Notes on Pseudoinverses

Application to Identification

Söderström, Torsten

1970

Document Version:
Publisher's PDF, also known as Version of record

NOTES ON PSEUDOINVERSES. APPLICATION TO IDENTIFICATION.

T. SÖDERSTRÖM

REPORT 7003 JULI 1970
LUND INSTITUTE OF TECHNOLOGY
DIVISION OF AUTOMATIC CONTROL

NOTES ON PSEUDOINVERSES. APPLICATION TO IDENTIFICATION.

T. Söderström

ABSTRACT

In this report we will discuss possibilities of defining a "good solution" of a system of linear equations

$$Ax = b$$

where A is not necessarily quadratic. Especially the least squares problem will be treated. The concepts of pseudoinverses are introduced and its application to the least squares problem is shown. Geometrical interpretation of the solution in different cases is given. Different algorithms for computing the pseudoinverse are briefly discussed. One is programmed and compared with another program, given by Golub-Reinsch. Numerical examples are given.

The concepts of least squares problem are applied to identification of a linear time-invariant, discrete, single input, single output system. Some recursive equations are given. The consequences of using a model of too high an order for systems with and systems without noise are discussed.

The results in chapter 10 and the last half of chapter 11 are believed to be new.

TABLE OF CONTENTS

1. INTRODUCTION.

In this chapter we will consider systems of linear equations, which may not have a solution in the ordinary sense. Some possibilities of defining a "best" solution will be discussed.

Let us consider the system

$$Ax = b \tag{1.1}$$

where A is an m × n matrix, x an unknown n vector to be determined, and b is an m vector.

In identification problems one usually has an overdetermined system, i.e. m is greater than or equal to n, but we will not limit ourselves to that case now. If i) m = n and ii) A^{-1} exists, then it is known, that the (exact) solution, which is unique, exists and is $x = A^{-1}b$.

When A^{-1} does not exist, the least squares solution, introduced already by Gauss, is very often used. This means that we search an x, such that

$$|| Ax - b ||^2 \tag{1.2}$$

is minimized. In this report we mostly use the Frobenius norm,

$$|| A ||_E = \sqrt{\text{tr } AA^T} = \sqrt{\text{tr } A^TA} = \sqrt{\sum_{i,k} a_{ik}^2}$$

This norm coincides with the usual Euclidian norm in R^n, when applied to vectors. It is, however, not in general equal to the operator norm, defined by

$$|| A ||_2 = \sup_{|| x ||=1} || Ax ||$$

with || x || as the Euclidian norm of x.

When A^{-1} exists, we will get $x = A^{-1}b$, which is the solution in the ordinary sense to (1.1). When rank A = n (or equivalently $(A^TA)^{-1}$

exists), it is shown in |28| by completing the squares, that the least squares solution is

$$x = (A^TA)^{-1}A^Tb$$

When rank A is less than n, there is no unique x, which minimizes (1.2). Further conditions on x must be added. We will already now point out, that this case should be handled with care. This case means, that the number of linear independent equations is less than the number of unknown ones.

A usual way to handle this case is to add the condition, that among all x, which minimize (1.2), we search for that one of least norm, "the least squares solution of minimum length". In chapter 3 we will introduce the pseudoinverse $A^\dagger$ of a matrix A. In chapter 5 it will be shown that $x = A^\dagger b$ is the solution of the least squares problem in the sense just described.

Another condition is given in |24|, where Rosen defines x_b as a basic approximate solution to (1.1) by i) x_b minimizes (1.2), ii) x_b has a most r nonzero component, with r = rank A. This solution is not unique.

A generalized least squares solution to (1.1) is given in |13|. For this solution it is assumed that rank A = n. The problem, previously discussed, can be regarded as follows. Disturb the vector b to b + Δb, so that an exact solution exists. Of all possible Δb determine the one, for which $|| \Delta b ||^2$ is minimized. The solution to Ax = b + Δb is then the least squares solution. In the generalized least squares solution also A is allowed to be disturbed. The problem is now the following: Disturb A and b so that (A + ΔA)x = (b + Δb) has a solution. Of all possible ΔA and Δb, choose those which satisfy $K|| \Delta A ||^2 + || \Delta b ||^2$ is minimum, where K is a given weight. This problem is solved in |13|, which also gives ALGOL programs, which can be used for computing the solution.

Björck |7| considers least squares problems with linear constraints. The problem is to determine a vector x, which satisfy $A_1x = b_1$ exactly, and such that $|| A_2x - b_2 ||^2$ is minimum. The solution of this problem and ALGOL programs are given in |7|.

2. STATEMENT OF THE PROBLEM.

Now we give the exact statement of the problem after the introducing discussion.

Given the system of linear equations $Ax = b$, we search for the $x = x_0$, which satisfies:

i) $\| Ax_0 - b \|^2 < \| Ax - b \|^2 \quad \forall x \neq x_0$

or

ii) $\| Ax_0 - b \|^2 \leq \| Ax - b \|^2 \quad \forall x$ and

$\| x_0 \| < \| x \| \quad \forall x; \quad \| Ax_0 - b \|^2 = \| Ax - b \|^2, \quad x \neq x_0$

The solution of this problem exists and is given in chapter 5.

3. PSEUDOINVERSE. DEFINITION AND BASIC PROPERTIES.

In this chapter we want to introduce the pseudoinverse of a matrix. Before starting with the definition we give some concepts of linear transformations.

Let A be an m × n matrix. In the following we will not distinguish between the matrix A and the corresponding linear transformation A: $R^n \to R^m$, which maps the vectors of R^n into R^m.

We start with

Definition 3.1

The nullspace of A, written N(A), is the set of vectors in R^n, transformed by A to the origin in R^m or $N(A) = \{x|Ax = 0\}$.

Definition 3.2

The range of A, written R(A), is the subset of R^n, described by $R(A) = \{y|\exists x;\ y = Ax\}$. Loosely speaking, R(A) consists of all vectors y = Ax, which are obtained when x is varied in R^n.

Definition 3.3

Rank A = dim R(A).

We also use the following partition of R^n and R^m

$$R^n = N(A) \oplus N(A)^{\perp} \tag{3.1}$$

$$R^m = R(A) \oplus R(A)^{\perp} \tag{3.2}$$

$M^{\perp}$ means the orthogonal complement of the set M. $\oplus$ means direct sum. In several sources, e.g. |27|, it is shown that

$$N(A)^{\perp} = R(A^T) \tag{3.3}$$

$$N(A^T) = R(A)^{\perp} \tag{3.4}$$

We have also (see e.g. |27|):

Theorem 3.1

A is a one-one mapping

$$R(A^T) \rightarrow R(A) \tag{3.5}$$

Now, let us turn to the pseudoinverse. There are several ways of defining the pseudoinverse of a matrix. Here we have chosen to follow Zadeh-Desoer |27|.

Definition 3.4

$A^\dagger$ is the pseudoinverse of A if

$$\text{i)} \quad A^\dagger A x = x \qquad \forall x \in R(A^T) \tag{3.6}$$

$$\text{ii)} \quad A^\dagger z = 0 \qquad \forall z \in N(A^T) \tag{3.7}$$

$$\text{iii)} \quad A^\dagger (y+z) = A^\dagger y + A^\dagger z \qquad \forall y \in R(A),\ \forall z \in N(A^T) \tag{3.8}$$

The following picture is instructive to explain the pseudoinverse.

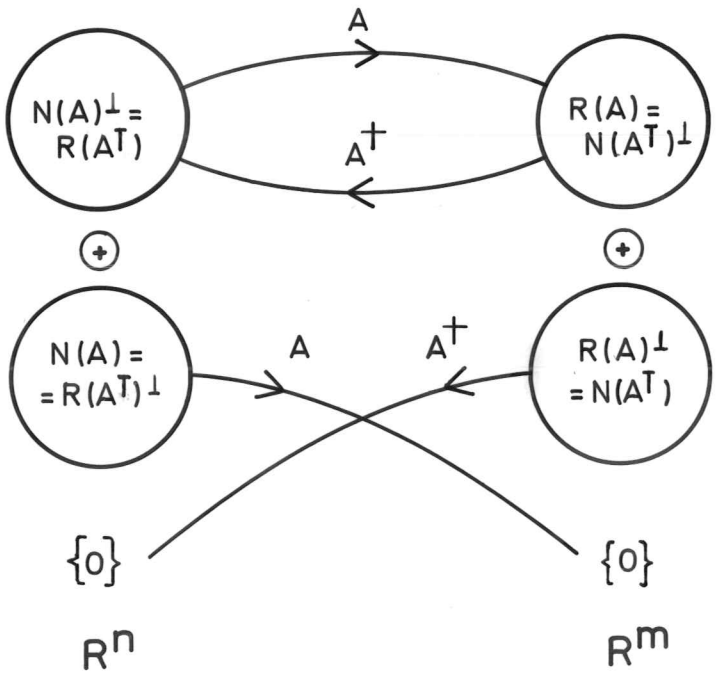

Fig. 3.1

When dim N(A) and dim $N(A^T)$ are not both equal to zero, the ordinary inverse A^{-1} does not exist. As pointed out in (3.5) the restriction of A to A: $R(A^T) \to R(A)$ is a one-one mapping. A good substitution to inverse ought to map Ax back to x, when $x \in R(A^T)$. The pseudoinverse has this property.

Penrose has an alternative definition which could be shown to be equivalent to def. 3.4.

Definition 3.4'

$A^\dagger$ is the pseudoinverse of A if $A^\dagger$ satisfies

i) $AA^\dagger A = A$ (3.9)

ii) $A^\dagger AA^\dagger = A^\dagger$ (3.10)

iii) $AA^\dagger$ symmetric (3.11)

iv) $A^\dagger A$ symmetric (3.12)

This definition requires a proof to show that $A^\dagger$ always exists and is unique. For some purposes, e.g. making proofs, def. 3.4' may be as good as def. 3.4.

Other ways of defining $A^\dagger$ can be found in |4|, |10|, |15|, |16|. For example the solution of the problem stated in chapter 2 can be used for making a definition.

Some simple properties of the pseudoinverse can be found e.g. in |19|, |27|.

Corr

$A^\dagger$ is a linear transformation.

Corr

$R(A^\dagger) = R(A^T)$, $N(A^\dagger) = N(A^T)$

Especially we have $A^\dagger$ is $n \times m$, if A is $m \times n$.

Theorem 3.2

i) $A^{\dagger}A$ is the orthogonal projection of R^n on $R(A^T)$ (3.13)

ii) $AA^{\dagger}$ is the orthogonal projection of R^m on $R(A)$ (3.14)

iii) $(A^{\dagger})^{\dagger} = A$ (3.15)

iv) $AA^{\dagger}A = A$ (3.16)

v) $A^{\dagger}AA^{\dagger} = A^{\dagger}$ (3.17)

Theorem 3.3

$$(A^T)^{\dagger} = (A^{\dagger})^T \tag{3.18}$$

Theorem 3.4

$$A^{\dagger} = A^{-1}, \text{ if } A^{-1} \text{ exists} \tag{3.19}$$

Theorem 3.5

Def. 3.4' is equivalent to

$$\begin{cases} A^{\dagger}AA^T = A^T & (3.20) \\ AA^{\dagger}(A^{\dagger})^T = (A^{\dagger})^T & (3.21) \end{cases}$$

Most of the results can easily be carried out from definition 1 or from figure 3.1. Strict proofs are found in |19| and |27|.

We conclude this chapter by pointing out that the transformation $A \to A^{\dagger}$ is discontinuous. Let

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{bmatrix}$$

Then A^{-1} exists and

$$A^{\dagger} = A^{-1} = \frac{1}{\varepsilon}\begin{bmatrix} 1+\varepsilon & -1 \\ -1 & 1 \end{bmatrix}$$

However, if $\varepsilon = 0$ we get

$$A^{\dagger} = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}$$

which cannot be obtained as a limit of the previous result.

This, in some cases "bad", property depends on the change in rank of A.

If we restrict to matrices of a fixed type and fixed rank, then the pseudoinverse is continuous.

Note, that this may cause numerical difficulties, since two matrices may have different rank but still be very close to another in norm. The determination of the correct rank is in fact the critical point in every algorithm for computing the pseudoinverse.

4. FURTHER PROPERTIES OF THE PSEUDOINVERSE.

In this chapter we will give some more formulas concerning pseudo-inverses. Some references containing rather difficult, but interesting, expressions will be mentioned.

As mentioned in the introduction in the special case rank A = n we have

$$A^{\dagger} = (A^{T}A)^{-1}A^{T} \tag{4.1}$$

which is easily verified. More generally we have (see |19|)

Theorem 4.1

$$\begin{cases} A^{\dagger} = (A^{T}A)^{\dagger}A^{T} & (4.2) \\ A^{\dagger} = A^{T}(AA^{T})^{\dagger} & (4.3) \end{cases}$$

In |19| Kalman and Englar give a recursive formula for the pseudoinverse. Other references of this topic are |8|, |16|. The result in |19| is the following:

Let A be an m × n matrix with a known pseudoinverse $A^{\dagger}$. Add a column vector a of dimension m to form the m × (n+1) matrix $[A \mid a]$. Express the pseudoinverse $[A \mid a]^{\dagger}$ making use of $A^{\dagger}$.

We will have to separate two cases:

i) if $a \notin R(A) \iff \text{rank } [A \mid a] > \text{rank } [A] \iff (I - AA^{\dagger})a \neq 0$

$$[A \mid a]^{\dagger} = \begin{bmatrix} A^{\dagger}\left(I - \dfrac{aa^{T}(I-AA^{\dagger})}{a^{T}(I-AA^{\dagger})a}\right) \\ - - - - - - - - - \\ \dfrac{a^{T}(I-AA^{\dagger})}{a^{T}(I-AA^{\dagger})a} \end{bmatrix} \tag{4.4}$$

ii) if $a \in R(A) \iff (I - AA^{\dagger})a = 0$

$$[A \mid a]^{\dagger} = \begin{bmatrix} A^{\dagger}\left(I - \dfrac{aa^{T}(A^{\dagger})^{T}A^{\dagger}}{1+a^{T}A^{\dagger T}A^{\dagger}a}\right) \\ \hline \dfrac{a^{T}A^{\dagger T}A^{\dagger}}{1+a^{T}A^{\dagger T}A^{\dagger}a} \end{bmatrix} \tag{4.5}$$

In |8| Cline goes further and studies the pseudoinverse of a partioned matrix, i.e. $A = [U \mid V]$. He also gives some formulas for $U^{\dagger}$ in terms of submatrices of $A^{\dagger}$.

In |9| Cline gives representations for the pseudoinverse of sums of matrices.

Greville |17| has given conditions on A and B so that

$$(AB)^{\dagger} = B^{\dagger}A^{\dagger}$$

This equality is not always true. For example $A = |1 \quad 0|$, $B = \begin{vmatrix} 1 \\ 1 \end{vmatrix}$ implies $(AB)^{\dagger} = 1$ and $B^{\dagger}A^{\dagger} = 0.5$.

Some general theory of the pseudoinverse can be found in |2|.

Note, that if U and V are orthogonal matrices then

$$(UAV)^{\dagger} = V^{\dagger}A^{\dagger}U^{\dagger} = V^{T}A^{\dagger}U^{T} \tag{4.6}$$

This fact is used in several algorithms for computing the pseudoinverse as mentioned in chapter 7.

5. SOLUTION OF THE PROBLEM.

In this chapter we want to show that the use of pseudoinverse will give the solution of the problem stated in chapter 2. We will also give some results about the modified problem and the solution in the ordinary sense of $Ax = b$ when it exists.

We start with

Theorem 5.1

Let $x_0 = A^{\dagger}b$. Then

i) $\quad || Ax_0 - b ||^2 < || Ax - b ||^2 \qquad \forall x \neq x_0$

or

ii) $\quad || Ax_0 - b ||^2 \leq || Ax - b ||^2 \qquad \forall x \neq x_0$ and

$$|| x_0 || < || x || \qquad \forall x \neq x_0;\ || Ax_0 - b || = || Ax - b ||$$

The proof of this theorem is well-known (see e.g. |19|, |27|). We give a proof of the theorem here on account of its central part in this report.

Proof

Let the vector b be decomposed $b = b_1 + b_2$ where b_1 is in $N(A^T)$ and b_2 is in $R(A)$. Let the arbitrary x be written as $x = x_0 + x_1 + x_2$, where x_1 is an arbitrary vector in $N(A)$ and x_2 an arbitrary vector in $R(A^T)$.

We have

$$|| Ax - b ||^2 = || Ax_0 + Ax_1 + Ax_2 - b_1 - b_2 ||^2 =$$

$$= || Ax_0 + Ax_2 - b_2 ||^2 + || b_1 ||^2$$

since $Ax_1 = 0$, and further $N(A^T)$ and $R(A)$ are orthogonal complements.

We also have

$$Ax_0 = AA^{\dagger}b = AA^{\dagger}(b_1 + b_2) = AA^{\dagger}b_2 = b_2$$

(cf. (3.7), (3.14)). Hence

$$|| Ax - b ||^2 = || Ax_2 ||^2 + || b_1 ||^2$$

We can conclude that the minimum of

$$|| Ax - b ||^2 = || b_1 ||^2$$

and is obtained with $x_2 = 0$, x_1 arbitrary.

Let us now separate two cases.

i) rank A = n (A is as before an m × n matrix). This implies dim N(A) = 0 and $x_1 = 0$. Thus we have a unique minimum.

ii) rank A < n, and dim N(A) > 0. $|| Ax - b ||^2$ is minimized by all $x = x_0 + x_1$. The first part of ii) is proved. The second relation follows from

$$|| x ||^2 = || x_0 ||^2 + || x_1 ||^2$$

and hence $|| x || > || x_0 ||$ if $x \neq x_0$. Q.E.D.

The problem can be modified to minimize $|| Ax - b ||_P^2$, where P is a positive definite matrix. ($|| y^2 ||_P$ means $y^T Py$.) Since P can be decomposed as $P = P^{1/2}P^{1/2}$, $P^{1/2}$ symmetric we can write

$$|| Ax - b ||_P^2 = || P^{1/2}Ax - P^{1/2}b ||^2$$

The result from theorem 5.1 is then easily modified and the corresponding least squares solution is

$$x_0 = (P^{1/2}A)^{\dagger}P^{1/2}b$$

It can be shown, see |19|, that

$$x_0 = (A^T P A)^\dagger A^T P b$$

In a computation $P^{1/2}$ must not necessarily be computed.

In |19| Kalman and Englar show some further results, e.g. minimizing

$$\sum_{i=1}^{m} || A_i x - b_i ||^2_{P_i}$$

It is also interesting to know when the system Ax = b has any solution in the ordinary sense, i.e. when exists a vector x, satisfying Ax = b exactly. The answer is given by the following theorem. Although the proof is well-known it is thought valuable to include.

Theorem 5.2

Consider the system Ax = b, where A is an m × n matrix.

i) If $b \in R(A)$ then

$$x = A^\dagger b + (I - A^\dagger A)z$$

z arbitrary vector in R^n is the general solution (in the ordinary sense).

ii) If $b \notin R(A)$ there is no exact solution (in the ordinary sense).

Proof

i) The given x is a solution for

$$Ax = AA^\dagger b + A(I - A^\dagger A)z = b$$

where (3.9) and (3.14) are used. With figure 1 in mind it is easy to be convinced that the general solution must be of the form

$$x = A^{\dagger}b + x_1$$

where x_1 is an arbitrary vector in N(A). Now instead of x_1 we take $z \in R^n$ arbitrary and project z on to N(A). Then (3.13) gives the relation

$$x_1 = (I - A^{\dagger}A)z$$

ii) As

$$\min || Ax - b ||^2 = || b_1 ||^2 \neq 0$$

with notations from theorem 5.1 there cannot be any solution in the ordinary sense.

Q.E.D.

Remark

The criterion $b \in R(A)$ is indeed easy to understand. Ax = b has solution is just equivalent to that there is at least some x satisfying this relation. By definition of R(A) this is always true if $b \in R(A)$, and always false if $b \notin R(A)$.

6. GEOMETRICAL INTERPRETATION.

Now we will point out the geometrical interpretation of theorem 5.1. We will also show how the result of this theorem works in some figures illustrating simple examples.

Consider again the system of equations $Ax = b$ and define $x_0 = A^{\dagger}b$.

x_0 is obtained as:

i) Make an orthogonal projection of b on to R(A). This projection is Ax_0.

ii) Take the shortest vector x, that satisfies Ax = this projection. We once again note that step i) is a very natural one when trying to get some "solution". We know from theorem 5.2 that on exact solution requires $b \in R(A)$. In the least squares sense step i) is the best way we have to produce such an $b \in R(A)$.

Now, let us turn to the examples.

Example 1

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$m = n = 2$, A^{-1} exists and we get $x = A^{-1}b$ or

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

The graphic solution to this example is shown in fig. 6.1

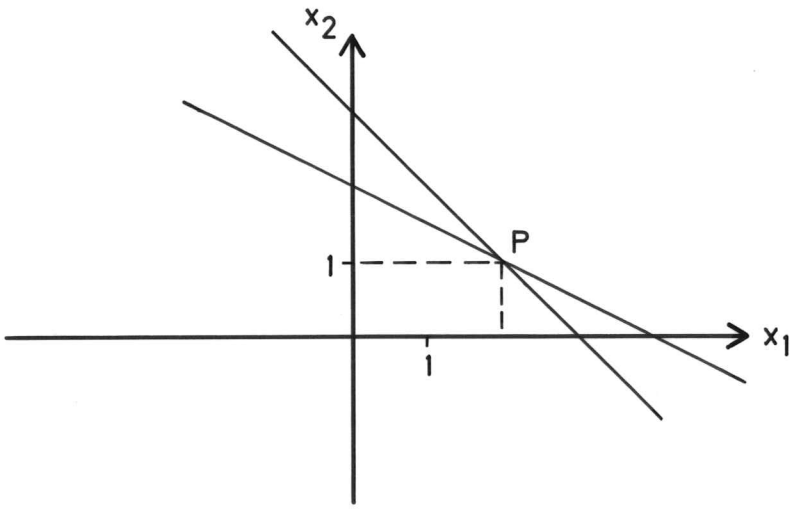

Fig. 6.1 - Graphic solution of example 1.
P is the obtained solution.

Example 2

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

$m = 3$, $n = 2$. We have an overdetermined system with rank $A = n = 2$. Then we use

$$A^{\dagger} = (A^T A)^{-1} A^T$$

to get the result

$$x = A^{\dagger} b = \frac{1}{11} \begin{bmatrix} 7 \\ 7 \end{bmatrix}$$

Graphic solution in fig. 6.2.

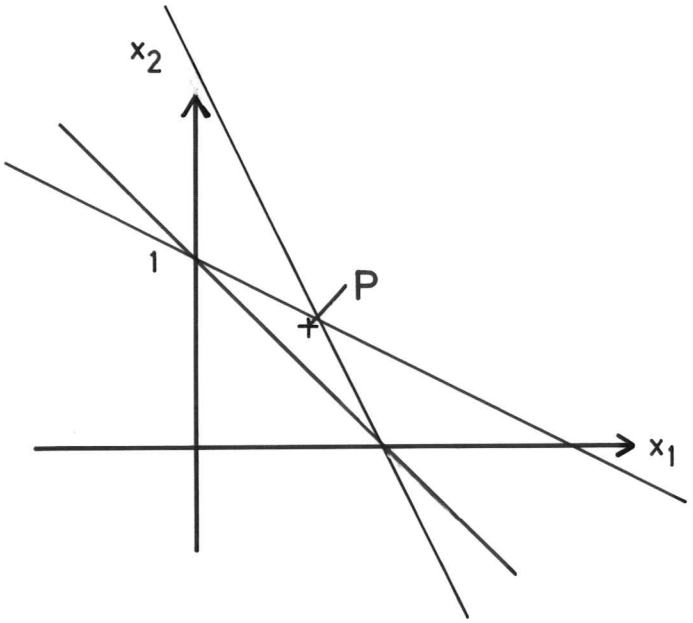

Fig. 6.2 - Graphic solution of example 2.
P is the obtained least squares solution.

The solution has the property that the sum of the squares of the distances to the equation lines is minimum.

Example 3

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2$$

$m = 1$, $n = 2$. We have too few equations to get an exact solution. The formula

$$A^{\dagger} = A^{T}(AA^{T})^{-1}$$

gives the least squares solution

$$x = A^{\dagger}b$$

or

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Applying theorem 5.2 we get the general solution

$$x = A^{\dagger}b + (I - A^{\dagger}A)z = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} z$$

Shifting z to 2z we get

$$\begin{cases} x_1 = 1 + [\,1 \quad -1\,]z \\ x_2 = 1 - [\,1 \quad -1\,]z \end{cases}$$

The graphic solution is given in figure 6.3.

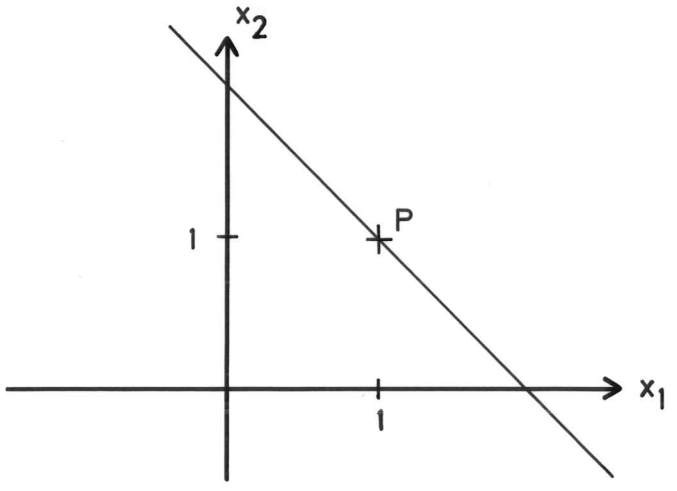

Fig. 6.3 - Graphic solution of example 3.
P is the least squares solution.

The exact solution we obtained is just the whole line, and the point of least norm on the line is P.

Example 4

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$m = 2$, $n = 2$. Here rank $A = 1$, and the two equations are linear dependent. The least squares solution is

$$x = A^{\dagger} b$$

or

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$

Since $b \notin R(A)$ there is no exact solution.

The graphic solution is shown in figure 6.4.

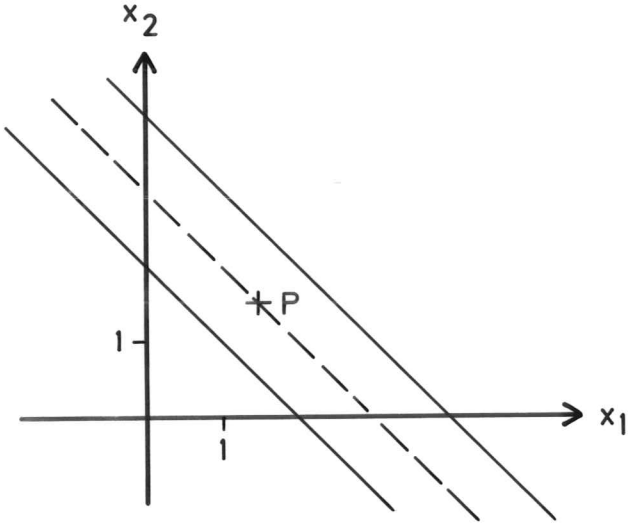

Fig. 6.4 - Graphic solution of example 4.
P is the obtained least squares solution.

To get the correct x we first minimize $|| Ax - b ||^2$ and get the whole dotted line. (No unique minimum since rank A < n.) The point nearest to the origin on that line is P. This example illustrates case ii) of theorem 5.1.

In connection with the last example we return to the fact that $A^\dagger$ is not continuous. Consider the system

$$\begin{bmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

with the solution (in the ordinary sense)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{\varepsilon} \begin{bmatrix} -2+2\varepsilon \\ 2 \end{bmatrix}$$

The solution is graphically shown in fig. 6.5.

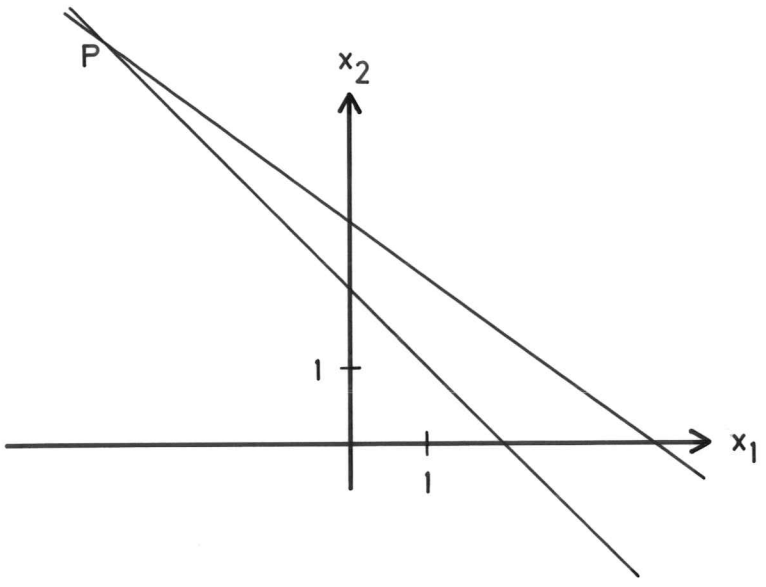

Fig. 6.5 - Graphic solution of

$$\begin{bmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

P is the obtained solution.

When $\varepsilon \to 0$ the lines become parallel and P goes to infinity, which does not coincide with the previous solution

$$\begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$

In this case it seems that the pseudoinverse means: the intersection between two parallel lines is the line halfway between them.

With risk of being tedious we note again that a change of rank A, which may easily be done numerically, can change the result drastically. Further it is not sure that the second condition (minimizing $|| x ||$) has any physical meaning in an actual problem.

7. ALGORITHMES.

Several algorithmes for computing the pseudoinverse of a matrix have been developed. For the computation of numerical examples we have chosen two algorithmes, given by Mayne in |21| and by Golub - Reinsch |13|. The reason for this choice is mainly that they seem to be simple and straightforward. The first one is, however, not without disadvantages, as pointed out in |13|. In the last part of this chapter we briefly discuss some other algorithmes.

In |21| Mayne considers an $m \times n$ matrix A with $m \leq n$. The algorithm can be outlined as follows:

i) By elementary row operations, or equivalently by premultiplying A with a square matrix P, a new matrix A_1 of type $q \times n$ is received.

$$PA = \begin{bmatrix} A_1 \\ \hline 0 \end{bmatrix}$$

The rows of A_1 should be linear independent.

ii) Compute $C = A_1 A^T$ (C is a $q \times m$ matrix)

iii) Compute $A^\dagger = A_1^T (CC^T)^{-1} C$ ($A^\dagger$ is an $n \times m$ matrix). Then $A^\dagger$ is the pseudoinverse of A.

In |21| Mayne makes a proof of the algorithm. First he shows that CC^T is invertible and then that $A^\dagger$ fulfils def. 4 of chapter 3. Here we give an alternative proof, which gives some more understanding of the geometrical meaning of the different matrices involved.

Consider the system $Ax = b$. We want $x_0 = A^\dagger b$. Then we will i) project b on R(A) and get b_1, ii) take x_0 (uniquely determined) in $R(A^T)$, satisfying $Ax_0 = b_1$.

The rows of A span the set $R(A^T)$. Let us pick up linearly independent row vectors of A, and take as many as possible. The number of them will be $q = \text{rank } A$. We call the result $e_1^T, \ldots, e_q^T$, and we form the matrix

$$A_1 = \begin{bmatrix} e_1^T \\ \vdots \\ e_q^T \end{bmatrix} \tag{7.1}$$

Now $f_i = Ae_i$, $i = 1, \ldots, q$, are a base for $R(A)$, since $A:R(A^T) \to R(A)$ is a one-one mapping (theorem 3.1). Introduce the matrix

$$C = \begin{bmatrix} f_1^T \\ \vdots \\ f_q^T \end{bmatrix} \tag{7.2}$$

The relation $f_i = Ae_i$ gives us

$$C^T = AA_1^T \quad \text{or} \quad C = A_1A^T \tag{7.3}$$

Introduce the vectors $f_{q+1}, \ldots, f_m$ as a base in $R(A)^\perp = N(A^T)$. We can now write b with components in R^m, with the use of the base $f_1, \ldots, f_m$.

$$b = \sum_{i=1}^{q} d_i f_i + \sum_{i=q+1}^{m} d_i f_i \tag{7.4}$$

The first sum is just b_1. Now the components $d_1, \ldots, d_q$ are wanted. Let us form the inner products

$$\langle f_j|b\rangle = \sum_{i=1}^{q} d_i \langle f_j|f_i\rangle \quad \text{with} \quad 1 \leq j \leq q$$

We get the following system of linear equations:

$$\begin{vmatrix} \langle f_1|f_1\rangle \ldots\ldots \langle f_1|f_q\rangle \\ \vdots \qquad\qquad \vdots \\ \langle f_q|f_1\rangle \ldots\ldots \langle f_q|f_q\rangle \end{vmatrix} \begin{vmatrix} d_1 \\ \vdots \\ d_q \end{vmatrix} = \begin{vmatrix} \langle f_1|b\rangle \\ \vdots \\ \langle f_q|b\rangle \end{vmatrix} \tag{7.5}$$

which we equivalently write:

$$Fd = B \tag{7.6}$$

We have

$$F = \begin{bmatrix} f_1^T \\ \vdots \\ f_q^T \end{bmatrix} \begin{bmatrix} f_1 & \cdots & f_q \end{bmatrix} = CC^T \tag{7.7}$$

$$B = \begin{bmatrix} f_1^T \\ \vdots \\ f_q^T \end{bmatrix} \begin{bmatrix} b \end{bmatrix} = Cb \tag{7.8}$$

Rank C = q implies rank F = q and we conclude that F is a positive definite matrix and hence F^{-1} exists. The solution of (7.6) then is:

$$d = F^{-1}B = (CC^T)^{-1}Cb \tag{7.9}$$

Now assume

$$x_0 = \sum_{i=1}^{q} x_i e_i \tag{7.10}$$

We search for the components $x_1, \ldots, x_q$. The relation $Ax_0 = b_1$ implies

$$Ax_0 = A \sum_{i=1}^{q} x_i e_i = \sum_{i=1}^{q} x_i Ae_i = \sum_{i=1}^{q} x_i f_i = b_1 = \sum_{i=1}^{q} d_i f_i$$

We then have

$$x_i = d_i \qquad i = 1, \ldots, q \tag{7.11}$$

The solution of the least squares problem is then given by:

$$x_0 = \Sigma\, d_i e_i = \begin{bmatrix} e_1 & \cdots & e_q \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_q \end{bmatrix} = A_1^T d = A_1^T (CC^T)^{-1} Cb \tag{7.12}$$

Finally $x_0 = A^\dagger b$ combined with the fact that (7.12) holds for all b gives

$$A^\dagger = A_1^T (CC^T)^{-1} C \tag{7.13}$$

The other way of calculating the pseudoinverse in the numerical examples is the following, described in |13|. To a given matrix $m \times n$ matrix A with $m \geq n$ there exists U (of type $m \times n$) and V (of type $n \times n$) satisfying

$$A = U \Sigma V^T \tag{7.14}$$

$$U^T U = I_n \tag{7.15}$$

$$V V^T = I_n \tag{7.16}$$

$$\Sigma = \operatorname{diag} (\sigma_1 \ \cdots \ \sigma_n) \tag{7.17}$$

The pseudoinverse then is

$$A^\dagger = V \Sigma^\dagger U^T \tag{7.18}$$

with

$$\Sigma^\dagger = \operatorname{diag} (\sigma_1^\dagger, \ \ldots, \ \sigma_n^\dagger) \tag{7.19}$$

$$\sigma_i^{\dagger} = \begin{cases} 1/\sigma_i & \sigma_i \neq 0 \\ 0 & \sigma_i = 0 \end{cases} \qquad (7.20)$$

The numbers σ_i, which are nonnegative, are called the singular values. The number of non-zero σ_i equals rank A. An advantage of this method is that it is possible to make a more adequate determination of rank A using the singular values. Moreover, the smallest non-zero singular value gives a measure of how critical this determination is. The spectral norm of A is defined by

$$|| A ||_2 = \sup_{|| x || = 1} || Ax || = \sigma_1$$

σ_1 = the greatest singular value, and the condition number

$$\text{cond } A = || A ||_2 \cdot || A^{\dagger} ||_2 = \sigma_1/\sigma_r$$

σ_r is the smallest non-zero singular value. These two numbers are thus easily obtained from the algorithm.

Similar methods (e.g. orthogonal triangularizations) are used in |12|, |19|, |22|, |25|, |26|.

In |4| an algorithm similar to the one used is given. As shown in |21| this algorithm can be derived from Mayne's algorithm.

In |6| iterative refinement of least squares solution is considered. (rank A = n is assumed). In |7| Björck treats the constrained least squares problem and also the case rank A < n. ALGOL programs, which also consider the case with a constrained problem, are listed in this reference.

Some other methods of probably less interest will now be shortly commented.

In |20| Mayne gives an algorithm, which uses Gram-Schmidt orthogonalization.

A modified Gram-Schmidt orthogonalization is used in |5|, |6|, where rank A = n is assumed.

Bauer seems to consider the case rank A = n in |1|. An ALGOL program is given.

An iterative computation scheme is given in |3|. Let α satisfy

$$0 < \alpha < \frac{2}{\lambda_1(A^TA)}$$

$\lambda_1(A^TA)$ is the greatest eigenvalue of A^TA. It is shown that the sequence

$$\begin{cases} Y_0 = \alpha A^T \\ Y_{k+1} = Y_k(2I - AY_k) \end{cases} \tag{7.21}$$

converges to $A^\dagger$ as $k \to \infty$. The convergence is given by

$$|| A^\dagger - Y_{k+1} || \leq || A || \cdot || A^\dagger - Y_k ||^2$$

This method seems to be instable in some sense. Introduce

$$X_k = A^\dagger - Y_k$$

and let

$$A = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$$

We have

$$|| X_{k+1} || \leq || A || \cdot || X_k ||^2$$

Let

$$X_k = \begin{vmatrix} -\varepsilon & \varepsilon \\ \varepsilon & -\varepsilon \end{vmatrix}$$

Then

$$X_{k+1} = 2X_k + \theta(\varepsilon)$$

and Y_{k+1} is not better than Y_k. The algorithm cannot be used for refinements of previously received results.

In |18| the following method is suggested. Let A be hermitian (no restriction, see (4.1), (4.2)). Then solve $A^2X^T = A$. There is in general no unique solution, but any of them may be used to get $A^\dagger = XAX^T$.

We finish this chapter by pointing out some references, which contain error analysis of the computations of the pseudoinverse. Such references are |5|, |6|, |14|, |22|.

8. PROGRAMS AND NUMERICAL EXAMPLES.

As mentioned in chapter 7 two different algorithmes have been used for the numerical examples. Below we will discuss how some details of the algorithm by Mayne have been programmed. After that some numerical examples will be given.

The algorithm by Golub-Reinsch is fully discussed in |13|, which also contains an ALGOL program. A FORTRAN version of this program is used in the examples. This subroutine is called SVD (Singular Value Decomposition).

The program, using the algorithm by Mayne, consists of two subroutines, called PSINV and SOLVE1. They are written in FORTRAN and have been used partly on a CD 3600 (machine accuracy $\sim 10^{-10}$), partly on a Univac 1108 (machine accuracy $\sim 10^{-7}$). SVD is used only on the Univac 1108, according to the rearrangement of Lunds Datacentral.

Now we will describe the subroutines PSINV and SOLVE1. The description is illustrated by two flow-charts.

Subroutine PSINV (EPS, A, APSINV, IRANK, IA, IB, NA, NB, ITMAX)

Parameters:

A	- input matrix of order NB× NA, NA ≤ NB
APSINV	- at output equals $A^{\dagger}$
IRANK	- rank A. If rank A is known = NA, it is possible to put IRANK = NA on entry to simplify the computations.
EPS	- value to be used as a tolerance for acceptance of small vectors
IA, IB	- dimension parameters
NA, NB	- parameters determining the order of A
ITMAX	- max number of iterations in SOLVE1.

Since in the algorithm m ≤ n, while here the corresponding relation is NB ≥ NA, some transposition of matrices must be done. The reason for m ≤ n is probably numerical accuracy. The reason for the choice NB ≥ NA is that this case is the most common one, at least in applications to identification.

A matrix A0 is computed by column pivoting from A. A slight modification of the subroutine DECOM in |11| is used. A vector is regarded as zero if all components are less than EPS in magnitude. In this manner we find linearly independent columns of A. The matrix A1 in the algorithm has row vectors equal to the found linearly independent column vectors of A. (The vectors received after the elimination procedure are not used in order not to increase the errors.) Simultaneously rank A is obtained as the number of linearly independent column vectors of A.

If now IRANK = NA then SOLVE1 is used directly to give the result $A^{\dagger}$, using (4.1). To a given matrix S SOLVE1 computes $(SS^T)^{-1}S = S^{T\dagger}$. If IRANK < NA then C in the algorithm is first computed. SOLVE1 is then used to get $(CC^T)^{-1}C$ and finally APSINV is computed from (7.13).

Subroutine SOLVE1 (D, C, ITMAX, NM, NIQ, IA, IB)

Parameters:

C	- input matrix of order NIQ * NM, Rank C = NIQ, NIQ ≤ NM
D	- output matrix, = $(CC^T)^{-1}C$
ITMAX	- greatest number of iterations
IA, IB	- dimension parameters
NM, NIQ	- parameters, determining the order of C

D is computed as the solution of $(CC^T)D = C$. It is easier to solve this equation than to compute the inverse $(CC^T)^{-1}$ explicitely. First $C1 = CC^T$ is computed. C1 is positive definite. C1 is then factorized $C1 = GG^T$, where G is a lower triangular matrix. A simple algorithm for this computation is found in |11|. The system

$$(GG^T)D = C \tag{8.1}$$

is then solved by introducing Y as the solution of

$$GY = C \tag{8.2}$$

Then we have

$$G^T D = Y \qquad (8.3)$$

(8.2) and (8.3) are easily solved recursively since G is a triangular matrix.

The solution to (8.1) is easy to iterate to machine accuracy as shown in |11|. Suppose D is not an exact solution, while D + D1 is. Then $(GG^T)(D + D1) = C$ and $(CC^T)D1 = C - (GG^T)D$. We get the correction matrix from (8.2) and (8.3) if in (8.2) C is substituted with

$$C' = C - C1 \cdot D \qquad (8.4)$$

The iteration is repeated until either ITMAX stops the procedure or || D1 || is smaller than some test quantity. In (8.4) double precision is used, since it is a difference between two great, almost equal, matrices.

A subroutine PART has been written, which computes and refines the G matrix iteratively. When C is ill-conditioned this procedure will make the result better. Maybe, in these cases, it is easier to use double precision.

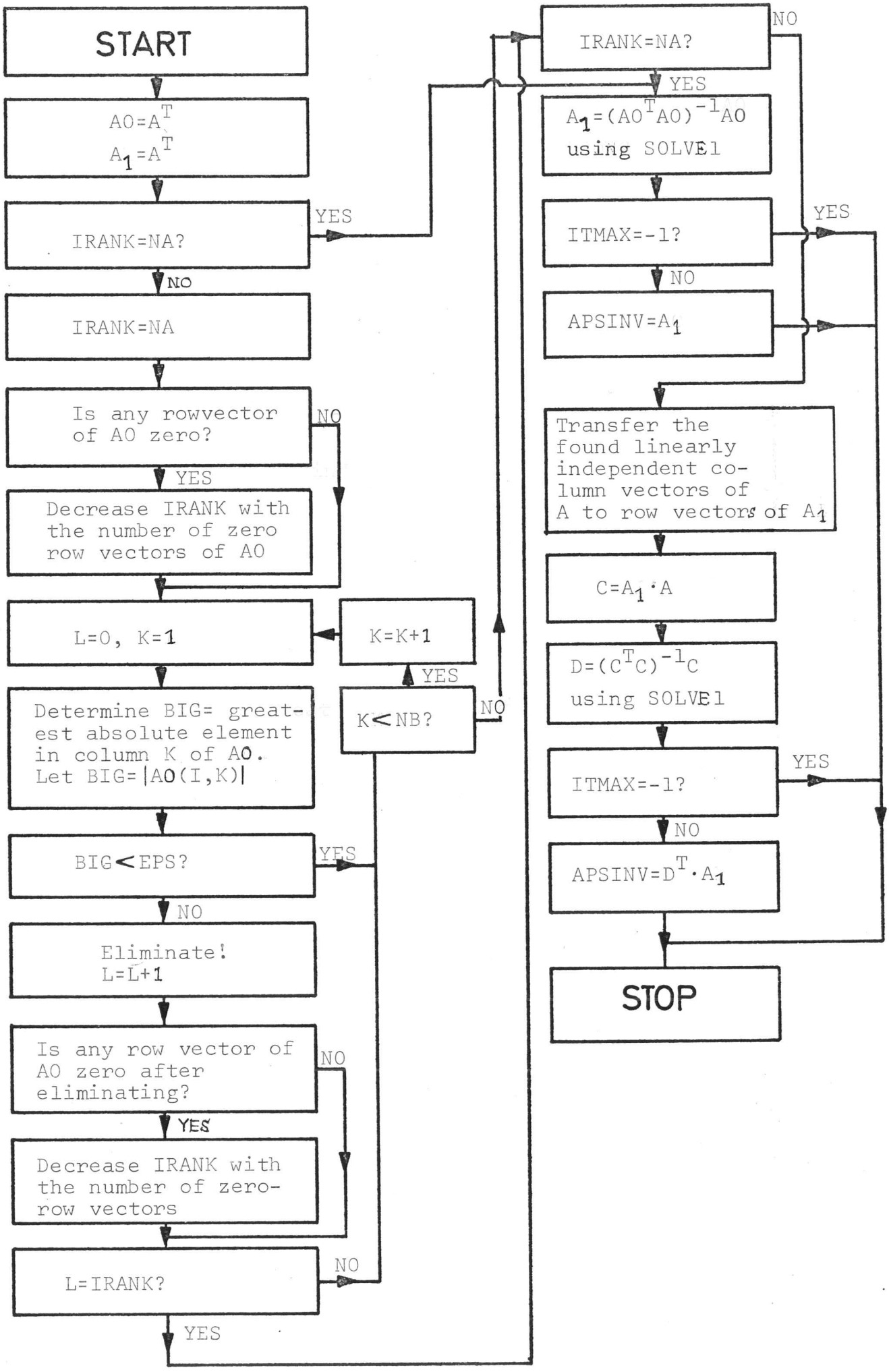

Fig. 8.1 - Flow chart for PSINV.

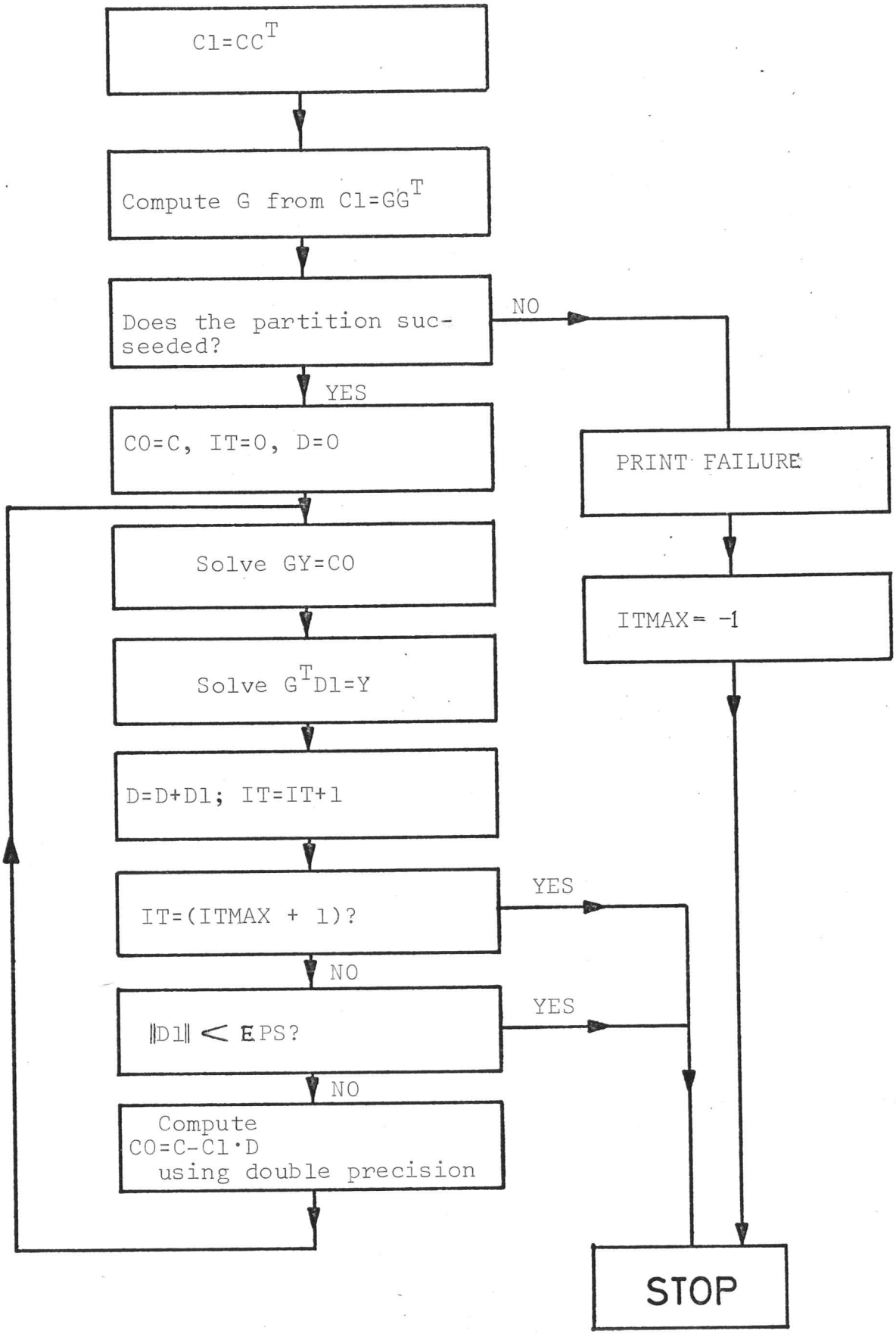

Fig. 8.2 - Flow chart for SOLVE1

Now we turn to some numerical examples.

Example 1

This example is a simple test example.

Consider the matrix A, given by

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix}$$

The rank of A equals 2. The correct pseudoinverse is

$$A^{\dagger} = \frac{1}{18} \begin{bmatrix} 2 & 4 & -2 \\ -1 & 7 & -8 \\ 5 & 1 & 4 \end{bmatrix}$$

The subroutines PSINV and SOLVE1 were used on the CD 3600. The parameters were EPS = 10^{-9} and ITMAX = 0. The received result differs from the exact solution with just one unit in the last significant digit (the relative error is $\sim 10^{-10}$).

Example 2

Consider the linear system Ax = b, where

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1+\delta_1 & 1 & 1 \\ 1 & 1 & 1+\delta_2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 10 \\ 2 \\ 10+2\delta_1 \\ 2+3\delta_2 \end{bmatrix} \tag{8.5}$$

The system is constructed such that Ax = b with

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \tag{8.6}$$

Thus if $\delta_1 \neq 0$, $\delta_2 \neq 0$ then x given by (8.6) is the solution of minimizing $|| Ax - b ||^2$.

If, however, $\delta_1 = 0$ or $\delta_2 = 0$ then A^{-1} does not exist and there is no unique minimum. After some computations the following (theoretical) result is obtained for $x = A^{\dagger}b$.

δ_1	$\neq 0$	$\neq 0$	$= 0$	$= 0$
δ_2	$\neq 0$	$= 0$	$\neq 0$	$= 0$
rank A	4	3	3	2
x_1	1	2	1.5	2
x_2	2	2	1.5	2
x_3	3	2	3	2
x_4	4	4	4	4

Table 1 - Theoretical solution $x = A^{\dagger}b$, with A and b given from (8.5).

The similarity of the second column to the fourth is due to the special feature of the system. The system is treated in two separate ways.

a) EPS varied with δ_1 and δ_2 fixed, PSINV used.

This example will illustrate the influence of EPS on the result. Intuitively, a great EPS may cause that the rank of A will be too small and then the result can be everything. The simulations were made on the CD 3600. The parameter ITMAX was 0. At a first simulation we chose $\delta_1 = 10^{-5}$ and $\delta_2 = 10^{-3}$. The result is given in table 2.

EPS	1	10^{-1}, 10^{-2}, 10^{-3}	10^{-4}, 10^{-5}	10^{-6}, 10^{-7}
rank A	0	2	3	4
x_1	0	2.00025	1.99990	14.549
x_2	0	2.00026	1.99991	-11.553
x_3	0	1.99975	2.00044	3.004
x_4	0	3.99975	3.99975	4.000

Table 2 - Result from example 2 with $\delta_1 = 10^{-5}$, $\delta_2 = 10^{-3}$. Solution computed on the CD 3600.

With EPS = 1 all elements are regarded as zero.

The results with EPS = 10^{-1}, 10^{-2}, 10^{-3} are expected. Rank A is considered as 2 (or equivalently $\delta_1 = 0$, $\delta_2 = 0$) and the theoretical result from table 1 is $x^T = [2, 2, 2, 4]$.

When EPS = 10^{-4}, 10^{-5} rank A is regarded as 3. From table 1 we expect the result $x^T = [1.5, 1.5, 3, 4]$. Also the smallest EPS will give a result, which differs from the theoretical one.

In the last case the reason for this discrepancy is that A is too ill-conditioned. From b) we have cond A ≈ 10^6. Since we invert CC^T in SOLVE1 we have to square this quantity. When the condition number × the machine accuracy is greater than 1 we cannot expect to get any correct result. Since the actual condition number is 10^{12} and the machine accuracy $\sim 10^{-10}$ the result should not be astonishing. Note that the x given in the last column of table 4 will make $|| Ax - b ||$ very small.

At a second simulation we chose $\delta_1 = 10^{-2}$ and $\delta_2 = 10^{-3}$. Then cond A is $\sim 10^4$ so we hope to get correct results for small values of EPS.

EPS	1	10^{-1}, 10^{-2}	10^{-3}	10^{-4}, 10^{-5}
rank A	0	2	3	4
x_1	0	1.995	2.016	1.000 079
x_2	0	2.010	1.969	1.999 995
x_3	0	1.995	2.015	2.999 926
x_4	0	4.000	4.000	4.000 000

Table 3 - Result from example 2 with $\delta_1 = 10^{-2}$, $\delta_2 = 10^{-3}$. Solution computed on the CD 3600.

In this simulation the obtained results coincide very well with the expected values of x from table 1.

b) δ_1 and δ_2 varied. PSINV and SVD compared.

In this case the simulations were made on the Univac 1108. In PSINV we used EPS = 10^{-7}, ITMAX = 0. δ_1 and δ_2 were varied and the results from PSINV and SVD were compared. The accuracy is measured as

$$\max_i \frac{|\delta x_i|}{|x_i|}$$

where the component x_i is taken from table 1, and δx_i is the deviation of x_i.

From table 4 it is obvious that in this example SVD will give results with higher accuracy than PSINV. It could be noted that $\delta_1 = 0$, $\delta_2 = 10^{-3}$ will give $x^T \approx [2, 2, 2, 4]$ when PSINV is used. The same deviation from the expected result was obtained in table 2 when EPS = 10^{-4}, 10^{-5}.

δ_1	δ_2	PSINV		SVD		
		rank A	Accuracy	rank A	Accuracy	cond A
10^{-1}	10^{-1}	4	10^{-5}	4	10^{-6}	10^{2}
	10^{-2}	4	10^{-4}	4	10^{-5}	10^{3}
	10^{-3}	4	0.12	4	10^{-4}	10^{4}
	10^{-4}	4	FAILED	4	10^{-4}	10^{5}
10^{-2}	10^{-1}	4	10^{-4}	4	10^{-5}	10^{3}
	10^{-2}	4	10^{-3}	4	10^{-4}	10^{3}
	10^{-3}	4	10^{-2}	4	10^{-4}	10^{4}
	10^{-4}	4	FAILED	4	10^{-3}	10^{5}
10^{-3}	10^{-1}	4	10^{-2}	4	10^{-4}	10^{4}
	10^{-2}	4	0.2	4	10^{-4}	10^{4}
	10^{-3}	4	0.2	4	10^{-4}	10^{4}
	10^{-4}	4	FAILED	4	10^{-4}	10^{5}
10^{-4}	10^{-1}	4	FAILED	4	10^{-3}	10^{5}
	10^{-2}	4	"	4	10^{-3}	10^{5}
	10^{-3}	4	"	4	10^{-3}	10^{5}
	10^{-4}	4	"	4	10^{-3}	10^{5}
10^{-5}	10^{-1}	4	FAILED	4	10^{-2}	10^{6}
	10^{-2}	4	"	4	0.05	10^{6}
	10^{-3}	4	"	4	10^{-3}	10^{6}
	10^{-4}	4	"	4	0.02	10^{6}
0	10^{-1}	3	10^{-3}	3	10^{-7}	10^{2}
	10^{-2}	3	FAILED	3	10^{-5}	10^{3}
	10^{-3}	3	0.5	3	10^{-4}	10^{4}
	10^{-4}	3	FAILED	3	10^{-3}	10^{5}

Table 4 - Result from example 2.
Solutions are computed on the Univac 1108.
Comparison between PSINV and SVD.

Example 3

In this example Hilbert matrices were used. It is well-known, |11|, that they are very ill-conditioned. A subroutine, given in |11| was used to generate the inverses of the Hilbert matrices. The inverses T_n (n denotes the order) were inverted with PSINV and with SVD to get H_n. Afterwards the result T_n^{-1} was compared with HA_n, which was generated as

$$(HA_n)_{i,j} = \text{FLOAT}(1/(I + J - 1))$$

The computations were made partly on the Univac 1108 (PSINV and SVD used), partly on the CD 3600 (only PSINV used).

The result from computations on the CD 3600 are given in table 5.

n	$\lvert\text{cond}(T_n)\rvert^2$	ITMAX	Number of iterations	$\max_{i,j}\lvert(T_n^{-1} - HA_n)_{i,j}\rvert$
1	$1 \cdot 10^0$	0 5	0 1	0 0
2	$4 \cdot 10^2$	0 5	0 1	$5 \cdot 10^{-10}$ 0
3	$3 \cdot 10^5$	0 5	0 1	$1 \cdot 10^{-6}$ 0
4	$2 \cdot 10^8$	0 5	0 3	$2 \cdot 10^{-4}$ 0
5	$2 \cdot 10^{11}$	0 10	0 10	$1 \cdot 10^{-1}$ $2 \cdot 10^{-10}$
6	$2 \cdot 10^{14}$	10	FAILED	-

Table 5 - Results from example 3.
Computed on the CD 3600.

The result is good. The failure when n = 6 depends on the accuracy of the machine ($\sim 10^{-10}$).

Table 6 gives the result from the computations on the Univac 1108.

	PSINV			SVD	
n	ITMAX	Number of iterations	$\max_{i,j}\lvert(T_n^{-1} - HA_n)_{ij}\rvert$	cond A	$\max_{i,j}\lvert(T_n^{-1} - HA_n)_{ij}\rvert$
1	0 10	0 1	0 0	1	0
2	0 10	0 2	$4 \cdot 10^{-7}$ $1 \cdot 10^{-8}$	20	$1 \cdot 10^{-8}$
3	0 10	0 2	$1 \cdot 10^{-5}$ $1 \cdot 10^{-8}$	$5 \cdot 10^{2}$	$6 \cdot 10^{-7}$
4	0 10	0 5	$2 \cdot 10^{-2}$ $4 \cdot 10^{-9}$	$2 \cdot 10^{4}$	$3 \cdot 10^{-5}$
5	0 10		FAILED	$5 \cdot 10^{5}$	$7 \cdot 10^{-5}$
6	0 10		FAILED	10^{7}	$8 \cdot 10^{-3}$
7	0 10		FAILED	$2 \cdot 10^{8}$	$7 \cdot 10^{-1}$
8	0 10		FAILED	(10^{8})	1

Table 6 - Results from example 3.
Solutions are computed on the Univac 1108.
Comparison between PSINV and SVD.

From table 6 we conclude that using PSINV we can calculate H_n with n up to 4. When a suitable number of iterations is used the result is brought to machine accuracy.

The subroutine SVD will give result for n = 1, ..., 7. When n = 8 the inverse T_8^{-1} differs very much from the true Hilbert matrix H_8. The failure is also shown in the obtained value of cond A. The correct result is $1.53 \cdot 10^{10}$. It is also shown in the table that with low values of n, PSINV with iteration will give the best accuracy.

9. APPLICATION TO IDENTIFICATION I. INTRODUCTION.

In this and the following three chapters we will consider an identification problem. First we give some preliminaries and a statement of the problem. It will also be shown that it turns out to be an ordinary least squares problem. In the next chapter we give some recursive formulas which allow the parameter estimates to be computed recursively as the data are obtained. Finally, in chapter 11 we discuss what happens when our model of the process is of wrong order. Chapter 12 contains some numerical examples.

Now consider a linear, time-invariant, discrete, single input, single output system. Let the input signal u be piece-wise constant over the sampling intervals, which are of constant length. Then the system can be represented by

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = b_1 u(t-1) + \ldots + b_n u(t-n) + e(t) \tag{9.1}$$

e(t) is assumed to be a sequence of independent, equally distributed random variables with zero mean and finite variance.

The coefficients $a_1, \ldots, a_n, b_1, \ldots, b_n$ are assumed to be unknown and the problem is to find them. We then perform experiments on the systems by changing the input u and observing the output y. Let us introduce a matrix notation for the problem.

$$Y = \begin{bmatrix} y(n+1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y(n+N) \end{bmatrix} \qquad \phi = \begin{bmatrix} -y(n) & \ldots\ldots & -y(1) & u(n) & \ldots\ldots & u(1) \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ -y(n+N-1) & \ldots & -y(N) & u(n+N-1) & \ldots & u(N) \end{bmatrix}$$

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \qquad e = \begin{bmatrix} e(n+1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ e(n+N) \end{bmatrix} \tag{9.2}$$

It is worth pointing out that in most cases the number of measurements (n+N) is much greater than n.

The system equation (9.1) implies

$$Y = \phi\theta + e \tag{9.3}$$

A usual way of determining an estimate $\hat{\theta}$ of θ is to do it in such a way that the loss function

$$V = e^T e = || Y - \phi\hat{\theta} ||^2 \tag{9.4}$$

is minimized, see |28|. From the previous chapters we know that the solution of this problem is given by

$$\hat{\theta} = \phi^{\dagger} Y \tag{9.5}$$

If rank $\phi = 2n$, then there is just this $\hat{\theta}$, which minimizes (9.4). Further we have in this case

$$\phi^{\dagger} = (\phi^T \phi)^{-1} \phi^T$$

If, however, rank $\phi < 2n$ then there is no unique minimum and it is not trivially true (in fact it is not true) that we will get a correct result, or a result as good as possible. We note that, if possible, rank ϕ should be = 2n, that is at least that the input signal should be chosen irregularly enough. In other cases the last n columns of ϕ may be linearly dependent.

Anticipating the result in chapter 11 we mention that with e = 0 (noisefree system) $\phi^T\phi$ will not be invertible if the model is of too high order.

10. APPLICATION TO IDENTIFICATION II. RECURSIVE FORMULAS.

Now we will give some recursive formulas for the computation of $\hat{\theta}$. In practice it often happens that the observations are obtained recursively. The formulas are to handle this fact.

It could also be pointed out that with rank $\phi = 2n$ the given formulas coincide exactly with those in |28|. Åström, however, uses $\phi^{\dagger} = (\phi^T\phi)^{-1}\phi^T$ and consequently must have some approximate initial conditions. This fact is the essential difference between Åström's formulas and those given here.

For this case (rank $\phi = 2n$) it is known from practical computations that the method converges.

Let $\hat{\theta}_N$ be the estimate based on the N first equations. We then want to get $\hat{\theta}_{N+1}$ from $\hat{\theta}_N$ without finding the pseudoinverse of a matrix of order $(N+1) \times 2n$.

The matrices introduced in (9.2) we now call Y_N, ϕ_N, θ and e_N respectively. Further we introduce

$$Y_{N+1} = \begin{vmatrix} Y_N \\ \cdots \\ y_{N+1} \end{vmatrix}, \qquad \phi_{N+1} = \begin{vmatrix} \phi_N \\ \cdots \\ \varphi_{N+1}^T \end{vmatrix} \tag{10.1}$$

We have

$$\hat{\theta}_{N+1} = \phi_{N+1}^{\dagger} Y_{N+1} \tag{10.2}$$

The formulas (4.4) and (4.5) are easily transposed to get the following relations. Making it comfortable we sometimes drop the index of ϕ_N and φ_{N+1}.

i) $$\begin{bmatrix} \phi \\ \cdots \\ \varphi^T \end{bmatrix}^{\dagger} = \begin{bmatrix} \phi^{\dagger} - \dfrac{(I - \phi^{\dagger}\phi)\varphi\varphi^T\phi^{\dagger}}{\varphi^T(I - \phi^{\dagger}\phi)\varphi} & \dfrac{(I - \phi^{\dagger}\phi)\varphi}{\varphi^T(I - \phi^{\dagger}\phi)\varphi} \end{bmatrix} \tag{10.3}$$

ii) $$\begin{bmatrix} \phi \\ \cdots \\ \varphi^T \end{bmatrix}^{\dagger} = \begin{bmatrix} \phi^{\dagger} - \dfrac{\phi^{\dagger}\phi^{\dagger T}\varphi\varphi^T\phi^{\dagger}}{1 + \varphi^T\phi^{\dagger}\phi^{\dagger T}\varphi} & \dfrac{\phi^{\dagger}\phi^{\dagger T}\varphi}{\varphi^T\phi^{\dagger}\phi^{\dagger T}\varphi} \end{bmatrix} \tag{10.4}$$

Case i) is equivalently to rank ϕ_{N+1} > rank ϕ_N. This condition can be written $(I - \phi_N^{\dagger}\phi_N)\varphi_{N+1}^{T} \neq 0$.

It is natural to introduce the matrices

$$P_N = I - \phi_N^{\dagger}\phi_N \tag{10.5}$$

$$B_N = \phi_N^{\dagger}\phi_N^{\dagger T} \tag{10.6}$$

Both P_N and B_N are of constant type $2n \times 2n$.

Equations (10.1) to (10.6) now give

i)
$$\hat{\theta}_{N+1} = \left[\begin{array}{c|c} \phi^{\dagger} - \dfrac{P_N\varphi\varphi^T\phi^T}{\varphi^T P_N \varphi} & \dfrac{P_N\varphi}{\varphi^T P_N \varphi} \end{array}\right]\left[\begin{array}{c} Y_N \\ \cdots \\ y_{N+1} \end{array}\right]$$

$$= \phi^{\dagger}Y_N + \frac{P_N\varphi}{\varphi^T P_N \varphi}\left(y_{N+1} - \varphi^T\phi^{\dagger}Y_N\right)$$

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{P_N\varphi}{\varphi^T P_N \varphi}\left(y_{N+1} - \varphi^T\hat{\theta}_N\right) \tag{10.7}$$

ii)
$$\hat{\theta}_{N+1} = \left[\begin{array}{c|c} \phi^{\dagger} - \dfrac{B_N\varphi\varphi^T\phi^{\dagger}}{1 + \varphi^T B_N \varphi} & \dfrac{B_N\varphi}{1 + \varphi^T B_N \varphi} \end{array}\right]\left[\begin{array}{c} Y_N \\ \cdots \\ y_{N+1} \end{array}\right]$$

$$= \phi^{\dagger}Y_N + \frac{B_N\varphi}{1 + \varphi^T B_N \varphi}\left(y_{N+1} - \varphi^T\phi^{\dagger}Y_N\right)$$

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{B_N\varphi}{\varphi^T B_N \varphi}\left(y_{N+1} - \varphi^T\hat{\theta}_N\right) \tag{10.8}$$

As a summary we get

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K(N)(y_{N+1} - \varphi^T_{N+1}\hat{\theta}_N) \tag{10.9}$$

The term $\varphi^T_{N+1}\hat{\theta}_N$ is the value of y_{N+1} if the model was perfect and the system was without noise. The brackets are just the difference between the measured and the predicted value of y_{N+1}.

The weighting factor K(N) is given by (10.7) or (10.8) according to the actual case.

If the described method should be of any value we need recursive formulas for P_N and B_N. Such equations are derived in Appendix A, and the result is:

$$P_0 = I \tag{10.10}$$

case i)
$$P_{N+1} = P_N - \frac{c_N c_N^T}{\varphi^T_{N+1} c_N} \tag{10.11}$$

case ii)
$$P_{N+1} = P_N - \frac{d_N c_n^T}{1 + \varphi^T_{N+1} \cdot d_n} \tag{10.12}$$

$$B_0 = 0 \tag{10.13}$$

case i)
$$B_{N+1} = B_N - \frac{c_N d_N^T}{\varphi^T_{N+1} c_N} - \frac{d_N c_N^T}{\varphi^T_{N+1} c_N} +$$

$$+ \frac{c_N c_N^T}{(\varphi^T_{N+1} c_N)^2} (1 + \varphi^T_{N+1} d_N) \tag{10.14}$$

case ii)
$$B_{N+1} = B_N - \frac{d_N d_N^T}{1 + \varphi^T_{N+1} d_N} \tag{10.15}$$

where

$$c_N = P_N \varphi_{N+1} \tag{10.16}$$

$$d_N = B_N \varphi_{N+1} \tag{10.17}$$

With these notations we also have

$$\text{case i)} \qquad K(N) = \frac{c_N}{\varphi_{N+1}^T c_N} \tag{10.18}$$

$$\text{case ii)} \qquad K(N) = \frac{d_N}{1 + \varphi_{N+1}^T d_N} \tag{10.19}$$

We repeat the criterion of cases is that case i) is equivalently to

$$P_N \varphi_{N+1} \neq 0 \tag{10.20}$$

We also note that if rank $\Phi = 2n$ then $P_N = 0$ and as this is the maximum of rank ϕ, P_N is not needed in the future. The rank of ϕ is obtained as the number of times case i) is used.

Finally we summarize. The equations (10.9) - (10.20) are the recursive formulas.

11. APPLICATION TO IDENTIFICATION III. IDENTIFICATION WITH MODEL OF TOO HIGH ORDER.

In practical cases the exact order of the system is not known. Therefore a fundamental question is what happens when the order of the model differs from the order of the system. When the order of the model is too low, we will get estimates of the parameters of a system, which is an approximation of the true system to a system of this lower order. In this chapter we discuss what happens when the order of the model is too high.

It could be mentioned that Åström gives some interesting results in |28|. He states a theorem by which it is possible to test if the decrease of the loss function, when the order of model is increased, is of statistical significance. Here we will consider the problem from another point of view. It turns out to be necessary to distinguish between systems with noise and noise-free ones.

We need some results about the rank ϕ as preliminaries. Unfortunately we are not able to give strict proofs to all of them.

In the following we first deal with systems with noise. The essential result is theorem 11.2.

The same discussion cannot be applied to systems without noise. As shown in theorem 11.6 a noiseless system will almost always give quite other results, when the order of the model is too high.

Before starting we will point out, that the next chapter contains numerical examples, illustrating the results from this chapter.

System with Noise.

Now consider the system

$$y(t) + a_1 y(t-1) + \ldots + a_p y(t-p) =$$

$$= b_1 u(t-1) + \ldots + b_r u(t-r) + e(t) \qquad (11.1)$$

As before e(t) is a sequence of independent, equally distributed random variables with zero mean and finite covariance.

Let the model be

$$y(t) + \hat{a}_1 y(t-1) + \ldots + \hat{a}_n y(t-n) =$$

$$= \hat{b}_1 u(t-1) + \ldots + \hat{b}_n u(t-n) \qquad (11.2)$$

where the case $n > \max(p,r)$ is allowed. Then we have from Appendix 3

Theorem 11.1

With the assumptions above and ϕ defined by (9.2) rank $\phi = 2n$ with probability one, if

$$\operatorname{rank} \begin{bmatrix} u(n) & \ldots & u(1) \\ \vdots & & \vdots \\ u(N-1) & \ldots & u(N-n) \end{bmatrix} = n$$

Remark

The conclusion of the theorem is:

A system with noise and an input signal, which varies irregularly enough, has the pseudoinverse

$$\phi^{\dagger} = (\phi^T \phi)^{-1} \phi^T$$

The minimum of

$$V = || Y - \phi\hat{\theta} ||^2$$

is unique.

Introduce

$$\theta_0 = \begin{bmatrix} a_1 & 1 \\ \vdots & \\ a_0 & p \\ 0 & \\ \vdots & \\ 0 & n \\ b_1 & n+1 \\ \vdots & \\ b_r & n+r \\ 0 & \\ \vdots & \\ 0 & 2n \end{bmatrix} \tag{11.3}$$

which contains the correct values of the parameters. Let us put $\hat{\theta} = \theta_0 + \theta_1$. We get

$$V = || Y - \phi(\theta_0 + \theta_1) ||^2 \tag{11.4}$$

According to the equation of the system (9.3) we have

$$Y = \phi\theta_0 + e$$

and

$$V = || e - \phi\theta_1 ||^2 \tag{11.5}$$

When V is minimized we get

$$\theta_1 = \phi^{\dagger} e$$

We have the following

Theorem 11.2

A system with noise and with rank $\phi = 2n$ will give the estimates

$$\hat{\theta} = \theta_0 + \phi^{\dagger} e \tag{11.6}$$

Remark

For a noisefree system ($e \equiv 0$) there is perhaps no unique θ_1 since rank $\phi = 2n$ may be impossible.

An essential question is the behaviour of the term $\phi^{\dagger} e$. It is clear that it decreases when the variance of the errors is decreased. In |28| Åström states a theorem which shows that with suitable assumptions on the input signal $u(t)$, $\phi^{\dagger} e \to 0$ in mean square as $N \to \infty$. We note that it can be shown that $|| \phi^{\dagger} || \to 0$, $N \to \infty$ with use of (10.15). This is, however, not enough since $E || e ||^2 = N\sigma^2 \to \infty$, $N \to \infty$.

System without Noise.

Now we want to consider a noisefree system

$$y(t) + a_1 y(t-1) + \ldots + a_p y(t-p) =$$

$$= b_1 u(t-1) + \ldots + b_r u(t-r) \tag{11.7}$$

and the model

$$y(t) + \hat{a}_1 y(t-1) + \ldots + \hat{a}_n y(t-n) =$$

$$= \hat{b}_1 u(t-1) + \ldots + \hat{b}_n u(t-n) \tag{11.8}$$

From Appendix B we have

Theorem 11.3

Given the system (11.7) and the model (11.8). Let $n = \max(p,r)$, $k = n - \min(p,r)$. Assume that $u(t)$ is white noise, rank $\phi_{00} = p+r$, where

$$\phi_0 = \begin{bmatrix} y(n) & \dots & y(n-p+1) & u(n) & \dots & u(n-r+1) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(n+N-1-k) & \dots & y(n-p+N-k) & u(n+N-1-k) & \dots & u(n-r+N-k) \end{bmatrix}$$

and that rank ϕ = rank $q^k\phi$, where q is the forward shift operator working on all the elements of ϕ. Then rank $\phi = 2n$ with probability one.

and

Theorem 11.4

Given the system (11.7) and the model (11.8). Let $k = n - \max(p,r) > 0$, $m = \min(p,r)$, $\ell = |p-r|$, $j = k+\ell$. Assume that $u(t)$ is white noise, rank $\phi_1 = p+r$ and rank $q^j\phi$ = rank ϕ. Here ϕ_1 denotes the matrix

$$\phi_1 = \begin{bmatrix} y(n) & \dots & y(n-p+1) & u(n) & \dots & u(n-r+1) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(n+N-1-m) & \dots & y(n-p+N-m) & u(n+N-1-m) & \dots & u(n-r+N-m) \end{bmatrix}$$

q is the forward shift operator working on all the elements of ϕ. Then rank $\phi = 2n-k = n + \max(p,r)$ with probability one.

Since rank $\phi < 2n$ if the order of the model is too high, the result for systems with noise is not applicable. Further we cannot expect that an assumption as $\hat{\theta} = \theta_0 + \theta_1$ will give $\theta_1 = 0$. What we can hope is to get common factors in the pulse transfer function.

$$H(q^{-1}) = \frac{b_1q^{-1} + \dots + b_rq^{-r}}{1 + a_1q^{-1} + \dots + a_pq^{-p}} \tag{11.9}$$

where q is the forward shift operator. In theorem 11.6 we show that this in fact occurs.

At first we consider the case when n = max(p,r). We have

Theorem 11.5

Given a noisefree system described by (11.7), a model described by (11.9) with n = max(p,r), rank ϕ = 2n. Then the model will give the estimates $\hat{\theta} = \theta_0$.

Proof

Since rank ϕ = 2n there is a unique minimum of the loss function

$$V = || Y - \phi\theta ||^2$$

Now using $\theta = \theta_0$ we have

$$Y = \phi\theta_0 \tag{11.10}$$

according to (9.3) and (11.3). Then $V(\theta_0) = 0$. Naturally V is minimized and the existence of a unique minimum completes the argumentation.

Q.E.D.

Now we turn to the case n > max(p,r). With use of theorem 11.5 there is no restriction in assuming the system to be (p=r).

$$y(t) + a_1y(t-1) + \ldots + a_py(t-p) =$$

$$= b_1u(t-1) + \ldots + b_pu(t-p) \tag{11.11}$$

We have the model

$$y(t) + \hat{a}_1y(t-1) + \ldots + \hat{a}_ny(t-n) =$$

$$= \hat{b}_1u(t-1) + \ldots + \hat{b}_nu(t-n) \tag{11.12}$$

with n > p.

Theorem 11.6

Given the system (11.11), the model (11.12), the relations $n > p$, rank $\phi = n+p$. Then the least squares identification will give estimated parameters, which physically means that the pulse transfer function contains factors in common.

Proof.

In the proof we will first assume that the pulse transfer function will get common factors. Then we will derive some formulas, expressing what this means to the solution $\hat{\theta}$. The last part of the proof is to show that this $\hat{\theta}$ must be solution $\phi^{\dagger}Y$.

Now let us assume that we will get common factors in the pulse transfer function. This can be written:

$$\frac{\hat{b}_1 q^{-1} + \ldots + \hat{b}_n q^{-n}}{1 + \hat{a}_1 q^{-1} + \ldots + \hat{a}_n q^{-n}} = \frac{b_1 q^{-1} + \ldots + b_p q^{-p}}{1 + a_1 q^{-1} + \ldots + a_p q^{-p}} \cdot$$

$$\cdot \frac{1 + \lambda_1 q^{-1} + \ldots + \lambda_k q^{-k}}{1 + \lambda_1 q^{-1} + \ldots + \lambda_k q^{-k}} \qquad (11.13)$$

where $n = p+k$. The constants $\lambda_1, \ldots, \lambda_k$ are arbitrary.

(11.13) determinates

$$\hat{\theta} = \left[\hat{a}_1, \ldots, \hat{a}_n, \hat{b}_1, \ldots, \hat{b}_n\right]^T$$

as a function of $a_1, \ldots, a_p, b_1, \ldots, b_p, \lambda_1, \ldots, \lambda_k$. This relation is now to be expressed explicitly.

The polynomials shall be identical which gives us:

$$
\begin{aligned}
&\hat{b}_1 = b_1 \\
&\hat{b}_2 = b_1\lambda_1 + b_2 \\
&\quad\vdots \\
&\hat{b}_i = b_{i-k}\lambda_k + b_{i-k+1}\lambda_{k-1} + \dots b_i \qquad i = k+1, \dots p \\
&\quad\vdots \\
&\hat{b}_{p+k-1} = b_p\lambda_{k-1} + b_{p-1}\lambda_k \\
&\hat{b}_{p+k} = b_p\lambda_k
\end{aligned}
\tag{11.14}
$$

$$
\begin{aligned}
&\hat{a}_1 = a_1 + \lambda_1 \\
&\hat{a}_2 = a_2 + a_1\lambda_1 + \lambda_2 \\
&\quad\vdots \\
&\hat{a}_i = a_i + a_{i-1}\lambda_1 + \dots + a_{i-k}\lambda_k \qquad i = k, \dots, p \\
&\quad\vdots \\
&\hat{a}_{p+k-1} = a_p\lambda_{k-1} + a_{p-1}\lambda_k \\
&\hat{a}_{p+k} = a_p\lambda_k
\end{aligned}
$$

If we define $\lambda_0 = 1$ all the formulas could be written in the form

$$
\hat{b}_i = \sum_{\{j\}} \hat{b}_j\lambda_{i-j}, \qquad \hat{a}_i = \sum_{\{k\}} \hat{a}_k\lambda_{i-k} \tag{11.15}
$$

Now let us change the elements in $\hat{\theta}$ to

$$
\hat{\theta} = \begin{bmatrix} \hat{a}_1 \\ \hat{b}_1 \\ \vdots \\ \hat{a}_n \\ \hat{b}_n \end{bmatrix} \tag{11.16}
$$

This implies

$$\phi = \begin{bmatrix} -y(n) & u(n) & \dots & -y(1) & u(1) \\ \vdots & \vdots & & \vdots & \vdots \\ -y(n+N-1) & u(n+N-1) & \dots & -y(N) & u(N) \end{bmatrix} \tag{11.17}$$

Introduce

$$x_0 = \begin{bmatrix} a_1 \\ b_1 \\ \vdots \\ a_p \\ b_p \end{bmatrix} \tag{11.18}$$

and $0_i = [0, \dots 0]^T$ of dimension i. Further introduce the vectors

$$\theta_0 = \begin{bmatrix} x_0 \\ --- \\ 0_{2k} \end{bmatrix} \qquad \theta_i = \begin{bmatrix} 0_{2i-2} \\ ------ \\ 1 \\ 0 \\ ------ \\ x_0 \\ ------ \\ 0_{2k-2i} \end{bmatrix} \quad i = 1,\dots, k \tag{11.19}$$

This definition of θ_0 coincides with (11.3), i.e. θ_0 contains the correct parameters.

The equations (11.14) can now be written in the compact form

$$\hat{\theta} = \theta_0 + \lambda_1\theta_1 + \dots + \lambda_k\theta_k \tag{11.20}$$

It is rather easy to see that (11.11) implies

$$\phi\theta_i = 0 \qquad i = 1, \dots, k \tag{11.21}$$

Summarizing we note that common factors in the pulse transfer function mean that we have the estimate (11.20), where the vectors θ_i are defined by (11.19). They satisfy (11.21). It is also obvious from (11.19) that the vectors θ_i, $i = 1, \ldots, k$, are linearly independent.

What we now have to do is to show that $\hat{\theta}$, given by (11.20), in fact is the solution, i.e. $\hat{\theta} = \phi^{\dagger}Y$.

It was given rank $\phi = n+p$. Thus dim $N(\phi) = 2n - (n+p) = k$. Using (11.21) we conclude that the vectors θ_i, $i = 1, \ldots, k$ are a base of the set $N(\phi)$.

Introduce the notation $\theta^* = \phi^{\dagger}Y$. Let us show $\theta^* = \hat{\theta}$.

θ^* can be fully defined by the properties (see the proof of theorem 5.1):

i) $V(\theta) = || \phi\theta - Y ||^2$ is minimized when $\theta = \theta^*$.

ii) $\theta^* \in N(\phi)^{\perp} = R(\phi^T)$

We have

$$V(\hat{\theta}) = || \phi\theta_0 + \sum_{i=1}^{k} \phi\theta_i - Y ||^2 = 0$$

since $Y = \phi\theta_0$ (11.10) and $\phi\theta_i = 0$ (11.21).

In general θ_0 has a component in $N(\phi)$. Since θ_i, $i = 1, \ldots, k$, are bases of $N(\phi)$, we can subtract this component from θ_0 to obtain $\hat{\theta}$ in (11.20) if the coefficients $\lambda_1, \ldots, \lambda_k$ are chosen suitable.

Hence with "suitable" values of λ_i, $i = \ldots k$, $\hat{\theta}$ is a vector in $R(\phi^T)$. We have shown that $\hat{\theta}$ fulfils the uniquely definition of θ^* and then $\hat{\theta} = \theta^*$.

Q.E.D.

It could be of some value to have a more explicit formula, from which the constants $\lambda_1, \ldots, \lambda_k$ can be determined. The derivation of such a formula is now given.

The problem is: given θ_0, the base $\theta_1, \ldots, \theta_k$ of $N(\phi)$ then express the component of θ_0 in $N(\phi)$ as a linear combination of $\theta_1, \ldots, \theta_k$. Introduce θ_0' = the component of θ_0 in $N(\phi)^{\perp}$. Then we have

$$\theta_0 = -\sum_{i=1}^{k} \lambda_i \theta_i + \theta_0' \tag{11.22}$$

which is similar to (11.20). Forming inner products $\langle \theta_j | \theta_0 \rangle$ with $j = 1, \ldots, k$ we get the system

$$-\begin{bmatrix} \langle \theta_1|\theta_1\rangle & \cdots & \langle \theta_1|\theta_k\rangle \\ \vdots & & \vdots \\ \langle \theta_k|\theta_1\rangle & \cdots & \langle \theta_k|\theta_k\rangle \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{bmatrix} = \begin{bmatrix} \langle \theta_1|\theta_0\rangle \\ \vdots \\ \langle \theta_k|\theta_0\rangle \end{bmatrix} \tag{11.23}$$

Compare with (7.5). Since $\theta_1, \ldots, \theta_k$ is a base and thus linearly independent the inverse exists. Introducing

$$\textcircled{H} = \begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_k^T \end{bmatrix} \tag{11.24}$$

we get the solution

$$\Lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{bmatrix} = -(\textcircled{H}\textcircled{H}^T)^{-1}\textcircled{H}\theta_0 \tag{11.25}$$

The formulas (11.24) and (11.25) can be connected with (11.20) to get

$$\hat{\theta} = \theta_0 + \begin{bmatrix} \theta_1 & \cdots & \theta_k \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{bmatrix}$$

$$= \theta_0 + \textcircled{H}^T \Lambda$$

$$= \theta_0 - \textcircled{H}^T(\textcircled{H}\textcircled{H}^T)^{-1}\textcircled{H}\theta_0 \tag{11.26}$$

$$\hat{\theta} = [I - \textcircled{H}^{+}\textcircled{H}]\theta_0 \qquad (11.27)$$

Geometrically this means that $\hat{\theta}$ is the orthogonal projection of θ_0 on $N(\textcircled{H})$. An alternative characterization of $\hat{\theta}$ is then the following: $\hat{\theta}$ minimizes $|| \hat{\theta} - \theta_0 ||$ with the constraints $(\hat{\theta}|\theta_i) = 0$, $i = 1, \ldots, k$.

12. APPLICATION TO IDENTIFICATION IV: NUMERICAL EXAMPLES.

The main intention with this chapter is to give some computational results illustrating the theory developed in chapter 11. All the computations are made on the CD 3600.

In the first example a first order system is treated. Identifications are made with models of order 1, 2, 3. In the second example the order of the system was 3. The orders of the models were 1, 2, 3, 4. The third example differs from the others. Theoretically we will have quite different results when there is noise in the system compared with a noisefree system. What will happen in a computation, if the noise is small? Will there be a smooth change from one result to another? When will this change occur? By varying EPS compared with the variance σ of the noise the third example is intended to illustrate these questions and to give some insight of how the program treats these cases.

In all the examples the input signal was a PRBS (Pseudo Random Binary Signal). The actual period was 83. The noise signal was a sequence of normal $N(0,\sigma)$ variables. The number N was 96.

Example 1

The system was

$$y(t) + 0.5y(t-1) = u(t-1) + e(t)$$

The parameter σ was 1, 10^{-1}, ..., 10^{-4} and 0. The system was identified with models of order one, two and three. The value of EPS was 10^{-8}. The result is listed in tables 1 - 3.

The loss function decreases just a little when the order of the model is increased, as it should since the true order of the system is one.

When the order of the model is one, table 1 shows that the estimated parameters become better when σ is decreased.

The effect is not always true with models of higher order. For instance when $\sigma = 10^{-3}$ we have rather great differences between estimated and true parameters. The reason is maybe the following: The errors from the estimation is $\phi^{\dagger}e$ (11.6). If σ is great then e is great

and the error vector $\phi^{\dagger}e$ also. When σ is smaller, e becomes smaller, but ϕ becomes more ill-conditioned, and thus $\phi^{\dagger}e$ may increase. It is not impossible the balancing between small e and ill-conditioned ϕ causes the phenomena when $\sigma = 10^{-3}$.

Table 2 shows that we get right values of the parameters when a model of right order is used (no noise).

In Table 3 expected and computed values of the parameters are shown. In the case $\sigma = 0$, $n = 2, 3$. The expected parameters are calculated from (11.26). The accordance is very good.

In some other simulations of the same system ($\sigma = 0$, $n = 2, 3$) we did not get these parameter estimates. Nevertheless, we obtained approximate factors in common. Further we obtained rank $\phi = 2n$. This situation can be geometrically understood in the following way.

We will minimize

$$V = || \phi\theta - Y ||^2$$

If rank $\phi < 2n$ we know that there is no unique minimum. Geometrically this means a horizontal valley, where all the points in the bottom minimizes V. With the pseudoinverse $\phi^{\dagger}$ we pick up the one of them, which is nearest to the "origin". Now if computational round-off error disturbs V a little, it may cause, that the disturbed V has a unique minimum. If the disturbance is small, this new minimum is approximately in the bottom of the original valley, but it may very well differ from the first point we obtained.

Table 1 - Identified parameters from example 1.

		Parameters	a_1	a_2	a_3	b_1	b_2	b_3	Value of loss function
Example 1		True values	0.5	-	-	1.0	-	-	
σ	n	Estimated parameters	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	
	1		0.488	-	-	0.960	-	-	85.7
1	2		0.396	-0.05	-	0.907	-0.17	-	84.2
	3		0.412	-0.113	-0.040	0.915	-0.173	-0.126	83.5
	1		0.506	-	-	0.992	-	-	$6.92 \cdot 10^{-1}$
10^{-1}	2		0.515	0.0009	-	0.992	0.012	-	$6.89 \cdot 10^{-1}$
	3		0.516	0.092	0.048	0.992	0.012	0.090	$6.83 \cdot 10^{-1}$
	1		0.502	-	-	1.002	-	-	$8.95 \cdot 10^{-3}$
10^{-2}	2		0.524	0.012	-	1.002	0.022	-	$8.85 \cdot 10^{-3}$
	3		0.528	-0.029	-0.007	1.002	0.026	-0.015	$8.83 \cdot 10^{-3}$
	1		0.500	-	-	1.000	-	-	$10.72 \cdot 10^{-5}$
10^{-3}	2		0.331	-0.085	-	1.000	-0.169	-	$10.40 \cdot 10^{-5}$
	3		0.337	-0.160	-0.039	1.000	-0.163	-0.078	$10.33 \cdot 10^{-5}$
	1		0.500	-	-	1.000	-	-	$8.60 \cdot 10^{-7}$
10^{-4}	2		0.553	0.026	-	1.000	0.052	-	$8.54 \cdot 10^{-7}$
	3		0.565	0.185	0.076	1.000	0.065	0.153	$8.32 \cdot 10^{-7}$
	1		0.500	-	-	1.000	-	-	0.0
0	2	see also tab. 2	0.278	-0.111	-	1.000	-0.222	-	$5.41 \cdot 10^{-19}$
	3	see also tab. 3	0.266	-0.065	0.026	1.000	-0.234	0.052	$4.16 \cdot 10^{-20}$

Table 2 - Identified parameters from example 1.
No noise. Correct order of the model.

n	True parameters		Estimated parameters	
1	a_1	0.5	$\hat{a}_1$	0.50000 00000
	b_1	1.0	$\hat{b}_1$	1.00000 00000

Table 3 - Identified parameters from example 1.
No noise. Too high an order of the model.

n	Expected parameters		Estimated parameters	
2	a_1	0.27777 77777...	$\hat{a}_1$	0.27777 77777 9
	a_2	-0.11111 11111...	$\hat{a}_2$	-0.11111 11111 2
	b_1	1.0	$\hat{b}_1$	1.00000 00000 1
	b_2	-0.22222 22222...	$\hat{b}_2$	-0.22222 22222 4
3	a_1	0.26623 37662 3	$\hat{a}_1$	0.26623 37662 4
	a_2	-0.06493 50649 35	$\hat{a}_2$	-0.06493 50649 35
	a_3	0.02597 40259 74	$\hat{a}_3$	0.02597 40259 78
	b_1	1.0	$\hat{b}_1$	1.00000 00000
	b_2	-0.23376 62337 7	$\hat{b}_2$	-0.23376 62337 6
	b_3	0.05194 80519 48	$\hat{b}_3$	0.05194 80519 46

Example 2

The system was

$$y(t) + 0.5y(t-1) = u(t-1) - 1.1u(t-1) + 0.24u(t-2) + e(t)$$

The parameter σ was 1.0, 10^{-2}, 10^{-4} and 0, and the value of EPS was 10^{-8}. Identifications were made with models of orders one to four. The results are given in tables 4 - 6.

When $\sigma \neq 1$ it is very evident that the loss function does not decrease very much when n is increased from 3 to 4. Looking on the loss functions in the case $\sigma = 1$ one would believe that the order of the system is two. When $n = 2$ we have also very good estimates of the parameters.

It is also remarkable that the estimates when $\sigma = 10^{-2}$, 10^{-4} and 0 for models with $n = 1$, $n = 2$ are nearly just the same.

In table 5 it is shown that $\sigma = 0$, $n = 3$ will give us the correct values of the parameters. The errors are now approx. 10^{-8} and greater than in example 1, when n was 1.

Expected and computed values of parameters when $\sigma = 0$, $n = 4$ are given in table 6. The expected parameters are calculated from (11.26). The differences between expected and computed values are $\sim 10^{-8}$, which is a quite acceptable result.

Table 4 - Identified parameters from example 2.

		Parameters	a_1	a_2	a_3	a_4
		True values	0.5	-	-	-
σ	n	Estimated values	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$
1	1		0.713	-	-	-
	2		0.535	-0.017	-	-
	3		0.409	-0.091	-0.028	-
	4		0.416	-0.103	0.053	-0.024
10^{-2}	1		0.797	-	-	-
	2		0.731	0.136	-	-
	3		0.513	0.011	0.003	-
	4		0.495	0.002	0.004	0.000
10^{-4}	1		0.797	-	-	-
	2		0.731	0.136	-	-
	3		0.500	-0.000	-0.000	-
	4		0.553	0.027	-0.000	0.000
0	1		0.797	-	-	-
	2		0.731	0.136	-	-
	3	See also table 5	0.500	0.000	0.000	-
	4	See also table 6	0.746	0.123	0.000	0.000

Table 4 (Contd.)

σ	n	b_1	b_2	b_3	b_4	Value of loss function
		1.0	-1.1	0.24	-	
		$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	
1	1	0.975	-	-	-	$1.97 \cdot 10^{2}$
	2	0.927	-1.156	-	-	$0.86 \cdot 10^{2}$
	3	0.911	-1.274	0.237	-	$0.84 \cdot 10^{+2}$
	4	0.915	-1.269	0.214	0.011	$0.83 \cdot 10^{2}$
10^{-2}	1	1.002	-	-	-	$8.06 \cdot 10^{1}$
	2	0.998	-0.869	-	-	$1.25 \cdot 10^{-1}$
	3	0.999	-1.087	0.230	-	$6.69 \cdot 10^{-3}$
	4	0.999	-1.106	0.251	-0.005	$6.68 \cdot 10^{-3}$
10^{-4}	1	1.003	-	-	-	$8.05 \cdot 10^{1}$
	2	0.999	-0.869	-	-	$1.12 \cdot 10^{-1}$
	3	1.000 02	-1.100	0.240	-	$8.69 \cdot 10^{-7}$
	4	1.000 02	-1.047	0.181	0.012	$8.68 \cdot 10^{-7}$
0	1	1.003	-	-	-	$8.05 \cdot 10^{1}$
	2	0.999	-0.869	-	-	$1.13 \cdot 10^{-1}$
	3	1.000	-1.100	0.240	-	$1.57 \cdot 10^{-16}$
	4	1.000	-0.854	-0.030	0.059	$2.32 \cdot 10^{-14}$

Table 5 - Identified parameters from example 2.
No noise. Correct order of the model.

n	True parameters		Estimated parameters	
3	a_1	0.5	$\hat{a}_1$	0.49999 99786
	a_2	0.0	$\hat{a}_2$	0.00000 00155
	a_3	0.0	$\hat{a}_3$	-0.00000 00029
	b_1	1.0	$\hat{b}_1$	0.99999 99999
	b_2	-1.1	$\hat{b}_2$	-1.10000 00215
	b_3	0.24	$\hat{b}_3$	0.24000 00187

Table 6 - Identified parameters from example 2.
No noise. Too high an order of the model.

n	Expected parameters		Estimated parameters	
4	a_1	0.74562 20150	$\hat{a}_1$	0.74562 19990
	a_2	0.12281 10075	$\hat{a}_2$	0.12281 10925
	a_3	0.0	$\hat{a}_3$	-0.00000 00593
	a_4	0.0	$\hat{a}_4$	-0.00000 00131
	b_1	1.0	$\hat{b}_1$	0.99999 99991
	b_2	-0.85437 79850	$\hat{b}_2$	-0.85437 80029
	b_3	-0.03018 42165	$\hat{b}_3$	-0.03018 42743
	b_4	0.05894 92836	$\hat{b}_4$	0.05894 93409

Example 3

This example was intended to illustrate the influence of EPS. If EPS is greater than σ then maybe the result is approximately as expected for a noiseless system.

The system was the same as in example 1

$$y(t) + 0.5y(t-1) = u(t-1) + e(t)$$

The parameter σ was $10^{-2}(10^{-2})10^{-6}$ and EPS was $\sigma/5$, σ and 5σ for every value of σ. Models of orders one, two and three were used. The result is shown in table 7.

For all used σ rank ϕ = 2n if EPS is less than σ , while rank ϕ = = n + max(p,r) = n+1 when EPS is greater than σ. The loss functions are a little greater with the greatest values of EPS. However, the accuracy (unfortunately defined in different ways) is better with the great values of EPS.

In the table it is shown that the greatest values of EPS results in estimates which are approximately those which would be got with $\sigma = 0$. A quantitative reason for this fact is the following:

Consider the matrix ϕ. Let $n_0 = n + \max(p,r)$. When EPS is great then rank ϕ is considered as n_0. This means that A_1 in the program contains n_0 column vectors from ϕ. Let ϕ_2 be the limit of ϕ when $\sigma \to 0$. (The noise is $e(t) = \sigma e_1(t)$, where $e_1(t) \in N(0,1)$ are assumed not to be changed.) Rank $\phi_2 = n_0$ according to theorem (11.4). The matrices A and A_1 in the algorithm are denoted ϕ_2 and ϕ_{21} in this case, (limit when $\sigma \to 0$), resp. $\phi_1 = \phi$ and ϕ_{11} with EPS great. Let us assume σ small. Then $\phi_2 \approx \phi_1$. If $\phi_{21} \approx \phi_{11}$ then it is clear from the algorithm that $\phi_2^\dagger \approx \phi_1^\dagger$ which implies $\hat{\theta}_2 \approx \hat{\theta}_1$. One expects that the accordance $\hat{\theta}_2 \approx \hat{\theta}_1$ grows with decreasing σ.

The different models have been compared with the noisefree system ($\sigma = 0$) in the following way:

A new PRBS signal has been used as input, and the outputs have been compared. The loss function 2 is used as a measure of the accordance. It is computed as the sum of the squares of the differences between the outputs from the system and from the model.

$$\text{Loss function 2} = \sum_{t=1}^{50} \left| y_{SYST}(t) - Y_{MODEL}(t) \right|^2$$

When $\sigma = 10^{-2}$ the new loss function indicates that small values of EPS will give the best result. When $\sigma = 10^{-6}$, n = 2, 3, however, the smallest values of EPS fail. Probably ϕ is too ill-conditioned in this case, since the factorization $GG^T = \phi^T\phi$ fails. In such a case it seems to be a good solution to choose EPS great enough to give a well-conditioned matrix of lower rank. In another simulation of the same systems with the same values of σ and EPS the procedure succeeded, but the accuracy was just 0.3.

Table 7 - Identified parameters from example 3.

σ	EPS	n	rank ϕ	Loss function	Accuracy	Loss function 2
10^{-2}	$\sigma/5$	1	2	$8.56 \cdot 10^{-3}$	10^{-3}	$2.9 \cdot 10^{-5}$
	σ	2	4	$8.41 \cdot 10^{-3}$	0.1	$5.4 \cdot 10^{-5}$
		3	6	$8.35 \cdot 10^{-3}$	0.1	$5.2 \cdot 10^{-5}$
	5σ	1	2	$8.56 \cdot 10^{-3}$	10^{-3}	$2.9 \cdot 10^{-5}$
		2	3	$8.54 \cdot 10^{-3}$	10^{-2}	$7.1 \cdot 10^{-5}$
		3	4	$8.62 \cdot 10^{-3}$	10^{-2}	$8.0 \cdot 10^{-5}$
10^{-4}	$\sigma/5$	1	2	$6.63 \cdot 10^{-7}$	10^{-5}	$2.3 \cdot 10^{-5}$
	σ	2	4	$6.59 \cdot 10^{-7}$	10^{-2}	$2.0 \cdot 10^{-8}$
		3	6	$6.47 \cdot 10^{-7}$	10^{-1}	$2.4 \cdot 10^{-8}$
	5σ	1	2	$6.63 \cdot 10^{-7}$	10^{-5}	$2.3 \cdot 10^{-8}$
		2	3	$7.13 \cdot 10^{-7}$	10^{-4}	$1.8 \cdot 10^{-8}$
		3	4	$7.04 \cdot 10^{-7}$	10^{-3}	$2.1 \cdot 10^{-8}$
10^{-6}	$\sigma/5$	1	2	$9.17 \cdot 10^{-11}$	10^{-7}	$4.7 \cdot 10^{-12}$
	σ	2	4	FAILED	-	-
		3	6	FAILED	-	-
	5σ	1	2	$9.17 \cdot 10^{-11}$	10^{-7}	$4.7 \cdot 10^{-12}$
		2	3	$9.67 \cdot 10^{-11}$	10^{-6}	$6.3 \cdot 10^{-12}$
		3	4	$9.68 \cdot 10^{-11}$	10^{-5}	$7.3 \cdot 10^{-12}$

Accuracy:

1. Rank $\phi = 2n$ Accuracy = $\max(|\hat{a}_1 - a_1|, \ldots, |\hat{b}_n - b_n|)$

2. Rank $\phi < 2n$ Accuracy = $\max\left(\left|\frac{\hat{a}_1 - \bar{a}_1}{\bar{a}_1}\right|, \ldots, \left|\frac{\hat{b}_n - \bar{b}_n}{\bar{b}_n}\right|\right)$,

where $\bar{a}_1, \ldots, \bar{b}_n$ are expected estimates for a noiseless system (see table 3).

ACKNOWLEDGEMENTS.

The author wants to express his thanks to prof. K. Eklund, civ.ing. Per Hagander and civ.ing. Ivar Gustavsson. Their helpful advices and suggestions have been of great value. I am also grateful to Mrs. G. Christensen, who typed the manuscript, and to Mr. Bengt Lander, who drew the figures.

REFERENCES.

|1| F.L. Bauer: Elimination with Weighted Row Combinations for Solving Linear Equations and Least Squares Problems, Num. Math., 7 (1965).

|2| A. Ben-Israel, A. Charnes: Contributions to the Theory of Generalized Inverses, J. Soc. Indust. Appl. Math., 11 (1963).

|3| A. Ben-Israel, D. Cohen: On Iterative Computation of Generalized Inverses and Associated Projection, SIAM J. Numer. Anal., 3 (1966).

|4| A. Ben-Israel, S.J. Wersan: An Elimination Method for Computing the Generalized Inverse of an Arbitrary Complex Matrix, J. Assoc. Comp. Mach., 10 (1963).

|5| Å. Björck: Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization, BIT, 7 (1967).

|6| Å. Björck: Iterative Refinement of Linear Least Squares Solutions I, BIT, 7 (1967).

|7| Å. Björck: Iterative Refinement of Linear Least Squares Solutions II, BIT, 8 (1968).

|8| R.E. Cline: Representations for the Generalized Inverse of a Partitioned Matrix, J. Soc. Indust. Appl. Math., 12 (1964).

|9| R.E. Cline: Representations for the Generalized Inverse of Sums of Matrices, SIAM J. Numer. Anal., Serie B, 2 (1965).

|10| R. Deutsch: Estimation Theory, Prentice-Hall, 1965.

|11| G. Forsythe, C.B. Moler: Computer Solution of Linear Algebraic Systems, Prentice-Hall, 1967.

|12| G.H. Golub, W. Kahan: Calculating the Singular Values and Pseudo-inverse of a Matrix, SIAM J. Numer. Anal., Serie B, 2(1965)

|13| G.H. Golub, C. Reinsch: Singular Value Decomposition and Least Squares Solutions. Technical report No. CS133, 1969, Computer Science Department, Stanford Univ.

|14| G.H. Golub, J.H. Wilkinson: Iterative Refinement of Least Squares Solution, Num. Math., 9 (1966).

|15| T.N.E. Greville: The Pseudoinverse of a Rectangular or Singular Matrix and Its Application to the Solution of Systems of Linear Equations, SIAM Review, 1 (1959).

|16| T.N.E. Greville: Some Applications of the Pseudoinverse of a Matrix, SIAM Review, 2 (1960).

|17| T.N.E. Greville: Note on the Generalized Inverse of a Matrix Product, SIAM Review, 8 (1966).

|18| F.A. Graybill, C.D. Meyer, R.J. Painter: Note on the Computation of the Generalized Inverse of a Matrix, SIAM Review, 8 (1966).

|19| R.E. Kalman, T.S. Englar: A User's Manual for the Automatic Synthesis Program, 1966, Washington D.C.

|20| D.Q. Mayne: An Algorithm for the Calculation of the Pseudo-Inverse of a Singular Matrix, Comp. J., 9 (1966).

|21| D.Q. Mayne: On the Calculation of Pseudo-Inverses, IEEE Trans. Automatic Control (USA), Vol. AC-14, (1969).

|22| E.E. Osborne: On Least Squares Solutions of Linear Equations, J. Assoc. Comp. Mach., 8 (1961).

|23| E.E. Osborne: Smallest Least Squares Solution of Linear Equations, SIAM J. Numer. Anal., Serie B, 2 (1965).

|24| J.B. Rosen: Minimum and Basic Solutions to Singular Linear Systems, J. Soc. Indust. Appl. Math., 12 (1964).

|25| R.P. Tewarson: A Computational Method for Evaluating Generalized Inverses.

|26| R.P. Tewarson: On Computing Generalized Inverses, Computing, 4 (1969).

|27| L.A. Zadeh, C.A. Desoer: Linear System Theory, McGraw-Hill, 1963.

|28| K.J. Åström: Lectures on the Identification Problem - The Least Squares Method, Report 6806, Lund Inst. of Technology, Div. of Automatic Control, Lund.

APPENDIX A

RECURSIVE EQUATIONS FOR P_N AND B_N.

Now we want to derive recursive equations for P_N and B_N. First we repeat some equations from chapter 10. Sometimes we will drop the indexes of P_N, B_N, ϕ_N, and φ_{N+1} to simplify the expressions.

$$P_N = I - \phi_N^{\dagger}\phi_N \tag{A.1}$$

$$B_N = \phi_N^{\dagger}\phi_N^{\dagger T} \tag{A.2}$$

Case i)

$$\phi_{N+1}^{\dagger} = \left[\begin{array}{c} \phi \\ \hline \varphi^T \end{array}\right]^{\dagger} = \left[\begin{array}{c:c} \phi^{\dagger} - \dfrac{P\varphi\varphi^T\phi^{\dagger}}{\varphi^T P \varphi} & \dfrac{P\varphi}{\varphi^T P \varphi} \end{array}\right] \tag{A.3}$$

Case ii)

$$\phi_{N+1}^{\dagger} = \left[\begin{array}{c} \phi \\ \hline \varphi^T \end{array}\right]^{\dagger} = \left[\begin{array}{c:c} \phi^{\dagger} - \dfrac{B\varphi\varphi^T\phi^{\dagger}}{1+\varphi^T B \varphi} & \dfrac{B\varphi}{1+\varphi^T B \varphi} \end{array}\right] \tag{A.4}$$

Introduce the vectors

$$c_N = P_N\varphi_{N+1} \tag{A.5}$$

$$d_N = B_N\varphi_{N+1} \tag{A.6}$$

Now we get

Case i)

$$P_{N+1} = I - \left[\begin{array}{c:c} \phi^{\dagger} - \dfrac{c_N\varphi^T\phi^{\dagger}}{\varphi^T c_N} & \dfrac{c_N}{\varphi^T c_N} \end{array}\right]\left[\begin{array}{c} \phi \\ \hline \varphi^T \end{array}\right]$$

$$= I - \phi^{\dagger}\phi + \frac{c_N \varphi^T \phi^{\dagger}\phi}{\varphi^T c_N} - \frac{c_n \varphi^T}{\varphi^T c_N}$$

$$= P_N - \frac{c_N \varphi^T}{\varphi^T c_N}(I - \phi^{\dagger}\phi)$$

$$= P_N - \frac{c_N \varphi^T P_N}{\varphi^T c_N}$$

$$P_{N+1} = P_N - \frac{c_N c_N^T}{\varphi_{N+1}^T c_N} \qquad (A.7)$$

We have used the fact that P_N is symmetric, which follows from (3.7).

Case ii)

$$P_{N+1} = I - \left[\phi^{\dagger} - \frac{d_N \varphi^T \phi^{\dagger}}{1+\varphi^T d_N} \;\vdots\; \frac{d_N}{1+\varphi^T d_N}\right]\left[\begin{array}{c}\phi \\ \hline \varphi^T\end{array}\right]$$

$$= I - \phi^{\dagger}\phi + \frac{d_N \varphi^T \phi^{\dagger}\phi}{1+\varphi^T d_N} - \frac{d_N \varphi^T}{1+\varphi^T d_N}$$

$$= P_N - \frac{d_N \varphi^T}{1+\varphi^T d_N}(I - \phi^{\dagger}\phi)$$

$$= P_N - \frac{d_N \varphi^T P_N}{1+\varphi^T d_N}$$

$$P_{N+1} = P_N - \frac{d_N c_N^T}{1+\varphi_{N+1}^T d_N} \tag{A.8}$$

Now we calculate the corresponding formulas for B_N.

Case i)

$$B_{N+1} = \left[\phi^\dagger - \frac{c_N \varphi^T \phi^\dagger}{\varphi^T c_N} \;\vdots\; \frac{c_N}{\varphi^T c_N}\right] \left[\begin{array}{c} \phi^{\dagger T} - \dfrac{(c_N \varphi^T \phi^\dagger)^T}{\varphi^T c_N} \\ \hline \dfrac{c_N^T}{\varphi^T c_N} \end{array}\right] =$$

$$= \phi^\dagger \phi^{\dagger T} - \frac{c_N \varphi^T \phi^\dagger \phi^{\dagger T}}{\varphi^T c_N} - \frac{\phi^\dagger \phi^{\dagger T} \varphi c_N^T}{\varphi^T c_N} +$$

$$+ \frac{c_N \varphi^T \phi^\dagger \phi^{\dagger T} \varphi c_N^T}{(\varphi^T c_N)^2} + \frac{c_N c_N^T}{(\varphi^T c_N)^2}$$

$$= B_N - \frac{c_N \varphi^T B_N}{\varphi^T c_N} - \frac{B_N \varphi c_N^T}{\varphi^T c_N} + \frac{c_N \varphi^T B_N \varphi c_N^T}{(\varphi^T c_N)^2} + \frac{c_N c_N^T}{(\varphi^T c_N)^2}$$

Since $\varphi^T B_N \varphi$ is a scalar we get

$$B_{N+1} = B_N - \frac{c_N d_N^T + d_N c_N^T}{\varphi_{N+1}^T c_N} + \frac{c_N c_N^T}{(\varphi_{N+1}^T c_N)^2} (1 + \varphi_{N+1}^T d_N) \tag{A.9}$$

Case ii)

$$B_{N+1} = \left[\phi^{\dagger} - \frac{d_N \varphi^T \phi^{\dagger}}{1+\varphi^T d_N} \;\vdots\; \frac{d_N}{1+\varphi^T d_N}\right] \left[\begin{array}{c} \phi^{\dagger T} - \dfrac{(d_N \varphi^T \phi^{\dagger})^T}{1+\varphi^T d_N} \\ \hline \dfrac{d_N^T}{1+\varphi^T d_N} \end{array}\right]$$

$$= \phi^{\dagger}\phi^{\dagger T} - \frac{\phi^{\dagger}\phi^{\dagger T}\varphi d_N^T}{1+\varphi^T d_N} - \frac{d_N \varphi^T \phi^{\dagger}\phi^{\dagger T}}{1+\varphi^T d_N} + \frac{d_N \varphi^T \phi^{\dagger}\phi^{\dagger T}\varphi d_N^T}{(1+\varphi^T d_N)^2} + \frac{d_N d_N^T}{(1+\varphi^T d_N)^2}$$

$$= B_N - \frac{B_N \varphi d_N^T}{1+\varphi^T d_N} - \frac{d_N d_N^T}{1+\varphi^T d_N} + \frac{d_N d_N^T}{(1+\varphi^T d_N)^2}(1 + \varphi^T B_N \varphi)$$

$$= B_N - 2\frac{d_N d_N^T}{1+\varphi^T d_N} + \frac{d_N d_N^T}{1+\varphi^T d_N}$$

$$B_{N+1} = B_N - \frac{d_N d_N^T}{1+\varphi_{N+1}^T d_N} \qquad (A.10)$$

The initial values P_0 and B_0 remain to be determined. We will show that

$$P_0 = I \qquad (A.11)$$

$$B_0 = 0 \qquad (A.12)$$

a) $\underline{\phi_1} \neq \underline{0}$

(A.7) gives

$$P_1 = I - \frac{\phi_1^T \phi_1}{\phi_1 \phi_1^T}$$

A direct computation gives

$$\phi_1^{\dagger} = \phi_1^T(\phi_1\phi_1^T)^{-1} = \frac{\phi_1^T}{\phi_1\phi_1^T}$$

and the use of (A.1) will give

$$P_1 = I - \frac{\phi_1^T\phi_1}{\phi_1\phi_1^T}$$

(A.9) gives

$$B_1 = \frac{c_0c_0^T}{(\phi_1c_0)^2} = \frac{\phi_1^T\phi_1}{(\phi_1\phi_1^T)^2}$$

The direct computation gives

$$B_1 = \frac{\phi_1^T}{\phi_1\phi_1^T} \cdot \left[\frac{\phi_1^T}{\phi_1\phi_1^T}\right]^T = \frac{\phi_1^T\phi_1}{(\phi_1\phi_1^T)^2}$$

b) $\underline{\phi_1 = 0}$

(A.8) gives $P_1 = I$.

$\phi_1 = 0$ implies $\phi_1^{\dagger} = 0$. Thus a direct computation gives the same result $P_1 = I$.(A.10) gives $B_1 = 0$ in accordance with the direct computation

$$B_1 = \phi_1^{\dagger}\phi_1^{\dagger T} = 0$$

Summary

The equations (A.5) to (A.11) are the iterative formulas for P_N and B_N.

APPENDIX B.

SOME THEOREMS ABOUT RANK ϕ.

In this appendix we will derive some theorems about rank ϕ, which are used in chapter 11.

After repeating some notations we start with two lemmas. Then systems with noise are treated and finally we take systems without noise in consideration.

Let us introduce the system:

$$y(t) + a_1 y(t-1) + \ldots + a_p y(t-p) =$$

$$= b_1 u(t-1) + \ldots + b_r u(t-r) + e(t) \qquad \text{(B.1)}$$

and a corresponding noisefree system:

$$y(t) + a_1 y(t-1) + \ldots + a_p y(t-p) =$$

$$b_1 u(t-1) + \ldots + b_r u(t-r) \qquad \text{(B.2)}$$

We will have use of the following two models:

$$y(t) + \hat{a}_1^0 y(t-1) + \ldots + \hat{a}_p^0 y(t-p) =$$

$$= \hat{b}_1^0 u(t-1) + \ldots + \hat{b}_r^0 u(t-r) \qquad \text{(B.3)}$$

$$y(t) + \hat{a}_1 y(t-1) + \ldots + \hat{a}_n y(t-n) =$$

$$= \hat{b}_1 u(t-1) + \ldots + \hat{b}_n u(t-n) \qquad \text{(B.4)}$$

It is assumed that $n \geq \max(p,r)$.

The ϕ matrix, which corresponds to the model (B.3), will be denoted by ϕ_0.

$$\phi_0 = \begin{bmatrix} y(n) & \dots & y(n-p+1) & u(n) & \dots & u(n-r+1) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(n+N-1) & \dots & y(n-p+N) & u(n+N-1) & \dots & u(n-r+N) \end{bmatrix} \qquad \text{(B.5)}$$

In the following ϕ will mean the ϕ matrix, which corresponds to the model (B.4).

Now let us start with two lemmas, which are useful later.

Lemma B.1

Let ξ and η be continuous, independent random variables with finite distribution functions. Then the probability $P(\xi=\eta)$ is zero.

Proof

The proof is a straightforward computation.

$$P(\xi=\eta) = \iint_{x=y} f_{\xi,\eta}(x,y)dxdy = \iint_{x=y} f_{\xi}(x)f_{\eta}(y)dxdy$$

This is an integral over a domain of measure zero. Since the integrand is finite the integral must be zero.

Q.E.D.

Remark

If ξ or η is quite or partly discrete then $f_{\xi}(x)$ resp. $f_{\eta}(y)$ contains dirac distributions and are not finite. If such a dirac distribution gives contributions in the domain of integration the proof above is not valid. Such a situation may also occur if the variables are dependent.

To simplify the expressions in the following we will introduce the concept "backwards independent". Note that this is not a general concept, but it will be used here for practical reasons.

Definition B.1

Let the vector ξ and the matrix A be given by:

$$\xi = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix} \qquad A = \begin{bmatrix} \eta_{11} & \cdots & \eta_{1m} \\ \vdots & & \vdots \\ \eta_{n1} & \cdots & \eta_{nm} \end{bmatrix} \tag{B.6}$$

where all components are random variables. We will say that ξ is backwards independent of A if ξ_i is independent of ξ_j, $j = 1, \ldots, i-1$, and of η_{jk}, $j = 1, \ldots, i$, $k = 1, \ldots, m$. This must be true for $i = 1, \ldots, n$.

Now we have:

Lemma 2

Given ξ and A by (B.6), where all the random variables are assumed to be continuous with finite distribution functions. The matrix A_0 given by

$$A_0 = \begin{bmatrix} \eta_{11} & \cdots & \eta_{1m} \\ \vdots & & \vdots \\ \eta_{n-1,1} & \cdots & \eta_{n-1,m} \end{bmatrix} \tag{B.7}$$

is obtained from A by dropping the last row. Now assume that ξ is backwards independent of A and that rank A_0 = m. Then ξ does not belong to the range of A $(\xi \notin R(A))$ with probability one or equivalently rank of $[\, \xi \mid A \,]$ = m+1 with probability one.

Proof

We will compute the probability of the complement (ξ does belong to the range of A)

$$P(\xi \in R(A)) = P(\exists \lambda;\ \xi = A\lambda)$$

where λ is an m vector. If ξ belongs to R(A) there must be such a λ.

Among the (n-1) rows of A_0 there are m linearly independent ones according to the assumptions. These rows determine the vector λ uniquely. We then have:

$$P(\xi \in R(A)) \leq P\left(\xi_n = \begin{bmatrix} \eta_{n1} & \cdots & \eta_{nm} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{bmatrix}\right)$$

Introduce

$$\eta = \begin{bmatrix} \eta_{n1} & \cdots & \eta_{nm} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{bmatrix}$$

According to the construction ξ_n and η are independent. Since Lemma 1 can be used we conclude that $P(\xi \in R(A)) = 0$ and then the lemma is proved.

Q.E.D.

Systems with Noise.

Consider the system (B.1) where as before e(t) is a sequence of independent, equally distributed random variables with zero mean and finite covariance.

Let the model be (B.4) where the case $n > \max(p,r)$ is allowed.

Before stating the theorem we repeat the definition of ϕ.

$$\phi = \begin{bmatrix} -y(n) & \cdots & -y(1) & u(n) & \cdots & u(1) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y(n+N-1) & \cdots & -y(N) & u(n+N-1) & \cdots & u(N) \end{bmatrix} \qquad \text{(B.8)}$$

Introduce the matrix A.

$$A = \begin{bmatrix} u(n) & \dots & u(1) \\ \vdots & & \vdots \\ u(N-1) & \dots & u(N-n) \end{bmatrix} \qquad (B.9)$$

which is a part of the ϕ matrix.

Theorem B.1

With the assumptions of the system and model above the following is true. If rank A = n the rank ϕ = 2n with probability one.

Proof

Let us introduce the vector

$$y_1 = \begin{bmatrix} -y(1) \\ \vdots \\ -y(N-n+1) \end{bmatrix}$$

which we partition as $y_1 = y_{10} + e$. y_{10} is the part of y_1, which depends on earlier values of y and u. According to (B.1) y_1 consists of these two parts, although we cannot partition y_1 numerically, if the noise is not known.

Introduce the matrix B of order (N-n+1) × (n+1) by

$$B = \begin{bmatrix} y_1 & & A \\ & & \\ & u(N) & \dots \; u(N-n+1) \end{bmatrix}$$

We will also use the matrix A_0 of order (N-n+1) × n

$$A_0 = \begin{bmatrix} & A & \\ & & \\ u(N) & \dots & u(N-n+1) \end{bmatrix}$$

Thus we have

$$B = \left[\begin{array}{c|c} y_1 & A_0 \end{array}\right]$$

Since rank A = n we have rank A_0 = n.

Note that also B is a part of ϕ. We can obtain it from A by "adding" a column and a row. We will first show that rank B = n+1 with probability one. The argumentation can then be repeated, "adding" a column and a row each time. By this procedure the whole ϕ matrix will be considered at last.

$$P(\text{rank } B = n+1) = P\{y_1 \notin R(A_0)\} =$$

$$= P\{y_{10} + e \notin R(A_0)\} \leq P\{e \notin \{R(A_0) \oplus R(y_{10})\}\}$$

The inequality holds since we introduce further restrictions of e.

Let us separate two cases.

Case a

$R(y_{10})$ is not a subset of $R(A_0)$. Then let us form the matrix (of order (N-n+1) × (n+1))

$$A_1 = \left[\begin{array}{c|c} y_{10} & A_0 \end{array}\right]$$

We have in this case rank A_1 = n+1. Now e is backwards independent of A_1. By use of lemma 2 we conclude that the probability

$$P\{e \notin R(A_1)\} = P(\text{rank } B = n+1) = 1$$

Case b

$R(y_{10})$ is a subset of $R(A_0)$. We have

$$P(\text{rank } B = n+1) = P\{e \notin R(A_0)\}$$

but rank A_0 = n and e is backwards independent of A_0. Consequently we have P(rank B = n+1) = 1.

Hence in both cases we have P(rank B = n+1) = 1. Since the procedure can be repeated in totally n steps as outlined above the theorem is proved.

Q.E.D.

Systems without Noise.

Now we will consider the case with a system without noise (B.2). Three cases will be treated:

i) model 1 used,
ii) model 2 used n = max(p,r)
iii) model 2 used n > max(p,r)

Case i)

We want to have sufficient conditions on u(t) so that rank ϕ_0 = p+r. We immediately see that it is necessary that the last r column vectors of ϕ_0 should be linearly independent. In many practical cases this condition seems to be sufficient. We now state the following theorem, which is not proved.

Theorem B.2

Given the system (B.2) and the model (B.3). Then there exist input signals u(t), such that rank ϕ_0 = p+r. Further, if the input signal is white noise this holds with probability one.

Case ii)

In this case and in the following we will sometimes assume that a time translation does not change rank ϕ. Let $q^s\phi$ (s a given integer) means that the argument is shifted s steps forward in all the components of ϕ. The relation rank ϕ = rank $q^s\phi$ will then sometimes be assumed. This will always be denoted explicitely as well as the integer s will be given. This relation can be motivated in two ways.

Physically, this is a consequence of stationarity. Secondly, we note that $q^s\phi$ mathematically means shifting the column vectors of ϕ s steps upwards. If now N >> n we can hope that if s is small compared with N this operation does not change rank ϕ. We have many (N) rows. A small number of them (s) will be removed and substituted with another s rows.

Further, introduce the vectors:

$$y_i = \begin{bmatrix} y(i) \\ \vdots \\ y(i+N-1) \end{bmatrix}, \qquad u_i = \begin{bmatrix} u(i) \\ \vdots \\ u(i+N-1) \end{bmatrix} \tag{B.10}$$

Then

$$\phi_0 = \begin{bmatrix} y_n \cdots y_{n-p+1} & u_n \cdots u_{n-r+1} \end{bmatrix} \tag{B.11}$$

The equation of the system (B.2) implies

$$y_t + a_1 y_{t-1} + \ldots + a_n y_{t-n} = b_1 u_{t-1} + \ldots + b_r u_{t-r} \tag{B.12}$$

Introduce $k = n - \min(p,r)$ and the abbreviated ϕ_0 matrix of order $(N-k)*(p+r)$.

$$\phi_{00} = \begin{bmatrix} y(n) & \ldots & y(n-p+1) & u(n) & \ldots & u(n-r+1) \\ \vdots & & & \vdots & & \\ y(n+N-1-k) & \ldots & y(n-p+N-k) & u(n+N-1-k) & \ldots & u(n-r+N-k) \end{bmatrix} \tag{B.13}$$

We have

Theorem B.3

Given the system (B.2) and the model (B.4). Let $n = \max(p,r)$. Assume that u(t) is white noise, rank ϕ_{00} = p+r, rank $q^k\phi$ = rank ϕ. Then rank ϕ = 2n with probability one.

Proof

Let us first consider the case when $p > r$. Then $p = n$, $k = n-r$ are valid. We have

$$\begin{aligned} \text{rank } \phi &= \text{rank } \begin{bmatrix} y_n & \cdots & y_1 & u_n & \cdots & u_1 \end{bmatrix} = \\ &= \text{rank } \begin{bmatrix} y_{n+k} & \cdots & y_{1+k} & u_{n+k} & \cdots & u_{1+k} \end{bmatrix} \end{aligned} \tag{B.14}$$

where we have used the time translation, which was assumed to be allowed.

Now y_{n+k} is a linear combination of $y_{n+k-1}, \cdots, y_k, u_{n+k-1}, \cdots, u_{n+k-r}$. All these vectors except y_k are included in $q^k\phi$. Thus we can substitute y_{n+k} with y_k in (B.14). This procedure can be repeated (k-1) times to get

$$\begin{aligned} \text{rank } \phi &= \text{rank } \begin{bmatrix} y_k & \cdots & y_1 & y_n & \cdots & y_{1+k} & u_{n+k} & \cdots & u_{1+k} \end{bmatrix} = \\ &= \text{rank } \begin{bmatrix} u_{n+k} & \cdots & u_{n+1} & y_n & \cdots & y_1 & u_n & \cdots & u_{k+1} \end{bmatrix} = \\ &= \text{rank } \left[\begin{array}{ccc|c} u_{n+k} & \cdots & u_{n+1} & \phi_0 \end{array}\right] \end{aligned}$$

The rest of the argumentation is similar to the one in the proof of theorem B.1. The vector

$$u'_{n+1} = \begin{bmatrix} u(n+1) \\ \vdots \\ u(n+N-k) \end{bmatrix}$$

is backwards independent of ϕ_{00}. By lemma 2 we conclude that rank $\left[\begin{array}{c|c} u'_{n+1} & \phi_{00} \end{array}\right] = 1 + \text{rank } \phi_{00} = 1 + n + r$. In this way we get rank ϕ = $= k + \text{rank } \phi_{00} = k + n + r = 2n$ by "adding" a column and a row each time.

The case $p < r$ ($k = n-p$, $r = n$) is analogous, so the comments are omitted.

$$\text{rank } \phi = \text{rank } \begin{bmatrix} y_{n+k} & \cdots & y_{k+1} & u_{n+k} & \cdots & u_{k+1} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_k & \cdots & u_1 & y_n & \cdots & y_{k+1} & u_{n+k} & \cdots & u_{k+1} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+k} & \cdots & u_{n+1} & y_n & \cdots & y_{k+1} & u_n & \cdots & u_1 \end{bmatrix}$$

$$= \text{rank } \left[\begin{array}{ccc|c} u_{n+k} & \cdots & u_{n+1} & \phi_0 \end{array}\right]$$

$$= k + (p+r) = n-p+p+n = 2n$$

Thus we have shown that rank ϕ = 2n with probability one.

Q.E.D.

Case iii)

Now we have $n > \max(p,r)$.

Let us define $k = n - \max(p,r)$, $\ell = |p-r|$, $m = \min(p,r)$, $j = \ell+k$. Further define the matrix ϕ_1 of order $(N-m) \times (p+r)$.

$$\phi_1 = \begin{bmatrix} y(n) & \cdots & y(n-p+1) & u(n) & \cdots & u(n-r+1) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(n+N-1-m) & \cdots & y(n-p+N-m) & u(n+N-1-m) & \cdots & u(n-r+N-m) \end{bmatrix} \quad \text{(B.15)}$$

The matrix ϕ_1 is a shortened ϕ_0 matrix, which could be compared with ϕ_{00} (B.13). In fact, they play the same roles in the proofs.

Theorem B.4

Given the system (B.2) and the model (B.4). Let $k = n - \max(p,r) > 0$, $\ell = |p-r|$, $j = k+\ell$. Assume that u(t) is white noise, rank $\phi_1 = p+r$, and rank $q^j\phi$ = rank ϕ. Then rank $\phi = 2n - k = n + \max(p,r)$ with probability one.

Proof

Let us first consider the case $p \geq r$. We have $k = n-p$, $\ell = p-r$, $j = \ell+k = n-r$. Using rank $q^j\phi$ = rank ϕ we get

$$\text{rank } \phi = \text{rank } \begin{bmatrix} y_{n+j} & \cdots & y_{1+j} & u_{n+j} & \cdots & u_{1+j} \end{bmatrix}$$

Now y_{n+j} is a linear combination of $y_{n+j-1} \cdots y_{n+j-p}$, $u_{n+j-1} \cdots u_{n+j-r}$ which all are included in ϕ. Thus we can just drop y_{n+j}. Repeating we get:

$$\text{rank } \phi = \text{rank } \begin{bmatrix} y_{p+j} & \cdots & y_{1+j} & u_{n+j} & \cdots & u_{1+j} \end{bmatrix}$$

We have the inequality rank $\phi \leq p+n$. Now we want to show that with the given assumptions we in fact have an equality.

Just as in theorem B.3 we can substitute y_{p+j} with y_j. Repeat this procedure $(\ell-1)$ times to obtain

$$\text{rank } \phi = \text{rank } \begin{bmatrix} y_j & \cdots & y_{1+j-\ell} & y_{p+j-\ell} & \cdots & y_{1+j} & u_{n+j} & \cdots & u_{1+j} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{p+1+j-\ell} & y_{p+j-\ell} & \cdots & y_{1+j-\ell} & u_{p+j-\ell} & \cdots & u_{1+j} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{p+1+k} & y_{p+k} & \cdots & y_{1+k} & u_{p+k} & \cdots & u_{1+\ell+k} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{n+1} & y_n & \cdots & y_{n-p+1} & u_n & \cdots & u_{n-r+1} \end{bmatrix}$$

$$= \text{rank } \left[\begin{array}{ccc|c} u_{n+j} & \cdots & u_{n+1} & \phi_0 \end{array}\right]$$

Quite analogous to the proof of theorem B.3 we get rank $\phi = j + (p+r)$ by successive use of lemma B.2. Thus rank $\phi = n-r+p+r = n+p = n + \max(p,r)$.

The case $p \leq r$ is analogous, so the comments are omitted. We have in this case

$$p \leq r,\ k = n-r,\ \ell = r-p,\ j = k+\ell = n-p$$

$$\text{rank } \phi = \text{rank } \begin{bmatrix} y_{n+j} & \cdots & y_{1+j} & u_{n+j} & \cdots & u_{1+j} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} y_{r+j} & \cdots & y_{1+j} & u_{n+j} & \cdots & u_{1+j} \end{bmatrix}$$

Note that rank $\phi \leq r+n = n + \max(p,r)$.

$$\text{rank } \phi = \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{r+j+1} & y_{r+j} & \cdots & y_{1+j} & u_{r+j} & \cdots & u_{1+j} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{r+j+1} & u_j & \cdots & u_{1+j-\ell} & y_{p+j} & \cdots & y_{1+j} & u_{r+j} & \cdots & u_{1+j} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{r+j+1} & u_{r+j} & \cdots & u_{r+j-\ell+1} & y_{p+j} & \cdots & y_{1+j} & u_{r+j-\ell} & \cdots & u_{1+j-\ell} \end{bmatrix}$$

$$= \text{rank } \begin{bmatrix} u_{n+j} & \cdots & u_{n+1} & y_n & \cdots & y_{1+n-p} & u_n & \cdots & u_{1+n-r} \end{bmatrix}$$

$$= \text{rank } \left[\begin{array}{ccc|c} u_{n+j} & \cdots & u_{n+1} & \phi_0 \end{array}\right]$$

The repeated use of lemma B.2 gives us rank $\phi = j + p + r = n + r$.

Thus we have shown that rank $\phi = n + \max(p,r)$ with probability one.

Q.E.D.

Remark

As pointed out in the proof we have an inequality rank $\phi \leq n + \max(p,r)$ which was easily established. The theorem gives conditions so that the equality sign holds.

Remark

The case ii), which was considered in theorem B.3, is a special case of case iii). The only differences in the assumptions are that k in theorem B.3 is substituted with m in (B.15) and with j in the time translation relation.

Remark

The weakness of all this appendix is the lack of a proof of theorem B.2, since the rest of the appendix uses this theorem.

Remark

The conclusions of the theorems are mostly valid with milder assumptions. For the numerical examples in chapter 11 a PRBS was used and the results of this appendix were fulfilled.