

NUMERICAL OPTIMIZATION-BASED DESIGN

V. Balakrishnan, Purdue University

A. L. Tits, University of Maryland

1 Introduction

The fundamental control system design problem is the following: *Given a system that does not perform satisfactorily, devise a strategy to improve its performance.* Often, quantitative measures are available to evaluate the performance of the system. **Numerical optimization-based design**, or for short, optimization-based design, is design with the goal of optimizing these measures of performance, using iterative optimization methods implemented on a computer.

Optimization-based design has a relatively short history compared to other areas of control; this is because the very factors that have led to the evolution of optimization-based design into a popular and useful design procedure are themselves of recent vintage. The first is the exploding growth in computer power; the second, recent breakthroughs in optimization theory and algorithms, especially **convex optimization**; and the third, recent advances in numerical linear algebra.

The optimization-based design methods that we will consider in this article must be contrasted against classical optimal control methods. Loosely speaking, classical optimal control methods rely on theoretical techniques (variational arguments or dynamic programming, for example) to derive the optimal control strategy (or conditions that characterize it). This is a very powerful approach because “exact” strategies can be obtained and limits of performance can be derived. However, this approach only applies to very restricted classes of models (such as lumped linear time-invariant (LTI) systems, for instance) and the class of performance measures that can be handled is very limited. Also, many design constraints cannot be handled by this approach. The methods we consider in this article, on the other hand, do not yield analytical solutions; instead the design problem is posed as a numerical optimization problem, typically either minimization (or maximization) or min-max

(or max-min). This optimization problem is then solved numerically on a computer. This approach can tackle a much wider variety of problems when compared with the analytical approach, usually at the cost of much-increased computation. Still, we should stress that in some cases (especially with convex optimization methods), the computation required for optimization-based design is comparable to that required for the evaluation of related analytical solutions. For this reason, optimization-based design methods often compete quite favorably with related analytical design methods.

Our intent in this article is to provide the reader with a quick introduction to several important control system analysis and design methods that rely in an essential way on numerical optimization. Whenever possible, we have striven to keep tedious details to the minimum, since these tend to obscure the ideas underlying the methods; the reader interested in details or technical conditions can turn to the list of references appearing at the end. We will also not cover several important optimization-based methods that appear elsewhere in the handbook; see the other articles under “Design Methods for MIMO LTI Systems”. We will only be considering continuous-time systems in this article; most design methods that we discuss can be extended to discrete-time systems in a straightforward manner.

2 A framework for controller design

The framework on which we will focus our discussion is shown in Figure 1. P is the model of the plant, that is, of the system to be controlled. K is the controller that implements the control strategy for improving the performance of the system. y is the signal that the controller has access to, and u is the output of the controller that drives the plant. w and z represent inputs and outputs of interest, so that the map from w to z contains all the input-output maps of interest. We let n_w , n_u , n_z and n_y denote the sizes (i.e., number of components) of w , u , z and y respectively.

The choice of an approach for tackling a control design problem within this framework must take the following three factors into consideration.¹

¹Also note that these factors are in general not “God-given”, and that a trial and error approach is

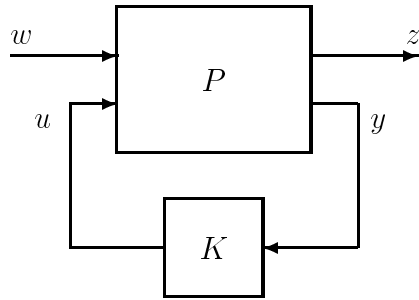


Figure 1: A standard controller design framework

The plant model P . Depending on the particular modeling, system identification and validation procedures used, the plant model P may or may not be linear, may or may not be time-invariant; it may also be unspecified except for the constraint that it lie within a given set Π (“uncertainty modeling”).

The class of controllers K . The controller could be static or dynamic, linear or nonlinear, time-invariant or not; it could be a state-feedback (i.e., the measured output y is the state of P) or an output-feedback controller; nonlinear controllers could be restricted to, say, gain-scheduled controllers, or to linear controllers augmented with output saturation; the order of the controller or its internal structure also could be specified or restricted.

The performance specifications. Perhaps the most important of these is the requirement that the nominal closed-loop system be stable. Typical performance specifications, beyond stability, require some desirable property for the map from w to z . Often, it is required that K be designed to minimize some norm of the map from w to z , subject to constraints on other norms, as well as other constraints on the behavior of the closed-loop system (e.g., constraints on the decay rate of trajectories, constraints on the response to some specific reference inputs such as steps, etc.). In cases when the plant is not completely specified, but is known only to lie in some set Π , these performance specifications (in particular, closed-loop stability) may be required for every $P \in \Pi$ (this is called “worst-case design”), or they may be required “on the average” (roughly speaking, this means that with the controller K in generally required).

place, the “average behavior” of the closed-loop system is satisfactory).

In the simplest cases, numerical optimization can be bypassed altogether. A typical example is the linear quadratic Gaussian (LQG) problem, which applies to LTI models with a quadratic performance index; see the articles under “Kalman Filter and Observers”, and the articles “LQG/LTR” and “ \mathbf{H}_2 , \mathbf{H}_∞ ” for details. In most cases, however, numerical optimization is called for. These cases constitute the focus of this article. Sections 3 and 4 deal with a class of problems that enjoy certain convexity properties, making them, in a sense that we will describe, easier to solve. In Section 3, we consider perhaps the simplest paradigm, that of designing LTI controllers for a plant which is assumed to be LTI and known exactly. It will turn out that for a wide variety of performance specifications, this problem can be solved “easily”, using convex optimization. Then, in Section 4, we turn to the harder problem of worst-case design of LTI controllers when the plant is known to belong to a certain class. It turns out that when the controller is restricted to be static linear state-feedback, the design problem can be solved via convex optimization in a number of cases. The class of problems that can be tackled by convex optimization is large enough to be of definite interest. However, in practical situations, it is often necessary to deal with models or impose restrictions that rule out convex optimization. In such cases one can resort to local optimization methods. While it is not guaranteed that the global optimum will be achieved (at least with reasonable computation), the class of problems that can be tackled is very large; in particular, there is little restriction on the possible plant models, and the main restriction on the class of controllers is that it must be parametrized by a finite number of real parameters. This is discussed in Section 5. When the plant model is highly nonlinear and the controller is allowed to be time-varying, the solution of an open-loop optimal control problem is often a key stepping stone. Numerical methods for the solution of such problems are considered in Section 6. Finally, in Section 7, we discuss multiobjective problems and tradeoff exploration.

3 LTI controllers for LTI systems: Design based on the Youla parameter

For the special case when the plant is modeled as a finite-dimensional LTI system, and LTI controllers are sought, the **Youla parametrization** of the set of all achievable stable closed-loop maps can be combined with convex optimization to design optimal (or, more often, suboptimal) controllers.

Consider the controller design framework in Figure 1, where the plant P has a transfer function

$$P(s) = \begin{bmatrix} P_{zw}(s) & P_{zu}(s) \\ P_{yw}(s) & P_{yu}(s) \end{bmatrix}, \quad (1)$$

where P_{zw} is the open-loop (i.e., with the controller removed from the loop) transfer function of the plant from w to z , P_{zu} that from u to z , P_{yw} that from w to y , and P_{yu} that from u to y . Then, with $K(s)$ denoting the transfer function of the LTI controller, the transfer function from w to z is

$$H_{\text{cl}}(s) = P_{zw}(s) + P_{zu}(s)K(s)(I - P_{yu}(s)K(s))^{-1}P_{yw}(s). \quad (2)$$

H_{cl} is called the “closed-loop” transfer function from w to z . Note that z can include components of the control input u , so that specifications on the control input such as bounds on the control effort can be handled. The set of achievable, stable closed-loop maps from w to z is given by

$$\mathcal{H}_{\text{cl}} = \{H_{\text{cl}} : H_{\text{cl}} \text{ satisfies (2), } K \text{ stabilizes the system}\}. \quad (3)$$

The set of the controllers K that stabilize the system is in general not a convex set. Thus optimizing over \mathcal{H}_{cl} using the description (3), with K as the (infinite-dimensional) optimization variable, is a very hard numerical problem. However, it turns out (see [Boyd and Barratt, 1991] for details) that the set \mathcal{H}_{cl} can be also written as

$$\mathcal{H}_{\text{cl}} = \{H_{\text{cl}} : H_{\text{cl}}(s) = T_1(s) + T_2(s)Q(s)T_3(s), Q \text{ is stable}\}, \quad (4)$$

where T_1 , T_2 and T_3 are fixed, stable transfer matrices that can be computed from the data characterizing P . Moreover, given Q , the corresponding controller K can be immediately

computed as $K = \mathcal{K}(Q)$ (where $\mathcal{K}(\cdot)$ is a certain rational function of Q). The most important observation about this reparametrization of \mathcal{H}_{cl} is that it is *affine* in the infinite-dimensional parameter Q ; it is therefore a convex parametrization of the set of achievable stable closed-loop maps from w to z . (The parameter Q is also referred to as the Youla parameter.) This obvious fact has an important ramification—it is possible now to use convex optimization techniques to find an optimal parameter Q_{opt} , and therefore an optimal controller $\mathcal{K}(Q_{\text{opt}})$.

Example: We demonstrate the affine reparametrization of \mathcal{H}_{cl} with a simple example. Let

$$P(s) = \begin{bmatrix} P_{zw}(s) & P_{zu}(s) \\ P_{yw}(s) & P_{yu}(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{s+2} & 1 \\ 1 & \frac{1}{s+2} \end{bmatrix}. \quad (5)$$

Then,

$$\mathcal{H}_{\text{cl}} = \left\{ H_{\text{cl}} : \frac{1}{s+2} + K(s) \left(1 - \frac{1}{s+2} K(s) \right)^{-1}, K \text{ stabilizes system} \right\}. \quad (6)$$

The affine parametrization is simply

$$\mathcal{H}_{\text{cl}} = \left\{ H_{\text{cl}} : \frac{1}{s+2} + Q(s), Q \text{ is stable} \right\}. \quad (7)$$

Moreover, given the Youla parameter Q , the corresponding controller is given by

$$K(s) = \left(1 + \frac{Q(s)}{s+2} \right)^{-1} Q(s). \quad (8)$$

The reparametrization in this example is particularly simple— $T_1 = P_{zw}$, $T_2 = P_{zu}$ and $T_3 = P_{yw}$ —since the open-loop transfer function from u to y (i.e., P_{yu}) is stable to start with. In cases when P_{yu} is unstable, the parametrization is more complicated; see for example [Boyd and Barratt, 1991]. \diamond

The general procedure for designing controllers using the Youla parametrization proceeds as follows. Let $\phi_0, \phi_1, \dots, \phi_m$ be (not necessarily differentiable) convex functionals on the closed-loop map that represent performance measures. These performance measures may be norms (typically \mathbf{H}_2 or \mathbf{H}_∞ norms), certain time-domain quantities (step response overshoot, steady-state errors), etc.

Then the problem

$$\begin{aligned} & \text{minimize, w.r.t. } Q: && \phi_0(T_1 + T_2 Q T_3) \\ & \text{subject to:} && \phi_1(T_1 + T_2 Q T_3) \leq \alpha_1 \\ & && \vdots \\ & && \phi_m(T_1 + T_2 Q T_3) \leq \alpha_m \end{aligned} \quad (9)$$

is a convex optimization problem (with an infinite-dimensional optimization variable Q), since it has the form “Minimize a convex function subject to convex constraints”. This problem has the interpretation of minimizing a measure of performance of the closed-loop system subject to other performance constraints.

In practice, problem (9) is solved by searching for Q over a finite-dimensional subspace. Typically, Q is restricted to lie in the set

$$\{Q : Q = \beta_1 Q_1 + \dots + \beta_n Q_n\}, \quad (10)$$

where Q_1, \dots, Q_n are stable, fixed, transfer matrices, and scalar β_1, \dots, β_n are the optimization variables. This enables us to solve problem (9) “approximately” by solving the following problem with a finite number of scalar optimization variables:

$$\begin{aligned} & \text{minimize, w.r.t. } \beta_1, \dots, \beta_n: && \phi_0(T_1 + T_2 Q(\beta) T_3) \\ & \text{subject to:} && \phi_1(T_1 + T_2 Q(\beta) T_3) \leq \alpha_1 \\ & && \vdots \\ & && \phi_m(T_1 + T_2 Q(\beta) T_3) \leq \alpha_m \end{aligned} \quad (11)$$

where $Q(\beta) = \beta_1 Q_1 + \dots + \beta_n Q_n$.

Approximating the infinite-dimensional parameter Q by a finite-dimensional quantity is referred to as a “Ritz approximation”. Evidently, the transfer matrices Q_i and their number, n , should be so chosen that the optimal parameter Q can be approximated with sufficient accuracy.

Example: With the same plant as in (5), let us take ϕ_0 to be the \mathbf{H}_∞ norm (i.e., peak value of the transfer function magnitude) of H_{cl} , ϕ_1 to be the steady-state magnitude of z for a unit-step input at w (this is just the transfer function magnitude at DC), and $\alpha_1 = 0.1$. The design problem may thus be viewed as the minimization of the closed-loop \mathbf{H}_∞ norm, subject to a DC disturbance rejection constraint.

Let us approximate Q by $Q(\beta) = \beta_1 + \beta_2 \frac{1}{s+1}$. Thus,

$$\phi_0(T_1 + T_2 Q T_3) = \left\| \frac{1}{s+2} + \beta_1 + \beta_2 \frac{1}{s+1} \right\|_\infty, \quad \phi_1(T_1 + T_2 Q T_3) = |0.5 + \beta_1 + \beta_2|. \quad (12)$$

Then, the optimization problem (11) becomes

$$\begin{aligned} & \text{minimize, w.r.t. } \beta_1, \beta_2: && \left\| \frac{1}{s+2} + \beta_1 + \beta_2 \frac{1}{s+1} \right\|_\infty \\ & \text{subject to:} && |0.5 + \beta_1 + \beta_2| \leq 0.1 \end{aligned} \quad (13)$$

◇

The most important observation concerning our reduction of the LTI controller design to problem (11) is that it is a convex optimization problem with a finite number of optimization variables. Convexity has several important implications:

- Every stationary point of the optimization problem (11) is also a global minimizer.
- The problem can be solved in polynomial-time.
- We can immediately write down necessary and sufficient optimality conditions.
- There is a well-developed duality theory.

From a practical standpoint, there are effective and powerful algorithms for the solution of problems such as (11), that is, algorithms that rapidly compute the global optimum, with non-heuristic stopping criteria. These algorithms range from simple descent-type or quasi-Newton methods for smooth problems to sophisticated cutting-plane or interior-point methods for non-smooth problems. A comprehensive literature is available on algorithms for convex programming; see for example, [Hiriart-Urruty and Lemaréchal, 1993] and [Nesterov and Nemirovskii, 1994]; see also [Boyd and Barratt, 1991].

4 LTI controllers for uncertain systems: LMI synthesis

We now outline one convex optimization-based approach that is applicable when the plant P is not known exactly, but is known only to belong to a set Π of a certain type. This approach is controller design based on **Linear Matrix Inequalities** (LMIs).

4.1 Optimization over Linear Matrix Inequalities

A linear matrix inequality is a matrix inequality of the form

$$F(\zeta) \triangleq F_0 + \sum_{i=1}^m \zeta_i F_i > 0, \quad (14)$$

where $\zeta \in \mathbf{R}^m$ is the variable, and $F_i = F_i^T \in \mathbf{R}^{n \times n}$, $i = 0, \dots, m$ are given. The inequality symbol in (14) means that $F(\zeta)$ is positive-definite, i.e., $u^T F(\zeta) u > 0$ for all nonzero $u \in \mathbf{R}^n$.

The set $\{\zeta \mid F(\zeta) > 0\}$ is convex. (We have used strict inequality mostly as a convenience; inequalities of the form $F(\zeta) \geq 0$ are also readily handled.)

Multiple LMIs

$$F_1(\zeta) > 0, \dots, F_n(\zeta) > 0 \tag{15}$$

can be expressed as the single LMI

$$\begin{bmatrix} F_1(\zeta) & 0 & \dots & 0 \\ 0 & F_2(\zeta) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_n(\zeta) \end{bmatrix} > 0. \tag{16}$$

Therefore we will make no distinction between a set of LMIs and a single LMI, i.e., “the LMI $F_1(\zeta) > 0, \dots, F_n(\zeta) > 0$ ” will mean “the LMI $\mathbf{diag}(F_1(\zeta), \dots, F_n(\zeta)) > 0$ ”. When the matrices F_i are diagonal, the LMI $F(\zeta) > 0$ is just a set of linear inequalities.

For many problems, the variables are matrices, e.g.,

$$A^T P + P A < 0 \tag{17}$$

where $A \in \mathbf{R}^{n \times n}$ is given and $P = P^T$ is the variable. In this case we will not write out the LMI explicitly² in the form $F(\zeta) > 0$, but instead make clear which matrices are the variables. The phrase “the LMI $A^T P + P A < 0$ in P ” means that the matrix P is the variable.

LMI feasibility problems. Given an LMI $F(\zeta) > 0$, the corresponding LMI Problem (LMIP) is to find ζ^{feas} such that $F(\zeta^{\text{feas}}) > 0$ or determine that the LMI is infeasible. Of course, this is a convex feasibility problem. We will say “solving the LMI $F(\zeta) > 0$ ” to mean solving the corresponding LMIP.

Eigenvalue problems. The eigenvalue problem (EVP) is to minimize³ the maximum eigenvalue of a matrix, subject to an LMI, which is equivalently expressed as

$$\begin{aligned} &\text{minimize, w.r.t. } \zeta \text{ and } \lambda: && \lambda \\ &\text{subject to: } && \lambda I - A(\zeta) > 0, \quad B(\zeta) > 0. \end{aligned} \tag{18}$$

²This can be done as follows. Let P_1, \dots, P_m be a basis for symmetric $n \times n$ matrices ($m = n(n+1)/2$). Then take $F_0 = 0$ and $F_i = -A^T P_i - P_i A$.

³Technically, we seek the infimum, rather than the minimum (and corresponding parameter values in the closure of the constraint set).

Here, A and B are symmetric matrices that depend affinely on the optimization variable ζ . This is a convex optimization problem.

Another LMI-based optimization problem that arises frequently in optimization-based design is the generalized eigenvalue problem or GEVP; see [Boyd et al., 1994] for details.

4.2 Analysis and design of uncertain controls systems using LMIs

We next describe the set of plants for which we will synthesize controllers using LMI-based optimization. The set Π is described by the following state equations:

$$\begin{aligned} \dot{x} &= A(t)x + B_u(t)u + B_w(t)w, & x(0) &= x_0, \\ z &= C_z(t)x + D_{zu}(t)u + D_{zw}(t)w, \end{aligned} \tag{19}$$

where the matrices in (19) are unknown except for the fact that they satisfy

$$\begin{bmatrix} A(t) & B_u(t) & B_w(t) \\ C_z(t) & D_{zu}(t) & D_{zw}(t) \end{bmatrix} \in \Omega \quad \text{for all } t \geq 0, \tag{20}$$

where $\Omega \subseteq \mathbf{R}^{(n+n_z) \times (n+n_u+n_w)}$ is a convex set of a certain type. In some applications we can have one or more of the integers n_u , n_w , and n_z equal to zero, which means that the corresponding variable is not used. For example, when $n_u = n_w = n_z = 0$, the set Π is described by $\{\dot{x} = A(t)x \mid A(t) \in \Omega\}$.

With the appropriate choice of Ω , a number of common control system models can be described in the framework of (20): LTI systems; polytopic systems, norm-bound systems, structured norm-bound systems; systems with parametric perturbations; systems with structured, bounded LTI perturbations, etc. For details, see [Boyd et al., 1994]. For purposes of illustration, we now consider one particular system model in detail, *viz.* polytopic systems (PS).

Polytopic system models arise when the uncertain plant is modeled as a linear time-varying system with state-space matrices that lie in a polytope—thus, Ω is a polytope described as the convex hull of its vertices

$$\mathbf{Co} \left\{ \begin{bmatrix} A_1 & B_{u,1} & B_{w,1} \\ C_{z,1} & D_{zu,1} & D_{zw,1} \end{bmatrix}, \dots, \begin{bmatrix} A_L & B_{u,L} & B_{w,L} \\ C_{z,L} & D_{zu,L} & D_{zw,L} \end{bmatrix} \right\}, \tag{21}$$

where the matrices in (21) are given, and \mathbf{Co} stands for the convex hull, defined by

$$\mathbf{Co} \{G_1, \dots, G_L\} \triangleq \left\{ G : G = \sum_{i=1}^L \lambda_i G_i, \lambda_i \geq 0, \sum_{i=1}^L \lambda_i = 1 \right\}. \quad (22)$$

Example: Consider the linear time-varying system with two parameters $q_1(t)$ and $q_2(t)$:

$$\frac{dx}{dt} = \begin{bmatrix} 0 & 1 \\ -q_1(t) & -q_2(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t); \quad y(t) = [1 \ 0] x(t), \quad (23)$$

with $q_1(t) \in [-1, 1]$ and $q_2(t) \in [-2, 2]$ for all $t \geq 0$. The corresponding polytopic system model is

$$\frac{dx}{dt} = A(t) x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t); \quad y(t) = [1 \ 0] x(t), \quad (24)$$

where

$$A(t) \in \mathbf{Co} \left\{ \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \right\}. \quad (25)$$

◇

Suppose we have a physical plant that is fairly well modeled as a (possibly time-varying) linear system. We collect many sets of input-output measurements, obtained at different times, under different operating conditions, or perhaps from different instances of the system (e.g., different units from a manufacturing run). It is important that we have data sets from enough plants or plant conditions to characterize or at least give some idea of the plant variation that can be expected. For each data set we estimate a time-invariant linear system model of the plant. To simplify the problem we will assume that the state in this model is accessible, so the different models refer to the same physical state vector. These models should be fairly close, but of course not exactly the same. Then, the original plant can be modeled as a PS with the vertices given by the estimated linear system models. In other words, we model the plant as a time-varying linear system, with system matrices that can jump around among any of the models we estimated. It seems reasonable to conjecture that a controller that works well with this PS is likely to work well when connected to the original plant.

A number of problems in the analysis and design for the aforementioned system models can be reformulated as one of the LMI optimization problems, namely LMIP or

EVP. We will demonstrate this reformulation for PS, for a sample set of simple problems; we will also list a few more problems in Section 4.2.4. For a more comprehensive list, we refer the reader to [Boyd et al., 1994].

4.2.1 Stability of polytopic systems

First let us consider the problem of stability of the polytopic system PS, i.e., we consider the problem of determining whether all the trajectories of the system

$$\frac{dx}{dt} = A(t)x(t), \quad A(t) \in \mathbf{Co}\{A_1, \dots, A_L\} \quad (26)$$

converge to zero as $t \rightarrow \infty$. A sufficient condition for this is the existence of a quadratic positive function $V(z) = z^T P z$ such that $dV(x(t))/dt < 0$ for any trajectory of (26). Since

$$\frac{d}{dt}V(x(t)) = x(t)^T (A(t)^T P + P A(t)) x(t), \quad (27)$$

a sufficient condition for stability is the existence of a P satisfying the condition

$$P > 0, \quad A(t)^T P + P A(t) < 0, \quad A(t) \in \mathbf{Co}\{A_1, \dots, A_L\}. \quad (28)$$

If there exists such a P , we say the PS (26) is “quadratically stable”.

Condition (28) is equivalent to

$$P > 0, \quad A_i^T P + P A_i < 0, \quad i = 1, \dots, L, \quad (29)$$

which is an LMI in P . V is sometimes called a “simultaneous quadratic Lyapunov function” since it proves stability of each of A_1, \dots, A_L . Thus, determining quadratic stability is an LMIP.

4.2.2 Stabilizing state-feedback synthesis for polytopic systems

Consider the system (26) with state-feedback:

$$\frac{dx}{dt} = A(t)x(t) + B_u(t)u(t), \quad u(t) = Kx(t), \quad (30)$$

where

$$[A(t) \ B_u(t)] \in \mathbf{Co}\{[A_1 \ B_{u,1}], \dots, [A_L \ B_{u,L}]\}. \quad (31)$$

Our objective is to design the matrix K such that (30) is quadratically stable. This is the “quadratic stabilizability” problem.

System (30) is quadratically stable for some state-feedback K if there exist P and K such that

$$P > 0, \quad (A_i + B_i K)^T P + P(A_i + B_i K) < 0, \quad i = 1, \dots, L. \quad (32)$$

Note that this matrix inequality is *not* jointly convex in P and K . However, with the bijective transformation $Y \triangleq P^{-1}$, $W \triangleq K P^{-1}$, we may rewrite it as

$$Y > 0, \quad (A_i + B_i W Y^{-1})^T Y^{-1} + Y^{-1}(A_i + B_i W Y^{-1}) < 0 \quad i = 1, \dots, L. \quad (33)$$

Multiplying the second inequality on the left and right by Y (such a congruence preserves the inequality) we get an LMI in Y and W :

$$Y > 0, \quad Y A_i^T + W^T B_i^T + A_i Y + B_i W < 0, \quad i = 1, \dots, L. \quad (34)$$

If this LMIP in Y and W has a solution, then the Lyapunov function $V(z) = z^T Y^{-1} z$ proves the quadratic stability of the closed-loop system with state-feedback $u(t) = W Y^{-1} x(t)$.

In other words, we can synthesize a linear state-feedback for the PS (26) by solving an LMIP.

4.2.3 More general stabilizing feedback synthesis for PS

We have shown that synthesizing state-feedback that renders a PS quadratically stable is equivalent to solving an LMIP. However, in general, the entire state of the PS may not be available for feedback. What is typically available is some linear combination of the states, so that we have

$$\frac{dx}{dt} = A(t)x(t) + B_u(t)u(t), \quad y(t) = C_y x(t) \quad (35)$$

where once again,

$$[A(t) \ B_u(t)] \in \mathbf{Co} \{ [A_1 \ B_{u,1}], \dots, [A_L \ B_{u,L}] \}. \quad (36)$$

Our objective is to design the *output-feedback* matrix K such that with $u(t) = K y(t)$, the PS (30) is quadratically stable.

Though this problem appears to be only a minor variation of the state-feedback synthesis problem, it turns out to be *much harder*. Here is why: Using the same steps as with state-feedback synthesis, we conclude that system (35) is quadratically stable for some output-feedback K if there exist $P > 0$ and K such that

$$(A_i + B_i K C)^T P + P(A_i + B_i K C) < 0, \quad i = 1, \dots, L. \quad (37)$$

This matrix inequality is *not* jointly convex in P and K ; moreover, unlike the state-feedback case, there are no known procedures (change of variables, for instance) that convert this problem to a convex feasibility problem. We are therefore left with having to solve a nonconvex feasibility problem.

A simple heuristic iterative procedure, which is along the lines of the D - K iteration of μ -synthesis (see the article “ μ -Analysis and Synthesis”) can be employed to tackle this nonconvex optimization problem. The basic idea is this. The nonconvex matrix inequality (37) is feasible if and only if the minimum value of the optimization problem

$$\begin{aligned} & \text{minimize, w.r.t. } P, K \text{ and } \gamma: \quad \gamma \\ & \text{subject to:} \quad (A_i + B_i K C)^T P + P(A_i + B_i K C) < \gamma I, \quad i = 1, \dots, L \end{aligned} \quad (38)$$

is negative. Now, problem (38) is an EVP in P and γ for fixed K , and is an EVP in K and γ for fixed P . Therefore, one can iterate between solving EVPs with respect to P and γ , and K and γ respectively, in order to attempt to solve the inequality (37). Details can be found in [El Ghaoui and Balakrishnan, 1994].

4.2.4 Other problems from control

A number of other control problems can be solved using LMI-based optimization; see [Boyd et al., 1994] for details. The list includes: computation of ellipsoidal bounds on the state; computation of bounds on the decay-rate for state trajectories; stability margin bounds; ellipsoidal bounds on the set of states reachable with various input constraints; bounds on the output energy given a certain initial state condition; bounds on the RMS value of the output for white noise inputs; bounds on the RMS gain from given inputs to outputs,

etc. Moreover, feedback design to optimize all these bounds can be tackled as well. Other problems on the list are synthesis of gain-scheduled state-feedback; some linear controller design problems using the Youla parameter and Ritz approximations; synthesis of multipliers for stability and performance analysis of systems with structured perturbations, etc.

4.3 Solving LMI problems

We have already observed that the LMI-based optimization problems that we have described are convex optimization problems, with a finite number of variables. Thus the comments that we made about convex optimization problems at the end of Section 3 apply to LMIP and EVP. Furthermore, a number of specialized interior-point algorithms that exploit the special structure of LMI problems have been recently developed, for example [Nesterov and Nemirovskii, 1994]; software packages implementing these algorithms are available as well (LMI Lab [Gahinet and Nemirovskii, 1993] and LMITOOL [El Ghaoui et al., 1995]).

5 More general optimization for controller design

The design approaches outlined in Sections 3 and 4 are efficient and mathematically elegant. However this is achieved at the cost of reduced flexibility in various respects. Namely, (i) specifications without certain convexity properties cannot be handled gracefully; (ii) constraints on the controller structure (e.g., controller order) cannot be dealt with; and (iii) the approaches are restricted to LTI models, or to cases when the plant sets Π are restricted to certain specific classes. In this section, we briefly survey approaches that are applicable to some of these more general situations.

5.1 Branch and bound methods

Often, the controller design problem can be posed as one of scalar parameter selection so as to optimize some performance objective, where each parameter is to be chosen from some specified interval (thus the parameters are required to lie in a “rectangle”). Such a situation arises, for instance, when one wishes to directly choose the coefficients of the transfer function

of the controller.

Suppose that it is easy to compute upper and lower bounds for the optimum value of the performance objective over any given rectangle of values for the parameters, with the property that the difference between the upper and lower bounds uniformly converges to zero when the rectangle shrinks to a point. This is the case, for example, with common bounds on quantities such as the decay-rate of trajectories, or \mathbf{H}_2 and \mathbf{H}_∞ norms of transfer functions of interest [Balakrishnan and Boyd, 1992]. Then **branch and bound algorithms** can be used to find the globally optimal parameter values and the corresponding value of the performance objective. The branch and bound scheme proceeds as follows. First, upper and lower bounds are computed over the original parameter rectangle. These bounds are further refined by breaking up the parameter rectangle into subrectangles (“branching”) to derive bounds for the global optimum over the original rectangle (“bounding”). The branching is done based on some heuristic rules. As they progress, branch and bound algorithms maintain upper and lower bounds for the global optimum; thus termination at any time yields guaranteed bounds for the optimum.

For an account of the solution of some parameter problems from control via branch and bound methods, see [Balakrishnan and Boyd, 1992].

5.2 Local methods

Suppose first that we are still dealing with an LTI model and LTI controller, with no pre-specified controller structure, but that some constraints are not readily expressible by means of convex or quasi-convex functions. Note that the stability requirement still can be handled implicitly via the Youla parametrization. If all other specifications are “smooth enough”, such problems can be readily tackled by “classical” **nonlinear programming** techniques. Under mild assumptions, such techniques will construct a local minimizer (maximizer), which in fact might well be global. Often, such problems will include “functional” constraints, i.e., constraints of the type

$$\phi(\zeta, \alpha) \leq 0 \quad \forall \alpha \in \Omega, \tag{39}$$

where ζ is the vector of optimization variables (for instance, the vector of components β_i as in (11)), α is an index (e.g., time or frequency), and Ω is a compact set. A functional constraint of this type arises, for instance, when the frequency response of the controller is required to satisfy $\|K(j\omega)\| \leq 1$ for $\omega \in \mathbf{R}$ (where $\|\cdot\|$ is some suitable norm). Such a constraint would be useful in limiting the size of the actuator signal u .

Problems with functional constraints of the form (39) are often called “semi-infinite”; see, e.g., [Hettich and Kortanek, 1993] or [Polak, 1996]. Note that, even if all specifications are convex (or quasi-convex), classical techniques may constitute a valid alternative to modern convex optimization techniques for such problems.

A more drastic departure from the approach discussed in the previous section is necessary when the controller structure is constrained, e.g., when the order of the controller is prescribed. Such restrictions cannot be expressed in a simple manner in terms of the Q parameter. In such cases, it is much more natural to use a controller parametrization directly linked to the structure, e.g., coefficients of transfer functions numerators and denominators, location of zeros and poles, etc. However, stability of the closed-loop system is then no longer automatically ensured, but must be enforced explicitly. A similar situation arises when the model P or the class of controller under consideration is not LTI.

The simplest way to impose closed-loop stability in such cases is perhaps to make direct use of its definition. For most practical engineering problems, this is tantamount to the condition that bounded inputs give rise to bounded outputs, provided, of course, that the inputs and outputs are chosen wisely (in particular, any internal instability of the system should be reflected in its input-output behavior). In practice, it clearly makes sense to require a little more than this, namely, that for bounded inputs, the outputs be less than some specific values at all times or, in practice, on a long enough time interval. If both the plant and the controller are LTI, however, closed-loop stability can also be enforced by invoking stability criteria such as those of Routh-Hurwitz, Nyquist, or Lyapunov. For instance, suppose that ψ denotes the vector of controller parameters, and $A(\psi)$ denotes the state dynamics matrix of the closed-loop system. Then, as discussed in Section 4.2.1,

stability is equivalent to the existence of a symmetric matrix P satisfying

$$P > 0, \quad A(\psi)^T P + P A(\psi) < 0, \quad (40)$$

or equivalently,

$$\langle u, P u \rangle > 0 \quad \forall u \in \{u \in \mathbf{R}^n : \|u\|_2 = 1\} \quad \text{and} \quad (41)$$

$$\langle u, (A(\psi)^T P + P A(\psi)) u \rangle > 0 \quad \forall u \in \{u \in \mathbf{R}^n : \|u\|_2 = 1\}. \quad (42)$$

The latter constraints are of the form (39) (semi-infinite problem) with the components of ζ being those of P and ψ , and where the role of α is now played by u . This and other schemes are discussed in [Mayne and Polak, 1993].

In Section 4 it was stressed that LMIs make a powerful tool for the design of *robust* control systems when the models $P \in \Pi$ are linear and certain other conditions are met. In other cases when the model is LTI, tools such as \mathbf{H}_∞ synthesis (see the articles “LQG/LTR” and “ \mathbf{H}_2 , \mathbf{H}_∞ ”), μ -synthesis (see the article “ μ -Analysis and Synthesis”) and ℓ_1 -synthesis (see the article “ ℓ_1 Robust Control: Theory, Computation and Design”) can be used. Constraints on the spectral norm (largest singular value; see the articles entitled “Matrices and Linear Algebra” and “MIMO Frequency Response, MIMO Nyquist Criterion, Singular Values”), on the structured singular value (or, rather, on its upper bound), or on the ℓ_1 structured norm of selected transfer functions (e.g., sensitivity and complementary sensitivity) can also be included together with the other constraints in an optimization problem to be solved by general nonlinear programming methods (see [Mayne and Polak, 1993] for a semi-infinite formulation of spectral norm constraints). Note however that the resulting problem is in general nonsmooth and may require sophisticated techniques for its solution (see, e.g., [Lemaréchal, 1989]) (but classical techniques often perform satisfactorily). In situations where the nominal model is not LTI, however, it is typically necessary to resort to heuristics, e.g., replace the model set Π with a finite set and split each specification into a set of specifications, one for each model in this finite set.

Finally, note that the general nonlinear programming approach is not limited to the case of LTI models, but can be used also in the general case of nonlinear, time-varying

models.

5.3 A general approach for LTI models

The ultimate approach, in the case of LTI models, may be a combination of convex optimization techniques with the Youla parametrization and local techniques. The following scheme can be contemplated. First, ignore any restriction on controller structure and other nonconvex constraints (or approximate them with convex constraints). Globally solve the simplified problem using efficient convex optimization techniques. Next, use controller order reduction techniques to obtain a suboptimal controller of required order. Finally, use this suboptimal controller as an “initial guess” for the full optimization problem to be solved using local techniques.

6 Open-loop Optimal Control

An entire arsenal of techniques have been developed for problems of **open-loop optimal control** of very general systems, namely, problems of the following type. Given the dynamics

$$\dot{x}(t) = f(x(t), u(t)), \quad t \in [0, T], \quad (43)$$

where $T > 0$ is given, and given an initial state

$$x(0) = x_0, \quad (44)$$

determine a function $u(\cdot)$ over the interval $[0, T]$ such that a certain objective function depending on $u(\cdot)$ and on the corresponding state trajectory $x_u(\cdot)$ achieves its minimum (maximum) value, possibly subject to constraints on the control and state trajectories.

One may wonder however how useful such an open-loop control would be in practice. Indeed, the state is bound to be affected by external disturbances, so that any pre-computed control signal is unlikely to be anywhere close to optimal after even a short amount of time. There are however important classes of applications, in the context of the framework of Section 2, for which knowledge of an optimal (or suboptimal) open-loop control can be of great interest. We consider two such classes.

First suppose that the system to be controlled is highly nonlinear and that the effect of external disturbances is reasonably small compared to desired variations of the state (in particular, when the time horizon of interest is relatively short). A typical example here is that of a rendezvous between two spacecraft or that of a moon landing. The idea would then be as follows. First, compute an open-loop optimal (or suboptimal) control $u_0(\cdot)$ based on the nonlinear model, and compute the corresponding state trajectory $x_0(\cdot)$. Next, linearize the system dynamics around $u_0(\cdot)$ and $x_0(\cdot)$. Finally, design a controller for the linearized system, using, e.g., LMI techniques. The resulting controller is thus closed-loop and nonlinear (or affine-linear).

Alternatively, suppose that the system to be controlled has a long time horizon but evolves relatively slowly, as in, for example, a continuously operating chemical process. A promising technique for such systems is known as “model predictive control” or sometimes “receding horizon control” (see the articles “Internal Model Control” and “Predictive Control” in the handbook). The idea is, at discrete intervals of time t_i , to minimize the objective function of interest (possibly subject to constraints) for the time interval $[t_i, t_i + T]$ with the actual current state $x(t_i)$ as initial state, where T is a fixed quantity larger than $t_{i+1} - t_i$. Then the computed optimal control is applied until time t_{i+1} , at which point a new optimal control is computed. (It is assumed that the CPU time necessary to compute the optimal control is small compared to $t_{i+1} - t_i$.) This control scheme can be thought of as being intermediate between open-loop and closed-loop control schemes.

7 Multiple Objectives—Tradeoff Exploration

In the previous sections, we have considered problems of minimizing (or maximizing) one objective function possibly subject to constraints. Some cost functions considered were the \mathbf{H}_∞ (or, say, \mathbf{H}_2) gain from w to z or, say, some measure of the stability margin. (We also have encountered pure feasibility problems, not involving any objective functions.)

In practical design situations however, it is generally the case that several, often conflicting objective functions are to be jointly minimized/maximized, i.e., a (possibly con-

strained) **multiobjective** (or **multi-criterion**) **optimization problem** is to be solved (see for example [Maciejowski, 1989]). Singling out one of the specifications as an objective function, as we did in Section 3 for example, is often artificial. Obviously, in a strict sense, several objectives can seldom be jointly minimized (when one of them achieves its minimum, the others typically do not). Suppose ζ is the vector of optimization variables and ϕ_1, \dots, ϕ_n are the objective functions to be minimized. For simplicity of exposition, suppose that there are no constraints. To give a precise meaning to a multiobjective optimization problem, a “utility function” must then be defined, one that encapsulates ϕ_1 through ϕ_n . Two such functions are commonly used: weighted sum and weighted maximum, i.e.,

$$\phi_S(\zeta) = \sum w_i \phi_i(\zeta), \quad (45)$$

where $w_i > 0$, $i = 1, \dots, n$, and

$$\phi_M(\zeta) = \max \frac{\phi_i(\zeta)}{c_i}, \quad (46)$$

where $c_i > 0$, $i = 1, \dots, n$. It may also be appropriate to replace each ϕ_i in (45) and (46) with $\phi_i(\zeta) - \hat{\phi}_i$, where $\hat{\phi}_i$ is a “goal” or “good value⁴” associated with ϕ_i . If (45) is used, a standard “smooth” single objective problem is obtained. In contrast ϕ_M is generally nondifferentiable (it has “corners” at values of ζ where the “max” is achieved by more than one function). This nonsmoothness is benign, however, and can be removed by means of a simple trick: include an additional, artificial scalar optimization variable ζ^0 , and solve

$$\text{minimize } \zeta^0 \text{ with respect to } (\zeta, \zeta^0) \quad (47)$$

$$\text{subject to } \frac{\phi_i(\zeta)}{c_i} \leq \zeta^0 \quad i = 1, \dots, n, \quad (48)$$

which is a smooth constrained problem. Note however that the transformation just outlined hides the structure of the minimax problem. For this reason, numerical methods specifically designed for the latter are generally preferable (e.g., [Han, 1981; Polak, 1996; Zhou and Tits, 1996]).

⁴ c_i can also be thought of as the difference $\phi_i^b - \phi_i^g$ between a “bad” value to be avoided and a “good value” to aspire to; see [Nye and Tits, 1986].

Whether utility function (45) or utility function (46) is selected, appropriate values of the weights are seldom obvious. Rather, it is typical that, after a trial solution has been obtained, the designer would feel the need to modify the relative emphases on the various objective functions. This is the process of tradeoff exploration, best performed in a congenial graphical environment in which the graphical output conveys to the designer information on the current design, on which specifications are competing, and on how much improvement can be expected with respect to selected specifications if others are relaxed. Moreover, the graphical input tools must allow the designer to painlessly convey his/her inclinations to the optimization process. One example of such an environment is described in [Nye and Tits, 1986].

8 Defining Terms

Numerical optimization-based design: Design with the goal of optimizing measures of performance, using iterative optimization methods implemented on a computer.

Youla Parametrization: An affine parametrization of the set of achievable stable closed-loop transfer functions for a control system with a linear time-invariant plant and a linear time-invariant controller.

Convex optimization: The solution of an optimization problem requiring the minimization of a convex function subject to convex constraints on the optimization variables.

Linear Matrix Inequality: A matrix inequality of the form

$$F(\zeta) \triangleq F_0 + \sum_{i=1}^m \zeta_i F_i > 0, \quad (49)$$

where $\zeta \in \mathbf{R}^m$ is the variable, and $F_i = F_i^T \in \mathbf{R}^{n \times n}$, $i = 0, \dots, m$ are given. The inequality symbol in (49) means that $F(\zeta)$ is positive-definite, i.e., $u^T F(\zeta) u > 0$ for all nonzero $u \in \mathbf{R}^n$.

Nonlinear Programming: The solution of an optimization problem requiring the minimization (or maximization) of a nonlinear objective subject to nonlinear constraints on the optimization variables. The term “nonlinear” helps to contrast the methods with Linear Programming methods.

Branch and bound algorithm: An exhaustive search method for finding the maximum (or minimum) of a function over a rectangular region of parameters. The algorithm proceeds by systematically breaking up the parameter region into sub-rectangles (“branching”), computing bounds on the optimal objective over these sub-rectangles (“bounding”), and thereby computing bounds on the optimal objective over the original rectangle.

Robustness: The ability of a control system (a closed-loop system consisting of a plant and controller) to continue performing satisfactorily despite variations in the plant dynamics.

Multiobjective or multi-criterion optimization problem: An optimization problem where conflicting objective functions are to be jointly minimized/maximized, subject to constraints.

Open-loop optimal control: For the system

$$\dot{x}(t) = f(x(t), u(t)), \quad t \in [0, T], \quad (50)$$

where $T > 0$ is given, and given an initial state

$$x(0) = x_0, \quad (51)$$

the problem of determining a function $u(\cdot)$ over the interval $[0, T]$ such that a certain objective function depending on $u(\cdot)$ and on the corresponding state trajectory $x_u(\cdot)$ achieves its minimum (maximum) value, possibly subject to constraints on the control and state trajectories.

9 References

S. Boyd and C. Barratt. 1991. *Linear Controller Design: Limits of Performance*. Prentice-Hall, Englewood Cliffs, NJ.

V. Balakrishnan and S. Boyd. 1992. Global optimization in control system analysis and design. In *Control and Dynamic Systems: Advances in Theory and Applications*, volume 53, ed. C. T. Leondes, p. 1–55. Academic Press, New York, New York.

- S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. 1994. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA.
- J. Dunn. 1993. Gradient-related constrained minimization algorithms in function spaces: Convergence properties and computational implications. In *Large Scale Optimization: State of the Art*, ed. W. W. Hager, D. W. Hearn and P. M. Pardalos, p. 99–118. Kluwer Academic Publishers, Dordrecht.
- L. El Ghaoui and V. Balakrishnan. 1994. Synthesis of fixed-structure controllers via numerical optimization. In *Proc. IEEE Conf. on Decision and Control*, p. 2678–2683. Orlando, Florida.
- L. El Ghaoui, F. Delebecque, and R. Nikoukhah. 1995. *LMITool: A User-friendly Interface for LMI Optimization*. ENSTA/INRIA, Paris, France.
- P. Gahinet and A. Nemirovskii. 1993. *LMI Lab: A Package for Manipulating and Solving LMIs*. INRIA, Paris, France.
- P. E. Gill, W. Murray, and M. Wright. 1981. *Practical Optimization*. Academic Press, London.
- S. P. Han. 1981. Variable metric methods for minimizing a class of nondifferentiable functions. *Mathematical Programming*. 20:1–13.
- R. Hettich and K. O. Kortanek. 1993. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*. 35(3):380–429.
- J-B. Hiriart-Urruty and C. Lemaréchal. 1993. *Convex Analysis and Minimization Algorithms I & II*, volume 305, 306 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin.
- C. Lemaréchal. 1989. Nondifferentiable optimization. In *Optimization*, ed. G. Nemhauser, A. Rinnooy-Kan and M. Todd. *Handbooks in Operations Research and Management Science*. North Holland.

- J. M. Maciejowski. 1989. *Multivariable Feedback Design*. Addison-Wesley, Wokingham, England.
- D. Q. Mayne and E. Polak. 1975. First-order strong variation algorithms for optimal control. *Journal of Optimization Theory and Applications*. 16(3/4):277–301.
- D. Q. Mayne and E. Polak. 1993. Optimization based design and control of constrained dynamic systems. In *Proceedings of 12th IFAC World Congress on Automatic Control*, p. 129–138. Sydney, Australia.
- J. J. Moré and S. J. Wright. 1993. *Optimization Software Guide*, volume 14 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, PA.
- Yu. Nesterov and A. Nemirovskii. 1994. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA.
- W. T. Nye and A. L. Tits. 1986. An application-oriented, optimization-based methodology for interactive design of engineering systems. *Int. J. Control*. 43(6):1693–1721.
- E. Polak. 1996. *Computational Methods in Optimization: A Unified Approach*, 2nd ed. Academic Press, New York.
- E. Polak and D. Q. Mayne. 1975. First-order strong variation algorithms for optimal control problems with terminal inequality constraints. *Journal of Optimization Theory and Applications*. 16(3/4):303–325.
- J. L. Zhou and A. L. Tits. 1996. An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions. *SIAM J. on Optimization*. 6(2).

10 For Further Information

The details of control system design using a combination of Youla parametrization and convex optimization can be found in the book *Linear Controller Design: Limits of Performance* by

Boyd and Barratt. This book also discusses the framework for controller design that we presented in Section 2.

Control system analysis and design using Linear Matrix Inequalities is discussed in great detail in the book *Linear Matrix Inequalities in System and Control Theory*, by Boyd et al. This book also contains an extensive bibliography.

Local nonlinear programming methods are discussed, e.g., in *Practical Optimization* by Gill et al.. The book *Optimization Software Guide*, by Moré and Wright, has useful pointers to software. The survey paper “Optimization based design and control of constrained dynamic systems” by Mayne and Polak gives a quick idea of the versatility of local optimization methods, in particular semi-infinite optimization methods, in their application to controller design.

A major chapter of the book *Computational Methods in Optimization: A Unified Approach*, by Polak, is devoted to algorithms for open-loop optimal control. There the focus is on the extension to function spaces of algorithms initially developed for finite dimensional optimization problems. The review paper by Dunn also addresses this issue. Algorithms that make use of optimal control theory, namely of Pontryagin’s Maximum Principle, are discussed, e.g., in two 1975 papers by Mayne and Polak.