1 The MATLAB Files

This site contains three main MATLAB script files, `Fig1.m`, `Fig2.m` and `Fig3.m` to be run with MATLAB. The scrips reproduce the essence of the corresponding figures in article

- *Kernel methods in system identification, machine learning and function estimation: A survey*, by Gianluigi Pillonetto, Francesco Dinuzzo,Tianshi Chen, Giuseppe De Nicolao and Lennart Ljung, *Automatica, March 2014*

The scrips rely upon some m-file functions that implement the routines describes in the article:

- `arx_with_regularization_using_SITB`

  This implements the routines `TC,SS,DC` in Sections 7.2 and 7.3, using the general regularization software in the Identification Toolbox.

- `oe_with_oracle_and_with_AIC`

  This implements the routines `oe+Or1` and `oe+Or2` in Section 7.2

- `oe_with_CV`

  This implements the routine `oe+CV` in Section 7.2

- `armax_with_oracle_and_with_AIC`

  This implements the routines `PEM+Oracle, PEM+BIC` and, `PEM+ AICc` in Section 7.3.

- `armax_with_CV`

  This implements the routine `PEM+CV` in Section 7.3

- `DataGeneration_Sect7_2`

  This routine generates random systems and data used in Section 7.2.

- `DataGeneration_Sect7_3`

  This routine generates random systems and data used in Section 7.3.

- `arx_with_regularization_special_code`

  This code is written from scratch for both FIR and ARX regularization

as well as the following mat file

- `decimatedRobotArm.mat`

  This is the data file generated by experiments with a robot arm. The authors thank Jan Swevers (dept Mechanical Engineering, KU Leuven, Belgium) for making the experimental data available so that the users can verify the proposed algorithms themselves. A detailed description of the setup and the experiment is given in

> *Comparison of Two Feedforward Design Methods Aiming at Accurate Trajectory Tracking of the End Point of a Flexible Robot Arm* by Dirk E. Torfs, Rudi Vuerinckx, Jan Swevers, and Johan Schoukens *IEEE Transactions on Control Systems Technology, Vol. 6, No. 1, January 1998*, pp 2-14.
>
> In this mat-file the data have been decimated by a factor 10.

2 Two Versions of the Code

Two versions are provided: One that uses the regulariation software provided in the System Identification Toolbox (version RMatlab2014a or higher, System Identitication toolbox ver 9.0 or higher). The other version uses special code developed in he course of writing the paper (`arx-with-regularization-special-code`). That code is also provided.

The reason for providing the two version is to be able to reasonable well reproduce the figures in the paper with the special code, but also to show that very similar results are obtainable with commercially available, general purpose software.

The special code has given some ideas for how the toolbox routines can be improved, especially for short data records. Such updates will be included in upcoming versions of the toolbox.

3 How to Run the Tests

Download the files and run the script files.

4 Possible Discrepancies

The results will not necessarily be exactly as in the article. First, the systems and data sets are generated by random numbers, so the systems do not exactly match those in the article. Second, the simulations in the article were run with code specially developed for the purpose. The algorithms have since been included in the System Identification Toolbox with slightly different implementations, but following the same principles.

One notable deviation is that the kernels for the regularized ARX models in Section 7.3 were tuned to have the scale factors for all inputs and indicidually tuned decay rates, while they are all separately tuned in the SITB code here.

Also, the robot arm data has been decimated for reasonable file size. That affects the figures of fit, but not the main message.

5 Timing

Most of the time in the code is spent on estimating PEM models of all orders up to MaxOrder (=30 in the article). One such command, like `oe-with-CV` takes about 2 minutes, while a call to `arx-with-regularization` takes about 2 seconds. You could start by just running a fex Monte-Carlo runs, to see the timing, and then do a bigger batch for the statistics. You could also lower MaxOrder, since most selected orders are in fact less than 10.