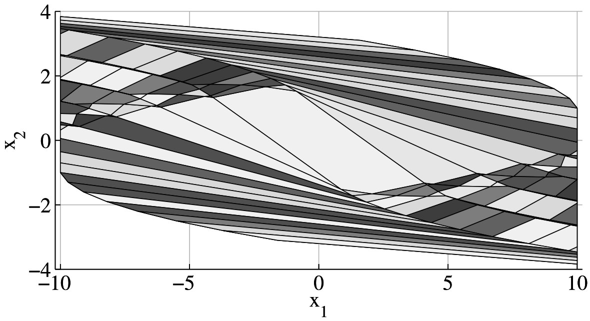
Example 4.2 We consider now the explicit interpolating controller for Example 4.1. Using Algorithm 4.2, the state space partition is obtained in Fig. 4.7. Merging the regions with identical control laws, the reduced state space partition is obtained in Fig. 4.9.

Table 4.2 Number of regions for explicit interpolating control and for explicit MPC for Example 4.2

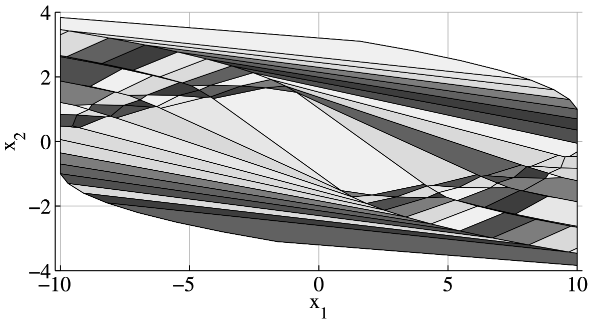
	Before merging	After merging
Explicit interpolating control	25	11
Explicit MPC	127	97

Figure 4.10(a) shows the Lyapunov function as a piecewise affine function of state. It is well known[1] that the level sets of the Lyapunov function for vertex control are simply obtained by scaling the boundary of the set C_N. For the interpolating controller (4.7), (4.8), (4.10), the level sets of the Lyapunov function $V(x) = c^*$ depicted in Fig. 4.10(b) have a more complicated form and generally are not parallel to the boundary of C_N. From Fig. 4.10, it can be observed that the Lyapunov level sets $V(x) = c^*$ have the outer set C_N as an external level set (for $c^* = 1$). The inner level sets change the polytopic shape in order to approach the boundary of the inner set $\Omega_{\max}$.

Fig. 4.11 State space partition before and after merging for Example 4.2 using explicit MPC



(a) Before merging



(b) After merging

[1] See Sect. 3.4.

The control law over the state space partition is,

$$u(k) = \begin{cases}
-1 & \text{if } \begin{bmatrix} 0.45 & 0.89 \\ 0.24 & 0.97 \\ 0.16 & 0.99 \\ -0.55 & 0.84 \\ 0.14 & 0.99 \\ -0.50 & -0.87 \\ 0.20 & 0.98 \\ 0.32 & 0.95 \\ 0.37 & -0.93 \\ 0.70 & 0.71 \end{bmatrix} x(k) \leq \begin{bmatrix} 5.50 \\ 3.83 \\ 3.37 \\ 1.75 \\ 3.30 \\ -0.89 \\ 3.53 \\ 4.40 \\ 2.73 \\ 7.78 \end{bmatrix} \\
-0.38x_1(k) + 0.59x_2(k) - 2.23 & \text{if } \begin{bmatrix} 0.54 & -0.84 \\ -0.37 & 0.93 \\ -0.12 & -0.99 \end{bmatrix} x(k) \leq \begin{bmatrix} -1.75 \\ 2.30 \\ -1.41 \end{bmatrix} \\
-0.02x_1(k) - 0.32x_2(k) + 0.02 & \text{if } \begin{bmatrix} 0.37 & -0.93 \\ 0.06 & 1.00 \\ -0.26 & -0.96 \end{bmatrix} x(k) \leq \begin{bmatrix} -2.30 \\ 3.20 \\ -1.06 \end{bmatrix} \\
-0.43x_1(k) - 1.80x_2(k) + 1.65 & \text{if } \begin{bmatrix} 0.16 & -0.99 \\ 0.26 & 0.96 \\ -0.39 & -0.92 \end{bmatrix} x(k) \leq \begin{bmatrix} -1.97 \\ 1.06 \\ 0.38 \end{bmatrix} \\
0.16x_1(k) - 0.41x_2(k) + 2.21 & \text{if } \begin{bmatrix} 0.39 & 0.92 \\ -1.00 & 0 \\ 0.37 & -0.93 \end{bmatrix} x(k) \leq \begin{bmatrix} -0.38 \\ 10.00 \\ -2.73 \end{bmatrix} \\
1 & \text{if } \begin{bmatrix} -0.14 & -0.99 \\ -0.37 & 0.93 \\ -0.24 & -0.97 \\ -0.71 & -0.71 \\ -0.45 & -0.89 \\ -0.32 & -0.95 \\ -0.20 & -0.98 \\ -0.16 & -0.99 \\ 0.50 & 0.87 \\ 0.54 & -0.84 \end{bmatrix} x(k) \leq \begin{bmatrix} 3.30 \\ 2.73 \\ 3.83 \\ 7.78 \\ 5.50 \\ 4.40 \\ 3.53 \\ 3.37 \\ -0.89 \\ 1.75 \end{bmatrix} \\
-0.38x_1(k) + 0.59x_2(k) + 2.23 & \text{if } \begin{bmatrix} 0.12 & 0.99 \\ 0.37 & -0.93 \\ -0.54 & 0.84 \end{bmatrix} x(k) \leq \begin{bmatrix} -1.41 \\ 2.30 \\ -1.75 \end{bmatrix} \\
-0.02x_1(k) - 0.32x_2(k) - 0.02 & \text{if } \begin{bmatrix} 0.26 & 0.96 \\ -0.06 & -1.00 \\ -0.37 & 0.93 \end{bmatrix} x(k) \leq \begin{bmatrix} -1.06 \\ 3.20 \\ -2.30 \end{bmatrix} \\
-0.43x_1(k) - 1.80x_2(k) - 1.65 & \text{if } \begin{bmatrix} 0.39 & 0.92 \\ -0.26 & -0.96 \\ -0.16 & 0.97 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.38 \\ 1.06 \\ -1.98 \end{bmatrix} \\
0.16x_1(k) - 0.41x_2(k) - 2.21 & \text{if } \begin{bmatrix} 1.00 & 0 \\ -0.37 & 0.93 \\ -0.39 & -0.92 \end{bmatrix} x(k) \leq \begin{bmatrix} 10.00 \\ -2.73 \\ -0.38 \end{bmatrix} \\
-0.56x_1(k) - 0.98x_2(k) & \text{if } \begin{bmatrix} 0.16 & -0.99 \\ -0.16 & 0.99 \\ -0.12 & -0.99 \\ 0.12 & 0.99 \\ -0.50 & -0.87 \\ 0.50 & 0.87 \end{bmatrix} x(k) \leq \begin{bmatrix} 1.97 \\ 1.97 \\ 1.41 \\ 1.41 \\ 0.89 \\ 0.89 \end{bmatrix}
\end{cases}$$

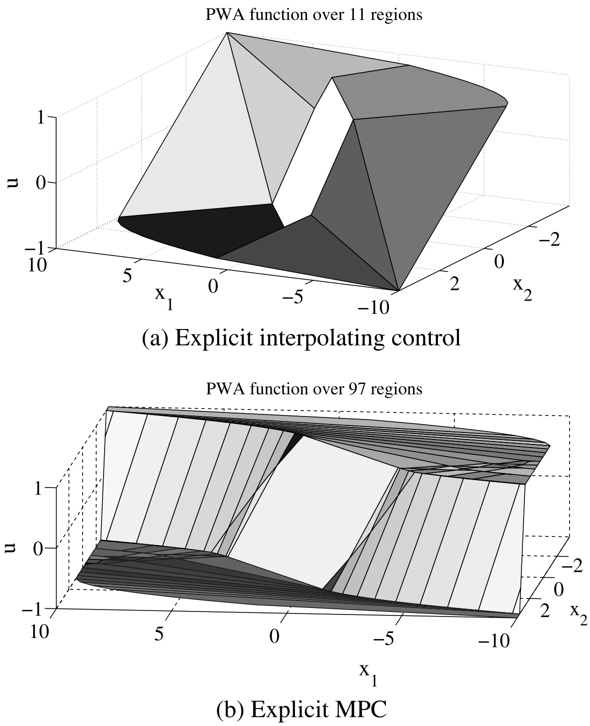


Fig. 4.12 Explicit interpolating control law and explicit MPC control law as piecewise affine functions of state for Example 4.2

In view of comparison, consider the explicit MPC solution in Example 4.1, Fig. 4.11(a) presents the state space partition of the explicit MPC with the same setup parameters as in Example 4.1. Merging the polyhedral regions with an identical piecewise affine control function, the reduced state space partition is obtained in Fig. 4.11(b).

The comparison of explicit interpolating control and explicit MPC in terms of the number of regions before and after merging is given in Table 4.2.

Figure 4.12 shows the explicit interpolating control law and the explicit MPC control law as piecewise affine functions of state, respectively.

4.4 Improved Interpolating Control

The interpolating controller in Sect. 4.2 and Sect. 4.3 can be considered as an approximate model predictive control law, which in the last decade has received significant attention in the control community [18, 60, 63, 78, 108, 114]. From this point of view, it is worthwhile to obtain an interpolating controller with some given level of accuracy in terms of performance compared with the optimal MPC one. Naturally, the approximation error can be a measure of the level of accuracy. The methods of computing bounds on the approximation error are known, see e.g. [18, 60, 114].

Obviously, the simplest way of improving the performance of the interpolating controller is to use an intermediate s-step controlled invariant set C_s with $1 \le s < N$. Then there will be not only one level of interpolation but *two* or virtually *any* number of interpolation as necessary from the performance point of view. For simplicity, we provide in the following a study of the case when only one intermediate controlled invariant set C_s is used. Let C_s be in the form,

$$C_s = \left\{x \in \mathbb{R}^n : F_s x \le g_s\right\} \tag{4.32}$$

and satisfy the condition $\Omega_{\max} \subset C_s \subset C_N$.

Remark 4.7 It has to be noted however that, the expected increase in performance comes at the price of complexity as long as the intermediate set needs to be stored along with its vertex controller.

For further use, the vertex control law applied for the set C_s is denoted as u_s. Using the same philosophy as in Sect. 4.2, the state x is decomposed as,

1. If $x \in C_N$ and $x \notin C_s$, then

$$x = c_1 x_v + (1 - c_1) x_s \tag{4.33}$$

with $x_v \in C_N$, $x_s \in C_s$ and $0 \le c_1 \le 1$. The control law is,

$$u = c_1 u_v + (1 - c_1) u_s \tag{4.34}$$

2. Else $x \in C_s$,

$$x = c_2 x_s + (1 - c_2) x_o \tag{4.35}$$

with $x_s \in C_s$, $x_o \in \Omega_{\max}$ and $0 \le c_2 \le 1$. The control law is,

$$u = c_2 u_s + (1 - c_2) u_o \tag{4.36}$$

Depending on the value of x, at each time instant, either c_1 or c_2 is minimized in order to be as close as possible to the optimal controller. This can be done by solving the following nonlinear optimization problems,

1. If $x \in C_N \setminus C_s$,

$$c_1^* = \min_{x_v, x_s, c_1} \{c_1\} \tag{4.37}$$

subject to

$$\begin{cases} F_N x_v \le g_N, \\ F_s x_s \le g_s, \\ c_1 x_v + (1 - c_1) x_s = x, \\ 0 \le c_1 \le 1 \end{cases}$$

2. Else $x \in C_s$,

$$c_2^* = \min_{x_s, x_o, c_2} \{c_2\} \tag{4.38}$$

subject to

$$\begin{cases} F_s x_s \leq g_s, \\ F_o x_o \leq g_o, \\ c_2 x_s + (1 - c_2) x_o = x, \\ 0 \leq c_2 \leq 1 \end{cases}$$

or by changing variables $r_v = c_1 x_v$ and $r_s = c_2 x_s$, the nonlinear optimization problems (4.37) and (4.38) can be transformed in the following LP problems, respectively,

1. If $x \in C_N \setminus C_s$

$$c_1^* = \min_{r_v, c_1} \{c_1\} \tag{4.39}$$

subject to

$$\begin{cases} F_N r_v \leq c_1 g_N, \\ F_s (x - r_v) \leq (1 - c_1) g_s, \\ 0 \leq c_1 \leq 1 \end{cases}$$

2. Else $x \in C_s$

$$c_2^* = \min_{r_s, c_2} \{c_2\} \tag{4.40}$$

subject to

$$\begin{cases} F_s r_s \leq c_2 g_s, \\ F_o (x - r_s) \leq (1 - c_2) g_o, \\ 0 \leq c_2 \leq 1 \end{cases}$$

The following theorem shows recursive feasibility and asymptotic stability of the interpolating controller (4.33), (4.34), (4.35), (4.36), (4.39), (4.40),

Theorem 4.8 *The control law* (4.33), (4.34), (4.35), (4.36), (4.39), (4.40) *guarantees recursive feasibility and asymptotic stability of the closed loop system for all initial states* $x(0) \in C_N$.

Proof The proof is omitted here, since it follows the same steps as those presented in the feasibility proof of Theorem 4.1 and the stability proof of Theorem 4.2 in Sect. 4.2. □

Remark 4.8 Clearly, instead of the second level of interpolation (4.35), (4.36), (4.40), the MPC approach can be applied for all states inside the set C_s. This has very practical consequences in applications, since it is well known [34, 88] that the main issue of MPC for time-invariant linear discrete-time systems is the trade-off between the overall complexity (computational cost) and the size of the domain of attraction. If the prediction horizon is short then the domain of attraction is small. If the prediction horizon is long then the computational cost may be very burdensome for the available hardware. Here MPC with the short prediction horizon is employed inside C_s for the performance and then for enlarging the domain of attraction, the control law (4.33), (4.34), (4.39) is used. In this way one can achieve the performance and the domain of attraction with a relatively small computational cost.

Theorem 4.9 *The control law* (4.33), (4.34), (4.35), (4.36), (4.39), (4.40) *can be represented as a continuous function of the state.*

Proof Clearly, the discontinuity of the control law may arise only on the boundary of the set C_s, denoted as ∂C_s. Note that for $x \in \partial C_s$, the LP problems (4.39), (4.40) have the trivial solution,

$$c_1^* = 0, \qquad c_2^* = 1$$

Therefore, for $x \in \partial C_s$ the control law (4.33), (4.34), (4.39) is $u = u_s$ and the control law (4.35), (4.36), (4.40) is $u = u_s$. Hence the continuity of the control law is guaranteed. □

Remark 4.9 It is interesting to note that by using $N - 1$ intermediate sets C_i together with the sets C_N and $\Omega_{\max}$, a continuous minimum-time controller is obtained, i.e. a controller that steers all state $x \in C_N$ into $\Omega_{\max}$ in no more than N steps.

Concerning the explicit solution of the control law (4.33), (4.34), (4.35), (4.36), (4.39), (4.40), with the same argument as in Sect. 4.3, it can be concluded that,

- If $x \in C_N \setminus C_s$ (or $x \in C_s \setminus \Omega_{\max}$), the smallest value c_1 (or c_2) will be reached when the region $C_N \setminus C_s$ (or $C_S \setminus \Omega_{\max}$) is decomposed into polyhedral partitions in form of simplices with vertices both on ∂C_N and on ∂C_s (or on ∂C_s and on $\partial \Omega_{\max}$). The control law in each simplex is a piecewise affine function of the state, whose gains are obtained by interpolation of control values at the vertices of the simplex.
- If $x \in \Omega_{\max}$, then the control law is the optimal unconstrained controller.

Example 4.3 Consider again Example 4.1. Here one intermediate set C_s with $s = 4$ is introduced. The set of vertices V_s of C_s is,

$$V_s = \begin{bmatrix} 10.00 & -5.95 & -7.71 & -10.00 & -10.00 & 5.95 & 7.71 & 10.00 \\ -0.06 & 2.72 & 2.86 & 1.78 & 0.06 & -2.72 & -2.86 & -1.78 \end{bmatrix} \tag{4.41}$$

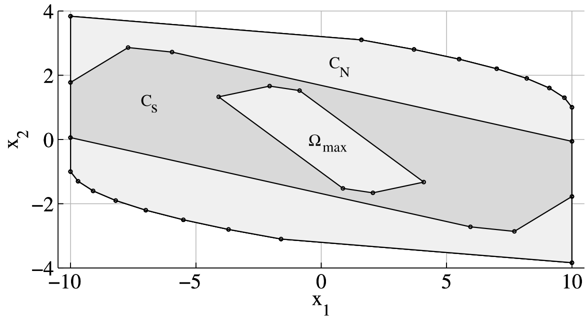


Fig. 4.13 Two-level interpolation for improving the performance

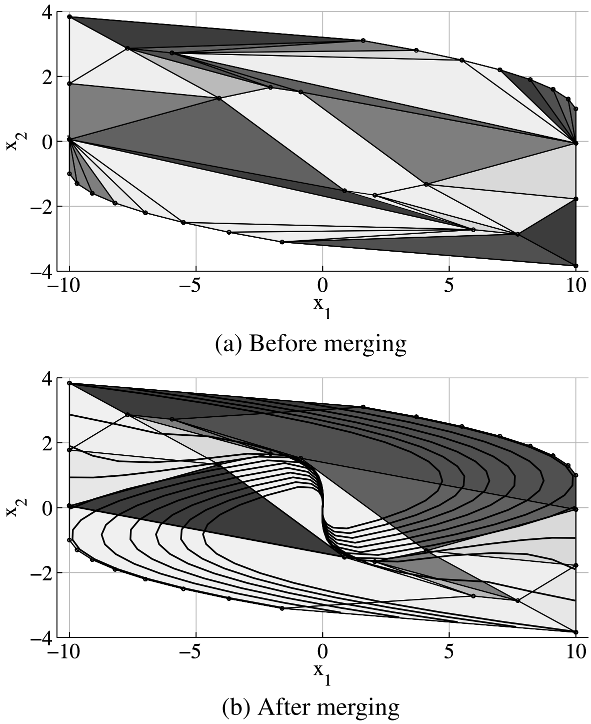


Fig. 4.14 State space partition before merging (number of regions: $N_r = 37$) and after merging ($N_r = 19$), and state trajectories for Example 4.3

and the set of the corresponding control actions at the vertices V_s is,

$$U_s = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.42}$$

The sets C_N, C_s and $\Omega_{\max}$ are depicted in Fig. 4.13. For the explicit solution, the state space partition of the control law (4.33), (4.34), (4.35), (4.36), (4.39), (4.40) is shown in Fig. 4.14(a). Merging the regions with identical control laws, the reduced state space partition is obtained in Fig. 4.14(b). This figure also shows state trajectories of the closed-loop system for different initial conditions.

Figure 4.15 shows the control law with two-level interpolation.

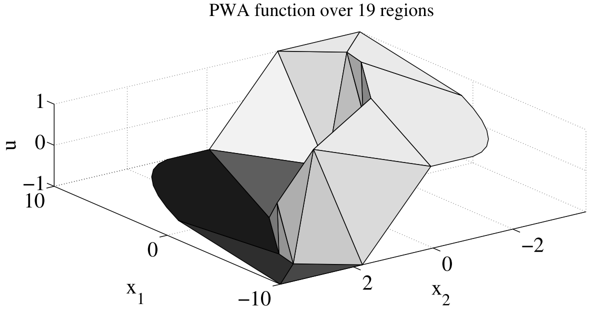
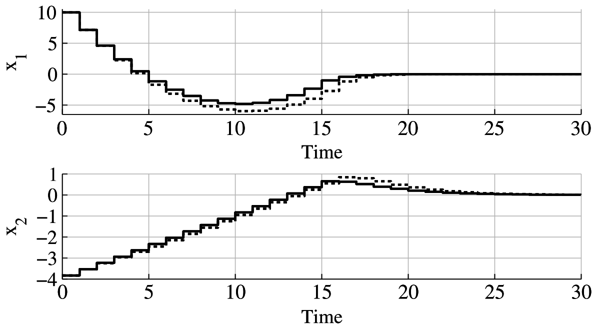
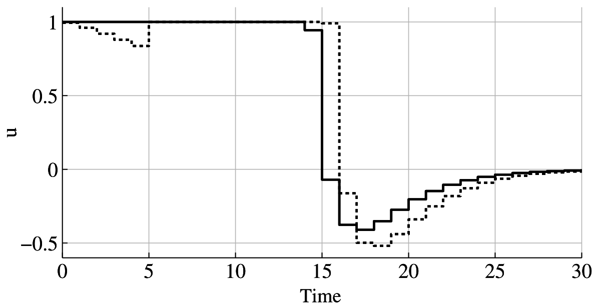


Fig. 4.15 Control value as a piecewise affine function of the state using two-level interpolation for Example 4.3



(a) State trajectories



(b) Input trajectories

Fig. 4.16 State and input trajectories for one-level interpolating control (*dashed*), and for two-level interpolating control (*solid*) for Example 4.3

For the initial condition $x(0) = [9.9800\ -3.8291]^T$, Fig. 4.16 shows the results of a time-domain simulation. The two curves correspond to the one-level and two-level interpolating control, respectively.

Figure 4.17 presents the interpolating coefficients c_1^* and c_2^*. As expected c_1^* and c_2^* are positive and non-increasing. It is also interesting to note that $\forall k \geq 10$, $c_1^*(k) = 0$, indicating that x is inside C_s and $\forall k \geq 14$, $c_2^*(k) = 0$, indicating that x is inside $\Omega_{\max}$.

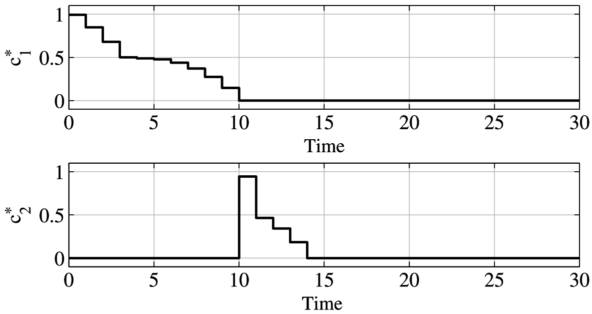


Fig. 4.17 Interpolating coefficients as functions of time for Example 4.3

4.5 Interpolating Control via Quadratic Programming

The interpolating controller in Sect. 4.2 and Sect. 4.4 makes use of linear programming, which is extremely simple. However, the main issue regarding the implementation of Algorithm 4.1 is the non-uniqueness of the solution. Multiple optima are undesirable, as they might lead to a fast switching between the different optimal control actions when the LP problem (4.10) is solved on-line. In addition, MPC traditionally has been formulated using a quadratic criterion [92]. Hence, also in interpolating control it is worthwhile to investigate the use of quadratic programming.

Before introducing a QP formulation, let us note that the idea of using QP for interpolating control is not new. In [10, 110], Lyapunov theory is used to compute an upper bound of the infinite horizon cost function,

$$J = \sum_{k=0}^{\infty} \left\{ x(k)^T Q x(k) + u(k)^T R u(k) \right\} \tag{4.43}$$

where $Q \succeq 0$ and $R \succ 0$ are the state and input weighting matrices. At each time instant, the algorithms in [110] use an on-line decomposition of the current state, with each component lying in a separate invariant set, after which the corresponding controller is applied to each component separately in order to calculate the control action. Polytopes are employed as candidate invariant sets. Hence, the on-line optimization problem can be formulated as a QP problem. The approach taken in this section follows ideas originally proposed in [10, 110]. In this setting we provide a QP based solution to the constrained control problem.

This section begins with a brief summary on the works [10, 110]. For this purpose, it is assumed that a set of unconstrained asymptotically stabilizing feedback controllers $u(k) = K_i x(k)$, $i = 1, 2, \ldots, s$ is available such that the corresponding invariant set $\Omega_i \subseteq X$

$$\Omega_i = \left\{ x \in \mathbb{R}^n : F_o^{(i)} x \le g_o^{(i)} \right\} \tag{4.44}$$

is non-empty for $i = 1, 2, \ldots, s$.

Denote Ω as the convex hull of Ω_i, $i = 1, 2, \ldots, s$. It follows that $\Omega \subseteq X$, since $\Omega_i \subseteq X$, $\forall i = 1, 2, \ldots, s$ and the fact that X is convex. Any state $x(k) \in \Omega$ can be decomposed as,

$$x(k) = \lambda_1(k)\widehat{x}_1(k) + \lambda_2(k)\widehat{x}_2(k) + \cdots + \lambda_s(k)\widehat{x}_s(k) \tag{4.45}$$

where $\widehat{x}_i(k) \in \Omega_i$, $\forall i = 1, 2, \ldots, s$ and $\sum_{i=1}^{s} \lambda_i(k) = 1$, $\lambda_i(k) \geq 0$.

Define $r_i = \lambda_i \widehat{x}_i$. Since $\widehat{x}_i \in \Omega_i$, it follows that $r_i \in \lambda_i \Omega_i$ or equivalently,

$$F_o^{(i)} r_i \leq \lambda_i g_o^{(i)}, \quad \forall i = 1, 2, \ldots, s \tag{4.46}$$

From (4.45), one obtains

$$x(k) = r_1(k) + r_2(k) + \cdots + r_s(k) \tag{4.47}$$

Consider the following control law,

$$u(k) = \sum_{i=1}^{s} \lambda_i K_i \widehat{x}_i = \sum_{i=1}^{s} K_i r_i \tag{4.48}$$

where $u_i(k) = K_i r_i(k)$ is the control law in Ω_i. One has,

$$x(k+1) = Ax(k) + Bu(k) = A\sum_{i=1}^{s} r_i(k) + B\sum_{i=1}^{s} K_i r_i(k) = \sum_{i=1}^{s}(A + BK_i) r_i(k)$$

or,

$$x(k+1) = \sum_{i=1}^{s} r_i(k+1) \tag{4.49}$$

where $r_i(k+1) = A_{ci} r_i(k)$ and $A_{ci} = A + BK_i$.

Define the vector $z \in \mathbb{R}^{sn}$ as,

$$z = \begin{bmatrix} r_1^T & r_2^T & \ldots & r_s^T \end{bmatrix}^T \tag{4.50}$$

Using (4.49), one obtains,

$$z(k+1) = \Phi z(k) \tag{4.51}$$

where

$$\Phi = \begin{bmatrix} A_{c1} & 0 & \ldots & 0 \\ 0 & A_{c2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A_{cs} \end{bmatrix}$$

For the given state and control weighting matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$, consider the following quadratic function,

$$V(z) = z^T P z \tag{4.52}$$

where matrix $P \in \mathbb{R}^{sn \times sn}$, $P \succ 0$ is chosen to satisfy,

$$V\big(z(k+1)\big) - V\big(z(k)\big) \leq -x(k)^T Qx(k) - u(k)^T Ru(k) \tag{4.53}$$

Using (4.51), the left hand side of (4.53) can be rewritten as,

$$V\big(z(k+1)\big) - V\big(z(k)\big) = z(k)^T \big(\Phi^T P \Phi - P\big) z(k) \tag{4.54}$$

and using (4.47), (4.48), (4.50), the right hand side of (4.53) becomes,

$$-x(k)^T Qx(k) - u(k)^T Ru(k) = z(k)^T (Q_1 + R_1) z(k) \tag{4.55}$$

where

$$Q_1 = -\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} Q \begin{bmatrix} I & I & \dots & I \end{bmatrix}, \qquad R_1 = -\begin{bmatrix} K_1^T \\ K_2^T \\ \vdots \\ K_s^T \end{bmatrix} R \begin{bmatrix} K_1 & K_2 & \dots & K_s \end{bmatrix}$$

Combining (4.53), (4.54) and (4.55), one gets,

$$\Phi^T P \Phi - P \preceq Q_1 + R_1$$

or by using the Schur complements, one obtains,

$$\begin{bmatrix} P + Q_1 + R_1 & \Phi^T P \\ P\Phi & P \end{bmatrix} \succeq 0 \tag{4.56}$$

Problem (4.56) is linear with respect to matrix P. Since matrix Φ has a sub-unitary spectral radius (4.51), problem (4.56) is always feasible. One way to obtain P is to solve the following LMI problem,

$$\min_{P} \big\{\text{trace}(P)\big\} \tag{4.57}$$

subject to constraints (4.56).

At each time instant, for a given current state x, consider the following optimization problem,

$$\min_{r_i, \lambda_i} \left\{ \begin{bmatrix} r_1^T & r_2^T & \dots & r_s^T \end{bmatrix} P \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_s \end{bmatrix} \right\} \tag{4.58}$$

subject to

$$\begin{cases} F_o^{(i)} r_i \le \lambda_i g_o^{(i)}, & \forall i = 1, 2, \dots, s, \\ \sum_{i=1}^{s} r_i = x, \\ \sum_{i=1}^{s} \lambda_i = 1, \\ \lambda_i \ge 0, & \forall i = 1, 2, \dots, s \end{cases}$$

and implement as input the control action $u = \sum_{i=1}^{s} K_i r_i$.

Theorem 4.10 [10, 110] *The control law* (4.45), (4.48), (4.58) *guarantees recursive feasibility and asymptotic stability for all initial states* $x(0) \in \Omega$.

Note that using the approach in [10, 110], for a given state x we are trying to minimize $r_1, r_2, \dots, r_s$ in the weighted Euclidean norm sense. This is somehow a conflicting task, since,

$$r_1 + r_2 + \cdots + r_s = x$$

In addition, if the first controller is optimal and plays the role of a performance controller, then one would like to have a control law as close as possible to the first controller. This means that in the interpolation scheme (4.45), one would like to have $r_1 = x$ and

$$r_2 = r_3 = \cdots = r_s = 0$$

whenever it is possible. And it is not trivial to do this with the approach in [10, 110].

Below we provide a contribution to this line of research by considering one of the interpolation factors, i.e. control gains to be a performance related one, while the remaining factors play the role of degrees of freedom to enlarge the domain of attraction. This alternative approach can provide the appropriate framework for the constrained control design which builds on the unconstrained optimal controller (generally with high gain) and subsequently need to adjusted them to cope with the constraints and limitations (via interpolation with adequate low gain controllers). From this point of view, in the remaining part of this section we try to build a bridge between the linear interpolation scheme presented in Sect. 4.2 and the QP based interpolation approaches in [10, 110].

For a given set of state and control weighting matrices $Q_i \succeq 0$, $R_i \succ 0$, consider the following set of quadratic functions,

$$V_i(r_i) = r_i^T P_i r_i, \quad \forall i = 2, 3, \dots, s \tag{4.59}$$

where matrix $P_i \in \mathbb{R}^{n \times n}$ and $P_i \succ 0$ is chosen to satisfy

$$V_i\big(r_i(k+1)\big) - V_i\big(r_i(k)\big) \le -r_i(k)^T Q_i r_i(k) - u_i(k)^T R_i u_i(k) \tag{4.60}$$

Since $r_i(k+1) = A_{ci} r_i(k)$ and $u_i(k) = K_i r_i(k)$, equation (4.60) can be written as,

$$A_{ci}^T P_i A_{ci} - P_i \preceq -Q_i - K_i^T R_i K_i$$

By using the Schur complements, one obtains

$$\begin{bmatrix} P_i - Q_i - K_i^T R_i K_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} \succeq 0 \tag{4.61}$$

Since matrix A_{ci} has a sub-unitary spectral radius, problem (4.61) is always feasible. One way to obtain matrix P_i is to solve the following LMI problem,

$$\min_{P_i} \big\{ \text{trace}(P_i) \big\} \tag{4.62}$$

subject to constraint (4.61).

Define the vector $z_1 \in \mathbb{R}^{(s-1)(n+1)}$ as,

$$z_1 = \begin{bmatrix} r_2^T & r_3^T & \dots & r_s^T & \lambda_2 & \lambda_3 & \dots & \lambda_s \end{bmatrix}^T$$

Consider the following quadratic function,

$$J(z_1) = \sum_{i=2}^{s} r_i^T P_i r_i + \sum_{i=2}^{s} \lambda_i^2 \tag{4.63}$$

We underline the fact that the sum is built on indices $\{2, 3, \dots, s\}$, corresponding to the more poorly performing controllers. At each time instant, consider the following optimization problem,

$$V_1(z_1) = \min_{z_1} \big\{ J(z_1) \big\} \tag{4.64}$$

subject to the constraints

$$\begin{cases} F_o^{(i)} r_i \leq \lambda_i g_o^{(i)}, \forall i = 1, 2, \dots, s, \\ \sum_{i=1}^{s} r_i = x, \\ \sum_{i=1}^{s} \lambda_i = 1, \\ \lambda_i \geq 0, \forall i = 1, 2, \dots, s \end{cases}$$

and apply as input the control signal $u = \sum_{i=1}^{s} \{K_i r_i\}$.

Theorem 4.11 *The control law (4.45), (4.48), (4.64) guarantees recursive feasibility and asymptotic stability for all initial states* $x(0) \in \Omega$.

Proof Theorem 4.11 makes two important claims, namely the recursive feasibility and the asymptotic stability. These can be treated sequentially.

Recursive feasibility: It has to be proved that $F_u u(k) \leq g_u$ and $x(k+1) \in \Omega$ for all $x(k) \in \Omega$. It holds that,

$$F_u u(k) = F_u \sum_{i=1}^{s} \lambda_i K_i \widehat{x}_i = \sum_{i=1}^{s} \lambda_i F_u K_i \widehat{x}_i \leq \sum_{i=1}^{s} \lambda_i g_u = g_u$$

and

$$x(k+1) = Ax(k) + Bu(k) = \sum_{i=1}^{s} \lambda_i A_{ci} \widehat{x}_i(k)$$

Since $A_{ci}\widehat{x}_i(k) \in \Omega_i \subseteq \Omega$, it follows that $x(k+1) \in \Omega$.

Asymptotic stability: Consider the positive function $V_1(z_1)$ as a candidate Lyapunov function. From the recursive feasibility proof, it is apparent that if $\lambda_1^*(k)$, $\lambda_2^*(k), \ldots, \lambda_s^*(k)$ and $r_1^*(k), r_2^*(k), \ldots, r_s^*(k)$ is the solution of the optimization problem (4.64) at time instant k, then $\lambda_i(k+1) = \lambda_i^*(k)$ and

$$r_i(k+1) = A_{ci} r_i^*(k)$$

$\forall i = 1, 2, \ldots, s$ is a feasible solution to (4.64) at time instant $k+1$. Since at each time instant we are trying to minimize $J(z_1)$, it follows that

$$V_1\big(z_1^*(k+1)\big) \leq J\big(z_1(k+1)\big)$$

and therefore

$$V_1\big(z_1^*(k+1)\big) - V_1\big(z_1^*(k)\big) \leq J\big(z_1(k+1)\big) - V_1\big(z_1^*(k)\big)$$

together with (4.60), one obtains

$$V_1\big(z_1^*(k+1)\big) - V_1\big(z_1^*(k)\big) \leq -\sum_{i=2}^{s}\big(r_i^T Q_i r_i + u_i^T R_i u_i\big)$$

Hence $V_1(z_1)$ is a Lyapunov function and the control law (4.45), (4.48), (4.64) assures asymptotic stability for all $x \in \Omega$. □

The constraints of the problem (4.64) can be rewritten as,

$$\begin{cases} F_o^{(1)}(x - r_2 - \cdots - r_s) \leq (1 - \lambda_2 - \cdots - \lambda_s) g_o^{(1)} \\ F_o^{(2)} r_2 \leq \lambda_2 g_o^{(2)} \\ \quad \vdots \\ F_o^{(s)} r_s \leq \lambda_s g_o^{(s)} \\ \lambda_i \geq 0, \quad \forall i = 2, \ldots, s \\ \lambda_2 + \lambda_3 + \cdots + \lambda_s \leq 1 \end{cases}$$

or, equivalently

$$G z_1 \leq S + E x \tag{4.65}$$

where

$$G = \begin{bmatrix} -F_o^{(1)} & -F_o^{(1)} & \ldots & -F_o^{(1)} & g_o^{(1)} & g_o^{(1)} & \ldots & g_o^{(1)} \\ F_o^{(2)} & 0 & \ldots & 0 & -g_o^{(2)} & 0 & \ldots & 0 \\ 0 & F_o^{(3)} & \ldots & 0 & 0 & -g_o^{(3)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & F_o^{(s)} & 0 & 0 & \ldots & -g_o^{(s)} \\ 0 & 0 & \ldots & 0 & -1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & -1 \\ 0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 \end{bmatrix},$$

$$S = \begin{bmatrix} (g_o^{(1)})^T & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix}^T$$

$$E = \begin{bmatrix} -(F_o^{(1)})^T & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix}^T$$

And the objective function (4.64) can be written as,

$$\min_{z_1} \{ z_1^T H z_1 \} \tag{4.66}$$

Algorithm 4.3 Interpolating control via quadratic programming

1. Measure the current state $x(k)$.
2. Solve the QP problem (4.66), (4.65).
3. Apply the control input (4.48).
4. Wait for the next time instant $k := k + 1$.
5. Go to step 1 and repeat.

where

$$H = \begin{bmatrix} P_2 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & P_3 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_s & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Hence, the optimization problem (4.64) is transformed into the quadratic programming problem (4.66), (4.65).

It is worth noticing that for all $x \in \Omega_1$, the QP problem (4.66), (4.65) has the trivial solution, namely

$$\begin{cases} r_i = 0, \\ \lambda_i = 0 \end{cases} \quad \forall i = 2, 3, \dots, s$$

Hence $r_1 = x$ and $\lambda_1 = 1$. That means, inside the set Ω_1, the interpolating controller (4.45), (4.48), (4.64) becomes the optimal unconstrained controller.

Remark 4.10 Note that Algorithm 4.3 requires the solution of the QP problem (4.66) of dimension $(s-1)(n+1)$ where s is the number of interpolated controllers and n is the dimension of state. Clearly, solving the QP problem (4.66) can be computationally expensive when the number of interpolated controllers is big. However, it is usually enough with $s = 2$ or $s = 3$ in terms of performance and in terms of the size of the domain of attraction.

Example 4.4 Consider again the system in Example 4.2 with the same state and control constraints. Two linear feedback controllers are chosen as,

$$\begin{cases} K_1 = [-0.0942 \quad -0.7724] \\ K_2 = [-0.0669 \quad -0.2875] \end{cases} \tag{4.67}$$

The first controller $u(k) = K_1 x(k)$ is an optimal controller and plays the role of the performance controller, and the second controller $u(k) = K_2 x(k)$ is used to enlarge the domain of attraction.

Figure 4.18(a) shows the invariant sets Ω_1 and Ω_2 correspond to the controllers K_1 and K_2, respectively. Figure 4.18(b) shows state trajectories obtained by solving the QP problem (4.66), (4.65) for different initial conditions.

The sets Ω_1 and Ω_2 are presented in minimal normalized half-space representation as,

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.1211 & -0.9926 \\ 0.1211 & 0.9926 \end{bmatrix} x \le \begin{bmatrix} 10.0000 \\ 10.0000 \\ 1.2851 \\ 1.2851 \end{bmatrix} \right\}$$

$$\Omega_2 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.2266 & -0.9740 \\ 0.2266 & 0.9740 \\ 0.7948 & 0.6069 \\ -0.7948 & -0.6069 \\ -0.1796 & -0.9837 \\ 0.1796 & 0.9837 \\ -0.1425 & -0.9898 \\ 0.1425 & 0.9898 \\ -0.1117 & -0.9937 \\ 0.1117 & 0.9937 \\ -0.0850 & -0.9964 \\ 0.0850 & 0.9964 \\ -0.0610 & -0.9981 \\ 0.0610 & 0.9981 \\ -0.0386 & -0.9993 \\ 0.0386 & 0.9993 \\ -0.0170 & -0.9999 \\ 0.0170 & 0.9999 \end{bmatrix} x \le \begin{bmatrix} 10.0000 \\ 10.0000 \\ 3.3878 \\ 3.3878 \\ 8.5177 \\ 8.5177 \\ 3.1696 \\ 3.1696 \\ 3.0552 \\ 3.0552 \\ 3.0182 \\ 3.0182 \\ 3.0449 \\ 3.0449 \\ 3.1299 \\ 3.1299 \\ 3.2732 \\ 3.2732 \\ 3.4795 \\ 3.4795 \end{bmatrix} \right\}$$

For the weighting matrices $Q_2 = I$, $R_2 = 1$, and by solving the LMI problem (4.62), one obtains,

$$P_2 = \begin{bmatrix} 5.1917 & 9.9813 \\ 9.9813 & 101.2651 \end{bmatrix} \tag{4.68}$$

For the initial condition $x(0) = [6.8200\ 1.8890]^T$, Fig. 4.19(a) and 4.19(b) present the state and input trajectories of the closed loop system for our approach (solid), and for the approach in [110] (dashed).

For [110], the matrix P in the problem (4.57) is computed as,

$$P = \begin{bmatrix} 4.8126 & 2.9389 & 4.5577 & 13.8988 \\ 2.9389 & 7.0130 & 2.2637 & 20.4391 \\ 4.5577 & 2.2637 & 5.1917 & 9.9813 \\ 13.8988 & 20.4391 & 9.9813 & 101.2651 \end{bmatrix}$$

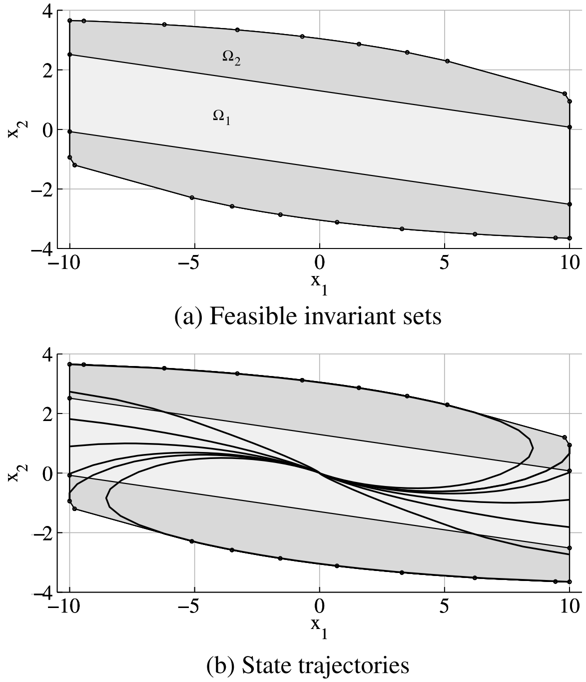


Fig. 4.18 Feasible invariant sets and state trajectories of the closed loop system for Example 4.4

for the weighting matrices $Q = I$, $R = 1$.

The interpolating coefficient λ_2^* and the Lyapunov function $V_1(z_1)$ are depicted in Fig. 4.20. As expected $V_1(z_1)$ is a positive and non-increasing function.

4.6 Interpolating Control Based on Saturated Controllers

In this section, in order to fully utilize the capability of actuators and to enlarge the domain of attraction, an interpolation between several saturated controllers will be proposed. For simplicity, only single-input single-output system is considered, although extensions to multi-input multi-output systems are straightforward.

From Lemma 2.1 in Sect. 2.4.1, recall that for a given stabilizing controller $u(k) = Kx(k)$, there exists an auxiliary stabilizing controller $u(k) = Hx(k)$ such that the saturation function can be expressed as, $\forall x$ such that $Hx \in U$,

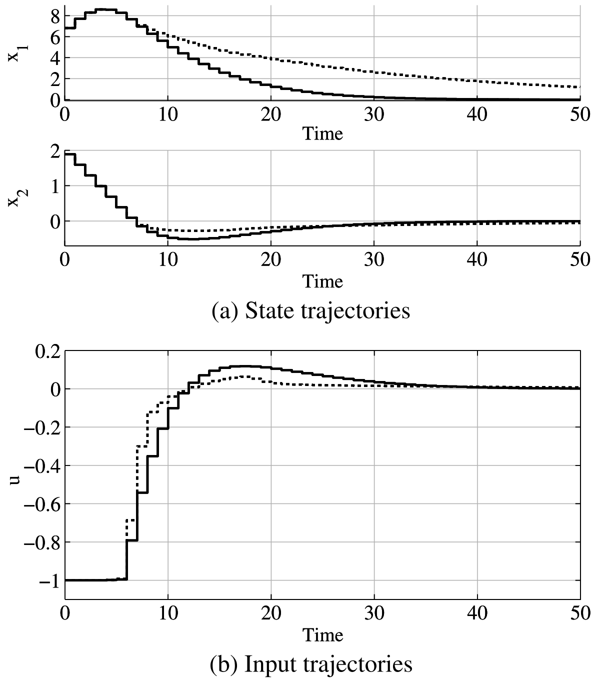
$$\mathrm{sat}\big(Kx(k)\big) = \alpha(k)Kx(k) + \big(1 - \alpha(k)\big)Hx(k) \tag{4.69}$$

where $0 \le \alpha(k) \le 1$. Matrix $H \in \mathbb{R}^n$ can be computed using Theorem 2.3. Using Procedure 2.5 in Sect. 2.4.1, the polyhedral set Ω_s^H can be computed, which is invariant for system,

$$x(k+1) = Ax(k) + B\,\mathrm{sat}\big(Kx(k)\big) \tag{4.70}$$

and with respect to the constraints (4.2).

Fig. 4.19 State and input trajectories of the closed loop system for our approach (*solid*), and for the approach in [110] (*dashed*) for Example 4.4



It is assumed that a set of asymptotically stabilizing feedback controllers $K_i \in \mathbb{R}^n$, $i = 1, 2, \dots, s$ is available as well as a set of auxiliary matrices $H_i \in \mathbb{R}^n$, $i = 2, \dots, s$ such that the corresponding invariant sets $\Omega_1 \subseteq X$

$$\Omega_1 = \left\{x \in \mathbb{R}^n : F_o^{(1)} x \le g_o^{(1)}\right\} \tag{4.71}$$

for the linear controller $u = K_1 x$ and $\Omega_s^{H_i} \subseteq X$

$$\Omega_s^{H_i} = \left\{x \in \mathbb{R}^n : F_o^{(i)} x \le g_o^{(i)}\right\} \tag{4.72}$$

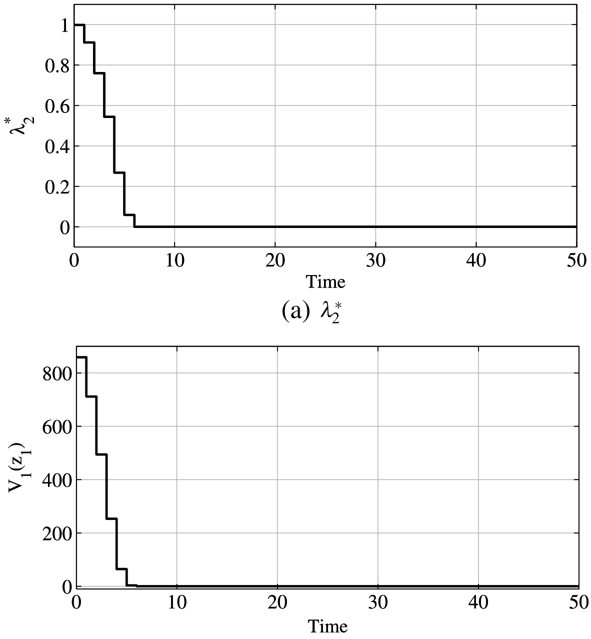
for the saturated controllers $u = \text{sat}(K_i x)$, $\forall i = 2, 3, \dots, s$, are non-empty. Denote Ω_s as the convex hull of the sets Ω_1 and $\Omega_s^{H_i}$, $i = 2, 3, \dots, s$. It follows that $\Omega_s \subseteq X$, since $\Omega_1 \subseteq X$, $\Omega_s^{H_i} \subseteq X$, $\forall i = 2, 3, \dots, s$ and the fact that X is a convex set.

Remark 4.11 We use one linear control law here in order to show that interpolation can be done between any kind of controllers: *linear or saturated*. The main requirement is that there exists for each of these controllers its own convex invariant set as the domain of attraction.

Any state $x(k) \in \Omega_s$ can be decomposed as,

$$x(k) = \lambda_1(k)\widehat{x}_1(k) + \sum_{i=2}^{s} \lambda_i(k)\widehat{x}_i(k) \tag{4.73}$$

Fig. 4.20 Interpolating coefficient λ_2^* and the Lyapunov function $V_1(z_1)$ as functions of time for Example 4.4



(a) λ_2^*

(b) $V_1(z_1)$

where $\widehat{x}_1(k) \in \Omega_1$, $\widehat{x}_i(k) \in \Omega_s^{H_i}$, $i = 2, 3, \ldots, s$ and

$$\sum_{i=1}^{s} \lambda_i(k) = 1, \quad \lambda_i(k) \geq 0.$$

Consider the following control law,

$$u(k) = \lambda_1(k) K_1 \widehat{x}_1(k) + \sum_{i=2}^{s} \lambda_i(k) \,\mathrm{sat}\big(K_i \widehat{x}_i(k)\big) \tag{4.74}$$

Using Lemma 2.1, one obtains,

$$u(k) = \lambda_1(k) K_1 \widehat{x}_1(k) + \sum_{i=2}^{s} \lambda_i(k)\big(\alpha_i(k) K_i + \big(1 - \alpha_i(k)\big) H_i\big)\widehat{x}_i(k) \tag{4.75}$$

where $0 \leq \alpha_i(k) \leq 1$ for all $i = 2, 3, \ldots, s$.

Similar with the notation employed in Sect. 4.5, we denote $r_i = \lambda_i \widehat{x}_i$. Since $\widehat{x}_1 \in \Omega_1$ and $\widehat{x}_i \in \Omega_s^{H_i}$, it follows that $r_1 \in \lambda_1 \Omega_1$ and $r_i \in \lambda_i \Omega_s^{H_i}$ or, equivalently

$$F_o^{(i)} r_i \leq \lambda_i g_o^{(i)}, \quad \forall i = 1, 2, \ldots, s \tag{4.76}$$

Based on (4.73) and (4.75), one obtains,

$$\begin{cases} x = r_1 + \sum_{i=2}^{s} r_i, \\ u = u_1 + \sum_{i=2}^{s} u_i \end{cases} \tag{4.77}$$

where $u_1 = K_1 r_1$ and $u_i = (\alpha_i K_i + (1-\alpha_i)H_i)r_i$, $i = 2, 3, \dots, s$.

As in Sect. 4.5, the first controller, identified by the high gain K_1, will play the role of a performance controller, while the remaining controllers $u = \text{sat}(K_i x)$, $i = 2, 3, \dots, s$ will be used to extend the domain of attraction.

It holds that,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= A\sum_{i=1}^{s} r_i(k) + B\sum_{i=1}^{s} u_i = \sum_{i=1}^{s} r_i(k+1) \end{aligned}$$

where $r_1(k+1) = Ar_1 + Bu_1 = (A + BK_1)r_1$ and

$$r_i(k+1) = Ar_i(k) + Bu_i(k) = \big\{A + B\big(\alpha_i K_i + (1-\alpha_i)H_i\big)\big\} r_i(k) \tag{4.78}$$

or, equivalently

$$r_i(k+1) = A_{ci} r_i(k) \tag{4.79}$$

with $A_{ci} = A + B(\alpha_i K_i + (1-\alpha_i)H_i)$, $\forall i = 2, 3, \dots, s$.

For a given set of state and control weighting matrices $Q_i \succeq 0$ and $R_i \succ 0$, $i = 2, 3, \dots, s$, consider the following set of quadratic functions,

$$V_i(r_i) = r_i^T P_i r_i, \quad i = 2, 3, \dots, s \tag{4.80}$$

where the matrix $P_i \in \mathbb{R}^{n\times n}$, $P_i \succ 0$ is chosen to satisfy,

$$V_i\big(r_i(k+1)\big) - V_i\big(r_i(k)\big) \le -r_i(k)^T Q_i r_i(k) - u_i(k)^T R_i u_i(k) \tag{4.81}$$

With the same argument as in Sect. 4.5, equation (4.81) can be rewritten as,

$$A_{ci}^T P_i A_{ci} - P_i \preceq -Q_i - \big(\alpha_i K_i + (1-\alpha_i)H_i\big)^T R_i \big(\alpha_i K_i + (1-\alpha_i)H_i\big)$$

Using the Schur complements, the above condition can be transformed into,

$$\begin{bmatrix} P_i - Q_i - Y_i^T R_i Y_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} \succeq 0$$

where $Y_i = \alpha_i K_i + (1-\alpha_i)H_i$. Or, equivalently

$$\begin{bmatrix} P_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} - \begin{bmatrix} Q_i + Y_i^T R_i Y_i & 0 \\ 0 & 0 \end{bmatrix} \succeq 0$$

Denote $\sqrt{Q_i}$ and $\sqrt{R_i}$ as the Cholesky factor of the matrices Q_i and R_i, which satisfy

$$\sqrt{Q_i}^T \sqrt{Q_i} = Q_i \quad \text{and} \quad \sqrt{R_i}^T \sqrt{R_i} = R_i.$$

The previous condition can be rewritten as,

$$\begin{bmatrix} P_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} - \begin{bmatrix} \sqrt{Q_i}^T & Y_i^T \sqrt{R_i}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{Q_i} & 0 \\ \sqrt{R_i} Y_i & 0 \end{bmatrix} \succeq 0$$

or by using the Schur complements, one obtains,

$$\begin{bmatrix} P_i & A_{ci}^T P_i & \sqrt{Q_i}^T & Y_i^T \sqrt{R_i}^T \\ P_i A_{ci} & P_i & 0 & 0 \\ \sqrt{Q_i} & 0 & I & 0 \\ \sqrt{R_i} Y_i & 0 & 0 & I \end{bmatrix} \succeq 0 \tag{4.82}$$

Since $Y_i = \alpha_i K_i + (1 - \alpha_i) H_i$, and $A_{ci} = A + BY_i$ the left hand side of inequality (4.82) is linear in α_i, and hence reaches its minimum at either $\alpha_i = 0$ or $\alpha_i = 1$. Consequently, the set of LMI conditions to be checked is following,

$$\begin{cases} \begin{bmatrix} P_i & (A + BK_i)^T P_i & \sqrt{Q_i}^T & (\sqrt{R_i} K_i)^T \\ P_i (A + BK_i) & P_i & 0 & 0 \\ \sqrt{Q_i} & 0 & I & 0 \\ \sqrt{R_i} K_i & 0 & 0 & I \end{bmatrix} \succeq 0 \\ \begin{bmatrix} P_i & (A + BH_i)^T P_i & \sqrt{Q_i}^T & (\sqrt{R_i} H_i)^T \\ P_i (A + BH_i) & P_i & 0 & 0 \\ \sqrt{Q_i} & 0 & I & 0 \\ \sqrt{R_i} H_i & 0 & 0 & I \end{bmatrix} \succeq 0 \end{cases} \tag{4.83}$$

Condition (4.83) is linear with respect to the matrix P_i. One way to calculate P_i is to solve the following LMI problem,

$$\min_{P_i} \{\text{trace}(P_i)\} \tag{4.84}$$

subject to constraint (4.83).

Once the matrices P_i, $i = 2, 3, \ldots, s$ are computed, they can be used in practice for real-time control based on the following algorithm, which can be formulated as an optimization problem that is efficient with respect to structure and complexity. At each time instant, for a given current state x, minimize on-line the quadratic cost function,

$$\min_{r_i,\lambda_i}\left\{\sum_{i=2}^{s} r_i^T P_i r_i + \sum_{i=2}^{s} \lambda_i^2\right\} \tag{4.85}$$

subject to the linear constraints

$$\begin{cases} F_o^{(i)} r_i \le \lambda_i g_o^{(i)}, \quad \forall i = 1,2,\ldots,s, \\ \sum_{i=1}^{s} r_i = x, \\ \sum_{i=1}^{s} \lambda_i = 1, \\ \lambda_i \ge 0, \quad \forall i = 1,2,\ldots,s \end{cases}$$

Theorem 4.12 *The control law* (4.73), (4.74), (4.85) *guarantees recursive feasibility and asymptotic stability of the closed loop system for all initial states* $x(0) \in \Omega_s$.

Proof The proof is similar to Theorem 4.11. Hence it is omitted here. □

Example 4.5 Consider again the system in Example 4.1 with the same state and control constraints. Two gain matrices are chosen as,

$$\begin{cases} K_1 = [-0.9500 \quad -1.1137], \\ K_2 = [-0.4230 \quad -2.0607] \end{cases} \tag{4.86}$$

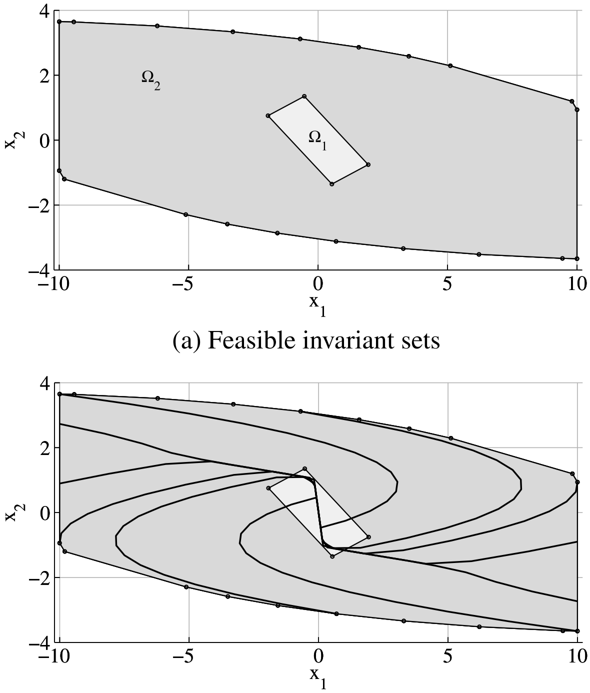
Using Theorem 2.3, matrix H_2 is computed as,

$$H_2 = [-0.0669 \quad -0.2875] \tag{4.87}$$

The invariant sets Ω_1 and $\Omega_s^{H_2}$ are, respectively constructed for the controllers $u = K_1 x$ and $u = \text{sat}(K_2 x)$, see Fig. 4.21(a). Figure 4.21(b) shows state trajectories for different initial conditions.

The sets Ω_1 and $\Omega_s^{H_2}$ are presented in minimal normalized half-space representation as,

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.3919 & -0.9200 \\ -0.3919 & 0.9200 \\ -0.6490 & -0.7608 \\ 0.6490 & 0.7608 \end{bmatrix} x \le \begin{bmatrix} 1.4521 \\ 1.4521 \\ 0.6831 \\ 0.6831 \end{bmatrix} \right\}$$



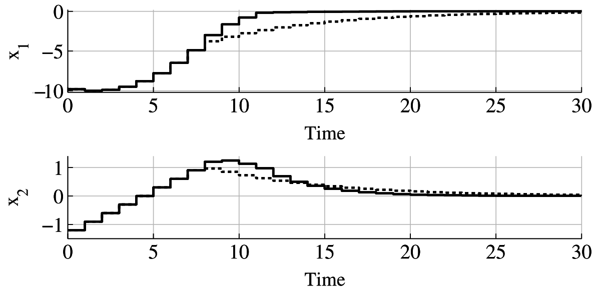
(a) Feasible invariant sets

(b) State trajectories

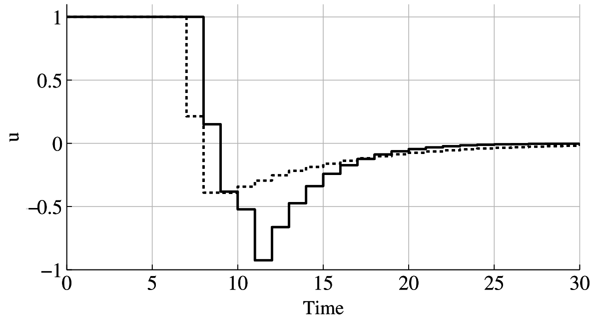
Fig. 4.21 Feasible invariant sets and state trajectories of the closed loop system for Example 4.5

$$\Omega_s^{H_2} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.0170 & -0.9999 \\ 0.0170 & 0.9999 \\ -0.0386 & -0.9993 \\ 0.0386 & 0.9993 \\ -0.0610 & -0.9981 \\ 0.0610 & 0.9981 \\ -0.0850 & -0.9964 \\ 0.0850 & 0.9964 \\ -0.1117 & -0.9937 \\ 0.1117 & 0.9937 \\ -0.1425 & -0.9898 \\ 0.1425 & 0.9898 \\ 0.7948 & 0.6069 \\ -0.7948 & -0.6069 \\ -0.1796 & -0.9837 \\ 0.1796 & 0.9837 \\ 1.0000 & 0 \\ -1.0000 & 0 \\ -0.2266 & -0.9740 \\ 0.2266 & 0.9740 \end{bmatrix} x \le \begin{bmatrix} 3.4795 \\ 3.4795 \\ 3.2732 \\ 3.2732 \\ 3.1299 \\ 3.1299 \\ 3.0449 \\ 3.0449 \\ 3.0182 \\ 3.0182 \\ 3.0552 \\ 3.0552 \\ 8.5177 \\ 8.5177 \\ 3.1696 \\ 3.1696 \\ 10.0000 \\ 10.0000 \\ 3.3878 \\ 3.3878 \end{bmatrix} \right\}$$

Fig. 4.22 State and input trajectories of the closed loop system as functions of time for Example 4.5 for the interpolating controller (*solid*) and for the saturated controller $u = \text{sat}(K_2 x)$ (*dashed*)



(a) State trajectories



(b) Input trajectories

With the weighting matrices $Q_2 = I$, $R_2 = 0.001$ and by solving the LMI problem (4.84), one obtains,

$$P_2 = \begin{bmatrix} 5.4929 & 9.8907 \\ 9.8907 & 104.1516 \end{bmatrix}$$

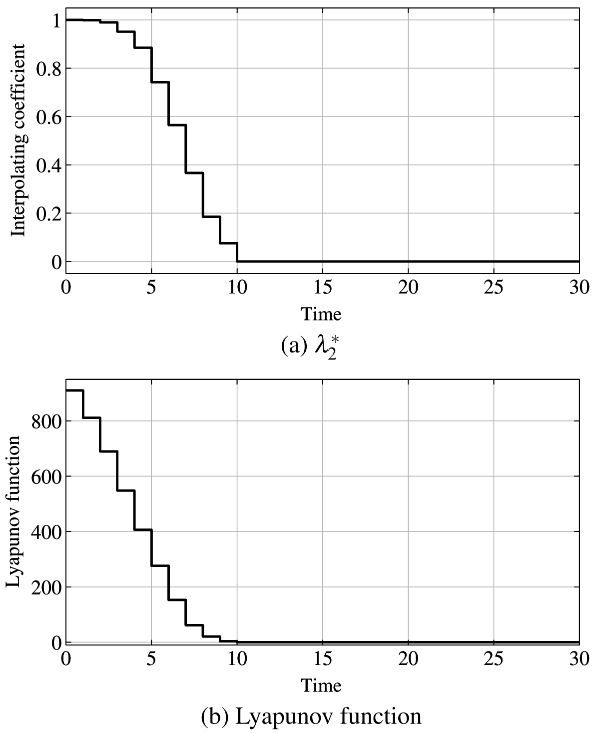
For the initial condition $x(0) = [-9.79 \ -1.2]^T$, Fig. 4.22 presents the state and input trajectories for the interpolating controller (solid blue) and for the saturated controller $u = \text{sat}(K_2 x)$ (dashed red), which is the controller corresponding to the set $\Omega_s^{H_2}$. The interpolating coefficient λ_2^* and the objective function as a Lyapunov function are shown in Fig. 4.23.

4.7 Convex Hull of Ellipsoids

For high dimensional systems, the polyhedral based interpolation approaches in Sects. 4.2, 4.3, 4.4, 4.5, 4.6 might be impractical due to the huge number of vertices or half-spaces in the representation of polyhedral sets. In that case, ellipsoids might be a suitable class of sets for interpolation.

Note that the idea of using ellipsoids for a constrained control system is well known, for time-invariant linear continuous-time systems, see [56], and for time-invariant linear discrete-time systems, see [10]. In these papers, a method to construct a continuous control law based on a set of *linear* control laws was proposed

Fig. 4.23 Interpolating coefficient λ_2^* and Lyapunov function as functions of time for Example 4.5



to make the convex hull of an associated set of invariant ellipsoids *invariant*. However these results do not allow to impose priority among the control laws.

In this section, an interpolation using a set of *saturated* controllers and its associated set of invariant ellipsoid is presented. The main contribution with respect to [10, 56] is to provide a new type of controller, that uses interpolation.

It is assumed that a set of asymptotically stabilizing saturated controllers $u = \text{sat}(K_i x)$ is available such that the corresponding ellipsoidal invariant sets $E(P_i)$

$$E(P_i) = \left\{x \in \mathbb{R}^n : x^T P_i^{-1} x \le 1\right\} \tag{4.88}$$

are non-empty for $i = 1, 2, \ldots, s$. Recall that for all $x(k) \in E(P_i)$, it follows that $\text{sat}(K_i x) \in U$ and $x(k+1) = Ax(k) + B\,\text{sat}(K_i x(k)) \in X$. Denote $\Omega_E \subset \mathbb{R}^n$ as the convex hull of $E(P_i), i = 1, 2, \ldots, s$. It follows that $\Omega_E \subseteq X$, since X is convex and $E(P_i) \subseteq X, i = 1, 2, \ldots, s$.

Any state $x(k) \in \Omega_E$ can be decomposed as,

$$x(k) = \sum_{i=1}^{s} \lambda_i(k) \widehat{x}_i(k) \tag{4.89}$$

where $\widehat{x}_i(k) \in E(P_i)$ and $\lambda_i(k)$ are interpolating coefficients, that satisfy

$$\sum_{i=1}^{s} \lambda_i(k) = 1, \quad \lambda_i(k) \geq 0$$

Consider the following control law,

$$u(k) = \sum_{i=1}^{s} \lambda_i(k)\,\mathrm{sat}\big(K_i \widehat{x}_i(k)\big) \tag{4.90}$$

where $\mathrm{sat}(K_i \widehat{x}_i(k))$ is the saturated control law in $E(P_i)$.

Theorem 4.13 *The control law* (4.89), (4.90) *guarantees recursive feasibility for all initial conditions* $x(0) \in \Omega_E$.

Proof One has to prove that $u(k) \in U$ and $x(k+1) = Ax(k) + Bu(k) \in \Omega_E$ for all $x(k) \in \Omega_E$. For the input constraints, from equation (4.90) and since $\mathrm{sat}(K_i \widehat{x}_i(k)) \in U$, it follows that $u(k) \in U$.

For the state constraints, it holds that,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= A \sum_{i=1}^{s} \lambda_i(k)\widehat{x}_i(k) + B \sum_{i=1}^{s} \lambda_i(k)\,\mathrm{sat}(K_i \widehat{x}_i(k)) \\ &= \sum_{i=1}^{s} \lambda_i(k)(A\widehat{x}_i(k) + B\,\mathrm{sat}(K_i \widehat{x}_i(k))) \end{aligned}$$

One has $A\widehat{x}_i(k) + B\,\mathrm{sat}(K_i \widehat{x}_i(k)) \in E(P_i) \subseteq \Omega_E$, $i = 1, 2, \ldots, s$, which ultimately assures that $x(k+1) \in \Omega_E$. □

As in Sects. 4.5 and 4.6, the first high gain controller will be used for the performance, while the rest of available low gain controllers will be used to enlarge the domain of attraction. For a given current state x, consider the following optimization problem,

$$\lambda_i^* = \min_{\widehat{x}_i, \lambda_i} \left\{ \sum_{i=2}^{s} \lambda_i \right\} \tag{4.91}$$

subject to

$$\begin{cases} \widehat{x}_i^T P_i^{-1} \widehat{x}_i \leq 1, \quad \forall i = 1, 2, \ldots, s, \\ \sum_{i=1}^{s} \lambda_i \widehat{x}_i = x, \\ \sum_{i=1}^{s} \lambda_i = 1, \\ \lambda_i \geq 0, \quad \forall i = 1, 2, \ldots, s \end{cases}$$

Theorem 4.14 *The control law* (4.89), (4.90), (4.91) *guarantees asymptotic stability for all initial states* $x(0) \in \Omega_E$.

Proof Consider the following non-negative function,

$$V(x) = \sum_{i=2}^{s} \lambda_i^*(k) \tag{4.92}$$

for all $x \in \Omega_E \setminus E(P_1)$. $V(x)$ is a Lyapunov function candidate.

For any $x(k) \in \Omega_E \setminus E(P_1)$, by solving the optimization problem (4.91) and by applying (4.89), (4.90), one obtains

$$\begin{cases} x(k) = \sum_{i=1}^{s} \lambda_i^*(k) \widehat{x}_i^*(k) \\ u(k) = \sum_{i=1}^{s} \lambda_i^*(k) \operatorname{sat}(K_i \widehat{x}_i^*(k)) \end{cases}$$

It follows that,

$$\begin{aligned} x(k+1) = Ax(k) + Bu(k) &= A \sum_{i=1}^{s} \lambda_i^*(k) \widehat{x}_i^*(k) + B \sum_{i=1}^{s} \lambda_i^*(k) \operatorname{sat}(K_i \widehat{x}_i^*(k)) \\ &= \sum_{i=1}^{s} \lambda_i^*(k) \widehat{x}_i(k+1) \end{aligned}$$

where $\widehat{x}_i(k+1) = A\widehat{x}_i^*(k) + B \operatorname{sat}(K_i \widehat{x}_i^*(k)) \in E(P_i)$, $\forall i = 1, 2, \ldots, s$. Hence $\lambda_i^*(k)$, $\forall i = 1, 2, \ldots, s$ is a feasible solution of (4.91) at time $k+1$.

At time $k+1$, by soling the optimization problem (4.91), one obtains

$$x(k+1) = \sum_{i=1}^{s} \lambda_i^*(k+1) \widehat{x}_i^*(k+1)$$

where $\widehat{x}_i^*(k+1) \in E(P_i)$. It follows that $\sum_{i=2}^{s} \lambda_i^*(k+1) \le \sum_{i=2}^{s} \lambda_i^*(k)$ and $V(x)$ is a non-increasing function.

The contractive property of the ellipsoids $E(P_i), i = 1, 2, \ldots, s$ assures that there is no initial condition $x(0) \in \Omega_E \setminus E(P_1)$ such that $\sum_{i=2}^{s} \lambda_i^*(k+1) = \sum_{i=2}^{s} \lambda_i^*(k)$ for sufficiently large and finite k. It follows that $V(x) = \sum_{i=2}^{s} \lambda_i^*(k)$ is a Lyapunov function for all $x \in \Omega_E \setminus E(P_1)$.

The proof is completed by noting that inside $E(P_1)$, $\lambda_1 = 1$ and $\lambda_i = 0$, $i = 2, 3, \ldots, s$, the saturated controller $u = \text{sat}(K_1\widehat{x})$ is contractive and thus the control laws (4.89), (4.90), (4.91) assures asymptotic stability for all $x \in \Omega_E$. □

Denote $r_i = \lambda_i \widehat{x}_i$. Since $\widehat{x}_i \in E(P_i)$, it follows that $r_i \in \lambda_i E(P_i)$, and hence $r_i^T P_i^{-1} r_i \le \lambda_i^2$. The non-linear optimization problem (4.91) can be rewritten as,

$$\min_{r_i, \lambda_i} \left\{ \sum_{i=2}^{s} \lambda_i \right\} \tag{4.93}$$

subject to

$$\begin{cases} r_i^T P_i^{-1} r_i \le \lambda_i^2, \quad \forall i = 1, 2, \ldots, s, \\ \sum_{i=1}^{s} r_i = x, \\ \sum_{i=1}^{s} \lambda_i = 1, \quad \lambda_i \ge 0, \quad \forall i = 1, 2, \ldots, s \end{cases}$$

By using the Schur complements, (4.93) is converted into the following LMI problem,

$$\min_{r_i, \lambda_i} \left\{ \sum_{i=2}^{s} \lambda_i \right\} \tag{4.94}$$

subject to

$$\begin{cases} \begin{bmatrix} \lambda_i & r_i^T \\ r_i & \lambda_i P_i \end{bmatrix} \succeq 0, \quad \forall i = 1, 2, \ldots, s, \\ \sum_{i=1}^{s} r_i = x, \\ \sum_{i=1}^{s} \lambda_i = 1, \ \lambda_i \ge 0, \quad \forall i = 1, 2, \ldots, s \end{cases}$$

Algorithm 4.4 Interpolating control—Convex hull of ellipsoids

1. Measure the current state $x(k)$.
2. Solve the LMI problem (4.94).
3. Apply as input the control signal (4.90).
4. Wait for the next time instant $k := k + 1$.
5. Go to step 1 and repeat.

Remark 4.12 It is worth noticing that for all $x(k) \in E(P_1)$, the LMI problem (4.94) has the trivial solution,

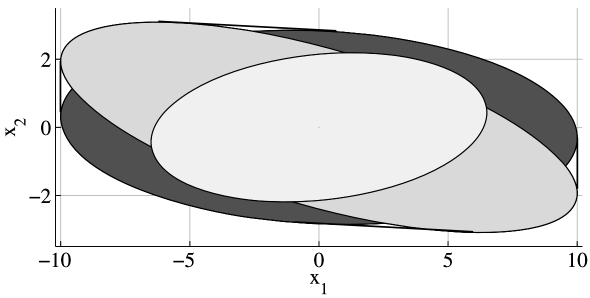
$$\lambda_i = 0, \quad \forall i = 2, 3, \ldots, s$$

Hence $\lambda_1 = 1$ and $x = \widehat{x}_1$. In this case, the interpolating controller becomes the saturated controller $u = \text{sat}(K_1 x)$.

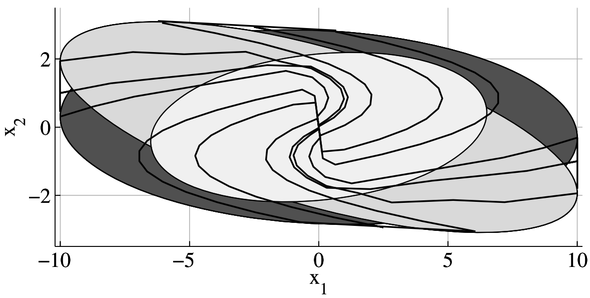
Example 4.6 Consider again the system in Example 4.1 with the same state and control constraints. Three gain matrices are chosen as,

$$\begin{cases} K_1 = [-0.9500 \quad -1.1137], \\ K_2 = [-0.4230 \quad -2.0607], \\ K_3 = [-0.5010 \quad -2.1340] \end{cases} \tag{4.95}$$

Fig. 4.24 Invariant ellipsoids and state trajectories of the closed loop system for Example 4.6

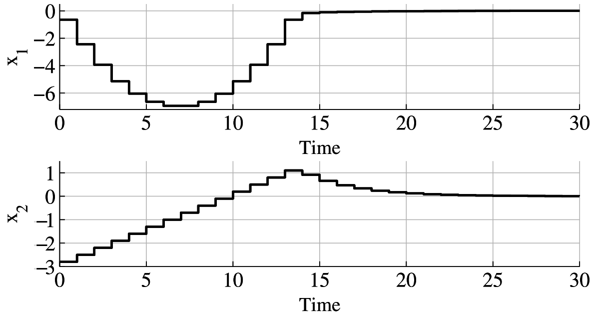


(a) Invariant ellipsoids

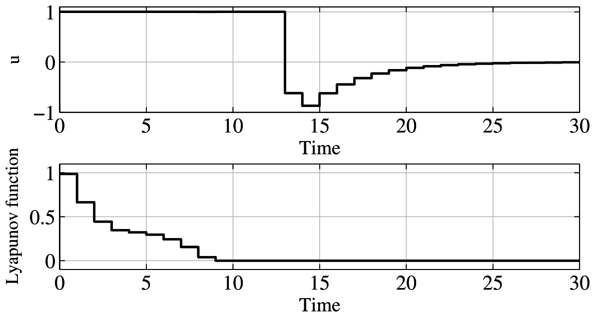


(b) State trajectories

Fig. 4.25 State trajectory, input trajectory and the sum $(\lambda_2^* + \lambda_3^*)$ of the closed loop system for Example 4.6



(a) State trajectory



(b) Input trajectory and $(\lambda_2^* + \lambda_3^*)$

By solving the LMI problem (2.55) three invariant ellipsoids $E(P_1)$, $E(P_2)$, $E(P_3)$ are computed corresponding to the saturated controllers $u = \text{sat}(K_1 x)$, $u = \text{sat}(K_2 x)$ and $u = \text{sat}(K_3 x)$. The sets $E(P_1)$, $E(P_2)$, $E(P_3)$ and their convex hull are depicted in Fig. 4.24(a). Figure 4.24(b) shows state trajectories for different initial conditions.

The matrices P_1, P_2 and P_3 are,

$$P_1 = \begin{bmatrix} 42.27 & 2.82 \\ 2.82 & 4.80 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 100.00 & -3.10 \\ -3.10 & 8.12 \end{bmatrix}, \qquad P_3 = \begin{bmatrix} 100.00 & -19.40 \\ -19.40 & 9.54 \end{bmatrix}$$

For the initial condition $x(0) = [-0.64 \ -2.8]^T$, using Algorithm 4.4, Fig. 4.25 presents the state and input trajectories and the sum $(\lambda_2^* + \lambda_3^*)$. As expected, the sum $(\lambda_2^* + \lambda_3^*)$, i.e. the Lyapunov function is positive and non-increasing.