Table of Contents

```
%
 #########################################################################
% TUHH :: Institute for Control Systems :: Control Lab
%
 #########################################################################
% Experiment CSTD2: Magnetic Levitation Plant
%
% Copyright Herbert Werner and Hamburg University of Technology, 2014
%
 #########################################################################
% This file is to be completed by the student.
% The completed version is to be published using
%   publish('cstd2_ident.m','pdf')
% and submitted as a pdf-file at least one week prior to the scheduled
 date
% for the experiment
%
% HINT 1:
% if you want to find out more about a certain command, just type
% 'help command' into the matlab window
% HINT 2:
% use section evaluations (Ctrl+Enter) to run the code within a single
% section

%----------------------------
% v.0.9 - 13-11-2014
% by Michael Heuer
%----------------------------
% Last modified on 25-11-2014
% by Julian Theis
% ---------------------------

clear all; clc; close all
```

I. Setup some global Variables which will be needed later

```
t_s = 0.01; % Sampling Time of the datasets
```

II. Load the measurment data into an iddata object

The data are stored in the vector called 'data' The first row is the time, the second and third u_1 and u_2, and the fourth and fifth y_1 and y_2. Those are copied into a iddata object which can be used for the identification.

```
load d_steps.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_steps = iddata([y_1, y_2], [u_1, u_2], t_s);

load d_chirp_1.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_chirp_1 = iddata([y_1, y_2], [u_1, u_2], t_s);

load d_chirp_2.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_chirp_2 = iddata([y_1, y_2], [u_1, u_2], t_s);

load d_prbs_1.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_prbs_1 = iddata([y_1, y_2], [u_1, u_2], t_s);

load d_prbs_2.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_prbs_2 = iddata([y_1, y_2], [u_1, u_2], t_s);

load d_noise_1.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_noise_1 = iddata([y_1, y_2], [u_1, u_2], t_s);

load d_noise_2.mat;
u_1 = data(:,2); u_2 = data(:,3);
y_1 = data(:,4); y_2 = data(:,5);
d_noise_2 = iddata([y_1, y_2], [u_1, u_2], t_s);
```

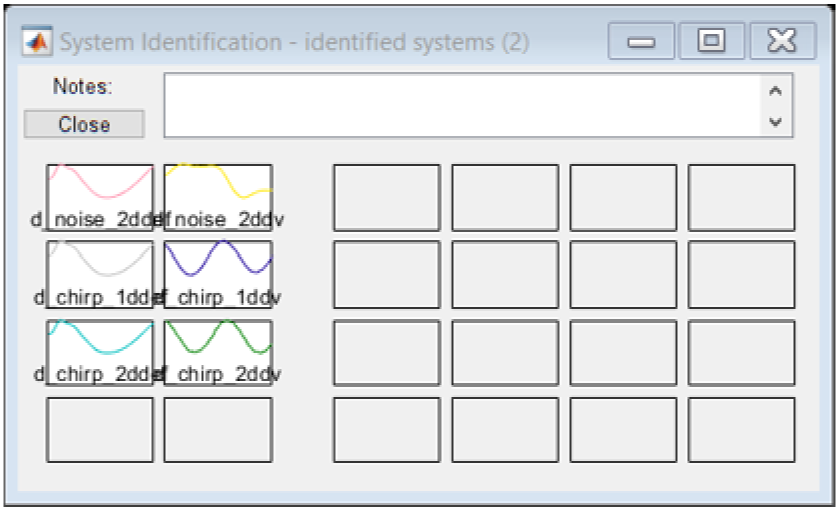
III. Open the System Identification Toolbox and performe the identification

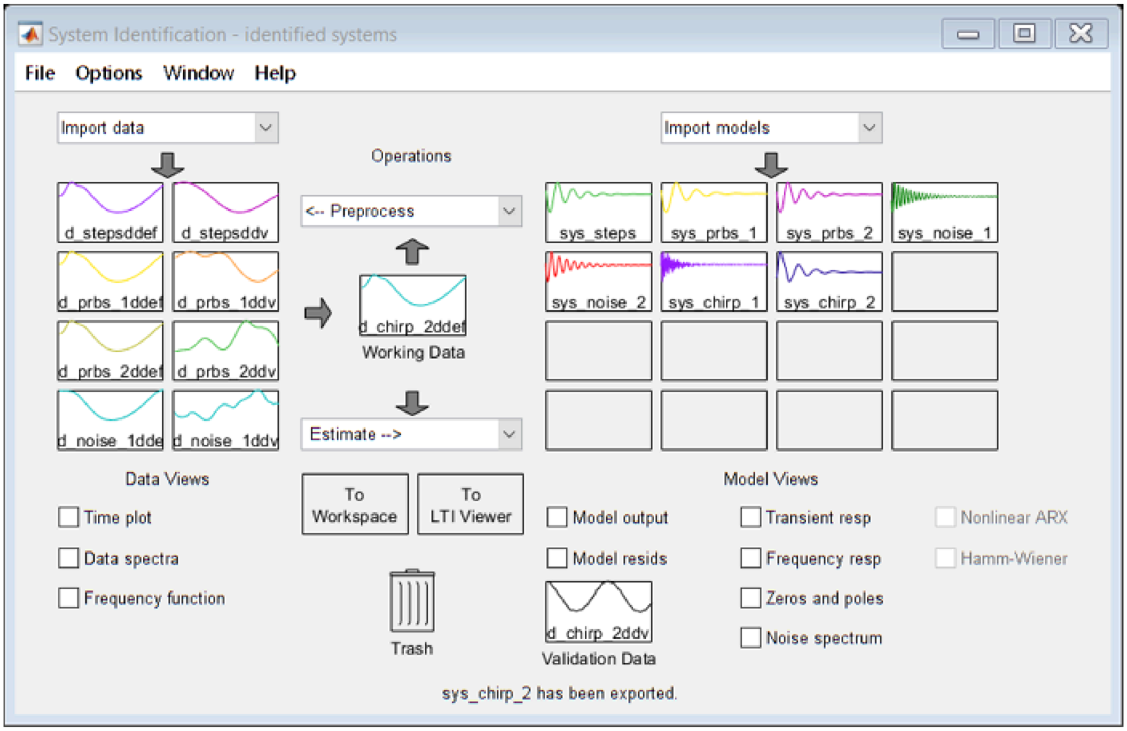
In this step the identification should be performed using the gui. Take a look at the lab document and performe the required steps.

Store the identified system in the pattern sys_* Where * is the dataset you used for the identification.

```
systemIdentification;
```

```
pause();
```





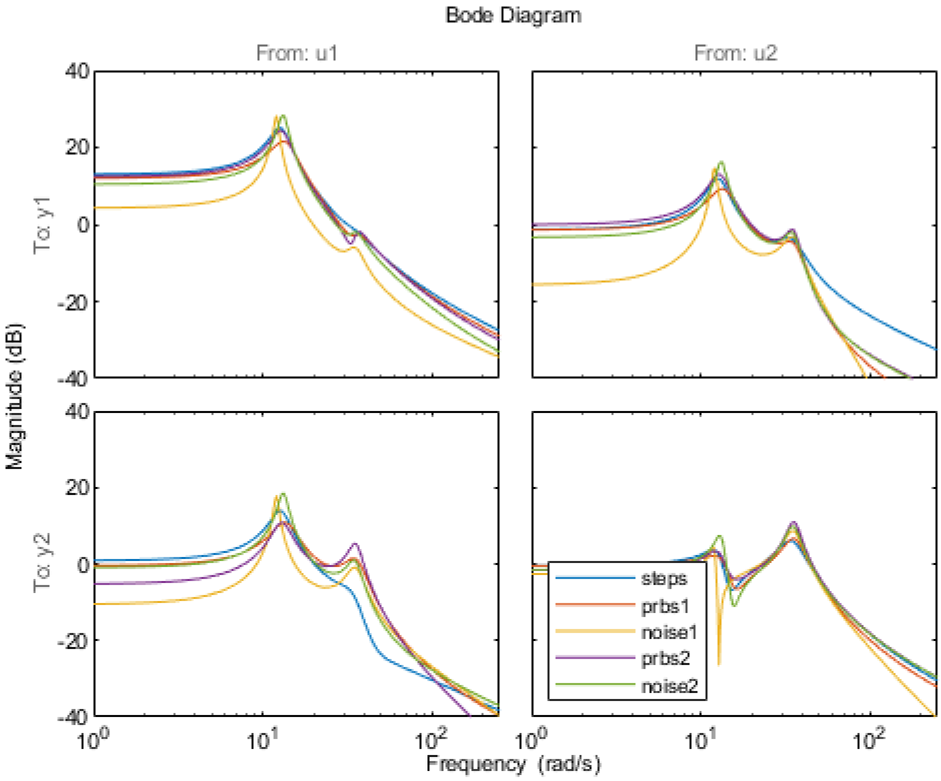
sys_chirp_2 has been exported.

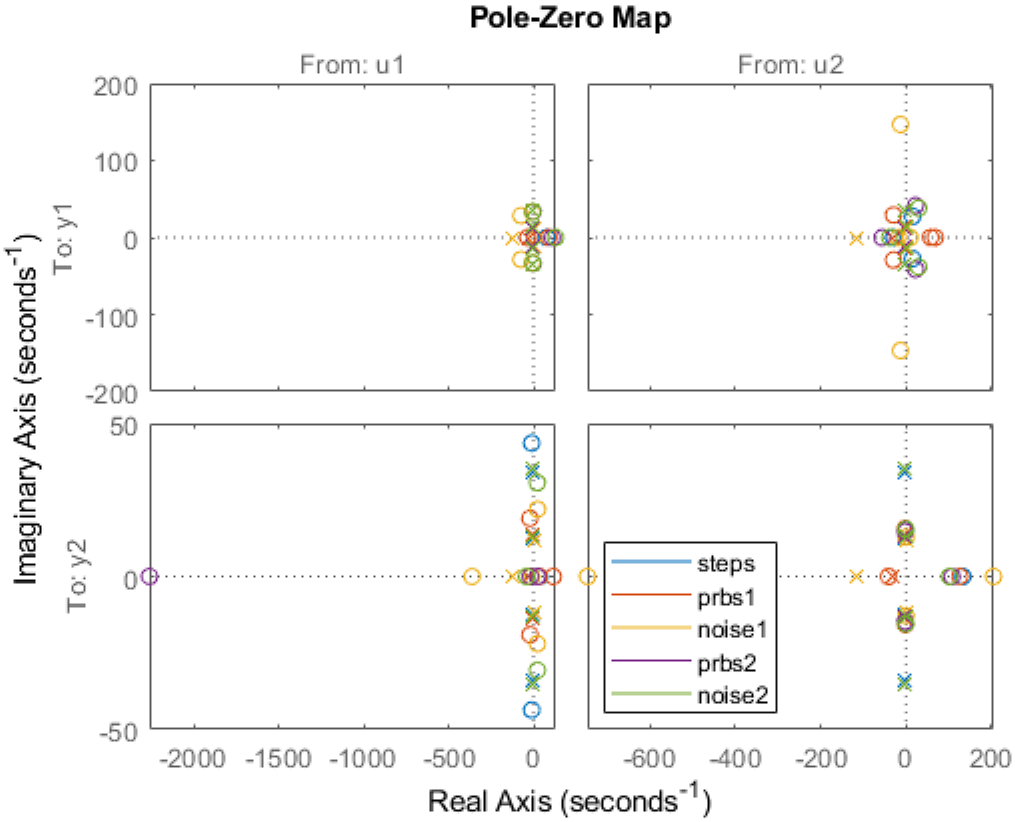
IV. Save the identified models

```
if (sys_steps.ts ~= 0) || (sys_chirp_1.ts ~= 0) || (sys_chirp_2.ts ~=
 0) || ....
   (sys_prbs_1.ts ~= 0) || (sys_prbs_2.ts ~= 0) || (sys_noise_1.ts ~=
 0) || (sys_noise_2.ts ~= 0)
    error('Please convert the systems to continuous time');
end

save models.mat sys_steps sys_chirp_1 sys_chirp_2 sys_prbs_1 sys_noise_1 sys_prbs_
```

V. Compare the models

```
% Compare Bode Plots
figure(1);
b = bodeoptions();
b.XLim = {[1,250]};
b.YLim = {[-40 40]};
bodemag(sys_steps, sys_prbs_1, sys_noise_1, sys_prbs_2,
 sys_noise_2,b);
legend({'steps','prbs1','noise1','prbs2','noise2'},'Location','SouthWest');

% Compare Poles
figure(2);
iopzmap(sys_steps, sys_prbs_1, sys_noise_1, sys_prbs_2, sys_noise_2);
legend({'steps','prbs1','noise1','prbs2','noise2'},'Location','SouthWest');
```





Published with MATLAB® R2019b