Computer lab 5

Kristiaan Pelckmans, Viktor Bro

1 Computer Lab 5: Subspace Identification

1.1 Goals

This computerlab brings you along some computational tools that implement subspace identification methods for deterministic, stochastic and combined deterministic-stochastic systems. Specifically, we advocate of the tools implemented in the MATLAB SI toolbox.

Again, I need for each of you a named 1-page report (for official reasons), but do send me an email if you have specific content related questions. This computer lab should be seen as a step-stone for the project works, and I advise you to explore the function files and help as a function of what you actually need for your project.

1.2 MATLAB SI

Let us first walk through the functionality of the MATLAB SI toolbox w.r.t. subspace identification. As an example consider the 3-by-2 MIMO system generating the signals u^1, u^2, u^3, y^1, y^2 as given as follows

$$\begin{cases} \mathbf{x}_{t+1} &= \begin{bmatrix} 1 & 0.81 \\ -0.11 & \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} 1 & 2 & 3 \\ -1 & -2 & -3 \end{bmatrix} \mathbf{u}_t \\ \mathbf{y}_t &= \begin{bmatrix} 2 & 1 \\ 11 & \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} 1 & 2 & 3 \\ -1 & -2 & -3 \end{bmatrix} \mathbf{u}_t + \mathbf{e}_t \end{cases} \quad \forall t = 1, 2, \ldots, n$$

for $n = 100$ and $\mathbf{x}_0 = (0, 0)^T$. In MATLAB this system is simulated as

```
>> U = randn(100,3);
>> A = [1,-0.99 ;0.1, 0.7];
>> B = [1 2 3; -1 -2 -3];
>> C = [1 0; -1 1];
>> D= [1 2 3; -1 -2 -3];
>> K = [0 0; 0 0];
>> m = idss(A,B,C,D,K);
>> Y = sim(m,U,'Noise');
>> bode(m)
```

Then the N4SID implementation of the MATLAB SI toolbox is called as

```
>> z = iddata(Y,U);
>> m1 = n4sid(z,1:5,'Display','on');
>> bode(m1,'sd',3);
```

Observe that the order n rolls out quite naturally from the algorithm implementing the subspace technique (specifically, from the number of significant nonzero singular values from the 'realization' step in the implementation).

Another approach to compare the dynamics of the system `m` and of the estimated model `m1` is given by comparing the eigenvalues of $\mathbf{A}$ and `m1.A`. This approach is to be contrasted to a PEM approach which was extended for handling MIMO data

```
>> z = iddata(Y,U);
>> m2 = pem(z,5,'ss','can')
```

Now this can be done using the GUI of the SI toolbox, type

```
>> ident
```

Now the question for you is how you can compare the models m, m_1, m_2 on new input signals

```
>> Ut = randn(100,3);
```

Which one is better? What is the price for this superiority? Happy clicking!

1.3 Timeseries

Now consider the case of a multivariate timeseries instead, where at each timestep t one has that $\mathbf{y}_t \in \mathbb{R}^3$. Consider the system underlying the observations $\mathbf{y}_1, \mathbf{y}_2, \ldots$ given as

$$\begin{cases} \mathbf{x}_{t+1} & = \begin{bmatrix} 1 & 0.81 \\ -0.11 & \end{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t & = \begin{bmatrix} 2 & 1 \\ 11 & \end{bmatrix} \mathbf{x}_t + \mathbf{e}_t \end{cases} \quad \forall t = 1, 2, \ldots, n$$

for $n = 100$ and $\mathbf{x}_0 = (0, 0)^T$. In MATLAB this system is simulated as

```
>> A = [1,-0.81 ;0.11, 0];
>> C = [2 1; 11 0];
>> m = idss(A,[],C,[]);
>> data=sim(m, iddata(zeros(n,1),[]), simOptions('AddNoise',true));
>> Y=get(data,'OutputData');
```

Then the N4SID implementation of the MATLAB SI toolbox is called as

```
>> z = iddata(Y,[]);
>> m2 = n4sid(z,2);
```

Compare m and $m2$. What values of n are good?