Notes on Signals and Systems

Marnix Heskamp

Contents

Chapter 1

Linear equations

1.1 Introduction

Assume we have two linear spaces V and W with dimensions n and m, and a linear transform $\boldsymbol{A} : V \to W$, which we can write as

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}$$

Our goal is to solve $\mathbf{x}$ from this equation when both $\boldsymbol{A}$ and $\mathbf{b}$ are known. There are roughly three possibilities:

1) There exist one unique solution
2) There are multiple solutions
3) There are no solutions

When there are multiple solutions, we can ask if some solutions are preverable over others, and if there are no solutions, we can ask if there are vectors which approximate a solution, i.e. vectors $\mathbf{x}$ that when transformed by $\boldsymbol{A}$ end up close to vector $\mathbf{b}$.

Since $\boldsymbol{A}$ is a function, it maps every vector of the domain V to exactly one vector in the co-domain W. In other words, all vectors in V participate in the mapping, and every mapping from V to W is unique. The opposite is in general not true, i.e. there may be vectors in W that are not the image of some vector from V, and as such don't participate in the mapping, and there might be vectors in V that can be reached from different vectors in V, which means that the inverse mapping from W back to V is in general not unique.

These situations can be described by the following two properties:

A transform is *injective* when:
$\boldsymbol{A}\mathbf{x_1} = \boldsymbol{A}\mathbf{x_2} \implies \mathbf{x_1} = \mathbf{x_2}$

which we can easily rewrite as:
$\boldsymbol{A}\mathbf{x} = \mathbf{0} \implies \mathbf{x} = \mathbf{0}$

A transform is *surjective* when:
$\mathbf{b} \in W \implies \mathbf{b} = \boldsymbol{A}\mathbf{x}$ for some $\mathbf{x} \in V$

When both properties hold for a transform $\boldsymbol{A}$ we call the transform *bijective* and for such transforms there exist an inverse $\boldsymbol{A}^{-1}$, and the equation has a unique solution which we can write as:

$$\mathbf{x} = \boldsymbol{A}^{-1}\mathbf{b}$$

When $\boldsymbol{A}^{-1}$ does not exist, this can be caused by $\boldsymbol{A}$ not injective, or not surjective, or neither of both. This does not mean, however, that there is nothing sensible to say about $\mathbf{x}$ because there can still be multiple solutions or approximate solutions, because it turns out that we usually can find subspaces of V and W between which $\boldsymbol{A}$ actually does make a bijective mapping. So if we stay within these subspaces, $\boldsymbol{A}$ is still invertible.

1.1.1 The four fundamental subspaces

In order to understand invertibility of linear transforms better, it is instructive to consider certain sub spaces of the domain and co-domain, as shown in the following figure:

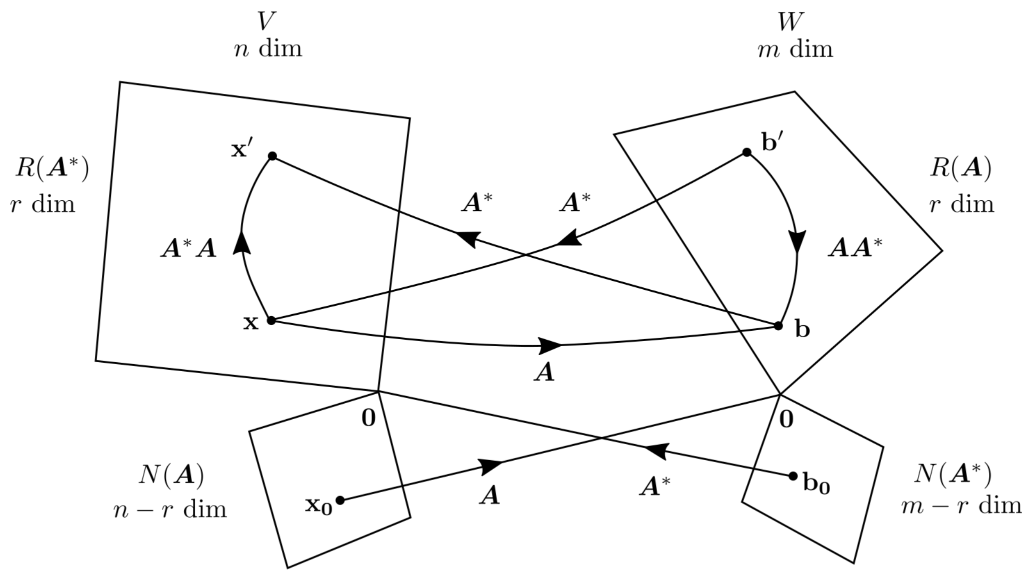

Figure 1.1: The four fundamental subspaces.

1.1.2 The singular value decomposition

Say we have a $m \times n$ matrix $\boldsymbol{A}$ which is (in general) not square, so that (in general) we can not do an eigenvalue decomposition. The next best thing is to consider the eigenvalue decomposition of the products $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$. Since these products are square and symmetrical, all its eigenvalues are non-negative, so that we can write every eigenvalue λ_i as the square of a so called singular value σ_i, i.e. $\lambda_i = \sigma_i^2$. Furthermore, all eigenvectors are orthogonal to each other.

The eigenvector relations are as follows:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^* \ \mathbf{u}_i &= \sigma_i^2 \mathbf{u}_i \quad \text{for } i = 1 \cdots m \\ \boldsymbol{A}^*\boldsymbol{A} \ \mathbf{v}_j &= \sigma_j^2 \mathbf{v}_j \quad \text{for } j = 1 \cdots n \end{aligned}$$

which we can rewrite as the following eigen decomposition:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^* &= \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^* \\ \boldsymbol{A}^*\boldsymbol{A} &= \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^* \end{aligned}$$

in which $\boldsymbol{U}$ and $\boldsymbol{V}$ are an orthogonal matrices in which the columns are formed by the eigenvectors $\mathbf{u}$ and $\mathbf{v}$ respectively, and $\boldsymbol{\Sigma}$ a diagonal matrix with the singular values σ on its diagonal.

It is easy to verify that from this follows:

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \\ \boldsymbol{A}^* &= \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^* \end{aligned}$$

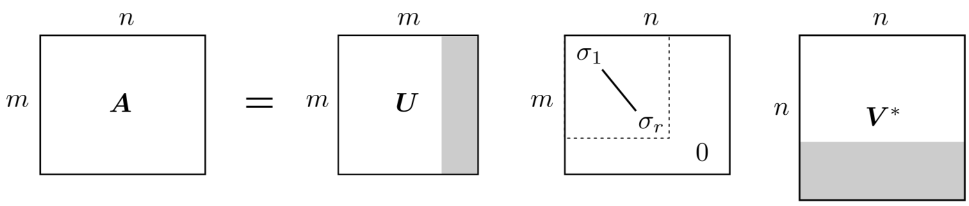

Figure 1.2: The Singular value decomposition.

1.2 The underdetermined system

When we have fewer equations than unknowns in a linear system we call the system under determined. The matrix $\boldsymbol{A}$ of such system has more columns than rows (i.e. a fat matrix). For such matrix it is impossible for all columns to be linearly independent, because we have too many of them. In this section we consider such system with too many columns, but we assume that we don't have too many rows, i.e. we assume that all rows are linearly independent. Such a system is called full (row) rank, and the rank r is equal to the number of rows, i.e. $r = m$. When the rows are not linearly independent (i.e. we also have too many rows) we call the system rank deficient.

An under determined system transforms in some sense a large space to a smaller space, i.e. the domain V has more dimensions than the co-domain W, so we lose dimensions (i.e. degrees of freedom) when applying the transform. This also means that it is inevitable that multiple vectors from V map to the same vector in W, because every vector in V has to participate (because the transform is a function), and in some sense there are just more vectors in V than there are target vectors in W. In other words, the transform is not injective, which means that the null space $N(\boldsymbol{A})$ is not empty; Say we have two vector: $\mathbf{x_1}$ and $\mathbf{x_2}$ which are linearly independent (i.e. not a scalar multiple of each other) and they both map to $\mathbf{b}$. This means that $\boldsymbol{A}(\mathbf{x_1} - \mathbf{x_2}) = \mathbf{0}$, so $\mathbf{x_1} - \mathbf{x_2} \in N(\boldsymbol{A})$.

Figure 1.3 shows the sub space diagram of an under determined system. The domain V is split-up in two subspaces: The space $R(\boldsymbol{A}^*)$, called the column space of $\boldsymbol{A}^*$ which is spanned by the columns of $\boldsymbol{A}^*$, and a space $N(\boldsymbol{A})$ which is called the null space of $\boldsymbol{A}$, which consist of all vectors from the domain which are mapped to zero. Both spaces are orthogonal to each other, i.e. the inner product between every vector from $R(\boldsymbol{A}^*)$ with every vector from $N(\boldsymbol{A})$ is zero.

Every vector $\mathbf{x}$ can be split into two parts: $\mathbf{x} = \mathbf{x_p} + \mathbf{x_0}$, in which $\mathbf{x_p} \in R(\boldsymbol{A}^*)$ and $\mathbf{x_0} \in N(\boldsymbol{A})$. In other words $\mathbf{x}$ can be split into a part which bijectively (and thus invertibly) maps into W and a part which maps to $\mathbf{0}$.

Our goal is again to solve $\mathbf{x}$ (the state) from the equation

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}$$

in which $\boldsymbol{A}$ (the system) and $\mathbf{b}$ (the observation or measurement) are known.

Since we lose dimensions with the transform $\boldsymbol{A}$, we sort of lose information about $\mathbf{x}$ so that in general it becomes impossible to perfectly reconstruct $\mathbf{x}$ based on an observation $\mathbf{b}$. But we can make an estimation $\hat{\mathbf{x}}$ which is optimal in the least-squares sense, which we obtain by applying the so-called pseudo inverse to $\mathbf{b}$:

$$\hat{\mathbf{x}} = \mathbf{x_p} = \boldsymbol{A}_R^+ \mathbf{b}$$

in which $\boldsymbol{A}_R^+ = \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}$ is the *right* pseudo inverse (note that $\boldsymbol{A}\boldsymbol{A}_R^+ = \boldsymbol{I}$).

Note that $\mathbf{x_p}$, which is sometimes called the *particular solution*, is an exact solution of the equation, i.e. $\boldsymbol{A}\mathbf{x_p} = \mathbf{b}$ without any error. But it is not a unique solution, because we can add any vector $\mathbf{x_0}$ from $N(\boldsymbol{A})$, which is called the homogeneous solution, and still have an exact solution:

$$\hat{\mathbf{x}}' = \mathbf{x_p} + \mathbf{x_0}$$

But among these solutions, only $\mathbf{x_p}$ has the smallest norm, which makes it optimal in the least-squares sense.

Figure 1.5 illustrates how the right inverse works in terms of singular values; the m non zero singular vales cancel each other to become the 1s on the diagonal of the identity matrix, and the singular values which are zero are just ignored. And if we look in Figure 1.3 we can see that if we start with an arbitrary vector $\mathbf{b}$ we can make a complete round-trip in the sub-space diagram by first applying $\boldsymbol{A}_R^+$ followed by $\boldsymbol{A}$.

It's even more interesting to see what happens is we apply the right inverse from the *left*, as is shown in Figure 1.6. Now we don't get an identity matrix, but an *projection matrix*:

$$\boldsymbol{P} = \boldsymbol{A}_R^+\boldsymbol{A}$$

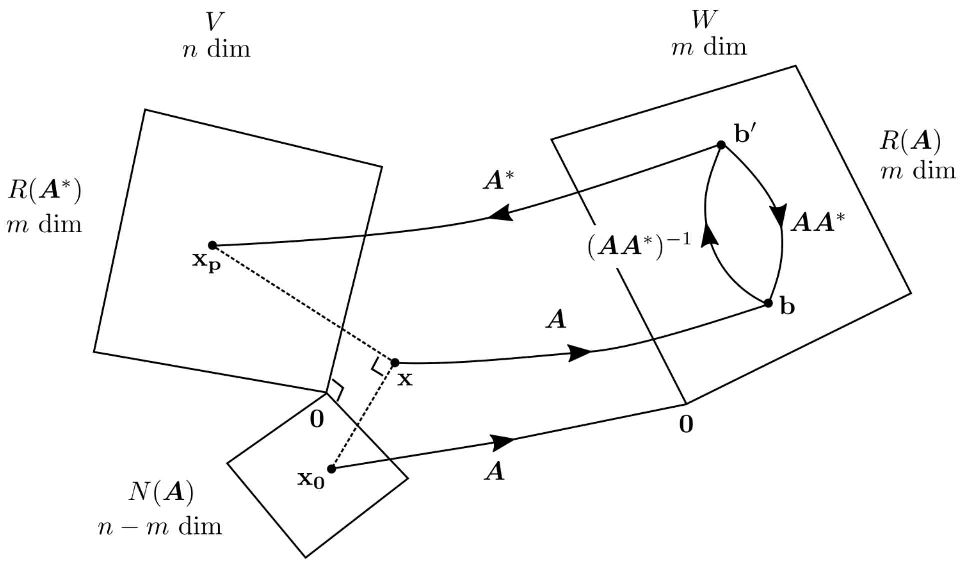

Figure 1.3: The subspaces of an underdetermined system.

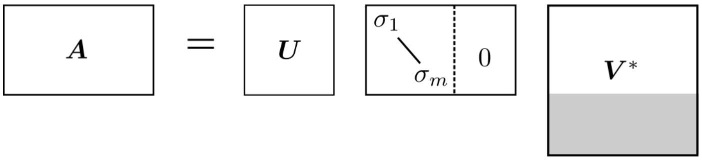

Figure 1.4: Singular value decomposition of an underdetermined system.

A projection matrix has the very interesting property that you can effectively only apply it once; if you repeat it you get the same result back, i.e. $\boldsymbol{P}^k = \boldsymbol{P}$ for $k = 1, 2, \cdots$. In this sense a projection matrix is somewhat similar to an identity matrix. If we look at Figure 1.3 we see that with $\boldsymbol{P}$ we project a vector $\mathbf{x}$ onto the subspace $R(\boldsymbol{A}^*)$, so it turns an arbitrary solution into the minimum norm solution.

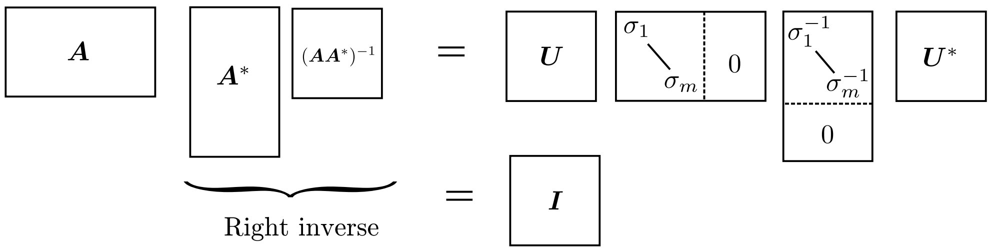

Figure 1.5: Right inverse applied from the right gives the identity.

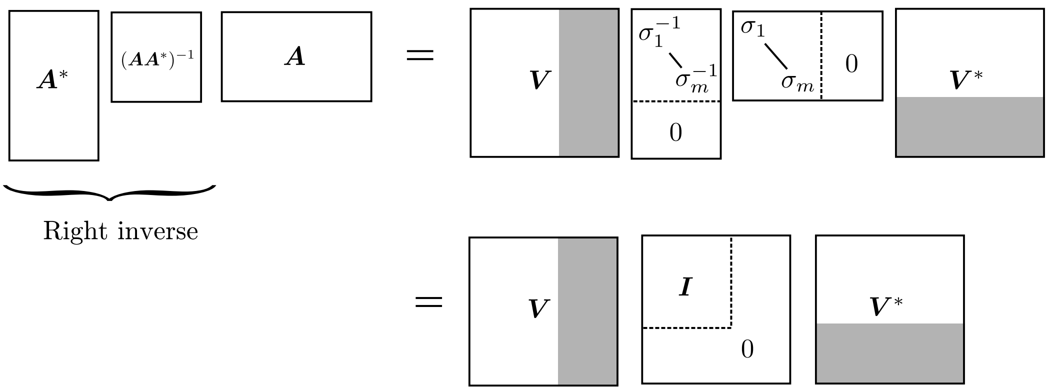

Figure 1.6: Right inverse applied from the left gives an projection onto $R(\boldsymbol{A}^*)$.

1.2.1 Example

Assume we have the system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ from which we want to solve $\mathbf{x}$, with the following values:

$$\begin{pmatrix} 2 & -3 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -7 \\ 6 \end{pmatrix}$$

With a little effort we might find that $(1, 2, 3)^T$ is a solution, but so is $(-1, -1, 8)^T$. In fact there are infinitely many solutions. Among those solutions, there is one which has the smallest norm,

$$\hat{\mathbf{x}} = \boldsymbol{A}_R^+ \, \mathbf{b} = \begin{pmatrix} 2 & 1 \\ -3 & 1 \\ -1 & 1 \end{pmatrix} \left(\begin{pmatrix} 2 & -3 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ -3 & 1 \\ -1 & 1 \end{pmatrix} \right)^{-1} \begin{pmatrix} -7 \\ 6 \end{pmatrix} = \frac{1}{38} \begin{pmatrix} 52 \\ 97 \\ 79 \end{pmatrix}$$

All the other solutions are equal to this minimum norm solution plus some multiple of the vector

$$\mathbf{n} = \begin{pmatrix} -2 \\ -3 \\ 5 \end{pmatrix}$$

For example

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \hat{\mathbf{x}} + \frac{7}{38} \, \mathbf{n}$$

and

$$\begin{pmatrix} -1 \\ -1 \\ 8 \end{pmatrix} = \hat{\mathbf{x}} + \frac{45}{38} \, \mathbf{n}$$

Figure 1.7 shows how the subspace $R(\boldsymbol{A}^*)$ forms a plane in the 3 dimensional space V which is spanned by the two rows of $\boldsymbol{A}$. The vector $\mathbf{n}$ is the normal vector to this plane.

The vector $\hat{\mathbf{x}}$ lies in this plane. I fact,

$$\hat{\mathbf{x}} = \frac{-9}{38} \begin{pmatrix} 2 \\ -3 \\ -1 \end{pmatrix} + \frac{70}{38} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Note that the two coefficients in the above equation are the components of the vector $\mathbf{b}'$ from Figure 1.3.

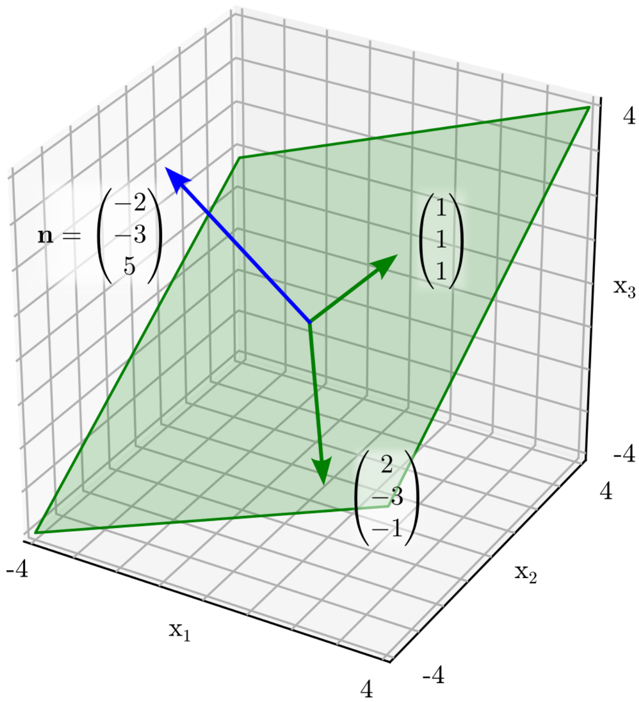

Figure 1.7: The two rows of $\boldsymbol{A}$ span the plane $R(\boldsymbol{A}^*)$ in the domain. All minimal norm solutions lie in this plane. the vector $\mathbf{n}$ is orthogonal to this plane and spans the sub space $N(\boldsymbol{A})$.

1.3 The over-determined system

In an over-determined system there there are more equations than there are unknowns. The matrix of such system has more rows than columns, i.e. its a skinny matrix. Also in this case we assume the system is still full rank, i.e. the columns are linearly independent. In some sense, the over-determined case is the opposite of the under-determined case we have seen in the previous section, and we can deal with it in a very similar way.

Since we have too many rows, it's impossible for them to all be linearly independent. And since each row defines an equation, we have in some sense too many equations. For the solution of the system, this leads to two possibilities. Either the set of equations is consistent, which leads to one unique exact solution, or the system is inconsistent, in which case there is no exact solution. In this case we have to settle for an approximate solution.

The subspace diagram in Figure 1.8 makes the situation clear; An arbitrary vector $\mathbf{b}$ can be considered to be the sum of two parts: A vector $\mathbf{b_p}$ which can be reached by $\boldsymbol{A}$ and a vector $\mathbf{b_0}$ which can not be reached by $\boldsymbol{A}$. So if we have the equation

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}$$

and we are asked to estimate $\mathbf{x}$ when $\boldsymbol{A}$ and $\mathbf{b}$ are given, we propose the following estimation:

$$\hat{\mathbf{x}} = \boldsymbol{A}_L^+ \,\mathbf{b}$$

in which $\boldsymbol{A}_L^+$ is the *left* pseudo inverse of $\boldsymbol{A}$

$$\boldsymbol{A}_L^+ = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*$$

(note that $\boldsymbol{A}_L^+\boldsymbol{A} = \boldsymbol{I}$). This estimation yield the part of $\mathbf{b}$ which is in the range of $\boldsymbol{A}$ and ignores the rest, i.e. the part of $\mathbf{b}$ which is in the null space of $\boldsymbol{A}^*$. So

$$\boldsymbol{A}\hat{\mathbf{x}} = \mathbf{b_p}$$

.

1.3.1 A note on interpretation

There might arise some confusion about how to interpret math expressions such as $\boldsymbol{A}\mathbf{x} = \mathbf{b}$. Such an expression does not definitely asserts that $\boldsymbol{A}\mathbf{x}$ is equal to $\mathbf{b}$. Instead, it must be read as a proposition, which either can be true of false. In case the equation has no solutions, the expression is false, and everything you derive from it remains false. For example, let's assume we fill in a value of $\mathbf{b}$ for which the equation does not have a solution. One might still multiply both sides with $\boldsymbol{A}_L^+$ to get

$$\boldsymbol{A}_L^+\boldsymbol{A}\ \ \mathbf{x} = \boldsymbol{A}_L^+ \,\mathbf{b} \implies \mathbf{x} = \boldsymbol{A}_L^+ \,\mathbf{b}$$

but this expression is still false; $\mathbf{x}$ is not equal to $\boldsymbol{A}_L^+ \,\mathbf{b}$, because we started out with an

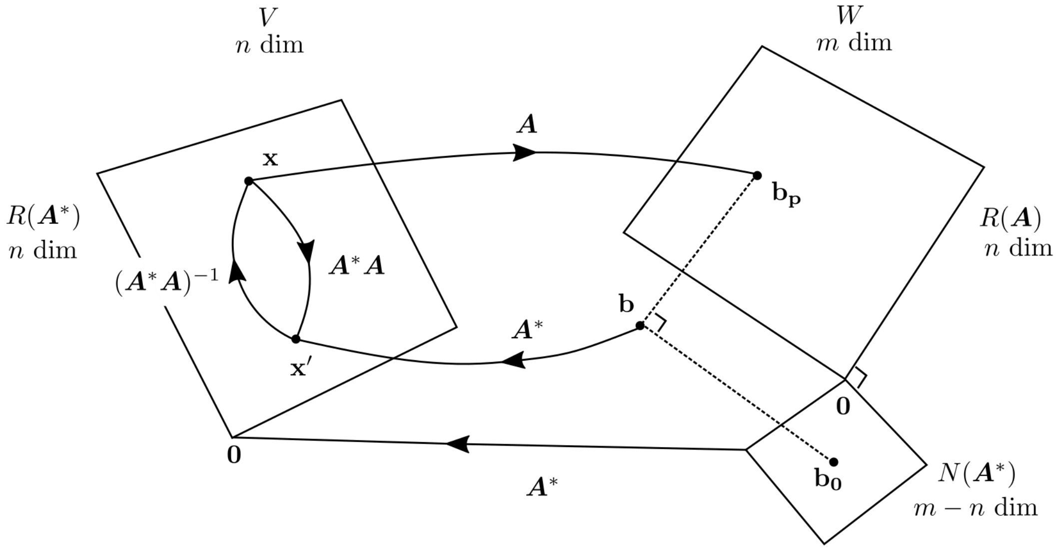

Figure 1.8: The subspaces of an over-determined system.

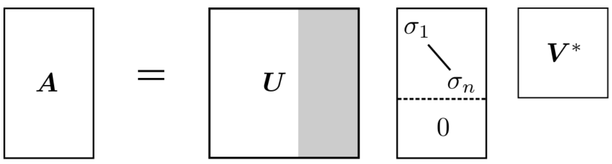

Figure 1.9: Singular value decomposition of an over-determined system.

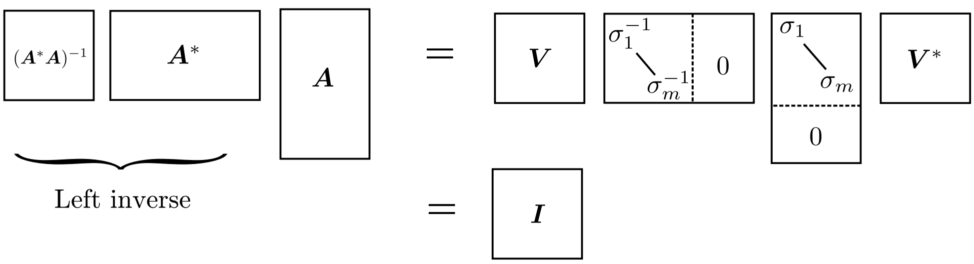

Figure 1.10: The left inverse applied from the left gives the identity.

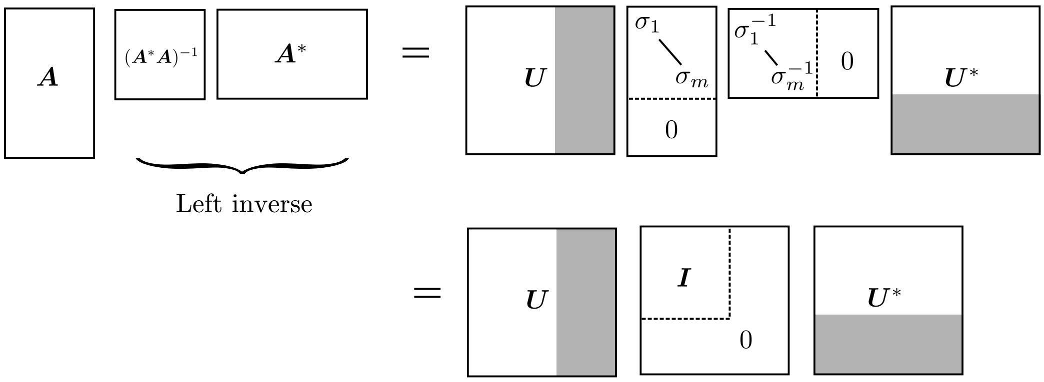

Figure 1.11: The left inverse applied from the right gives a projection onto $R(\boldsymbol{A})$.

expression which is false to begin with. This is why we introduced the new variable $\hat{\mathbf{x}}$. The hat symbol on top of the $\mathbf{x}$ means it's an estimation of $\mathbf{x}$, which is a different variable.

1.3.2 Example

Assume we have the system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ from which we want to solve $\mathbf{x}$, with the following values:

$$\begin{pmatrix} 2 & 1 \\ -3 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 10 \\ 8 \end{pmatrix}$$

In this case we are lucky because there is an exact solution:

$$\mathbf{x} = \begin{pmatrix} -1 \\ 7 \end{pmatrix}$$

which means that the the vector $\mathbf{b}$ lies completly in $R(\boldsymbol{A})$ and had no components in $N(\boldsymbol{A}^*)$.

Next, consider the equation:

$$\begin{pmatrix} 2 & 1 \\ -3 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

In this case $\mathbf{b}$ has component in both $R(\boldsymbol{A})$ and $N(\boldsymbol{A}^*)$ which means we can not find an $\mathbf{x}$ which exactly solves the equation. We can write $\mathbf{b}$ as follows:

$$\mathbf{b} = \mathbf{b_p} + \mathbf{b_0} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \frac{1}{38} \begin{pmatrix} 52 \\ 97 \\ 79 \end{pmatrix} + \frac{7}{38} \begin{pmatrix} -2 \\ -3 \\ 5 \end{pmatrix}$$

By using the left speudo inverse, we can find the vector $\hat{\mathbf{x}}$ which when transformed by $\boldsymbol{A}$ yields $\mathbf{b_p}$ and as such is the best possible solution of the equation, i.e the solution in the least squares sense:

$$\hat{\mathbf{x}} = \boldsymbol{A}_L^+ \mathbf{b} = (\boldsymbol{A}^* \boldsymbol{A})^{-1} \boldsymbol{A}^* \mathbf{b} = \left(\begin{pmatrix} 2 & -3 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ -3 & 1 \\ -1 & 1 \end{pmatrix} \right)^{-1} \begin{pmatrix} 2 & -3 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \frac{1}{38} \begin{pmatrix} -9 \\ 70 \end{pmatrix}$$

Chapter 2

Network analysis

2.1 Introduction

In this chapter we are going to look how to convert analog and digital circuits into matrix form. We do this for two reasons. First, circuit may look nice to humans, but a computer can not do much with them. But once converted to matrix form, it can very efficiently analyze the circuit. And secondly, by using matrices, we can use linear algebra to reason about the circuit on a higher abstraction level.

2.2 Time discrete signal flow graphs

Figure 2.1 shows a simple signal flow graph (SFG), in which we see the three basic components from which every SFG can be build: The gain element, the summation node, and the junction.

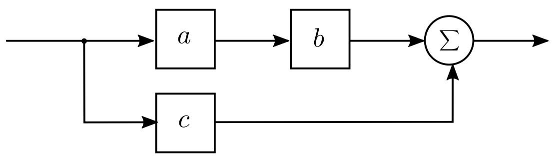

Figure 2.1: Simple signal flow graph

We will make two modification to this picture. First, we can obviously combine the gain elements a and b into a single gain block ab. And we need to define not three, but just two basic building blocks: The junction can be regarded as just a summation node with multiple outputs, each of which is equal to the sum of all inputs. If we do this, we then get 2.1.

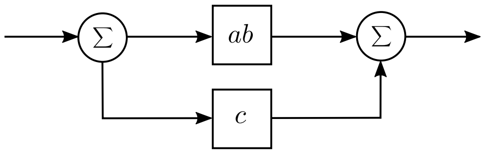

Figure 2.2: Simpler signal flow graph

Now we have defined a node as an implicit summation (in case it has multiple incoming arrows), and branches as implicit gain, we can draw the SFG in a more compact style, as shown in 2.3

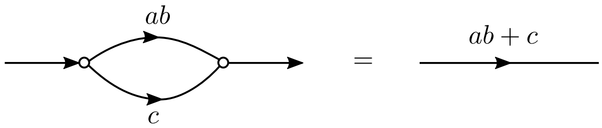

Figure 2.3: Signal flow graph in Roberts and Mullis style

Before we are going to look at a more complex flow graph, there is one simple flow graph which deserves special attention, and that is the feedback loop, as shown in 2.4.

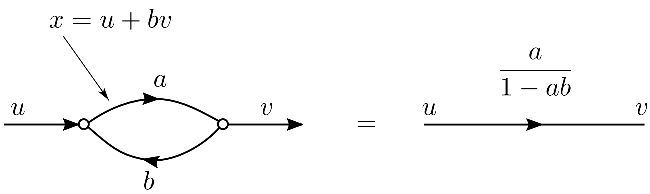

Figure 2.4: SFG of a feedback loop

To find the transfer function of such a loop, it is easiest to introduce an intermediate signal $x = u + bv$ just after the first node. Then we have $v = a(u + bv)$ from which we readily find the loop gain

$$\frac{v}{u} = \frac{a}{1 - ab}.$$

Figure 2.5 shows an example of a SFG which is a bit more complex. You can see that the flow graph gives a rough idea what is going on, but it's hard to tell exactly what the relation between input and output is. So, we will convert this SFG into matrix equations.

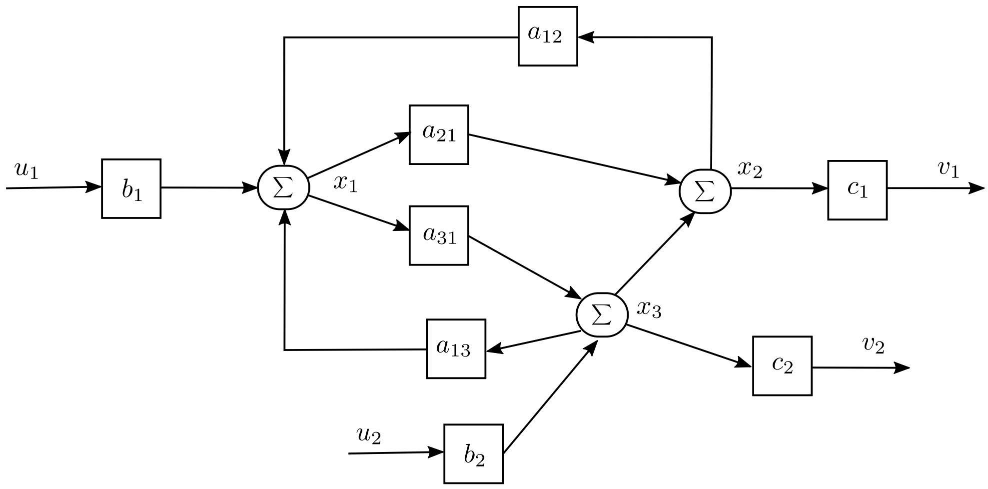

Figure 2.5: Other example of an SFG

The first step is that we write down the equations for all the output and intermediate signals:

$$
\begin{aligned}
x_1 &= a_{12}x_2 + a_{13}x_3 + b_1u_1 \\
x_2 &= a_{21}x_1 + x_3 \\
x_3 &= a_{31}x_1 + b_2u_2 \\
v_1 &= c_1x_2 \\
v_2 &= c_2x_3
\end{aligned}
$$

Next, we write this in matrix form

$$
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & 1 \\ a_{31} & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 & 0 \\ 0 & 0 \\ 0 & b_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}
$$

$$
\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 & c_1 & 0 \\ 0 & 0 & c_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}
$$

or symbolically:

$$
\begin{aligned}
\mathbf{x} &= \boldsymbol{A}\mathbf{x} + \boldsymbol{B}\mathbf{u} \\
\mathbf{v} &= \boldsymbol{C}\mathbf{x}
\end{aligned}
$$

From this form, its not hard to see that the relation between input and output is:

$$
\mathbf{v} = \boldsymbol{C}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}\mathbf{u}
$$

Next, we are going to discuss what happens if we reverse the flow graph, i.e. we go trough it in the opposite direction. In Figure. 2.5 the reverse of our previous example is shown. We have reversed the direction of all arrows, and reversed the role of input and output, i.e. we supply the input on $\mathbf{v}$ and measure the output on $\mathbf{u}$.

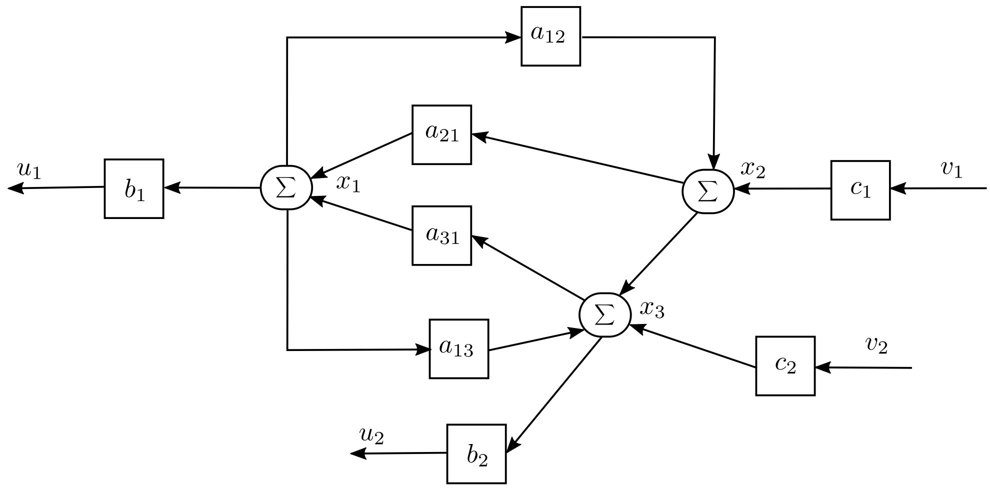

Figure 2.6: Revere of Figure 2.5

This is the list of equations:

$$\begin{aligned} x_1 &= a_{21}x_2 + a_{31}x_3 \\ x_2 &= a_{12}x_1 + c_1v_1 \\ x_3 &= a_{13}x_1 + x_2 + c_2v_2 \\ u_1 &= b_1x_1 \\ u_2 &= b_2x_3 \end{aligned}$$

which in matrix form look like this:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & a_{21} & a_{31} \\ a_{12} & 0 & 0 \\ a_{13} & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ c_1 & 0 \\ 0 & c_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} b_1 & 0 & 0 \\ 0 & 0 & b_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

If we compare this with the matrix equations of the original circuit, we see the resemblance: All matrices are transposed, and the position of the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices has switched. In symbolic form it looks like:

$$\begin{aligned} \mathbf{x} &= \boldsymbol{A}^*\mathbf{x} + \boldsymbol{C}^*\mathbf{v} \\ \mathbf{u} &= \boldsymbol{B}^*\mathbf{x} \end{aligned}$$

For the input-output relation we have:

$$\mathbf{u} = \boldsymbol{B}^*(\boldsymbol{I} - \boldsymbol{A}^*)^{-1}\boldsymbol{C}^*\mathbf{v}$$

The resemblance with the original input-output relation becomes even more apparent if we write it like this:

$$\mathbf{u} = \left[\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}\right]^* \mathbf{v}$$

Chapter 3

Time invariant linear systems

3.1 Introduction

When a linear system is time invariant, it can be completely characterized by it's impulse response.

3.2 Time discrete filters

In this section we will consider time discrete single-input, single-output systems with finite memory. Such a system is fully characterized by the following equation:

$$v[n] = \sum_{i=0}^{N} b[i]u[n-i] - \sum_{i=1}^{M} a[i]v[n-i]$$

Figure 3.1 shows the block diagram of such system.

If we take the Z-transform, we get:

$$U(z)\sum_{i=0}^{N} b[i]z^{-i} = V(z)\left[1 + \sum_{i=1}^{M} a[i]z^{-i}\right],$$

which we can rewrite in system function form:

$$H(z) = \frac{V(z)}{U(z)} = \frac{b[0] + b[1]z^{-1} + b[2]z^{-2} \cdots + b[N]z^{-N}}{1 + a[1]z^{-1} + a[2]z^{-2} \cdots + a[M]z^{-M}}$$

Next, we want to factor the polynomials in the numerator and denominator so that we get the pole-zero form of the system function. To make factorization possible, both polynomials should be written in such way that all terms have non-negative powers of z and the leading coefficient is 1.

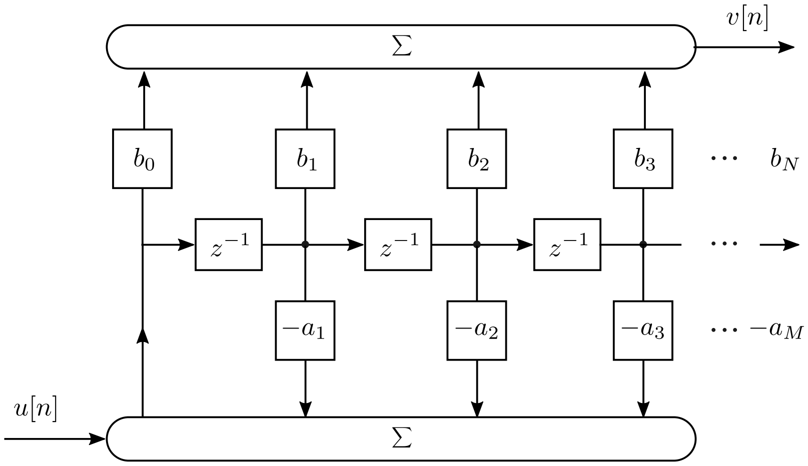

Figure 3.1: Time discrete filter

In the denominator, the leading coefficient is 1 by definition, but in the numerator, which represents the FIR part of the filter, the first n coefficient might be zero. So, we account for this by writing the system function as:

$$H(z) = \frac{b[n]z^{-n} + b[n+1]z^{-(n+1)} \cdots + b[N]z^{-N}}{1 + a[1]z^{-1} + a[2]z^{-2} \cdots + a[M]z^{-M}}$$

Next, we factor out $b[n]$ and z^{-n},

$$H(z) = b[n]\ z^{-n}\ \frac{1 + b[n+1]/b[n]z^{-1} \cdots + b[N]/b[n]z^{n-N}}{1 + a[1]z^{-1} + a[2]z^{-2} \cdots + a[M]z^{-M}}$$

and factor out the lowest power of z,

$$H(z) = b[n]\ z^{-n}\ \frac{z^{n-N}(z^{N-n} + b[n+1]/b[n]z^{N-n-1} \cdots + b[N]/b[n])}{z^{-M}(z^{M} + a[1]z^{M-1} + a[2]z^{-2} \cdots + a[M])}.$$

This form can be factored as:

$$H(z) = b[n]\ \frac{z^{M}(z - z_1)(z - z_2) \cdots (z - z_{N-n})}{z^{N}(z - p_1)(z - p_2) \cdots (z - p_M)}$$

The *order* of the system is equal to the number of delay elements, which equals $\max(N, M)$.

Note that if the FIR part has no leading 0 coefficients (i.e. $n = 0$), the number of poles equals the number of zeros. In such a system, the output reacts directly on the input. Although this is a causal system, sometimes a distinction is made by defining a *strictly causal* system as a system in which there is at least one delay between input and output. In a strictly causal system, the number of poles is always higher than the number of zeros.

In theory one could consider a system which has more zeros than poles, but such a system would be not causal (it produces output based on input samples which are not yet available), and thus not realizable in practice.