16.7 Computer Lab 5: Subspace Identification

16.7.1 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, and in the software accompanying the book of Van Overschee and De Moor. Other implementations can be found e.g. at

- MATLAB SI (Linköping) toolbox.
- N4SID (KULeuven) - see the files coming with this computer lab.
- CUEDSID (Cambridge): `http://www-control.eng.cam.ac.uk/jmm/cuedsid/`
- MACEC (KULeuven): `http://bwk.kuleuven.be/bwm/macec/`
- LTI toolbox (TUDelft): `http://www.dcsc.tudelft.nl/~datadriven/lti/ltitoolbox_product_page.html`

Do have a look at those tools in function of your chosen project.

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. However, this computer lab should be seen as a step-stone for the project works, and explore the function files and help in function what you need for your project.

16.7.2 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 $n = 100$. 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!

16.7.3 N4SID

While the MATLAB SI is carefully designed, it is not so well suited for dealing with large dimensional systems. The tools provided by Van Overschee and De Moor's book are perhaps better suited in such case. To get acquainted with those tools, run the tool

```
>> sta_demo
```

And follow the instructions on the screen. Follow the basic steps in the demo to identify the data as computed in the previous subsection. Specifically, use the command `subid` and `predic` and see how results deviate from the n4sid function in the MATLAB SI toolbox.

16.7.4 Timeseries

The previous tool does as well implement subspace techniques for purely stochastic models. Again, walk through the demo

```
>> sto_demo
```

Use this insight to identify a model from the observations y^1, y^2 in `stodata.mat`. What order is the underlying system? Now use this data in the MATLAB SI toolbox. Are the estimated models comparable? Compare the order selection methods implemented in either toolbox.