ADVANCED COMPUTATIONAL TECHNIQUES IN ENGINEERING

Assignment 2

Luigi Vanacore 48543518

July 23, 2024

1 Part A

A signal, denoted as **b**, is collected at **M** sampling points, represented as **b(1), b(2), ...b(m), ...,b(M)**. The task is to strategically insert **N** components, denoted as **x (x(1), x(2), ...,x(n),...x(N))**, into the system **A** to alter the signal profile. The altered signal profile is represented as **B=Ax+b**

The signal B will satisfy the following condition:

$$\left| \frac{\mathbf{B}}{B_0} - \mathbf{1} \right| \leq \epsilon,$$

where B_0 is the mean value of B and $\mathbf{1}$ is a vector of ones of M components. The tolerance ϵ is a vector with the following components:

$$\epsilon(m) = 0.0001, \quad m = 1, 2, \ldots, M.$$

Additionally, both b and B are positive vectors. The range of components of x is:

$$|x(i)| \leq 0.01, \quad i = 1, 2, \ldots, N.$$

You are asked to minimise the 0-norm, 1-norm using linprog() and 2-norm of vector $\mathbf{x}$

1.1 L1 minimisation

The following optimisation program represents the minimisation of the L1 norm of the vector x enforcing the requested constraints:

$$\begin{aligned} &\min \|x\|_1 \\ &\text{s.t.} \\ &\left| \frac{Ax + b}{\text{mean}(Ax + b)} - \mathbf{1} \right| \leq \epsilon \\ &Ax + b \geq 0 \\ &|x(i)| \leq 0.01 \quad i = 1, 2, \ldots, N \end{aligned} \tag{1}$$

However, this program is not linear in this formulation. We need to reformulate it in order to solve it with linprog().

First, let's recall the definition of L1 norm of x:

$$\|x\|_1 = |x(1)| + |x(2)| + \ldots + |x(N)|$$

We can transform the objective function in a linear one by using the following trick:

$$\begin{aligned} x(i) &= x^+(i) - x^-(i) \\ |x(i)| &= x^+(i) + x^-(i) \\ x^+(i) &\geq 0 \\ x^-(i) &\geq 0 \end{aligned} \tag{2}$$

So now the objective function is:

$$\min \sum_{i=1}^{N} \left(x^+(i) + x^-(i)\right)$$

Furthermore, it is needed to add an optimisation variable B_0 which is equal to the mean of (Ax+b). To do so, it is necessary to add the following constraint:

$$B_0 = \frac{(\mathbf{1})' * (Ax + b)}{M} \tag{3}$$

where $\mathbf{1}$ is a vector of ones of M components. Hence, the vector of optimisation variables is:

$$\tilde{x} = \begin{bmatrix} x^+(1) \\ \vdots \\ x^+(N) \\ x^-(1) \\ \vdots \\ x^-(N) \\ B_0 \end{bmatrix}$$

With 2N+1=1057 components

It is useful to remind for the further calculations that using the vector of optimisation variable $\tilde{x}$ and reminding equation 2 we have this equality:

$$\begin{bmatrix} A & -A & 0 \end{bmatrix} \tilde{x} = Ax$$

The following necessary step is to express the constraint 1 linearly with respect to the optimisation variables. First, by using $\tilde{x}$ and by splitting the absolute value in two inequalities we have:

$$\begin{aligned} \frac{\begin{bmatrix} A & -A & \mathbf{0} \end{bmatrix} \tilde{x} + b}{S} - \mathbf{1} \leq \epsilon \\ \frac{\begin{bmatrix} A & -A & \mathbf{0} \end{bmatrix} \tilde{x} + b}{S} - \mathbf{1} \geq -\epsilon \end{aligned}$$

by manipulating these inequalities it is possible to write:

$$\begin{aligned} \begin{bmatrix} A & -A & -\mathbf{1} - \epsilon \end{bmatrix} \tilde{x} \leq -b \\ \begin{bmatrix} -A & A & \mathbf{1} - \epsilon \end{bmatrix} \tilde{x} \leq b \end{aligned}$$

The same can be done for 3, which can be written as:

$$\begin{bmatrix} \mathbf{1'} * A & \mathbf{1'} * (-A) & -M \end{bmatrix} \tilde{x} = -\mathbf{1}^T * b$$

It's now possible to write the program linearly as:

$$
\begin{aligned}
&\min \sum_{i=1}^{N} \left(x^+(i) + x^-(i)\right) \\
&\text{s.t.} \\
&\begin{bmatrix} -A & A & \mathbf{0} \end{bmatrix} \tilde{x} \leq b \\
&\begin{bmatrix} A & -A & -\mathbf{1} - \epsilon \end{bmatrix} \tilde{x} \leq -b \\
&\begin{bmatrix} -A & A & \mathbf{1} - \epsilon \end{bmatrix} \tilde{x} \leq b \\
&x^+(i) - x^-(i) \leq 0.01 \quad i = 1, \dots, N \\
&- x^+(i) + x^-(i) \leq 0.01 \quad i = 1, \dots, N \\
&x^+(i) \geq 0 \quad i = 1, \dots, N \\
&x^-(i) \geq 0 \quad i = 1, \dots, N \\
&\begin{bmatrix} \mathbf{1}'A & \mathbf{1}'(-A) & -M \end{bmatrix} \tilde{x} = -\mathbf{1}'b
\end{aligned}
$$

Hence we can now define the matrices that are used in the linporg() MATLAB function to solve the program.

$$
\begin{aligned}
&\min \mathbf{f}^T \tilde{x} \\
&\text{s.t.} \\
&\quad \tilde{A}\tilde{x} \leq \tilde{b}, \\
&A_{\text{eq}}\tilde{x} = b_{\text{eq}},
\end{aligned}
$$

where

$$
f = \begin{bmatrix} \mathbf{1}_{N\times 1} \\ \mathbf{1}_{N\times 1} \\ 0 \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} -A & A & 0 \\ A & -A & -\mathbf{1} - \epsilon \\ -A & A & \mathbf{1} - \epsilon \\ I & -I & 0 \\ -I & I & 0 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} b \\ -b \\ b \\ 0.01 * \mathbf{1}_{N\times 1} \\ 0.01 * \mathbf{1}_{N\times 1} \end{bmatrix},
$$

and

$$
A_{\text{eq}} = \begin{bmatrix} \mathbf{1}' * A & \mathbf{1}' * (-A) & -M \end{bmatrix}, \quad b_{\text{eq}} = -\mathbf{1}' * b.
$$

Using linprog(), the minimiser $\tilde{x}^*$ is found. From this, we can construct the original vector to get the desired x^*. The objective function value is: $\|x^*\|_1$=0.1321 and the mean B_0=2.9

By analysing x^* is possible to see that 487 components of 528 are zeros.

The plot of x^* is shown in Figure 1.1

1.2 L2 minimisation

For the minimisation of the L2 norm of x the only change with the L1 minimisation is the objective function:

$$
\begin{aligned}
&\min \|x\|_2 \\
&\text{s.t.} \\
&\left| \frac{Ax + b}{\text{mean}(Ax + b)} - \mathbf{1} \right| \leq \epsilon \\
&Ax + b \geq 0 \\
&|x(i)| \leq 0.01 \quad i = 1, 2, \dots, N
\end{aligned} \tag{4}
$$

where,

$$
\|x\|_2 = \sqrt{\sum_{i=1}^{N} x(i)^2}
$$

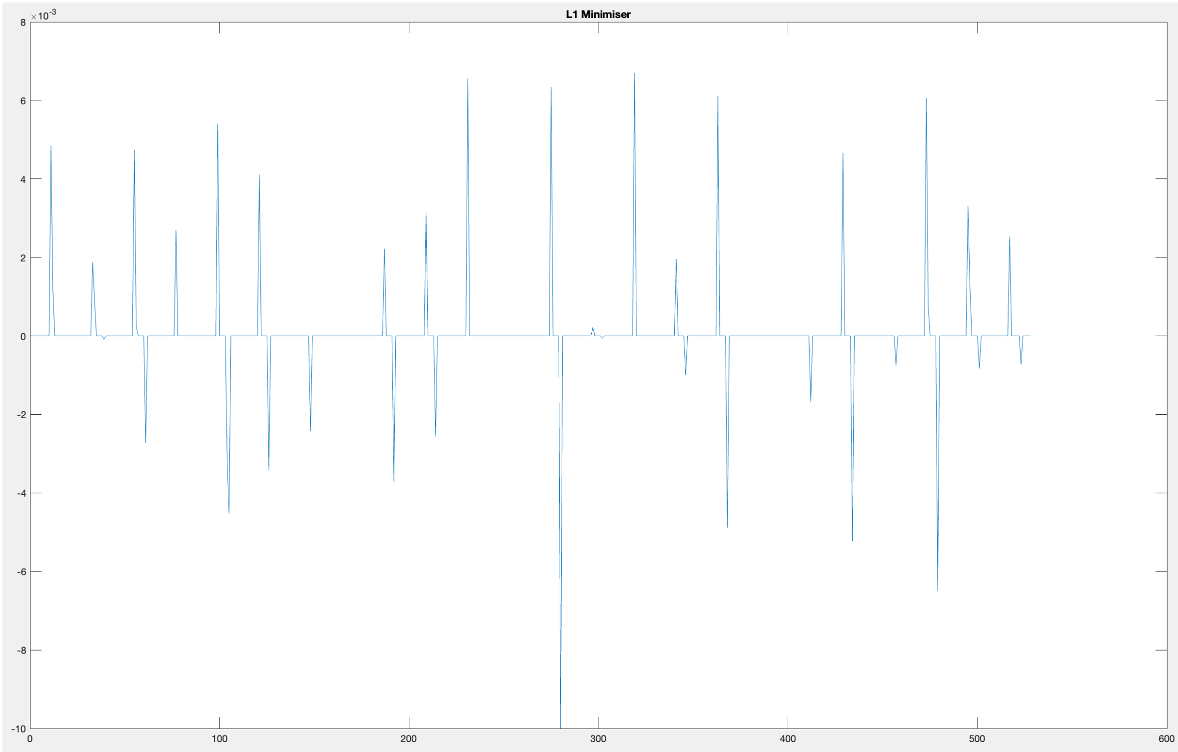

Figure 1: Minimiser of L1 minimisation

This objective function cannot be expressed as a linear function with the previous trick 2, this is why the optimisation routine has to be performed using the MATLAB function fmincon() and the objective function has to be specified with a MATLAB function as done in the codes.

As already done fot he L1 minimisation, the mean B_0 has to be considered as an additional optimisation variable by adding the constraint 3

In using fmincon(), we can choose to change the optimisation algorithm. The default algorithm is interior-point, however this procedure does not converge to a feasible solution since it stops prematurely due to the limit on the maximum function evaluations. To reach a feasible solution the algorithm Sequential Quadratic Programming (SQP) has to be used and specified in the options of fmincon().

In addition, there are two possible approaches of solving this program on MATLAB. The first one is by using the linear constraints defined already for the L1 minimisation. The second is by defining a MATLAB function nonlincon() to express the constraint (1) By comparing the two approaches resulted that the one using the non linear definition of the the constraint (1) is faster of 9 percent reaching a solution around 3 minutes.

The result of the optimisation shows that the mean B_0=2.9 and the objective function is $\|x^*\|_2$= 0.0108

In Figure 1.2 the L2 minimiser is depicted. It is possible to see that the L1 solution is sparser than this one since in the L2 minimisation there are no components of x^* that are actually equal to zero. On the other hand, in the L2 minimisation these components show to be an order of magnitude lower than the ones that are different to zero in the L1minimisation.

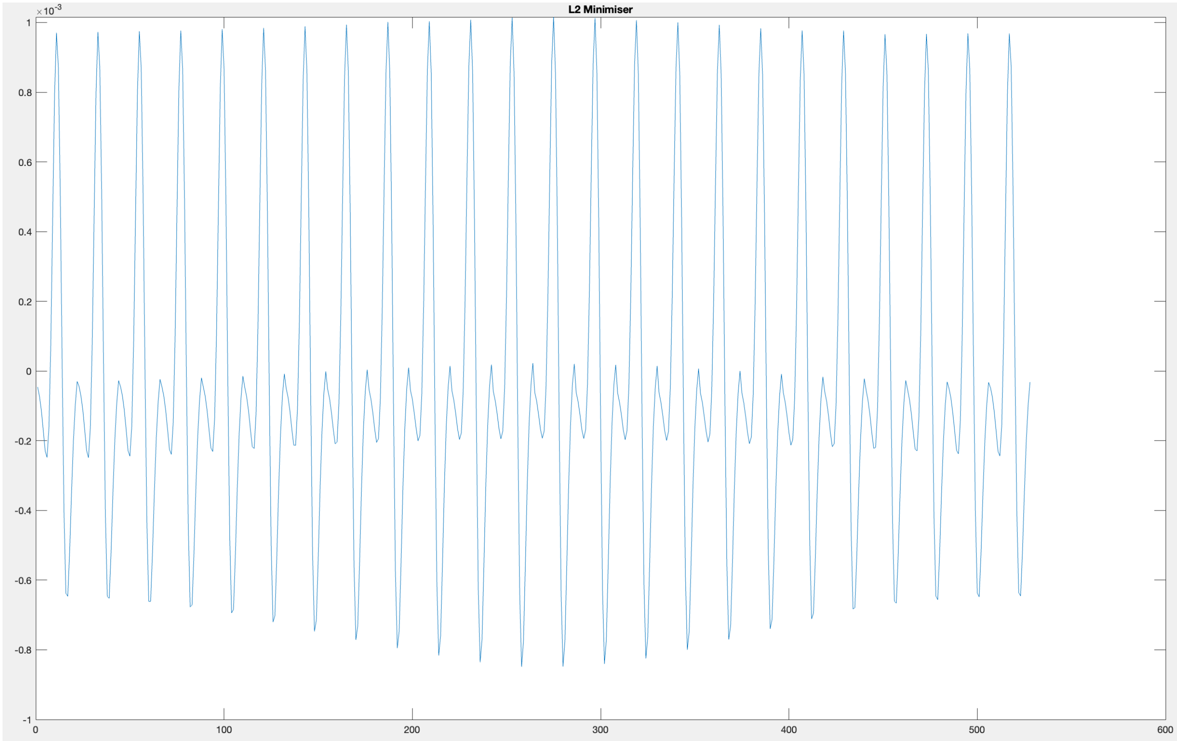

Figure 2: Minimiser of L2 minimisation

1.3 L0 minimisation

Minimizing the L0 norm of a vector, which counts the number of non-zero elements in the vector, is a non-convex and nondeterministic Polynomial time hard (NP-hard) problem in most cases. Since the L0 norm is a not differentiable and not smooth non linear function, direct minimization is computationally intensive and impractical for large optimisation variables.

The approach that is followed is to use a function that approximate the L0 norm function. In particular, it is shown that the sum of improved sigmoid function approximates the L0 norm with different property [5]. The aforementioned function is:

$$F_\sigma(x) = N - \sum_{i=1}^{N} \frac{2}{1 + \exp\left(\frac{\|x_i\|}{\sigma^2}\right)}$$

where σ is parameter that can be tuned with the property that when $\sigma \to 0$, then $F_\sigma(x) = N - \sum_{i=1}^{N} \frac{2}{1+\exp\left(\frac{\|x_i\|}{\sigma^2}\right)} \approx \|x\|_0$.

As done for L1 and L2 minimisation the mean B_0 has to be added in the optimisation variables. The first approach has been to minimise the improved sigmoid function optimisation by using the non linear constraints employed in L2 minimisation. With this strategy the solution was obtained after 12 minutes and was not sparse .

The second approach resulted in a very improved solution.
As done for L1 minimisation 2 the absolute of x_i value has been substitute by two optimisation variable and the definition of the linear constraints has been employed as in L1 minimisation.

The following program is representative

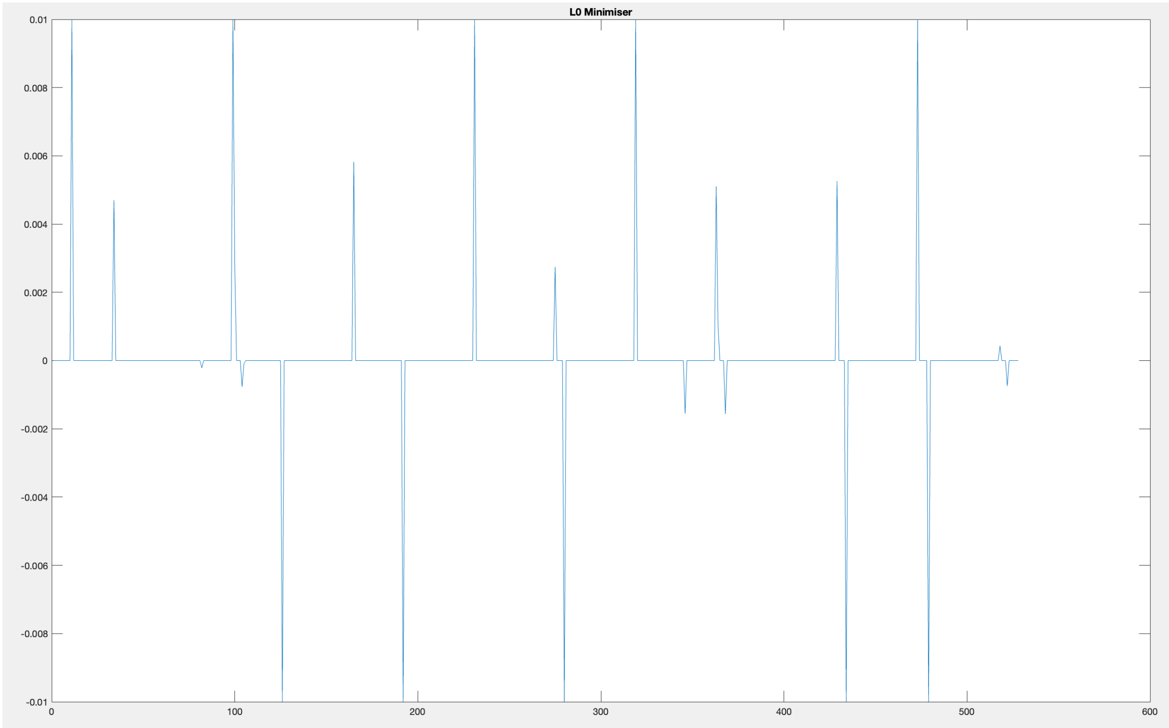

Figure 3: L0 minimiser

$$
\begin{aligned}
&\min N - \sum_{i=1}^{N} \frac{2}{1+\exp\left(\frac{x^+(i)+x^-(i)}{\sigma^2}\right)} \\
&\text{s.t.} \\
&\begin{bmatrix} -A & A & \mathbf{0} \end{bmatrix} \tilde{x} \leq b \\
&\begin{bmatrix} A & -A & -\mathbf{1}-\epsilon \end{bmatrix} \tilde{x} \leq -b \\
&\begin{bmatrix} -A & A & \mathbf{1}-\epsilon \end{bmatrix} \tilde{x} \leq b \\
&x^+(i) - x^-(i) \leq 0.01 \quad i = 1, \dots, N \\
&-x^+(i) + x^-(i) \leq 0.01 \quad i = 1, \dots, N \\
&x^+(i) \geq 0 \quad i = 1, \dots, N \\
&x^-(i) \geq 0 \quad i = 1, \dots, N \\
&\begin{bmatrix} \mathbf{1}'A & \mathbf{1}'(-A) & -M \end{bmatrix} \tilde{x} = -\mathbf{1}'b
\end{aligned}
$$

The solution x^* in Figure 3 is achieved in 18 seconds. By inspecting the vector solution is possible to see that some components are not exactly equal to zero but are lower than 10^{-19}. Hence, by rounding to zero these components is possible to see that the number of non zero components of x^* is 26. This solution can be compared to the one obtained in the L1 which showed 41 components different from zero.

2 Part B

2.1 Introduction

Linear algebra is utilised in solving a myriad of practical problems. In this report, it will be illustrated how the concepts of Rank and Singular Value Decomposition (SVD) are employed in the field of control theory to obtain a model of a system from a practical experiment. The focus is on the Subspace-based state-space system identification algorithm (4SID), which provides a numerically reliable way to retrieve a Linear Time Invariant (LTI) state-space model for a complex multivariable dynamical system from measured data.

2.2 Theory and Methods

A dynamical system is a system whose evolution is described through differential equations. In particular, LTI systems behave following linear differential equations which do not vary in time. The state space representation of an LTI system is used in several control techniques and allows to have a clear understanding of how the system evolves in time and how the system parameters are involved. The state-space representation of a LTI system is given by the equations:

$$\mathbf{x}(t+1) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) \tag{5}$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{6}$$

where

- $\mathbf{x}(t)_{n\times 1}$is the state vector,
- $\mathbf{u}(t)_{m\times 1}$ is the input vector,
- $\mathbf{y}(t)_{l\times 1}$ is the output vector,
- $\mathbf{F}_{n\times n}$ is the system matrix,
- $\mathbf{G}_{n\times m}$ is the input matrix,
- $\mathbf{H}_{l\times n}$ is the output matrix,
- $\mathbf{D}_{l\times m}$ is the feedthrough (or direct transmission) matrix,
- n is the order of the system and represents the minimum number of the components of the state that describe the behavior of the system.

Furthermore, real world physical systems are Strictly Proper, this means that the input at time t influences the output at time $t+1$ and is represented as $\mathbf{D} = \mathbf{0}$.

The aim of the 4SID algorithm is to retrieve the matrices F,G,H,D from measured data. The experiment that has to be carried out to perform this identification process is the response to impulse. It consists in gathering the output of the system when the input is an impulse.
Before delving deeper in the theory of the 4SID, some considerations have to be done about the observability and controllability matrices of a system.

Given the state-space representation, a system is fully obeservable if the observability matrix

$$\Theta_n = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} \tag{7}$$

is fully rank ($rank(\Theta_n) = n$).

In practice, when a system is fully observable it means that by watching the output signal $\mathbf{y}(t)$ is possible to observe the full state of the system. Regarding controllability, a system is fully controllable if the controllability matrix:

$$\Gamma_n = \begin{bmatrix} G & FG & \cdots & F^{n-1}G \end{bmatrix} \tag{8}$$

is full rank ($rank(\Gamma_n) = n$).

The meaning of this property is that by choosing the input signal $\mathbf{u}(t)$ is possible to control the full state of a system.

Having done these considerations, we can start discussing the steps of the 4SID algorithm. Firstly, an impulse is injected in the system through the input signal $\mathbf{u}(t)$ and

$$N \approx 100 \text{ or } 1000 \text{ samples}$$

of the output signal $\mathbf{y}(t)$ are gathered. Let's call these samples $h_1, h_2, h_3, \cdots, h_N$.

This data-set is used to build a matrix called Hankel matrix:

$$\mathbf{H}_{q\times d} = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & h_d \\ h_2 & h_3 & h_4 & \cdots & h_{d+1} \\ h_3 & h_4 & h_5 & \cdots & h_{d+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_q & h_{q+1} & h_{q+2} & \cdots & h_{q+d-1} \end{bmatrix}$$

where $q + d - 1 = N$ and the choice of q and d generically does not change the performance of the algorithm. As rule of thumb the choice is usually $q > d/2$ [2].

Furthermore, it is possible to prove that in absence of noise the Hankel matrix can be constructed only using $2n$ samples and showing the following property [6]:

$$\mathbf{H}_{n+1\times n+1} = \Theta_{n+1}\Gamma_{n+1} = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^n \end{bmatrix} \begin{bmatrix} G & FG & \cdots & F^nG \end{bmatrix} \tag{9}$$

with $rank(\mathbf{H}_{n+1\times n+1}) = n$. However, this route cannot be taken because of the presence of noise in the experimental data. In this case we need N samples from the experiment resulting in a larger and more complex identification procedure.

The Hankel matrix will typically be of full rank when noise is present. It means that the rank of the Hankel matrix is equal to the $max(q, d)$. The goal now is to perform an optimal rank reduction of the Hankel matrix to estimate the order of the system. n To do so the SVD method is employed [6].

$$\mathbf{H}_{q\times d} = \mathbf{U}_{q\times q}\mathbf{S}_{q\times d}\mathbf{V}^T_{d\times d}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{S}$ is a rectangular diagonal matrix that shows the singular values on his diagonal . Considering that the rule of thumb $q > d/2$ and the constraint $q + d - 1 = N$ are respected, the rank of the Hankel matrix is q ,it means that there are q singular values in S.

$$S = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_q & 0 & \cdots & 0 \end{bmatrix}$$

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_q$$

The scope now is to estimate the order of the system from the Hankel matrix. By analysing the singular values distribution in S is possible to properly truncate the SVD reaching the optimal rank reduction of the Hankel matrix. The Figure 4 from [1], is representative of the case where the

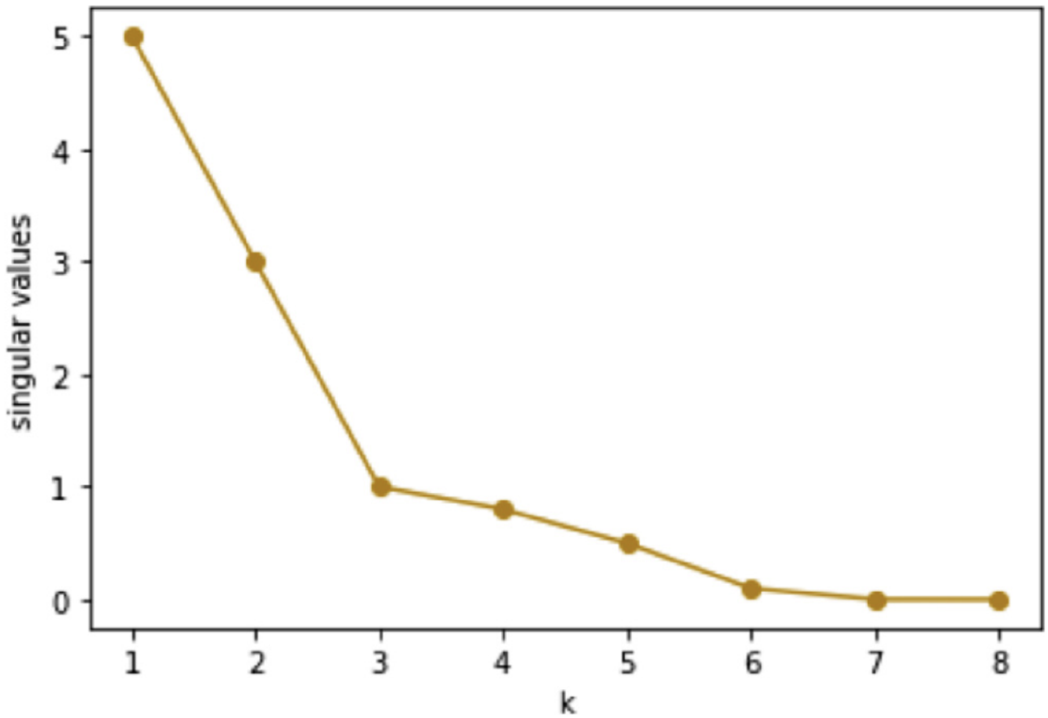

Figure 4: Singular values distribution example

sequence of singular values can be cut off at $k = 3$ resulting in an estimation of the order of the system equal to three. This is possible since the sequence of singular values shows a clear change in behavior describing a "knee" distribution. The procedure corresponds in separating the useful information about the system behavior from the experimental noise.

Hence, by truncating $\mathbf{S}_{q\times d}$ in $\hat{S}_{n\times n}$, $\mathbf{U}_{q\times q}$ in $\hat{U}_{q\times n}$ and $\mathbf{V}^T_{d\times d}$ in $\hat{V}^T_{n\times d}$ is possible to obtain the Hankel matrix $\hat{H}_{q\times d}$ with the rank reduced to n :

$$\hat{H}_{q\times d} = \hat{U}_{q\times n}\hat{S}_{n\times n}\hat{V}^T_{n\times d} = \hat{U}\hat{S}^{1/2}\hat{S}^{1/2}\hat{V}^T \tag{10}$$

It can be proven from here that $\hat{H}_{q\times d}$ can be obtained from the extended observability and controllability matrix [6]:

$$\hat{H}_{q\times d} = \hat{\Theta}_{q\times n}\hat{\Gamma}_{n\times d} \tag{11}$$

Hence, by using equations 10 and 11 it possible to write:

$$\hat{\Theta}_{q\times n} = \hat{U}_{q\times n}\hat{S}^{1/2}_{n\times n},$$

$$\hat{\Gamma}_{n\times d} = \hat{S}^{1/2}_{n\times n}\hat{V}^T_{n\times d}$$

Furthermore, we know from 8 and 7 that the first row of the observability matrix is H and the first column of the controllabolity matrix is G. Thus, the estimated $\hat{H}$ can be obtained by taking the first row of the extended observability matrix $\hat{\Theta}$ and the estimated $\hat{G}$ can be obtained by taking the first column of the extended controllabilty matrix$\hat{\Gamma}$.

Finally to retrieve F is necessary to point out that from the definition of the observability matrix (the controllability matrix can be used as well) :

$$\hat{\Theta}_1 = \hat{F}\hat{\Theta}_2$$

where $\hat{\Theta}_1$ is equal to the observability matrix excluded the last row $\hat{\Theta}(1 : q-1; :)$ and $\hat{\Theta}_2$ is equal to the observability matrix excluded the first row $\hat{\Theta}(2 : q; :)$.

From here is possible to obtain to obtain the estimation of F by using the "pseudoinverse" of $\hat{\Theta}_1$:

$$\hat{F} = \left(\Theta_1^T \Theta_1\right)^{-1} \Theta_1^T \Theta_2 = \Theta_1^{\dagger} \Theta_2 \tag{12}$$

2.3 Results and discussions

Signal processing, control engineering, and system identification are the main applications for the 4SID method. Two instances from real life where the 4SID algorithm is used are:

1. Structural health monitoring of civil infrastructure: Applications for the 4SID method include modal analysis of civil structures, such as high-rise buildings and bridges, and structural health monitoring. The 4SID approach was applied for output-only modal identification of a high-rise tower under abnormal loading conditions, such as strong winds, in the publication "Stochastic subspace identification for output-only modal analysis: Application to super high-rise tower under abnormal loading condition" by Liu et al [4]. The tower's dynamics and structural integrity were evaluated with the aid of the identified model.

2. Modeling of building thermal dynamics: Building heat dynamics have been modelled using the 4SID method for purposes such as model predictive control of heating systems. Bacher and Madsen's study "Identifying suitable models for the heat dynamics of buildings" [3] describes how they applied the 4SID method to find linear state-space models that accurately represented the thermal properties of different types of structures. The temperature of the inside was then predicted using these models.

These are only two examples of how the 4SID is used in many engineering practical application. It must be emphasised that without the help of the SVD this algohythm would have remained only a theoretical method and not feasible in practice due to the noise presence in empirical data.

2.4 Conclusions

The practical implementation of the 4SID approach for finding models of Linear Time-Invariant (LTI) systems has shown the importance of the Singular Value Decomposition (SVD) and the notion of matrix rank. However, this is only an example of how advanced linear algebra is crucial for real world applications. The reported case wanted to highlight how the use of SVD in lowering data dimensions and cutting off the noisy data, as well as the rank's capacity to identify system observability and controllability can open to simple and effective methodologies in a fundamental task such as the identification of system's models.

References

[1] Antonella Falini. A review on the selection criteria for the truncated svd in data science applications. *Journal of Computational Mathematics and Data Science*, 5:100064, 2022.

[2] Maryam Fazel, Ting Kei Pong, Defeng Sun, and Paul Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977, 2013.

[3] Thomas Hasfjord. Design and implementation of a kalman filter based estimator for temperature control. Master's thesis, Institutt for teknisk kybernetikk, 2014.

[4] Shieh-Kung Huang, Fu-Chung Chi, and Yuan-Tao Weng. Practical implementation of recursive subspace identification on seismically excited structures with fixed window. *Applied Sciences*, 12(21):10841, 2022.

[5] JianJun Li, Hangjun Che, and Xiaoyang Liu. Circuit design and analysis of smoothed l 0 norm approximation for sparse signal reconstruction. *Circuits, Systems, and Signal Processing*, 42(4):2321–2345, 2023.

[6] Mats Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851, 1995.