Properties of the Singular Value Decomposition

A good reference on numerical linear algebra is

G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1983.

Preliminary definitions:

Hermitian: Consider $x \in \mathbf{C}^n$. Then we define the vector "x Hermitian" by $x^H := \bar{x}^T$. That is, x^H is the complex conjugate transpose of x. Similarly, for a matrix $A \in \mathbf{C}^{m \times n}$, we define $A^H \in \mathbf{C}^{n \times m}$ by $\bar{A}^T$. We say that $A \in \mathbf{C}^{n \times n}$ is a *Hermitian matrix* if $A = A^H$.

Euclidean inner product: Given $x, y \in \mathbf{C}^n$. Let the elements[1] of x and y be denoted $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$.
Then the Euclidean inner product is defined as

$$\begin{aligned} \langle x, y \rangle &:= x^H y \\ &= \bar{x}_1 y_1 + \bar{x}_2 y_2 + \cdots + \bar{x}_n y_n \end{aligned}$$

Euclidean vector norm: Let "$\langle \bullet, \bullet \rangle$" denote the Euclidean inner product. Then the vector norm associated with this inner product is given by

$$\begin{aligned} \|x\|_2 &:= \sqrt{\langle x, x \rangle} \\ &= \sum_{i=1}^{n} |x_1|^2 \end{aligned}.$$

We often omit the "2" subscript when we are discussing the Euclidean norm (or "2-norm") exclusively.

---

[1] Sometimes we use subscripts to denote the elements of a vector, and sometimes to denote different members of a set of vectors. The meaning will be clear from context.

Euclidean matrix norm: Given $A \in \mathbf{C}^{m\times n}$. Then the *matrix norm induced by the Euclidean vector norm* is given by:

$$\begin{aligned}\|A\|_2 &:= \max_{v\neq 0} \frac{\|Av\|_2}{\|v\|_2} \\ &= \sqrt{\lambda_{\max}\left(A^H A\right)}\end{aligned}$$

where $\lambda_{\max}\left(A^H A\right)$ denotes the largest eigenvalue of the matrix $A^H A$. (It is a fact that all the eigenvalues of a matrix having the form $A^H A$ are all real and nonnegative.)

Orthogonality: Two vectors $x, y \in \mathbf{C}^n$ are *orthogonal* if $\langle x, y\rangle = 0$.

Orthonormal Set: A collection of vectors $\{x_1, x_2, \mathsf{K}, x_m\} \in \mathbf{C}^n$ is said to be an *orthonormal set* if

$$\langle x_i, x_j \rangle = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

(Hence $\|x_i\| = 1, \ \forall i$.)

Orthogonal Complement: Consider a subspace $\mathbf{X} \subseteq \mathbf{C}^n$. The *orthogonal complement* of $\mathbf{X}$, denoted $\mathbf{X}^\perp$, is defined as

$$\mathbf{X}^\perp := \left\{ x \in \mathbf{C}^n : \langle x, y\rangle = 0 \ \forall\, y \in \mathbf{X} \right\}.$$

That is, every vector in $\mathbf{X}^\perp$ is orthogonal to every vector in $\mathbf{X}$.

Unitary Matrix: A matrix $U \in \mathbf{C}^{n\times n}$ is *unitary* if $U^H U = U U^H = I_n$.

Fact: If U is a unitary matrix, then the columns of U form an orthonormal basis (ONB) for $\mathbf{C}^n$.

Proof of Fact: Denote the columns of U as $U = \begin{bmatrix} u_1 & u_2 & \mathsf{L} & u_n \end{bmatrix}$. Then

$$U^H U = \begin{bmatrix} u_1^H \\ u_2^H \\ \mathsf{M} \\ u_n^H \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \mathsf{L} & u_n \end{bmatrix}$$

$$= \begin{bmatrix} u_1^H u_1 & u_1^H u_2 & \mathsf{L} & u_1^H \\ u_2^H u_1 & u_2^H u_2 & \mathsf{L} & u_2^H \\ \mathsf{M} & \mathsf{M} & \mathsf{O} & \mathsf{M} \\ u_n^H u_1 & u_n^H u_2 & \mathsf{L} & u_n^H \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mathsf{L} & 0 \\ 0 & 1 & \mathsf{L} & 0 \\ \mathsf{M} & \mathsf{M} & \mathsf{O} & \mathsf{M} \\ 0 & 0 & \mathsf{L} & 1 \end{bmatrix}$$

Singular Value Decomposition:

Consider $M \in \mathbf{C}^{m \times n}$. Then there exist unitary matrices

$$U = \begin{bmatrix} u_1 & u_2 & \mathsf{K} & u_m \end{bmatrix}$$
$$V = \begin{bmatrix} v_1 & v_2 & \mathsf{K} & v_n \end{bmatrix}$$

such that

$$A = \begin{cases} U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^H, & m \geq n \\ \\ U \begin{bmatrix} \Sigma & 0 \end{bmatrix} V^H, & m \leq n \end{cases}$$

where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \mathsf{L} & 0 \\ 0 & \sigma_2 & \mathsf{L} & 0 \\ \mathsf{M} & \mathsf{M} & \mathsf{O} & \mathsf{M} \\ 0 & 0 & \mathsf{L} & \sigma_p \end{bmatrix}, \quad p = \min(m,n)$$

and

$$\sigma_1 \geq \sigma_2 \geq \mathsf{K} \geq \sigma_p \geq 0.$$

Terminology: We refer to σ_i as the i'th singular value, to u_i as the i'th left singular vector, and to v_i as the i'th right singular vector.

Properties of Singular Values and Vectors:

(1) Each singular value and associated singular vectors satisfy

$$Av_i = \sigma_i u_i, \quad i = 1,\mathsf{K}\ , p$$

(2) The largest singular value, $\sigma_{\max} := \sigma_1$, satisfies

$$\sigma_{\max} := \max_{v \neq 0} \frac{\|Av\|_2}{\|v\|_2}$$
$$= \|A\|_2$$

(3) The smallest singular value, $\sigma_{\min} := \sigma_p$, satisfies

$$\sigma_{\min} := \min_{v \neq 0} \frac{\|Av\|_2}{\|v\|_2}$$

If A is square and invertible, then $\|A^{-1}\|_2 = 1/\sigma_{\min}$.

(4) Suppose that $\sigma_1 \geq \sigma_2 \geq \mathsf{K} \geq \sigma_r > \sigma_{r+1} = \mathsf{K} = \sigma_p = 0$. Then

$$\text{rank}(A) = r$$

(5) Suppose that $\text{rank}(A) = r$. Then

$$\mathbf{R}(A) = \text{span}\{u_1, u_2, \mathsf{K}\ , u_r\},$$

where $\mathbf{R}(A)$ denotes the range, or columnspace, of A.

(6) Suppose that $\text{rank}(A) = r$. Then

$$\mathbf{N}(A) = \text{span}\{v_{r+1}, v_{r+2}, \mathsf{K}\ , v_n\},$$

where $\mathbf{N}(A)$ denotes the (right) nullspace of A.

(7) Suppose that $\text{rank}(A) = r$. Then

$$\mathbf{R}^{\perp}(A) = \text{span}\{u_{r+1}, u_{r+2}, \mathsf{K}\ , u_m\},$$

where $\mathbf{R}^{\perp}(A)$ denotes the orthogonal complement of $\mathbf{R}(A)$.

(8) Suppose that $\text{rank}(A) = r$. Then

$$\mathbf{N}^{\perp}(A) = \text{span}\{v_1, v_2, \mathsf{K}\ , v_r\},$$

where $\mathbf{N}^{\perp}(A)$ denotes the orthogonal complement of $\mathbf{N}(A)$.

(9) Suppose that $\text{rank}(A) = r$. Then

$$\begin{aligned} A &= \sum_{i=1}^{r} \sigma_i u_i v_i^H \\ &= U_r \Sigma_r V_r^H \end{aligned}$$

where

$$U_r := \begin{bmatrix} u_1 & u_2 & \mathsf{K} & u_r \end{bmatrix},\ V_r := \begin{bmatrix} v_1 & v_2 & \mathsf{K} & v_r \end{bmatrix},\ \Sigma_r := \text{diag}\begin{pmatrix} \sigma_1 & \sigma_2 & \mathsf{K} & \sigma_r \end{pmatrix}.$$

(10) If $A \in \mathbf{C}^{n \times n}$ is invertible, then

$$A^{-1} = \sum_{i=1}^{n} \tfrac{1}{\sigma_i} v_i u_i^H$$

Definition: The *pseudoinverse* of A, denoted $A^{\#}$, is given by

$$A^{\#} = V_r \Sigma_r^{-1} U_r\ .$$

The pseudoinverse has the following properties:

- If $m = n$ and $\text{rank}(A) = n$, then $A^{\#} = A^{-1}$.
- If $m > n$, and $\text{rank}(A) = n$, then A is left invertible, and $A^{\#} = A^{-\text{L}}$.
- If $m < n$, and $\text{rank}(A) = m$, then A is right invertible, and $A^{\#} = A^{-R}$.
- Note that $A^{\#}$ is well defined even if $\text{rank}(A) < \min(m,n)$.

Many results for linear systems have the form "if such and such a matrix has full rank, then such and such a property holds" (paraphrased from Golub and Van Loan). Such results are *naive*, in that they neglect the the fact that a matrix generated from physical data is almost always full rank. The more important question is *not* "does the matrix have full rank?", but rather "how close is the matrix

to one which does not have full rank?". The SVD is a very useful tool in making this concept precise.

Consider $A \in \mathbf{C}^{m \times n}$, and that A has *full rank*:

$$\text{rank}(A) = p := \min(m,n).$$

Suppose that we perturb the elements of A, yielding $\hat{A} = A + \Delta A$. We wish to know how large the error ΔA can become before $\hat{A}$ loses rank.

Proposition:

(a) Suppose that $\|\Delta A\|_2 < \sigma_{\min}(A)$. Then

$$\text{rank}(A + \Delta A) = p.$$

(b) There exists a matrix ΔA, with $\|\Delta A\|_2 = \sigma_{\min}(A)$ such that

$$\text{rank}(A + \Delta A) < p.$$

Proof:

(a) Using the triangle inequality (and dropping the subscript):

$$\begin{aligned}
\sigma_{\min}(A + \Delta A) &:= \min_{\|v\|=1} \|(A + \Delta A)v\| \\
&\geq \min_{\|v\|=1} \{\|Av\| - \|\Delta A v\|\} \\
&\geq \min_{\|v\|=1} \|Av\| - \max_{\|v\|=1} \|\Delta A v\| \\
&\geq \sigma_{\min}(A) - \sigma_{\max}(\Delta A) \\
&> 0 \\
&\Rightarrow \text{rank}(A + \Delta A) = p
\end{aligned}$$

(b) Let $A = \sum_{i=1}^{p} \sigma_i u_i v_i^H$, and consider $\Delta A = -\sigma_p u_p v_p^H$ (where $\sigma_p = \sigma_{\min}$). It is easy to see that $A + \Delta A = \sum_{i=1}^{p-1} \sigma_i u_i v_i^H$, and thus that $\text{rank}(A + \Delta A) = p - 1$.

Suppose that $\text{rank}(A) = p$, but A has very small singular values. Then A is "close" to a singular matrix in the sense that there exists a small perturbation ΔA to the elements of A that causes $\hat{A}$ to lose rank. Indeed, such a matrix should possibly be treated in applications as though it were singular.

In practice, people do not look always look for small singular values to determine distance to singularity. It is often more useful to compute the *ratio* between the maximum and minimum singular values.

Definition (Condition Number): Consider $A \in \mathbf{C}^{m \times n}$ and suppose that $\text{rank}(A) = p = \min(m,n)$. Then the (Euclidean) condition number of A is defined as

$$\kappa(A) := \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} .$$

###

Suppose $\text{rank}(A) = p$, but that $\kappa(A) >> 1$. It follows that A is "almost rank deficient". In fact, the process of calculating an inverse[2] for A may not be numerically robust, and calculations involving A^{-1} may be prone to error. Let's explore this idea.

Consider the linear system of equations

$$Ax = b, \quad A \in \mathbf{C}^{m \times n}, \quad b \in \mathbf{C}^n, \quad m \geq n, \quad \text{rank}(A) = n.$$

Suppose that we are given the data for A and b and need to solve for x. Let $\mathbf{R}(A)$ denote the range of A. If $b \in \mathbf{R}(A)$, then we may find x from

$$x = A^{\#} b.$$

In reality, the elements of A and b will be corrupted by errors. These may arise due to uncertainty in the methods used to generate the data. Errors also arise due to numerical roundoff in the computer representation of the data. We would like to know how these errors affect the accuracy of the solution vector x.

---

[2] If A is not square, then we calculate $A^{\#}$ which will be equal to the left or right inverse, whichever is appropriate.

Let

$$\hat{A} := A + \Delta A$$
$$\hat{b} := b + \Delta b$$
$$\hat{x} := x + \Delta x$$

so that $\hat{A}\hat{x} = \hat{b}$. Our next result relates the errors ΔA and Δb to errors in the computed value of x.

Proposition: Consider $A \in \mathbf{C}^{m\times n}$ has rank n, and that

$$\frac{\|\Delta A\|}{\sigma_{\min}(A)} \le \alpha < 1.$$

Suppose that ΔA and Δb satisfy the bounds $\frac{\|\Delta A\|}{\|A\|} \le \delta$ and $\frac{\|\Delta b\|}{\|b\|} \le \delta$, where δ is a constant. Then

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{2\delta}{1-\alpha}\kappa(A).$$

###

The proof is obtained as a sequence of homework exercises in Golub and van Loan.

Note that we can, in principle, calculate A^{-1} by successively solving the above linear system for $b = \begin{bmatrix}1\\0\\M\\0\end{bmatrix}, \begin{bmatrix}0\\1\\M\\0\end{bmatrix}, \mathrm{K}, \begin{bmatrix}0\\0\\M\\1\end{bmatrix}$.

It follows from this result that if $\kappa(A) >> 1$, then small relative errors in A and b *may* result in large relative errors in the computed value of x. One should be cautioned, however, that this error estimate is only an *upper bound*. Hence, the computed answer is not *guaranteed* to be incorrect. However, there are nonpathological examples for which the upper bound on the error is achieved! Hence, if no additional information is available, *the answer to any calculation involving the inverse of a matrix that is ill-conditioned (i.e., $\kappa(A) >> 1$) should be viewed dubiously.*

Example: Consider $A=\begin{bmatrix}1 & 100\\ 0 & 1\end{bmatrix}$. MATLAB says that $\text{rank}(A)=2$:

```
»A=[1 100;0 1];
»rank(A)
ans =
     2
»cond(A)
ans =
    1.0002e+04
```

The condition number looks pretty large; however, for the purposes of solving linear systems of two equations in two unknowns, this matrix is not particularly ill-conditioned *with respect to numerical roundoff error in the computer computations*. Suppose, on the other hand, that A and b are constructed from physical data. How reliable are calculations based upon A^{-1}?

For example, are we sure that the zero in the (2,1) element of A is *really* zero?

Suppose that the (2,1) element of A is perturbed:

$$\hat{A}=\begin{bmatrix}1 & 100\\ 0.009 & 1\end{bmatrix},$$

and consider the product $A^{-1}\hat{A}$; if $\hat{A}=A$, this product will equal the identity matrix. Let us see what the error is for our example:

```
»inv(A)*Ahat
ans =
   0.1          0.0
   0.009        1.0
```

We see that a small error in one of the elements of A results in a large error in the product $A^{-1}\hat{A}$!

On the other hand, consider an error of the same magnitude in the (1,1) element:

$$\hat{A}=\begin{bmatrix}1.009 & 100\\ 0 & 1\end{bmatrix}$$

```
»inv(A)*Ahat
ans =
      1.009          0
          0        1.0
```

In this case, the error in the product $A^{-1}\hat{A}$ is also small.

What to do with an "almost rank deficient" matrix:

Suppose we conclude that $\kappa(A)$ is so large that A should be viewed as rank deficient. How do we find an approximation to A? An obvious approach is to attempt to approximate A by a matrix of lower rank.

Approach 1: Motivated by the discussion above, let's use the information contained in the SVD and try to find a matrix of *lower rank* that is "close" to A.

We shall do this for the case $m \geq n$, so that

$$A = U\begin{bmatrix}\Sigma\\ 0\end{bmatrix}V^H,$$

where

$U = [u_1 \quad \mathsf{K} \quad u_m] \in \mathbf{C}^{m\times m}$, $V = [v_1 \quad \mathsf{K} \quad v_n] \in \mathbf{C}^{n\times n}$, and $\Sigma = \mathrm{diag}(\sigma_1 \quad \sigma_2 \quad \mathsf{K} \quad \sigma_n)$.

Suppose that

$$\sigma_1 \geq \mathsf{K} \geq \sigma_r > \varepsilon > \sigma_{r+1} \geq \mathsf{K} \ \sigma_n \geq 0,$$

where ε is so small (or so much smaller than σ_r) that the last (n-r+1) singular values are all effectively zero. Then we say that A has *effective rank* equal to r.

Furthermore, let $\mathbf{R}_{eff}(A)$ and $\mathbf{N}_{eff}(A)$ denote the *effective range* and *effective nullspace* of A, respectively. Then we can calculate bases for these subspaces by choosing appropriate singular vectors:

$$\mathbf{R}_{eff}(A) := \mathrm{span}\{u_1, \mathsf{K}, u_r\} \text{ and } \mathbf{N}_{eff}(A) := \mathrm{span}\{v_{r+1}, \mathsf{K}, v_n\}.$$

Similarly,

$$\mathbf{R}_{eff}^{\perp}(A) := \text{span}\{u_{r+1}, \text{K}, u_m\} \text{ and } \mathbf{N}_{eff}^{\perp}(A) := \text{span}\{v_1, \text{K}, v_r\}.$$

That is, the effective range of A is spanned by the left singular vectors corresponding to the nonnegligible singular values of A, the effective nullspace is spanned by the right singular vectors corresponding to the negligible left singular values, and so forth.

Define

$$\begin{aligned} A_{eff} &= \sum_{i=1}^{r} \sigma_i u_i v_i^H \\ &= U_r \Sigma_r V_r^H \end{aligned}$$

where

$$U_r := \begin{bmatrix} u_1 & u_2 & \text{K} & u_r \end{bmatrix}, \; V_r := \begin{bmatrix} v_1 & v_2 & \text{K} & v_r \end{bmatrix}, \; \Sigma_r := \text{diag}\begin{pmatrix} \sigma_1 & \sigma_2 & \text{K} & \sigma_r \end{pmatrix}.$$

If have chosen wisely, then

$$\kappa\left(A_{eff}\right) = \frac{\sigma_1}{\sigma_r}$$

will not be "too large". Note that

$$b \in \mathbf{R}\left(A_{eff}\right) \Leftrightarrow b \in \mathbf{R}_{eff}(A).$$

Hence, if $b \in \mathbf{R}\left(A_{eff}\right)$, then the equation $Ax = b$ has a solution that is *robust* against small errors in A. This solution is given by $x = A_{eff}^{\#} b$.

Example(continued):

Once again, consider $A = \begin{bmatrix} 1 & 100 \\ 0 & 1 \end{bmatrix}$. Let's look at the SVD of A, and use the information from the singular values and vectors to construct a rank 1 matrix, A_{eff}, that is "close" to A.

»[U,S,V] = svd(A)

U =

```
    1.0      -0.01
    0.01     1.0

S =
  100.01          0
       0          0.01

V =
   0.01       -1.0
   1.0         0.01
```

Let's calculate basis vectors for $\mathbf{R}_{eff}$ and $\mathbf{N}_{eff}$:

```
»Reff = U(:,1)
Reff =
     1.0000
     0.0100
```

The value of R_{eff} makes sense, because both columns of A have much larger entries in the first row than in the second.

```
»Neff = V(:,2)
Neff =
    -1.0000
     0.0100
```

Check to see that N_{eff} "almost" gets multiplied by zero, and thus is in the "effective" nullspace of A:

```
»A*Neff
ans  =
    -0.0001
     0.01
```

Now, let's construct a rank 1 matrix that approximates A by deleting the small singular value and associated singular vectors:

$$A_{eff} = \sigma_1 u_1 v_1^H$$

Note that $\mathbf{R}\left(A_{eff}\right) = \mathbf{R}_{eff}(A)$ and $\mathbf{N}\left(A_{eff}\right) = \mathbf{N}_{eff}(A)$.

```
»Aeff = Reff*S(1,1)*V(:,1)'
Aeff =
    0.9999   100.0000
    0.0100     0.9999

»rank(Aeff)
ans =
    1
```

We see that, as expected, $\text{rank}\left(A_{eff}\right)=1$.

Approach 2: Suppose we try to apply this idea to a matrix obtained from physical data. Unfortunately, it is sometimes difficult to relate the singular values and vectors to physical quantities. An alternate approach which may avoid this difficulty is to merely delete columns of the matrix. For example, suppose that we merely zeroed the first column of A:

$$A_1 = \begin{bmatrix} 0 & 100 \\ 0 & 1 \end{bmatrix}$$

It is easy to see that $\mathbf{R}(A_1) \cong \mathbf{R}_{eff}(A)$ and $\mathbf{N}(A_1) \cong \mathbf{N}_{eff}(A)$.

Shortly, we shall see how to apply these concepts to evaluate the rank of the DC gain matrix of a plant we wish to control. Before doing this, we need to review one more topic from numerical linear algebra. This topic is one which can cause a great deal of headaches in control applications.

Scaling and Choice of Units

There is a difficulty associated with using $\sigma_{\min}(A)$ and/or $\kappa(A)$ as measures of distance to singularity. Namely, the size of these quantities varies with *scaling*. For example, consider the equation

$$y = Au,\ y \in \mathbf{C}^q,\ u \in \mathbf{C}^p,$$

where the elements of the vectors represent physical quantities.

Suppose that we change the *units* used to measure the elements of u and y:

$$y_{new} := D_1^{-1} y, \quad u_{new} := D_2^{-1} u$$

where D_1 and D_2 are diagonal matrices whose diagonal elements are all real and positive. (The use of inverses is to conform with the notation in Golub and Van Loan; it also suggests the common practice of scaling signals by their nominal values.)

It is easy to show that y_{new} and u_{new} satisfy

$$y_{new} = A_{new} u_{new} \ , \quad \text{where } A_{new} := D_1^{-1} A D_2 .$$

Hence changing units for the physical quantities related by A corrsponds to *scaling* A by diagonal matrices. It is easy to show that the singular values and condition number of A_{new} may be very different than those of the original matrix. Indeed, let us consider our example

$$A = \begin{bmatrix} 1 & 100 \\ 0 & 1 \end{bmatrix},$$

for which $\sigma_{\min}(A) = 0.01$ and $\kappa(A) = 10{,}000$. If we choose $D_1 = \begin{bmatrix} d_1 & 0 \\ 0 & 1 \end{bmatrix}$ and $D_2 = \begin{bmatrix} 1 & 0 \\ 0 & d_2 \end{bmatrix}$, then

$$D_1^{-1} A D_2 = \begin{bmatrix} 1 & 100 \frac{d_2}{d_1} \\ 0 & 1 \end{bmatrix}.$$

It follows that, by changing units, we can make the condition number and singular values of A_{new} arbitrarily close to one. This means that the matrix is no longer "almost singular"? Does it also mean that the physical property that depended upon A being full rank is now present *robustly*? Maybe and maybe not. It all depends if the units of A_{new} are physically reasonable. The latter condition is a qualitative one that depends upon good engineering sense. For example, if one were studying deposition rates of a metal oxide film on a silicon substrate, measuring speed in terms of microns/minute might be very appropriate. If, in order to obtain a nice "A" matrix, we had to change these units to furlongs/fortnight, then it would be very difficult to interpret our answer.

Unfortunately, finding a physically reaonable set of units with which to work is sometimes problematic. In a MIMO feedback system, we may be comparing very different types of signals, for example, voltages, pressures, speeds, temperatures. Our choice of units will dictate the conditioning of various matrices, such as the DC gain matrix. To draw conclusions from the condition number, we must nevertheless have units that make comparisons between different physical quantities meaningful.

Systems Interpretations

Consider the system

$$\dot{x} = Ax + Bu, \qquad x \in \mathbf{R}^n,\ u \in \mathbf{R}^p$$
$$y = Cx, \qquad y \in \mathbf{R}^q$$
$$P(s) = C(sI - A)^{-1}B$$

We will assume the system is controllable and observable and (for simplicity) that $p = q$, and that A has stable eigenvalues. Consider the problem of achieving zero steady state error to a step command. We have seen that this problem is solvable precisely when the DC gain matrix satisfies $\text{rank}(P(0)) = q$. What happens when the DC gain matrix is *almost* rank deficient? Intuitively, this means that the command tracking problem is "almost" unsolvable.

We shall suppose that in *physically relevant units* the DC gain matrix satisfies:

(i) $\sigma_{\max}(P(0)) > 1$

and

(i) $\sigma_{\min}(P(0)) << 1.$

Together, (i) and (ii) imply

(iii) $\kappa(P(0)) >> 1.$

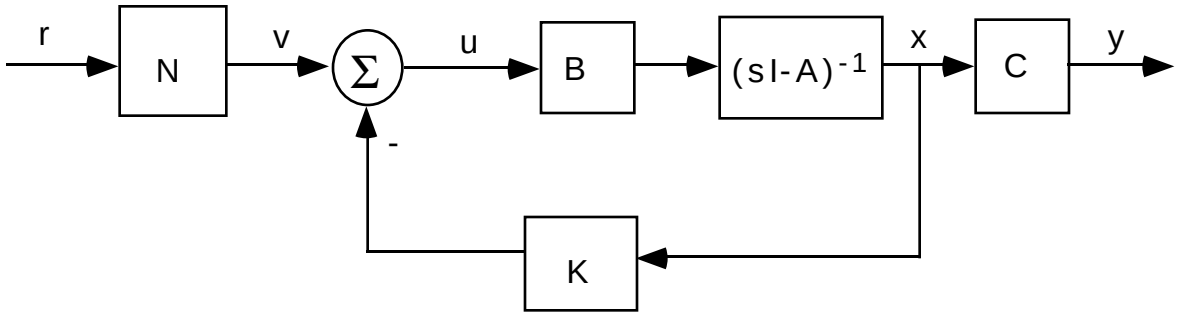
(For example, consider the matrix $A = \begin{bmatrix} 1 & 10 \\ 0 & 1 \end{bmatrix}$, which has $\sigma_{\max} = 100.01$, $\sigma_{\min} = 0.01$, and $\kappa \cong 10000.$)

It follows from (i) that there does exist some control authority at DC (i.e., the DC gain matrix is nonzero). However, (ii) implies that some control inputs are relatively ineffective. Ineffective control inputs can arise in two ways.

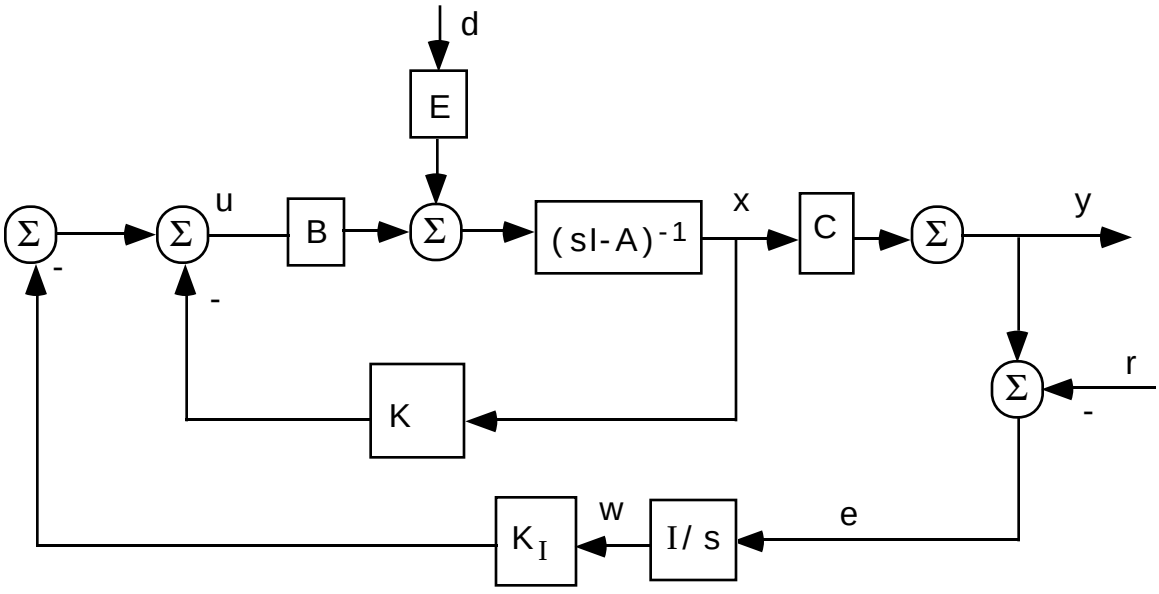
- If a column of $P(0)$ has small magnitude, then the particular input associated with that column has relatively little DC authority.
- If two or more columns of $P(0)$ are almost linearly dependent, then the control inputs associated with those columns are almost redundant.

Finally, it follows from (iii) that any calculations involving $P(0)^{-1}$ are sensitive to errors in the data for $P(0)$.

Let's now consider the implications of (i)-(iii) for the constant command tracking problem. Recall that this control problem is solvable only if $P(0)$ is nonsingular. We developed two solutions to this problem. First, we used state feedback and a constant gain precompensator, $N = \left[C(-A+BK)^{-1}B\right]^{-1}$:



Second, we augmented the system with integrators, and fed back both the plant and integrator states:



In each case, closed loop stability implies that the steady state response of the system output to a step command $r(t) = r_0 \mathbf{1}(t)$ satisfies $y_{ss} = r_0$. Note also that, in each case, $y_{ss} = P(0)u_{ss}$. It follows that the steady state control signal must satisfy:

$$u_{ss} = P(0)^{-1} r_0$$

Since $\sigma_{\max}\left(P(0)^{-1}\right) = 1 \big/ \sigma_{\min}\left(P(0)\right)$, it follows from (ii) that relatively large control signals will be required to track certain step commands. Large control signals are undesirable because they may saturate control actuators or cause other undesirable nonlinear behavior.

Furthermore, (iii) tells us that small changes in the data of $P(0)$ may cause large changes in $P(0)^{-1}$. It follows that *small errors in* $P(0)$ *may cause the size of the control signal generated to force command tracking to be much larger than that indicated from the nominal value of* $P(0)$. Hence, even if the control signals obtained by examining the nominal model of $P(0)$ are reasonable, the control signals obtained from the true plant may not be.

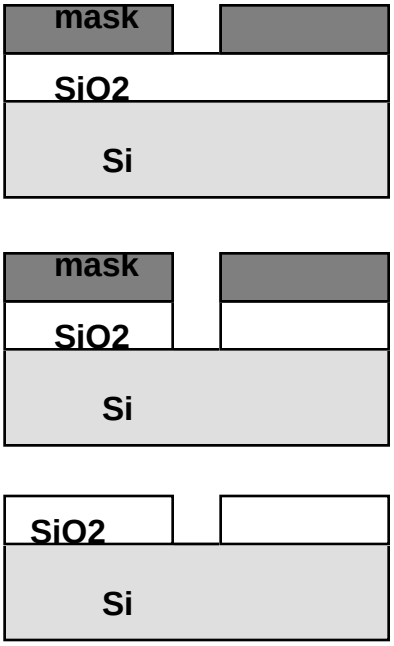
It follows from our discussion of effective range and nullspace that if $r_0 \in \mathbf{R}_{eff}(A)$, then the command $r(t) = r_0 \mathbf{1}(t)$ can be tracked without generating excessively large control signals.

Let's see an application of these ideas to data from a real system. First, we shall describe the system, and then analyze its DC gain matrix.

Example: Reactive Ion Etching of Silicon Wafers

Current practice in the semiconductor and flat panel display manufacturing industries uses little real-time feedback control. We are engaged in a project that will incorporate real-time feedback control into the process of Reactive Ion Etching (RIE).

The ultimate design goal is to reliably and reproducibly etch a pattern on a wafer; this is one of many steps needed to produce a semiconductor chip...



There are a number of characteristics of the etch we would like to control. Among these are:

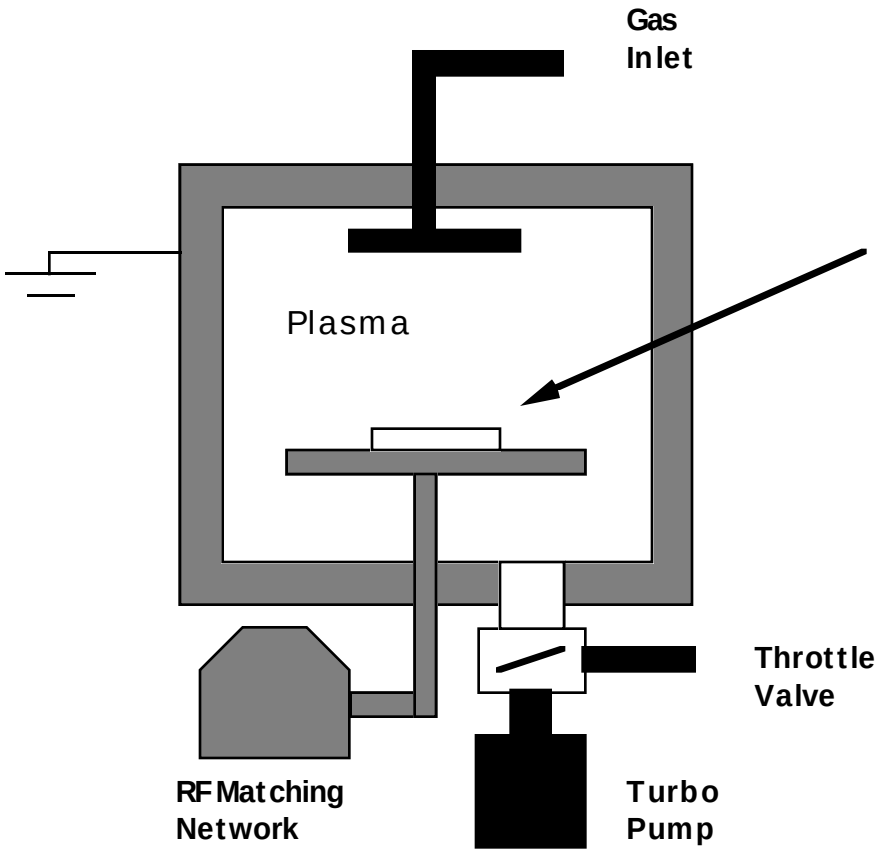
1. Sidewall angle: Are the sides of the etched feature vertical or sloped?

2. Selectivity: What are the relative etch rates of SiO_2, Si, and the photoresist mask?

3. Etch rate: Is this constant over the course of a single etch, and from etch to etch for many etches?

4. Uniformity: Are etch properties consistent across the wafer?

Wafer etching takes place inside a low pressure plasma chamber; a rough diagram is shown below:



We have the following three actuators that we can use for control:

RF Power, Throttle angle, and CF_4 Gas Flow Rate.

We also have sensors to measure three signals that we can feed back and attempt to regulate:

V_{bias}, $[F]$, Pressure.

These actuators and sensors are related to the plasma, not the wafer. Currently, we cannot sense features on the wafer surface in real time. Hence, our control strategy is to use the control inputs to regulate the plasma properties, which we *can* measure. The plasma properties are only indirectly related to the wafer etch; however, they do determine the environment in which the etch takes place. *It*

is our hypothesis that by better regulating the plasma environment, we will achieve a higher yield of usable etched wafers.

We now describe the control inputs and sensed outputs in more detail.

Inputs:

The flow input determines the rate at which CF_4 gas enters the etch chamber.

The RF power input has two effects: (i) it disassociates $CF_4 \rightarrow CF_3^+ + F$ (a charged ion and reactive fluorine radical whose density we denote by $[F]$), and (ii) it sets up a bias voltage, V_{bias}, across the plasma. This bias voltage accelerates the ions so that they bombard the wafer surface. The physical energy thus imparted to the surface, combined with the chemical reactions between $[F]$ and Si are responsible for etching the exposed portion of the wafer surface.

The throttle input determines the rate at which gases are exhausted from the chamber.

Outputs:

The V_{bias} output is responsible for the *physical* component of etching.

The $[F]$ output is responsible for the *chemical* component of etching.

The Pressure output determines (among other things) the mean free path between collisions in the chamber. The longer the mean free path, the more energy the ions have when they impact the wafer surface, and the greater the physical component of the etching process.

Modelling:

Using small signal step response data together with black box system identification techniques, we obtained a linear model of the system at the nominal operating condition:

Throttle: 10% open
CF_4 Flow: 15 sccm

RF power: 800 watts

V_{bias}: 400 volts
Pressure: 10 millitorr
$[F]$: 5

The DC gain matrix of the resulting transfer function model is:

$$P(0) = \begin{bmatrix} 13.2 & -6.77 & 0.37 \\ -1.0 & 0.973 & 0 \\ 0.123 & 0.0682 & 0.011 \end{bmatrix}$$

To aid in analyzing this matrix, we normalize each input and output by its nominal value. This yields a normalized DC gain matrix:

$$P_N(0) = \begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.02 \end{bmatrix} \begin{bmatrix} 13.2 & -6.77 & 0.37 \\ -1.0 & 0.973 & 0 \\ 0.123 & 0.0682 & 0.011 \end{bmatrix} \begin{bmatrix} \boldsymbol{10.0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{15.0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{800} \end{bmatrix}$$

$$= \begin{bmatrix} 0.3300 & -0.2539 & 0.7400 \\ -1.0000 & 1.4595 & 0.0 \\ 0.2460 & 0.2046 & 1.7600 \end{bmatrix}$$

$$\cong \begin{bmatrix} 0.3 & -0.3 & 0.7 \\ -1.0 & 1.5 & 0.0 \\ 0.2 & 0.2 & 1.8 \end{bmatrix}$$

Note that both CF_4 flow and Throttle angle each primarily affect pressure. Specifically, if we open the exhaust throttle, then the steady state pressure will *decrease*. On the other hand, if we increase the rate of CF_4 flow, then the steady state pressure will *increase*. These effects are plausible, because flow affects the rate at which gases enter the chamber, and throttle affects the rate at which gases leave the chamber. Power has no steady state effect on pressure, but it does affect both V_{bias} and $[F]$ relatively more strongly than do flow and throttle.

»[U,S,V] = svd(P_N)

```
U =                                   % left singular vectors
    0.4302   -0.0407   -0.9018
   -0.3499    0.9134   -0.2082
    0.8321    0.4051    0.3787


S =                                   % singular values
   1.9663         0         0
        0    1.7846         0
        0         0    0.0046


V =                                   % right singular vectors
    0.3543   -0.4635    0.8122
   -0.2287    0.7992    0.5558
    0.9067    0.3827   -0.1771

»kappa = cond(P_N)

kappa =
          425.0254
```

The large condition number is consistent with the fact that the throttle and flow inputs are almost redundant, and thus do not yield two independent degrees of control authority. As we have noted, both these inputs primarily affect pressure.

As a consequence of the poorly conditioned DC gain matrix, we cannot use our three actuators to independently control all three outputs unless we can tolerate very large control signals without saturating the actuators. As it happens, both the throttle and the mass flowmeter regulating CF_4 flow into the chamber are quite nonlinear, and operate fairly close to saturation. These actuators certainly cannot tolerate large control signals.

An approach to the problem of two redundant inputs is to keep one of them fixed at its nominal value, and use only the other for the purpose of feedback control. Engineering insight must be used to decide which input to keep, and which to ignore. In this case, we keep flow fixed because the mass flowmeter saturates easily. (In particular, the flow variable cannot be negative -- CF_4 cannot be sucked out of the chamber using the flowmeter!)

With flow removed as an actuator, the DC gain matrix becomes:

```
»DCnew1 = DCnew(:,[1 3])
DCnew1 =
    0.3300    0.7400
   -1.0000         0
    0.2460    1.7600
```

We now have a control problem with two inputs and three outputs. Becasue we cannot independently control all three outputs, we must choose two of them. (More generally, we can choose any two independent linear combinations of the three outputs.) The physics of the etch process suggests that V_{bias} and $[F]$ are relatively more important than Pressure (although this is a debatable point). Deleting Pressure from the DC gain matrix yields

```
»DCnew2 = DCnew1([1 3],:)
DCnew2 =
    0.3300    0.7400
    0.2460    1.7600

»svd(DCnew2)
ans =
    1.9423
    0.2053
```

The condition number is now $\cong 10$, and the control problem is reasonably well-conditioned.