The Behavioral Toolbox

Ivan Markovsky

The Behavioral Toolbox is a collection of Matlab functions for analysis and design of dynamical systems using the behavioral approach to systems theory and control. It implements newly emerged nonparameteric data-driven methods for linear time-invariant systems. In order to install it, download and unpack the archive file of the functions. A tutorial for its usage (and implementation details) are given in this document. In order to cite it, please refer to:

> I. Markovsky. The behavioral toolbox. In Proc. of Machine Learning Research, volume 242, pages 130–141, 2024.

```
@InProceedings{bt-l4dc,
  Author =    {I. Markovsky},
  Title =     {The Behavioral Toolbox},
  booktitle = {Proc. of Machine Learning Research},
  volume =    {242},
  pages =     {130--141},
  Year =      {2024}
}
```

Examples

In the examples shown below, the given data is a trajectory of an underlying data-generating system. This trajectory is not partitioned into an input signal and an output signal and is assumed that it fully specifies the system. The goal then is to find a property of the (unknown) data-generating system or solve a problem involving the system directly from the data.

Finding the number of inputs and the order

The number of inputs and the order are properties of the system. Theie importance is due to the fact that they determine the system's complexity: The more variables are inputs and the higher the order is, the more complex the system is.

The function `w2c` of the toolbox computes the number of inputs and the order directly from data. Here is an example:

```
m = 1; p = 2; n = 3; Td = 100;  % simulation parameters
B = drss(n, p, m);              % random stable LTI system
wd = B2w(B, Td);                % random data trajectory
[c, mh, ell, nh] = w2c(wd);     % direct data-driven computation
[m == mh, n == nh] % -> [1 1]   % check if the computed result is correct
```

A sufficiently long random trajectory almost surely fully specifies the system. Quantifying what sufficiently long means, however, requires knowledge of the model's complexity (which is precisely what we aim to compute in this case). There is no way out of the dilemma: We either have to assume that the data fully specifies the system or else we have to assume that the complexity is given and then we can check whether the data fully specifies the system.

Finding an input/output partitioning of the variables

In general, a system with inputs (called an open system) admits multiple input/output partitionings. The function `BT2IO` finds all possible input/output partitionings.

Continuing the example above, first, using the function `w2BT`, we obtain a non-parametric representation of the data-generating system from the given data:

```
T = 2 * ell + 1; BT = w2BT(wd, T, c);
```

The finite-horizon non-parametric representation is the main one used in the toolbox. The horizon T should be chosen "sufficiently" large for solving the problem at hand. For find input/output partitionings of the variables, it should be at least $2\ell+1$, where ℓ is the lag the system—another property of the system related to its complexity. Since the lag was already computed by using the function `w2c` in the previous example, we reuse it here.

Once the representation `BT` is computed, we call `BT2IO` for finding all valid input/output partitionings of the variables:

```
IO = BT2IO(BT, m + p)
```

Its output `IO` is a matrix, the rows of which indicate the valid input/output partitionings as follows: has as first m variables `IO(:, 1:m)` indicate the indeces of the input variables and the remaining p variables `IO(:, m+1:end)` indicate the indeces of the output variables. For example,

```
ud = wd(:, IO(1, 1:m)); yd = wd(:, IO(1, m+1:end));
```

is the first input/output partitioning obtained.

For a random example with $m = 1$ inputs and $p = 2$ outputs, a possible output may be:

```
IO =

     1     2     3
     1     3     2
     2     1     3
     2     3     1
```

which shows that variables 1 and 2 but not 3 can be inputs.

Distance of a signal to a system

Distance between systems

Distance of a signal to being a trajectory of a bounded complexity system