Properties of Numerical Algorithms Related to Computing Controllability

CHRIS C. PAIGE

Abstract—**The numerical properties of some methods for computing controllability are used in an expository way to motivate a wider understanding of numerical computations. In particular, the numerical rank of a matrix, the numerical stability of algorithms, the sensitivity of problems, and the scaling of problems are discussed. A numerically stable algorithm is given for computing controllability, but it is pointed out that a measure of the distance of the given system from the nearest uncontrollable system would be more useful, and this appears to be an open computational problem.**

I. Introduction

THERE are several mathematically equivalent approaches to determining controllability or observability of time-invariant linear systems of the form

$$\dot{x} = \underset{n\times n}{A}x + \underset{n\times m}{B}u\,, \quad y = \underset{q\times n}{C}x \tag{1}$$

where A has eigenvalues $\lambda_1, \cdots, \lambda_n$. Unfortunately, different approaches lead to computational methods which can give markedly different results. Such discrepancies are caused by the rounding errors that occur during the computation, and one aim of this paper is to give some understanding of this behavior. Thus, although determining controllability of (1) is a simple enough problem mathematically, it is not such a straightforward computational problem, and it brings out some rather subtle numerical properties of algorithms. As a result, it is an excellent problem for exhibiting several general aspects of numerical algorithms that may help the control engineer in the choice or design of reliable numerical algorithms for more general problems. A second aim of this paper is to examine the controllability of (1) more closely, since some questions related to computing controllability of (1) remain unanswered. A numerically stable algorithm is given for computing controllability, but it is indicated why this is not really the full answer to the problem.

Three widely used mathematically equivalent results on controllability and observability of (1) are the following.

R1 [1]: The system (1) is controllable if and only if

$$\operatorname{rank}(B, AB, \cdots, A^{n-1}B) = n \tag{2}$$

and observable if and only if

$$\operatorname{rank}\left(C^T, A^TC^T, \cdots, (A^T)^{n-1}C^T\right) = n \tag{3}$$

where the superscript T denotes transpose.

R2 [2], [3]: The system (1) is controllable if and only if

$$\operatorname{rank}(B, A-\lambda_i I) = n, \qquad i = 1, \cdots, n \tag{4}$$

and observable if and only if

$$\operatorname{rank}\left(C^T, A^T-\lambda_i I\right) = n, \qquad i = 1, \cdots, n. \tag{5}$$

R3: The system (1) is controllable and observable if and only if there exists a matrix K such that the eigenvalues of

$$A + BKC \tag{6}$$

are all different from each eigenvalue of A.

These results can lead to several different computational approaches, three of which are the following.

C1: Form the matrices in (2) and (3) and compute their ranks.

C2: Compute the eigenvalues of A and compute the ranks of the matrices in (4) and (5).

C3 [4]: Compute the eigenvalues of A, compute a random matrix K, compute the eigenvalues of $A + BKC$, and compare these with those of A.

The first five sections of this paper will point out some general properties of numerical algorithms as they relate to the above computational approaches. Because of the similarity of the two problems, it will save space to consider controllability alone, for example, taking $C = I$ in (6). The treatment for observability will be obvious. The remainder of the paper will describe a different computational approach which is closely related to both R2 and R3, and will discuss scaling and make a comment on deriving some measure of controllability. Any uncontrollable system is in a certain sense arbitrarily close to some controllable system, but a controllable system may or may not be close to some uncontrollable one. Thus, a measure of how far a controllable system is from the nearest uncontrollable system would be of far greater use than just knowing that the system is controllable.

The results R1, R2, and R3 are valid for discrete systems, and everything in this paper will also hold for such systems.

Manuscript received April 11, 1979; revised June 13, 1980. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant A8652.

The author is with the School of Computer Science, McGill University, Montreal, P.Q., Canada.

0018-9286/81/0200-0130$00.75

The paper is intended to be largely expository. It is not a survey paper, and the references will be somewhat incomplete as a result of the author's limited knowledge of the control literature. It is hoped that a combination of numerical experience with some understanding of control systems will allow the author to communicate some useful numerical ideas to a fairly general audience in the area of control systems. The paper is a rewrite and extension of [26].

II. Numerical Rank of a Matrix

Equations (2) and (4) for determining controllability are based on finding the rank of a matrix. This is a reasonably straightforward task computationally, and is most reliably done by computing the singular value decomposition (SVD) of the matrix (see [7], [8], and the implementation in [9]). If we assume that $n \geqslant m$, the SVD of an $n \times m$ matrix B is

$$\begin{aligned} &B = U_1 S_1 V_1^H, \\ &U_1^H U_1 = V_1^H V_1 = V_1 V_1^H = I_m, \\ &S_1 = \operatorname{diag}(\sigma_1, \cdots, \sigma_m), \quad \sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_m \geqslant 0 \end{aligned} \tag{7}$$

where the superscript H denotes complex conjugate transpose and can be replaced by T if B is real. Here the real scalars $\sigma_1, \cdots, \sigma_m$ are called the singular values of B, V_1 is a unitary matrix of right singular vectors, and the left singular vectors are the columns of U_1, which is only part of a unitary matrix if $m < n$.

An attractive property of the singular values is that they are not very sensitive to changes in the matrix. In fact, if $\tilde{\sigma}_1 \geqslant \cdots \geqslant \tilde{\sigma}_m$ are the singular values of the perturbed matrix $B + \delta B$, then (see, for example, [10, p. 321] or [11, p. 25])

$$|\sigma_i - \tilde{\sigma}_i| \leqslant \|\delta B\|_2, \qquad i = 1, \cdots, m, \tag{8}$$

$$\|B\|_2 \triangleq \sigma_{\max}(B) = \sigma_1. \tag{9}$$

Thus, singular values are "well conditioned" with respect to perturbations in the matrix.

Ideally, if $\sigma_1 \geqslant \cdots \geqslant \sigma_r > 0$ and $\sigma_{r+1} = \cdots = \sigma_m = 0$, then B has rank r, and from (8), for any $j \leqslant r$ it will require a change $\|\delta B\|_2 \geqslant \sigma_j$ to produce a perturbed matrix $B + \delta B$ of rank less than j. In fact, $B + \delta B \triangleq U_1 \operatorname{diag}(\sigma_1, \cdots, \sigma_{j-1}, 0, \cdots, 0) V_1^H$ is such a matrix with $\|\delta B\|_2 = \sigma_j$. Thus, the singular values of a matrix tell us not only the rank, but also how far the matrix is from the nearest matrix of a given lower rank, and the SVD gives us such a matrix.

Any matrix is arbitrarily close to a matrix of full rank as can be seen by altering every zero σ_i in (7) to an arbitrarily small value $\epsilon > 0$. So unless the elements of a matrix are known to infinite accuracy, the exact rank of the matrix will usually be unobtainable. But what is of interest is the closeness of the matrix to one of a certain rank, and the singular values tell us just this.

In a practical problem, the actual matrix B in (7) may not be available; instead we will often have a matrix $\bar{B}$ and a measure of accuracy BTOL such that

$$\|B - \bar{B}\|_2 \leqslant \text{BTOL} \cdot \|\bar{B}\|_2. \tag{10}$$

A numerically stable algorithm [9] applied to $\bar{B}$ on a floating-point digital computer will give the singular values $\tilde{\sigma}_1 \geqslant \cdots \geqslant \tilde{\sigma}_m$ not of $\bar{B}$, but of a nearby matrix $\tilde{B}$ where it is known from a rounding error analysis that

$$\|\bar{B} - \tilde{B}\|_2 \leqslant \text{STOL} \cdot \|\bar{B}\|_2, \tag{11}$$

the scalar STOL being the product of the relative precision of the computer, the number of iterations required in the computation, and a low order polynomial in n. Note that

$$\begin{aligned} \|B - \tilde{B}\|_2 &\leqslant (\text{BTOL} + \text{STOL}) \cdot \|\bar{B}\|_2 \\ &\leqslant \text{TOL} \cdot \tilde{\sigma}_1 \end{aligned}$$

$$\text{TOL} \triangleq (\text{BTOL} + \text{STOL})/(1 - \text{STOL}), \tag{12}$$

so we see from (8) that if

$$\tilde{\sigma}_r > \text{TOL} \cdot \tilde{\sigma}_1, \quad \tilde{\sigma}_{r+1} \leqslant \text{TOL} \cdot \tilde{\sigma}_1, \tag{13}$$

then σ_r could not be zero, but is is possible that $\sigma_{r+1}, \cdots, \sigma_m$ could be, and so we can say that B has effective rank r, meaning we know that B has rank at least r, but we are unable to say anything more with this precision of data and computation. If BTOL $= 0$ above, we sometimes say that B has numerical rank r. In general, B would have to be perturbed by a matrix with norm at least

$$\tilde{\sigma}_r - \text{TOL} \cdot \tilde{\sigma}_1 \tag{14}$$

to obtain a matrix of rank $r - 1$.

Computing the SVD is fairly expensive, and another approach to determining rank is to compute an orthogonal decomposition

$$Q^H B P = \begin{pmatrix} R & S \\ 0 & 0 \end{pmatrix} \tag{15}$$

where Q is unitary, R is upper triangular and nonsingular, and P is a permutation matrix; see, for example, [11, ch. 3]. In theory, the dimension of R gives the rank of B, but it can be difficult computationally to decide the effective rank of (R, S), and the result does not give the distance to the nearest matrix of lower rank. If economy of computation is important, this approach can be tried, and if it does not give a clear enough result, the computation can be continued so that in (15), using a general unitary matrix P, R is altered to $\tilde{R}$ and S becomes zero [11, ch. 3]. The singular values of $\tilde{R}$ are the nonzero singular values of B, and the smallest one can be estimated as in [12] or [13]. If this result is still not satisfactory, the SVD of $\tilde{R}$ can be computed.

The main purpose of this section was to emphasize that we cannot, in general, expect to find the true mathemati-

cal rank of a given matrix, as this is meaningless when we are dealing with numbers of a limited precision. However, it is straightforward to compute the effective rank of the given matrix, that is, the rank of a matrix which to within the precision of data and computation could be the given matrix. The SVD also gives a reliable indicator of the distance to the nearest matrix of lesser rank, and, in fact, it is more useful to know the computed singular values than the "rank."

Many of the ideas in this section have already been presented in the control literature, for example, in [25]. Much of the early work on the subject appeared in the statistical and numerical literatures.

III. Sensitivity of Problems

Perturbation analysis is the study of the sensitivity of the results of problems to changes in data. We saw in (8) that small changes in B led to small changes in its singular values, and so the singular value problem is well conditioned. On the other hand, if we consider the solution of equations problem $Az=d$ with $n\times n$ nonsingular A, and perturb d to $d+\delta d$, the bound we obtain for the size of the change δz in the solution (see [10, p. 194]) is $\|\delta z\|_2/\|z\|_2 \leqslant \chi(A)\|\delta d\|_2/\|d\|_2$ where $\chi(A)$ is called the condition number of A for solution of equations. Here the multiplying factor

$$\chi(A) \triangleq \sigma_1(A)/\sigma_n(A) \geqslant 1 \tag{16}$$

is the ratio of the largest to smallest singular values of A, and the solution of equations problem for A is ill conditioned if this is large.

Perturbation analysis is a purely mathematical study, and the sensitivity of a problem is independent of how that problem is solved. However, the numerical analyst must always keep the sensitivity of problems clearly in mind when designing algorithms, for if in the process of solution the original problem is transformed to a more sensitive problem, the error in the computed solution will usually reflect this and be larger than necessary.

This is the main weakness of computational approach C1. To exhibit this, consider the clearly controllable system

$$\dot{x}=\begin{bmatrix} 1 & & & \\ & 2^{-1} & & \\ & & \ddots & \\ & & & 2^{1-n} \end{bmatrix} x+\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u. \tag{17}$$

Here $(B, AB, \cdots, A^{n-1}B)$ has (i, j) element $2^{(i-1)(1-j)}$ which can be formed and stored with full accuracy on most computers. Even so, with $n=10$ this matrix has its three smallest singular values 0.613×10^{-12}, 0.364×10^{-9}, 0.712×10^{-7}, and its numerical rank will be less than 10 using the SVD on a computer with relative precision no smaller than 10^{-12}. In this case, C1 would conclude that an obviously controllable system was uncontrollable. In fact, a perturbation of about 10^{-3} must be made to A to obtain an uncontrollable system, a very large change in comparison with 10^{-12}. Fortunately, we can do better by using C2. It is fairly straightforward to show that in this case, the smallest singular value of $(B, A-\lambda_i I)$ is greater than $2^{-(n+1)}$ for every i, and for $n=10$, C2 would certainly give the correct result on a computer with precision 10^{-5} or smaller.

The weakness of C1 is that the original problem is transformed to the intermediate problem (2) which can be sensitive to far smaller perturbations than would affect either the original problem or the intermediate problem (4), and it is widely recognized that C1 must be discarded as a general computational approach. As is often the case, one good example is sufficient to condemn a poor algorithm. To pinpoint the difficulty, the example was chosen so that the matrix in (2) could be accurately computed, showing that it was the very small singular values of this intermediate matrix that led to the trouble; the problem would often be made worse by the rounding errors that would normally occur in the computation of this matrix.

A more general lesson can be learned from this example, and this is the danger of transforming problems by matrix multiplication. If the transforming matrix is ill conditioned for solution of equations [see (16)], the transformed problem will usually be more sensitive than the original problem. This is what happened in C1. The well-known approach [14] of avoiding forming C^TC in solving

$$\text{minimize}\|y-Cx\|_2$$

is an important instance of eliminating this danger, and has led, for example, to the introduction of "square-root" filtering; see [15]. The warning against multiplication by general matrices applies equally well to multiplication by inverses, and it is advisable to avoid forming inverses wherever possible, and to arrange computations so that any unavoidable solution of equations occurs as late in the process as possible.

However, not all matrix multiplications should be avoided. For example, if $\chi(A)$ in (16) is close to unity, then there is negligible loss in forming AB. Since $\chi(Q)=1$ for any orthogonal matrix Q, transforming data with orthogonal matrices is quite acceptable, and is often the best way of solving many matrix problems numerically; see [7]–[11], [14]–[18].

IV. Sensitivity of Eigenvalues

The approaches C2 and C3 depend on finding eigenvalues, and so it is important to consider the sensitivity of eigenvalues of general square matrices. This subject is treated in depth in [16]. Unlike singular values, the eigenvalues of some matrices can be very ill conditioned, that is, very sensitive to small changes in the matrix, and this can be the case even when the eigenvalues are well separated. Wilkinson [16, p. 90] gives the following illuminat-

ing example of a 20×20 matrix:

$$\hat{A}=\begin{bmatrix} 20 & 20 & & & & & \\ & 19 & 20 & & & & \\ & & 18 & 20 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 2 & 20 \\ & & & & & & 1 \end{bmatrix} \tag{18}$$

with well-separated eigenvalues $\lambda_i = i$, $i = 1, \cdots, 20$. If we perturb $\hat{A}$ by a very small quantity ϵ in the (20, 1) position, we obtain eigenvalues, to first order in ϵ,

$$\lambda_i(\epsilon) \sim \lambda_i + \alpha_i \epsilon.$$

The distressing fact is that these α_i are very large, for example,

$$\alpha_{20} = -\alpha_1 \doteq 4\times 10^7$$
$$\alpha_{10} = -\alpha_{11} = 4\times 10^{12}$$

so the eigenvalues of $\hat{A}$ are extremely ill conditioned.

Numerically stable algorithms for finding the eigenvalues of a given matrix A [17], [18] will give the eigenvalues of a nearby matrix $\tilde{A}$ [16] where, in a similar manner to (11),

$$\|A-\tilde{A}\|_2 \leqslant \text{ETOL}\cdot\|A\|_2. \tag{19}$$

However, unlike the singular value case, if an eigenvalue λ_i of A is ill conditioned, then we have seen that the corresponding eigenvalue $\tilde{\lambda}_i$ of $\tilde{A}$ may be very different.

An immediate conclusion of the foregoing is that it is poor practice to attempt to determine if a square matrix is singular by computing its eigenvalues; it is the singular values that are required. Note that if A has singular values $\sigma_1 \geqslant \cdots \geqslant \sigma_n$ and eigenvalues ordered so that $|\lambda_1| \geqslant \cdots \geqslant |\lambda_n|$, then it is straightforward to show that

$$0 \leqslant \sigma_n \leqslant |\lambda_n| \leqslant |\lambda_1| \leqslant \sigma_1, \tag{20}$$

but these inequalities can be very loose. For instance, with large α, the matrix

$$\begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix}$$

has $\sigma_2 \sim |\alpha|^{-1}$ and $\sigma_1 \sim |\alpha|$. Thus, we see that nonzero eigenvalues may give no indication of how close a matrix is to singular.

The possibility of ill conditioned eigenvalues explains why approach C3 will sometimes give wrong answers. Even if A and $A+BK$ do have an equal eigenvalue λ_i, it will be difficult to decide this computationally when λ_i is ill conditioned. Instead of finding the eigenvalue λ_i of A, we will compute an eigenvalue $\tilde{\lambda}_i$ of $\tilde{A} = A + \delta A$ where

$$\|\delta A\|_2 \leqslant \text{ETOL}\cdot\|A\|_2, \tag{21}$$

and instead of finding an eigenvalue μ_i of $A+BK$, we will compute an eigenvalue $\tilde{\mu}_i$ of $A+BK+\delta A'$:

$$\|\delta A'\|_2 \leqslant \text{ETOL}\cdot\|A+BK\|_2. \tag{22}$$

There is no reason to expect any correlation between δA and $\delta A'$, and even if $\mu_i = \lambda_i$, we could have $\tilde{\mu}_i$ very different from μ_i and $\tilde{\lambda}_i$ very different from λ_i, and in different directions, so that $\tilde{\mu}_i$ and $\tilde{\lambda}_i$ look like distinctly different eigenvalues.

As an example, computations were carried out on the uncontrollable system with

$$A = Q^T\hat{A}Q, \quad B^T = (1, 1, \cdots, 1, 0)Q \tag{23}$$

where $\hat{A}$ is the matrix in (18) and Q is a random orthogonal matrix found by transforming a square matrix whose elements are uniform random numbers on $(-1, 1)$ to upper triangular form using the QR decomposition (see, for example, [10], [11]) and taking Q as the transforming matrix. The computed eigenvalues $\tilde{\lambda}_i$ of A and $\tilde{\mu}_i$ of $A+BK$ were obtained using single precision (six significant hexadecimal digits) on the AMDAHL 470 V7 at McGill University. The elements of K were random numbers from a uniform distribution on $(-1, 1)$. The results are given in Table I where, for comparison with C2, the values

$$\rho(B, A-\tilde{\lambda}_i I) \triangleq \text{ratio of smallest to largest singular value of } (B, A-\tilde{\lambda}_i I) \tag{24}$$

were also computed. This value was the same for both eigenvalues of a complex conjugate pair.

TABLE I
RESULTS OF COMPUTATIONAL TESTS ON (23)

Eigenvalues $\tilde{\lambda}_i(A)$	Eigenvalues $\mu_i(A+BK)$	$\rho(B, A-\tilde{\lambda}_i I)$
$-.32985 \pm j1.06242$	.99999	.002
$.92191 \pm j3.13716$	$-8.95872 \pm j\ 3.73260$	.004
$3.00339 \pm j4.80414$	$-5.11682 \pm j\ 9.54329$	.007
$5.40114 \pm j6.17864$	$-\ .75203 \pm j14.148167$	.012
$8.43769 \pm j7.24713$	$5.77659 \pm j15.58436$	.018
$11.82747 \pm j7.47463$	$11.42828 \pm j14.28694$	.026
$15.10917 \pm j6.90721$	$13.30227 \pm j12.90197$	.032
$18.06886 \pm j5.66313$	$18.59961 \pm j14.34739$	.040
$20.49720 \pm j3.81950$	$23.94877 \pm j11.80677$	.052
$22.06287 \pm j1.38948$	$28.45618 \pm j\ 8.45907$	.064
	32.68478	

In theory, one eigenvalue of A should be the same as an eigenvalue of $A+BK$, this one being unity. But as can be seen from Table I, the computed eigenvalues $\tilde{\lambda}_i$ of $\tilde{A}$ are almost unrelated to the true eigenvalues of A and the computed eigenvalues $\tilde{\mu}_i$ of $A+BK$. A user seeing only the $\tilde{\lambda}_i$ and $\tilde{\mu}_i$ would probably conclude from C3 that the system is clearly controllable. The algorithm C2 fared

little better. In theory, one value of ρ should be zero, but there is only the slightest hint that some modes could be "less controllable" than others. C2 was also carried out with the true eigenvalue unity of A, and this gave the computed value in (24):

$$\rho(B, A-I)=5\times 10^{-8}, \tag{25}$$

indicating that it is the eigenvalue computation that caused the failure of C2, just as for C3.

A good method that is not based on an eigensolution will be described in Section VI. It can be shown that this method is numerically stable, and so Section V will briefly consider numerical stability as related to the problem of determining controllability.

V. Numerically Stable Algorithms

It was mentioned that a numerically stable algorithm for finding the eigenvalues of a given matrix A can be relied on to give eigenvalues which are exact for a nearby matrix $\tilde{A}$, with "nearby" defined in (19). To understand this further, we note that the most we can hope for when working with floating-point numbers on a finite precision computer is to obtain the correct answer for a set of numbers which is in some sense close to the given data; for example, our data values are usually slightly altered just by storing them in the computer. In line with these ideas, we would like to find a numerical algorithm for determining controllability which, when applied to the data A and B in (1), would always give the exact answer for a nearby system:

$$\begin{aligned} &\dot{x}=(A+\delta A)x+(B+\delta B)u,\\ &\|\delta A\|_2 \leqslant \mathrm{CTOL}\cdot\|A\|_2,\\ &\|\delta B\|_2 \leqslant \mathrm{CTOL}\cdot\|B\|_2, \end{aligned} \tag{26}$$

the scalar CTOL being the product of the relative precision of the computer, the number of transformations required in the computation, and a low-order polynomial in the dimension of the system. For a reliable algorithm, a careful rounding error analysis would seek to prove such a result, and perhaps give an expression for CTOL which must be independent of A and B apart from their dimensions.

There exist effective algorithms for which it cannot be said that the computed results are exact for slightly perturbed data; nevertheless, such results hold for most basic matrix computations. The major work of realizing the importance of this approach in understanding the numerical properties of algorithms and of developing the tools and techniques for rounding error analyses was carried out by Wilkinson in [16] and elsewhere, and for many algorithms, a few easy observations and a reference to Wilkinson's work are all that is required to prove numerical stability. Note that a result of the form (19) by itself tells us nothing about the accuracy of the computed eigenvalues. The accuracy of the computed results will depend on their sensitivity to such equivalent changes in data.

VI. Transformation to an Equivalent System

The aim here will be to transform the original system (1) to an equivalent system for which controllability is obvious. Careful use will be made of orthogonal matrices so that the computed system will be equivalent to a system very close to the original system in the sense of (26), and the algorithm will be numerically stable in this ideal sense. The algorithm will be presented for the more general model,

$$E\dot{x}=Ax+Bu, \qquad E \text{ nonsingular}, \tag{27}$$

since little will be lost in simplicity. The mathematical formulation of a realistic system will sometimes take this form naturally (see [19, p. 121]), and it will usually be preferable numerically to transform E, A, and B rather than to work with the often more poorly conditioned model of the form (1) with $E^{-1}A$ and $E^{-1}B$; see Section III.

The model (27) will be transformed by orthogonal matrices P, Q, Z to give

$$\begin{aligned} \tilde{E}\dot{\tilde{x}} &\triangleq (Q^TEZ)(Z^T\dot{x})\\ &=(Q^TAZ)(Z^Tx)+(Q^TBP)(P^Tu)\\ &\triangleq \tilde{A}\tilde{x}+\tilde{B}\tilde{u} \end{aligned} \tag{28}$$

where for some integer $k \leqslant n$ and $n_0 \triangleq m$

$$\tilde{E}=\begin{bmatrix} E_{11} & E_{12} & \cdot & \cdot\\ 0 & E_{22} & \cdot & \cdot\\ 0 & 0 & \cdot & \cdot\\ 0 & 0 & 0 & E_{kk} \end{bmatrix},$$

$$\tilde{A}=\begin{bmatrix} A_{11} & A_{12} & \cdot & \cdot\\ A_{21} & A_{22} & \cdot & \cdot\\ 0 & \cdot & \cdot & \cdot\\ 0 & 0 & A_{k,k-1} & A_{kk} \end{bmatrix}, \quad \tilde{B}=\begin{bmatrix} A_{10}\\ 0\\ 0\\ 0 \end{bmatrix}, \tag{29}$$

$$E_{ii} \text{ and } A_{ii} \text{ are } n_i\times n_i, \qquad i=1,\cdots,k, \tag{30}$$

$$A_{i,i-1} \text{ is } n_i\times n_{i-1} \text{ of rank } n_i, \qquad i=1,\cdots,k-1. \tag{31}$$

Here $\tilde{E}$ is block upper triangular and $\tilde{A}$ is block upper Hessenberg. Since E is nonsingular, the E_{ii} are too. This transformation is a direct computation rather than an iterative one, such as must be used in an eigenvalue computation, and it stops when either

$$A_{k,k-1} \text{ has rank } n_k, \text{ meaning that (27) is controllable,} \tag{32}$$

or

$$A_{k,k-1}=0, \text{ meaning that (27) is not controllable.} \tag{33}$$

To justify this criterion, note that the criterion (4) for (1) is equivalent to

$$\text{rank}\,(B, A-\lambda I)=n \qquad \text{for all } \lambda \tag{34}$$

since if $\lambda \neq \lambda_i$, then $A-\lambda I$ has rank n. If we transform (28) to the form of (1), then (34) becomes rank $(\tilde{E}^{-1}\tilde{B}, \tilde{E}^{-1}A-\lambda I)=n$ for all λ, which is equivalent to

$$\text{rank}\,(\tilde{B}, \tilde{A}-\lambda \tilde{E})=n \qquad \text{for all } \lambda, \tag{35}$$

so this is a criterion for (28) to be controllable. If the matrices in (29) are substituted in (35) and (32) holds, then (35) holds since $A_{1,0}, \cdots, A_{k,k-1}$ all have full row rank and (28) is controllable. If (33) holds, then (35) will not be true for λ an eigenvalue of the generalized eigenproblem

$$A_{kk}z=\lambda E_{kk}z, \qquad z\neq 0 \tag{36}$$

and the system is uncontrollable. The transformed system (28) is controllable if and only if the original system is, so now we just have to show how to obtain the forms in (29), (30), and (31).

Note that if $E=I$ in (27), then taking $Z=Q$ gives $\tilde{E}=I$ in (28), (29), and (30), and, in fact, for models of the form (1), we would use

$$Z=Q \tag{37}$$

and save time by not carrying out the computation Q^TEZ. The following algorithm description is easier to understand initially if this is assumed.

There are several ways to transform the system to the form (28)–(31). The algorithm here requires k major steps, the first being on

$$\left(A_{10}^{(1)}, A_{11}^{(1)}, E_{11}^{(1)}\right) \triangleq (B, A, E). \tag{38}$$

In the ith step, orthogonal matrices Q_i, P_i, and then Z_i are chosen to give

$$Q_i^T\left(A_{i,i-1}^{(i)}, A_{ii}^{(i)}, E_{ii}^{(i)}\right)\begin{bmatrix} P_i & 0 & 0 \\ 0 & Z_i & 0 \\ 0 & 0 & Z_i \end{bmatrix}$$

$$=\left[\begin{array}{c|cc|cc} A_{i,i-1} & A_{ii}^{(i+1)} & A_{i,i+1}^{(i+1)} & E_{ii}^{(i+1)} & E_{i,i+1}^{(i+1)} \\ 0 & A_{i+1,i}^{(i+1)} & A_{i+1,i+1}^{(i+1)} & 0 & E_{i+1,i+1}^{(i+1)} \end{array}\right] \tag{39}$$

with $A_{i,i-1}$ as in (31) and $E_{ii}^{(i+1)}$ and $A_{ii}^{(i+1)}$ being $n_i \times n_i$. Orthogonal Q_i and P_i are chosen purely to transform $A_{i,i-1}^{(i)}$ to full row rank $A_{i,i-1}$, for example, as in (15). If $A_{i,i-1}^{(i)}$ is already zero or has full row rank, the algorithm stops with $k=i$. Otherwise, this ith step is completed by choosing Z_i to give upper triangular $Q_i^TE_{ii}^{(i)}Z_i$ which can be partitioned as in (39). Since $E_{ii}^{(i)}$ is nonsingular, this last is a straightforward computation; see [10], [11], or [14]. The Q_i and Z_i are also applied to $A_{ii}^{(i)}$ in (39). If the algorithm did not stop this general step in repeated on the nonzero matrices in the bottom block of rows on the right-hand side of (39).

Of course, the transformations are applied to the complete matrices in (29), the final subscripts in (39) corresponding to those in (29). The full k steps can be visualized by considering (29). In the first step, A_{10} is finalized and the $\tilde{E}$ matrix becomes upper triangular. In the second step, A_{21}, A_{11}, and E_{11} are finalized and the rest of the $\tilde{E}$ matrix again becomes upper triangular, and so on. In the kth, step $A_{k,k-1}^{(k)}$ is either zero or of full row rank, and the last two blocks of columns of $\tilde{E}$ and $\tilde{A}$ have been finalized. Thus, the kth step only entails computations to check if $A_{k,k-1}^{(k)}$ is zero or has full row rank. Obviously, $k \leqslant n$, and this is a finite computation. In practice, this algorithm would be refined somewhat for efficiency.

If reliable orthogonal decompositions are used in this transformation, then a straightforward application of the rounding error results of Wilkinson [16] shows that the computed matrices in (29) are exact for a system which is very close to (27):

$$\begin{aligned} &(E+\delta E)\dot{x}=(A+\delta A)x+(B+\delta B)u, \\ &\|\delta E\|_2 \leqslant \text{CTOL}\cdot\|E\|_2, \\ &\|\delta A\|_2 \leqslant \text{CTOL}\cdot\|A\|_2, \\ &\|\delta B\|_2 \leqslant \text{CTOL}\cdot\|B\|_2. \end{aligned} \tag{40}$$

Here CTOL is as described for (26). Since the controllability of the transformed system is obvious, it means that we will decide the controllability of a system very close to the given system (27), and so this is, by definition, a numerically stable algorithm for deciding controllability.

Finding full row rank $A_{i,i-1}$ requires a rank determination in (39), and as always, a tolerance must be used for testing for numerically zero elements. If a computation of the form (15) for this reduction does not give sufficient confidence, then it should be augmented, or replaced, by the singular value decomposition (SVD). In either case, using a numerically stable SVD, the result (40) holds.

Finally, if the system does turn out to be uncontrollable, then the generalized eigenproblem (36) could be solved to determine if the system is stabilizable. The uncontrollable subspace is obvious from the form of (28) and (29).

This algorithm will give the correct answer for a nearby system, but to compare its actual performance with C2 and C3, it was applied to the same ill conditioned problem (23) using the same precision of computation. Since in (23) B is a column vector, the $A_{i,i-1}$ in (29) are scalars, and the present algorithm will never require the SVD and will be much faster than C1, C2, or C3. Ideally,

$$A_{i,i-1}\neq 0, \qquad i=1,2,\cdots,19; \quad A_{20,19}=0.$$

The computations gave $A_{1,0}=4.35887$; $A_{2,1}=8.29969$; $17<|A_{i,i-1}|<22$, $i=3,4,\cdots,19$; and $A_{20,19}=0.0000027$, showing that the system is uncontrollable to this precision of computation. In light of the earlier computations with C2 and C3, this was a satisfyingly good result. To see if

this was just luck, the algorithm was applied to (23) with 100 different random orthogonal matrices Q. In all cases, similar good results were obtained.

Matrix transformations like (39) form some of the tools of the numerical analyst's trade, and related algorithms are used by Wilkinson [20], [21] and Van Dooren [22] in analyzing the generalized eigenvalue problem, and by Van Dooren [23] in analyzing general linear systems. An earlier work by Tse *et al.* [24] also suggested reducing the system to the form of (29), but instead of using orthogonal transformation matrices [24] uses matrices which can be very illconditioned for the solution of equations, and the method appears to be numerically unstable. The effect on numerical computations of general transformations and scaling in particular will be examined further in the next section.

VII. Coordinate Transformations and Scaling

If D_1, D_2, and D_3 are nonsingular matrices,

$$\dot{x}_1 = D_1 A D_1^{-1} x_1 + D_1 B D_2^{-1} u_1, \quad y_1 = D_3 C D_1^{-1} x_1 \quad (41)$$

is mathematically equivalent to (1), as can be seen by taking $x_1 = D_1 x$, $u_1 = D_2 u$, and $y_1 = D_3 y$. This is called coordinate transformation. If the D_i matrices are also positive definite and diagonal, it is usually referred to as scaling. With exact scaling, the elements of each D_i are chosen to be integer powers of the floating-point computer number base, and in this case, a transformation such as $D_1 B D_2^{-1}$ will introduce no rounding errors where here we consider B to be already stored in the computer. However, when a numerically stable algorithm is applied to the scaled data, we find that even exact scaling can have a significant effect on the computed results.

There are differing opinions as to what an ideal scaling should be. Some of the confusion can be resolved by realizing that there are two fairly distinct reasons for scaling. In the first place, the engineer would like to choose coordinate axes (transformations) and units (scaling) so that the mathematical problem accurately reflects the sensitivity of the physical problem. For example, if the physical system is close to being uncontrollable, the mathematical model should reflect this accurately. Later the numerical analyst might want to scale to minimize the effect of rounding errors on the computed solution. The first reason is by far the more important one, and if the mathematical problem accurately reflects the sensitivity of the physical problem, then the numerical analyst should be careful not to alter this sensitivity in any way so that he or she can compute this sensitivity accurately. The beauty of orthogonal transformations is that they do not alter norms, and effectively only rotate axes, each axis maintaining its exact relation to the others throughout. As a result, it can usually be shown that orthogonal transformations will not alter the sensitivity of a problem. On the other hand, general coordinate transformations with D_i in (41) being ill conditioned for the solution of equations will usually not only alter the sensitivity of the problem, but introduce excessive rounding errors as well.

Setting up a mathematical model so that the mathematical problem accurately reflects the sensitivity of the physical problem may not be straightforward. The correct choice of coordinate axes will hopefully be clear from the physics of the problem, but the choice of scaling is often not so clear since physical quantities can be expressed in different units. The scaling may depend on the particular problem being solved for the model. For the controllability problem, Moore [5] suggests that if the state variables are in close correspondence with physical variables, then a useful scaling would be "on a per unit basis so that if in (1) it happens that $\int_0^\infty x_i^2\,dt$ is relatively small for some input u, we may conclude that the associated physical variable is only slightly excited by the input." Perhaps such a scaling could be obtained *a priori* by deciding on the smallest significant change for each element of x, y, and u and scaling so these are all equal. If the decision can be made as to what each smallest significant change is, then this scaling is trivial to apply, but it is not at first glance clear that this necessarily achieves Moore's aim. However, balancing variables like this seems reasonable in that a significant change in a physical variable would correspond to a significant change in the model variable.

In the next section, an apparently different scaling will be suggested. This will depend on the model constants A, B, and C rather than the model variables x, y, and u, but seems justified for the situation to be considered.

As a result of this discussion, it can be seen that the best a numerical analyst can do is to design algorithms which are numerically stable for the data to which they are applied. However, such algorithms can only be assured of giving physically meaningful results if the data to which they are applied have been sufficiently well scaled, and the axes for the mathematical model have been correctly chosen and so are physically meaningful for the problem being examined. This choice is best handled by the engineer abstracting the mathematical model from the physical system.

VIII. Distance from an Uncontrollable System

So far we have been asking whether a system is controllable or not, and this has led us to ask, for example, if (4) holds or if it does not. Such yes–no answers are suspect since it is well known that any uncontrollable system (1) with $m > 0$ is arbitrarily close to a controllable system, whereas if we have to carry out general computational transformations, we have seen in Section V that we can at best test the controllability of a nearby system. This

nearby system may well be controllable even if the original system is not, and so it is hardly sufficient to just ask if a system is controllable.

One is thus naturally led to seek some measure of how far a given system is from an uncontrollable one. One obvious algebraic approach for (1) is to take

$$\mu(A, B) \triangleq \text{minimum} \|(\delta A, \delta B)\|_2 \text{ such that the system defined by } (A+\delta A, B+\delta B) \text{ is uncontrollable} \quad (42)$$

as a measure of controllability. This measure with either δA or δB set to zero might also be of interest. The measure $\mu(A, B)$ will be invariant under orthogonal transformations, but could be altered by nonunitary transformations and scaling. A numerically stable computation for finding $\mu(A, B)$ would ideally give the exact result for a nearby system. But it is clear that $\mu(A, B)$ is perfectly conditioned with respect to changes in A and B, and so the computed $\mu(A, B)$ would be very close to the true value.

Such an approach would have the following practical advantage. Suppose that the model matrices A and B are known to differ from the true system matrices $\bar{A}$ and $\bar{B}$ within

$$\|A-\bar{A}\|_2 \leqslant \text{ATOL} \cdot \|A\|_2, \quad \|B-\bar{B}\|_2 \leqslant \text{BTOL} \cdot \|B\|_2,$$

and the computed result $\mu(A, B)$ is exact for a system as in (26); then if

$$\mu(A, B) > (\text{ATOL}+\text{CTOL})\|A\|_2 + (\text{BTOL}+\text{CTOL})\|B\|_2,$$

the designer would be confident that the system is controllable. If this is not the case, there is the possibility that the system could be uncontrollable.

The measure $\mu(A, B)$ is obtained by allowing all possible perturbations, but if the above bounds on model uncertainties are dominated by the uncertainties in just a few elements, then the above criterion is liable to be a pessimistic one—large changes in these few elements may not even alter the controllability of the system. For this problem, a good scaling then appears to be to scale so that as far as possible the uncertainties in the elements of A and B are all of the same order of magnitude. This follows the recommendation in [9, p. I.11] for scaling in solution of linear equations. Of course, accurately known elements such as 0 or 1 are ignored in such scaling.

It is salutary to note that of the methods for computing controllability that have been mentioned here, only the new one in Section VI gave a useful measure of $\mu(A, B)$ in its application to the uncontrollable system (23). For controllable systems, no way of obtaining such a measure has been hinted at here. Related theoretical ideas are developed in [6], but the development of a practical computational algorithm still appears to be an open question. One possible advantage of the reduction in Section VI is that it may prove useful as a preprocessor for a numerically stable way of computing a measure $\mu(A, B)$.

IX. Conclusion

The preceding contents have been somewhat destructive, setting up computations for determining controllability, and then showing some of their drawbacks. On top of this, the previous section emphasized the weakness of any such approach which seeks only to determine if a system is controllable or not. Nevertheless, the reader has been taken on a tour of some important numerical ideas. The paper concluded by drawing attention to a basic, but apparently unresolved, problem in controllability—that of finding a reliable and efficient numerical algorithm for determining the distance of the closest unconrollable system from the given one.

From a tutorial viewpoint, it is useful to reemphasize two clear distinctions that must be made in understanding the numerical properties of algorithms. The first is the distinction between the sensitivity of a problem, and the numerical stability of any algorithm applied to the problem. The sensitivity of a problem is a mathematical property of the problem which is independent of which algorithm is used to solve the problem. On the other hand, if an algorithm is numerically stable, it is ideally numerically stable no matter what the sensitivity of the problem to which it is applied. Much of the power of Wilkinson's reverse error analysis of algorithms comes from clearly separating these two aspects. A successful reverse error analysis of a numerically stable algorithm will give bounds on equivalent perturbations in the initial data, and these bounds will be independent of the sensitivity of the problem. If it is then necessary to see what effect such equivalent perturbations could have on the computed solution, the sensitivity of the problem must be considered.

The other distinction is between the scaling of a problem and the numerical stability of an algorithm for its solution. There are two distinct problems here that have to be solved in order to give useful results. First a well-scaled model has to be formulated, and then a numerically stable algorithm has to be used to solve the problem. A numerically stable algorithm on a computer of a given precision will give as good a result as is possible for the data it is given; however, the results will only be physically meaningful if the scaling chosen was itself meaningful. Thus, the scaling and choice of axes is the crucial front end and should depend on the physical properties of the system. The numerical algorithm comes later and depends on the mathematical characteristics of the whole class of problems.

A connection between these two distinctions is that the choice of scaling can alter the condition of the mathematical problem. There is little point in choosing a scaling on the basis of making the resulting mathematical problem

well conditioned; it is far more important that the scaling and choice of axes be such to ensure that the mathematical problem accurately reflects the sensitivity of the physical problem.

The original version of this paper [26] was written without the benefit of computed results, and partly because of this, the weakness of C2 as exhibited in Table I was not detected. A referee thoughtfully pointed this out, thereby initiating the present computations.

Acknowledgment

I am very grateful for the time and guidance given me by M. Denham of Kingston Polytechnic, and by D. Mayne, P. Antsaklis, and R. Vintner at Imperial College where this work was originally done. A. Laub and the referees gave a great many thoughtful and helpful comments which were invaluable in the rewrite. G. Miminis and B. Bhattacharya carried out the computations mentioned here.

References

[1] R. E. Kalman, "On the general theory of control systems," in *Proc. 1st IFAC Congr.*, vol. 1. London: Butterworth, 1960, pp. 481–491.
[2] M. L. J. Hautus, "Controllability and observability conditions of linear autonomous systems," in *Proc. Kon. Ned. Akad. Wetensh. Ser. A.*, vol. 72, 1969, pp. 443–448.
[3] H. H. Rosenbrock, *State-Space and Multivariable Theory*. London: Nelson, 1970.
[4] E. J. Davison, W. Gesing, and S. H. Wang, "An algorithm for obtaining the minimal realization of a linear time-invariant system and determining if a system is stabilizable-detectable," *IEEE Trans. Automat. Contr.*, vol. AC-23, pp. 1048–1054, Dec. 1978.
[5] B. C. Moore, "Computational problems with modal analysis," presented at the 16th Annu. Allerton Conf. Commun., Contr., Comput., Oct. 1978.
[6] ——, "Singular value analysis of linear systems," in *Proc. 1978 IEEE Conf. Decision Contr.*, San Diego, CA, Jan. 1979, pp. 66–73.
[7] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *J. SIAM Numer. Anal.*, ser. B., vol. 2, pp. 205–224, 1965.
[8] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, pp. 403–420, 1970.
[9] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK User's Guide*. Philadelphia, PA: SIAM 1979.
[10] G. W. Stewart, *Introduction to Matrix Computations*. New York: Academic, 1973.
[11] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
[12] N. Anderson and I. Karasalo, "On computing bounds for the least singular value of a triangular matrix," *BIT*, vol. 15, pp. 1–4, 1975.
[13] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, "An estimate for the condition number of a matrix," *SIAM J. Numer. Anal.*, vol. 16, pp. 368–375, 1979.
[14] G. H. Golub, "Numerical methods for solving linear least squares problems," *Numer. Math.*, vol. 7, pp. 206–216, 1965.
[15] P. G. Kaminski, "Square root filtering and smoothing for discrete processes," Ph.D. dissertation, Dep. Elec. Eng., Stanford Univ., Stanford, CA, 1971.
[16] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. London: Oxford Univ. Press, 1965.
[17] J. H. Wilkinson and C. Reinsch, *Handbook for Automatic Computation, Vol. II. Linear Algebra*. Berlin: Springer-Verlag, 1971.
[18] B. T. Smith *et al.*, *Matrix Eigensystem Routines—EISPACK Guide, 2nd ed., Lecture Notes in Comput. Sci.*, vol. 6. New York: Springer-Verlag, 1976.
[19] D. G. Luenberger, *Introduction to Dynamic Systems*. New York: Wiley, 1979.
[20] J. H. Wilkinson, "Linear differential equations and Kronecker's canonical form," in *Recent Advances in Numerical Analysis*, C. de Boor and G. H. Golub, Eds. New York: Academic, 1978.
[21] ——, "Kronecker's canonical form and the *QZ* algorithm," Nat. Phys. Lab., Teddington, Middlesex, England, Rep. DNACS 10/78, Nov. 1978.
[22] P. Van Dooren, "The computation of Kronecker's canonical form of a singular pencil," *Linear Algebra and Its Appl.*, vol. 27, pp. 103–140, 1979.
[23] ——, "The generalized eigenstructure problem applications in linear system theory," Ph.D. dissertation, Katholieke Univ. Leuven, Leuven, Belgium, May 1979.
[24] E. C. Y. Tse, J. V. Medanic, and W. R. Perkins, "Generalized Hessenberg transformations for reduced-order modelling of large scale systems," *Int. J. Contr.*, vol. 27, no. 4, pp. 493–512, 1978.
[25] V. C. Klema and A. J. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Trans. Automat. Contr.*, vol. AC-25, Apr. 1980.
[26] C. C. Paige, "Computing controllability and observability of time-invariant linear systems," Imperial Coll., London, England, Comput. and Contr. Rep. 79/13, Mar. 1979.

Chris C. Paige was born in Sydney, Australia, in 1939. He received the B.Sc. (mathematics and physics) degree in 1960 and the B.E. (electrical engineering) degree in 1962, both from Sydney University, and an Academic Postgraduate Diploma in numerical analysis in 1965 and the Ph.D. degree in computer science in 1971 from the University of London, London, England.

He worked for the Australian Department of Supply from 1962 to 1964 where he was involved in the mathematical and computer modeling of systems, and was a lecturer at London University Institute of Computer Science from 1967 to 1972. Since 1972 he has been with the McGill University School of Computer Science, Montreal, P.Q., Canada, where he is now an Associate Professor. In 1978–1979 he spent a sabbatic year with the Department of Mathematics, Imperial College, London, and has spent summers with the Departments of Computer Science and Operations Research at Stanford University, Stanford, CA, and with the Department of Scientific and Industrial Research, Wellington, New Zealand. His main research interests are related to the design and analysis of effective numerical algorithms and the application of such algorithms in engineering and statistics.