

Sensitivity of Computational Control Problems*

Nicholas J. Higham*

Mihail Konstantinov[†]

Volker Mehrmann[‡]

Petko Petkov[§]

March 7, 2003

Abstract

It is well-known that many factors contribute to the accurate and efficient numerical solution of mathematical problems such as those arising in computational control system design. In simple terms these are the arithmetic of the machine on which the calculations are carried out, sensitivity (or conditioning) of the mathematical model to small changes of the data and the numerical stability of the algorithms. It happens quite often that these concepts are confused. We define these concepts and demonstrate some of the subtleties that often lead to confusion. In particular we demonstrate with several examples what may happen when a problem is modularized, i.e., split into subproblems for which computational modules are available.

For three classical problems in computational control, pole placement, linear quadratic control and optimal H_∞ control, we then discuss the conditioning of the problems and point out sources of difficulties. We give some ill-conditioned examples for which even numerically stable methods fail.

We also stress the need for condition and error estimators that supplement the numerical algorithm and inform the user about potential or actual difficulties, and we explain what can be done to avoid these difficulties.

Keywords Sensitivity and conditioning, numerical stability, machine arithmetic, pole placement, linear quadratic control, algebraic Riccati equation, H_∞ control.

1 Introduction

For many centuries numerical methods have been used to solve various problems in science and engineering, but the importance of numerical methods grew tremendously with the advent of digital computers. It became clear immediately that many of the classical analytic and numerical methods and algorithms could not be implemented directly as computer codes although they were perfectly suited for hand computations. What was the reason? When

*Numerical Analysis Report 424, Manchester Centre for Computational Mathematics, 2003 and Preprint 2003/5, Institut für Mathematik, TU Berlin, 2003. To appear in IEEE Control Systems Magazine.

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (higham@ma.man.ac.uk, <http://www.ma.man.ac.uk/~higham/>). This work was supported by Engineering and Physical Sciences Research Council grant GR/R22612.

[†]University of Architecture and Civil Engineering, 1 Hr. Smirnenski Blvd., 1046 Sofia, Bulgaria (mmk_fte@uacg.bg).

[‡]Institut für Mathematik, MA 4-5, TU Berlin, D-10623 Berlin, Germany (mehrmann@math.tu-berlin.de). Supported by *Deutsche Forschungsgemeinschaft*, through DFG Research Center FZT86, 'Mathematics for key technologies' in Berlin.

[§]Department of Automatics, Technical University of Sofia, 1756 Sofia, Bulgaria (php@tu-sofia.acad.bg).

doing computations “by hand” a person can choose the accuracy of each elementary calculation and estimate—based on intuition and experience—its influence on the final result. On the contrary, when computations are done automatically such an error control is usually not possible and the effect of errors in the intermediate calculations must be estimated in a more formal way. Due to this observation, starting essentially with the work of J. Von Neumann, modern numerical analysis evolved as a fundamental basis of computer computations. One of the central themes of this analysis is to study the solution of computational problems in finite precision (or machine) arithmetic taking into account the properties of both the mathematical problem and the numerical algorithm for its solution. On the basis of such an analysis numerical methods may be evaluated and compared with respect to the accuracy that can be achieved.

When solving a computational problem on a digital computer, the accuracy of the computed solution generally depends on three major factors:

1. The properties of the *machine arithmetic*—in particular, the rounding unit (or the relative machine precision) and the range of this arithmetic.
2. The properties of the computational problem—in particular, *the sensitivity* of its solution relative to changes in the data, often estimated by the *conditioning* of the problem.
3. The properties of the computational algorithm—in particular, the *numerical stability* of this algorithm.

It should be noted that only by taking into account all three factors we are able to estimate the accuracy of the computed solution.

In this article we will discuss the sensitivity of three important problems of linear control theory that are solved very frequently in a large number of applications everyday. These are pole placement, linear quadratic optimal control and optimal H_∞ control. Let us briefly recall these problems.

Consider a linear constant coefficient dynamical system in state space form

$$\dot{x} = Ax + Bu, \quad x(t_0) = x^0, \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state at the time t , x^0 is an initial vector, $u(t) \in \mathbb{R}^m$ is the control input of the system and the matrices $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$ are constant. The classical *pole placement problem* is to find a state feedback control law

$$u = Kx \quad (2)$$

such that the closed loop system

$$\dot{x} = (A + BK)x \quad (3)$$

has prescribed desired poles or, in linear algebra terminology, that the spectrum of the closed loop system matrix $A + BK$ is a given collection \mathcal{P} of complex numbers symmetric about the real axis.

For a discussion of the theory of the pole placement problem and related problems, we refer the reader to classical monographs in linear control theory, e.g., [28, 64]. In Section 3 we will discuss the conditioning of the pole placement problem. This topic has generated some controversy in the literature and we will bring different viewpoints together.

Another important basic problem in control is the *linear quadratic control problem*. For this problem the objective is to find a control $u(t)$ such that the closed loop system is asymptotically stable and such that the performance index

$$\mathcal{S}(x, u) = \int_{t_0}^{\infty} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \quad (4)$$

is minimized. Here $Q = Q^T \in \mathbb{R}^{n,n}$, $R = R^T \in \mathbb{R}^{m,m}$ is positive definite and $\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}$ is positive semidefinite. An important feature of this problem is that the optimal control can be realized as a linear state feedback $u(t) = Kx(t)$. The classical theory for this problem can be found in the monographs [5, 11, 15, 28, 38, 40, 50, 54]. We discuss this problem in Section 4 and we will show, in particular, that the classical approach of using Riccati equations is sometimes not the best way to solve this problem.

The third problem that we include into our discussion is the *optimal H_∞ control problem*, which arises in the context of robust control in frequency domain, see the recent monographs [21, 57, 65]. In this problem one studies the linear system

$$\begin{aligned} \dot{x} &= Ax + B_1u + B_2w, & x(t_0) &= x^0, \\ z &= C_1x + D_{11}u + D_{12}w, \\ y &= C_2x + D_{21}u + D_{22}w, \end{aligned} \tag{5}$$

where $A \in \mathbb{R}^{n,n}$, $B_k \in \mathbb{R}^{n,m_k}$, $C_k \in \mathbb{R}^{p_k,n}$ for $k = 1, 2$, and $D_{ij} \in \mathbb{R}^{p_i,m_j}$ for $i, j = 1, 2$. Here $w(t) \in \mathbb{R}^{m_2}$ describes noise, modelling errors or an unknown part of the system, $y(t) \in \mathbb{R}^{p_2}$ describes measured outputs, while $z \in \mathbb{R}^{p_1}$ describes the regulated outputs. The objective of optimal H_∞ control is to find a controller

$$\begin{aligned} \dot{q} &= \tilde{A}q + \tilde{B}y, \\ u &= \tilde{C}q + \tilde{D}y, \end{aligned} \tag{6}$$

that internally stabilizes the system and minimizes the closed loop transfer function T_{zw} from w to z in H_∞ -norm.

Although this problem is frequently solved in practice, the sensitivity analysis and the development of reliable numerical methods for this problem is far from mature. We will point out some of the questions that need to be studied.

2 Basic concepts of numerical analysis

In order to refresh the memory of the reader we discuss in this section the three factors that determine the accuracy of the results of a numerical computation in more detail. Readers familiar with floating point arithmetic, conditioning and stability may jump to Section 3.

2.1 Floating point arithmetic

In this subsection we recall some of the basics of floating point arithmetic. A digital computer has only a finite number of internal states and hence it can operate with a finite, although possibly very large, set of numbers called *machine numbers*. As a result we have the so called *machine arithmetic*, which consists of the set of machine numbers together with the rules for performing algebraic operations on these numbers.

There are different machine arithmetics, the most widely used being the ANSI/IEEE 754-1985 Standard for Binary Floating Point Arithmetic [1], [26, Chap. 2], [47].

In the following we will not deal with a particular machine arithmetic but rather consider several issues which are essential in every computing environment. For a detailed treatment of this topic see [26, 62]. For simplicity we consider a real arithmetic.

Let $\mathbb{M} \subset \mathbb{R}$ be the set of machine numbers, where \mathbb{R} is the set of real numbers. The set \mathbb{M} is finite, contains the zero 0 and is symmetric about 0, i.e., if $x \in \mathbb{M}$ then $-x \in \mathbb{M}$.

Let \bullet be one of the four arithmetic operations (summation $+$, subtraction $-$, multiplication \times and division $/$). Thus for each two operands $x, y \in \mathbb{R}$ we have the result $x \bullet y \in \mathbb{R}$, where

$y \neq 0$ if \bullet is the division. For each operation $\bullet \in \{+, -, \times, /\}$ there is a *machine analogue* \odot of \bullet which, given $x, y \in \mathbb{M}$, yields

$$x \odot y \in \mathbb{M}.$$

But it is possible that the operation \odot cannot be performed in \mathbb{M} . We also note that even if $x, y \in \mathbb{M}$, then the number $x \bullet y$ may not be a machine number and hence $x \odot y \neq x \bullet y$.

As a result some strange things happen in \mathbb{M} :

- an arithmetic operation \odot may not be performed even if the operands are from \mathbb{M} ;
- the associative law is violated in the sense that $(x \odot y) \odot z \neq x \odot (y \odot z)$, where \bullet stands for $+$ or \times ;
- the distributive law may be violated in the sense that $(x \oplus y) \otimes z \neq x \otimes z \oplus y \otimes z$.

In order to map $x \in \mathbb{R}$ into \mathbb{M} , *rounding* is used to represent x by the number $\hat{x} \in \mathbb{M}$ (denoted also as $\text{rd}(x)$) which is closest to x , with some rule to break ties when x is equidistant from two machine numbers [47]. Of course, $\hat{x} = x$ if and only if $x \in \mathbb{M}$. We shall use the hat notation to denote also other quantities computed in machine arithmetic.

Since \mathbb{M} is finite, there is a very large positive number $L \in \mathbb{M}$ such that any $x \in \mathbb{R}$ can be represented in \mathbb{M} if and only if $|x| \leq L$. Moreover, there is a very small positive number $l \in \mathbb{M}$ such that if $|x| < l$ then $\hat{x} = 0$ even when $x \neq 0$. We say that a number $x \in \mathbb{R}$ is in the *standard range* of \mathbb{M} if $l \leq |x| \leq L$. In the IEEE double precision arithmetic we have

$$L \approx 10^{308}, \quad l \approx 10^{-324}, \quad (7)$$

where asymmetry is a consequence of the use of subnormal numbers [1], [26, Chap. 2], [47].

If a number x with $|x| > L$ appears as an initial data or as an intermediate result in a computational procedure realized in \mathbb{M} , then the computations are usually terminated. This is called an *overflow* and must be avoided. If a number $x \neq 0$ with $|x| < l$ appears during the computations then it is rounded to $\hat{x} = 0$ and this phenomenon is known as *underflow*. Although not so destructive as overflow, underflow should also be avoided. Over- and underflow may be avoided by appropriate *scaling* of the data as the next example suggests. It should be noted that often numerical problems occur because the data are represented in units that are widely differing in magnitude.

Example 1 Consider the computation of the norm $y = \|x\| = \sqrt{x_1^2 + x_2^2}$ of the vector $x = [x_1, x_2]^T$, where the data x_1, x_2 and the result y are in the standard range $[l, L]$ of \mathbb{M} . In particular, we have $l \leq |x_i| \leq L$. If, however, we have $x_1^2 > L$ then the direct calculation of y will give overflow. Another difficulty arises when $x_1^2, x_2^2 < l$. Then we have the underflow $\text{rd}(x_1^2) = \text{rd}(x_2^2) = 0$ resulting in the wrong answer $\hat{y} = 0$, while the correct answer is $y \geq l\sqrt{2}$. Overflow may be avoided by using the scaling $\xi_i := x_i/s$, $s := |x_1| + |x_2|$ (provided $s \leq L$) and computing the result from $y = s\sqrt{\xi_1^2 + \xi_2^2}$. Underflow can also be avoided by this scaling (we shall have at least $\hat{y} \geq l$ when $x_1^2, x_2^2 < l$).

Another important characteristic of \mathbb{M} is the *rounding unit* (*relative machine precision*, or *machine epsilon*), denoted by ε , which is half the distance from 1 to the next larger floating point number. If x is in the standard range of \mathbb{M} then the relative error in the approximation of x by its machine analogue \hat{x} satisfies the important bound

$$\frac{|x - \hat{x}|}{|x|} \leq \varepsilon. \quad (8)$$

In IEEE double precision arithmetic we have

$$\varepsilon \approx 1.1 \times 10^{-16}. \quad (9)$$

This means that rounding is performed with a tiny relative error. Most machine arithmetics, including IEEE arithmetics, are built to satisfy the following property.

Let $x, y \in \mathbb{R}$ and $x \bullet y \in \mathbb{R}$ be in the standard range of \mathbb{M} . Then the result of the machine analogue \odot of \bullet satisfies

$$x \odot y = (x \bullet y)(1 + \delta), \quad (10)$$

where $|\delta| \leq \varepsilon$.

If this property holds, then arithmetic operations on two numbers are performed very accurately in \mathbb{M} , with a relative error of order of the rounding unit ε .

Example 2 In IEEE double precision arithmetic the associative rule for summation and multiplication may be violated as follows. The computation of $1+10^{17}-10^{17}$ as $1 \oplus (10^{17} \ominus 10^{17})$ gives the correct answer 1, while the result of the machine summation $(1 \oplus 10^{17}) \ominus 10^{17}$ is 0. In turn, the computation of $10^{155}10^{155}10^{-250}$ as $10^{155} \otimes (10^{155} \otimes 10^{-250})$ will produce an accurate approximation to the correct answer 10^{60} , while the attempt to compute $(10^{155} \otimes 10^{155}) \otimes 10^{-250}$ will give an overflow.

One of the most dangerous operations during the computations is the subtraction of numbers that are very close to each other. In this case a *subtractive cancellation* may occur.

Example 3 Consider $x = 0.1234567892$ and $y = 0.1234567891$, which agree in their first 9 significant digits. The result of the subtraction is $z = x - y = 0.0000000001$. If the original x and y are subject to errors, likely to be at least of order 10^{-10} , then the subtraction has brought these errors into prominence and z may be dominated by error. This phenomenon has nothing to do with the errors of the machine subtraction (if any), except for the fact that the inaccuracies in x and y may be due to rounding errors at previous steps.

Example 4 If we compute

$$y = \frac{(1+x)^2 - (2x+1)}{x^2}, \quad x \neq 0,$$

in the ANSI/IEEE 754-1985 Standard for $x = 10^{-1}, \dots, 10^{-p}, \dots$, we see that for $p > 8$ the computed result \hat{y} is far from the correct answer $y = 1$; even negative values for \hat{y} will appear.

The reason for this phenomenon is that we have cancellation in the numerator for small x . For $p > 8$ the result of the subtraction is of order $\varepsilon \approx 10^{-16}$ and there is no correct digit in the computed numerator. To worsen the situation, this wrong result is divided by a denominator that is less than 10^{-16} . This example is very instructive because the input of order 10^{-8} and an output 1 are by no means close to the boundaries of the range $[10^{-324}, 10^{308}]$ of \mathbb{M} .

Often subtractive cancellation can be avoided by a simple reformulation of the problem.

Example 5 The expression $\sqrt{1+x} - 1$ may be computed as $x/(1 + \sqrt{1+x})$ to avoid cancellation for small x .

2.2 The computational problem

The second important feature in assessing the results of computations in finite arithmetic is the formulation of the computational problem. Most problems can be written in explicit form as $y = f(x)$ or in implicit form via the equation $\varphi(x, y) = 0$. Here typically the *data* x and the *result* y are elements of vector spaces X and Y , respectively, and $f : X \rightarrow Y$, $\varphi : X \times Y \rightarrow Y$ are given functions.

Suppose that the data x are perturbed to $x + \delta x$, where the perturbation may result from measurement, modelling or rounding errors. Then the result y is changed to $y + \delta y$, where

$$\delta y = f(x + \delta x) - f(x).$$

Thus δy depends on both the data x and its perturbation δx .

The estimation of some quantitative measure $\mu(\delta y)$ of the size of δy as a function of the corresponding measure $\mu(\delta x)$ of δx is the aim of *perturbation analysis* of computational problems. If $x = [x_1, \dots, x_m]^T$ and $y = [y_1, \dots, y_n]^T$ are vectors, then we may use some vector norm, $\mu(x) = \|x\|$ as such quantitative measure.

In order illustrate the idea of perturbation analysis we consider the solution of Lyapunov equations, which is another basic problem in computational control.

Example 6 Consider the Lyapunov equation

$$A^T X + X A = C, \tag{11}$$

where the coefficients A, C and the solution X are real 6×6 matrices. For a particular example of this equation we generated 10000 perturbations $\delta c_{11}, \delta c_{12}, \delta c_{22}$ in the corresponding entries of the right hand side C each of size $10^{-6} \times \|C\|_F$ and computed the variations $\delta x_{11}, \delta x_{12}, \delta x_{22}$ in the entries of the solution X . In Figure 1 we show the perturbations in the right hand side (the small sphere), the corresponding perturbations in the solution (the ellipsoid) and an appropriate sensitivity estimate (the large sphere). The sensitivity estimate should be an upper bound on the size of perturbations in the solution and in this case it is in the form of the linear estimate

$$\|\delta X\|_F \leq \beta \|\delta C\|_F$$

for some positive constant β .

Clearly, for some directions the corresponding perturbations lead to relatively small changes in the solution which makes the sensitivity estimate pessimistic for these particular perturbations.

To derive sensitivity estimates, we need some basic mathematical concepts. Recall that a function $f : X \rightarrow Y$ is *Lipschitz continuous* at a point $x \in X$ if there is $r_0 > 0$ such that for every $r \in (0, r_0]$ there exists a quantity M such that

$$\|f(x + h) - f(x)\| \leq M \|h\| \quad \text{for} \quad \|h\| \leq r.$$

The smallest such quantity

$$M = M(x, r) := \inf \left\{ \frac{\|f(x + h) - f(x)\|}{\|h\|} : h \neq 0, \|h\| \leq r \right\} \tag{12}$$

is called the *Lipschitz constant* of f in the r -neighborhood of x . Lipschitz continuous functions satisfy the perturbation bound

$$\|\delta y\| \leq M(x, r) \|\delta x\| \quad \text{for} \quad \|\delta x\| \leq r.$$

Computational problems $y = f(x)$, where f is Lipschitz continuous at x , are called *regular* at x . Otherwise they are called *singular*.

Perturbations in the solution of a Lyapunov equation

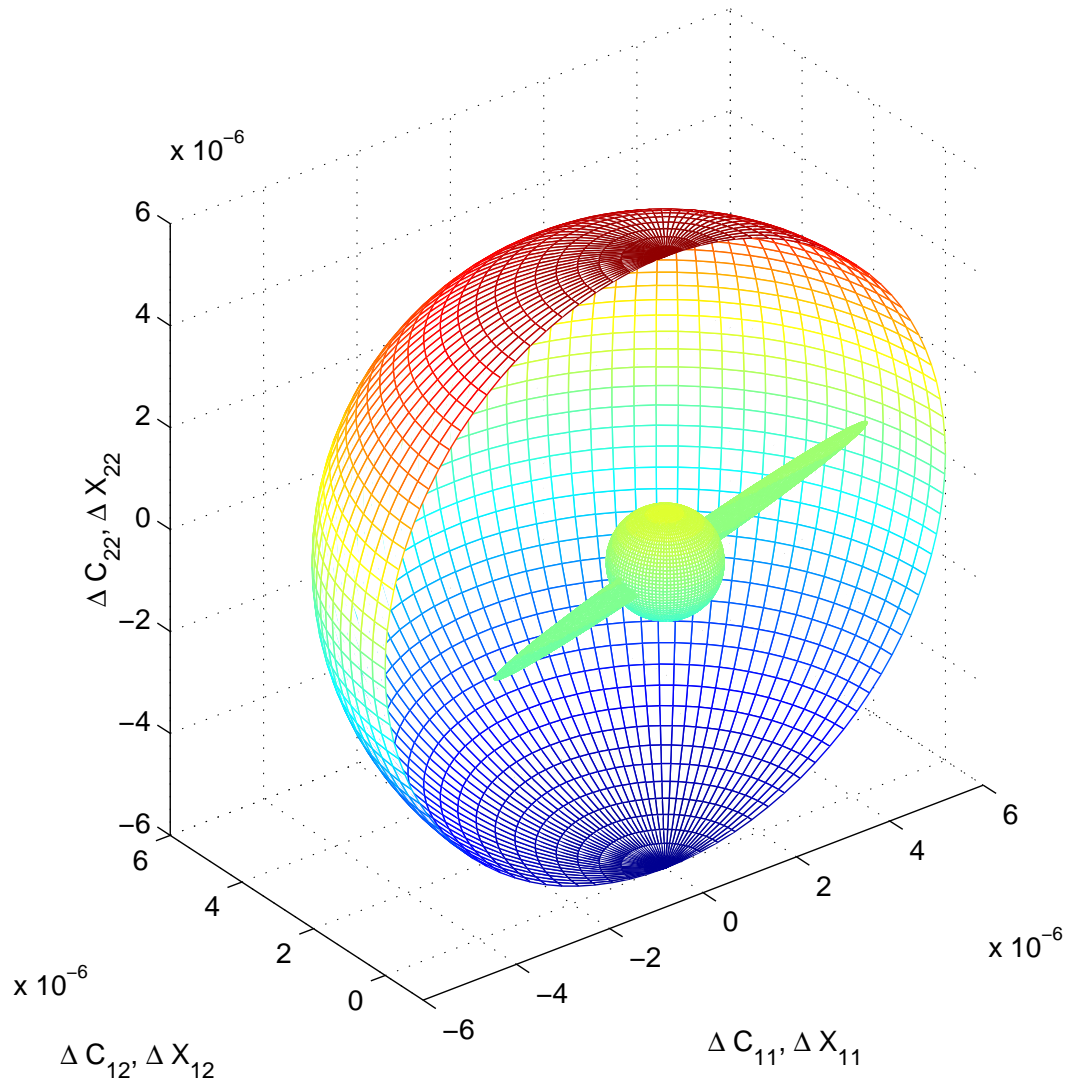


Figure 1: Perturbed solutions of Lyapunov equation and a sensitivity estimate

Example 7 Consider the polynomial equation

$$(y - 1)^p = y^p - py^{p-1} + \dots + (-1)^p = 0$$

which has a multiple solution $y = 1$. If the constant term $(-1)^p$ is changed to $(-1)^p - 10^{-p}$, then the perturbed equation will have p different roots $y_i = 1 + 0.1\varepsilon_i$, $i = 1, \dots, p$, where $\varepsilon_1, \dots, \varepsilon_p$ are the primitive p -roots of 1. Thus a relative change of 10^{-p} in one of the coefficients leads to a relative change of 0.1 in the solution.

In order to characterize when a problem has the property that small changes in the data lead to large changes in the result we introduce the concept of condition number. For a regular problem, let $M(x, r)$ be as in (12). Then the number

$$K(x) := \lim_{r \rightarrow 0} M(r, x)$$

is called the *absolute condition number* of the computational problem $y = f(x)$. For singular problems we set $K(x) = \infty$.

We have

$$\|\delta y\| \leq K(x)\|\delta x\| + \Omega(\delta x), \quad (13)$$

where the scalar quantity $\Omega(h)$ satisfies $|\Omega(h)|/\|h\| \rightarrow 0$ for $h \rightarrow 0$.

If $f = [f_1, \dots, f_n]^T$ is differentiable in the sense that all partial derivatives $(\partial f_i / \partial x_j)(x)$ exist and are continuous then

$$K(x) = \left\| \frac{\partial f}{\partial x}(x) \right\|,$$

where $\partial f / \partial x$ is the matrix with elements $\partial f_i / \partial x_j$ and $\|\cdot\|$ denotes both a vector norm and the corresponding subordinate matrix norm.

Suppose now that $x \neq 0$ and $y = f(x) \neq 0$. Then setting $\delta_x := \|\delta x\|/\|x\|$, $\delta_y := \|\delta y\|/\|y\|$ we have the bound

$$\delta_y \leq k(x)\delta_x + \omega(\delta x), \quad \omega(h) := \Omega(h)/\|y\|,$$

where $\|\omega(h)\|/\|h\| \rightarrow 0$ for $h \rightarrow 0$ and

$$k(x) := K(x) \frac{\|x\|}{\|y\|}$$

is the *relative condition number* of the problem $y = f(x)$.

Condition numbers can be defined analogously for implicit problems, defined via the equation $\varphi(x, y) = 0$, where x is the data and y is the solution.

A regular problem $y = f(x)$ is called *well-conditioned* (respectively *ill-conditioned*) if the relative condition number $k(x)$ is small (respectively large) in the context of the given machine arithmetic. In particular the problem is very well conditioned if $k(x)$ is of order 1, and very ill conditioned if $\varepsilon k(x) \simeq 1$.

The computer solution of ill-conditioned problems typically leads to large errors. In practice, the following **rule of thumb** may be used for the computational problem $y = f(x)$.

Suppose that $\varepsilon k(x) < 1$. Then one can expect approximately $-\log_{10}(\varepsilon k(x))$ correct decimal digits in the computed solution.

Indeed, as a result of rounding the data x we work with $\hat{x} = x + \delta x$, where $\|\delta x\| \leq \varepsilon\|x\|$. Even if no additional errors are made during the computation, then the computed result is $\hat{y} = f(\hat{x})$ and we have

$$\|f(\hat{x}) - f(x)\| \leq K(x)\|\delta x\| + \Omega(x) \leq \varepsilon K(x)\|x\| + \Omega(x).$$

Thus the relative error in the computed result satisfies the approximate inequality

$$\frac{\|\hat{y} - y\|}{\|y\|} \leq \frac{\varepsilon K(x)\|x\|}{\|y\|} = \varepsilon k(x). \quad (14)$$

Closely related to the sensitivity is the problem of estimating the *distance to the nearest singular problem*. Consider a computational problem $y = f(x)$. We call the quantity

$$\text{Dist}(f, x) = \min\{\|h\| : \text{the problem } y = f(x + h) \text{ is singular}\},$$

the *absolute distance* to singularity of the problem $y = f(x)$. Similarly, for $x \neq 0$, the quantity $\text{Dist}(f, x)/\|x\|$ is called the *relative distance* to singularity of the problem. Typically the relative distance to singularity and the relative condition number of the problem are inversely proportional, see [16].

Example 8 The problem of solving the linear system $Ay = b$ with a square matrix A and data $x = (A, b)$ is regular if and only if the matrix A is nonsingular. The relative distance to singularity for an invertible matrix A is $1/\text{cond}(A)$, where $\text{cond}(A) := \|A\| \|A^{-1}\|$ is the relative condition number of A relative to inversion [26, Thm. 6.5].

Another difficulty that needs to be mentioned is the mathematical representation of the computational problem that one needs to solve. In particular in control theory there are several different frameworks that are used. A classical example for such different frameworks is the representation of linear systems via matrices and vectors, as in the classical state space form (1), as rational matrix functions (via the Laplace transform) or even in a polynomial setting [18, 52]. These different approaches have different mathematical properties and often it is a matter of taste which framework is preferred.

From a numerical point of view, however, this is typically not a matter of taste, since the sensitivity is drastically different. Numerical analysts usually prefer the matrix/vector setting over the representations via polynomial or rational functions, while for users of computer algebra systems the polynomial or rational approach is often more attractive. The reason for the preference for the matrix/vector approach in numerical methods is that the sensitivity of the polynomial or rational representation is usually much higher than that of a matrix/vector representation and this fact is often ignored in order to favor frameworks which are mathematically more elegant but numerically inadequate. The reason for the higher sensitivity is often an over-condensation of the data, i.e. a representation of the problem with as few data as possible. Let us demonstrate this issue with a well-known example.

Example 9 [63] Consider the computation of the eigenvalues of the matrix

$$A = Q^T \text{diag}(1, 2, \dots, 20)Q,$$

where Q is a random orthogonal matrix. Clearly the matrix is symmetric and therefore diagonalizable with nicely separated eigenvalues $1, 2, \dots, 20$. The problem of computing the eigenvalues of A is very well-conditioned and numerical methods such as the symmetric QR algorithm lead to highly accurate results, see [20, 48]. For example `eig` from MATLAB [39] yields all eigenvalues to at least 15 correct digits.

The usual textbook approach to compute eigenvalues that is taught in first year linear algebra is that the eigenvalues of A are the roots of the characteristic polynomial

$$\det(\lambda I - A) = (\lambda - 1)(\lambda - 2) \cdots (\lambda - 20).$$

Using a numerical method such as `roots` from MATLAB to compute the roots of this polynomial, however, yields highly inaccurate large eigenvalues 20.0003, 18.9970, 18.0117, 16.9695,

16.0508, 14.9319, 14.0683, 12.9471, 12.0345, 10.9836, 10.0062, 8.9983, 8.0003. The accuracy for the small eigenvalues is slightly better. There are several reasons for the inaccuracy. First the coefficients of the polynomial range in the interval $[1, 20!]$ and cannot all be represented accurately in the IEEE double precision arithmetic, while the elements of the matrix range in the ball of radius 20 around the origin. Second, the sensitivity of the larger roots with respect to perturbations in the coefficients is very large in this case.

In this section we have discussed the sensitivity (conditioning in particular) of a computational problem. This is a property of the problem and its mathematical representation in the context of the machine arithmetic used, and should not be confused with the properties of the computational method that is implemented to solve the problem. In practice, linear sensitivity estimates of the type $\delta_y \leq k(x)\delta_x$ are usually used. This may sometimes lead to underestimation of the actual perturbation in the solution. Rigorous perturbation bounds can be derived by using the technique of non-linear perturbation analysis [34].

2.3 Computational algorithms

In this subsection we discuss properties of computational algorithms and the accuracy of the computed result.

An algorithm to compute $y = f(x)$ is a decomposition

$$f = F_r \circ F_{r-1} \circ \cdots \circ F_1 \quad (15)$$

which gives a sequence $x_k = F_k(x_{k-1})$, $k = 1, \dots, r$, with $x_0 = x$ and $y = x_r$. Usually the computation of $F_k(\xi)$ requires simple algebraic operations on ξ such as arithmetic operations or taking roots, but it may also be a more complicated subproblem like solving a system of linear equations or computing the eigenvalues of a matrix.

The algorithm either gives the exact answer in exact arithmetic or for some problems, like eigenvalue problems or the solution of differential equations, the answer is an approximation to the exact answer in exact arithmetic. We will not analyze the latter case here but we will investigate only what happens with the computed value of x_r when the computations are done in machine arithmetic.

It is important to mention that two different algorithms, say (15) and $f = \Phi_s \circ \Phi_{s-1} \circ \cdots \circ \Phi_1$ for computing $y = f(x)$ may give completely different results in machine arithmetic although in exact arithmetic they are equivalent.

In what follows we suppose that the data x is in the standard range of the machine arithmetic with characteristics L, l, ε , and that the computations do not lead to overflow or to a destructive underflow. As a result the answer computed by the algorithm (15) is \hat{y} . Our goal is to estimate the absolute error $E := \|\hat{y} - y\|$ and the relative error $e := E/\|y\|$ (for $y \neq 0$) of the computed solution \hat{y} in case of a regular problem $y = f(x)$ when the data x belongs to a given set X_0 .

Definition 1 *The algorithm (15) is numerically stable on the set X_0 if the computed quantity \hat{y} for $y = f(x)$, $x \in X_0$, is close to the solution $f(\hat{x})$ of a problem with data \hat{x} near to x in the sense that*

$$\|\hat{y} - f(\hat{x})\| \leq \varepsilon a \|y\|, \quad \|\hat{x} - x\| \leq \varepsilon b \|x\|, \quad (16)$$

where the constants $a, b > 0$ do not depend on $x \in X_0$.

If the computed value \hat{y} is equal to the rounded value $\text{rd}(y)$ of the exact answer y , then the result is obtained with the *maximum achievable accuracy*.

For a problem for which the data is inexact, perhaps itself being subject to rounding errors, numerical stability is in general the most we can ask of an algorithm. If in Definition 1 we take $a = 0$, then the algorithm is called *numerically backward stable*.

The concept of backward stability, introduced by Wilkinson [62], tries to represent the error of a computational algorithm by showing that the computed solution is the exact solution of a perturbed problem, where the perturbation is called the *equivalent data error*. The elementary floating point operations are carried out in a backward stable way, because (10) shows that the computed answer is the correct one for perturbed data $x(1 + \delta_1)$ and $y(1 + \delta_2)$, where δ_i are of order ε . It is clear from the definitions that backward stability implies stability, but the converse is not true.

Example 10 The computational problem $y = f(x) := 1 + 1/x$, is solved directly and very accurately for $x \geq 1$. However, for $x > 1/\varepsilon$ we have $\hat{y} = 1$ (a very accurate result!), but the method is not backward stable, since $f(x) > 1$ for all $x > 0$.

As in [50], using the inequalities (16) and $\|f(x+h) - f(x)\| \leq K(x)\|h\| + \Omega(h)$ (see (13)), we obtain the estimate for the absolute error

$$\begin{aligned} E := \|\hat{y} - y\| &= \|\hat{y} - f(\hat{x}) + f(\hat{x}) - f(x)\| \\ &\leq \|\hat{y} - f(\hat{x})\| + \|f(\hat{x}) - f(x)\| \\ &\leq \varepsilon a \|y\| + K(x)\|\hat{x} - x\| + \Omega(\hat{x} - x) \\ &\leq \varepsilon a \|y\| + \varepsilon b K(x)\|x\| + \Omega(\hat{x} - x). \end{aligned}$$

Dividing by $\|y\|$ we get an estimate for the relative error

$$e := \frac{\|\hat{y} - y\|}{\|y\|} \leq \varepsilon \left(a + bK(x) \frac{\|x\|}{\|y\|} + \frac{\omega(\hat{x} - x)}{\varepsilon} \right).$$

Since $\omega(\hat{x} - x)/\varepsilon \rightarrow 0$ for $\varepsilon \rightarrow 0$, by ignoring this term we have the approximate estimate

$$e \leq \varepsilon \left(a + bK(x) \frac{\|x\|}{\|y\|} \right) = \varepsilon(a + bk(x)) \quad (17)$$

for the relative error in the computed solution.

Inequality (17) shows clearly the influence of all the three major factors that determine the accuracy of the computed solution:

- the machine arithmetic (the rounding unit ε and implicitly the range of \mathbb{M} through the requirement to avoid over- and underflow);
- the computational problem (the relative condition number $k(x)$);
- the computational algorithm (the constants a and b).

Inequality (17) is an example of a *condition number based accuracy estimate* for the solution, computed in a machine arithmetic. In order to assess the accuracy of results and to be able to trust numerical results, such condition and accuracy estimates should accompany every computational procedure. Many modern software packages provide such estimates. It is, however, unfortunately common practice in industrial use to turn these facilities off, even though this service will warn the user of numerical methods about possible failure.

As we have seen in (15), computational problems are typically modularized, i.e., they are decomposed and solved by a sequence of subproblems. This facilitates the use of computational modules and is one of the reasons for success of numerical analysis. But one should be

aware that this modularization may lead to substantial numerical difficulties. This happens if one or more of the created subproblems F_i that are combined to get the decomposition is ill-conditioned. Consider for simplicity the decomposition $f = F_2 \circ F_1$ of a regular problem $y = f(x)$ with absolute condition number $K = K(x)$ which is of order 1. Thus the original problem is very well conditioned.

First of all it may happen that one of the subproblems F_k is not regular.

Example 11 The scalar identity function $y = f(x) = x$ may be decomposed as $f = F_2 \circ F_1$, where $F_1(x) = x^3$ and $F_2(z) = z^{1/3}$. Here the function F_2 is not Lipschitz continuous at 0.

But even if the functions F_1, F_2 are Lipschitz continuous with constants K_1, K_2 respectively, then it may happen that one (or both) of these constants is large. We obtain the estimate

$$\begin{aligned} \|f(x+h) - f(x)\| &= \|F_2(F_1(x+h)) - F_2(F_1(x))\| \\ &\leq K_2 \|F_1(x+h) - F_1(x)\| \leq K_2 K_1 \|h\|, \end{aligned}$$

where the quantity $K_2 K_1$ may be much larger than the actual Lipschitz constant K of f .

Example 12 Consider the identity function $y = f(x) = x$ in \mathbb{R}^2 . Define $F_1(x) = A^{-1}x$ and $F_2(z) = Az$, where the matrix $A \in \mathbb{R}^{2,2}$ is non-singular. Then $K = 1$ while both $K_1 = \|A^{-1}\|$ and $K_2 = \|A\|$ may be arbitrarily large.

If the computations are carried out with maximum achievable accuracy, then the computed value for $A^{-1}x$ is $\widehat{F}_1(x) = (I_2 + E_1)A^{-1}x$, where $E_1 := \text{diag}(\varepsilon_1, \varepsilon_2)$ and $|\varepsilon_1|, |\varepsilon_2| \leq \varepsilon$. Similarly, the computed value for $A(A^{-1}x)$ becomes $(I_2 + E_2)A\widehat{F}_1(x) = (I_2 + E_2)A(I_2 + E_1)A^{-1}x$, where $E_2 := \text{diag}(\varepsilon_3, \varepsilon_4)$ and $|\varepsilon_3|, |\varepsilon_4| \leq \varepsilon$. Suppose that $\varepsilon_1 = -\varepsilon_2 = \varepsilon \simeq 10^{-16}$, $\varepsilon_3 = \varepsilon_4 = 0$ and

$$A = \begin{bmatrix} a & a+1 \\ a-1 & a \end{bmatrix}, \quad x = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

where $a = 10^8$. Then the computed result is $\widehat{x} = x + \varepsilon\xi$, where $\xi = [\xi_1, \xi_2]^T$ and $\xi_1 = 4a^2 + 2a - 1$, $\xi_2 = 4a^2 - 2a - 1$. Thus the actual relative error in the solution of the decomposed problem is

$$\varepsilon \frac{\|\xi\|}{\|x\|} \simeq 4a^2 \varepsilon \simeq 4$$

and there are no correct digits in the computed result.

In this section we have reviewed some of the general principles of numerical analysis. In the following sections we look at some basic problems in control theory and analyze their sensitivity.

3 Pole-Placement

Pole-placement is a very important tool in many applications in modern control theory. As we have discussed in the introduction, in linear algebra terminology the pole placement problem is as follows.

Problem 1 For a given pair of matrices $S = (A, B)$ with $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$ and a given collection of n complex numbers $\mathcal{P} = \{\lambda_1, \dots, \lambda_n\} \subset \mathbb{C}$ (closed under conjugation), find a matrix $K \in \mathbb{R}^{m,n}$ such that the set of eigenvalues of $A + BK$ is equal to \mathcal{P} .

It is well-known, see e.g., [28, 64], that a *feedback gain* matrix K exists for all collections $\mathcal{P} \subset \mathbb{C}$, (symmetric relative to the real axis) if and only if (A, B) is *controllable*, i.e.,

$$\text{rank}[A - \lambda I_n, B] = n, \quad \forall \lambda \in \mathbb{C}. \quad (18)$$

There is a large literature on this problem, in particular on numerical methods for its solution [29, 46, 49, 60]. Even though numerical backward stability has been shown for some of these methods, see e.g. [4, 13, 14, 46, 49], it is often observed that the numerical results are very inaccurate. In view of our discussion in Section 2, if a numerically stable method yields highly inaccurate results, then this is most likely due to ill-conditioning of the problem. The analysis of the conditioning of the pole placement problem, however, led to interesting observations, see [3, 37, 41, 42]. We will discuss these issues in more detail, since they illuminate some of the misconceptions that arise in the evaluation of numerical methods.

Since controllability is a requirement for the ability to assign arbitrary sets of poles, from the discussion in Subsection 2.2 it must be expected that numerical difficulties arise when the problem is very near to an uncontrollable problem. The *distance to uncontrollability* is defined as the minimum of the quantity $\|[\delta A, \delta B]\|$, where the pair $(A + \delta A, B + \delta B)$ is uncontrollable, see [17]. This distance may be determined by computing $\min_{\lambda \in \mathcal{C}} \sigma_n[A - \lambda I, B]$, see [17], where $\sigma_n[A - \lambda I, B]$ denotes the smallest singular value of the matrix $[A - \lambda I, B]$.

As we have discussed in Subsection 2.3 numerical problems may also arise when a problem is approached via a multi-step procedure, where an intermediate step is ill-conditioned. In the case of pole-placement this is usually a two-step procedure which first brings the pair (A, B) to a simpler form [36, 50, 58, 59] and then assigns the poles in this simpler form [45, 60]. For the evaluation of a particular numerical method the conditioning of both subproblems needs to be analyzed and good results can only be expected if both subproblems are well-conditioned.

If one studies the large literature of the pole-placement problem this is only partially reflected and the discussion is quite controversial. This has several reasons, which have to do with the non-uniqueness of the solution in the multi-input case, but also with the representation of the data. Another reason for the resulting confusion in the evaluation of the pole placement problem is that one has to define clearly what the *solution* of the problem is. This could be the feedback K , the closed loop matrix $A + BK$ or its spectrum, respectively. All of these are solutions of the pole placement problem but they exhibit largely different perturbation behavior. We will now summarize these different viewpoints.

The perturbation analysis for the gain matrix consists in studying the change δK in the gain matrix K as a function of the changes $\delta A, \delta B$ in the system matrices A, B and the changes $\delta \lambda_1, \dots, \delta \lambda_n$ in the desired poles.

In this case, whether or not the closed-loop system matrix $A + BK$ or its spectrum is sensitive is not the subject of the sensitivity analysis. Of course, in the multi-input case it is possible to use the $n(m - 1)$ -parametric freedom in the gain matrix K to minimize some measure of the sensitivity of the eigenstructure of $A + BK$, or to achieve other design purposes, e.g., minimization of $\|K\|$, maximization of the stability radius of $A + BK$, see [29, 43, 61]. Since for $m > 1$ the general solution for the gain matrix K is an unbounded $n(m - 1)$ -dimensional algebraic variety in $\mathbb{R}^{m,n}$, the sensitivity analysis has to guarantee that there exists at least one solution to the perturbed problem for which the perturbation bounds hold. At the same time both the original and perturbed problems may have solutions of arbitrary large norm. Explicit perturbation bounds for K , both local and non-local, have been derived in [37].

Let $\Lambda := \text{diag}(\lambda_1, \dots, \lambda_n)$ and $\delta \Lambda := \text{diag}(\delta \lambda_1, \dots, \delta \lambda_n)$. An estimate in terms of relative perturbations $\delta_K := \|\delta K\|_F / \|K\|_F$ is given by

$$\delta_K \leq c_A \delta_A + c_B \delta_B + c_\Lambda \delta_\Lambda + O(\|\delta\|^2), \quad (19)$$

Table 1: Norms of feedback gain matrix and error in assigned spectrum.

m	\widehat{K}	err
2	2.5×10^6	2.0×10^1
3	1.3×10^6	1.2×10^1
4	2.3×10^5	1.2×10^{-3}
5	3.4×10^5	1.6×10^{-6}
6	1.0×10^4	3.1×10^{-8}
7	4.2×10^3	1.3×10^{-9}
8	2.1×10^3	1.3×10^{-10}
9	1.1×10^3	1.9×10^{-11}
10	8.9×10^2	6.3×10^{-12}

where $c_A = C_A \|A\|_F / \|K\|_F$, $c_B = C_B \|B\|_F / \|K\|_F$, $c_\Lambda = C_\Lambda \|\Lambda\|_F / \|K\|_F$ are the relative condition numbers in respect to the perturbations in A , B , Λ , respectively, $\delta_A = \|\delta A\|_F / \|A\|_F$, $\delta_B = \|\delta B\|_F / \|B\|_F$, $\delta_\Lambda = \|\delta \Lambda\|_F / \|\Lambda\|_F$, C_A , C_B , C_Λ are the corresponding absolute condition numbers and $\delta := [\delta_A, \delta_B, \delta_\Lambda]^T$.

This analysis shows that the problem of computing the feedback gain K is well- or ill-conditioned if the overall relative condition number

$$c_{PA} := c_A + c_B + c_\Lambda \quad (20)$$

is small or large in the context of the machine arithmetic used, see [37].

In general, the sensitivity of the computation of K does not depend substantially on the desired spectrum \mathcal{P} . At the same time the eigenstructure (the eigenvalues in particular) of the matrix $A + BK$ may be very sensitive to perturbations in the data. As a result the spectrum of the computed closed-loop system matrix $A + B\widehat{K}$ may be far from \mathcal{P} , even if \widehat{K} is computed by a numerically stable algorithm or even if K is exact. A striking example for this effect is the following.

Example 13 [41] Let $A = \text{diag}(1, \dots, 20)$, $\mathcal{P} = \{-1, \dots, -20\}$, let B be formed from the first m columns of a random 20×20 orthogonal matrix.

The MATLAB pole placement code `place` of the *Control System Toolbox* Version 4.1 which is an implementation of the method given in [29], was used to compute the feedback gain K . For m from 1 to 10 the feedback was computed 20 times with 20 random matrices B with orthogonal columns. In Table 1 the geometric means (over the 20 experiments) of the norm of the computed feedback matrix \widehat{K} and $\text{err} = \max_{1 \leq i \leq 20} |\widehat{\lambda}_i - \lambda_i|$ are listed, with λ_i and the real parts of the resulting poles $\widehat{\lambda}_i$ arranged in increasing order.

It should be noted that for all 400 tests the pair (A, B) was controllable with a large distance to uncontrollability. Nevertheless for $m = 1$ the method produced an error message ‘‘Can’t place eigenvalues there’’ and for $m = 2, 3$ a warning ‘‘Pole locations are more than 10% in error’’ was displayed. Other pole placement algorithms have similar difficulties for small m , see [41, 42]. The eigenvalues of the closed loop system are highly sensitive (and may have even negative real part) regardless how the feedback is computed (even if it is computed analytically).

The analysis of the sensitivity of the spectrum and the eigenvectors of the closed-loop matrix $A + BK$ has been carried out in [41, 42]. The major factors in the conditioning of the closed loop spectrum include the norm of K , the distance to uncontrollability and the condition number of the closed loop eigenvector matrix. We have the following possibilities.

- The gain matrix K is very sensitive, for example since the distance to uncontrollability is small. Then small equivalent changes in A, B may lead to a large difference between \widehat{K} and K . This difference will in general also result in large errors for the eigenvalues of the computed closed-loop system matrix.
- The norm of the gain matrix K is very large. Then the difference $\|\widehat{K} - K\|$, which is of order at least $\varepsilon\|K\|$, may also be large and this will affect the eigenvalues of $A + B\widehat{K}$.
- The eigenvalues of $A + BK$ are very sensitive to perturbations for any (or for the particular) choice of K . This, for example, will be the case in dead-beat control of discrete-time systems $x(t+1) = Ax(t) + Bu(t)$ when the closed loop poles are all equal to zero. Here the perturbations in the eigenvalues of $A + BK$ may be of order $\eta^{1/n}$, where η is the size of the perturbations in the data.

All these three reasons are independent and may appear alone or in some combination. Moreover, in some cases the minimum sensitivity of the pole assignment problem is achieved exactly when the eigenstructure of the closed-loop system matrix is maximally sensitive.

Example 14 Consider the pole placement problem for the simplest case $n = 2, m = 1$. Let

$$A = \begin{bmatrix} \lambda_1 & 1 \\ \beta & \lambda_2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

If the desired poles are λ_1, λ_2 then $K = [-\beta, 0]$ and we obtain

$$C_\Lambda = \sqrt{1 + 2\mu + \sqrt{1 + 4\mu^4}}, C_A = \sqrt{2 + 4\mu^2}, C_B := \mu + \sqrt{1 + \mu^2}$$

where $\mu := |\lambda_1 - \lambda_2|/2$. Here the minimum sensitivity of K is achieved for $\lambda_1 = \lambda_2$ which corresponds to maximum sensitivity of the closed-loop poles.

Example 15 [37] In this example we study the overall relative condition number c_{PA} in (20) for computing K for the controllable pair of matrices

$$A = \begin{bmatrix} 0 & 3 & 0 & 4 & 0 & -7 \\ -9 & 0 & -3 & 0 & 7 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ -4 & 0 & -1 & 0 & 4 & 0 \\ 0 & 3 & 0 & 4 & 0 & -7 \\ -9 & 0 & -2 & 0 & 8 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}.$$

We take $\lambda_3 = \dots = \lambda_6 = -1$ and vary λ_1, λ_2 . In Figure 2 we show the dependence of c_{PA} on the real and imaginary parts of λ_1, λ_2 . We see that the computation of K remains well conditioned for large variations in λ_1, λ_2 . The minimum of the overall conditioning is achieved for λ_1, λ_2 near to -1 . Choosing all desired poles equal to -1 we obtain the gain matrix

$$K = [6.3, 2.3, 0.7, 3.1, -5.3, -5.35]$$

and the relative condition numbers $c_\Lambda = 1.420, c_A = 37.27$ and $c_B = 2.360$.

In Figure 3 we show the distribution of the closed-loop poles (the so-called *pseudospectrum*) for 2000 perturbations in $A+BK$ of norm 10^{-8} , computed by the function `ps` from the Matrix Computation Toolbox [25]. Clearly, the large sensitivity of the closed-loop poles is not related to the conditioning of computing K .

Pole Assignment Conditioning

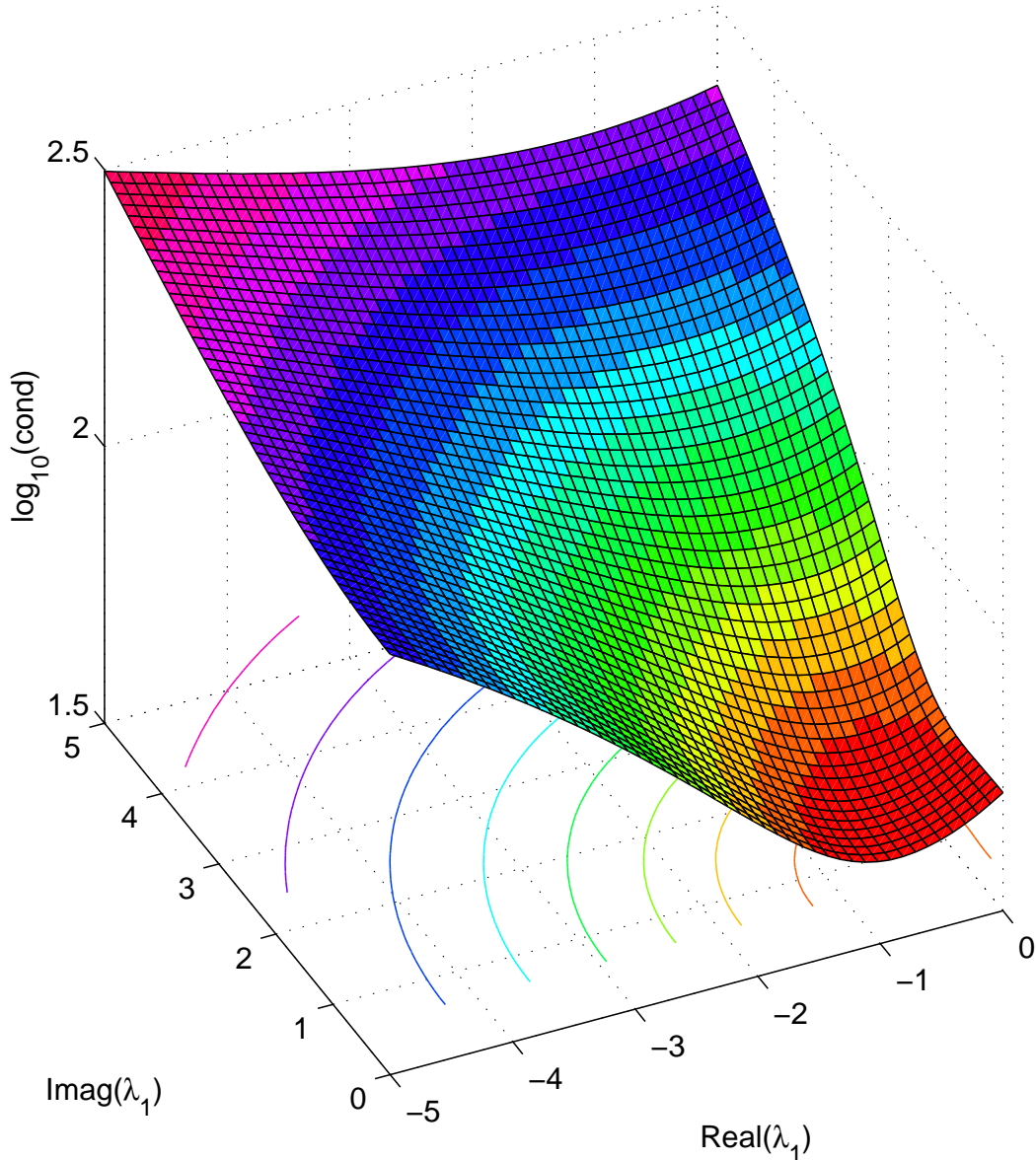


Figure 2: Pole assignment conditioning as a function of $\text{real}(\lambda_1)$, $\text{imag}(\lambda_1)$

So far we have only briefly discussed the use of the freedom in the choice of K in the multi-input case. There are several possibilities to use this freedom to optimize a robustness measure: one could minimize $\|K\|$, see [10, 30, 46, 51, 60], or the stability radius of $A + BK$, or the condition number of the closed loop eigenvector matrix as in [29] (in this case the poles must be pairwise disjoint) or the feedback norm and the eigenvalue sensitivity together [61]. In general one should also first ask the following question.

Does one really have a fixed collection of poles or does one rather have a specific region in the complex plane where one wants the closed loop poles to be?

If the latter is the case then not only the minimization over the freedom in K but also a minimization over the position of the poles in the given set should be used. This would lead to the *optimized pole placement problem* [43, 44].

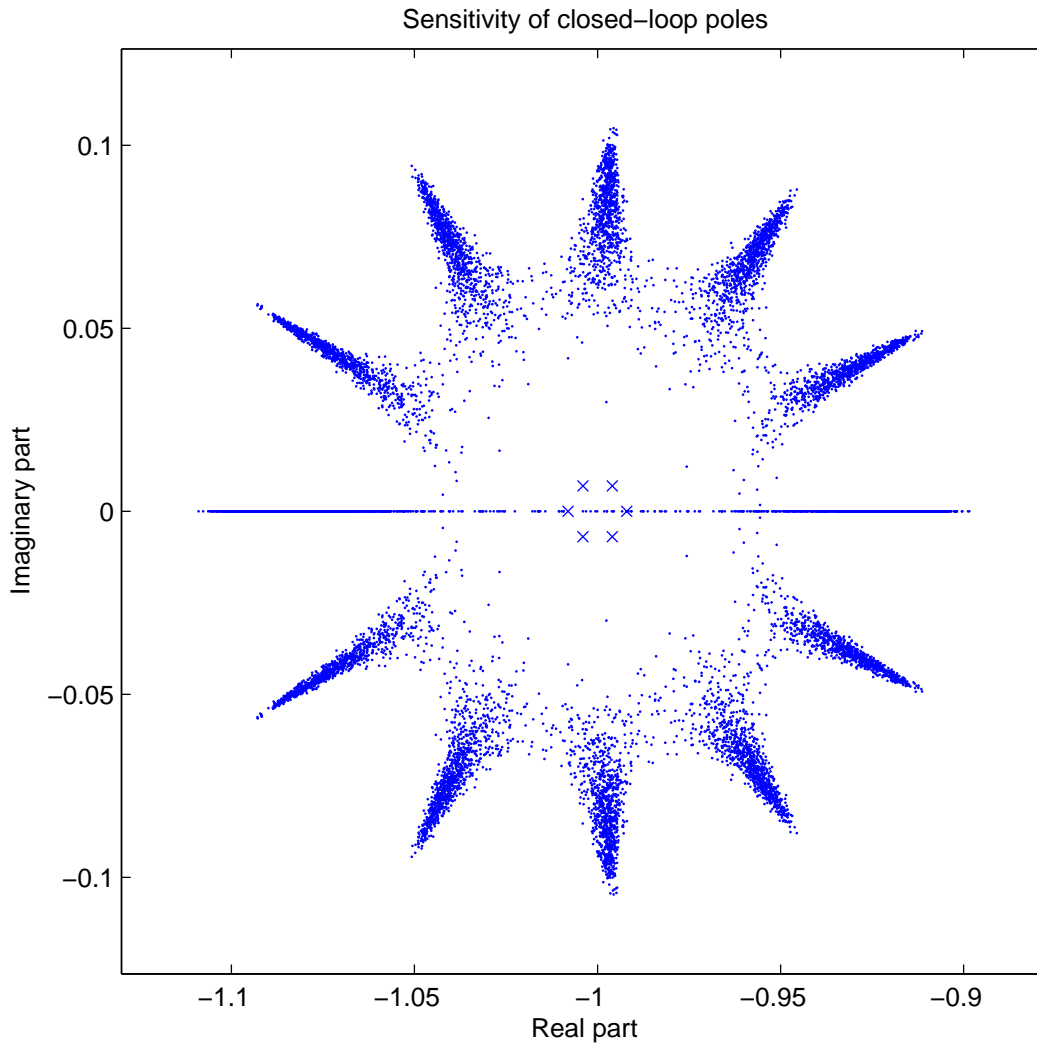


Figure 3: Sensitivity of closed-loop poles

Problem 2 For given matrices $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$ and a given set $\mathcal{P} \subset \mathbb{C}$, find a matrix $K \in \mathbb{R}^{m,n}$, such that the set of eigenvalues of $A+BK$ is contained in \mathcal{P} and at the same time a robustness measure is optimized.

A clear and practical formulation of such a general robust measure as well as suitable algorithms to determine this optimized pole assignment will depend on the application and on the set \mathcal{P} . In the stabilization problem this is the left half plane or in the case of damped stabilization a particular part of the left half plane, see [22]. If the set \mathcal{P} is too small, like when it has exactly n points, then even an optimization of some robustness measure may still yield a very sensitive closed loop spectrum, but if the set \mathcal{P} is large, then better results may be obtained. The general sensitivity analysis for this optimized pole placement problem is an open problem.

4 Linear quadratic control

In this section we discuss the linear quadratic control problem to minimize (4) subject to (1). Application of the maximum principle [40, 53] leads to the equivalent problem of finding an

asymptotically stable solution to the two-point boundary value problem of Euler-Lagrange equations

$$\mathcal{E}_c \begin{bmatrix} \dot{x} \\ \dot{\mu} \\ \dot{u} \end{bmatrix} = \mathcal{A}_c \begin{bmatrix} x \\ \mu \\ u \end{bmatrix}, \quad x(t_0) = x^0, \quad \lim_{t \rightarrow \infty} \mu(t) = 0, \quad (21)$$

with the matrix pencil

$$\alpha \mathcal{E}_c - \beta \mathcal{A}_c := \alpha \begin{bmatrix} I & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & 0 \end{bmatrix} - \beta \begin{bmatrix} A & 0 & B \\ Q & A^T & S \\ S^T & B^T & R \end{bmatrix}. \quad (22)$$

If R is well-conditioned with respect to inversion, then (21) may be reduced to the two-point boundary value problem

$$\begin{bmatrix} \dot{x} \\ -\dot{\mu} \end{bmatrix} = \mathcal{H} \begin{bmatrix} x \\ -\mu \end{bmatrix}, \quad x(t_0) = x^0, \quad \lim_{t \rightarrow \infty} \mu(t) = 0, \quad (23)$$

with the *Hamiltonian matrix*

$$\mathcal{H} = \begin{bmatrix} F & G \\ H & -F^T \end{bmatrix} := \begin{bmatrix} A - BR^{-1}S^T & BR^{-1}B^T \\ Q - SR^{-1}S^T & -(A - BR^{-1}S^T)^T \end{bmatrix}. \quad (24)$$

We again have different mathematical representations that may be used to compute the optimal control and these representations again observe very different sensitivity.

The classical way to solve the boundary value problems (21) and (23) [38, 40, 54], that is implemented in most design packages, is again a two step procedure. One computes first X , the positive semidefinite (stabilizing) solution of the associated algebraic Riccati equation

$$0 = H + XF + F^T X - XGX, \quad (25)$$

and then obtains the optimal stabilizing feedback as

$$u(t) = -R^{-1}B^T Xx(t). \quad (26)$$

Another approach is the deflating subspace approach of Van Dooren [59]. Suppose $(\mathcal{E}_c, \mathcal{A}_c)$ has an n -dimensional deflating subspace associated with eigenvalues in the left half plane. Let this subspace be spanned by the columns of a matrix \mathcal{U} , partitioned conformably with the pencil as

$$\mathcal{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}. \quad (27)$$

Then, if U_1 is invertible, the optimal control is a linear feedback of the form $u(t) = Kx(t) = U_3 U_1^{-1} x(t)$. The solution of the associated Riccati equation (25) is then $X = U_2 U_1^{-1}$, see [40] for details. In this case an explicit solution of the Riccati equation is not needed to determine the feedback.

In analogy to the discussion of the pole-placement problem we first discuss the question what is the distance to the nearest singular problem. The requirement that the closed loop system is asymptotically stable leads to the requirement that the system (1) is *stabilizable*, i.e.

$$\text{rank}[A - \lambda I_n, B] = n, \quad \forall \lambda \in \mathbb{C}_0^+, \quad (28)$$

where \mathbb{C}_0^+ denotes the closed right half plane. The *distance to unstabilizability* is defined as the minimum of the quantity $\|[\delta A, \delta B]\|$, where the pair $(A + \delta A, B + \delta B)$ is not stabilizable.

Table 2: Comparison of Riccati and subspace approach.

γ	Method	$\frac{\ \hat{X}-X\ _2}{\ X\ _2}$	$\frac{\ \hat{K}-K\ _2}{\ K\ _2}$
10^{-2}	care	7.0×10^{-16}	1.3×10^{-15}
	qz	2.4×10^{-16}	4.9×10^{-15}
10^{-6}	care	3.1×10^{-12}	3.2×10^{-9}
	qz	2.6×10^{-15}	4.7×10^{-11}
10^{-9}	care	2.1×10^{-8}	1.3×10^{-4}
	qz	1.6×10^{-15}	5.9×10^{-9}
10^{-13}	care	9.2×10^{-5}	3.9×10^1
	qz	1.7×10^{-15}	5.0×10^{-4}

This distance can be determined by studying the smallest perturbation so that the matrix pencil (22) ceases to have exactly n finite eigenvalues in the open left half of the complex plane and hence we have to discuss the perturbation theory of eigenvalues and invariant subspaces of matrix pencils. This is definitely beyond the scope of this paper, see [32, 55] for a detailed analysis of this problem.

It is clear that the three different approaches to determine the feedback gain K have different sensitivities. For example we see that in order to use the representation (23), the invertibility of R is required and thus it is clear that the sensitivity of the computation of $K = U_3 U_1^{-1} x(t)$ is different than that of the procedure to first compute $X = U_2 U_1^{-1}$ and then the feedback $K = -R^{-1} B^T X$ from this. Consider the following example.

Example 16 [44] Let U be a randomly generated real orthogonal matrix, $S = 0$, and let

$$A = U \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} U^T, \quad B = U, \quad R = \begin{bmatrix} 0.5 & 0 \\ 0 & \gamma \end{bmatrix}, \quad Q = U \begin{bmatrix} 6 & 0 \\ 0 & 3\gamma \end{bmatrix} U^T,$$

where $\gamma > 0$.

For the desired solution of the Riccati equation (25) and the associated feedback we have

$$X = U \begin{bmatrix} 3 & 0 \\ 0 & 3\gamma \end{bmatrix} U^T, \quad K = - \begin{bmatrix} 6 & 0 \\ 0 & 3 \end{bmatrix} U^T,$$

and the closed loop spectrum is $\{-4, -2\}$. Since K and the spectrum are independent of the value of γ and since U is orthogonal, we see that $\|K\|$ is small and hence we do not expect large perturbations in the solution. The solution via the Riccati equation, however, depends on γ .

In Table 2 we compare the MATLAB m-file **care** from the MATLAB Control Toolbox [39] which is a solver for algebraic Riccati equations and compare the results with those obtained by just computing the deflating subspace by the MATLAB implementation **qz** of the QZ-algorithm. The Riccati solution is used to compute $K = -R^{-1} B^T X$ while via the deflating subspace (27) of $\alpha \mathcal{E}_c - \beta \mathcal{A}_c$, the feedback K is directly obtained as $U_3 U_1^{-1}$. The relative error in X and K for the two methods and different values of γ are listed in Table 2.

We see that the direct computation of the optimal control via the computation of the invariant subspace (using **qz**) yields much smaller relative errors than the solution via the Riccati equation (using **care**).

As in the pole-placement problem we also have to ask the question what we are interested in as solution to the problem. This could be the feedback gain $K = -R^{-1} B^T X = U_3 U_1^{-1}$ or

the closed matrix $A + BK$ or its spectrum, Examples 13 and 14 (which can be constructed to come from optimal control) demonstrate that these may have very different sensitivity.

The discussion demonstrates again that it is important to analyze the sensitivity of the formulation of the computational problem and that a different modularization of the computational problem may lead to significantly different results. We see that the solution of the linear quadratic control problem via the solution of the algebraic Riccati equation presents a dangerous detour that may lead to very bad results. However, this detour is not necessary, since the feedback and the closed loop matrix may be computed from the deflating subspace. The situation is even worse in the case of descriptor systems, see [7, 40], where the Riccati equation may not have anything to do with the solution of the optimal control problem.

On the other hand, the approach via the Riccati equation is well analyzed and very efficient numerical software for the solution of algebraic Riccati equation is available, while the development of structure preserving solution methods for the eigenvalue problem (22) is not matured yet, [7]. For this reason we now discuss the conditioning of the algebraic Riccati equation (25). This then allows to decide when the conditioning of the Riccati equation is much worse than the conditioning of the optimization problem itself. For this reason we now discuss the conditioning of the algebraic Riccati equation (25), see [12, 31, 33, 56]. We assume that there exists a non-negative definite solution X such that $F - GX$ is stable.

Let the coefficient matrices F , G , H in (25) be subject to perturbations δF , δG , δH , respectively, so that instead of the initial data we have the matrices $\tilde{F} = F + \delta F$, $\tilde{G} = G + \delta G$, $\tilde{H} = H + \delta H$. The aim of the perturbation analysis of (25) is to investigate the variation δX in the solution $\tilde{X} = X + \delta X$ due to the perturbations δF , δG , δH . It is assumed here that the perturbations preserve the symmetric structure of the equation, i.e., the perturbations δG and δH are symmetric. If $\|\delta F\|$, $\|\delta G\|$ and $\|\delta H\|$ are sufficiently small, then the perturbed solution \tilde{X} is well defined [31]. The *condition number of the Riccati equation* (25) is defined as (see [12])

$$K_R = \limsup_{\alpha \rightarrow 0} \left\{ \frac{\|\delta X\|}{\alpha \|X\|} : \|\delta F\| \leq \alpha \|F\|, \|\delta G\| \leq \alpha \|G\|, \|\delta H\| \leq \alpha \|H\| \right\}.$$

For sufficiently small α we have (to first order)

$$\frac{\|\delta X\|}{\|X\|} \leq K_R \alpha.$$

Let \hat{X} be the solution of the Riccati equation computed by a numerical method in finite arithmetic with rounding unit ε . If the method is *backward stable*, then we can bound the relative error in the solution by

$$\frac{\|\hat{X} - X\|}{\|X\|} \leq p(n) K_R \varepsilon$$

with some low-order polynomial $p(n)$ of n . This shows the importance of the condition number in the accuracy estimation of the computed solution.

The determination of the exact condition number K_R is a difficult task. In first order approximation the equation for δX can be represented as

$$\delta X = -\Omega^{-1}(\delta H) - \Theta(\delta F) + \Pi(\delta G), \quad (29)$$

where

$$\begin{aligned} \Omega(Z) &= F_c^T Z + Z F_c, \\ \Theta(Z) &= \Omega^{-1}(Z^T X + X Z), \\ \Pi(Z) &= \Omega^{-1}(X Z X) \end{aligned}$$

are linear operators in the space of $n \times n$ matrices, which determine the sensitivity of X with respect to the perturbations in F , G , H , respectively, and $F_c = F - GX$. Based on (29) it was suggested in [12] to use the approximate condition number

$$K_B := \frac{\|\Omega^{-1}\| \|H\| + \|\Theta\| \|F\| + \|\Pi\| \|G\|}{\|X\|}, \quad (30)$$

where $\|\Omega^{-1}\|$, $\|\Theta\|$, $\|\Pi\|$ are the corresponding induced operator norms. Note that the quantity

$$\|\Omega^{-1}\|_F = \frac{1}{\text{sep}(F_c^T, -F_c)}$$

where

$$\text{sep}(F_c^T, -F_c) := \min_{Z \neq 0} \frac{\|F_c^T Z + Z F_c\|_F}{\|Z\|_F}$$

is connected to the sensitivity of the Lyapunov equation [23]

$$F_c^T X + X F_c = C.$$

In Figures 4 and 5 we show the relative variations $\|\delta X\|_F / \|X\|_F$ in the solutions of well-conditioned and ill-conditioned Riccati equations, respectively, for small relative perturbations in the matrices F and G . While, in the case of well-conditioned Riccati equations, the

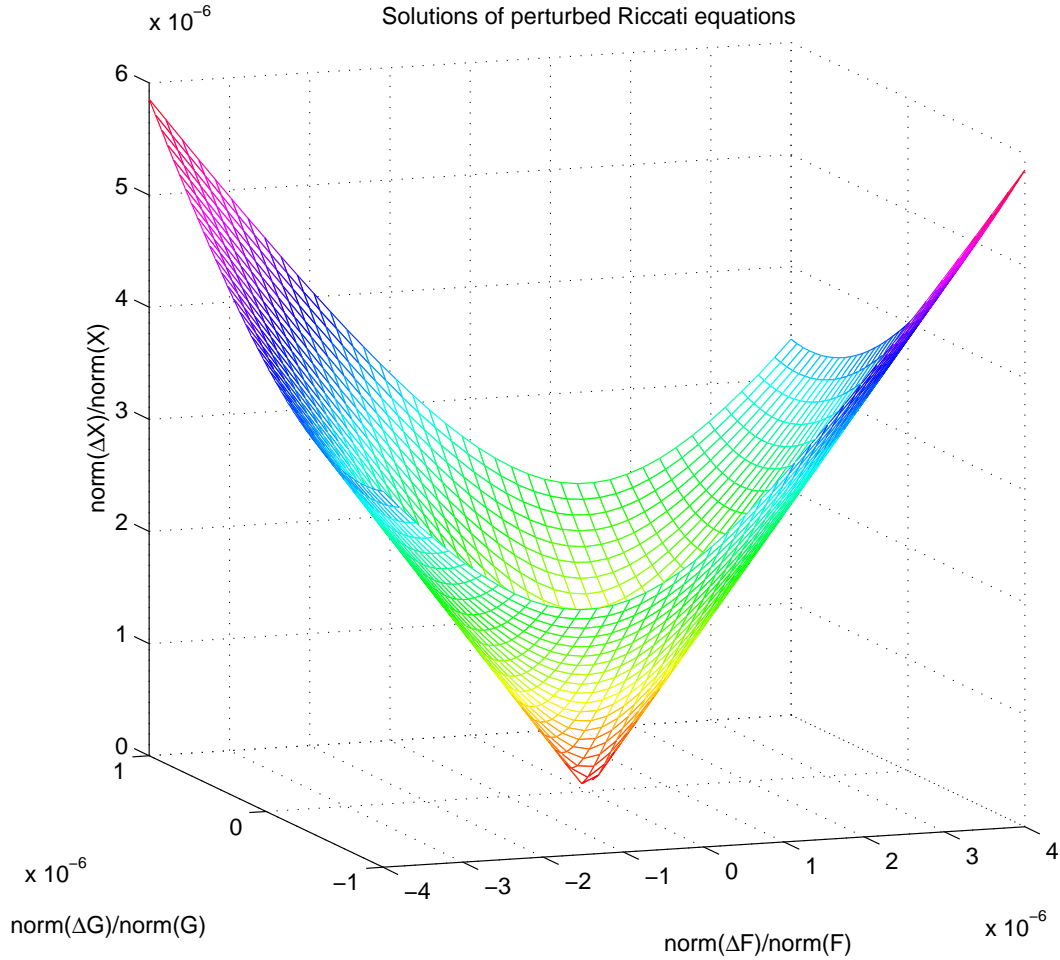


Figure 4: Perturbed solutions of well-conditioned Riccati equations

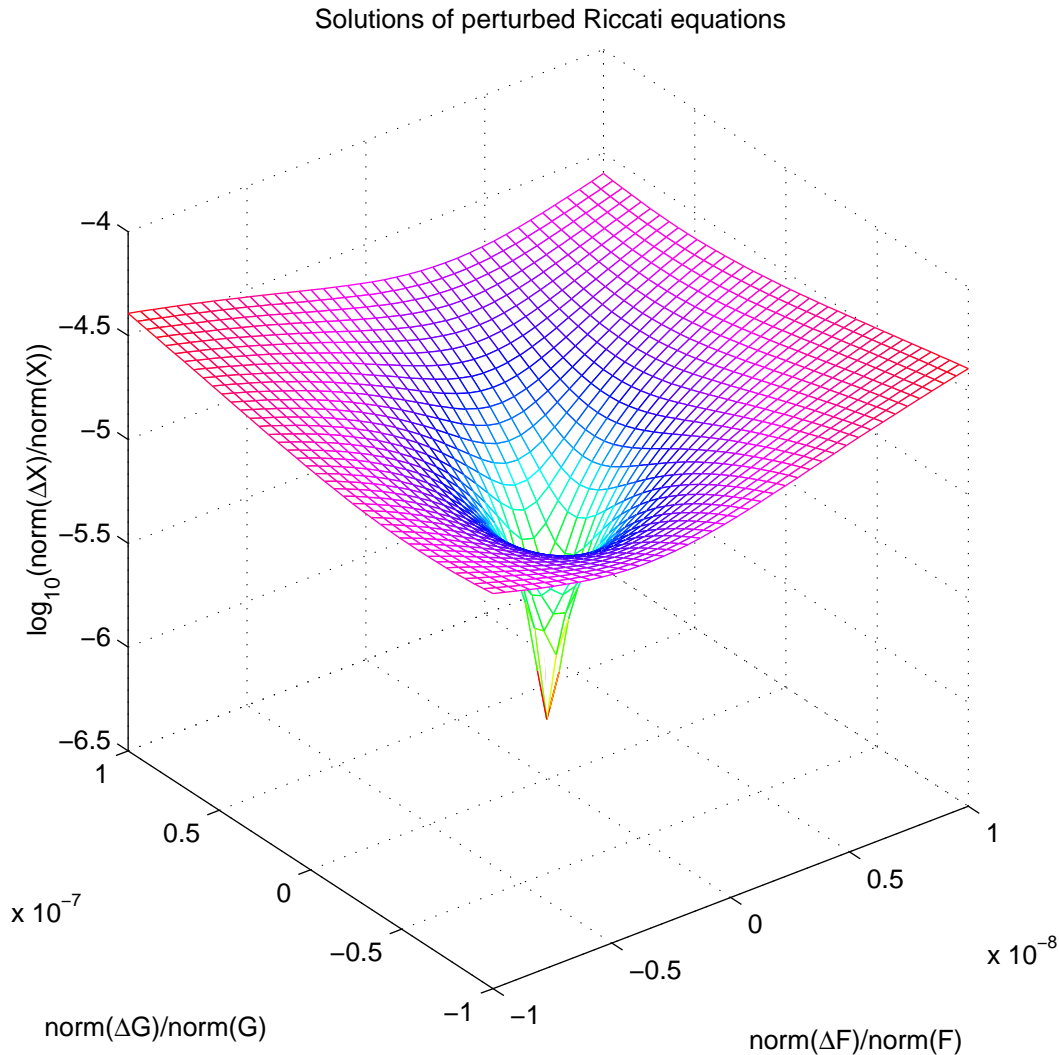


Figure 5: Perturbed solutions of ill-conditioned Riccati equations

change in the solution is of the order of the perturbations in the data, we see that in the case of ill-conditioned Riccati equations the change in the solution is 10000 times larger than the perturbations in the data.

An important practical issue is how to estimate cheaply the quantities in the condition number (30) and other condition numbers. This is now a routine matter thanks to the development of efficient matrix norm estimators, and in particular the LAPACK norm estimator `xLACON`, [2, 27]; see also [26, Chap. 15], that computes an estimate of the 1-norm $\|B\|_1$ given only the ability to evaluate matrix-vector products Bx and $B^T y$ for judiciously chosen x and y . The use of this estimator for condition estimation in nonsymmetric eigenproblems and matrix Sylvester equations was developed in [6] and [24], respectively. In the case of Riccati equations it is possible to take advantage of the solution symmetry, thus reducing significantly the cost of the estimation.

For the Riccati equation we may use the condition estimator to obtain

$$\|\Omega^{-1}\|_F = \frac{1}{\text{sep}_F(F_c^T, -F_c)}.$$

An estimate of $\|\Theta\|_1$ can be obtained in a similar way by solving the Lyapunov equations

$$\begin{aligned} F_c^T Y + Y F_c &= V^T X + X V \\ F_c Z + Z F_c^T &= V^T X + X V \end{aligned} \quad (31)$$

and to estimate $\|\Pi\|_1$ via `xLACON` it is necessary to solve the equations

$$\begin{aligned} F_c^T Y + Y F_c &= X V X \\ F_c Z + Z F_c^T &= X V X, \end{aligned} \quad (32)$$

where the matrix V is again symmetric.

As in the case of other condition estimators it is always possible to construct special examples where the value produced by `xLACON` underestimates the true value of the corresponding norm by an arbitrary factor. However, in practice severe underestimation happens only in rare circumstances. To demonstrate the performance of these estimators consider the following example.

Example 17 Consider a family of Riccati equations, constructed as

$$F = T F_0 T^{-1}, \quad G = T^{-T} G_0 T^{-1}, \quad H = T H_0 T^T,$$

where

$$F_0 = \text{diag}(F_1, F_1), \quad G_0 = \text{diag}(G_1, G_1), \quad H_0 = \text{diag}(H_1, H_1)$$

are diagonal matrices with

$$\begin{aligned} F_1 &= \text{diag}(-1 \times 10^{-k}, -2, -3 \times 10^k), \\ H_1 &= \text{diag}(3 \times 10^{-k}, 5, 7 \times 10^k), \\ G_1 &= \text{diag}(10^{-k}, 1, 10^k) \end{aligned}$$

and T is a nonsingular transformation matrix. The solution is then given by $X = T^{-T} X_0 T^{-1}$ where X_0 is a diagonal matrix whose elements are determined simply from the elements of F_0 , G_0 , H_0 . To avoid large rounding errors in constructing and inverting T , this matrix is chosen as $T = T_2 S T_1$ where T_1 and T_2 are elementary reflectors and S is a diagonal matrix,

$$\begin{aligned} T_1 &= I_n - 2e e^T / n, \quad e = [1, 1, \dots, 1]^T, \\ T_2 &= I_n - 2f f^T / n, \quad f = [1, -1, 1, \dots, (-1)^{n-1}]^T, \\ S &= \text{diag}(1, s, s^2, \dots, s^{n-1}), \quad s > 1. \end{aligned}$$

By varying the scalar s it is possible to control the condition number of the matrix T with respect to inversion, since $\text{cond}_2(T) = s^{n-1}$.

The solution is given by

$$X_0 = \text{diag}(X_1, X_1), \quad X_1 = \text{diag}(1, 1, 1).$$

The conditioning of these equations deteriorates with the increase of k and s .

In Figure 6 we show the ratio of the error in the solution and its estimate as functions of k and s . We see that for large k and s (i.e. for ill-conditioned equations) the error estimate may become pessimistic. This is due to the fact that the estimate is based on an error analysis that is inevitably pessimistic, and any poor estimate is then usually due not to the estimator but to the error bound. At the same time the numerical experiments show that generally the condition estimates are always of the same order as the true condition numbers.

Accuracy of the error estimate in the solution of Riccati equations

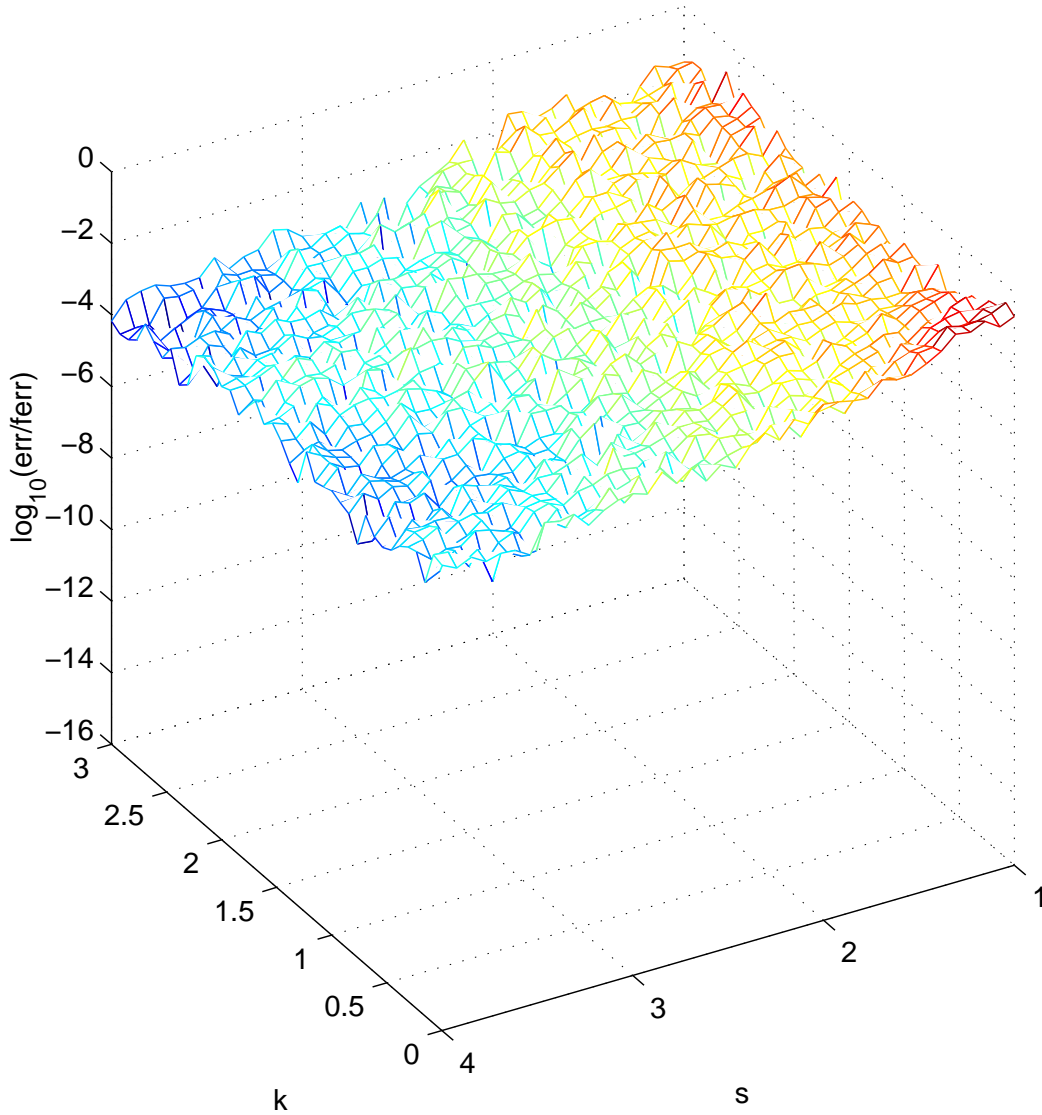


Figure 6: Accuracy of the error estimate for a family of Riccati equations

As in the pole-placement (where the choice of poles may represent an extra freedom), we may use the freedom in the choice of the weighting matrices Q, S, R to optimize other performance criteria to solve an *optimized linear quadratic control problem*.

Problem 3 [44] *Given matrices $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$ and a set $\mathcal{P} \subset \mathbb{C}$, determine cost matrices Q, S, R such the closed loop system obtained via the solution of the associated linear quadratic control problem has eigenvalues that are contained in \mathcal{P} and at the same time a robustness measure is optimized.*

In this section we have discussed the sensitivity of the linear quadratic optimal control problem and, in particular, the solution approach via the solution of algebraic Riccati equations. As we have demonstrated, the analysis is not complete.

5 H_∞ control

As final problem we consider the optimal H_∞ problem. Since, in general, it is difficult to compute the optimal controller, typically the following *modified optimal H_∞ problem* is solved. Instead of looking for the minimum of the transfer functions one determines the minimal parameter γ for which $\|T_{zw}\|_\infty < \gamma$. It should be noted that in general the optimal H_∞ norm of the transfer function is less than or equal to the minimal γ in the modified problem.

The advantage of the modified problem is, however, that it is a one-parameter optimization problem. Furthermore, under some extra assumptions, it is easy to classify when, for a given parameter $\gamma > 0$, a controller exists such that $\|T_{zw}\|_\infty < \gamma$. The computation of such an *admissible controller* is usually called the *suboptimal H_∞ problem*.

Consider the Assumptions:

A1 The pair (A, B_2) is *stabilizable* and the pair (A, C_2) is *detectible*.

A2 $D_{22} = 0$ and both D_{12} and D_{21} have full rank.

A3 The matrix $\begin{bmatrix} A-i\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all real ω .

A4 The matrix $\begin{bmatrix} A-i\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all real ω

and form the symmetric matrices

$$\begin{aligned} R_H(\gamma) &:= \begin{bmatrix} D_{11}^T \\ D_{12}^T \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} \end{bmatrix} - \begin{bmatrix} \gamma^2 I_{m_1} & 0 \\ 0 & 0 \end{bmatrix}, \\ R_J(\gamma) &:= \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix} \begin{bmatrix} D_{11}^T & D_{21}^T \end{bmatrix} - \begin{bmatrix} \gamma^2 I_{p_1} & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned} \quad (33)$$

Let, furthermore, γ_0 be the largest value of γ for which $R_H(\gamma)$ or $R_J(\gamma)$ is singular. Then the solvability of the suboptimal problem is classified via the following theorem.

Theorem 2 [65]. *Consider system (5), with R_H, R_J as in (33). Under assumptions A1–A4, there exists an internally stabilizing controller such that the transfer function from w to z satisfies $\|T_{zw}\|_\infty < \gamma$ if and only if the following four conditions hold.*

1. $\gamma > \gamma_0$.
2. *There exists a positive semidefinite solution X_H of the algebraic Riccati equation associated with the Hamiltonian matrix*

$$\begin{aligned} H(\gamma) &= \begin{bmatrix} A_H(\gamma) & G_H(\gamma) \\ H_H(\gamma) & -A_H^T(\gamma) \end{bmatrix} \\ &= \begin{bmatrix} A & 0 \\ -C_1^T C_1 & -A^T \end{bmatrix} - \begin{bmatrix} B_1 & B_2 \\ -C_1^T D_{11} & -C_1^T D_{12} \end{bmatrix} R_H^{-1}(\gamma) \begin{bmatrix} D_{11}^T C_1 & B_1^T \\ D_{12}^T C_1 & B_2^T \end{bmatrix}. \end{aligned} \quad (34)$$

3. *There exists a positive semidefinite solution X_J of the algebraic Riccati equation associated with the Hamiltonian matrix*

$$\begin{aligned} J(\gamma) &= \begin{bmatrix} A_J(\gamma) & G_J(\gamma) \\ H_J(\gamma) & -A_J^T(\gamma) \end{bmatrix} \\ &= \begin{bmatrix} A^T & 0 \\ -B_1 B_1^T & -A \end{bmatrix} - \begin{bmatrix} C_1^T & C_2^T \\ -B_1 D_{11}^T & -B_1 D_{21}^T \end{bmatrix} R_J^{-1}(\gamma) \begin{bmatrix} D_{11} B_1^T & C_1 \\ D_{21} B_1^T & C_2 \end{bmatrix}. \end{aligned} \quad (35)$$

4. $\gamma^2 > \rho(X_H X_J)$ (where $\rho(X_H X_J)$ is the spectral radius ρ of $X_H X_J$).

The optimal solution of the modified H_∞ control problem is then obtained by finding the smallest admissible γ so that conditions 1–4 in Theorem 2 still hold. In this way we have found a mathematical formulation of the problem that allows to compute the suboptimal controller.

As before, in order to assess the sensitivity, we must first decide which of the problems and in which mathematical formulation we wish to solve the problem. To analyze the sensitivity of the optimal H_∞ control problem is still an open problem and in general it is not clear how to compute this minimum. Also for the modified optimal H_∞ control problem the sensitivity is not completely understood but a lot of progress has been made in recent years, e.g. [8, 19].

We will not repeat the discussion of the previous sections but it should be clear by now that the sensitivity of different formulations may differ significantly, in this case we may even have completely different solutions in the optimal and suboptimal case. It is also obvious that many factors contribute to the distance of this problem to the nearest singular problem, including the distance to the nearest unstabilizable problem, see Section 4. But the situation is even more complicated here, since the method involves a nonlinear optimization procedure and hence the problem of computing the suboptimal controller may be singular or close to singular for different values of γ .

The part of the sensitivity analysis that is best analyzed [35] is that of the suboptimal H_∞ control problem, where for given matrices $A, B_1, B_2, C_1, C_2, D_{11}, D_{12}, D_{21}, D_{22}$ and for given $\gamma > \gamma_{\text{modopt}}$ the sensitivity of the resulting controller (6) under perturbations $\delta A, \delta B_1, \dots, \delta D_{22}, \delta \gamma$ in the data is studied. We do not present these formulas here, but it should be obvious that the conditioning of the two Riccati equations for X_H and X_J as well as the distance to singularity of the matrices R_H, R_J plays a major role. One of the major difficulties that frequently arises is the ill-conditioning of one or both Riccati equations near the suboptimal γ . In Figure 7 we show the conditioning of the Riccati equations involving X_H and X_J for a sixth order system. With γ going to $\gamma_0 = 10.1806399112943$ the sensitivity of the second Riccati equation is tending to infinity.

As a consequence of this problem most optimization methods will not be able to determine the suboptimal controller.

There are many more numerical difficulties in the computation of the optimal or suboptimal H_∞ controller. To present these is beyond the scope of this paper and work in progress, see [8].

Example 18 [8] Consider the system

$$\left[\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ \hline C_2 & D_{21} & 0 \end{array} \right] = \left[\begin{array}{cc|cc|c} -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ \hline 1 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} & 1 \\ \hline 1 & 1 & 0 & 1 & 0 \end{array} \right].$$

Then (33) becomes

$$R_H(\gamma) = R_J(\gamma) = \begin{bmatrix} \frac{1}{4} - \gamma^2 & 0 & 0 \\ 0 & \frac{1}{4} - \gamma^2 & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 \end{bmatrix}.$$

The Hamiltonian matrix (34) is

$$H(\gamma) = \left[\begin{array}{cc|cc} -1 & -1 & \frac{1}{4}\gamma^{-2} - 1 & \frac{1}{4}\gamma^{-2} - 1 \\ 0 & -1 - 1 & \frac{1}{4}\gamma^{-2} - 1 & \frac{1}{4}\gamma^{-2} - 1 \\ \hline \frac{1}{\frac{1}{4}\gamma^{-2} - 1} & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{array} \right],$$

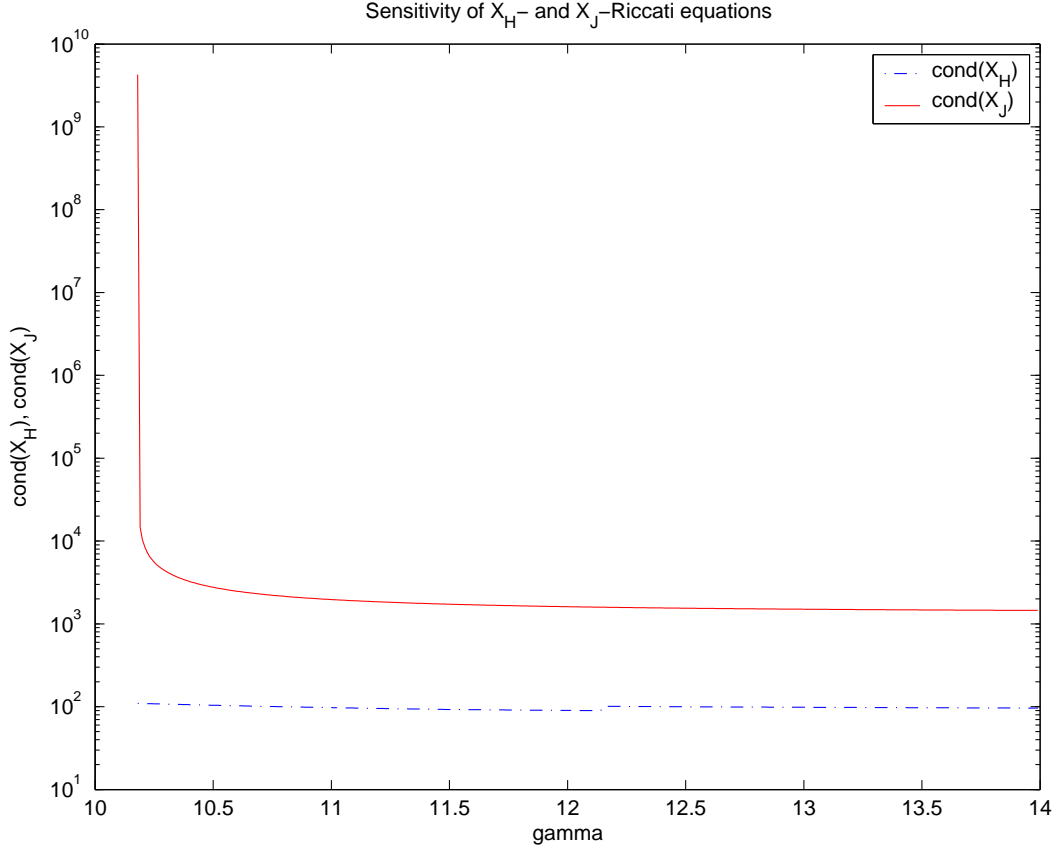


Figure 7: Conditioning of the solutions of Riccati equations as a function of γ

and (35) is

$$J(\gamma) = \left[\begin{array}{cc|cc} -1 & 0 & -\frac{4}{1-4\gamma^2} + \delta^2(\frac{1}{4}\gamma^{-2} - 1) & -\frac{1}{2}\gamma^{-2} + \frac{1}{4}\gamma^{-2} - 1 \\ 0 & -1 & -\frac{1}{2}\gamma^{-2} + \frac{1}{4}\gamma^{-2} - 1 & \frac{1}{4}\gamma^{-2} - 1 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right].$$

The positive semidefinite Riccati solution corresponding to $J(\gamma)$ is $X_J = 0$ and the positive semidefinite Riccati solution corresponding to $H(\gamma)$ is

$$X_H = \frac{3}{(1 - \frac{1}{4}\gamma^{-2})4} \times \left[\begin{array}{cc} \frac{1}{2} + \frac{1}{3(1+\sqrt{5})} & \frac{1}{1+\sqrt{5}} - \frac{1}{2} \\ \frac{1}{1+\sqrt{5}} - \frac{1}{2} & \left(\frac{1}{6} - \frac{1}{(1+\sqrt{5})(2+\sqrt{5})}\right) \end{array} \right].$$

As γ approaches the solution of the modified optimal H_∞ problem $\gamma_{\text{modopt}} = \frac{1}{2}$, the Riccati solution X_H converges to infinity, R_H and R_J become singular and the Hamiltonian matrix $H(\gamma)$ becomes ill-defined. The fourth condition in Theorem 2 never fails, because $\rho(X_J X_H) = 0$ for all $\gamma > \gamma_{\text{modopt}}$.

This example demonstrates that the conditioning of the modified optimal H_∞ control problem may deteriorate near the optimum and clearly in this case an iterative method that converges against γ_{modopt} will have to be terminated before the optimum is reached. Alternative formulations of the modified optimal H_∞ control problem where these difficulties do not occur are currently being investigated, [8], where in view of the discussion in Section 4 Riccati equations as well as the inversion of the matrices R_H, R_J is avoided.

6 Conclusion and challenges

We have discussed the sensitivity of several standard problems of linear control theory, including pole assignment, linear quadratic and H_∞ control. We have demonstrated that the mathematical formulation and the splitting of the problem into subproblems are essential factors in the conditioning of these problems. We have shown that standard approaches that are implemented in numerical toolboxes, which present widely accepted approaches in numerical control, may face problems coming from ill-conditioning. Some of these may be avoided by a reformulation of the problem but several open problems remain. In particular, complete software is missing for perturbation and error estimates for the following important problems in control theory:

- computation of matrix functions such as matrix exponential, matrix sign-function, etc.;
- solution of various classes of linear and non-linear matrix equations: Lyapunov and Sylvester equations, quadratic and fractional-affine equations (Riccati equations in particular);
- solution of unstructured and structured eigenstructure problems;
- computation of the matrices of the optimal and suboptimal controller for some H_∞ control problems;
- computation of the distance to uncontrollability (unobservability);
- computation of the distance (or some of its lower bounds) to unstabilizability (undetectability);
- investigation and computation of the sensitivity of general classes of H_∞ control problems.

In order to assess the accuracy of results and to be able to trust numerical results, such condition and accuracy estimates should accompany every computational procedure. Some modern software packages such as [9] provide such estimates. It is, however, unfortunately common practice in industrial use to turn these facilities off, even though this service will warn the user of numerical methods about possible failure. *What is the reason for such irresponsible action?* One reason is that the computation of the estimates somewhat slows down the method but, as we have learned in many discussions with engineers in industry, the main reason is that the user does not know how to interpret these warnings. But it should be clear that the developers of numerical software introduce these estimates not for the purpose of bothering the user with mathematical overcautioness. The idea is to make the user aware of difficulties with the computational problem or the numerical method and it should be an essential part of the curriculum to teach scientists and engineers how to interpret these warnings.

References

- [1] *IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985.* Institute of Electrical and Electronics Engineers, New York, 1985. Reprinted in SIGPLAN Notices, 22(2):9–25, 1987.
- [2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, PA, third edition, 1999.

- [3] M. Arnold. *Algorithms and conditioning for eigenvalue assignment*. PhD thesis, Northern Illinois University, De Kalb, Illinois, USA, 1993.
- [4] M. Arnold and B.N. Datta. Single-input eigenvalue assignment algorithms. A close-look. Report, Northern Illinois University, Department of Mathematical Sciences, De Kalb, IL. 60115, 1997.
- [5] M. Athans and P.L. Falb. *Optimal Control*. McGraw-Hill, New York, 1966.
- [6] Z. Bai, J. Demmel, and A. Mckenney. On computing condition numbers for the nonsymmetric eigenproblem. *ACM Trans. Math. Software*, 19:202–223, 1993.
- [7] P. Benner, R. Byers, V. Mehrmann, and H. Xu. Numerical computation of deflating subspaces of skew hamiltonian/hamiltonian pencils. *SIAM J. Matrix Anal. Appl.*, 24:165–190, 2002.
- [8] P. Benner, R. Byers, V. Mehrmann, and H. Xu. Robust methods for robust control. Technical report, Institut für Mathematik, TU Berlin, Str. des 17. Juni 136, D-10623 Berlin, FRG, 2003. in Preparation.
- [9] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT - a subroutine library in systems and control theory. *Appl. Comput. Contr., Sign., Circ.*, 1:499–532, 1999.
- [10] S.P. Bhattacharyya and E. De Souza. Pole assignment via Sylvester’s equations. *Systems Control Lett.*, 1(4):261–263, 1982.
- [11] S. Bittanti, A. Laub, , and J. C. Willems, editors. *The Riccati Equation*. Springer-Verlag, Berlin, 1991.
- [12] R. Byers. Numerical condition of the algebraic Riccati equation. *Contemp. Math.*, 47:35–49, 1985.
- [13] C.L. Cox and W.F. Moss. Backward error analysis for a pole assignment algorithm. *SIAM J. Matrix Anal. Appl.*, 10:446–456, 1989.
- [14] C.L. Cox and W.F. Moss. Backward error analysis for a pole assignment algorithm II: The complex case. *SIAM J. Matrix Anal. Appl.*, 13:1159–1171, 1992.
- [15] L. Dai. *Singular Control Systems*. Number 118 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 1989.
- [16] J.W. Demmel. On condition numbers and the distance to the nearest ill-posed problem. *Numer. Math.*, 51:251–289, 1987.
- [17] R. Eising. Between controllable and uncontrollable. *Systems and Control Letters*, 4:263–264, 1984.
- [18] P. Fuhrmann. *A Polynomial Approach to Linear Algebra*. Springer-Verlag, Berlin, 1996.
- [19] P. Gahinet and A. J. Laub. Numerically reliable computation of optimal performance in singular H_∞ control. *SIAM J. Cont. Optim.*, 35:1690–1710, 1997.
- [20] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, third edition, 1996.

- [21] M. Green and D.J.N Limebeer. *Linear Robust Control*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [22] J.J. Hench, C. He, V. Kučera, and V. Mehrmann. Dampening controllers via a Riccati equation approach. *IEEE Trans. Automat. Control*, AC-43:1280–1284, 1998.
- [23] G. Hewer and C. Kenney. The sensitivity of the stable Lyapunov equation. *SIAM J. Cont. Optim.*, 26:321–344, 1988.
- [24] N. J. Higham. Perturbation theory and backward error for $AX - XB = C$. *BIT*, 33:124–136, 1993.
- [25] Nicholas J. Higham. The Matrix Computation Toolbox. <http://www.ma.man.ac.uk/higham/mctoolbox>.
- [26] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.
- [27] N.J. Higham. FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (Algorithm 674). *ACM Trans. Math. Software*, 14:381–396, 1988.
- [28] T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [29] J. Kautsky, N. K. Nichols, and P. Van Dooren. Robust pole assignment in linear state feedback. *Internat. J. Control*, 41:1129–1155, 1985.
- [30] L.H. Keel, J.A. Fleming, and S.P. Bhattacharya. Minimum norm pole assignment via Sylvester’s equation. *Contemp. Math.*, 47:265–272, 1985.
- [31] C. Kenney and G. Hewer. The sensitivity of the algebraic and differential Riccati equations. *SIAM J. Cont. Optim.*, 28:50–69, 1990.
- [32] M. Konstantinov, V. Mehrmann, and P. Petkov. Perturbation analysis of Hamiltonian Schur and block-Schur forms. *SIAM J. Matrix Anal. Appl.*, 23:387–424, 2001.
- [33] M. Konstantinov and P. Petkov. Note on “Perturbation theory for algebraic Riccati equations”. *SIAM J. Matrix Anal. Appl.*, 21:327, 1999.
- [34] M. Konstantinov, P. Petkov, and D.W. Gu. Improved perturbation bounds for general quadratic matrix equations. *Numer. Func. Anal. Optim.*, 20:717–736, 1999.
- [35] M.M. Konstantinov, P.H. Petkov, and N.D. Christov. Conditioning of the continuous-time H_∞ optimisation problem. In *Proc. Third European Control Conference ECC’95*, pages 613–618, Rome, Italy, September 1995.
- [36] M.M. Konstantinov, P.Hr. Petkov, and N.D. Christov. Invariants and canonical forms for linear multivariable systems under the action of orthogonal transformation groups. *Kybernetika* (Prague), 17:413–424, 1981.
- [37] M.M. Konstantinov, P.Hr. Petkov, and N.D. Christov. Sensitivity analysis of the feedback synthesis problem. *IEEE Trans. Automat. Control*, 42:568–573, 1997.
- [38] P. Lancaster and L. Rodman. *The Algebraic Riccati Equation*. Oxford University Press, Oxford, 1995.

- [39] The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Mass, 01760. *MATLAB Version 6.5.0.180913a (R13)*, 2002.
- [40] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, 1991.
- [41] V. Mehrmann and H. Xu. An analysis of the pole placement problem. I. The single-input case. *Electr. Trans. Num. Anal.*, Vol 4:89–105, 1996.
- [42] V. Mehrmann and H. Xu. An analysis of the pole placement problem. II. The multi-input case. *Electr. Trans. Num. Anal.*, Vol 5:77–97, 1997.
- [43] V. Mehrmann and H. Xu. Choosing poles so that the single-input pole placement problem is well-conditioned. *SIAM J. Matrix Anal. Appl.*, 19:664–681, 1998.
- [44] V. Mehrmann and H. Xu. Numerical methods in control. *J. Comput. Appl. Math.*, 123:371–394, 2000.
- [45] G.S. Miminis and C.C. Paige. An algorithm for pole assignment of time-invariant linear systems. *Internat. J. Control*, 35:341–354, 1982.
- [46] G.S. Miminis and C.C. Paige. A direct algorithm for pole assignment of linear time-invariant multi-input linear systems using state feedback. *Automatica*, 24:343–356, 1988.
- [47] M. Overton. *Computing with IEEE Floating Point Arithmetic*. SIAM, Philadelphia, 2001.
- [48] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [49] P.Hr. Petkov, N.D. Christov, and M.M. Konstantinov. A computational algorithm for pole assignment of linear multiinput systems. *IEEE Trans. Automat. Control*, 31:1044–1047, 1986.
- [50] P.Hr. Petkov, N.D. Christov, and M.M. Konstantinov. *Computational Methods for Linear Control Systems*. Prentice-Hall, Hemel Hempstead, 1991.
- [51] P.Hr. Petkov, M.M. Konstantinov, D.W. Gu, and I. Postlethwaite. Optimal eigenstructure assignment of linear systems. In *Proc. 13 IFAC Congress*, volume C, pages 109–114, San Francisco, USA, 1996.
- [52] Polyx, Ltd, Prague, Czech Republic. *The Polynomial Toolbox, Version 2.5*, 2002.
- [53] L.S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishenko. *The Mathematical Theory of Optimal Processes*. Interscience, New York, 1962.
- [54] V. Sima. *Algorithms for Linear Quadratic Optimization*. Marcel Dekker Inc., New York, 1996.
- [55] G.W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, New York, 1990.
- [56] J.-G. Sun. Condition numbers of algebraic Riccati equations in the Frobenius norm. *Linear Algebra Appl.*, 350:237–261, 2002.

- [57] H.L. Trentelmann, M.L.J. Hautus, and A.A. Stoorvogel. *Control Theory for Linear Systems*. Springer Verlag, London, 2001.
- [58] P. Van Dooren. The generalized eigenstructure problem in linear system theory. *IEEE Trans. Automat. Control*, AC-26:111–129, 1981.
- [59] P. Van Dooren. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Sci. Statist. Comput.*, 2:121–135, 1981.
- [60] A. Varga. A Schur method for pole assignment. *IEEE Trans. Automat. Control*, AC-26:517–519, 1981.
- [61] A. Varga. Robust pole assignment techniques via state feedback. In *Proc. of IEEE Conference on Decision and Control CDC'2000*, pages 4655–4660, Sydney, Australia, 2000.
- [62] J.H. Wilkinson. *Rounding Errors in Algebraic Processes*. Prentice-Hall, Englewood Cliffs, 1963.
- [63] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, 1965.
- [64] W.M. Wonham. *Linear Multivariable Control: A Geometric Approach*. Springer-Verlag, New York, third edition, 1985.
- [65] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Englewood Cliffs, NJ, 1996.