6

Distributed Model Predictive Control

6.1 Introduction and Preliminary Results

In many large-scale control applications, it becomes convenient to break the large plantwide problem into a set of smaller and simpler subproblems in which the local inputs are used to regulate the local outputs. The overall plantwide control is then accomplished by the composite behavior of the interacting, local controllers. There are many ways to design the local controllers, some of which produce guaranteed properties of the overall plantwide system. We consider four control approaches in this chapter: decentralized, noncooperative, cooperative, and centralized control. The first three methods require the local controllers to optimize over only their local inputs. Their computational requirements are identical. The communication overhead is different, however. Decentralized control requires no communication between subsystems. Noncooperative and cooperative control require the input sequences and the current states or state estimates for all the other local subsystems. Centralized control solves the large, complex plantwide optimization over all the inputs. Communication is not a relevant property for centralized control because all information is available in the single plantwide controller. We use centralized control in this chapter to provide a benchmark of comparison for the distributed controllers.

We have established the basic properties of centralized MPC, both with and without state estimation, in Chapters 2, 3, and 5. In this chapter, we analyze some basic properties of the three distributed approaches: decentralized, noncooperative, and cooperative MPC. We show that the conditions required for closed-loop stability of decentralized control and noncooperative control are often violated for coupled multivariable systems under reasonable decompositions into subsystems. For ensuring closed-loop stability of a wide class of plantwide models and decomposition choices, cooperative control emerges as the most attractive option for distributed MPC. We then establish the closed-loop properties of cooperative MPC for unconstrained and constrained linear systems with and without state estimation. We also discuss current challenges facing this method, such as input constraints that are coupled between subsystems.

In our development of distributed MPC, we require some basic results on two topics: how to organize and solve the linear algebra of linear MPC, and how to ensure stability when using suboptimal MPC. We cover these two topics in the next sections, and then turn to the distributed MPC approaches.

6.1.1 Least Squares Solution

In comparing various forms of linear distributed MPC it proves convenient to see the MPC quadratic program for the sequence of states and inputs as a single large linear algebra problem. To develop this linear algebra problem, we consider first the *unconstrained* linear quadratic (LQ) problem of Chapter 1, which we solved efficiently with dynamic programming (DP) in Section 1.3.3

$$V(x(0), \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1}\big(x(k)'Qx(k) + u(k)'Ru(k)\big) + (1/2)x(N)'P_f x(N)$$

subject to

$$x^+ = Ax + Bu$$

In this section, we first take the direct but brute-force approach to finding the optimal control law. We write the model solution as

$$\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\mathcal{A}} x(0) + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\mathcal{B}} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} \tag{6.1}$$

or using the input and state sequences

$$\mathbf{x} = \mathcal{A}x(0) + \mathcal{B}\mathbf{u}$$

The objective function can be expressed as

$$V(x(0), \mathbf{u}) = (1/2)\left(x'(0)Qx(0) + \mathbf{x}'\mathcal{Q}\mathbf{x} + \mathbf{u}'\mathcal{R}\mathbf{u}\right)$$

in which

$$\mathcal{Q} = \operatorname{diag}\left(\begin{bmatrix} Q & Q & \ldots & P_f \end{bmatrix}\right) \in \mathbb{R}^{Nn \times Nn}$$
$$\mathcal{R} = \operatorname{diag}\left(\begin{bmatrix} R & R & \ldots & R \end{bmatrix}\right) \in \mathbb{R}^{Nm \times Nm} \tag{6.2}$$

Eliminating the state sequence. Substituting the model into the objective function and *eliminating* the state sequence gives a quadratic function of $\mathbf{u}$

$$V(x(0), \mathbf{u}) = (1/2)x'(0)(Q + \mathcal{A}'\mathcal{Q}\mathcal{A})x(0) + \mathbf{u}'(\mathcal{B}'\mathcal{Q}\mathcal{A})x(0) + (1/2)\mathbf{u}'(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})\mathbf{u} \tag{6.3}$$

and the optimal solution for the entire set of inputs is obtained in one shot

$$\mathbf{u}^0(x(0)) = -(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A}\, x(0)$$

and the optimal cost is

$$V^0(x(0)) = \left(\frac{1}{2}\right)x'(0)\left(Q + \mathcal{A}'\mathcal{Q}\mathcal{A} - \mathcal{A}'\mathcal{Q}\mathcal{B}(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A}\right)x(0)$$

If used explicitly, this procedure for computing $\mathbf{u}^0$ would be inefficient because $\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R}$ is an $(mN \times mN)$ matrix. Notice that in the DP formulation one has to invert instead an $(m \times m)$ matrix N times, which is computationally less expensive.[1] Notice also that unlike DP, the least squares approach provides *all* input moves as a function of the *initial* state, $x(0)$. The gain for the control law comes from the first input move in the sequence

$$K(0) = -\begin{bmatrix} I_m & 0 & \cdots & 0 \end{bmatrix}(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A}$$

It is not immediately clear that the $K(0)$ and V^0 given above from the least squares approach are equivalent to the result from the Riccati iteration, (1.10)–(1.14) of Chapter 1, but since we have solved the same optimization problem, the two results are the same.[2]

Retaining the state sequence. In this section we set up the least squares problem again, but with an eye toward improving its efficiency. Retaining the state sequence and adjoining the model equations as

[1]Would you prefer to invert by hand 100 (1×1) matrices or a single (100×100) dense matrix?

[2]Establishing this result directly is an exercise in using the partitioned matrix inversion formula. The next section provides another way to show they are equivalent.

equality constraints is a central idea in optimal control and is described in standard texts (Bryson and Ho, 1975, p. 44). We apply this standard approach here. Wright (1997) provides a discussion of this problem in the linear model MPC context and the extensions required for the quadratic programming problem when there are inequality constraints on the states and inputs. Including the state with the input in the sequence of unknowns, we define the enlarged vector $\mathbf{z}$ to be

$$\mathbf{z} = \begin{bmatrix} u(0) \\ x(1) \\ u(1) \\ x(2) \\ \vdots \\ u(N-1) \\ x(N) \end{bmatrix}$$

The objective function is

$$\min_{\mathbf{u}} (1/2)(x'(0)Qx(0) + \mathbf{z}'H\mathbf{z})$$

in which

$$H = \operatorname{diag}\left(\begin{bmatrix} R & Q & R & Q & \cdots & R & P_f \end{bmatrix}\right)$$

The constraints are

$$D\mathbf{z} = d$$

in which

$$D = -\begin{bmatrix} B & -I & & & & & \\ & A & B & -I & & & \\ & & & \ddots & & & \\ & & & & A & B & -I \end{bmatrix} \qquad d = \begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix} x(0)$$

We now substitute these results into (1.57) and obtain the linear algebra problem

$$\begin{bmatrix}
R & & & & & & & B' & & & \\
& Q & & & & & & -I & A' & & \\
& & R & & & & & & B' & & \\
& & & Q & & & & & -I & & \\
& & & & \ddots & & & & & \ddots & \\
& & & & & R & & & & & B' \\
& & & & & & P_f & & & & -I \\
B & -I & & & & & & & & & \\
& A & B & -I & & & & & & & \\
& & & & \ddots & & & & & & \\
& & & & & B & -I & & & &
\end{bmatrix}
\begin{bmatrix} u(0) \\ x(1) \\ u(1) \\ x(2) \\ \vdots \\ u(N-1) \\ x(N) \\ \lambda(1) \\ \lambda(2) \\ \vdots \\ \lambda(N) \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -A \\ 0 \\ \vdots \\ 0 \end{bmatrix} x(0)$$

Method	FLOPs
dynamic programming (DP)	Nm^3
dense least squares	N^3m^3
banded least squares	$N(2n+m)(3n+m)^2$

Table 6.1: Computational cost of solving finite horizon LQR problem.

This equation is rather cumbersome, but if we reorder the unknown vector to put the Lagrange multiplier together with the state and input from the same time index, and reorder the equations, we obtain the following banded matrix problem

$$\begin{bmatrix}
R & B' & & & & & & & & & \\
B & & -I & & & & & & & & \\
& -I & Q & & & & & & & & \\
& & & R & & & & & & & \\
& & & \ddots & \ddots & & & & & & \\
& & & & & R & B' & & & & \\
& & & & A & B & & -I & & & \\
& & & & & & -I & Q & & A' & \\
& & & & & & & & R & B' & \\
& & & & & & & A & B & & -I \\
& & & & & & & & & -I & P_f
\end{bmatrix}
\begin{bmatrix}
u(0) \\ \lambda(1) \\ x(1) \\ u(1) \\ \vdots \\ u(N-2) \\ \lambda(N-1) \\ x(N-1) \\ u(N-1) \\ \lambda(N) \\ x(N)
\end{bmatrix}
=
\begin{bmatrix}
0 \\ -A \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix} x(0) \tag{6.4}$$

The banded structure allows a more efficient solution procedure. The floating operation (FLOP) count for the factorization of a banded matrix is $O(LM^2)$ in which L is the dimension of the matrix and M is the bandwidth. This compares to the regular FLOP count of $O(L^3)$ for the factorization of a regular dense matrix. The bandwidth of the matrix in (6.4) is $3n+m$ and the dimension of the matrix is $N(2n+m)$. Therefore the FLOP count for solving this equation is $O(N(2n+m)(3n+m)^2)$. Notice that this approach reduces the N^3 dependence of the previous MPC solution method. That is the computational advantage provided by these adjoint methods for treating the model constraints. Table 6.1 summarizes the computational cost of the three approaches for the linear quadratic regulator (LQR) problem. As shown in the table, DP is highly efficient. When we add input and state inequality constraints to the control problem and the state dimension is large, however, we cannot conveniently apply DP. The dense least squares computational cost is high if we wish to compute a large number of moves in the horizon. Note the cost of dense least squares scales with the third power of horizon length N. As we have discussed in Chapter 2, considerations of control theory favor large N. Another factor increasing the computational cost is the trend in industrial MPC implementations to larger multivariable control problems with more states and inputs, i.e., larger m and n. Therefore, the adjoint approach using banded least squares method becomes important for industrial applications in which the problems are large and a solid theoretical foundation for the control method is desirable.

We might obtain more efficiency than the banded structure if we view (6.4) as a block tridiagonal matrix and use the method provided by Golub and Van Loan (1996, p. 174). The final fine tuning of the solution method for this class of problems is a topic of current research, but the important point is that, whatever final procedure is selected, the computational cost will be linear in N as in DP instead of cubic in N as in dense least squares.

Furthermore, if we wish to see the connection to the DP solution, we can proceed as follows. Substitute $\Pi(N) = P_f$ as in (1.11) of Chapter 1 and consider the last three-equation block of the matrix appearing in (6.4)

$$\begin{bmatrix} & R & B' & \\ A & B & & -I \\ & & -I & \Pi(N) \end{bmatrix} \begin{bmatrix} x(N-1) \\ u(N-1) \\ \lambda(N) \\ x(N) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We can eliminate this small set of equations and solve for $u(N-1)$, $\lambda(N)$, $x(N)$ in terms of $x(N-1)$, resulting in

$$\begin{bmatrix} u(N-1) \\ \lambda(N) \\ x(N) \end{bmatrix} = \begin{bmatrix} -(B'\Pi(N)B+R)^{-1}B'\Pi(N)A \\ \Pi(N)(I - B(B'\Pi(N)B+R)^{-1}B'\Pi(N))A \\ (I - B(B'\Pi(N)B+R)^{-1}B'\Pi(N))A \end{bmatrix} x(N-1)$$

Notice that in terms of the Riccati matrix, we also have the relationship

$$A'\lambda(N) = \Pi(N-1)x(N-1) - Qx(N-1)$$

We then proceed to the next to last block of three equations

$$\begin{bmatrix} & R & B' & & & \\ A & B & & -I & & \\ & & -I & Q & & A' \end{bmatrix} \begin{bmatrix} x(N-2) \\ u(N-2) \\ \lambda(N-1) \\ x(N-1) \\ u(N-1) \\ \lambda(N) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Note that the last equation gives

$$\lambda(N-1) = Qx(N-1) + A'\lambda(N) = \Pi(N-1)x(N-1)$$

Using this relationship and continuing on to solve for $x(N-1)$, $\lambda(N-1)$, $u(N-2)$ in terms of $x(N-2)$ gives

$$\begin{bmatrix} u(N-2) \\ \lambda(N-1) \\ x(N-1) \end{bmatrix} = \begin{bmatrix} -(B'\Pi(N-1)B+R)^{-1}B'\Pi(N-1)A \\ \Pi(N-1)(I-B(B'\Pi(N-1)B+R)^{-1}B'\Pi(N-1))A \\ (I-B(B'\Pi(N-1)B+R)^{-1}B'\Pi(N-1))A \end{bmatrix} x(N-2)$$

Continuing on through each previous block of three equations produces the Riccati iteration and feedback gains of (1.10)–(1.13). The other unknowns, the multipliers, are simply

$$\lambda(k) = \Pi(k)x(k) \qquad k = 1, 2, \ldots, N$$

so the cost to go at each stage is simply $x(k)'\lambda(k)$, and we see the nice connection between the Lagrange multipliers and the cost of the LQR control problem.

6.1.2 Stability of Suboptimal MPC

When using distributed MPC, it may be necessary or convenient to implement the control without solving the complete optimization. We then have a form of suboptimal MPC, which was first considered in Chapter 2, Section 2.7. Before adding the complexity of the distributed version, we wish to further develop a few features of suboptimal MPC in the centralized, single-player setting. These same features arise in the distributed, many-player setting as we discuss subsequently.

We consider a specific variation of suboptimal MPC in which a starting guess is available from the control trajectory at the previous time and we take a fixed number of steps of an optimization algorithm. The exact nature of the optimization method is not essential, but we do restrict the method so that each iteration is feasible and decreases the value of the cost function. To initialize the suboptimal controller, we are given an initial state $x(0) = x_0$, and we generate an initial control sequence $\mathbf{u}(0) = \mathbf{h}(x_0)$. We consider input constraints $u(i) \in \mathbb{U} \subseteq \mathbb{R}^m$, $i \in \mathbb{I}_{0:N-1}$, which we also write as $\mathbf{u} \in \mathbb{U}^N \subseteq \mathbb{R}^N$. As in Chapter 2 we denote the set of feasible states as $\mathcal{X}_N$. These are the states for which the initial control sequence $\mathbf{h}(x_0)$ is well defined. The suboptimal MPC algorithm is as follows.

Algorithm 6.1 (Suboptimal MPC (simplified))**.** Set current state $x = x_0$, current control sequence, $\mathbf{u} = \mathbf{h}(x_0)$, current warm start $\tilde{\mathbf{u}} = \mathbf{u}$. Then repeat

1. Obtain current measurement of state x.

2. The controller performs some number of iterations of a feasible path optimization algorithm to obtain an improved control sequence $\mathbf{u}$ such that $V_N(x, \mathbf{u}(0)) \leq V_N(x, \widetilde{\mathbf{u}}(0))$.

3. Inject the first element of the input sequence $\mathbf{u}$.

4. Compute the next warm start.

$$\widetilde{\mathbf{u}}^+ = (u(1), u(2), \ldots, u(N-1), 0)$$

This warm start is a simplified version of the one considered in Chapter 2, in which the final control move in the warm start was determined by the control law $\kappa_f(x)$. In distributed MPC it is simpler to use zero for the final control move in the warm start. We establish later in the chapter that the system cost function $V(x, \mathbf{u})$ satisfies the following properties for the form of suboptimal MPC generated by distributed MPC. There exist constants $a, b, c > 0$ such that

$$a\,|(x, \mathbf{u})|^2 \leq V(x, \mathbf{u}) \leq b\,|(x, \mathbf{u})|^2$$
$$V(x^+, \mathbf{u}^+) - V(x, \mathbf{u}) \leq -c\,|(x, u(0))|^2$$

These properties are similar to those required for a valid Lyapunov function. The difference is that the cost decrease here does not depend on the size of $\mathbf{u}$, but only x and the first element of $\mathbf{u}$, $u(0)$. This cost decrease is sufficient to establish that $x(k)$ and $u(k)$ converge to zero, but allows the possibility that $\mathbf{u}(k)$ is large even though $x(k)$ is small. That fact prevents us from establishing the solution $x(k) = 0$ for all k is Lyapunov stable. We can establish that the solution $x(k) = 0$ for all k is Lyapunov stable at $k = 0$ only. We cannot establish uniform Lyapunov stability nor Lyapunov stability for any $k > 0$. The problem is not that our proof technique is deficient. There is no reason to *expect* that the solution $x(k) = 0$ for all k is Lyapunov stable for suboptimal MPC. The lack of Lyapunov stability of $x(k) = 0$ for all k is a subtle issue and warrants some discussion. To make these matters more precise, consider the following standard definitions of Lyapunov stability at time k and uniform Lyapunov stability (Vidyasagar, 1993, p. 136).

Definition 6.2 (Lyapunov stability)**.** The zero solution $x(k) = 0$ for all k is stable (in the sense of Lyapunov) at $k = k_0$ if for any $\varepsilon > 0$ there exists a $\delta(k_0, \varepsilon) > 0$ such that

$$|x(k_0)| < \delta \implies |x(k)| < \varepsilon \quad \forall k \geq k_0 \tag{6.5}$$

Lyapunov stability is defined at a time k_0. Uniform stability is the concept that guarantees that the zero solution is not losing stability with time. For a uniformly stable zero solution, δ in Definition 6.2 is *not* a function of k_0, so that (6.5) holds for all k_0.

Definition 6.3 (Uniform Lyapunov stability)**.** The zero solution $x(k) = 0$ for all k is uniformly stable (in the sense of Lyapunov) if for any $\varepsilon > 0$ there exists a $\delta(\varepsilon) > 0$ such that

$$|x(k_0)| < \delta \implies |x(k)| < \varepsilon \quad \forall k \geq k_0 \quad \forall k_0$$

Exercise 6.6 gives an example of a linear system for which $x(k)$ converges exponentially to zero with increasing k for all $x(0)$, but the zero solution $x(k) = 0$ for all k is Lyapunov stable only at $k = 0$. It is not uniformly Lyapunov stable nor Lyapunov stable for any $k > 0$. Without further restrictions, suboptimal MPC admits this same type of behavior.

To ensure uniform Lyapunov stability, we add requirements to suboptimal MPC beyond obtaining only a cost decrease. Here we impose the constraint

$$|\mathbf{u}| \leq d\,|x| \qquad x \in r\mathcal{B}$$

in which $d, r > 0$. This type of constraint is also included somewhat indirectly by the suboptimal control approach discussed in Section 2.7. In that arrangement, this constraint is implied by the first case in (2.29), which leads to Proposition 2.44. For simplicity, in this chapter we instead include the constraint explicitly in the distributed MPC optimization problem. Both approaches provide (uniform) Lyapunov stability of the solution $x(k) = 0$ for all k.

The following lemma summarizes the conditions we use later in the chapter for establishing exponential stability of distributed MPC. A similar lemma establishing asymptotic stability of suboptimal MPC was given by Scokaert, Mayne, and Rawlings (1999) (Theorem 1).

First we recall the definition of exponential stability.

Definition 6.4 (Exponential stability)**.** Let $\mathbb{X}$ be positive invariant set for $x^+ = f(x)$. Then the origin is exponentially stable in $\mathbb{X}$ for $x^+ = f(x)$ if there exists $c > 0$ and $0 < \gamma < 1$ such that for each $x \in \mathbb{X}$

$$|\phi(i;x)| \leq c\,|x|\,\gamma^i$$

for all $i \geq \mathbb{I}_{\geq 0}$.

Consider next the suboptimal MPC controller. Let the system satisfy $(x^+, \mathbf{u}^+) = (f(x, \mathbf{u}), g(x, \mathbf{u}))$ with initial sequence $\mathbf{u}(0) = \mathbf{h}(x(0))$. The controller constraints are $x(i) \in \mathbb{X} \subseteq \mathbb{R}^n$ for all $i \in \mathbb{I}_{0:N}$ and $u(i) \in \mathbb{U} \subseteq \mathbb{R}^m$ for all $i \in \mathbb{I}_{0:N-1}$. Let $\mathcal{X}_N$ denote the set of states for which the MPC controller is feasible.

Lemma 6.5 (Exponential stability of suboptimal MPC)**.** *Assume that the suboptimal MPC system satisfies the following inequalities with $r, a, b, c > 0$*

$$
\begin{aligned}
a\,|(x,\mathbf{u})|^2 \le V(x,\mathbf{u}) &\le b\,|(x,\mathbf{u})|^2 && x \in \mathcal{X}_N \quad \mathbf{u} \in \mathbb{U}^N \\
V(x^+,\mathbf{u}^+) - V(x,\mathbf{u}) &\le -c\,|(x, u(0))|^2 && x \in \mathcal{X}_N \quad \mathbf{u} \in \mathbb{U}^N \\
|\mathbf{u}| &\le d\,|x| && x \in r\mathcal{B}
\end{aligned}
$$

Then the origin is exponentially stable for the closed-loop system under suboptimal MPC with region of attraction $\mathcal{X}_N$ if either of the following additional assumptions holds

(a) $\mathbb{U}$ is compact. In this case, $\mathcal{X}_N$ may be unbounded.

(b) $\mathcal{X}_N$ is compact. In this case $\mathbb{U}$ may be unbounded.

Proof. First we show that the origin of the extended state $(x, \mathbf{u})$ is exponentially stable for $x(0) \in \mathcal{X}_N$.

(a) For the case $\mathbb{U}$ compact, we have $|\mathbf{u}| \le d\,|x|\,, x \in r\mathcal{B}$. Consider the optimization

$$\max_{\mathbf{u} \in \mathbb{U}^N} |\mathbf{u}| = s > 0$$

The solution exists by the Weierstrass theorem since $\mathbb{U}$ is compact, which implies $\mathbb{U}^N$ is compact. Then we have $|\mathbf{u}| \le (s/r)\,|x|$ for $x \in \mathcal{X}_N \setminus r\mathcal{B}$, so we have $|\mathbf{u}| \le d'\,|x|$ for $x \in \mathcal{X}_N$ in which $d' = \max(d, s/r)$.

(b) For the case $\mathcal{X}_N$ compact, consider the optimization

$$\max_{x \in \mathcal{X}_N} V(x, \mathbf{h}(x)) = \bar{V} > 0$$

The solution exists because $\mathcal{X}_N$ is compact and $\mathbf{h}(\cdot)$ and $V(\cdot)$ are continuous. Define the compact set $\bar{\mathbb{U}}$ by

$$\bar{\mathbb{U}} = \{\mathbf{u} \mid V(x, \mathbf{u}) \le \bar{V}, \quad x \in \mathcal{X}_N\}$$

The set is bounded because $V(x, \mathbf{u}) \ge a\,|(x, \mathbf{u})|^2 \ge a\,|\mathbf{u}|^2$. The set is closed because V is continuous. The significance of this set is that for all $k \geq 0$ and all $x \in \mathcal{X}_N$, $\mathbf{u}(k) \in \bar{\mathbb{U}}$. Therefore we have established that $\mathcal{X}_N$ compact implies $\mathbf{u}(k)$ evolves in a compact set as in the previous case when $\mathbb{U}$ is assumed compact. Using the same argument as in that case, we have established that there exists $d' > 0$ such that $|\mathbf{u}| \leq d' |x|$ for all $x \in \mathcal{X}_N$.

For the two cases, we therefore have established for all $x \in \mathcal{X}_N$, $\mathbf{u} \in \mathbb{U}^N$ (case (a)) or $\mathbf{u} \in \bar{\mathbb{U}}$ (case (b))

$$|(x, \mathbf{u})| \leq |x| + |\mathbf{u}| \leq |x| + d' |x| \leq (1 + d') |x|$$

which gives $|x| \geq c' |(x, \mathbf{u})|$ with $c' = 1/(1 + d') > 0$. Hence, there exists $a_3 = c(c')^2$ such that $V(x^+, \mathbf{u}^+) - V(x, \mathbf{u}) \leq -a_3 |(x, \mathbf{u})|^2$ for all $x \in \mathcal{X}_N$. Therefore the extended state $(x, \mathbf{u})$ satisfies the standard conditions of an exponential stability Lyapunov function (see Theorem B.19 in Appendix B) with $a_1 = a, a_2 = b, a_3 = c(c')^2, \sigma = 2$ for $(x, \mathbf{u}) \in \mathcal{X}_N \times \mathbb{U}^N$ (case (a)) or $\mathcal{X}_N \times \bar{\mathbb{U}}$ (case (b)). Therefore for all $x(0) \in \mathcal{X}_N$, $k \geq 0$

$$|(x(k), \mathbf{u}(k))| \leq \alpha |(x(0), \mathbf{u}(0))| \gamma^k$$

in which $\alpha > 0$ and $0 < \gamma < 1$.

Finally we remove the input sequence and establish that the origin for the state (rather than the extended state) is exponentially stable for the closed-loop system. We have for all $x(0) \in \mathcal{X}_N$ and $k \geq 0$

$$\begin{aligned} |x(k)| &\leq |(x(k), \mathbf{u}(k))| \leq \alpha |(x(0), \mathbf{u}(0))| \gamma^k \\ &\leq \alpha(|x(0)| + |\mathbf{u}(0)|)\gamma^k \leq \alpha(1 + d') |x(0)| \gamma^k \\ &\leq \alpha' |x(0)| \gamma^k \end{aligned}$$

in which $\alpha' = \alpha(1 + d') > 0$, and we have established exponential stability of the origin on the feasible set $\mathcal{X}_N$. ■

Exercises 6.7 and 6.8 explore what to conclude about exponential stability when both $\mathbb{U}$ and $\mathcal{X}_N$ are unbounded.

We also consider later in the chapter the effects of state estimation error on the closed-loop properties of distributed MPC. For analyzing stability under perturbations, the following lemma is useful. Here e plays the role of estimation error.

Lemma 6.6 (Global asymptotic stability and exponential convergence with mixed powers of norm)**.** *Consider a dynamic system*

$$(x^+, e^+) = f(x, e)$$

with a zero steady-state solution, $f(0,0) = (0,0)$. Assume there exists a function $V : \mathbb{R}^{n+m} \to \mathbb{R}_{\geq 0}$ that satisfies the following for all $(x,e) \in \mathbb{R}^n \times \mathbb{R}^m$

$$a(|x|^\sigma + |e|^\gamma) \leq V((x,e)) \leq b(|x|^\sigma + |e|^\gamma) \tag{6.6}$$

$$V(f(x,e)) - V((x,e)) \leq -c(|x|^\sigma + |e|^\gamma) \tag{6.7}$$

with constants $a, b, c, \sigma, \gamma > 0$. Then the following holds for all $(x(0), e(0))$ and $k \in \mathbb{I}_{\geq 0}$

$$|x(k), e(k)| \leq \delta(|x(0), e(0)|)\lambda^k$$

with $\lambda < 1$ and $\delta(\cdot) \in \mathcal{K}_\infty$.

The proof of this lemma is discussed in Exercise 6.9. We also require a converse theorem for exponential stability.

Lemma 6.7 (Converse theorem for exponential stability)**.** *If the zero steady-state solution of $x^+ = f(x)$ is globally exponentially stable, then there exists Lipschitz continuous $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ that satisfies the following: there exist constants $a, b, c, \sigma > 0$, such that for all $x \in \mathbb{R}^n$*

$$a\,|x|^\sigma \leq V(x) \leq b\,|x|^\sigma$$

$$V(f(x)) - V(x) \leq -c\,|x|^\sigma$$

Moreover, any $\sigma > 0$ is valid, and the constant c can be chosen as large as one wishes.

The proof of this lemma is discussed in Exercise B.3.

6.2 Unconstrained Two-Player Game

To introduce clearly the concepts and notation required to analyze distributed MPC, we start with a two-player game. We then generalize to an M-player game in the next section.

Let (A_{11}, B_{11}, C_{11}) be a minimal state space realization of the (u_1, y_1) input-output pair. Similarly, let (A_{12}, B_{12}, C_{12}) be a minimal state space realization of the (u_2, y_1) input-output pair. The dimensions are $u_1 \in \mathbb{R}^{m_1}$, $y_1 \in \mathbb{R}^{p_1}$, $x_{11} \in \mathbb{R}^{n_{11}}$, $x_{12} \in \mathbb{R}^{n_{12}}$ with $n_1 = n_{11} + n_{12}$. Output y_1 can then be represented as the following, possibly nonminimal, state space model

$$\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}^+ = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{12} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ B_{12} \end{bmatrix} u_2$$

$$y_1 = \begin{bmatrix} C_{11} & C_{12} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$$

Proceeding in an analogous fashion with output y_2 and inputs u_1 and u_2, we model y_2 with the following state space model

$$\begin{bmatrix} x_{22} \\ x_{21} \end{bmatrix}^+ = \begin{bmatrix} A_{22} & 0 \\ 0 & A_{21} \end{bmatrix} \begin{bmatrix} x_{22} \\ x_{21} \end{bmatrix} + \begin{bmatrix} B_{22} \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ B_{21} \end{bmatrix} u_1$$

$$y_2 = \begin{bmatrix} C_{22} & C_{21} \end{bmatrix} \begin{bmatrix} x_{22} \\ x_{21} \end{bmatrix}$$

We next define player one's local cost functions

$$V_1(x_1(0), \mathbf{u}_1, \mathbf{u}_2) = \sum_{k=0}^{N-1} \ell_1(x_1(k), u_1(k)) + V_{1f}(x_1(N))$$

in which

$$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$$

Note that the first local objective is affected by the second player's inputs through the model evolution of x_1, i.e., through the x_{12} states. We choose the stage cost to account for the first player's inputs and outputs

$$\ell_1(x_1, u_1) = (1/2)(y_1' \overline{Q}_1 y_1 + u_1' R_1 u_1)$$
$$\ell_1(x_1, u_1) = (1/2)(x_1' Q_1 x_1 + u_1' R_1 u_1)$$

in which

$$Q_1 = C_1' \overline{Q}_1 C_1 \qquad C_1 = \begin{bmatrix} C_{11} & C_{12} \end{bmatrix}$$

Motivated by the warm start to be described later, for stable systems, we choose the terminal penalty to be the infinite horizon cost to go under zero control

$$V_{1f}(x_1(N)) = (1/2) x_1'(N) P_{1f} x_1(N)$$

We choose P_{1f} as the solution to the following Lyapunov equation assuming A_1 is stable

$$A_1' P_{1f} A_1 - P_{1f} = -Q_1 \tag{6.8}$$

We proceed analogously to define player two's local objective function and penalties

$$V_2(x_2(0), \mathbf{u}_1, \mathbf{u}_2) = \sum_{k=0}^{N-1} \ell_2(x_2(k), u_2(k)) + V_{2f}(x_2(N))$$

In centralized control and the cooperative game, the two players share a common objective, which can be considered to be the overall plant objective

$$V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) = \rho_1 V_1(x_1(0), \mathbf{u}_1, \mathbf{u}_2) + \rho_2 V_2(x_2(0), \mathbf{u}_2, \mathbf{u}_1)$$

in which the parameters ρ_1, ρ_2 are used to specify the relative weights of the two subsystems in the overall plant objective. Their values are restricted so $\rho_1, \rho_2 > 0$, $\rho_1 + \rho_2 = 1$ so that both local objectives must have some nonzero effect on the overall plant objective.

6.2.1 Centralized Control

Centralized control requires the solution of the systemwide control problem. It can be stated as

$$\begin{aligned}
&\min_{\mathbf{u}_1, \mathbf{u}_2} V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) \\
&\text{s.t. } x_1^+ = A_1 x_1 + \overline{B}_{11} u_1 + \overline{B}_{12} u_2 \\
&\qquad x_2^+ = A_2 x_2 + \overline{B}_{22} u_2 + \overline{B}_{21} u_1
\end{aligned}$$

in which

$$A_1 = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{12} \end{bmatrix} \qquad A_2 = \begin{bmatrix} A_{22} & 0 \\ 0 & A_{21} \end{bmatrix}$$

$$\overline{B}_{11} = \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} \qquad \overline{B}_{12} = \begin{bmatrix} 0 \\ B_{12} \end{bmatrix} \qquad \overline{B}_{21} = \begin{bmatrix} 0 \\ B_{21} \end{bmatrix} \qquad \overline{B}_{22} = \begin{bmatrix} B_{22} \\ 0 \end{bmatrix}$$

This optimal control problem is more complex than all of the distributed cases to follow because the decision variables include both $\mathbf{u}_1$ and $\mathbf{u}_2$. Because the performance is optimal, centralized control is a natural benchmark against which to compare the distributed cases: cooperative, noncooperative, and decentralized MPC. The plantwide stage cost and terminal cost can be expressed as quadratic functions of the subsystem states and inputs

$$\begin{aligned}
\ell(x, u) &= (1/2)(x'Qx + u'Ru) \\
V_f(x) &= (1/2)x'P_f x
\end{aligned}$$

in which

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad Q = \begin{bmatrix} \rho_1 Q_1 & 0 \\ 0 & \rho_2 Q_2 \end{bmatrix}$$

$$R = \begin{bmatrix} \rho_1 R_1 & 0 \\ 0 & \rho_2 R_2 \end{bmatrix} \qquad P_f = \begin{bmatrix} \rho_1 P_{1f} & 0 \\ 0 & \rho_2 P_{2f} \end{bmatrix} \tag{6.9}$$

and we have the standard MPC problem considered in Chapters 1 and 2

$$\begin{aligned} &\min_{\mathbf{u}} V(x(0), \mathbf{u}) \\ &\text{s.t. } x^+ = Ax + Bu \end{aligned} \tag{6.10}$$

in which

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \qquad B = \begin{bmatrix} \overline{B}_{11} & \overline{B}_{12} \\ \overline{B}_{21} & \overline{B}_{22} \end{bmatrix} \tag{6.11}$$

Given the terminal penalty in (6.8), stability of the closed-loop centralized system is guaranteed for all choices of system models and tuning parameters subject to the usual stabilizability assumption on the system model.

6.2.2 Decentralized Control

Centralized and decentralized control define the two extremes in distributing the decision making in a large-scale system. Centralized control has full information and optimizes the full control problem over all decision variables. Decentralized control, on the other hand, optimizes only the local objectives and has no information about the actions of the other subsystems. Player one's objective function is

$$V_1(x_1(0), \mathbf{u}_1) = \sum_{k=0}^{N-1} \ell_1(x_1(k), u_1(k)) + V_{1f}(x_1(N))$$

We then have player one's decentralized control problem

$$\begin{aligned} &\min_{\mathbf{u}_1} V_1(x_1(0), \mathbf{u}_1) \\ &\text{s.t. } x_1^+ = A_1 x_1 + \overline{B}_{11} u_1 \end{aligned}$$

We know the optimal solution for this kind of LQ problem is a linear feedback law

$$u_1^0 = K_1 x_1(0)$$

Notice that in decentralized control, player one's model does not account for the inputs of player two, and already contains model error. In the decentralized problem, player one requires no information about player two. The communication overhead for decentralized control is therefore minimal, which is an implementation advantage, but the resulting performance may be quite poor for systems with reasonably strong coupling. We compute an optimal K_1 for system one $(A_1, \overline{B}_{11}, Q_1, R_1)$ and optimal K_2 for system 2. The closed-loop system evolution is then

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 + \overline{B}_{11}K_1 & \overline{B}_{12}K_2 \\ \overline{B}_{21}K_1 & A_2 + \overline{B}_{22}K_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and we know only that $A_{11} + \overline{B}_{11}K_1$ and $A_{22} + \overline{B}_{22}K_2$ are stable matrices. Obviously the stability of the closed-loop, decentralized system is fragile and depends in a sensitive way on the sizes of the interaction terms $\overline{B}_{12}$ and $\overline{B}_{21}$ and feedback gains K_1, K_2.

6.2.3 Noncooperative Game

In the noncooperative game, player one optimizes $V_1(x_1(0), \mathbf{u}_1, \mathbf{u}_2)$ over $\mathbf{u}_1$ and player two optimizes $V_2(x_2(0), \mathbf{u}_1, \mathbf{u}_2)$ over $\mathbf{u}_2$. From player one's perspective, player two's planned inputs $\mathbf{u}_2$ are known disturbances affecting player one's output through the dynamic model. Part of player one's optimal control problem is therefore to compensate for player two's inputs with his optimal $\mathbf{u}_1$ sequence in order to optimize his local objective V_1. Similarly, player two considers player one's inputs as a known disturbance and solves an optimal control problem that removes their effect in his local objective V_2. Because this game is noncooperative $(V_1 \neq V_2)$, the struggle between players one and two can produce an outcome that is bad for both of them as we show subsequently. Notice that unlike decentralized control, there is no model error in the noncooperative game. Player one knows exactly the effect of the actions of player two and vice versa. Any poor nominal performance is caused by the noncooperative game, not model error.

Summarizing the noncooperative control problem statement, player one's model is

$$x_1^+ = A_1 x_1 + \overline{B}_{11} u_1 + \overline{B}_{12} u_2$$

and player one's objective function is

$$V_1(x_1(0), \mathbf{u}_1, \mathbf{u}_2) = \sum_{k=0}^{N-1} \ell_1(x_1(k), u_1(k)) + V_{1f}(x_1(N))$$

Note that V_1 here depends on $\mathbf{u}_2$ because the state trajectory $x_1(k)$, $k \geq 1$ depends on $\mathbf{u}_2$ as shown in player one's dynamic model. We then have player one's noncooperative control problem

$$\min_{\mathbf{u}_1} V_1(x_1(0), \mathbf{u}_1, \mathbf{u}_2)$$
$$\text{s.t. } x_1^+ = A_1 x_1 + \overline{B}_{11} u_1 + \overline{B}_{12} u_2$$

Solution to player one's optimal control problem. We now solve player one's optimal control problem. Proceeding as in Section 6.1.1 we define

$$\mathbf{z} = \begin{bmatrix} u_1(0) \\ x_1(1) \\ \vdots \\ u_1(N-1) \\ x_1(N) \end{bmatrix} \qquad H = \operatorname{diag}\left(\begin{bmatrix} R_1 & Q_1 & \cdots & R_1 & P_{1f} \end{bmatrix}\right)$$

and can express player one's optimal control problem as

$$\min_{\mathbf{z}} (1/2)(\mathbf{z}'H\mathbf{z} + x_1(0)'Q_1x_1(0))$$
$$\text{s.t. } D\mathbf{z} = d$$

in which

$$D = -\begin{bmatrix} \overline{B}_{11} & -I & & & & \\ & A_1 & \overline{B}_{11} & -I & & \\ & & & \ddots & & \\ & & & A_1 & \overline{B}_{11} & -I \end{bmatrix}$$

$$d = \begin{bmatrix} A_1x_1(0) + \overline{B}_{12}u_2(0) \\ \overline{B}_{12}u_2(1) \\ \vdots \\ \overline{B}_{12}u_2(N-1) \end{bmatrix}$$

We then apply (1.57) to obtain

$$\begin{bmatrix} H & -D' \\ -D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ -\widetilde{A}_1 \end{bmatrix} x_1(0) + \begin{bmatrix} 0 \\ -\widetilde{B}_{12} \end{bmatrix} \mathbf{u}_2 \tag{6.12}$$

in which we have defined

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda(1) \\ \lambda(2) \\ \vdots \\ \lambda(N) \end{bmatrix} \qquad \widetilde{A}_1 = \begin{bmatrix} A_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \widetilde{B}_{12} = \begin{bmatrix} \overline{B}_{12} & & & \\ & \overline{B}_{12} & & \\ & & \ddots & \\ & & & \overline{B}_{12} \end{bmatrix}$$

Solving this equation and picking out the rows of $\mathbf{z}$ corresponding to the elements of $\mathbf{u}_1$ gives

$$\mathbf{u}_1^0 = K_1x_1(0) + L_1\mathbf{u}_2$$

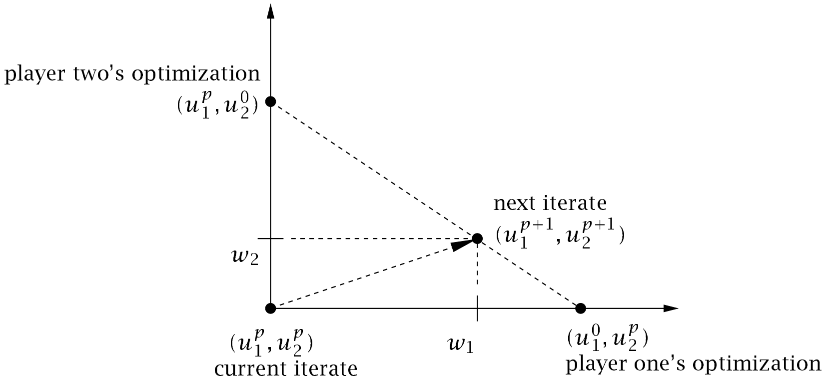

Figure 6.1: Convex step from (u_1^p, u_2^p) to (u_1^{p+1}, u_2^{p+1}); the parameters w_1, w_2 with $w_1 + w_2 = 1$ determine location of next iterate on line joining the two players' optimizations: (u_1^0, u_2^p) and (u_1^p, u_2^0).

and we see player one's optimal decision depends linearly on his initial state, but also on *player two's decision.* This is the key difference between decentralized control and noncooperative control. In noncooperative control, player two's decisions are communicated to player one and player one accounts for them in optimizing the local objective.

Convex step. Let $p \in \mathbb{I}_{\geq 0}$ denote the integer-valued iteration in the optimization problem. Looking ahead to the M-player game, we do not take the full step, but a convex combination of the the current optimal solution, $\mathbf{u}_1^0$, and the current iterate, $\mathbf{u}_1^p$

$$\mathbf{u}_1^{p+1} = w_1 \mathbf{u}_1^0 + (1 - w_1)\mathbf{u}_1^p \qquad 0 < w_1 < 1$$

This iteration is displayed in Figure 6.1. Notice we have chosen a distributed optimization of the Gauss-Jacobi type (see Bertsekas and Tsitsiklis, 1997, pp.219–223).

We place restrictions on the systems under consideration before analyzing stability of the controller.

Assumption 6.8 (Unconstrained two-player game)**.**

(a) All subsystems, $A_{ij}, i = 1, 2, j = 1, 2$, are stable.

(b) The controller penalties Q_1, Q_2, R_1, R_2 are positive definite.

The assumption of stable models is purely for convenience of exposition. We treat unstable, stabilizable systems in Section 6.3.

Convergence of the players' iteration. To understand the convergence of the two players' iterations, we express both players' moves as follows

$$\begin{aligned}
\mathbf{u}_1^{p+1} &= w_1 \mathbf{u}_1^0 + (1 - w_1)\mathbf{u}_1^p \\
\mathbf{u}_2^{p+1} &= w_2 \mathbf{u}_2^0 + (1 - w_2)\mathbf{u}_2^p \\
1 &= w_1 + w_2 \qquad 0 < w_1, w_2 < 1
\end{aligned}$$

or

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^{p+1} = \begin{bmatrix} w_1 I & 0 \\ 0 & w_2 I \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^0 \\ \mathbf{u}_2^0 \end{bmatrix} + \begin{bmatrix} (1-w_1)I & 0 \\ 0 & (1-w_2)I \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^p$$

The optimal control for each player is

$$\begin{bmatrix} \mathbf{u}_1^0 \\ \mathbf{u}_2^0 \end{bmatrix} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} 0 & L_1 \\ L_2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^p$$

Substituting the optimal control into the iteration gives

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1 K_1 & 0 \\ 0 & w_2 K_2 \end{bmatrix}}_{\overline{K}} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \underbrace{\begin{bmatrix} (1-w_1)I & w_1 L_1 \\ w_2 L_2 & (1-w_2)I \end{bmatrix}}_{L} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^p$$

Finally writing this equation in the plantwide notation, we express the iteration as

$$\mathbf{u}^{p+1} = \overline{K}x(0) + L\mathbf{u}^p$$

The convergence of the two players' control iteration is governed by the eigenvalues of L. If L is stable, the control sequence converges to

$$\mathbf{u}^\infty = (I - L)^{-1}\overline{K}x(0) \qquad |\lambda| < 1 \text{ for } \lambda \in \operatorname{eig}(L)$$

in which

$$\begin{aligned}
(I - L)^{-1}\overline{K} &= \begin{bmatrix} w_1 I & -w_1 L_1 \\ -w_2 L_2 & w_2 I \end{bmatrix}^{-1} \begin{bmatrix} w_1 K_1 & 0 \\ 0 & w_2 K_2 \end{bmatrix} \\
(I - L)^{-1}\overline{K} &= \begin{bmatrix} I & -L_1 \\ -L_2 & I \end{bmatrix}^{-1} \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}
\end{aligned}$$

Note that the weights w_1, w_2 do not appear in the converged input sequence. The $\mathbf{u}_1^\infty$, $\mathbf{u}_2^\infty$ pair have the equilibrium property that neither player can improve his position given the other player's current decision. This point is called a Nash equilibrium (Başar and Olsder, 1999, p. 4). Notice that the distributed MPC game does not have a Nash equilibrium if the eigenvalues of L are on or outside the unit circle. If the controllers have sufficient time during the control system's sample time to iterate to convergence, then the effect of the initial control sequence is removed by using the converged control sequence. If the iteration has to be stopped before convergence, the solution is

$$\mathbf{u}^{p+1} = L^p \mathbf{u}^{[0]} + \sum_{j=0}^{p-1} L^j \overline{K} x(0) \qquad 0 \leq p$$

in which $\mathbf{u}^{[0]}$ is the $p = 0$ (initial) input sequence. We use the brackets with $p = 0$ to distinguish this initial input sequence from an optimal input sequence.

Stability of the closed-loop system. We assume the Nash equilibrium is stable and there is sufficient computation time to iterate to convergence.

We require a matrix of zeros and ones to select the first move from the input sequence for injection into the plant. For the first player, the required matrix is

$$\begin{aligned} u_1(0) &= E_1 \mathbf{u}_1 \\ E_1 &= \begin{bmatrix} I_{m_1} & 0_{m_1} & \ldots & 0_{m_1} \end{bmatrix} \qquad m_1 \times m_1 N \text{ matrix} \end{aligned}$$

The closed-loop system is then

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \underbrace{\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \\ \underbrace{\begin{bmatrix} \overline{B}_{11} & \overline{B}_{12} \\ \overline{B}_{21} & \overline{B}_{22} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix} \begin{bmatrix} I & -L_1 \\ -L_2 & I \end{bmatrix}^{-1} \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}}_{K} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Using the plantwide notation for this equation and defining the feedback gain K gives

$$x^+ = (A + BK)x$$

The stability of the closed loop with converged, noncooperative control is therefore determined by the eigenvalues of $(A + BK)$.

We next present three simple examples to show that (i) the Nash equilibrium may not be stable (L is unstable), (ii) the Nash equilibrium may be stable but the closed loop is unstable (L is stable, $A + BK$ is unstable), and (iii) the Nash equilibrium may be stable and the closed loop is stable (L is stable, $A + BK$ is stable). Which situation arises depends in a nonobvious way on all of the problem data: A_1, A_2, $\overline{B}_{11}$, $\overline{B}_{12}$, $\overline{B}_{21}$, $\overline{B}_{22}$, Q_1, Q_2, P_{1f}, P_{2f}, R_1, R_2, w_1, w_2, N. One has to examine the eigenvalues of L and $A+BK$ for each application of interest to know how the noncooperative distributed MPC is going to perform. Even for a fixed dynamic model, when changing tuning parameters such as Q, P_f, R, w, one has to examine eigenvalues of L and $A + BK$ to know the effect on the closed-loop system. This is the main drawback of the noncooperative game. In many control system design methods, such as all forms of MPC presented in Chapter 2, closed-loop properties such as exponential stability are guaranteed for the *nominal* system for all choices of performance tuning parameters. Noncooperative distributed MPC does not have this feature and a stability analysis is required. We show in the next section that cooperative MPC does not suffer from this drawback, at the cost of slightly more information exchange.

Example 6.9: Nash equilibrium is unstable

Consider the following transfer function matrix for a simple two-input two-output system

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

in which

$$G(s) = \begin{bmatrix} \dfrac{1}{s^2 + 2(0.2)s + 1} & \dfrac{0.5}{0.225s + 1} \\ \dfrac{-0.5}{(0.5s + 1)(0.25s + 1)} & \dfrac{1.5}{0.75s^2 + 2(0.8)(0.75)s + 1} \end{bmatrix}$$

Obtain discrete time models (A_{ij}, B_{ij}, C_{ij}) for each of the four transfer functions $G_{ij}(s)$ using a sample time of $T = 0.2$ and zero-order holds on the inputs. Set the control cost function parameters to be

$$\overline{Q}_1 = \overline{Q}_2 = 1 \qquad \overline{P}_{1f} = \overline{P}_{2f} = 0 \qquad R_1 = R_2 = 0.01$$
$$N = 30 \qquad w_1 = w_2 = 0.5$$

Compute the eigenvalues of the L matrix for this system using noncooperative MPC. Show the Nash equilibrium is unstable and the closed-loop system is therefore unstable. Discuss why this system is problematic for noncooperative control.

Solution

For this problem L is a 60×60 matrix $(N(m_1 + m_2))$. The magnitudes of the largest eigenvalues are

$$\left|\operatorname{eig}(L)\right| = \begin{bmatrix} 1.11 & 1.11 & 1.03 & 1.03 & 0.914 & 0.914 & \cdots \end{bmatrix}$$

The noncooperative iteration does not converge. The steady-state gains for this system are

$$G(0) = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1.5 \end{bmatrix}$$

and we see that the diagonal elements are reasonably large compared to the nondiagonal elements. So the *steady-state* coupling between the two systems is relatively weak. The dynamic coupling is unfavorable, however. The response of y_1 to u_2 is more than four times faster than the response of y_1 to u_1. The faster input is the disturbance and the slower input is used for control. Likewise the response of y_2 to u_1 is three times faster than the response of y_2 to u_2. Also in the second loop, the faster input is the disturbance and the slower input is used for control. These pairings are unfavorable dynamically, and that fact is revealed in the instability of L and lack of a Nash equilibrium for the noncooperative dynamic regulation problem. □

Example 6.10: Nash equilibrium is stable but closed loop is unstable

Switch the outputs for the previous example and compute the eigenvalues of L and $(A+BK)$ for the noncooperative distributed MPC regulator for the system

$$G(s) = \begin{bmatrix} \dfrac{-0.5}{(0.5s+1)(0.25s+1)} & \dfrac{1.5}{0.75s^2 + 2(0.8)(0.75)s + 1} \\ \dfrac{1}{s^2 + 2(0.2)s + 1} & \dfrac{0.5}{0.225s + 1} \end{bmatrix}$$

Show in this case that the Nash equilibrium is stable, but the noncooperative regulator destabilizes the system. Discuss why this system is problematic for noncooperative control.

Solution

For this case the largest magnitude eigenvalues of L are

$$|\operatorname{eig}(L)| = \begin{bmatrix} 0.63 & 0.63 & 0.62 & 0.62 & 0.59 & 0.59 & \cdots \end{bmatrix}$$

and we see the Nash equilibrium for the noncooperative game is stable. So we have removed the first source of closed-loop instability by switching the input-output pairings of the two subsystems. There are seven states in the complete system model, and the magnitudes of the eigenvalues of the closed-loop regulator $(A + BK)$ are

$$|\operatorname{eig}(A+BK)| = \begin{bmatrix} 1.03 & 1.03 & 0.37 & 0.37 & 0.77 & 0.77 & 0.04 \end{bmatrix}$$

which also gives an unstable closed-loop system. We see the distributed noncooperative regulator has destabilized a stable open-loop system. The problem with this pairing is the steady-state gains are now

$$G(0) = \begin{bmatrix} -0.5 & 1.5 \\ 1 & 0.5 \end{bmatrix}$$

If one computes any steady-state interaction measure, such as the relative gain array (RGA), we see the new pairings are poor from a steady-state interaction perspective

$$\text{RGA} = \begin{bmatrix} 0.14 & 0.86 \\ 0.86 & 0.14 \end{bmatrix}$$

Neither pairing of the inputs and outputs is closed-loop stable with noncooperative distributed MPC.

Decentralized control with this pairing is discussed in Exercise 6.10.

□

Example 6.11: Nash equilibrium is stable and the closed loop is stable

Next consider the system

$$G(s) = \begin{bmatrix} \dfrac{1}{s^2 + 2(0.2)s + 1} & \dfrac{0.5}{0.9s + 1} \\ \dfrac{-0.5}{(2s+1)(s+1)} & \dfrac{1.5}{0.75s^2 + 2(0.8)(0.75)s + 1} \end{bmatrix}$$

Compute the eigenvalues of L and $A + BK$ for this system. What do you conclude about noncooperative distributed MPC for this system?

Solution

This system is not difficult to handle with distributed control. The gains are the same as in the original pairing in Example 6.9, and the steady-state coupling between the two subsystems is reasonably weak. Unlike Example 6.9, however, the responses of y_1 to u_2 and y_2 to u_1 have been slowed so they are not faster than the responses of y_1 to u_1 and y_2 to u_2, respectively. Computing the eigenvalues of L and $A + BK$ for noncooperative control gives

$$|\,\mathrm{eig}(L)\,| = \begin{bmatrix} 0.61 & 0.61 & 0.59 & 0.59 & 0.56 & 0.56 & 0.53 & 0.53 \cdots \end{bmatrix}$$

$$|\,\mathrm{eig}(A + BK)\,| = \begin{bmatrix} 0.88 & 0.88 & 0.74 & 0.67 & 0.67 & 0.53 & 0.53 \end{bmatrix}$$

The Nash equilibrium is stable since L is stable, and the closed loop is stable since both L and $A + BK$ are stable. □

These examples reveal the simple fact that communicating the actions of the other controllers does not guarantee acceptable closed-loop behavior. If the coupling of the subsystems is weak enough, both dynamically and in steady state, then the closed loop is stable. In this sense, noncooperative MPC has few advantages over completely decentralized control, which has this same basic property.

We next show how to obtain much better closed-loop properties while maintaining the small size of the distributed control problems.

6.2.4 Cooperative Game

In the cooperative game, the two players share a common objective, which can be considered to be the overall plant objective

$$V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) = \rho_1 V_1(x_1(0), \mathbf{u}_1, \mathbf{u}_2) + \rho_2 V_2(x_2(0), \mathbf{u}_2, \mathbf{u}_1)$$

in which the parameters ρ_1, ρ_2 are used to specify the relative weights of the two subsystems in the overall plant objective. In the cooperative problem, each player keeps track of *how his input affects the other player's output* as well as his own output. We can implement this cooperative game in several ways. The implementation leading to the simplest notation is to combine x_1 and x_2 into a single model

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} \\ \overline{B}_{21} \end{bmatrix} u_1 + \begin{bmatrix} \overline{B}_{12} \\ \overline{B}_{22} \end{bmatrix} u_2$$

and then express player one's stage cost as

$$\ell_1(x_1, x_2, u_1) = \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}' \begin{bmatrix} \rho_1 Q_1 & 0 \\ 0 & \rho_2 Q_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2} u_1' (\rho_1 R_1) u_1 + \text{const.}$$

$$V_{1f}(x_1, x_2) = \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}' \begin{bmatrix} \rho_1 P_{1f} & 0 \\ 0 & \rho_2 P_{2f} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Notice that u_2 does not appear because the contribution of u_2 to the stage cost cannot be affected by player one, and can therefore be neglected. The cost function is then expressed as

$$V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) = \sum_{k=0}^{N-1} \ell_1(x_1(k), x_2(k), u_1(k)) + V_{1f}(x_1(N), x_2(N))$$

Player one's optimal control problem is

$$\min_{\mathbf{u}_1} V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2)$$
$$\text{s.t.} \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} \\ \overline{B}_{21} \end{bmatrix} u_1 + \begin{bmatrix} \overline{B}_{12} \\ \overline{B}_{22} \end{bmatrix} u_2$$

Note that this form is identical to the noncooperative form presented previously if we redefine the terms (noncooperative $\longrightarrow$ cooperative)

$$x_1 \to \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad A_1 \to \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \quad \overline{B}_{11} \to \begin{bmatrix} \overline{B}_{11} \\ \overline{B}_{21} \end{bmatrix} \quad \overline{B}_{12} \to \begin{bmatrix} \overline{B}_{12} \\ \overline{B}_{22} \end{bmatrix}$$
$$Q_1 \to \begin{bmatrix} \rho_1 Q_1 & 0 \\ 0 & \rho_2 Q_2 \end{bmatrix} \qquad R_1 \to \rho_1 R_1 \quad P_{1f} \to \begin{bmatrix} \rho_1 P_{1f} & 0 \\ 0 & \rho_2 P_{2f} \end{bmatrix}$$

Any computational program written to solve either the cooperative or noncooperative optimal control problem can be used to solve the other.

Eliminating states $\mathbf{x}_2$. An alternative implementation is to remove states $x_2(k), k \geq 1$ from player one's optimal control problem by substituting the dynamic model of system two. This implementation reduces the size of the dynamic model because only states x_1 are retained. This reduction in model size may be important in applications with many players. The removal of states $x_2(k), k \geq 1$ also introduces linear terms into player one's objective function. We start by using the dynamic model for x_2 to obtain

$$\begin{bmatrix} x_2(1) \\ x_2(2) \\ \vdots \\ x_2(N) \end{bmatrix} = \begin{bmatrix} A_2 \\ A_2^2 \\ \vdots \\ A_2^N \end{bmatrix} x_2(0) + \begin{bmatrix} \overline{B}_{21} & & & \\ A_2\overline{B}_{21} & \overline{B}_{21} & & \\ \vdots & \vdots & \ddots & \\ A_2^{N-1}\overline{B}_{21} & A_2^{N-2}\overline{B}_{21} & \dots & \overline{B}_{21} \end{bmatrix} \begin{bmatrix} u_1(0) \\ u_1(1) \\ \vdots \\ u_1(N-1) \end{bmatrix} +$$
$$\begin{bmatrix} \overline{B}_{22} & & & \\ A_2\overline{B}_{22} & \overline{B}_{22} & & \\ \vdots & \vdots & \ddots & \\ A_2^{N-1}\overline{B}_{22} & A_2^{N-2}\overline{B}_{22} & \dots & \overline{B}_{22} \end{bmatrix} \begin{bmatrix} u_2(0) \\ u_2(1) \\ \vdots \\ u_2(N-1) \end{bmatrix}$$

Using more compact notation, we have

$$\mathbf{x}_2 = \mathcal{A}_2 x_2(0) + \mathcal{B}_{21}\mathbf{u}_1 + \mathcal{B}_{22}\mathbf{u}_2$$

We can use this relation to replace the cost contribution of $\mathbf{x}_2$ with linear and quadratic terms in $\mathbf{u}_1$ as follows

$$\sum_{k=0}^{N-1} x_2(k)'Q_2x_2(k) + x_2(N)'P_{2f}x_2(N) =$$
$$\mathbf{u}_1' \left[\mathcal{B}_{21}'\mathcal{Q}_2\mathcal{B}_{21}\right]\mathbf{u}_1 + 2\left[x_2(0)'\mathcal{A}_2' + \mathbf{u}_2'\mathcal{B}_{22}'\right]\mathcal{Q}_2\mathcal{B}_{21}\,\mathbf{u}_1 + \text{constant}$$

in which

$$\mathcal{Q}_2 = \text{diag}\left(\begin{bmatrix} Q_2 & Q_2 & \dots & P_{2f} \end{bmatrix}\right) \qquad Nn_2 \times Nn_2 \text{ matrix}$$

and the constant term contains products of $x_2(0)$ and $\mathbf{u}_2$, which are constant with respect to player one's decision variables and can therefore be neglected.

Next we insert the new terms created by eliminating $\mathbf{x}_2$ into the cost function. Assembling the cost function gives

$$\min_{\mathbf{z}} (1/2)\mathbf{z}'\widetilde{H}\mathbf{z} + h'\mathbf{z}$$
$$\text{s.t. } D\mathbf{z} = d$$

and (1.57) again gives the necessary and sufficient conditions for the optimal solution

$$\begin{bmatrix} \widetilde{H} & -D' \\ -D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ -\widetilde{A}_1 \end{bmatrix} x_1(0) + \begin{bmatrix} -\widetilde{A}_2 \\ 0 \end{bmatrix} x_2(0) + \begin{bmatrix} -\widetilde{B}_{22} \\ -\widetilde{B}_{12} \end{bmatrix} \mathbf{u}_2 \quad (6.13)$$

in which

$$\widetilde{H} = H + E'\mathcal{B}_{21}'\mathcal{Q}_2\mathcal{B}_{21}E \qquad \widetilde{B}_{22} = E'\mathcal{B}_{21}'\mathcal{Q}_2\mathcal{B}_{22} \qquad \widetilde{A}_2 = E'\mathcal{B}_{21}'\mathcal{Q}_2\mathcal{A}_2$$
$$E = I_N \otimes \begin{bmatrix} I_{m_1} & 0_{m_1,n_1} \end{bmatrix}$$

See also Exercise 6.13 for details on constructing the padding matrix E. Comparing the cooperative and noncooperative dynamic games, (6.13) and (6.12), we see the cooperative game has made three changes: (i) the quadratic penalty H has been modified, (ii) the effect of $x_2(0)$ has been included with the term $\widetilde{A}_2$, and (iii) the influence of $\mathbf{u}_2$ has been modified with the term $\widetilde{B}_{22}$. Notice that the size of the vector $\mathbf{z}$ has not changed, and we have accomplished the goal of keeping player one's dynamic model in the cooperative game the same size as his dynamic model in the noncooperative game.

Regardless of the implementation choice, the cooperative optimal control problem is no more complex than the noncooperative game considered previously. The extra information required by player one in the cooperative game is $x_2(0)$. Player one requires $\mathbf{u}_2$ in both the cooperative and noncooperative games. Only in decentralized control does player one not require player two's input sequence $\mathbf{u}_2$. The other extra required information, $A_2, B_{21}, Q_2, R_2, P_{2f}$, are fixed parameters and making their values available to player one is a minor communication overhead.

Proceeding as before, we solve this equation for $\mathbf{z}^0$ and pick out the rows corresponding to the elements of $\mathbf{u}_1^0$ giving

$$\mathbf{u}_1^0(x(0), \mathbf{u}_2) = \begin{bmatrix} K_{11} & K_{12} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + L_1 \mathbf{u}_2$$

Combining the optimal control laws for each player gives

$$\begin{bmatrix} \mathbf{u}_1^0 \\ \mathbf{u}_2^0 \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} 0 & L_1 \\ L_2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^p$$

in which the gain matrix multiplying the state is a full matrix for the cooperative game. Substituting the optimal control into the iteration gives

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1K_{11} & w_1K_{12} \\ w_2K_{21} & w_2K_{22} \end{bmatrix}}_{\overline{K}} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \underbrace{\begin{bmatrix} (1-w_1)I & w_1L_1 \\ w_2L_2 & (1-w_2)I \end{bmatrix}}_{L} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}^p$$

Finally writing this equation in the plantwide notation, we express the iteration as

$$\mathbf{u}^{p+1} = \overline{K}x(0) + L\mathbf{u}^p$$

Exponential stability of the closed-loop system. In the case of cooperative control, we consider the closed-loop system with a finite number of iterations, p. With finite iterations, distributed MPC becomes a form of *suboptimal* MPC as discussed in Sections 6.1.2 and 2.7. To analyze the behavior of the cooperative controller with a finite number of iterations, we require the cost decrease achieved by a single iteration, which we derive next. First we write the complete system evolution as in (6.10)

$$x^+ = Ax + Bu$$

in which A and B are defined in (6.11). We can then use (6.3) to express the overall cost function

$$V(x(0), \mathbf{u}) = (1/2)x'(0)(Q + \mathcal{A}'\mathcal{Q}\mathcal{A})x(0) + \mathbf{u}'(\mathcal{B}'\mathcal{Q}\mathcal{A})x(0) + (1/2)\mathbf{u}'H_{\mathbf{u}}\mathbf{u}$$

in which $\mathcal{A}$ and $\mathcal{B}$ are given in (6.1), the cost penalties $\mathcal{Q}$ and $\mathcal{R}$ are given in (6.2) and (6.9), and

$$H_{\mathbf{u}} = \mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R}$$

The overall cost is a positive definite quadratic function in $\mathbf{u}$ because R_1 and R_2 are positive definite, and therefore so are $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}$.

The iteration in the two players' moves satisfies

$$\begin{aligned}
(\mathbf{u}_1^{p+1}, \mathbf{u}_2^{p+1}) &= \Big((w_1\mathbf{u}_1^0 + (1-w_1)\mathbf{u}_1^p), (w_2\mathbf{u}_2^0 + (1-w_2)\mathbf{u}_2^p)\Big) \\
&= (w_1\mathbf{u}_1^0, (1-w_2)\mathbf{u}_2^p) + ((1-w_1)\mathbf{u}_1^p, w_2\mathbf{u}_2^0) \\
(\mathbf{u}_1^{p+1}, \mathbf{u}_2^{p+1}) &= w_1(\mathbf{u}_1^0, \mathbf{u}_2^p) + w_2(\mathbf{u}_1^p, \mathbf{u}_2^0) \qquad (6.14)
\end{aligned}$$

Exercise 6.18 analyzes the cost decrease for a convex step with a positive definite quadratic function and shows

$$\begin{aligned}
V(x(0), \mathbf{u}_1^{p+1}, \mathbf{u}_2^{p+1}) &= V(x(0), \mathbf{u}_1^p, \mathbf{u}_2^p) \\
&\quad - \frac{1}{2}\Big[\mathbf{u}^p - \mathbf{u}^0(x(0))\Big]' P \Big[\mathbf{u}^p - \mathbf{u}^0(x(0))\Big] \qquad (6.15)
\end{aligned}$$

in which $P > 0$ is given by

$$P = H_{\mathbf{u}}D^{-1}\widetilde{H}D^{-1}H_{\mathbf{u}} \qquad \widetilde{H} = D - N$$

$$D = \begin{bmatrix} w_1^{-1}H_{\mathbf{u},11} & 0 \\ 0 & w_2^{-1}H_{\mathbf{u},22} \end{bmatrix} \qquad N = \begin{bmatrix} -w_1^{-1}w_2H_{\mathbf{u},11} & H_{\mathbf{u},12} \\ H_{\mathbf{u},21} & -w_1w_2^{-1}H_{\mathbf{u},22} \end{bmatrix}$$

and $H_{\mathbf{u}}$ is partitioned for the two players' input sequences. Notice that the cost decrease achieved in a single iteration is quadratic in the distance from the optimum. An important conclusion is that *each iteration in the cooperative game reduces the systemwide cost.* This cost reduction is the key property that gives cooperative MPC its excellent convergence properties, as we show next.

The two players' warm starts at the next sample are given by

$$\begin{aligned}\widetilde{\mathbf{u}}_1^+ &= (u_1(1), u_1(2), \ldots, u_1(N-1), 0)\\ \widetilde{\mathbf{u}}_2^+ &= (u_2(1), u_2(2), \ldots, u_2(N-1), 0)\end{aligned}$$

We define the following linear time-invariant functions g_1^p and g_2^p as the outcome of applying the control iteration procedure p times

$$\begin{aligned}\mathbf{u}_1^p &= g_1^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\\ \mathbf{u}_2^p &= g_2^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\end{aligned}$$

in which $p \geq 0$ is an integer, x_1 and x_2 are the states, and $\mathbf{u}_1, \mathbf{u}_2$ are the input sequences from the previous sample, used to generate the warm start for the iteration. Here we consider p to be constant with time, but Exercise 6.20 considers the case in which the controller iterations may vary with sample time. The system evolution is then given by

$$\begin{aligned}x_1^+ &= A_1 x_1 + \overline{B}_{11} u_1 + \overline{B}_{12} u_2 & x_2^+ &= A_2 x_2 + \overline{B}_{21} u_1 + \overline{B}_{22} u_2\\ \mathbf{u}_1^+ &= g_1^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) & \mathbf{u}_2^+ &= g_2^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\end{aligned} \tag{6.16}$$

By the construction of the warm start, $\widetilde{\mathbf{u}}_1^+, \widetilde{\mathbf{u}}_2^+$, we have

$$\begin{aligned}V(x_1^+, x_2^+, \widetilde{\mathbf{u}}_1^+, \widetilde{\mathbf{u}}_2^+) = V(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) &- \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)\\ &+ (1/2)\rho_1 x_1(N)' \left[A_1' P_{1f} A_1 - P_{1f} + Q_1\right] x_1(N)\\ &+ (1/2)\rho_2 x_2(N)' \left[A_2' P_{2f} A_2 - P_{2f} + Q_2\right] x_2(N)\end{aligned}$$

From our choice of terminal penalty satisfying (6.8), the last two terms are zero giving

$$\begin{aligned}V(x_1^+, x_2^+, \widetilde{\mathbf{u}}_1^+, \widetilde{\mathbf{u}}_2^+) &= V(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\\ &\qquad - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)\end{aligned} \tag{6.17}$$

No optimization, $p = 0$. If we do no further optimization, then we have $\mathbf{u}_1^+ = \widetilde{\mathbf{u}}_1^+$, $\mathbf{u}_2^+ = \widetilde{\mathbf{u}}_2^+$, and the equality

$$V(x_1^+, x_2^+, \mathbf{u}_1^+, \mathbf{u}_2^+) = V(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

The input sequences add a zero at each sample until $\mathbf{u}_1 = \mathbf{u}_2 = 0$ at time $k = N$. The system decays exponentially under zero control and the closed loop is exponentially stable.

Further optimization, $p \geq 1$. We next consider the case in which optimization is performed. Equation 6.15 then gives

$$V(x_1^+, x_2^+, \mathbf{u}_1^+, \mathbf{u}_2^+) \leq V(x_1^+, x_2^+, \tilde{\mathbf{u}}_1^+, \tilde{\mathbf{u}}_2^+) - \\ \left[\tilde{\mathbf{u}}^+ - \mathbf{u}^0(x^+)\right]' P \left[\tilde{\mathbf{u}}^+ - \mathbf{u}^0(x^+)\right] \qquad p \geq 1$$

with equality holding for $p = 1$. Using this result in (6.17) gives

$$V(x_1^+, x_2^+, \mathbf{u}_1^+, \mathbf{u}_2^+) \leq V(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2) \\ - \left[\tilde{\mathbf{u}}^+ - \mathbf{u}^0(x^+)\right]' P \left[\tilde{\mathbf{u}}^+ - \mathbf{u}^0(x^+)\right]$$

Since V is bounded below by zero and ℓ_1 and ℓ_2 are positive functions, we conclude the time sequence $V(x_1(k), x_2(k), \mathbf{u}_1(k), \mathbf{u}_2(k))$ converges. and therefore $x_1(k)$, $x_2(k)$, $u_1(k)$, and $u_2(k)$ converge to zero. Moreover, since $P > 0$, the last term implies that $\tilde{\mathbf{u}}^+$ converges to $\mathbf{u}^0(x^+)$, which converges to zero because x^+ converges to zero. Therefore, the entire input sequence $\mathbf{u}$ converges to zero. Because the total system evolution is a linear time-invariant system, the convergence is exponential. Even though we are considering here a form of *suboptimal* MPC, we do not require an additional inequality constraint on $\mathbf{u}$ because the problem considered here is *unconstrained* and the iterations satisfy (6.15).

6.2.5 Tracking Nonzero Setpoints

For tracking nonzero setpoints, we compute steady-state targets as discussed in Section 1.5. The steady-state input-output model is given by

$$y_s = G u_s \qquad G = C(I - A)^{-1} B$$

in which G is the steady-state gain of the system. The two subsystems are denoted

$$\begin{bmatrix} y_{1s} \\ y_{2s} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}$$

For simplicity, we assume that the targets are chosen to be the measurements ($H = I$). Further, we assume that both local systems are square, and that the local targets can be reached exactly with the local inputs. This assumption means that G_{11} and G_{22} are square matrices of full rank. We remove all of these assumptions when we treat the constrained two-player game in the next section. If there is model error, integrating disturbance models are required as discussed in Chapter 1. We discuss these later.

The target problem also can be solved with any of the four approaches discussed so far. We consider each.

Centralized case. The centralized problem gives in one shot both inputs required to meet both output setpoints

$$\begin{aligned} u_s &= G^{-1} y_{\text{sp}} \\ y_s &= y_{\text{sp}} \end{aligned}$$

Decentralized case. The decentralized problem considers only the diagonal terms and computes the following steady inputs

$$u_s = \begin{bmatrix} G_{11}^{-1} & \\ & G_{22}^{-1} \end{bmatrix} y_{\text{sp}}$$

Notice these inputs produce offset in both output setpoints

$$y_s = \begin{bmatrix} I & G_{12}G_{22}^{-1} \\ G_{21}G_{11}^{-1} & I \end{bmatrix} y_{\text{sp}}$$

Noncooperative case. In the noncooperative game, each player attempts to remove offset in only its outputs. Player one solves the following problem

$$\begin{aligned} &\min_{u_1} (y_1 - y_{1sp})' \overline{Q}_1 (y_1 - y_{1sp}) \\ &\text{s.t. } y_1 = G_{11} u_1 + G_{12} u_2 \end{aligned}$$

Because the target can be reached exactly, the optimal solution is to find u_1 such that $y_1 = y_{1sp}$, which gives

$$u_{1s}^0 = G_{11}^{-1} \left(y_{1sp} - G_{12} u_2^p \right)$$

Player two solves the analogous problem. If we iterate on the two players' solutions, we obtain

$$\begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1 G_{11}^{-1} & \\ & w_2 G_{22}^{-1} \end{bmatrix}}_{\overline{K}_s} \begin{bmatrix} y_{1sp} \\ y_{2sp} \end{bmatrix} + \underbrace{\begin{bmatrix} w_2 I & -w_1 G_{11}^{-1} G_{12} \\ -w_2 G_{22}^{-1} G_{21} & w_1 I \end{bmatrix}}_{L_s} \begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}^p$$

This iteration can be summarized by

$$u_s^{p+1} = \overline{K}_s y_{\mathrm{sp}} + L_s u_s^p$$

If L_s is stable, this iteration converges to

$$\begin{aligned} u_s^\infty &= (I - L_s)^{-1} \overline{K}_s y_{\mathrm{sp}} \\ u_s^\infty &= G^{-1} y_{\mathrm{sp}} \end{aligned}$$

and we have no offset. We already have seen that we cannot expect the dynamic noncooperative iteration to converge. The next several examples explore the issue of whether we can expect at least the steady-state iteration to be stable.

Cooperative case. In the cooperative case, both players work on minimizing the offset in both outputs. Player one solves

$$\begin{aligned} \min_{u_1} (1/2) \begin{bmatrix} y_1 - y_{1sp} \\ y_2 - y_{2sp} \end{bmatrix}' \begin{bmatrix} \rho_1 \overline{Q}_1 & \\ & \rho_2 \overline{Q}_2 \end{bmatrix} \begin{bmatrix} y_1 - y_{1sp} \\ y_2 - y_{2sp} \end{bmatrix} \\ \text{s.t.} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} u_1 + \begin{bmatrix} G_{12} \\ G_{22} \end{bmatrix} u_2 \end{aligned}$$

We can write this in the general form

$$\begin{aligned} \min_{r_s} (1/2) r_s' H r_s + h' r_s \\ \text{s.t. } D r_s = d \end{aligned}$$

in which

$$r_s = \begin{bmatrix} y_{1s} \\ y_{2s} \\ u_{1s} \end{bmatrix} \qquad H = \begin{bmatrix} \rho_1 \overline{Q}_1 & & \\ & \rho_2 \overline{Q}_2 & \\ & & 0 \end{bmatrix} \qquad h = \begin{bmatrix} -Q y_{\mathrm{sp}} \\ 0 \end{bmatrix}$$

$$D = \begin{bmatrix} I & -G_1 \end{bmatrix} \qquad d = G_2 u_2 \qquad G_1 = \begin{bmatrix} G_{11} \\ G_{12} \end{bmatrix} \qquad G_2 = \begin{bmatrix} G_{12} \\ G_{22} \end{bmatrix}$$

We can then solve the linear algebra problem

$$\begin{bmatrix} H & -D' \\ -D & 0 \end{bmatrix} \begin{bmatrix} r_s \\ \lambda_s \end{bmatrix} = - \begin{bmatrix} h \\ d \end{bmatrix}$$

and identify the linear gains between the optimal u_{1s} and the setpoint y_{sp} and player two's input u_{2s}

$$u_{1s}^0 = K_{1s} y_{\text{sp}} + L_{1s} u_{2s}^p$$

Combining the optimal control laws for each player gives

$$\begin{bmatrix} u_{1s}^0 \\ u_{2s}^0 \end{bmatrix} = \begin{bmatrix} K_{1s} \\ K_{2s} \end{bmatrix} y_{\text{sp}} + \begin{bmatrix} 0 & L_{1s} \\ L_{2s} & 0 \end{bmatrix} \begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}^p$$

Substituting the optimal control into the iteration gives

$$\begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1 K_{1s} \\ w_2 K_{2s} \end{bmatrix}}_{\overline{K}_s} y_{\text{sp}} + \underbrace{\begin{bmatrix} (1-w_1)I & w_1 L_{1s} \\ w_2 L_{2s} & (1-w_2)I \end{bmatrix}}_{L_s} \begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}^p$$

Finally writing this equation in the plantwide notation, we express the iteration as

$$u_s^{p+1} = \overline{K}_s y_{\text{sp}} + L_s u_s^p$$

As we did with the cooperative regulation problem, we can analyze the optimization problem to show that this iteration is always stable and converges to the centralized target. Next we explore the use of these approaches in some illustrative examples.

Example 6.12: Stability and offset in the distributed target calculation

Consider the following two-input, two-output system with steady-state gain matrix and setpoint

$$\begin{bmatrix} y_{1s} \\ y_{2s} \end{bmatrix} = \begin{bmatrix} -0.5 & 1.0 \\ 2.0 & 1.0 \end{bmatrix} \begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix} \qquad \begin{bmatrix} y_{1sp} \\ y_{2sp} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

(a) Show the first 10 iterations of the noncooperative and cooperative steady-state cases starting with the decentralized solution as the initial guess.

Describe the differences. Compute the eigenvalues of L for the cooperative and noncooperative cases. Discuss the relationship between these eigenvalues and the result of the iteration calculations.

Mark also the solution to the centralized and decentralized cases on your plots.

(b) Switch the pairings and repeat the previous part. Explain your results.

Solution

(a) The first 10 iterations of the noncooperative steady-state calculation are shown in Figure 6.2. Notice the iteration is unstable and the steady-state target does not converge. The cooperative case is shown in Figure 6.3. This case is stable and the iterations converge to the centralized target and achieve zero offset. The magnitudes of the eigenvalues of L_s for the noncooperative (nc) and cooperative (co) cases are given by

$$\left|\text{eig}(L_{snc})\right| = \{1.12, 1.12\} \qquad \left|\text{eig}(L_{sco})\right| = \{0.757, 0.243\}$$

Stability of the iteration is determined by the magnitudes of the eigenvalues of L_s.

(b) Reversing the pairings leads to the following gain matrix in which we have reversed the labels of the outputs for the two systems

$$\begin{bmatrix} y_{1s} \\ y_{2s} \end{bmatrix} = \begin{bmatrix} 2.0 & 1.0 \\ -0.5 & 1.0 \end{bmatrix} \begin{bmatrix} u_{1s} \\ u_{2s} \end{bmatrix}$$

The first 10 iterations of the noncooperative and cooperative controllers are shown in Figures 6.4 and 6.5. For this pairing, the noncooperative case also converges to the centralized target. The eigenvalues are given by

$$\left|\text{eig}(L_{snc})\right| = \{0.559, 0.559\} \qquad \left|\text{eig}(L_{sco})\right| = \{0.757, 0.243\}$$

The eigenvalues of the cooperative case are unaffected by the reversal of pairings. □

Given the stability analysis of the simple unconstrained two-player game, we remove from further consideration two options we have been

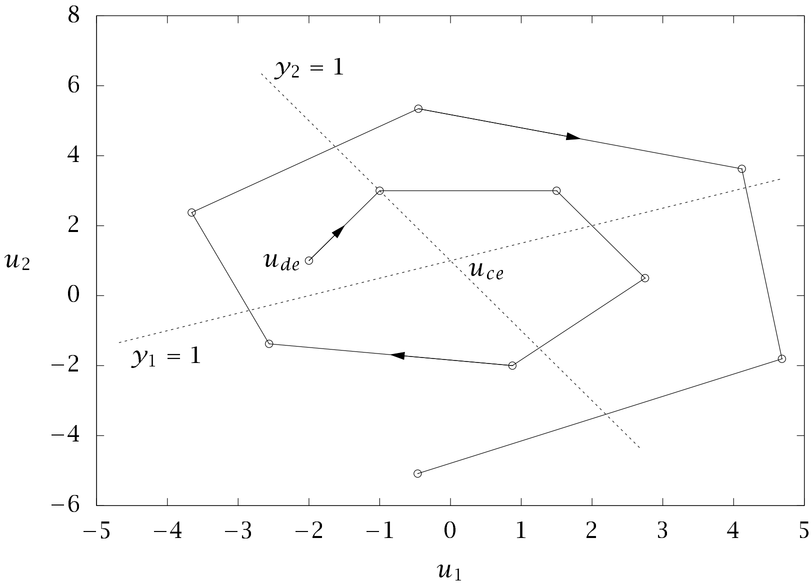

Figure 6.2: Ten iterations of noncooperative steady-state calculation, $u^{[0]} = u_{de}$; iterations are unstable, $u^p \to \infty$.

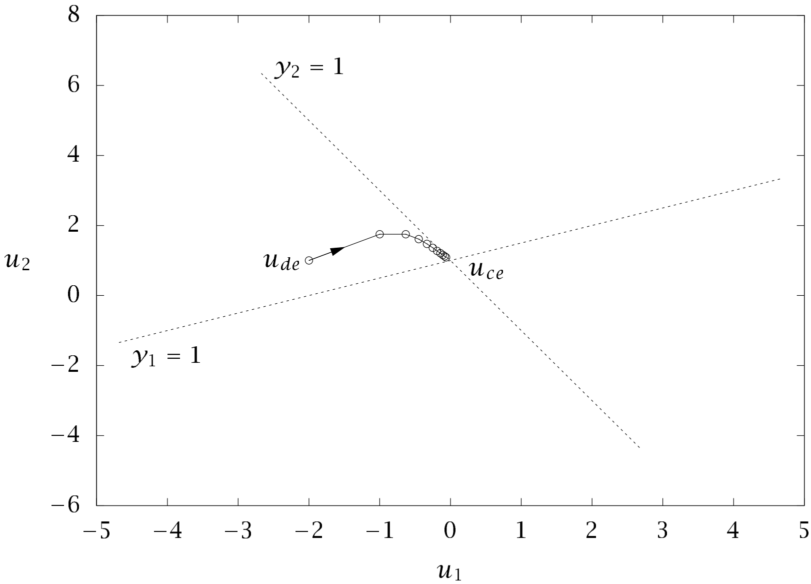

Figure 6.3: Ten iterations of cooperative steady-state calculation, $u^{[0]} = u_{de}$; iterations are stable, $u^p \to u_{ce}$.

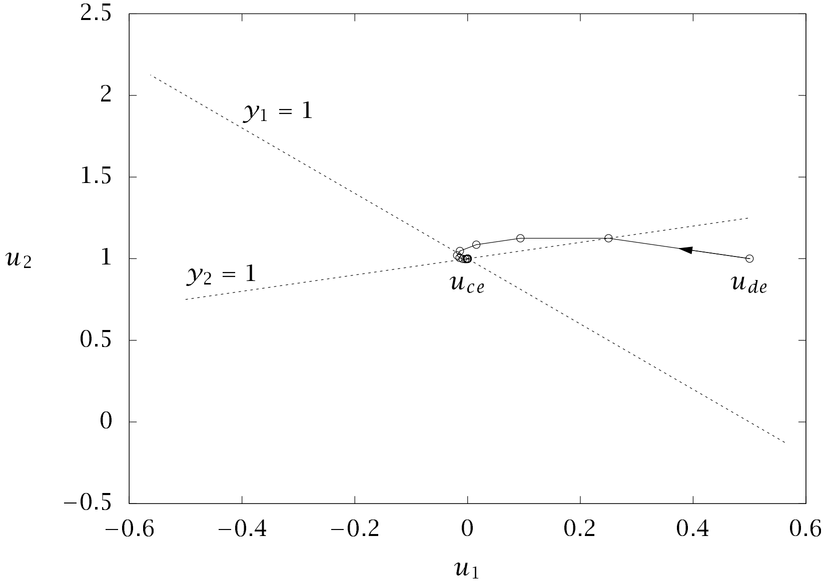

Figure 6.4: Ten iterations of noncooperative steady-state calculation, $u^{[0]} = u_{de}$; iterations are stable with reversed pairing.

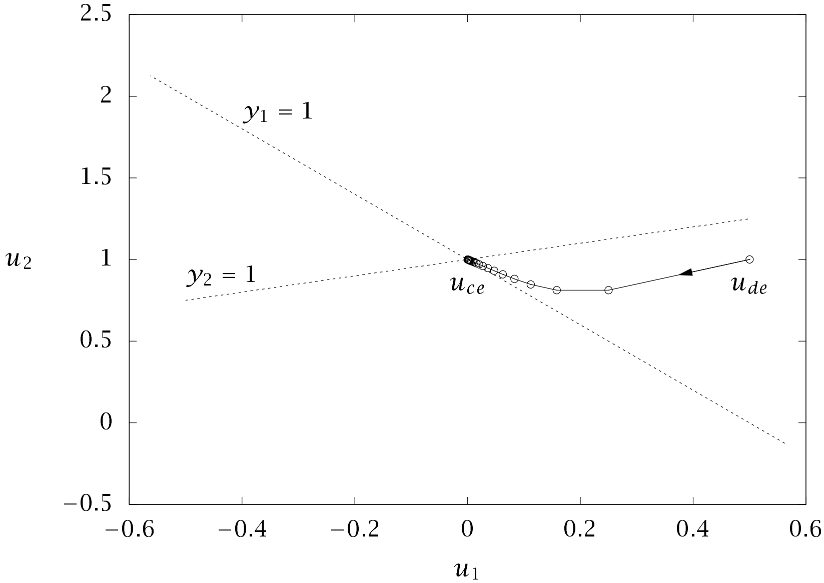

Figure 6.5: Ten iterations of cooperative steady-state calculation, $u^{[0]} = u_{de}$; iterations are stable with reversed pairing.

discussing to this point: noncooperative control and decentralized control. We next further develop the theory of cooperative MPC and compare its performance to centralized MPC in more general and challenging situations.

6.2.6 State Estimation

Given output measurements, we can express the state estimation problem also in distributed form. Player one uses local measurements of y_1 and knowledge of both inputs u_1 and u_2 to estimate state x_1

$$\hat{x}_1^+ = A_1\hat{x}_1 + \overline{B}_{11}u_1 + \overline{B}_{12}u_2 + L_1(y_1 - C_1\hat{x}_1)$$

Defining estimate error to be $e_1 = x_1 - \hat{x}_1$ gives

$$e_1^+ = (A_1 - L_1C_1)e_1$$

Because all the subsystems are stable, we know L_1 exists so that $A_1 - L_1C_1$ is stable and player one's local estimator is stable. The estimate error for the two subsystems is then given by

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}^+ = \begin{bmatrix} A_{L1} & \\ & A_{L2} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{6.18}$$

in which $A_{Li} = A_i - L_iC_i$.

Closed-loop stability. The dynamics of the estimator are given by

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} & \overline{B}_{12} \\ \overline{B}_{21} & \overline{B}_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} L_1C_1 & \\ & L_2C_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

In the control law we use the state estimate in place of the state, which is unmeasured and unknown. We consider two cases.

Converged controller. In this case the distributed control law converges to the centralized controller, and we have

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}$$

The closed-loop system evolves according to

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}^+ = \left\{ \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} & \overline{B}_{12} \\ \overline{B}_{21} & \overline{B}_{22} \end{bmatrix} \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \right\} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} L_1 C_1 & \\ & L_2 C_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

The $A+BK$ term is stable because this term is the same as in the stabilizing centralized controller. The perturbation is exponentially decaying because the distributed estimators are stable. Therefore $\hat{x}$ goes to zero exponentially, which, along with e going to zero exponentially, implies x goes to zero exponentially.

Finite iterations. Here we use the state plus input sequence description given in (6.16), which, as we have already noted, is a linear time-invariant system. With estimate error, the system equation is

$$\begin{bmatrix} \hat{x}_1^+ \\ \hat{x}_2^+ \\ \mathbf{u}_1^+ \\ \mathbf{u}_2^+ \end{bmatrix} = \begin{bmatrix} A_1 \hat{x}_1 + \overline{B}_{11} u_1 + \overline{B}_{12} u_2 \\ A_2 \hat{x}_2 + \overline{B}_{21} u_1 + \overline{B}_{22} u_2 \\ g_1^p(\hat{x}_1, \hat{x}_2, \mathbf{u}_1, \mathbf{u}_2) \\ g_2^p(\hat{x}_1, \hat{x}_2, \mathbf{u}_1, \mathbf{u}_2) \end{bmatrix} + \begin{bmatrix} L_1 C_1 e_1 \\ L_2 C_2 e_2 \\ 0 \\ 0 \end{bmatrix}$$

Because there is again only one-way coupling between the estimate error evolution, (6.18), and the system evolution given above, the composite system is exponentially stable.

6.3 Constrained Two-Player Game

Now that we have introduced most of the notation and the fundamental ideas, we consider more general cases. Because we are interested in establishing stability properties of the controlled systems, we focus exclusively on *cooperative distributed MPC* from this point forward. In this section we consider convex input constraints on the two players. We assume output constraints have been softened with exact soft constraints and added to the objective function, so do not consider output constraints explicitly. The input constraints break into two significant categories: coupled and uncoupled constraints. We treat each of these in turn.

We also allow unstable systems and replace Assumption 6.8 with the following more general restrictions on the systems and controller parameters.

Assumption 6.13 (Constrained two-player game)**.**

(a) The systems $(\underline{A}_i, \underline{B}_i)$, $i = 1, 2$ are stabilizable, in which $\underline{A}_i = \text{diag}(A_{1i}, A_{2i})$ and $\underline{B}_i = \begin{bmatrix} B_{1i} \\ B_{2i} \end{bmatrix}$.

(b) The systems (A_i, C_i), $i = 1, 2$ are detectable.

(c) The input penalties R_1, R_2 are positive definite, and the state penalties Q_1, Q_2 are semidefinite.

(d) The systems (A_1, Q_1) and (A_2, Q_2) are detectable.

(e) The horizon is chosen sufficiently long to zero the unstable modes, $N \geq \max_{i \in \mathbb{I}_{1:2}} \underline{n}_i^u$, in which $\underline{n}_i^u$ is the number of unstable modes of $\underline{A}_i$, i.e., number of $\lambda \in \text{eig}(\underline{A}_i)$ such that $|\lambda| \geq 1$.

Assumption (b) implies that we have L_i such that $(A_i - L_i C_i), i = 1, 2$ is stable. Note that the stabilizable and detectable conditions of Assumption 6.13 are automatically satisfied if we obtain the state space models from a minimal realization of the input/output models for (u_i, y_j), $i, j = 1, 2$.

Unstable modes. To handle unstable systems, we add constraints to zero the unstable modes at the end of the horizon. To set up this constraint, consider the real Schur decomposition of A_{ij} for $i, j \in \mathbb{I}_{1:2}$

$$A_{ij} = \begin{bmatrix} S_{ij}^s & S_{ij}^u \end{bmatrix} \begin{bmatrix} A_{ij}^s & - \\ & A_{ij}^u \end{bmatrix} \begin{bmatrix} {S_{ij}^s}' \\ {S_{ij}^u}' \end{bmatrix} \tag{6.19}$$

in which A_{ij}^s is upper triangular and stable, and A_{ij}^u is upper triangular with all unstable eigenvalues.[3] Given the Schur decomposition (6.19), we define the matrices

$$\begin{aligned} S_i^s &= \text{diag}(S_{i1}^s, S_{i2}^s) & \qquad A_i^s &= \text{diag}(A_{i1}^s, A_{i2}^s) \quad i \in \mathbb{I}_{1:2} \\ S_i^u &= \text{diag}(S_{i1}^u, S_{i2}^u) & \qquad A_i^u &= \text{diag}(A_{i1}^u, A_{i2}^u) \quad i \in \mathbb{I}_{1:2} \end{aligned}$$

These matrices satisfy the Schur decompositions

$$A_i = \begin{bmatrix} S_i^s & S_i^u \end{bmatrix} \begin{bmatrix} A_i^s & - \\ & A_i^u \end{bmatrix} \begin{bmatrix} {S_i^s}' \\ {S_i^u}' \end{bmatrix} \quad i \in \mathbb{I}_{1:2}$$

We further define the matrices Σ_1, Σ_2 as the solutions to the Lyapunov equations

$${A_1^s}' \Sigma_1 A_1^s - \Sigma_1 = -{S_1^s}' Q_1 S_1^s \qquad {A_2^s}' \Sigma_2 A_2^s - \Sigma_2 = -{S_2^s}' Q_2 S_2^s \tag{6.20}$$

[3] If A_{ij} is stable, then there is no A_{ij}^u and S_{ij}^u.

We then choose the terminal penalty for each subsystem to be the cost to go under zero control

$$P_{1f} = S_1^s \Sigma_1 S_1^{s\prime} \qquad\qquad P_{2f} = S_2^s \Sigma_2 S_2^{s\prime}$$

6.3.1 Uncoupled Input Constraints

We consider convex input constraints of the following form

$$Hu(k) \leq h \qquad k = 0, 1, \ldots, N$$

Defining convex set $\mathbb{U}$

$$\mathbb{U} = \{u | Hu \leq h\}$$

we express the input constraints as

$$u(k) \in \mathbb{U} \qquad k = 0, 1, \ldots, N$$

We drop the time index and indicate the constraints are applied over the entire input sequence using the notation $\mathbf{u} \in \mathbb{U}$. In the uncoupled constraint case, the two players' inputs must satisfy

$$\mathbf{u}_1 \in \mathbb{U}_1 \qquad \mathbf{u}_2 \in \mathbb{U}_2$$

in which $\mathbb{U}_1$ and $\mathbb{U}_2$ are convex subsets of $\mathbb{R}^{m_1}$ and $\mathbb{R}^{m_2}$, respectively. The constraints are termed *uncoupled* because there is no interaction or coupling of the inputs in the constraint relation. Player one then solves the following constrained optimization

$$\begin{gathered}
\min_{\mathbf{u}_1} V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) \\
\text{s.t.} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} \\ \overline{B}_{21} \end{bmatrix} u_1 + \begin{bmatrix} \overline{B}_{12} \\ \overline{B}_{22} \end{bmatrix} u_2 \\
\mathbf{u}_1 \in \mathbb{U}_1 \\
S_{j1}^{u\,\prime} x_{j1}(N) = 0 \quad j \in \mathbb{I}_{1:2} \\
|\mathbf{u}_1| \leq d_1(|x_{11}(0)| + |x_{21}(0)|) \quad x_{11}(0), x_{21}(0) \in r\mathcal{B}
\end{gathered}$$

in which we include the system's hard input constraints, the stability constraint on the unstable modes, and the Lyapunov stability constraints. Exercise 6.22 discusses how to write the constraint $|\mathbf{u}_1| \leq d_1 |x_1(0)|$ as a set of linear inequalities on $\mathbf{u}_1$. Similarly, player two solves

$$\min_{\mathbf{u}_2} V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2)$$
$$\text{s.t.} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} \\ \overline{B}_{21} \end{bmatrix} u_1 + \begin{bmatrix} \overline{B}_{12} \\ \overline{B}_{22} \end{bmatrix} u_2$$
$$\mathbf{u}_2 \in \mathbb{U}_2$$
$$S_{j2}^{u}{}' x_{j2}(N) = 0 \quad j \in \mathbb{I}_{1:2}$$
$$|\mathbf{u}_2| \leq d_2(|x_{21}(0)| + |x_{22}(0)|) \quad x_{12}(0), x_{22}(0) \in r\mathcal{B}$$

We denote the solutions to these problems as

$$\mathbf{u}_1^0(x_1(0), x_2(0), \mathbf{u}_2) \qquad \mathbf{u}_2^0(x_1(0), x_2(0), \mathbf{u}_1)$$

The feasible set $\mathcal{X}_N$ for the unstable system is the set of states for which the unstable modes can be brought to zero in N moves while satisfying the input constraints.

Given an initial iterate, $(\mathbf{u}_1^p, \mathbf{u}_2^p)$, the next iterate is defined to be

$$(\mathbf{u}_1, \mathbf{u}_2)^{p+1} = w_1(\mathbf{u}_1^0(x_1(0), x_2(0), \mathbf{u}_2^p), \mathbf{u}_2^p) + \\ w_2(\mathbf{u}_1^p, \mathbf{u}_2^0(x_1(0), x_2(0), \mathbf{u}_1^p))$$

To reduce the notational burden we denote this as

$$(\mathbf{u}_1, \mathbf{u}_2)^{p+1} = w_1(\mathbf{u}_1^0, \mathbf{u}_2^p) + w_2(\mathbf{u}_1^p, \mathbf{u}_2^0)$$

and the functional dependencies of $\mathbf{u}_1^0$ and $\mathbf{u}_2^0$ should be kept in mind.

This procedure provides three important properties, which we establish next.

1. The iterates are feasible: $(\mathbf{u}_1, \mathbf{u}_2)^p \in (\mathbb{U}_1, \mathbb{U}_2)$ implies $(\mathbf{u}_1, \mathbf{u}_2)^{p+1} \in (\mathbb{U}_1, \mathbb{U}_2)$. This follows from convexity of $\mathbb{U}_1$, $\mathbb{U}_2$ and the convex combination of the feasible points $(\mathbf{u}_1^p, \mathbf{u}_2^p)$ and $(\mathbf{u}_1^0, \mathbf{u}_2^0)$ to make $(\mathbf{u}_1, \mathbf{u}_2)^{p+1}$.

2. The cost decreases on iteration: $V(x_1(0), x_2(0), (\mathbf{u}_1, \mathbf{u}_2)^{p+1}) \leq V(x_1(0), x_2(0), (\mathbf{u}_1, \mathbf{u}_2)^p)$ for all $x_1(0)$, $x_2(0)$, and for all feasible $(\mathbf{u}_1, \mathbf{u}_2)^p \in (\mathbb{U}_1, \mathbb{U}_2)$. The systemwide cost satisfies the following inequalities

$$\begin{aligned} V(x(0), \mathbf{u}_1^{p+1}, \mathbf{u}_2^{p+1}) &= V\left(x(0), \left(w_1(\mathbf{u}_1^0, \mathbf{u}_2^p) + w_2(\mathbf{u}_1^p, \mathbf{u}_2^0)\right)\right) \\ &\leq w_1 V(x(0), (\mathbf{u}_1^0, \mathbf{u}_2^p)) + w_2 V(x(0), (\mathbf{u}_1^p, \mathbf{u}_2^0)) \\ &\leq w_1 V(x(0), (\mathbf{u}_1^p, \mathbf{u}_2^p)) + w_2 V(x(0), (\mathbf{u}_1^p, \mathbf{u}_2^p)) \\ &= V(x(0), \mathbf{u}_1^p, \mathbf{u}_2^p) \end{aligned}$$

The first equality follows from (6.14). The next inequality follows from convexity of V. The next follows from optimality of $\mathbf{u}_1^0$ and $\mathbf{u}_2^0$, and the last follows from $w_1 + w_2 = 1$. Because the cost is bounded below, the cost iteration converges.

3. The converged solution of the cooperative problem is equal to the optimal solution of the centralized problem. Establishing this property is discussed in Exercise 6.26.

Exponential stability of the closed-loop system. We next consider the closed-loop system. The two players' warm starts at the next sample are as defined previously

$$\begin{aligned}\tilde{\mathbf{u}}_1^+ &= (u_1(1), u_1(2), \ldots, u_1(N-1), 0)\\ \tilde{\mathbf{u}}_2^+ &= (u_2(1), u_2(2), \ldots, u_2(N-1), 0)\end{aligned}$$

We define again the functions g_1^p, g_2^p as the outcome of applying the control iteration procedure p times

$$\begin{aligned}\mathbf{u}_1^p &= g_1^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\\ \mathbf{u}_2^p &= g_2^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\end{aligned}$$

The important difference between the previous unconstrained and this constrained case is that the functions g_1^p, g_2^p are nonlinear due to the input constraints. The system evolution is then given by

$$\begin{aligned}x_1^+ &= A_1x_1 + \overline{B}_{11}u_1 + \overline{B}_{12}u_2 & x_2^+ &= A_2x_2 + \overline{B}_{21}u_1 + \overline{B}_{22}u_2\\ \mathbf{u}_1^+ &= g_1^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) & \mathbf{u}_2^+ &= g_2^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)\end{aligned}$$

We have the following cost using the warm start at the next sample

$$\begin{aligned}V(x_1^+, x_2^+, \tilde{\mathbf{u}}_1^+, \tilde{\mathbf{u}}_2^+) = V(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) &- \rho_1\ell_1(x_1, u_1) - \rho_2\ell_2(x_2, u_2)\\ &+ (1/2)\rho_1 x_1(N)'\left[A_1'P_{1f}A_1 - P_{1f} + Q_1\right]x_1(N)\\ &+ (1/2)\rho_2 x_2(N)'\left[A_2'P_{2f}A_2 - P_{2f} + Q_2\right]x_2(N)\end{aligned}$$

Using the Schur decomposition (6.19) and the constraints $S_{ji}^{u\,\prime}x_{ji}(N) = 0$ for $i, j \in \mathbb{I}_{1:2}$, the last two terms can be written as

$$\begin{aligned}&(1/2)\rho_1 x_1(N)'S_1^s\left[A_1^{s\,\prime}\Sigma_1A_1^s - \Sigma_1 + S_1^{s\,\prime}Q_1S_1^s\right]S_1^{s\,\prime}x_1(N)\\ &+(1/2)\rho_2 x_2(N)'S_2^s\left[A_2^{s\,\prime}\Sigma_2A_2^s - \Sigma_2 + S_2^{s\,\prime}Q_2S_2^s\right]S_2^{s\,\prime}x_2(N)\end{aligned}$$

These terms are zero because of (6.20). Using this result and applying the iteration for the controllers gives

$$V(x_1^+, x_2^+, \mathbf{u}_1^+, \mathbf{u}_2^+) \leq V(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

The Lyapunov stability constraints give (see also Exercise 6.28)

$$|(\mathbf{u}_1, \mathbf{u}_2)| \leq 2 \max(d_1, d_2)\, |(x_1, x_2)| \qquad (x_1, x_2) \in r\mathcal{B}$$

Given the cost decrease and this constraint on the size of the input sequence, we satisfy the conditions of Lemma 6.5, and conclude the solution $x(k) = 0$ for all k is exponentially stable on all of $\mathcal{X}_N$ if either $\mathcal{X}_N$ is compact or $\mathbb{U}$ is compact.

6.3.2 Coupled Input Constraints

By contrast, in the coupled constraint case, the constraints are of the form

$$H_1 \mathbf{u}_1 + H_2 \mathbf{u}_2 \leq h \quad \text{or} \quad (\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{U} \tag{6.21}$$

These constraints represent the players sharing some common resource. An example would be different subsystems in a chemical plant drawing steam or some other utility from a single plantwide generation plant. The total utility used by the different subsystems to meet their control objectives is constrained by the generation capacity.

The players solve the same optimization problems as in the uncoupled constraint case, with the exception that both players' input constraints are given by (6.21). This modified game provides only two of the three properties established for the uncoupled constraint case. These are

1. The iterates are feasible: $(\mathbf{u}_1, \mathbf{u}_2)^p \in \mathbb{U}$ implies $(\mathbf{u}_1, \mathbf{u}_2)^{p+1} \in \mathbb{U}$. This follows from convexity of $\mathbb{U}$ and the convex combination of the feasible points $(\mathbf{u}_1^p, \mathbf{u}_2^p)$ and $(\mathbf{u}_1^0, \mathbf{u}_2^0)$ to make $(\mathbf{u}_1, \mathbf{u}_2)^{p+1}$.

2. The cost decreases on iteration: $V(x_1(0), x_2(0), (\mathbf{u}_1, \mathbf{u}_2)^{p+1}) \leq V(x_1(0), x_2(0), (\mathbf{u}_1, \mathbf{u}_2)^p)$ for all $x_1(0)$, $x_2(0)$, and for all feasible $(\mathbf{u}_1, \mathbf{u}_2)^p \in \mathbb{U}$. The systemwide cost satisfies the same inequalities established for the uncoupled constraint case giving

$$V(x(0), \mathbf{u}_1^{p+1}, \mathbf{u}_2^{p+1}) \leq V(x(0), \mathbf{u}_1^p, \mathbf{u}_2^p)$$

   Because the cost is bounded below, the cost iteration converges.

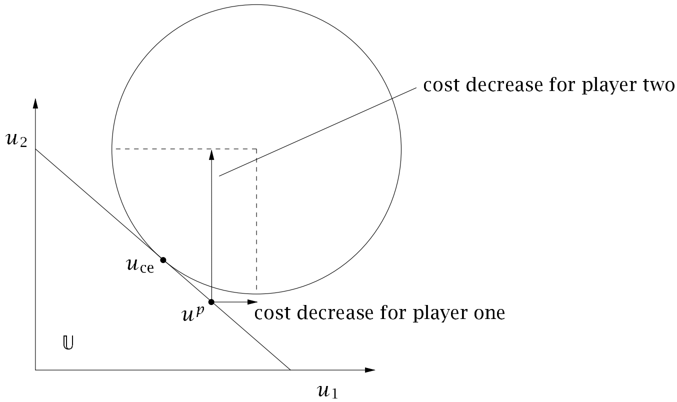

Figure 6.6: Cooperative control stuck on the boundary of $\mathbb{U}$ under coupled constraints; $u^{p+1} = u^p \neq u_{ce}$.

The converged solution of the cooperative problem is *not* equal to the optimal solution of the centralized problem, however. We have lost property 3 of the uncoupled case. To see how the convergence property is lost, consider Figure 6.6. Region $\mathbb{U}$ is indicated by the triangle and its interior. Consider point u^p on the boundary of $\mathbb{U}$. Neither player one nor player two can improve upon the current point u^p so the iteration has converged. But the converged point is clearly not the optimal point, u_{ce}.

Because of property 2, the nominal stability properties for the coupled and uncoupled cases are identical. The differences arise when the performance of cooperative control is compared to the benchmark of centralized control. Improving the performance of cooperative control in the case of coupled constraints is therefore a topic of current research. Current approaches include adding another player to the game, whose sole objective is to parcel out the coupled resource to the other players in a way that achieves optimality on iteration. This approach also makes sense from an engineering perspective because it is commonplace to design a dedicated control system for managing a shared resource such as steam or power among many plant units. The design of this single unit's control system is a reasonably narrow and well-defined task compared to the design of a centralized controller for the entire plant.

6.3.3 Exponential Convergence with Estimate Error

Consider next the constrained system evolution with estimate error

$$\begin{bmatrix} \hat{x}^+ \\ \mathbf{u}^+ \\ e^+ \end{bmatrix} = \begin{bmatrix} A\hat{x} + \overline{B}_1 u_1 + \overline{B}_2 u_2 + Le \\ g^p(\hat{x}, \mathbf{u}) \\ A_L e \end{bmatrix} \tag{6.22}$$

The estimate error is globally exponentially stable so we know from Lemma 6.7 that there exists a Lipschitz continuous Lyapunov function $J(\cdot)$ such that for all $e \in \mathbb{R}^n$

$$\overline{a}\,|e| \leq J(e) \leq \overline{b}\,|e|$$
$$J(e^+) - J(e) \leq -\overline{c}\,|e|$$

in which $\overline{b} > 0$, $\overline{a} > 0$, and we can choose constant $\overline{c} > 0$ as large as desired. In the subsequent development, we require this Lyapunov function to be based on the first power of the norm rather than the usual square of the norm to align with Lipschitz continuity of the Lyapunov function. From the stability of the solution $x(k) = 0$ for all k for the *nominal* system, the cost function $V(\hat{x}, \mathbf{u})$ satisfies for all $\hat{x} \in \mathcal{X}_N$, $\mathbf{u} \in \mathbb{U}^N$

$$\widetilde{a}\,|(\hat{x}, \mathbf{u})|^2 \leq V(\hat{x}, \mathbf{u}) \leq \widetilde{b}\,|(\hat{x}, \mathbf{u})|^2$$
$$V(A\hat{x} + \overline{B}_1 u_1 + \overline{B}_2 u_2, \mathbf{u}^+) - V(\hat{x}, \mathbf{u}) \leq -\widetilde{c}\,|\hat{x}|^2$$
$$|\mathbf{u}| \leq d\,|\hat{x}| \qquad \hat{x} \in \widetilde{r}\mathcal{B}$$

in which $\widetilde{a}, \widetilde{b}, \widetilde{c}, \widetilde{r} > 0$. We propose $W(\hat{x}, \mathbf{u}, e) = V(\hat{x}, \mathbf{u}) + J(e)$ as a Lyapunov function candidate for the perturbed system. We next derive the required properties of $W(\cdot)$ to establish exponential stability of the solution $(x(k), e(k)) = 0$. From the definition of $W(\cdot)$ we have for all $(\hat{x}, \mathbf{u}, e) \in \mathcal{X}_N \times \mathbb{U}^N \times \mathbb{R}^n$

$$\widetilde{a}\,|(\hat{x}, \mathbf{u})|^2 + \overline{a}\,|e| \leq W(\hat{x}, \mathbf{u}, e) \leq \widetilde{b}\,|(\hat{x}, \mathbf{u})|^2 + \overline{b}\,|e|$$
$$a(|(\hat{x}, \mathbf{u})|^2 + |e|) \leq W(\hat{x}, \mathbf{u}, e) \leq b(|(\hat{x}, \mathbf{u})|^2 + |e|) \tag{6.23}$$

in which $a = \min(\widetilde{a}, \overline{a}) > 0$, $b = \max(\widetilde{b}, \overline{b})$. Next we compute the cost change

$$W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) = V(\hat{x}^+, \mathbf{u}^+) - V(\hat{x}, \mathbf{u}) + J(e^+) - J(e)$$

The Lyapunov function V is quadratic in $(x, \mathbf{u})$ and therefore Lipschitz continuous on bounded sets. Therefore, for all $\hat{x}, u_1, u_2, \mathbf{u}^+, e$ in some bounded set

$$\left| V(A\hat{x} + \overline{B}_1 u_1 + \overline{B}_2 u_2 + Le, \mathbf{u}^+) - V(A\hat{x} + \overline{B}_1 u_1 + \overline{B}_2 u_2, \mathbf{u}^+) \right| \leq L_V \left| Le \right|$$

in which L_V is the Lipschitz constant for V with respect to its first argument. Using the system evolution we have

$$V(\hat{x}^+, \mathbf{u}^+) \leq V(A\hat{x} + \overline{B}_1 u_1 + \overline{B}_2 u_2, \mathbf{u}^+) + L'_V \left| e \right|$$

in which $L'_V = L_V \left| L \right|$. Subtracting $V(\hat{x}, \mathbf{u})$ from both sides gives

$$V(\hat{x}^+, \mathbf{u}^+) - V(\hat{x}, \mathbf{u}) \leq -\tilde{c} \left| \hat{x} \right|^2 + L'_V \left| e \right|$$

Substituting this result into the equation for the change in W gives

$$\begin{aligned} W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) &\leq -\tilde{c} \left| \hat{x} \right|^2 + L'_V \left| e \right| - \overline{c} \left| e \right| \\ &\leq -\tilde{c} \left| \hat{x} \right|^2 - (\overline{c} - L'_V) \left| e \right| \\ W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) &\leq -c(\left| \hat{x} \right|^2 + \left| e \right|) \end{aligned} \tag{6.24}$$

in which we choose $\overline{c} > L'_V$, which is possible because we may choose $\overline{c}$ as large as we wish, and $c = \min(\tilde{c}, \overline{c} - L'_V) > 0$. Notice this step is what motivated using the first power of the norm in $J(\cdot)$. Lastly, we require the constraint

$$\left| \mathbf{u} \right| \leq d \left| \hat{x} \right| \qquad \hat{x} \in \tilde{r}\mathcal{B} \tag{6.25}$$

Lemma 6.14 (Global asymptotic stability and exponential convergence of perturbed system)**.** *If either* $\mathcal{X}_N$ *or* $\mathbb{U}$ *is compact, there exist* $\lambda < 1$ *and* $\delta(\cdot) \in \mathcal{K}_\infty$ *such that the combined system* (6.22) *satisfies for all* $(x(0), e(0))$ *and* $k \geq 0$

$$\left| x(k), e(k) \right| \leq \delta(\left| x(0), e(0) \right|)\lambda^k$$

The proof is based on the properties (6.23), (6.24), and (6.25) of function $W(\hat{x}, \mathbf{u}, e)$, and is basically a combination of the proofs of Lemmas 6.5 and 6.6. The region of attraction is the set of states and initial estimate errors for which the unstable modes of the two subsystems can be brought to zero in N moves while satisfying the respective input constraints. If both subsystems are stable, for example, the region of attraction is $(x, e) \in \mathcal{X}_N \times \mathbb{R}^n$.

6.3.4 Disturbance Models and Zero Offset

Integrating disturbance model. As discussed in Chapter 1, we model the disturbance with an integrator to remove steady offset. The augmented models for the local systems are

$$\begin{bmatrix} x_i \\ d_i \end{bmatrix}^+ = \begin{bmatrix} A_i & B_{di} \\ 0 & I \end{bmatrix} \begin{bmatrix} x_i \\ d_i \end{bmatrix} + \begin{bmatrix} \overline{B}_{i1} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} \overline{B}_{i2} \\ 0 \end{bmatrix} u_2$$
$$y_i = \begin{bmatrix} C_i & C_{di} \end{bmatrix} \begin{bmatrix} x_i \\ d_i \end{bmatrix} \qquad i = 1, 2$$

We wish to estimate both x_i and d_i from measurements y_i. To ensure this goal is possible, we make the following restriction on the disturbance models.

Assumption 6.15 (Disturbance models)**.**

$$\operatorname{rank} \begin{bmatrix} I - A_i & -B_{di} \\ C_i & C_{di} \end{bmatrix} = n_i + p_i \qquad i = 1, 2$$

It is always possible to satisfy this assumption by proper choice of B_{di}, C_{di}. From Assumption 6.13 (b), (A_i, C_i) is detectable, which implies that the first n_i columns of the square $(n_i + p_i) \times (n_i + p_i)$ matrix in Assumption 6.15 are linearly independent. Therefore the columns of $\left[\begin{smallmatrix} -B_{di} \\ C_{di} \end{smallmatrix}\right]$ can be chosen so that the entire matrix has rank $n_i + p_i$. Assumption 6.15 is equivalent to detectability of the following augmented system.

Lemma 6.16 (Detectability of distributed disturbance model)**.** *Consider the augmented systems*

$$\widetilde{A}_i = \begin{bmatrix} A_i & B_{di} \\ 0 & I \end{bmatrix} \qquad \widetilde{C}_i = \begin{bmatrix} C_i & C_{di} \end{bmatrix} \qquad i = 1, 2$$

The augmented systems $(\widetilde{A}_i, \widetilde{C}_i), i = 1, 2$ are detectable if and only if Assumption 6.15 is satisfied.

Proving this lemma is discussed in Exercise 6.29. The detectability assumption then establishes the existence of $\widetilde{L}_i$ such that $(\widetilde{A}_i - \widetilde{L}_i \widetilde{C}_i)$, $i = 1, 2$ are stable and the local integrating disturbances can be estimated from the local measurements.

Centralized target problem. We can solve the target problem at the plantwide level or as a distributed target problem at the subunit controller level. Consider first the centralized target problem with the disturbance model discussed in Chapter 1, (1.45)

$$\min_{x_s, u_s} \frac{1}{2} \left| u_s - u_{\text{sp}} \right|^2_{R_s} + \frac{1}{2} \left| C x_s + C_d \hat{d}(k) - y_{\text{sp}} \right|^2_{Q_s}$$

subject to

$$\begin{bmatrix} I - A & -B \\ HC & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} B_d \hat{d}(k) \\ r_{\text{sp}} - HC_d \hat{d}(k) \end{bmatrix}$$
$$E u_s \leq e$$

in which we have removed the state inequality constraints to be consistent with the regulator problem. We denote the solution to this problem $(x_s(k), u_s(k))$. Notice first that the solution of the target problem depends only on the disturbance estimate, $\hat{d}(k)$, and not the solution of the control problem. So we can analyze the behavior of the target by considering only the exponential convergence of the estimator. We restrict the plant disturbance d so that the target problem is feasible, and denote the solution to the target problem for the plant disturbance, $\hat{d}(k) = d$, as (x_s^*, u_s^*). Because the estimator is exponentially stable, we know that $\hat{d}(k) \to d$ as $k \to \infty$. Because the target problem is a positive definite quadratic program (QP), we know the solution is Lipschitz continuous on bounded sets in the term $\hat{d}(k)$, which appears linearly in the objective function and the right-hand side of the equality constraint. Therefore, if we also restrict the initial disturbance estimate error so that the target problem remains feasible for all time, we know $(x_s(k), u_s(k)) \to (x_s^*, u_s^*)$ and the rate of convergence is exponential.

Distributed target problem. Consider next the cooperative approach, in which we assume the input inequality constraints are uncoupled. In the constrained case, we try to set things up so each player solves a local target problem

$$\min_{x_{1s}, u_{1s}} \frac{1}{2} \begin{bmatrix} y_{1s} - y_{1sp} \\ y_{2s} - y_{2sp} \end{bmatrix}' \begin{bmatrix} Q_{1s} & \\ & Q_{2s} \end{bmatrix} \begin{bmatrix} y_{1s} - y_{1sp} \\ y_{2s} - y_{2sp} \end{bmatrix} +$$
$$\frac{1}{2} \begin{bmatrix} u_{1s} - u_{1sp} \\ u_{2s} - u_{2sp} \end{bmatrix}' \begin{bmatrix} R_{1s} & \\ & R_{2s} \end{bmatrix} \begin{bmatrix} u_{1s} - u_{1sp} \\ u_{2s} - u_{2sp} \end{bmatrix}$$

subject to

$$\begin{bmatrix} I - A_1 & & -\overline{B}_{11} & -\overline{B}_{12} \\ & I - A_2 & -\overline{B}_{21} & -\overline{B}_{22} \\ H_1 C_1 & & & \\ & H_2 C_2 & & \end{bmatrix} \begin{bmatrix} x_{1s} \\ x_{2s} \\ u_{1s} \\ u_{2s} \end{bmatrix} = \begin{bmatrix} B_{d1}\hat{d}_1(k) \\ B_{d2}\hat{d}_2(k) \\ r_{1sp} - H_1 C_{d1}\hat{d}_1(k) \\ r_{2sp} - H_2 C_{d2}\hat{d}_2(k) \end{bmatrix}$$

$$E_1 u_{1s} \leq e_1$$

in which

$$y_{1s} = C_1 x_{1s} + C_{d1}\hat{d}_1(k) \qquad y_{2s} = C_2 x_{2s} + C_{d2}\hat{d}_2(k) \tag{6.27}$$

But here we run into several problems. First, the constraints to ensure zero offset in both players' controlled variables are not feasible with only the u_{1s} decision variables. We require also u_{2s}, which is not available to player one. We can consider deleting the zero offset condition for player two's controlled variables, the last equality constraint. But if we do that for both players, then the two players have *different and coupled* equality constraints. That is a path to instability as we have seen in the noncooperative target problem. To resolve this issue, we move the controlled variables to the objective function, and player one solves instead the following

$$\min_{x_{1s}, u_{1s}} \frac{1}{2} \begin{bmatrix} H_1 y_{1s} - r_{1sp} \\ H_2 y_{2s} - r_{2sp} \end{bmatrix}' \begin{bmatrix} T_{1s} & \\ & T_{2s} \end{bmatrix} \begin{bmatrix} H_1 y_{1s} - r_{1sp} \\ H_2 y_{2s} - r_{2sp} \end{bmatrix}$$

subject to (6.27) and

$$\begin{bmatrix} I - A_1 & & -\overline{B}_{11} & -\overline{B}_{12} \\ & I - A_2 & -\overline{B}_{21} & -\overline{B}_{22} \end{bmatrix} \begin{bmatrix} x_{1s} \\ x_{2s} \\ u_{1s} \\ u_{2s} \end{bmatrix} = \begin{bmatrix} B_{d1}\hat{d}_1(k) \\ B_{d2}\hat{d}_2(k) \end{bmatrix}$$

$$E_1 u_{1s} \leq e_1 \tag{6.28}$$

The equality constraints for the two players appear coupled when written in this form. Coupled constraints admit the potential for the optimization to become stuck on the boundary of the feasible region, and not achieve the centralized target solution after iteration to convergence. But Exercise 6.30 discusses how to show that the equality constraints are, in fact, uncoupled. Also, the distributed target problem as expressed here may not have a unique solution when there are more manipulated variables than controlled variables. In such cases, a regularization term using the input setpoint can be added to the objective function. The controlled variable penalty can be converted to a linear penalty with a large penalty weight to ensure exact satisfaction of the controlled variable setpoint.

If the input inequality constraints are coupled, however, then the distributed target problem may indeed become stuck on the boundary of the feasible region and not eliminate offset in the controlled variables. If the input inequality constraints are coupled, we recommend using the centralized approach to computing the steady-state target. As discussed above, the centralized target problem eliminates offset in the controlled variables as long as it remains feasible given the disturbance estimates.

Zero offset. Finally we establish the zero offset property. As described in Chapter 1, the regulator is posed in deviation variables

$$\widetilde{x}(k) = \hat{x}(k) - x_s(k) \qquad \widetilde{u}(k) = u(k) - u_s(k) \qquad \widetilde{\mathbf{u}} = \mathbf{u} - u_s(k)$$

in which the notation $\mathbf{u} - u_s(k)$ means to subtract $u_s(k)$ from each element of the $\mathbf{u}$ sequence. Player one then solves

$$\begin{aligned} &\min_{\widetilde{\mathbf{u}}_1} V(\widetilde{x}_1(0), \widetilde{x}_2(0), \widetilde{\mathbf{u}}_1, \widetilde{\mathbf{u}}_2) \\ \text{s.t.} \quad &\begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_{11} \\ \overline{B}_{21} \end{bmatrix} \widetilde{u}_1 + \begin{bmatrix} \overline{B}_{12} \\ \overline{B}_{22} \end{bmatrix} \widetilde{u}_2 \\ &\widetilde{\mathbf{u}}_1 \in \mathbb{U}_1 \ominus u_s(k) \\ &S_{1u}' \widetilde{x}_1(N) = 0 \\ &\left| \widetilde{\mathbf{u}}_1 \right| \leq d_1 \left| \widetilde{x}_1(0) \right| \end{aligned}$$

Notice that because the input constraint is shifted by the input target, we must retain feasibility of the regulation problem by restricting also the plant disturbance and its initial estimate error. If the two players' regulation problems remain feasible as the estimate error converges to zero, we have exponential stability of the zero solution from Lemma 6.14. Therefore we conclude

$$\begin{aligned} & (\widetilde{x}(k), \widetilde{u}(k)) \to (0,0) && \text{Lemma 6.14} \\ \Longrightarrow\ & (\hat{x}(k), u(k)) \to (x_s(k), u_s(k)) && \text{definition of deviation variables} \\ \Longrightarrow\ & (\hat{x}(k), u(k)) \to (x_s^*, u_s^*) && \text{target problem convergence} \\ \Longrightarrow\ & x(k) \to x_s^* && \text{estimator stability} \\ \Longrightarrow\ & r(k) \to r_{\text{sp}} && \text{target equality constraint} \end{aligned}$$

and we have *zero offset* in the plant controlled variable $r = Hy$. The rate of convergence of $r(k)$ to r_{sp} is also exponential. As we saw here, this convergence depends on maintaining feasibility in both the target problem and the regulation problem at all times.

6.4 Constrained M-Player Game

We have set up the constrained two-player game so that the approach generalizes naturally to the M-player game. We do not have a lot of work left to do to address this general case. Recall $\mathbb{I}_{1:M}$ denotes the set of integers $\{1, 2, \ldots, M\}$. We define the following systemwide variables

$$x(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_M(0) \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_M \end{bmatrix} \qquad B_i = \begin{bmatrix} \overline{B}_{1i} \\ \overline{B}_{2i} \\ \vdots \\ \overline{B}_{Mi} \end{bmatrix} \quad \underline{B}_i = \begin{bmatrix} B_{1i} \\ B_{2i} \\ \vdots \\ B_{Mi} \end{bmatrix} \quad i \in \mathbb{I}_{1:M}$$

$$V(x(0), \mathbf{u}) = \sum_{j \in \mathbb{I}_{1:M}} \rho_j V_j(x_j(0), \mathbf{u})$$

Each player solves a similar optimization, so for $i \in \mathbb{I}_{1:M}$

$$\min_{\mathbf{u}_i} V(x(0), \mathbf{u})$$

$$\text{s.t. } x^+ = Ax + \sum_{j \in \mathbb{I}_{1:M}} B_j u_j$$

$$\mathbf{u}_i \in \mathbb{U}_i$$

$$S_{ji}^{u\prime} x_{ji}(N) = 0 \quad j \in \mathbb{I}_{1:M}$$

$$|\mathbf{u}_i| \leq d_i \sum_{j \in \mathbb{I}_{1:M}} \left| x_{ji}(0) \right| \qquad \text{if } x_{ji}(0) \in r\mathcal{B},\ j \in \mathbb{I}_{1:M}$$

This optimization can be expressed as a quadratic program, whose constraints and linear cost term depend affinely on parameter x. The warm start for each player at the next sample is generated from purely local information

$$\tilde{\mathbf{u}}_i^+ = (u_i(1), u_i(2), \ldots, u_i(N-1), 0) \quad i \in \mathbb{I}_{1:M}$$

The controller iteration is given by

$$\mathbf{u}^{p+1} = \sum_{j \in \mathbb{I}_{1:M}} w_j \left(\mathbf{u}_1^p, \ldots, \mathbf{u}_j^0, \ldots, \mathbf{u}_M^p \right)$$

in which $\mathbf{u}_i^0 = \mathbf{u}_i^0\left(x(0), \mathbf{u}^p_{j\in\mathbb{I}_{1:M}, j\neq i}\right)$. The plantwide cost function then satisfies for any $p \geq 0$

$$V(x^+, \mathbf{u}^+) \leq V(x, \mathbf{u}) - \sum_{j\in\mathbb{I}_{1:M}} \rho_j \ell_j(x_j, u_j)$$
$$|\mathbf{u}| \leq d\,|x| \qquad x \in r\mathcal{B}$$

For the M-player game, we generalize Assumption 6.13 of the two-player game to the following.

Assumption 6.17 (Constrained M-player game)**.**

(a) The systems $(\underline{A}_i, \underline{B}_i)$, $i \in \mathbb{I}_{1:M}$ are stabilizable, in which $\underline{A}_i = \text{diag}(A_{1i}, A_{2i}, \cdots, A_{Mi})$.

(b) The systems (A_i, C_i), $i \in \mathbb{I}_{1:M}$ are detectable.

(c) The input penalties R_i, $i \in \mathbb{I}_{1:M}$ are positive definite, and Q_i, $i \in \mathbb{I}_{1:M}$ are semidefinite.

(d) The systems (A_i, Q_i), $i \in \mathbb{I}_{1:M}$ are detectable.

(e) The horizon is chosen sufficiently long to zero the unstable modes; $N \geq \max_{i\in\mathbb{I}_{1:M}}(\underline{n}_i^u)$, in which $\underline{n}_i^u$ is the number of unstable modes of $\underline{A}_i$.

(f) Zero offset. For achieving zero offset, we augment the models with integrating disturbances such that

$$\text{rank}\begin{bmatrix} I - A_i & -B_{di} \\ C_i & C_{di} \end{bmatrix} = n_i + p_i \qquad i \in \mathbb{I}_{1:M}$$

Applying Theorem 6.5 then establishes exponential stability of the solution $x(k) = 0$ for all k. The region of attraction is the set of states for which the unstable modes of each subsystem can be brought to zero in N moves, while satisfying the respective input constraints. These conclusions apply regardless of how many iterations of the players' optimizations are used in the control calculation. Although the closed-loop system is exponentially stable for both coupled and uncoupled constraints, the converged distributed controller is equal to the centralized controller only for the case of uncoupled constraints.

The exponential stability of the regulator implies that the states and inputs of the constrained M-player system converge to the steady-state target. The steady-state target can be calculated as a centralized or distributed problem. We assume the centralized target has a feasible, zero offset solution for the true plant disturbance. The initial state of the plant and the estimate error must be small enough that feasibility of the target is maintained under the nonzero estimate error.

6.5 Nonlinear Distributed MPC

In the nonlinear case, the usual model comes from physical principles and conservation laws of mass, energy, and momentum. The state has a physical meaning and the measured outputs usually are a subset of the state. We assume the model is of the form

$$\frac{dx_1}{dt} = f_1(x_1, x_2, u_1, u_2) \qquad y_1 = C_1 x_1$$
$$\frac{dx_2}{dt} = f_2(x_1, x_2, u_1, u_2) \qquad y_2 = C_2 x_2$$

in which C_1, C_2 are matrices of zeros and ones selecting the part of the state that is measured in subsystems one and two. We generally cannot avoid state x_2 dependence in the differential equation for x_1. But often only a small subset of the entire state x_2 appears in f_1, and vice versa. The reason in chemical process systems is that the two subsystems are generally coupled through a small set of process streams transferring mass and energy between the systems. These connecting streams isolate the coupling between the two systems and reduce the influence to a small part of the entire state required to describe each system.

Given these physical system models of the subsystems, the overall plant model is

$$\frac{dx}{dt} = f(x, u) \qquad y = Cx$$

with

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad C = \begin{bmatrix} C_1 & \\ & C_2 \end{bmatrix}$$

6.5.1 Nonconvexity

The basic difficulty in both the theory and application of nonlinear MPC is the nonconvexity in the control objective function caused by the nonlinear dynamic model. This difficulty applies even to centralized nonlinear MPC as discussed in Section 2.7, and motivates the development of suboptimal MPC. In the distributed case, nonconvexity causes extra

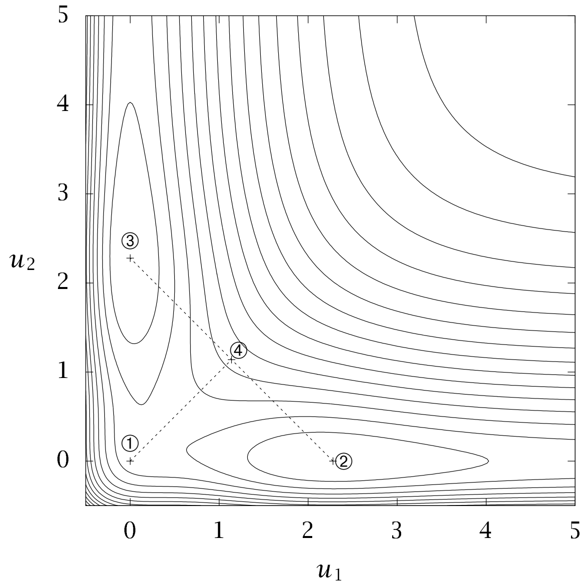

Figure 6.7: Cost contours for a two-player, nonconvex game; cost *increases* for the convex combination of the two players' optimal points.

difficulties. As an illustration, consider the simple two-player, nonconvex game depicted in Figure 6.7. The cost function is

$$V(u_1, u_2) = e^{-2u_1} - 2e^{-u_1} + e^{-2u_2} - 2e^{-u_2} \\ + a\exp(-\beta((u_1 + 0.2)^2 + (u_2 + 0.2)^2))$$

in which $a = 1.1$ and $\beta = 0.4$. Each player optimizes the cooperative objective starting at ① and produces the points (u_1^0, u_2^p), denoted ②, and (u_1^p, u_2^0), denoted ③. Consider taking a convex combination of the two players' optimal points for the next iterate

$$(u_1^{p+1}, u_2^{p+1}) = w_1(u_1^0, u_2^p) + w_2(u_1^p, u_2^0) \qquad w_1 + w_2 = 1, \quad w_1, w_2 \geq 0$$

We see in Figure 6.7 that this iterate causes the objective function to *increase* rather than decrease for most values of w_1, w_2. For $w_1 = w_2 = 1/2$, we see clearly from the contours that V at point ④ is greater than V at point ①.

The possibility of a cost increase leads to the possibility of closed-loop instability and precludes developing even a nominal control theory for this simple approach, which was adequate for the convex, linear plant case.[4] In the centralized MPC problem, this nonconvexity issue can be addressed in the optimizer, which can move both inputs simultaneously and always avoid a cost increase. One can of course consider adding another player to the game who has access to more systemwide information. This player takes the optimization results of the individual players and determines a search direction and step length that achieve a cost decrease for the overall system. This player is often known as a coordinator. The main drawback of this approach is that the design of the coordinator may not be significantly simpler than the design of the centralized controller.

Rather than design a coordinator, we instead let each player evaluate the effect of taking a combination of all the players' optimal moves. The players can then easily find an effective combination that leads to a cost decrease. We describe one such algorithm in the next section, which we call the *distributed gradient algorithm.*

6.5.2 Distributed Algorithm for Nonconvex Functions

We consider the problem

$$\min_u V(u) \quad \text{s.t.} \quad u \in \mathbb{U} \tag{6.29}$$

in which $u \in \mathbb{R}^m$ and $V : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is twice continuously differentiable and not necessarily convex. We assume $\mathbb{U}$ is closed, convex, and can be expressed as $\mathbb{U} = \mathbb{U}_1 \times \cdots \times \mathbb{U}_M$ with $\mathbb{U}_i \in \mathbb{R}^{m_i}$ for all $i \in \mathbb{I}_{1:M}$. We solve approximately the following subproblems at iterate $p \geq 0$ for all $i \in \mathbb{I}_{1:M}$

$$\min_{u_i \in \mathbb{U}_i} V(u_i, u_{-i}^p)$$

in which $u_{-i} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_M)$. Let $\overline{u}_i^p$ denote the approximate solution to these optimizations. We compute the approximate solutions via the standard technique of line search with gradient projection. At iterate $p \geq 0$

$$\overline{u}_i^p = \mathcal{P}_i(u_i^p - \nabla_i V(u^p)) \tag{6.30}$$

[4]This point marked the state of affairs at the time of publication of the first edition of this text. The remaining sections summarize one approach that addresses the nonconvexity problem (Stewart, Wright, and Rawlings, 2011).

in which $\nabla_i V(u^p)$ is the ith component of $\nabla V(u^p)$ and $\mathcal{P}_i(\cdot)$ denotes projection onto the set $\mathbb{U}_i$. Define the step $v_i^p = \overline{u}_i^p - u_i^p$. The stepsize α_i^p is chosen as follows; each suboptimizer initializes the stepsize with $\overline{\alpha}_i$, and then uses backtracking until α_i^p satisfies the Armijo rule (Bertsekas, 1999, p.230)

$$V(u^p) - V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p) \geq -\sigma \alpha_i^p \nabla_i V(u^p)' v_i^p \tag{6.31}$$

in which $\sigma \in (0,1)$. After all suboptimizers finish backtracking, they exchange proposed steps. Each suboptimizer forms a candidate step

$$u_i^{p+1} = u_i^p + w_i \alpha_i^p v_i^p \quad \forall i \in \mathbb{I}_{1:M} \tag{6.32}$$

and checks the following inequality

$$V(u^{p+1}) \leq \sum_{i \in \mathbb{I}_{1:M}} w_i V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p) \tag{6.33}$$

with $\sum_{i \in \mathbb{I}_{1:M}} w_i = 1$ and $w_i > 0$ for all $i \in \mathbb{I}_{1:M}$. If condition (6.33) is not satisfied, then we remove the direction with the least cost improvement, $i_{\max} = \arg\max_i \{V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p)\}$, by setting $w_{i_{\max}}$ to zero and repartitioning the remaining w_i so that they sum to one. The candidate step (6.32) is recalculated and condition (6.33) is checked again. This process is repeated until (6.33) is satisfied. It may happen that condition (6.33) is satisfied with only a single direction. The distributed algorithm thus eliminates poor suboptimizer steps and ensures that the objective function decreases at each iterate, even for nonconvex objective functions. The proposed algorithm has the following properties.

Lemma 6.18 (Distributed gradient algorithm properties)**.** *The distributed gradient projection algorithm has the following properties.*

(a) (Feasibility.) Given a feasible initial condition, the iterates u^p are feasible for all $p \geq 0$.

(b) (Objective decrease.) The objective function decreases at every iterate: $V(u^{p+1}) \leq V(u^p)$.

(c) (Convergence.) Every accumulation point of the sequence $(u^p)_{p \geq 0}$ is a stationary point.

The proof of Lemma 6.18 is given in Stewart et al. (2011). Note that the test of inequality (6.33) does not require a coordinator. At each iteration the subsystems exchange the solutions of the gradient

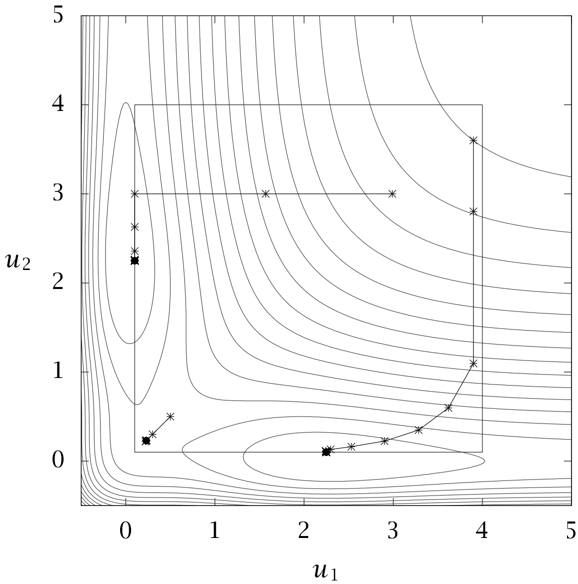

Figure 6.8: Nonconvex function optimized with the distributed gradient algorithm. Iterations converge to local minima from all starting points.

projection. Because each subsystem has access to the plantwide model, they can evaluate the objection function, and the algorithm can be run independently on each controller. This computation is likely a smaller overhead than a coordinating optimization.

Figure 6.8 shows the results of applying the proposed distributed gradient algorithm to the previous example. The problem has two global minima located at $(0.007, 2.28)$ and $(2.28, 0, 007)$, and a local minimum at $(0.23, 0.23)$. The inputs are constrained: $0.1 \leq u_i \leq 4$ for $i \in \mathbb{I}_{1:2}$. The algorithm is initialized at three different starting points $(0.5, 0.5)$, $(3.9, 3.6)$, and $(2.99, 3)$. From Figure 6.8 we see that each of these starting points converges to a different local minimum.

6.5.3 Distributed Nonlinear Cooperative Control

Next we design a controller based on the distributed optimization algorithm. For simplicity of presentation, we assume that the plant consists of two subsystems. We consider the standard MPC cost function for each system $i \in \mathbb{I}_{1:2}$

$$V_i(x(0), \mathbf{u}_1, \mathbf{u}_2) = \sum_{k=0}^{N-1} \ell_i(x_i(k), u_i(k)) + V_{if}(x(N))$$

with $\ell_i(x_i, u_i)$ denoting the stage cost, $V_{if}(x)$ the terminal cost of system i, and $x_i(i) = \phi_i(k; x_i, \mathbf{u}_1, \mathbf{u}_2)$. Because x_i is a function of both u_1 and u_2, V_i is a function of both $\mathbf{u}_1$ and $\mathbf{u}_2$. As in the case for linear plants, we define the plantwide objective

$$V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) = \rho_1 V_1(x(0), \mathbf{u}_1, \mathbf{u}_2) + \rho_2 V_2(x(0), \mathbf{u}_1, \mathbf{u}_2)$$

in which $\rho_1, \rho_2 > 0$ are weighting factors. To simplify notation we use $V(x, \mathbf{u})$ to denote the plantwide objective. Similarly we define the system stage cost and terminal cost as the combined stage costs $\ell(x, u) := \rho_1 \ell_1(x_1, u_1) + \rho_2 \ell_2(x_2, u_2)$, and terminal costs $V_f(x) := \rho_1 V_{1f}(x) + \rho_2 V_{2f}(x)$. Each subsystem has constraints of the form

$$u_1(k) \in \mathbb{U}_1 \qquad u_2(k) \in \mathbb{U}_2 \quad k \in \mathbb{I}_{0:N-1}$$

in which each $\mathbb{U}_i \in \mathbb{R}^{m_i}$ is compact, convex, and contains the origin. Finally, we define the terminal region $\mathbb{X}_f$ to be a sublevel set of V_f.

$$\mathbb{X}_f = \{x \mid V_f(x) \le a\}$$

for some $a > 0$.

We next modify the standard stability assumption to account for the distributed nature of the problem.

Assumption 6.19 (Basic stability assumption (distributed))**.** $V_f(\cdot)$, $\mathbb{X}_f$, and $\ell(\cdot)$ have the following properties.

(a) For all $x \in \mathbb{X}_f$, there exists (u_1, u_2) (such that $(x, u_1, u_2) \in \mathbb{R}^n \times U_1 \times U_2$) satisfying

$$\begin{aligned} &f(x, u_1, u_2) \in \mathbb{X}_f \\ &V_f(f(x, u_1, u_2)) - V_f(x) \le -\ell(x, u_1, u_2) \end{aligned}$$

(b) For each $i \in \mathbb{I}_{1:2}$, there exist $\mathcal{K}_\infty$ functions $\alpha_i(\cdot)$, and $\alpha_f(\cdot)$ satisfying

$$\begin{aligned} \ell_i(x_i, u_i) &\ge \alpha_i(|x_i|) && \forall (x_i, u_i) \in \mathcal{X}_N \times \mathbb{U}_i \\ V_f(x) &\le \alpha_f(|x|) && \forall x \in \mathbb{X}_f \end{aligned}$$

This assumption implies that there exist local controllers $\kappa_{if} : \mathbb{X}_f \to \mathbb{U}_i$ for all $i \in \mathbb{I}_{1:2}$ such that for all $x \in \mathbb{X}_f$

$$V_f(f(x, \kappa_{1f}(x), \kappa_{2f}(x))) - V_f(x) \leq -\ell(x, \kappa_{1f}(x), \kappa_{2f}(x)) \tag{6.34}$$

with $f(x, \kappa_{1f}(x), \kappa_{2f}(x)) \in \mathbb{X}_f$. Each terminal controller $\kappa_{if}(\cdot)$ may be found offline.

Removing the terminal constraint in suboptimal MPC. To show stability, we require that $\phi(N; x, \mathbf{u}) \in \mathbb{X}_f$. But the terminal constraint on the state shows up as a coupled input constraint in each subsystem's optimization problem. As we have already discussed, coupled input constraints may prevent the distributed algorithm from converging to the optimal plantwide control (Stewart, Venkat, Rawlings, Wright, and Pannocchia, 2010). The terminal constraint can be removed from the control problem by modifying the terminal penalty, however, as we demonstrate next.

For some $\beta \geq 1$, we define the objective function

$$V^{\beta}(x, \mathbf{u}) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) + \beta V_f(x(N)) \tag{6.35}$$

and the set of admissible initial $(x, \mathbf{u})$ as

$$\mathbb{Z}_0 = \{(x, \mathbf{u}) \in \mathbb{X} \times \mathbb{U}^N \mid V^{\beta}(x, \mathbf{u}) \leq \overline{V},\ \phi(N; x, \mathbf{u}) \in \mathbb{X}_f\} \tag{6.36}$$

in which $\overline{V} > 0$ is an arbitrary constant and $\mathbb{X} = \mathbb{R}^n$. The set of initial states $\mathbb{X}_0$ is the projection of $\mathbb{Z}_0$ onto $\mathbb{X}$

$$\mathbb{X}_0 = \{x \in \mathbb{X} \mid \exists \mathbf{u} \text{ such that } (x, \mathbf{u}) \in \mathbb{Z}_0\}$$

We have the following result.

Proposition 6.20 (Terminal constraint satisfaction)**.** *Let* $\{(x(k), \mathbf{u}(k)) \mid k \in \mathbb{I}_{\geq 0}\}$ *denote the set of states and control sequences generated by the suboptimal system. There exists a* $\overline{\beta} > 1$ *such that for all* $\beta \geq \overline{\beta}$*, if* $(x(0), \mathbf{u}(0)) \in \mathbb{Z}_0$*, then* $(x(k), \mathbf{u}(k)) \in \mathbb{Z}_0$ *with* $\phi(N; x(k), \mathbf{u}(k)) \in \mathbb{X}_f$ *for all* $k \in \mathbb{I}_{\geq 0}$*.*

The proof of this proposition is given in Stewart et al. (2011). We are now ready to define the cooperative control algorithm for nonlinear systems.

Cooperative control algorithm. Let $x(0)$ be the initial state and $\tilde{\mathbf{u}} \in \mathbb{U}$ be the initial feasible input sequence for the cooperative MPC algorithm such that $\phi(N; x(0), \tilde{\mathbf{u}}) \in \mathbb{X}_f$. At each iterate p, an approximate solution of the following optimization problem is computed

$$\begin{aligned}
\min_{\mathbf{u}} \quad & V(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2) \\
\text{s.t.} \quad & x_1^+ = f_1(x_1, x_2, u_1, u_2) \\
& x_2^+ = f_2(x_1, x_2, u_1, u_2) \\
& \mathbf{u}_i \in \mathbb{U}_i^N \quad \forall i \in \mathbb{I}_{1:2} \\
& |\mathbf{u}_i| \leq \delta_i(|x_i(0)|) \quad \text{if } x(0) \in \mathcal{B}r \quad \forall i \in \mathbb{I}_{1:2}
\end{aligned} \tag{6.37}$$

in which $\delta_i(\cdot) \in \mathcal{K}_\infty$ and $r > 0$ can be chosen as small as desired. We can express (6.37) in the form of (6.29) by eliminating the model equality constraints. To implement distributed control, we simply use the distributed gradient algorithm to solve (6.37).

Denote the solution returned by the algorithm as $\mathbf{u}^{\bar{p}}(x, \tilde{\mathbf{u}})$. The first element of the sequence, denoted $\kappa^{\bar{p}}(x(0)) = u^{\bar{p}}(0; x(0), \tilde{\mathbf{u}})$, is injected into the plant. To reinitialize the algorithm at the next sample time, we compute the warm start

$$\begin{aligned}
\tilde{\mathbf{u}}_1^+ &= \{u_1(1), u_1(2), \ldots, u_1(N-1), \kappa_{1f}(x(N))\} \\
\tilde{\mathbf{u}}_2^+ &= \{u_2(1), u_2(2), \ldots, u_2(N-1), \kappa_{2f}(x(N))\}
\end{aligned}$$

in which $x(N) = \phi(N; x(0), \mathbf{u}_1, \mathbf{u}_2)$. We expect that it is not possible to solve (6.37) to optimality in the available sample time, and the distributed controller is therefore a form of suboptimal MPC. The properties of the closed-loop system are therefore analyzed using suboptimal MPC theory.

6.5.4 Stability of Distributed Nonlinear Cooperative Control

We first show that the plantwide objective function decreases between sampling times. Let $(x, \mathbf{u})$ be the state and input sequence at some time. Using the warm start as the initial condition at the next sample time, we have

$$\begin{aligned}
V(x^+, \tilde{\mathbf{u}}^+) = V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2) \\
-\rho_1 V_{1f}(x(N)) - \rho_2 V_{2f}(x(N)) \\
+\rho_1 \ell_1(x_1(N), \kappa_{1f}(x(N))) + \rho_2 \ell_2(x_2(N), \kappa_{2f}(x(N))) \\
+\rho_1 V_{1f}\Big(f_1(x_1(N), x_2(N), \kappa_{1f}(x(N)), \kappa_{2f}(x(N)))\Big) \\
+\rho_2 V_{2f}\Big(f_2(x_1(N), x_2(N), \kappa_{1f}(x(N)), \kappa_{2f}(x(N)))\Big)
\end{aligned}$$

From (6.34) of the stability assumption, we have that

$$V(x^+, \tilde{\mathbf{u}}^+) \le V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

By Lemma 6.18(b), the objective function cost only decreases from this warm start, so that

$$V(x^+, \mathbf{u}^+) \le V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

and we have the required cost decrease of a Lyapunov function

$$V(x^+, \mathbf{u}^+) - V(x, \mathbf{u}) \le -\alpha(|(x, u)|) \tag{6.38}$$

in which $\alpha(|(x,u)|) = \rho_1 \alpha_1(|(x_1, u_1)|) + \rho_2 \alpha_2(|(x_2, u_2)|)$.

We can now state the main result. Let $\mathcal{X}_N$ be the admissible set of initial states for which the control optimization (6.37) is feasible.

Theorem 6.21 (Asymptotic stability)**.** *Let Assumptions 2.2, 2.3, and 6.19 hold, and let* $V(\cdot) \leftarrow V^{\overline{\beta}}(\cdot)$ *from Proposition 6.20. Then for every* $x(0) \in \mathcal{X}_N$*, the origin is asymptotically stable for the closed-loop system* $x^+ = f(x, \kappa^{\overline{p}}(x))$*.*

The proof follows, with minor modification, the proof that suboptimal MPC is asymptotically stable in Theorem 2.48. As in the previous sections, the controller has been presented for the case of two subsystems, but can be extended to any finite number of subsystems.

We conclude the discussion of nonlinear distributed MPC by revisiting the unstable nonlinear example system presented in Stewart et al. (2011).

Example 6.22: Nonlinear distributed control

We consider the unstable nonlinear system

$$\begin{aligned}
x_1^+ &= x_1^2 + x_2 + u_1^3 + u_2 \\
x_2^+ &= x_1 + x_2^2 + u_1 + u_2^3
\end{aligned}$$

with initial condition $(x_1, x_2) = (3, -3)$. The control objective is to stabilize the system and regulate the states to the origin. We use a standard quadratic stage cost

$$\ell_1(x_1, u_1) = \frac{1}{2}(x_1' Q_1 x_1 + u_1' R_1 u_1)$$
$$\ell_2(x_2, u_2) = \frac{1}{2}(x_2' Q_2 x_2 + u_2' R_2 u_2)$$

with $Q_1, Q_2 > 0$ and $R_1, R_2 > 0$. This stage cost gives the objective function

$$V(x, \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{N-1} x(k)' Q x(k) + u(k)' R u(k) + V_f(x(N))$$

in which $Q = \mathrm{diag}(Q_1, Q_2)$, $R = \mathrm{diag}(R_1, R_2)$. The terminal penalty is defined in the standard way for centralized MPC; we linearize the system at the steady state, and design an LQ controller for the linearized system. The terminal region is then a sublevel set of the terminal penalty chosen small enough to satisfy the input constraints. We use the following parameter values in the simulation study

$$Q = I \quad R = I \quad N = 2 \quad \overline{p} = 3 \quad \mathbb{U}_i = [-2.5, 2.5] \quad \forall i \in \mathbb{I}_{1:2}$$

Figure 6.9 shows that the controller is stabilizing for as few as $\overline{p} = 3$ iterations. Increasing the maximum number of iterations can significantly improve the performance. Figure 6.9 shows the performance improvement for $\overline{p} = 10$, which is close to the centralized MPC performance. To see the difficulty in optimizing the nonconvex objective function, iterations of the initial control optimization are shown in Figure 6.10 for the $N = 1$ case. Clearly the distributed optimization method is able to efficiently handle this nonconvex objective with only a few iterations. □

6.6 Notes

At least three different fields have contributed substantially to the material presented in this chapter. We attempt here to point out briefly what each field has contributed, and indicate what literature the interested reader may wish to consult for further pursuing this and related subjects.

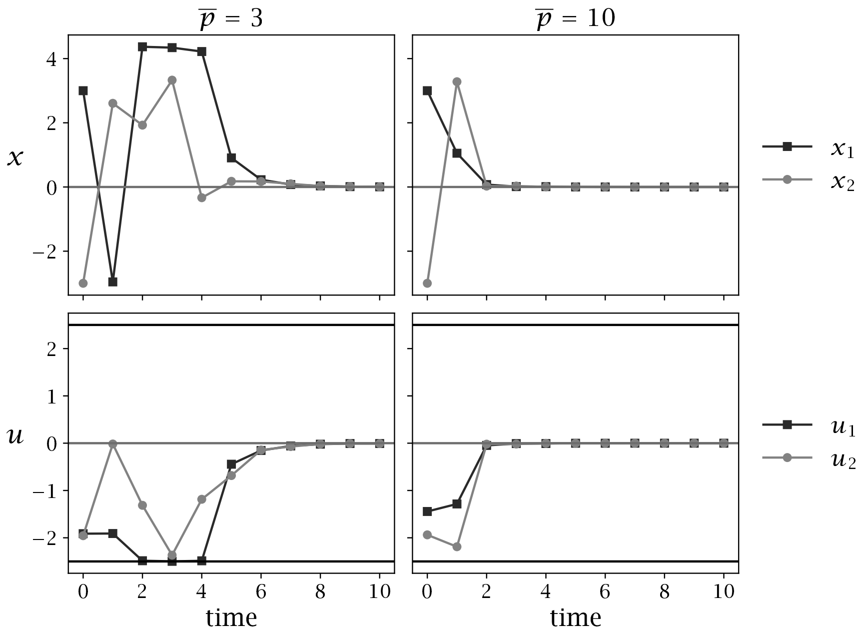

Figure 6.9: Closed-loop state and control evolution with $(x_1(0), x_2(0)) = (3, -3)$. Setting $\overline{p} = 10$ approximates the centralized controller.

Game theory. Game theory emerged in the mid-1900s to analyze situations in which multiple players follow a common set of rules but have their own and different objectives that they try to optimize in competition with each other. Von Neumann and Morgenstern introduced the classic text on this subject, “Theory of Games and Economic Behavior,” in 1944. A principle aim of game theory since its inception was to model and understand human *economic* behavior, especially as it arises in a capitalistic, free-market system. For that reason, much of the subsequent game theory literature was published in economics journals rather than systems theory journals. This field has contributed richly to the ideas and vocabulary used in this chapter to describe distributed control. For example, the game in which players have different objectives is termed *noncooperative*. The equilibrium of a noncooperative game is known as a *Nash equilibrium* (Nash, 1951). The Nash equilibrium is usually not Pareto optimal, which means that the outcomes for

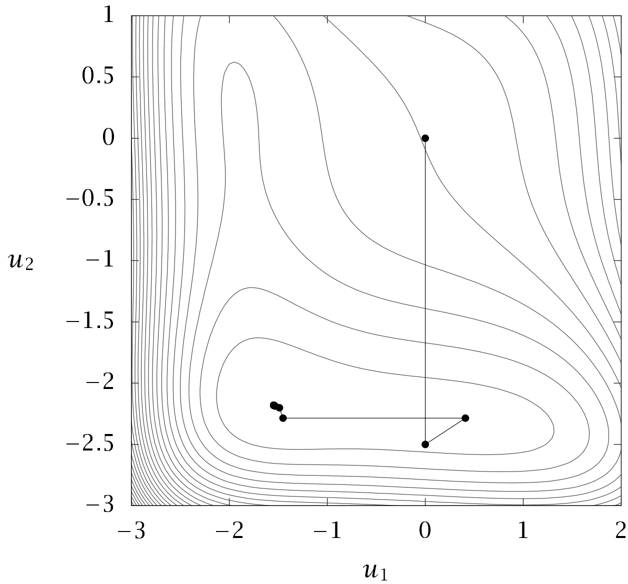

Figure 6.10: Contours of $V(x(0), \mathbf{u}_1, \mathbf{u}_2)$ with $N = 1$ at $k = 0$, $(x_1(0), x_2(0)) = (3, -3)$. Iterations of the subsystem controllers with initial condition $(u_1^0, u_2^0) = (0, 0)$.

all players can be improved simultaneously from the Nash solution. A comprehensive overview of the game theory literature, especially the parts relevant to control theory, is provided by Başar and Olsder (1999, Chapter 1), which is a highly recommended reference. Analyzing the equilibria of a noncooperative game is usually more complex than the cooperative game (optimal control problem). The closed-loop properties of a receding horizon implementation of any of these game theory solutions is not addressed in game theory. That topic is addressed by control theory.

Distributed optimization. The optimization community has extensively studied the issue of solving large-scale optimization problems using distributed optimization methods. The primary motivation in this field is to exploit parallel computing hardware and distributed data communication networks to solve large optimization problems faster. Bertsekas and Tsitsiklis provide an excellent and comprehensive overview of this field, focusing on numerical algorithms for implementing the distributed approaches. The important questions that are addressed in designing a distributed optimization are: task allocation, communication, and synchronization (Bertsekas and Tsitsiklis, 1997, Chapter 1).

These basic concepts arise in distributed problems of all types, and therefore also in the distributed MPC problem, which provides good synergy between these fields. But one should also note the structural distinctions between distributed optimization and distributed MPC. The primary obstacle to implementing centralized MPC for large-scale plants is not *computational* but *organizational*. The agents considered in distributed MPC are usually existing MPC systems already built for units or subsystems within an existing large-scale process. The plant management often is seeking to improve the plant performance by better coordinating the behavior of the different agents already in operation. Ignoring these structural constraints and treating the distributed MPC problem purely as a form of distributed optimization, ignores aspects of the design that are critical for successful industrial application (Rawlings and Stewart, 2008).

Control theory. Researchers have long studied the issue of how to distribute control tasks in a complex large-scale plant (Mesarović, Macko, and Takahara, 1970; Sandell Jr., Varaiya, Athans, and Safonov, 1978). The centralized controller and decentralized controller define two limiting design extremes. Centralized control accounts for all possible interactions, large and small, whereas decentralized control ignores them completely. In decentralized control the local agents have no knowledge of each others' actions. It is well known that the nominal closed-loop system behavior under decentralized control can be arbitrarily poor (unstable) if the system interactions are not small. The following reviews provide general discussion of this and other performance issues involving decentralized control (Šiljak, 1991; Lunze, 1992; Larsson and Skogestad, 2000; Cui and Jacobsen, 2002).

The next level up in design complexity from decentralized control is noncooperative control. In this framework, the agents have interaction models and communicate at each iteration (Jia and Krogh, 2002; Motee and Sayyar-Rodsari, 2003; Dunbar and Murray, 2006). The advantage of noncooperative control over decentralized control is that the agents have accurate knowledge of the effects of all other agents on their local objectives. The basic issue to analyze and understand in this setup is the competition between the agents. Characterizing the noncooperative equilibrium is the subject of noncooperative game theory, and the impact of using that solution for feedback control is the subject of control theory. For example, Dunbar (2007) shows closed-loop stability for an extension of noncooperative MPC described in (Dunbar and Murray, 2006) that handles systems with interacting subsystem dynamics. The key assumptions are the existence of a stabilizing *decentralized* feedback law valid near the origin, and an inequality condition limiting the coupling between the agents.

Cooperative MPC was introduced by Venkat, Rawlings, and Wright (2007). They show that a receding horizon implementation of a cooperative game with any number of iterates of the local MPC controllers leads to closed-loop stability for linear dynamics. Venkat, Rawlings, and Wright (2006a,b) show that state estimation errors (output instead of state feedback) do not change the system closed-loop stability if the estimators are also asymptotically stable. Most of the theoretical results on cooperative MPC of linear systems given in this chapter are presented in Venkat (2006) using an earlier, different notation. If implementable, this form of distributed MPC clearly has the best control properties. Although one can easily modify the agents' objective functions in a single large-scale process owned by a single company, this kind of modification may not be possible in other situations in which competing interests share critical infrastructure.

The requirements of the many different classes of applications continue to create exciting opportunities for continued research in this field. An excellent recent review provides a useful taxonomy of the different features of the different approaches (Scattolini, 2009). A recent text compiles no less than *35* different approaches to distributed MPC from more than *80* contributors (Maestre and Negenborn, 2014). The growth in the number and diversity of applications of distributed MPC shows no sign of abating.

6.7 Exercises

Exercise 6.1: Three looks at solving the LQ problem (LQP)

In the following exercise, you will write three codes to solve the LQR using Octave or MATLAB. The objective function is the LQR with mixed term

$$V = \frac{1}{2}\sum_{k=0}^{N-1}\left(x(k)'Qx(k) + u(k)'Ru(k) + 2x(k)'Mu(k)\right) + (1/2)x(N)'P_f x(N)$$

First, implement the method described in Section 6.1.1 in which you eliminate the state and solve the problem for the decision variable

$$\mathbf{u} = (u(0), u(1), \ldots, u(N-1))$$

Second, implement the method described in Section 6.1.1 in which you do *not* eliminate the state and solve the problem for

$$\mathbf{z} = (u(0), x(1), u(1), x(2), \ldots, u(N-1), x(N))$$

Third, use backward dynamic programming (DP) and the Riccati iteration to compute the closed-form solution for $u(k)$ and $x(k)$.

(a) Let

$$A = \begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} -2/3 & 1 \end{bmatrix} \qquad x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$Q = C'C + 0.001I \qquad P_f = \Pi \qquad R = 0.001 \qquad M = 0$$

in which the terminal penalty, P_f is set equal to Π, the steady-state cost to go. Compare the three solutions for $N = 5$. Plot $x(k)$, $u(k)$ versus time for the closed-loop system.

(b) Let $N = 50$ and repeat. Do any of the methods experience numerical problems generating an accurate solution? Plot the condition number of the matrix that is inverted in the first two methods versus N.

(c) Now consider the following unstable system

$$A = \begin{bmatrix} 27.8 & -82.6 & 34.6 \\ 25.6 & -76.8 & 32.4 \\ 40.6 & -122.0 & 51.9 \end{bmatrix} \qquad B = \begin{bmatrix} 0.527 & 0.548 \\ 0.613 & 0.530 \\ 1.06 & 0.828 \end{bmatrix} \qquad x(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Consider regulator tuning parameters and constraints

$$Q = I \qquad P_f = \Pi \qquad R = I \qquad M = 0$$

Repeat parts (a) and (b) for this system. Do you lose accuracy in any of the solution methods? What happens to the condition number of $H(N)$ and $S(N)$ as N becomes large? Which methods are still accurate for this case? Can you explain what happened?

Exercise 6.2: LQ as least squares

Consider the standard LQP

$$\min_{\mathbf{u}} V = \frac{1}{2}\sum_{k=0}^{N-1}\big(x(k)'Qx(k) + u(k)'Ru(k)\big) + (1/2)x(N)'P_f x(N)$$

subject to

$$x^+ = Ax + Bu$$

(a) Set up the dense Hessian least squares problem for the LQP with a horizon of three, $N = 3$. Eliminate the state equations and write out the objective function in terms of only the decision variables $u(0), u(1), u(2)$.

(b) What are the conditions for an optimum, i.e., what linear algebra problem do you solve to compute $u(0), u(1), u(2)$?

Exercise 6.3: Lagrange multiplier method

Consider the general least squares problem

$$\min_x V(x) = \frac{1}{2}x'Hx + \text{const}$$

subject to

$$Dx = d$$

(a) What is the Lagrangian L for this problem? What is the dimension of the Lagrange multiplier vector, λ?

(b) What are necessary and sufficient conditions for a solution to the optimization problem?

(c) Apply this approach to the LQP of Exercise 6.2 using the equality constraints to represent the model equations. What are H, D, d for the LQP?

(d) Write out the linear algebra problem to be solved for the optimum.

(e) Contrast the two different linear algebra problems in these two approaches. Which do you want to use when N is large and why?

Exercise 6.4: Reparameterizing an unstable system

Consider again the LQR problem with cross term

$$\min_{\mathbf{u}} V = \frac{1}{2}\sum_{k=0}^{N-1}\big(x(k)'Qx(k) + u(k)'Ru(k) + 2x(k)'Mu(k)\big) + (1/2)x(N)'P_f x(N)$$

subject to

$$x^+ = Ax + Bu$$

and the three approaches of Exercise 6.1.

1. The method described in Section 6.1.1 in which you eliminate the state and solve the problem for the decision variable

$$\mathbf{u} = (u(0), u(1), \ldots, u(N-1))$$

2. The method described in Section 6.1.1 in which you do *not* eliminate the state and solve the problem for

$$\mathbf{z} = (u(0), x(1), u(1), x(2), \ldots, u(N-1), x(N))$$

3. The method of DP and the Riccati iteration to compute the closed-form solution for $u(k)$ and $x(k)$.

(a) You found that unstable A causes numerical problems in the first method using large horizons. So let's consider a fourth method. Reparameterize the input in terms of a state feedback gain via

$$u(k) = Kx(k) + v(k)$$

in which K is chosen so that $A + BK$ is a stable matrix. Consider the matrices in a transformed LQP

$$\min_{\mathbf{v}} V = \frac{1}{2}\sum_{k=0}^{N-1}\left(x(k)'\widetilde{Q}x(k) + v(k)'\widetilde{R}v(k) + 2x(k)'\widetilde{M}v(k)\right) + (1/2)x(N)'\widetilde{P}_f x(N)$$

subject to $x^+ = \widetilde{A}x + \widetilde{B}v$.

What are the matrices $\widetilde{A}, \widetilde{B}, \widetilde{Q}, \widetilde{P}_f, \widetilde{R}, \widetilde{M}$ such that the two problems give the same solution (state trajectory)?

(b) Solve the following problem using the first method and the fourth method and describe differences between the two solutions. Compare your results to the DP approach. Plot $x(k)$ and $u(k)$ versus k.

$$A = \begin{bmatrix} 27.8 & -82.6 & 34.6 \\ 25.6 & -76.8 & 32.4 \\ 40.6 & -122.0 & 51.9 \end{bmatrix} \qquad B = \begin{bmatrix} 0.527 & 0.548 \\ 0.613 & 0.530 \\ 1.06 & 0.828 \end{bmatrix} \qquad x(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Consider regulator tuning parameters and constraints

$$Q = P_f = I \qquad R = I \qquad M = 0 \qquad N = 50$$

Exercise 6.5: Recursively summing quadratic functions

Consider generalizing Example 1.1 to an N-term sum. Let the N-term sum of quadratic functions be defined as

$$V(N, x) = \frac{1}{2}\sum_{i=1}^{N}(x - x(i))'X_i(x - x(i))$$

in which $x, x(i) \in \mathbb{R}^n$ are real n-vectors and $X_i \in \mathbb{R}^{n\times n}$ are positive definite matrices.

(a) Show that $V(N, x)$ can be found recursively

$$V(N, x) = (1/2)(x - v(N))'H(N)(x - v(N)) + \text{constant}$$

in which $v(i)$ and $H(i)$ satisfy the recursion

$$\begin{aligned} H(i+1) &= H_i + X_{i+1} & v(i+1) &= H^{-1}(i+1)\left(H_i v_i + X_{i+1}x(i+1)\right) \\ H_1 &= X_1 & v_1 &= x_1 \end{aligned}$$

Notice the recursively defined $v(m)$ and $H(m)$ provide the solutions and the Hessian matrices of the sequence of optimization problems

$$\min_x V(m, x) \qquad 1 \le m \le N$$

(b) Check your answer by solving the equivalent, but larger dimensional, constrained least squares problem (see Exercise 1.16)

$$\min_z (z - z_0)' \widetilde{H} (z - z_0)$$

subject to

$$Dz = 0$$

in which $z, z_0 \in \mathbb{R}^{nN}$, $\widetilde{H} \in \mathbb{R}^{nN \times nN}$ is a block diagonal matrix, $D \in \mathbb{R}^{n(N-1) \times nN}$

$$z_0 = \begin{bmatrix} x(1) \\ \vdots \\ x(N-1) \\ x(N) \end{bmatrix} \qquad \widetilde{H} = \begin{bmatrix} X_1 & & & \\ & \ddots & & \\ & & X_{N-1} & \\ & & & X_N \end{bmatrix} \qquad D = \begin{bmatrix} I & -I & & \\ & \ddots & \ddots & \\ & & I & -I \end{bmatrix}$$

(c) Compare the size and number of matrix inverses required for the two approaches.

Exercise 6.6: Why call the Lyapunov stability *nonuniform*?

Consider the following linear system

$$\begin{aligned} w^+ &= Aw \qquad w(0) = Hx(0) \\ x &= Cw \end{aligned}$$

with solution $w(k) = A^k w(0) = A^k Hx(0)$, $x(k) = CA^k Hx(0)$. Notice that $x(0)$ completely determines both $w(k)$ and $x(k)$, $k \geq 0$. Also note that zero is a solution, i.e., $x(k) = 0, k \geq 0$ satisfies the model.

(a) Consider the following case

$$A = \rho \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad H = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$\rho = 0.925 \qquad \theta = \pi/4 \qquad x(0) = 1$$

Plot the solution $x(k)$. Does $x(k)$ converge to zero? Does $x(k)$ achieve zero exactly for finite $k > 0$?

(b) Is the zero solution $x(k) = 0$ Lyapunov stable? State your definition of Lyapunov stability, and prove your answer. Discuss how your answer is consistent with the special case considered above.

Exercise 6.7: Exponential stability of suboptimal MPC with unbounded feasible set

Consider again Lemma 6.5 when both $\mathbb{U}$ and $\mathcal{X}_N$ are unbounded. Show that the suboptimal MPC controller is exponentially stable on the following sets.

(a) Any sublevel set of $V(x, \mathbf{h}(x))$

(b) Any compact subset of $\mathcal{X}_N$

Exercise 6.8: A refinement to the warm start

Consider the following refinement to the warm start in the suboptimal MPC strategy. First add the requirement that the initialization strategy satisfies the following bound

$$\mathbf{h}(x) \leq \bar{d}\,|x| \qquad x \in \mathcal{X}_N$$

in which $\bar{d} > 0$. Notice that all initializations considered in the chapter satisfy this requirement.

Then, at time k and state x, in addition to the shifted input sequence from time $k-1$, $\tilde{\mathbf{u}}$, evaluate the initialization sequence applied to the current state, $\mathbf{u} = \mathbf{h}(x)$. Select whichever of these two input sequence has lower cost as the warm start for time k. Notice also that this refinement makes the constraint

$$|\mathbf{u}| \leq d\,|x| \qquad x \in r\mathcal{B}$$

redundant, and it can be removed from the MPC optimization.

Prove that this refined suboptimal strategy is exponentially stabilizing on the set $\mathcal{X}_N$. Notice that with this refinement, we do not have to assume that $\mathcal{X}_N$ is bounded or that $\mathbb{U}$ is bounded.

Exercise 6.9: Global asymptotic stability and exponential convergence with mixed powers of the norm

Prove Lemma 6.6.

Hints: exponential convergence can be established as in standard exponential stability theorems. To establish Lyapunov stability, notice that $|x(0)| \leq |(x(0), e(0))|$ and $|e(0)| \leq |(x(0), e(0))|$ and that $(\cdot)^\alpha$ for $\alpha > 0$ is a $\mathcal{K}_\infty$ function.

Exercise 6.10: Decentralized control of Examples 6.9–6.11

Apply decentralized control to the systems in Examples 6.9–6.11. Which of these systems are closed-loop unstable with decentralized control? Compare this result to the result for noncooperative MPC.

Exercise 6.11: Cooperative control of Examples 6.9–6.11

Apply cooperative MPC to the systems in Examples 6.9–6.11. Are any of these systems closed-loop unstable? Compare the closed-loop eigenvalues of converged cooperative control to centralized MPC, and discuss any differences.

Exercise 6.12: Adding norms

Establish the following result used in the proof of Lemma 6.14. Given that $w \in \mathbb{R}^m$, $e \in \mathbb{R}^n$

$$\frac{1}{\sqrt{2}}(|w| + |e|) \leq |(w, e)| \leq |w| + |e| \qquad \forall w, e$$

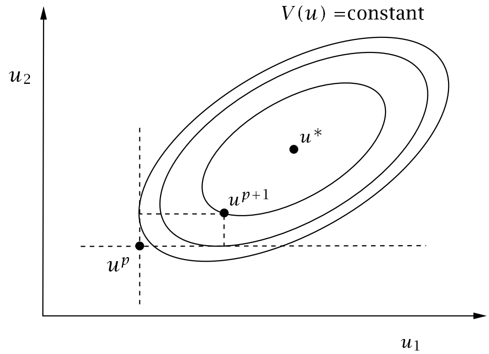

Figure 6.11: Optimizing a quadratic function in one set of variables at a time.

Exercise 6.13: Padding matrices

Given a vector $\mathbf{z}$ and subvector $\mathbf{u}$

$$\mathbf{z} = \begin{bmatrix} u(0) \\ x(1) \\ u(1) \\ x(2) \\ \vdots \\ u(N-1) \\ x(N) \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} \qquad x \in \mathbb{R}^n \qquad u \in \mathbb{R}^m$$

and quadratic function of $\mathbf{u}$

$$(1/2)\mathbf{u}'H\mathbf{u} + h'\mathbf{u}$$

Find the corresponding quadratic function of $\mathbf{z}$ so that

$$(1/2)\mathbf{z}'H_z\mathbf{z} + h_z'\mathbf{z} = (1/2)\mathbf{u}'H\mathbf{u} + h'\mathbf{u} \qquad \forall \mathbf{z}, \mathbf{u}$$

Hint: first find the padding matrix E such that $\mathbf{u} = E\mathbf{z}$.

Exercise 6.14: A matrix inverse

Compute the four partitioned elements in the two-player feedback gain $(I-L)^{-1}\overline{K}$

$$\mathbf{u}^\infty = (I-L)^{-1}\overline{K}x(0) \qquad \left|\text{eig}(L)\right| < 1$$

in which

$$(I-L)^{-1}\overline{K} = \begin{bmatrix} I & -L_1 \\ -L_2 & I \end{bmatrix}^{-1} \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}$$

Exercise 6.15: Optimizing one variable at a time

Consider the positive definite quadratic function partitioned into two sets of variables

$$V(u) = (1/2)u'Hu + c'u + d$$
$$V(u_1, u_2) = (1/2)\begin{bmatrix} u_1' & u_2' \end{bmatrix}\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} c_1' & c_2' \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + d$$

in which $H > 0$. Imagine we wish to optimize this function by first optimizing over the u_1 variables holding u_2 fixed and then optimizing over the u_2 variables holding u_1 fixed as shown in Figure 6.11. Let's see if this procedure, while not necessarily efficient, is guaranteed to converge to the optimum.

(a) Given an initial point (u_1^p, u_2^p), show that the next iteration is

$$\begin{aligned} u_1^{p+1} &= -H_{11}^{-1}\left(H_{12}u_2^p + c_1\right) \\ u_2^{p+1} &= -H_{22}^{-1}\left(H_{21}u_1^p + c_2\right) \end{aligned} \tag{6.39}$$

The procedure can be summarized as

$$u^{p+1} = Au^p + b \tag{6.40}$$

in which the iteration matrix A and constant b are given by

$$A = \begin{bmatrix} 0 & -H_{11}^{-1}H_{12} \\ -H_{22}^{-1}H_{21} & 0 \end{bmatrix} \qquad b = \begin{bmatrix} -H_{11}^{-1}c_1 \\ -H_{22}^{-1}c_2 \end{bmatrix} \tag{6.41}$$

(b) Establish that the optimization procedure converges by showing the iteration matrix is stable

$$\left|\operatorname{eig}(A)\right| < 1$$

(c) Given that the iteration converges, show that it produces the same solution as

$$u^* = -H^{-1}c$$

Exercise 6.16: Monotonically decreasing cost

Consider again the iteration defined in Exercise 6.15.

(a) Prove that the cost function is monotonically decreasing when optimizing one variable at a time

$$V(u^{p+1}) < V(u^p) \qquad \forall u^p \neq -H^{-1}c$$

(b) Show that the following expression gives the size of the decrease

$$V(u^{p+1}) - V(u^p) = -(1/2)(u^p - u^*)'P(u^p - u^*)$$

in which

$$P = HD^{-1}\widetilde{H}D^{-1}H \qquad \widetilde{H} = D - N \qquad D = \begin{bmatrix} H_{11} & 0 \\ 0 & H_{22} \end{bmatrix} \qquad N = \begin{bmatrix} 0 & H_{12} \\ H_{21} & 0 \end{bmatrix}$$

and $u^* = -H^{-1}c$ is the optimum.

Hint: to simplify the algebra, first change coordinates and move the origin of the coordinate system to u^*.

Exercise 6.17: One variable at a time with convex step

Consider Exercise 6.15 but with the convex step for the iteration

$$\begin{bmatrix} u_1^{p+1} \\ u_2^{p+1} \end{bmatrix} = w_1 \begin{bmatrix} u_1^0(u_2^p) \\ u_2^p \end{bmatrix} + w_2 \begin{bmatrix} u_1^p \\ u_2^0(u_1^p) \end{bmatrix} \qquad 0 \leq w_1, w_2 \quad w_1 + w_2 = 1$$

(a) Show that the iteration for the convex step is also of the form

$$u^{p+1} = Au^p + b$$

and the A matrix and b vector for this case are

$$A = \begin{bmatrix} w_2 I & -w_1 H_{11}^{-1} H_{12} \\ -w_2 H_{22}^{-1} H_{21} & w_1 I \end{bmatrix} \qquad b = \begin{bmatrix} -w_1 H_{11}^{-1} & \\ & -w_2 H_{22}^{-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

(b) Show that A is stable.

(c) Show that this iteration also converges to $u^* = -H^{-1}c$.

Exercise 6.18: Monotonically decreasing cost with convex step

Consider again the problem of optimizing one variable at a time with the convex step given in Exercise 6.17.

(a) Prove that the cost function is monotonically decreasing

$$V(u^{p+1}) < V(u^p) \qquad \forall u^p \neq -H^{-1}c$$

(b) Show that the following expression gives the size of the decrease

$$V(u^{p+1}) - V(u^p) = -(1/2)(u^p - u^*)'P(u^p - u^*)$$

in which

$$P = HD^{-1}\widetilde{H}D^{-1}H \qquad \widetilde{H} = D - N$$

$$D = \begin{bmatrix} w_1^{-1} H_{11} & 0 \\ 0 & w_2^{-1} H_{22} \end{bmatrix} \qquad N = \begin{bmatrix} -w_1^{-1} w_2 H_{11} & H_{12} \\ H_{21} & -w_1 w_2^{-1} H_{22} \end{bmatrix}$$

and $u^* = -H^{-1}c$ is the optimum.

Hint: to simplify the algebra, first change coordinates and move the origin of the coordinate system to u^*.

Exercise 6.19: Splitting more than once

Consider the generalization of Exercise 6.15 in which we repeatedly decompose a problem into one-variable-at-a-time optimizations. For a three-variable problem we have the three optimizations

$$u_1^{p+1} = \arg\min_{u_1} V(u_1, u_2^p, u_3^p)$$

$$u_2^{p+1} = \arg\min_{u_2} V(u_1^p, u_2, u_3^p) \qquad u_3^{p+1} = \arg\min_{u_3} V(u_1^p, u_2^p, u_3)$$

Is it true that

$$V(u_1^{p+1}, u_2^{p+1}, u_3^{p+1}) \leq V(u_1^p, u_2^p, u_3^p)$$

Hint: you may wish to consider the following example, $V(u) = (1/2)u'Hu + c'u$, in which

$$H = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \qquad c = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \qquad u^p = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Exercise 6.20: Time-varying controller iterations

We let $p_k \geq 0$ be a time-varying integer-valued index representing the iterations applied in the controller at time k.

$$\begin{aligned}
x_1(k+1) &= A_1 x_1(k) + \overline{B}_{11} u_1(0;k) + \overline{B}_{12} u_2(0;k) \\
x_2(k+1) &= A_2 x_2(k) + \overline{B}_{21} u_1(0;k) + \overline{B}_{22} u_2(0;k) \\
\mathbf{u}_1(k+1) &= g_1^{p_k}(x_1(k), x_2(k), \mathbf{u}_1(k), \mathbf{u}_2(k)) \\
\mathbf{u}_2(k+1) &= g_2^{p_k}(x_1(k), x_2(k), \mathbf{u}_1(k), \mathbf{u}_2(k))
\end{aligned}$$

Notice the system evolution is time-varying even though the models are time invariant because we allow a time-varying sequence of controller iterations.

Show that cooperative MPC is exponentially stabilizing for any $p_k \geq 0$ sequence.

Exercise 6.21: Stable interaction models

In some industrial applications it is preferable to partition the plant so that there are no unstable connections between subsystems. Any inputs u_j that have unstable connections to outputs y_i should be included in the ith subsystem inputs. Allowing an unstable connection between two subsystems may not be robust to faults and other kinds of system failures.[5] To implement this design idea in the two-player case, we replace Assumption 6.13 (b) with the following

Modified Assumption 6.13 (Constrained two-player game)**.**

(b) The interaction models $A_{ij}, i \neq j$ are stable.

Prove that Modified Assumption 6.13 (b) implies Assumption 6.13 (b). It may be helpful to first prove the following lemma.

Lemma 6.23 (Local detectability)**.** *Given partitioned system matrices*

$$\boldsymbol{A} = \begin{bmatrix} A & 0 \\ 0 & A_s \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} C & C_s \end{bmatrix}$$

in which A_s is stable, the system $(\boldsymbol{A}, \boldsymbol{C})$ is detectable if and only if the system (A, C) is detectable.

Hint: use the Hautus lemma as the test for detectability.

Next show that this lemma and Modified Assumption 6.13 (b) establishes the distributed detectability assumption, Assumption 6.13 (b).

Exercise 6.22: Norm constraints as linear inequalities

Consider the quadratic program (QP) in decision variable u with parameter x

$$\begin{aligned}
&\min_u (1/2) u' H u + x' D u \\
&\text{s.t. } Eu \leq Fx
\end{aligned}$$

[5] We are not considering the common instability of base-level inventory management in this discussion. It is assumed that level control in storage tanks (integrators) is maintained at all times with simple, local level controllers. The internal unit flowrates dedicated for inventory management are not considered available inputs in the MPC problem.

in which $u \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, and $H > 0$. The parameter x appears linearly (affinely) in the cost function and constraints. Assume that we wish to add a norm constraint of the following form

$$|u|_\alpha \leq c\,|x|_\alpha \qquad \alpha = 2, \infty$$

(a) If we use the infinity norm, show that this problem can be posed as an equivalent QP with additional decision variables, and the cost function and constraints remain linear (affine) in parameter x. How many decision variables and constraints are added to the problem?

(b) If we use the two norm, show that this problem can be approximated by a QP whose solution does satisfy the constraints, but the solution may be suboptimal compared to the original problem.

Exercise 6.23: Steady-state noncooperative game

Consider again the steady-state target problem for the system given in Example 6.12.

(a) Resolve the problem for the choice of convex step parameters $w_1 = 0.2$, $w_2 = 0.8$. Does the iteration for noncooperative control converge? Plot the iterations for the noncooperative and cooperative cases.

(b) Repeat for the convex step $w_1 = 0.8, w_2 = 0.2$. Are the results identical to the previous part? If not, discuss any differences.

(c) For what choices of w_1, w_2 does the target iteration converge using noncooperative control for the target calculation?

Exercise 6.24: Optimality conditions for constrained optimization

Consider the convex quadratic optimization problem

$$\min_u V(u) \qquad \text{subject to} \quad u \in \mathbb{U}$$

in which V is a convex quadratic function and $\mathbb{U}$ is a convex set. Show that u^* is an optimal solution if and only if

$$\langle z - u^*, -\nabla V|_{u^*} \rangle \quad \leq 0 \qquad \forall z \in \mathbb{U} \tag{6.42}$$

Figure 6.12(a) depicts this condition for $u \in \mathbb{R}^2$. This condition motivates defining the normal cone (Rockafellar, 1970) to $\mathbb{U}$ at u^* as follows

$$N(\mathbb{U}, u^*) = \{y \mid \langle z - u^*, y - u^* \rangle \ \leq 0 \quad \forall z \in \mathbb{U}\}$$

The optimality condition can be stated equivalently as u^* is an optimal point if and only if the negative gradient is in the normal cone to $\mathbb{U}$ at u^*

$$-\nabla V|_{u^*} \in N(\mathbb{U}, u^*)$$

This condition and the normal cone are depicted in Figure 6.12(b).

Exercise 6.25: Partitioned optimality conditions with constraints

Consider a partitioned version of the constrained optimization problem of Exercise 6.24 with uncoupled constraints

$$\min_{u_1, u_2} V(u_1, u_2) \qquad \text{subject to} \quad u_1 \in \mathbb{U}_1 \qquad u_2 \in \mathbb{U}_2$$

in which V is a quadratic function and $\mathbb{U}_1$ and $\mathbb{U}_2$ are convex and nonempty.

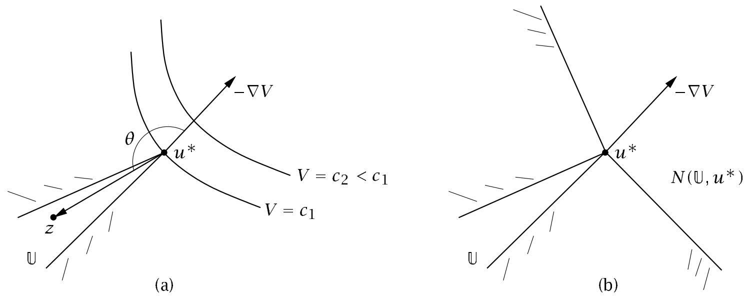

Figure 6.12: (a) Optimality of u^* means the angle between $-\nabla V$ and any point z in the feasible region must be greater than $90°$ and less than $270°$. (b) The same result restated: u^* is optimal if and only if the negative gradient is in the normal cone to the feasible region at u^*, $-\nabla V|_{u^*} \in N(\mathbb{U}, u^*)$.

(a) Show that (u_1^*, u_2^*) is an optimal solution if and only if

$$\begin{aligned} \langle z_1 - u_1^*, -\nabla_{u_1} V|_{(u_1^*, u_2^*)} \rangle &\leq 0 \qquad \forall z_1 \in \mathbb{U}_1 \\ \langle z_2 - u_2^*, -\nabla_{u_2} V|_{(u_1^*, u_2^*)} \rangle &\leq 0 \qquad \forall z_2 \in \mathbb{U}_2 \end{aligned} \tag{6.43}$$

(b) Extend the optimality conditions to cover the case

$$\min_{u_1,\ldots,u_M} V(u_1, \ldots, u_M) \qquad \text{subject to} \quad u_j \in \mathbb{U}_j \quad j = 1, \ldots, M$$

in which V is a quadratic function and the $\mathbb{U}_j$ are convex and nonempty.

Exercise 6.26: Constrained optimization of M variables

Consider an optimization problem with M variables and uncoupled constraints

$$\min_{u_1,u_2,\ldots,u_M} V(u_1, u_2, \ldots, u_M) \qquad \text{subject to} \quad u_l \in \mathbb{U}_j \quad j = 1, \ldots, M$$

in which V is a strictly convex function. Assume that the feasible region is convex and nonempty and denote the unique optimal solution as $(u_1^*, u_2^*, \ldots, u_M^*)$ having cost $V^* = V(u_1^*, \ldots, u_M^*)$. Denote the M one-variable-at-a-time optimization problems at iteration k

$$z_j^{p+1} = \arg\min_{u_j} V(u_1^p, \ldots, u_j, \ldots, u_M^p) \qquad \text{subject to } u_j \in \mathbb{U}_j$$

Then define the next iterate to be the following convex combination of the previous and new points

$$\begin{gathered} u_j^{p+1} = \alpha_j^p z_j^{p+1} + (1 - \alpha_j^p) u_j^p \qquad j = 1, \ldots, M \\ \varepsilon \leq \alpha_j^p < 1 \qquad 0 < \varepsilon \qquad j = 1, \ldots, M, \quad p \geq 1 \\ \sum_{j=1}^{M} \alpha_j^p = 1, \qquad p \geq 1 \end{gathered}$$

Prove the following results.

(a) Starting with any feasible point, $(u_1^0, u_2^0, \ldots, u_M^0)$, the iterations $(u_1^p, u_2^p, \ldots, u_M^p)$ are feasible for $p \geq 1$.

(b) The objective function decreases monotonically from any feasible initial point

$$V(u_1^{p+1}, \ldots, u_M^{p+1}) \leq V(u_1^p, \ldots, u_M^p) \qquad \forall u_l^0 \in \mathbb{U}_j, j = 1, \ldots, M, \quad p \geq 1$$

(c) The cost sequence $V(u_1^p, u_2^p, \ldots, u_M^p)$ converges to the optimal cost V^* from any feasible initial point.

(d) The sequence $(u_1^p, u_2^p, \ldots, u_M^p)$ converges to the optimal solution $(u_1^*, u_2^*, \ldots, u_M^*)$ from any feasible initial point.

Exercise 6.27: The constrained two-variable special case

Consider the special case of Exercise 6.26 with $M = 2$

$$\min_{u_1, u_2} V(u_1, u_2) \qquad \text{subject to} \quad u_1 \in \mathbb{U}_1 \qquad u_2 \in \mathbb{U}_2$$

in which V is a strictly positive quadratic function. Assume that the feasible region is convex and nonempty and denote the unique optimal solution as (u_1^*, u_2^*) having cost $V^* = V(u_1^*, u_2^*)$. Consider the two one-variable-at-a-time optimization problems at iteration k

$$u_1^{p+1} = \arg\min_{u_1} V(u_1, u_2^p) \qquad\qquad u_2^{p+1} = \arg\min_{u_2} V(u_1^p, u_2)$$

$$\text{subject to } u_1 \in \mathbb{U}_1 \qquad\qquad \text{subject to } u_2 \in \mathbb{U}_2$$

We know from Exercise 6.15 that taking the full step in the unconstrained problem with $M = 2$ achieves a cost decrease. We know from Exercise 6.19 that taking the full step for an unconstrained problem with $M \geq 3$ does *not* provide a cost decrease in general. We know from Exercise 6.26 that taking a reduced step in the constrained problem for all M achieves a cost decrease. That leaves open the case of a full step for a constrained problem with $M = 2$.

Does the full step in the constrained case for $M = 2$ guarantee a cost decrease? If so, prove it. If not, provide a counterexample.

Exercise 6.28: Subsystem stability constraints

Show that the following uncoupled subsystem constraints imply an overall system constraint of the same type. The first is suitable for asymptotic stability and the second for exponential stability.

(a) Given $r_1, r_2 > 0$, and functions γ_1 and γ_2 of class $\mathcal{K}$, assume the following constraints are satisfied

$$\begin{aligned} |\mathbf{u}_1| &\leq \gamma_1(|x_1|) \quad x_1 \in r_1\mathcal{B} \\ |\mathbf{u}_2| &\leq \gamma_2(|x_2|) \quad x_2 \in r_2\mathcal{B} \end{aligned}$$

Show that there exists $r > 0$ and function γ of class $\mathcal{K}$ such that

$$|(\mathbf{u}_1, \mathbf{u}_2)| \leq \gamma(|(x_1, x_2)|) \qquad (x_1, x_2) \in r\mathcal{B} \tag{6.44}$$

(b) Given $r_1, r_2 > 0$, and constants $c_1, c_2, \sigma_1, \sigma_2 > 0$, assume the following constraints are satisfied

$$\begin{aligned} |\mathbf{u}_1| &\leq c_1 \, |x_1|^{\sigma_1} \quad & x_1 \in r_1\mathcal{B} \\ |\mathbf{u}_2| &\leq c_2 \, |x_2|^{\sigma_2} \quad & x_2 \in r_2\mathcal{B} \end{aligned}$$

Show that there exists $r > 0$ and function $c, \sigma > 0$ such that

$$|(\mathbf{u}_1, \mathbf{u}_2)| \leq c \, |(x_1, x_2)|^{\sigma} \qquad (x_1, x_2) \in r\mathcal{B} \tag{6.45}$$

Exercise 6.29: Distributed disturbance detectability

Prove Lemma 6.16.

Hint: use the Hautus lemma as the test for detectability.

Exercise 6.30: Distributed target problem and uncoupled constraints

Player one's distributed target problem in the two-player game is given in (6.28)

$$\min_{x_{11s}, x_{21s}, u_{1s}} (1/2) \begin{bmatrix} H_1 y_{1s} - z_{1sp} \\ H_2 y_{2s} - z_{2sp} \end{bmatrix}' \begin{bmatrix} T_{1s} & \\ & T_{2s} \end{bmatrix} \begin{bmatrix} H_1 y_{1s} - z_{1sp} \\ H_2 y_{2s} - z_{2sp} \end{bmatrix}$$

subject to

$$\begin{bmatrix} I - A_1 & & -\overline{B}_{11} & -\overline{B}_{12} \\ & I - A_2 & -\overline{B}_{21} & -\overline{B}_{22} \end{bmatrix} \begin{bmatrix} x_{1s} \\ x_{2s} \\ u_{1s} \\ u_{2s} \end{bmatrix} = \begin{bmatrix} B_{1d} \hat{d}_1(k) \\ B_{2d} \hat{d}_2(k) \end{bmatrix}$$

$$E_1 u_{1s} \leq e_1$$

Show that the constraints can be expressed so that the target problem constraints are uncoupled.

Bibliography

T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* SIAM, Philadelphia, 1999.

D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, second edition, 1999.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation.* Athena Scientific, Belmont, Massachusetts, 1997.

A. E. Bryson and Y. Ho. *Applied Optimal Control.* Hemisphere Publishing, New York, 1975.

H. Cui and E. W. Jacobsen. Performance limitations in decentralized control. *J. Proc. Cont.*, 12:485–494, 2002.

W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Auto. Cont.*, 52(7):1249–1263, 2007.

W. B. Dunbar and R. M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.

D. Jia and B. H. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the American Control Conference*, pages 4507–4512, Anchorage,Alaska, May 2002.

T. Larsson and S. Skogestad. Plantwide control- A review and a new design procedure. *Mod. Ident. Control*, 21(4):209–240, 2000.

J. Lunze. *Feedback Control of Large Scale Systems.* Prentice-Hall, London, U.K., 1992.

J. M. Maestre and R. R. Negenborn. *Distributed Model Predictive Control Made Easy.* Springer Netherlands, 2014.

M. Mesarović, D. Macko, and Y. Takahara. *Theory of hierarchical, multilevel systems.* Academic Press, New York, 1970.

N. Motee and B. Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference*, pages 5300–5305, Denver,Colorado, June 2003.

J. Nash. Noncooperative games. *Ann. Math.*, 54:286–295, 1951.

J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *J. Proc. Cont.*, 18:839–845, 2008.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, N.J., 1970.

N. R. Sandell Jr., P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Trans. Auto. Cont.*, 23(2):108–128, 1978.

R. Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *J. Proc. Cont.*, 19(5):723–731, May 2009.

P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Auto. Cont.*, 44(3):648–654, March 1999.

D. D. Šiljak. *Decentralized Control of Complex Systems*. Academic Press, London, 1991.

B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Sys. Cont. Let.*, 59:460–469, 2010.

B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *J. Proc. Cont.*, 21(5):698–704, June 2011.

A. N. Venkat. *Distributed Model Predictive Control: Theory and Applications*. PhD thesis, University of Wisconsin–Madison, October 2006.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed, linear MPC. Part 1: state feedback. Technical Report 2006–03, TWMCC, Department of Chemical and Biological Engineering, University of Wisconsin–Madison (Available at http://jbrwww.che.wisc.edu/tech-reports.html), October 2006a.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed, linear MPC. Part 2: output feedback. Technical Report 2006–04, TWMCC, Department of Chemical and Biological Engineering, University of Wisconsin–Madison (Available at http://jbrwww.che.wisc.edu/tech-reports.html), October 2006b.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Distributed model predictive control of large-scale systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 591–605. Springer, 2007.

M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, second edition, 1993.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton and Oxford, 1944.

S. J. Wright. Applying new optimization algorithms to model predictive control. In J. C. Kantor, C. E. García, and B. Carnahan, editors, *Chemical Process Control–V*, pages 147–155. CACHE, AIChE, 1997.