3

Specialized Rollout Algorithms

Contents

In this chapter, we illustrate and extend the rollout algorithms of Chapter 2 in various specialized settings. We begin in Section 3.1 with model predictive control, a methodology of great importance in control system design. In Section 3.2, we consider multiagent problems, and we develop an effective approach to deal with control spaces that are large because the control consists of multiple components. In Section 3.3, we discuss how we can incorporate trajectory constraints into the rollout approach for deterministic optimal control. In Section 3.4, we provide illustrations of constrained rollout, which involve discrete and combinatorial optimization problems. In Section 3.5, we discuss rollout for Bayesian optimization, a partial state information DP formulation of a static optimization problem, whereby the outcomes of past experiments are used to design more effective future experiments. Finally, in Section 3.6, we show how to adapt the main DP principles and the rollout approach to the case of minimax control problems that involve a set membership description of uncertainty.

3.1 Model Predictive Control

We will consider a classical control problem, where the objective is to keep the state of a deterministic system close to the origin of the state space or close to a given trajectory. This problem has a long history, and has been addressed by a variety of methods. Starting in the late 50s and early 60s, approaches based on state variable system representations and optimal control became popular. The linear-quadratic approach whereby the system is represented by a linear model, the cost is quadratic in the state and the control, and there are no state and control constraints was developed during this period, and is still used extensively. Unfortunately, however, linear-quadratic models are often not satisfactory. There are two main reasons for this:

(1) The system may be nonlinear, and it may be inappropriate to use for control purposes a model that is linearized around the desired point or trajectory. Moreover, some of the control variables may be naturally discrete, and this is incompatible with the linear system viewpoint.

(2) There may be control and/or state constraints, which are not handled adequately through quadratic penalty terms in the cost function. For example, the motion of a car may be constrained by the presence of obstacles and hardware limitations (see Fig. 3.1.1). The solution obtained from a linear-quadratic model may not be suitable for such a problem, because quadratic penalties treat constraints "softly" and may produce trajectories that violate the constraints.

These inadequacies of the linear-quadratic formulation have motivated a methodology, called *model predictive control* (MPC for short), which combines elements of several ideas that we have discussed so far,

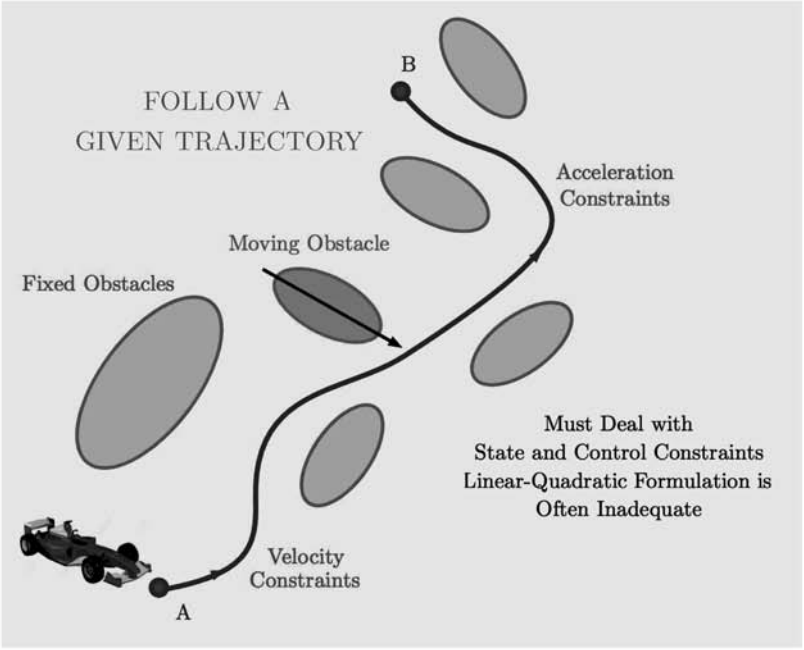

Figure 3.1.1 Illustration of constrained motion of a car from point A to point B. There are state (position/velocity) constraints, and control (acceleration) constraints. When there are mobile obstacles, the state constraints may change unpredictably, necessitating on-line replanning.

such as multistep lookahead, rollout with infinite control spaces, and certainty equivalence. Aside from resolving the difficulty with infinitely many Q-factors at x_k, while dealing adequately with state and control constraints, MPC is well-suited for on-line replanning, like all rollout methods.

The ideas of MPC were developed independently of the approximate DP/RL methodology and rollout in particular. However, the two fields are closely related, and there is much to be gained from understanding one field within the context of the other, as the subsequent development will aim to show.

We will focus primarily on the most common form of MPC, where the system is either deterministic, or else it is stochastic, but it is replaced with a deterministic version by using typical values in place of the uncertain quantities, or state estimates in place of exact state values, in the spirit of the certainty equivalent control approach. Moreover we will consider the case where the objective is to keep the state close to the origin (or more generally some point of interest, called the *set point*). This is called the *regulation problem*; see Fig. 3.1.2 for an example. Similar approaches have been developed for the problem of maintaining the state of a non-

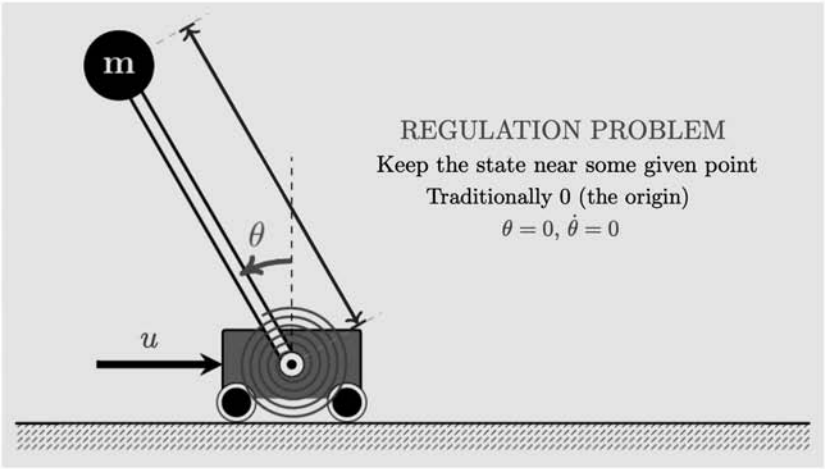

Figure 3.1.2 Illustration of a classical regulation problem, known as the "cartpole problem" or "inverse pendulum problem." The state is the two-dimensional vector of angular position and angular velocity. We aim to keep the pole at the upright position (state equal to 0) by exerting horizontal force u on the cart.

stationary system near a given state trajectory, and also, with appropriate modifications, to control problems involving disturbances. In particular, in some cases, the trajectory is treated like a sequence of set points, and the subsequently described algorithm is applied repeatedly.

We will consider a deterministic system

$$x_{k+1} = f_k(x_k, u_k)$$

whose state x_k and control u_k are finite-dimensional vectors. The cost per stage is assumed nonnegative

$$g_k(x_k, u_k) \geqslant 0, \qquad \text{for all } (x_k, u_k)$$

(e.g., a quadratic cost). We assume that the system can be kept at the origin at zero cost, i.e.,

$$f_k(0, \overline{u}_k) = 0, \quad g_k(0, \overline{u}_k) = 0\,, \qquad \text{for some control } \overline{u}_k \in U_k(0)$$

We also impose (possibly time-varying) state and control constraints

$$x_k \in X_k, \qquad u_k \in U_k(x_k), \qquad k = 0, 1, \cdots$$

We consider an infinite horizon version of the problem, i.e., for a given initial state $x_0 \in X_0$, we want to obtain a sequence $\{u_0, u_1, \cdots\}$ such that the states and controls of the system satisfy the state and control constraints, while minimizing the total cost.

Note that there are no restrictions on the sets X_k and $U_k(x_k)$: they are arbitrary, within the corresponding state and control spaces. In particular, they can be continuous/infinite or discrete/finite. The most common case in practice is when the system equation, the constraints, and the stage costs are stationary. However, nonstationary problems are also interesting, and there is no difficulty in allowing them in our analysis.

The MPC Algorithm

Let us describe the MPC algorithm for the deterministic problem just described. At the current state x_k:

(1) MPC solves an ℓ-step lookahead version of the problem, which requires that $x_{k+\ell} = 0$.† We assume that the positive integer ℓ satisfies a condition that guarantees the feasibility of this problem (see the constrained controllability condition that follows).

(2) If $\{\tilde{u}_k, \cdots, \tilde{u}_{k+\ell-1}\}$ is the optimal control sequence of this problem, MPC applies $\tilde{u}_k$ and discards the other controls $\tilde{u}_{k+1}, \cdots, \tilde{u}_{k+\ell-1}$.

(3) At the next stage, MPC repeats this process, once the next state x_{k+1} is revealed.

In particular, at the typical stage k and state $x_k \in X_k$, the MPC algorithm solves an ℓ-stage optimal control problem involving the same cost function and the requirement $x_{k+\ell} = 0$. This is the problem

$$\min_{u_t,\, t=k,\cdots,k+\ell-1} \sum_{t=k}^{k+\ell-1} g_t(x_t, u_t) \tag{3.1}$$

subject to the system equation constraints

$$x_{t+1} = f_t(x_t, u_t), \qquad t = k, \cdots, k+\ell-1$$

the state and control constraints

$$x_t \in X_t, \qquad u_t \in U_t(x_t), \qquad t = k, \cdots, k+\ell-1$$

and the terminal state constraint $x_{k+\ell} = 0$. Let $\{\tilde{u}_k, \tilde{u}_{k+1}, \cdots, \tilde{u}_{k+\ell-1}\}$ be a corresponding optimal control sequence. The MPC algorithm applies at stage k the first component $\tilde{u}_k$ of this sequence, and discards the remaining components; see Fig. 3.1.3.‡

† The constraint $x_{k+\ell} = 0$ is natural and simplifies the analysis. In practice it may be replaced by alternative more general conditions; see Section 3.1.2.

‡ In the case, where we want the system to follow a given nominal trajectory, rather than stay close to the origin, we should modify the MPC optimization to impose as a terminal constraint that the state $x_{k+\ell}$ should be a point on the nominal trajectory (instead of $x_{k+\ell} = 0$). We should also change the cost function to reflect a penalty for deviating from the given trajectory.

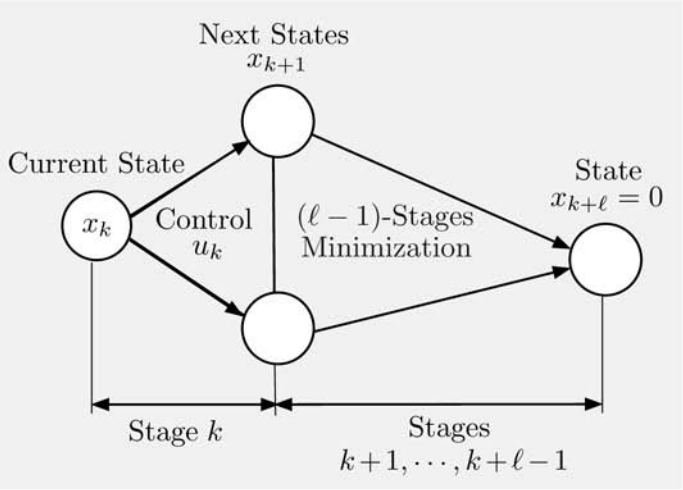

Figure 3.1.3 Illustration of the problem solved by MPC at state x_k. We minimize the cost function over the next ℓ stages while imposing the requirement that $x_{k+\ell}=0$. We then apply the first control of the optimizing sequence.

Note that we have not excluded the possibility that $U_k(x_k)$ has a discrete character, in which case the ℓ-stage MPC problem (3.1) may be an integer programming problem. We assume the following.

Constrained Controllability Condition

The integer ℓ is such that for every k and state $x_k \in X_k$, we can find a sequence of controls $u_k, \cdots, u_{k+\ell-1}$ that drive to 0 the state $x_{k+\ell}$ of the system at time $k+\ell$, while satisfying all the intermediate state and control constraints

$$u_k \in U_k(x_k),\ x_{k+1} \in X_{k+1},\ u_{k+1} \in U_{k+1}(x_{k+1}) \cdots,$$
$$x_{k+\ell-1} \in X_{k+\ell-1},\ u_{k+\ell-1} \in U_{k+\ell-1}(x_{k+\ell-1})$$

Finding an integer ℓ that satisfies the constrained controllability condition is an important issue. Generally the constrained controllability condition tends to be satisfied if the control constraints are not too stringent, and the state constraints do not allow a large deviation from the origin. In this case not only can we implement MPC, but also the resulting closed-loop system will tend to be stable; see the following discussion of stability. Note that the actual state trajectory produced by MPC may never reach the origin (see the subsequent Example 3.1.1). This is because we use only the first control $\tilde{u}_k$ of the kth stage sequence $\{\tilde{u}_k, \tilde{u}_{k+1}, \cdots, \tilde{u}_{k+\ell-1}\}$, which aims at $x_{k+\ell}=0$. At the next stage $k+1$ the control generated by MPC

may be different than $\tilde{u}_{k+1}$, because it will aim one step further to the terminal condition $x_{k+\ell+1} = 0$.

To make the connection of MPC with rollout, we note that *the one-step lookahead function $\tilde{J}$ implicitly used by MPC [cf. Eq. (3.1)] is the cost-to-go function of a certain base heuristic*. This is the heuristic that drives to 0 the state after $\ell-1$ stages (*not ℓ stages*) and keeps the state at 0 thereafter, while observing the state and control constraints, and minimizing the associated $(\ell-1)$-stages cost, in the spirit of our earlier discussion.

Sequential Improvement and Stability Analysis

It turns out that the base heuristic just described is sequentially improving, so MPC has a cost improvement property, of the type discussed in Section 2.3.1. To see this, let us denote by $\hat{J}_k(x_k)$ the optimal cost of the ℓ-stage problem solved by MPC when at a state $x_k \in X_k$. Let also $H_k(x_k)$ and $H_{k+1}(x_{k+1})$ be the optimal costs of the corresponding $(\ell-1)$-stage optimization problems that start at x_k and x_{k+1}, and drive the states $x_{k+\ell-1}$ and $x_{k+\ell}$, respectively, to 0. Thus, by the principle of optimality, we have the DP equation

$$\hat{J}_k(x_k) = \min_{u_k \in U_k(x_k)} \Big[g_k(x_k, u_k) + H_{k+1}\big(f_k(x_k, u_k)\big) \Big]$$

Since having one less stage at our disposal to drive the state to 0 cannot decrease the optimal cost, we have

$$\hat{J}_k(x_k) \leqslant H_k(x_k)$$

i.e., the cost of driving the system to 0 in $\ell-1$ stages and then staying at 0 for one more stage at no cost cannot be less than the optimal cost of driving the system to 0 in ℓ stages. By combining the preceding two relations, we obtain

$$\min_{u_k \in U_k(x_k)} \Big[g_k(x_k, u_k) + H_{k+1}\big(f_k(x_k, u_k)\big) \Big] \leqslant H_k(x_k) \tag{3.2}$$

which is the sequential improvement condition for the base heuristic (cf. Section 2.3.1).†

Often the primary objective in MPC, aside from fulfilling the state and control constraints, is to obtain a stable closed-loop system, i.e., a system that naturally tends to stay close to the origin. This is typically expressed adequately by the requirement of a finite cost over an infinite number of stages:

$$\sum_{k=0}^{\infty} g_k(x_k, u_k) < \infty \tag{3.3}$$

† Note that the base heuristic is not sequentially consistent, as it fails to satisfy the definition given in Section 2.3.1 (see the subsequent Example 3.1.1).

where $\{x_0, u_0, x_1, u_1, \cdots\}$ is the state and control sequence generated by MPC.

We will now show that the stability condition (3.3) is satisfied by the MPC algorithm. Indeed, from the sequential improvement condition (3.2), we have

$$g_k(x_k, u_k) + H_{k+1}(x_{k+1}) \leqslant H_k(x_k), \qquad k = 1, 2, \cdots \tag{3.4}$$

Adding this relation for all k in a range $[1, K]$, where $K \geqslant 1$, we obtain

$$H_{K+1}(x_{K+1}) + \sum_{k=0}^{K} g_k(x_k, u_k) \leqslant g_0(x_0, u_0) + H_1(x_1)$$

Since (in view of the nonnegativity of the cost per stage) we have

$$0 \leqslant H_{K+1}(x_{K+1})$$

it follows that

$$\sum_{k=0}^{K} g_k(x_k, u_k) \leqslant g_0(x_0, u_0) + H_1(x_1), \qquad K \geqslant 1$$

and taking the limit as $K \to \infty$, we obtain

$$\sum_{k=0}^{\infty} g_k(x_k, u_k) \leqslant g_0(x_0, u_0) + H_1(x_1) \tag{3.5}$$

The expression

$$g_0(x_0, u_0) + H_1(x_1)$$

in the right side above is the optimal cost of transfer from x_0 to $x_\ell = 0$ (i.e., the first ℓ-stage problem solved by MPC). Since this transfer is feasible by the constrained controllability condition, the above expression is finite and the stability condition (3.3) is satisfied.

The line of analysis just provided was based on rollout ideas and the sequential improvement condition (3.4). However, it is also related to well known lines of Lyapunov stability analysis in control theory; see e.g., the end-of-chapter textbook references on MPC.

Example 3.1.1

Consider a scalar linear system and a quadratic cost

$$x_{k+1} = x_k + u_k, \qquad g_k(x_k, u_k) = x_k^2 + u_k^2$$

where the state and control constraints are

$$x_k \in X_k = \{x \mid |x| \leqslant 1.5\}, \qquad u_k \in U_k(x_k) = \{u \mid |u| \leqslant 1\}$$

We apply the MPC algorithm with $\ell = 2$. For this value of ℓ, the constrained controllability assumption is satisfied, since the 2-step sequence of controls

$$u_0 = -\text{sgn}(x_0), \qquad u_1 = -x_1 = -x_0 + \text{sgn}(x_0)$$

drives the state x_2 to 0, for any x_0 with $|x_0| \leqslant 1.5$.

At state $x_k \in X_k$, MPC minimizes the two-stage cost

$$x_k^2 + u_k^2 + (x_k + u_k)^2 + u_{k+1}^2$$

subject to the control constraints

$$|u_k| \leqslant 1, \qquad |u_{k+1}| \leqslant 1$$

and the state constraints

$$|x_{k+1}| \leqslant 1.5, \qquad x_{k+2} = x_k + u_k + u_{k+1} = 0$$

This is a quadratic programming problem, which can be solved with available software, and in this case analytically, because of its simplicity. In particular, it can be verified that the minimization yields

$$\tilde{u}_k = -\frac{2}{3}x_k, \qquad \tilde{u}_{k+1} = -(x_k + \tilde{u}_k)$$

Thus the MPC algorithm selects $\tilde{u}_k = -\frac{2}{3}x_k$, which results in the closed-loop system

$$x_{k+1} = \frac{1}{3}x_k, \qquad k = 0, 1, \cdots$$

Note that while this closed-loop system is stable, its state is never driven to 0 if started from $x_0 \neq 0$. Moreover, it is easily verified that the base heuristic is not sequentially consistent. For example, starting from $x_k = 1$, the base heuristic generates the sequence

$$\left\{x_k = 1, u_k = -\frac{2}{3}, x_{k+1} = \frac{1}{3}, u_{k+1} = -\frac{1}{3}, x_{k+2} = 0, u_{k+2} = 0, \cdots\right\}$$

while starting from the next state $x_{k+1} = \frac{1}{3}$ it generates the sequence

$$\left\{x_{k+1} = \frac{1}{3}, u_{k+1} = -\frac{2}{9}, x_{k+2} = \frac{1}{9}, u_{k+2} = -\frac{1}{9}, x_{k+3} = 0, u_{k+3} = 0, \cdots\right\}$$

so the sequential consistency condition of Section 2.3.1 is violated.

Regarding the choice of the horizon length ℓ for the MPC calculations, note that if the constrained controllability assumption is satisfied for some value of ℓ, it is also satisfied for all larger values of ℓ. This argues for a large value of ℓ. On the other hand, the optimal control problem solved at each stage by MPC becomes larger and hence more difficult as ℓ increases. Thus, the horizon length is usually chosen on the basis of some experimentation: first ensure that ℓ is large enough for the constrained controllability assumption to hold, and then by further experimentation to ensure overall satisfactory performance.

3.1.1 Target Tubes and Constrained Controllability

We now return to the constrained controllability condition, which asserts that the state constraint sets X_k are such that starting from anywhere within X_k, it is possible to drive to 0 the state of the system within some number of steps ℓ, while staying within X_m at each intermediate step $m = k+1, \cdots, m = k+\ell-1$. Unfortunately, this assumption masks some major complications.† In particular, the control constraint set may not be sufficiently rich to compensate for natural instability tendencies of the system. As a result it may be impossible to keep the state within X_k over a sufficiently long period of time, something that may be viewed as a form of instability. Here is an example involving an unstable system and inadequate control constraints.

Example 3.1.2

Consider the scalar linear system

$$x_{k+1} = 2x_k + u_k$$

which is unstable, and the control constraint

$$|u_k| \leqslant 1$$

Then if $0 < x_0 < 1$, it can be seen that by using the control $u_0 = -1$, the next state satisfies,

$$x_1 = 2x_0 - 1 < x_0$$

and is closer to 0 than the preceding state x_0. Similarly, using controls $u_k = -1$, every subsequent state x_{k+1} will get closer to 0 than x_k. Eventually, after a sufficient number of steps $\bar{k}$, with controls $u_k = -1$ for $k < \bar{k}$, the state $x_{\bar{k}}$ will satisfy

$$0 \leqslant x_{\bar{k}} \leqslant \frac{1}{2}$$

Once this happens, the feasible control $u_{\bar{k}} = -2x_{\bar{k}}$ will drive the state $x_{\bar{k}+1}$ to 0.

Similarly, when $-1 < x_0 \leqslant 0$, by applying control $u_k = 1$ for a sufficiently large number of steps $\bar{k}$, the state $x_{\bar{k}}$ will be driven into the region $[-1/2, 0]$, and then the feasible control $u_{\bar{k}} = -2x_{\bar{k}}$ will drive the state $x_{\bar{k}+1}$ to 0.

Suppose now that the state constraint is of the form $X_k = [-\beta, \beta]$ for all k, and let us explore what happens for different values of β. The preceding discussion shows that if $0 < \beta < 1$ the constrained controllability assumption

† Fundamentally, when the constrained controllability condition is not known to hold, we are essentially dealing with a constrained form of rollout, whereby the present control choice may affect the feasibility of future state constraints. Constrained rollout will be discussed in Section 3.3.