Systematic review of SysID - Different models and prediction error methods

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Jingxiao Liu
% Inferlab, CEE, CMU
% June, 2019
% Matlab live script for introducing different dynamic models and
% prediction error methods for identifying them
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Part 3. Prediction error methods

a) Compute $\hat{y}[k;\theta|k-1]$;

b) Form the prediction error $\epsilon[k] = y[k] - \hat{y}[k;\theta|k-1]$;

c) Construct the loss function $L_N = \frac{1}{N}\sum_{k=1}^{N}\epsilon^2[k]$;

d) $\theta^* = \underset{\theta}{\operatorname{argmin}}\, L_N$.

3.1 Different model types

3.1.1 Finite impulse response (FIR) model

$$y(t) = b_1u(t-1) + b_2u(t-2) + \cdots + b_{n_b}(t-n_b) + e(t) = \left(\sum_{k=1}^{n_b} b_k q^{-k}\right)u(t) + e(t) = B(q)u(t) + e(t)$$

if n_b is infinity, this is a infinite impulse response model

3.1.2 Autoregressive exogenous (ARX) model

$$y(t) + a_1y(t-1) + \cdots + a_{n_a}y(t-n_a) = b_1u(t-1) + b_2u(t-2) + \cdots + b_{n_b}(t-n_b) + e(t)$$

$$\left(\sum_{k=0}^{n_a} a_k q^{-k}\right)y(t) = \left(\sum_{k=1}^{n_b} b_k q^{-k}\right)u(t) + e(t)$$

$$y(t) = \frac{B(q)}{A(q)}u(t) + \frac{1}{A(q)}e(t)$$

3.1.3 Autoregressive moving average exogenous (ARMAX) model (equation error model family)

$$y(t)+a_1y(t-1)+\cdots+a_{n_a}y(t-n_a)=b_1u(t-1)+b_2u(t-2)+\cdots+b_{n_b}(t-n_b)+e(t)+c_1e(t-1)+\cdots+c_{n_c}e(t-n_c)$$

$$\left(\sum_{k=0}^{n_a}a_kq^{-k}\right)y(t)=\left(\sum_{k=1}^{n_b}b_kq^{-k}\right)u(t)+\left(\sum_{k=0}^{n_c}c_kq^{-k}\right)e(t)$$

$$y(t)=\frac{B(q)}{A(q)}u(t)+\frac{C(q)}{A(q)}e(t)$$

3.1.4 Output-error model

If the linear difference equation is error-free, but that the noise consists of white measurement noise only:

$$x(t)+f_1x(t-1)+\cdots+f_{n_f}x(t-n_a)=b_1u(t-1)+b_2u(t-2)+\cdots+b_{n_b}(t-n_b)$$

$$y(t)=x(t)+e(t)$$

$$y(t)=\frac{\left(\sum_{k=1}^{n_b}b_kq^{-k}\right)}{\left(\sum_{k=0}^{n_f}f_kq^{-k}\right)}u(t)+e(t)$$

$$y(t)=\frac{B(q)}{F(q)}u(t)+e(t)$$

3.1.5 Box-Jenkins model (contains above all)

$$y(t)=\frac{B(q)}{F(q)}u(t)+\frac{C(q)}{A(q)}e(t)$$

3.2 2-DoF mass-spring-damper system

Define stiffness and mass matrices

|\/\/\/\/\O\/\/\/\/\O

k1 c1 m1 k2 c2 m2

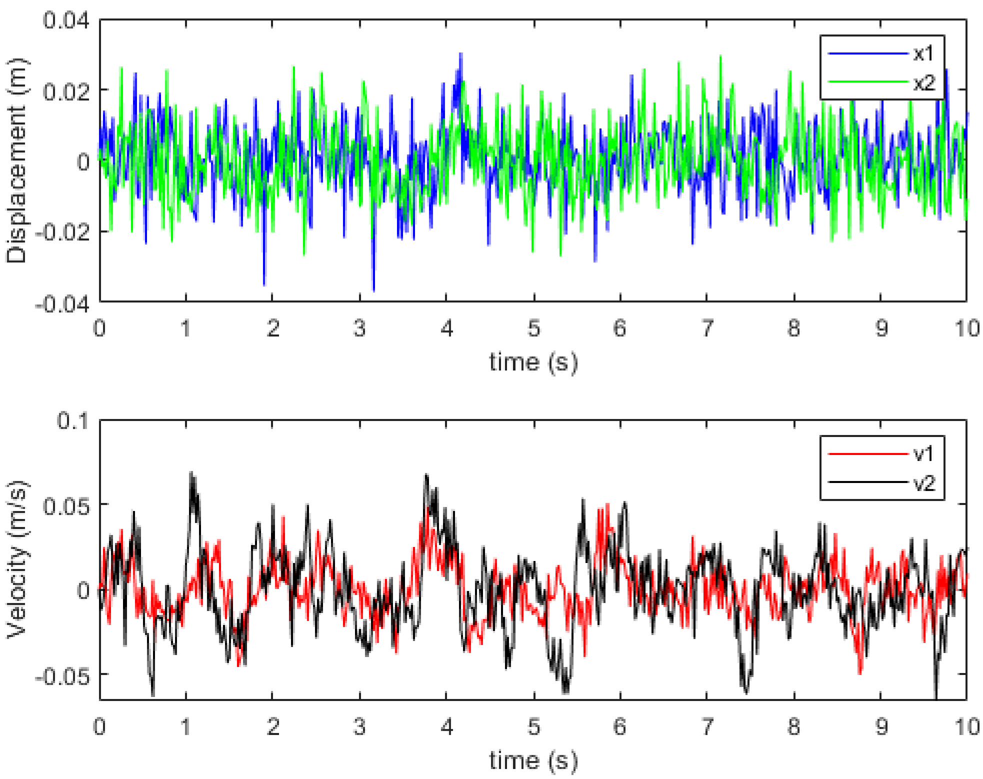
$$M\ddot{X}+C\dot{X}+KX=F$$

```
clc;
clear;
close all;
addpath ../functions/
m1=5; %mass 1 [kg]
m2=2; %mass 2 [kg]
k1=200; %spring 1 [N/m]
k2=100; %spring 2 [N/m]
c1=10; % damping coeff 1
c2=10; % damping coeff 2
M=[m1 0;0 m2]; %mass matrix
K=[k1+k2 -k2; -k2 k2]; %stiffness matrix
Damp=[c1+c2 -c2; -c2 c2]; % damping matrix
```

```
T = 10; %length of time duration
nt = 500; %number of time stamps
Fs = nt/T;
tspan=linspace(0,T,nt);
F = csvread(['../data/ambient.csv']); %load data
```

3.2.1 Load the data generated in script 1

```
ysol = csvread(['../data/2dof_noP.csv']); %load data
tspan=linspace(0,T,nt);
figure(1)
plot_dv(tspan,ysol')
```



3.2.2 FIR model identification

$$[y(n_b), \cdots, y(N)]^T = \begin{bmatrix} u(n_b-1) & u(n_b-2) & \cdots & u(0) \\ u(n_b) & u(n_b-1) & \cdots & u(1) \\ \vdots & \vdots & & \vdots \\ u(N_1) & u(N-2) & \cdots & u(N-n_b) \end{bmatrix} [b_1, b_2, \cdots, b_{n_b}]T$$

3.2.3 ARX model identification

$$[y(\max(n_a,n_b)),\cdots,y(N)]^T=\begin{bmatrix}-y(n_a-1) & \cdots & -y(0) & u(n_a-1) & \cdots & u(n_a-n_b)\\ -y(n_a) & \cdots & -y(1) & u(n_a) & \cdots & u(n_a-n_b+1)\\ \vdots & & \vdots & \vdots & & \vdots\\ -y(N-1) & \cdots & -y(N-n_a) & u(N-1) & \cdots & u(N-n_b)\end{bmatrix}[a_1,a_2,\cdots,a_{n_a},b_1,b_2,\cdots,b_{n_b}]T$$

3.2.4 ARMAX model identification

$$[y(\max(n_a,n_b,n_c)),\cdots,y(N)]^T=\begin{bmatrix}-y(n_a-1) & \cdots & -y(0) & u(n_a-1) & \cdots & u(n_a-n_b) & \epsilon(n_a-1) & \cdots & \epsilon(n_a-n_c)\\ -y(n_a) & \cdots & -y(1) & u(n_a) & \cdots & u(n_a-n_b+1) & \epsilon(n_a) & \cdots & \epsilon(n_a-n_c+1)\\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots\\ -y(N-1) & \cdots & -y(N-n_a) & u(N-1) & \cdots & u(N-n_b) & \epsilon(N-1) & \cdots & \epsilon(N-n_c)\end{bmatrix}$$

Need iterative solution

3.2.5 Output error identification

Similar to ARMAX, but error does not shift.

3.3 Prediction error method

All above can be generalizaed by considering the equation and out errors as prediction errors.

3.3.1 Deterministic model with one step information

$$x[k+1]=A\,x[k]+B\,u[k]$$
$$y[k+1]=C\,x[k+1]$$
$$y[k+1|k]=C\,A\,C^{-1}y[k]+C\,B\,u[k]$$
$$\theta^*=\underset{\theta}{\operatorname{argmin}}\,\frac{1}{N}\sum_{k=0}^{N-1}\left(y[k+1]-\hat{y}[k+1|k]\right)^2$$

```
rng(49)
params_ini = [1,1,100,100,5,5];
options = optimset('TolFun',10e-15);
params = zeros(6,6);
filenames = {'../data/2dof.csv','../data/2dof_noP.csv','../data/2dof_P6.csv','../data/2dof_P5.
for i=1:6
    ysol = csvread(filenames{i});
    pe = @(pa)pe_func(pa,tspan,F,ysol);
    params(i,:) = fmincon(pe,params_ini,[],[],[],[],[0,0,0,0,0,0],...
                            [10,10,300,300,20,20],[],options);
end
```

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>
```

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>
```

True parameters:

```
disp([m1,m2,k1,k2,c1,c2])
```

```
     5     2   200   100    10    10
```

Estimated parameters

```
disp(params)
```

```
    4.9997    2.0000  199.9864   99.9995   10.0017   10.0001
    2.9936    2.0927    6.1589    0.0001   19.9999   19.9993
    5.8706    2.3510    0.0000    0.0000    6.2452   19.0026
    5.7234    1.9322    0.0013   17.8977   19.9992    2.7900
    6.9048    2.4573  188.7143   96.9373   19.4295    8.9579
```

```
     3.1003    1.7399  123.2224   77.2227    6.5086    7.8601
```

3.3.2 Deterministic model without previous information

$$x[k+1] = A\,x[k] + B\,u[k]$$
$$y[k+1] = C\,x[k+1]$$
$$\hat{y}[k+1] = C\,A^k\,x[0] + C\,B\,u[k]$$
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \frac{1}{N}\sum_{k=1}^{N}(y[k] - \hat{y}[k])^2$$

```
x0=[0;0;0;0]; %initialization of states
C = [1,0,0,0;
     0,1,0,0;
     0,0,1,0;
     0,0,0,1]; %observation matrix
params1 = zeros(6,6);
for i=1:6
    ysol = csvread(filenames{i});
    pe = @(pa)pe1_func(pa,tspan,x0,C,F,ysol);
    params1(i,:) = fmincon(pe,params_ini,[],[],[],[],[0,0,0,0,0,0],...
                           [10,10,300,300,20,20],[],options);
end
```

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>
```

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>
```

```
disp(params1)
```

```
    5.0000    2.0000  199.9994  100.0005   10.0003    9.9999
    5.5827    1.9749  219.3015   98.8743    9.5965    8.5350
    3.3593    2.0498  188.5633   78.1357    5.6857    7.0251
    0.6686    2.2558   96.5856  299.9998    0.0000   11.8992
    1.4199    0.7358   95.1074    0.0016    0.0001   19.9982
    0.4877    0.3804   38.4027    0.0073    0.0001   19.9746
```