01211433 Vision and Control of Industrial Robots

Lecture 8

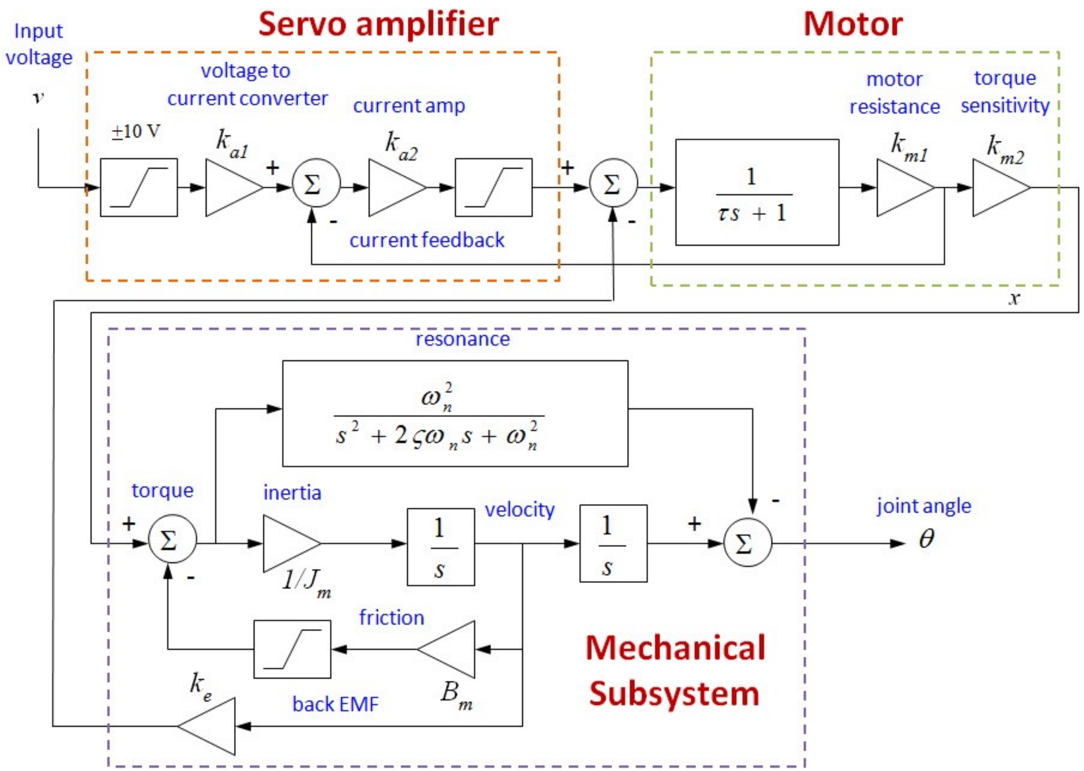
Dr.Varodom Toochinda

Dept. of Mechanical Engineering, Kasetsart University

System identification of robot-joint model

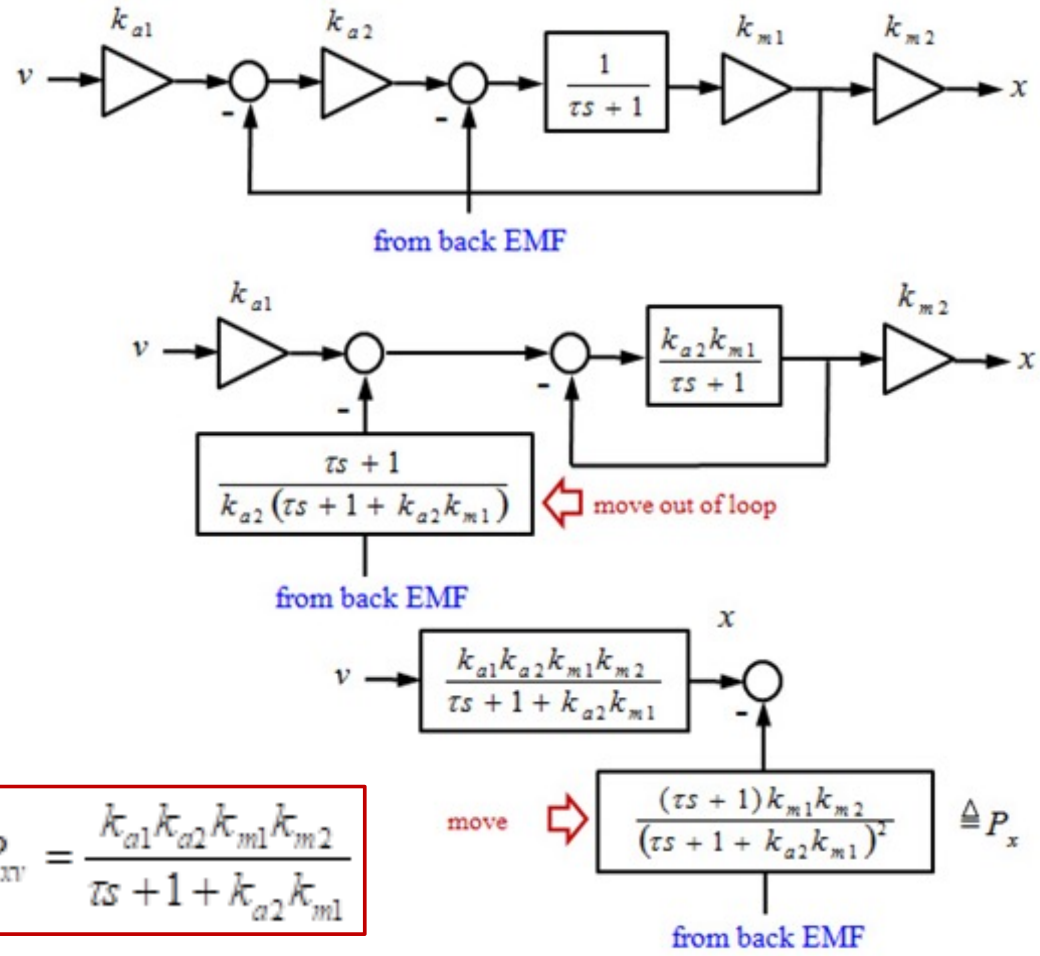
- Example of complicated robot model
- Least-square identification

Example of complicated robot joint model



$$P_{\theta v} = P_{\theta x} P_{xv}$$

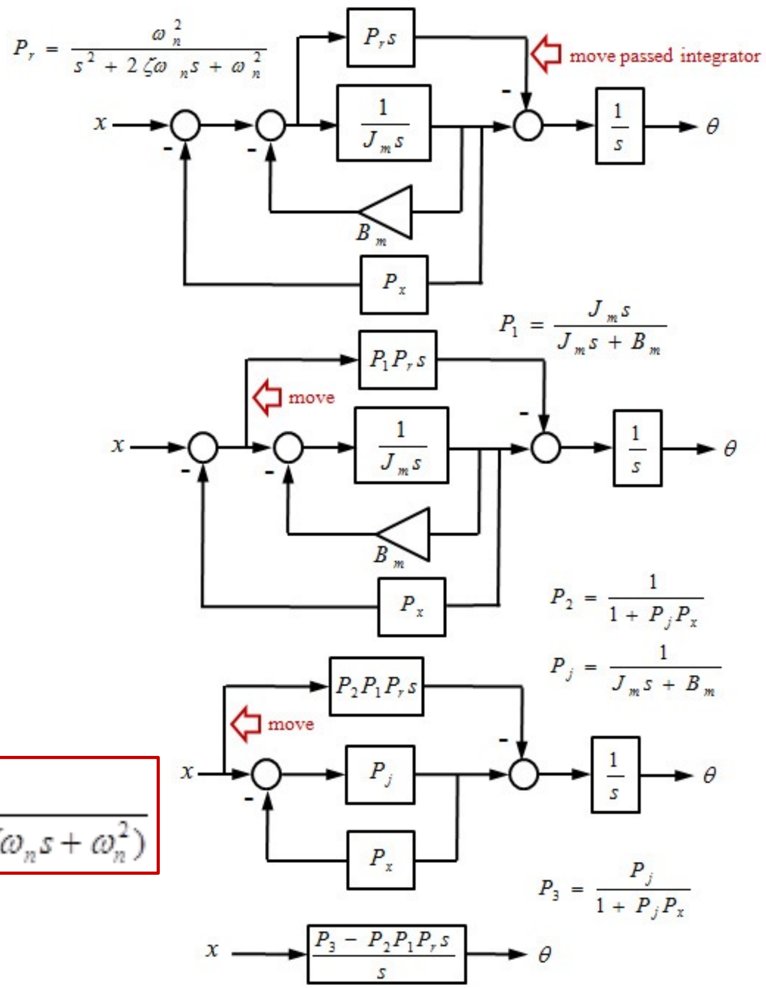
Block diagram reduction for P_{xv}



$$\boxed{P_{xv}=\frac{k_{a1}k_{a2}k_{m1}k_{m2}}{\tau s+1+k_{a2}k_{m1}}}$$

Block diagram for $P_{\theta x}$

$$P_{\theta x} = \frac{(\tau s+1+k_{a2}k_{m1})^2((1-J_m\omega_n^2)s^2+2\zeta\omega_n s+\omega_n^2)}{s((J_m s+B_m)(\tau s+1+k_{a2}k_{m1})^2+(\tau s+1)k_{m1}k_{m2}k_e)(s^2+2\zeta\omega_n s+\omega_n^2)}$$



Combined to get $P_{\theta x}$

$$P_{\theta x} = \frac{k_{a1}k_{a2}k_{m1}k_{m2}(\tau s+1+k_{a2}k_{m1})((1-J_m\omega_n^2)s^2+2\zeta\omega_n s+\omega_n^2)}{s((J_m s+B_m)(\tau s+1+k_{a2}k_{m1})^2+(\tau s+1)k_{m1}k_{m2}k_e)(s^2+2\zeta\omega_n s+\omega_n^2)}$$

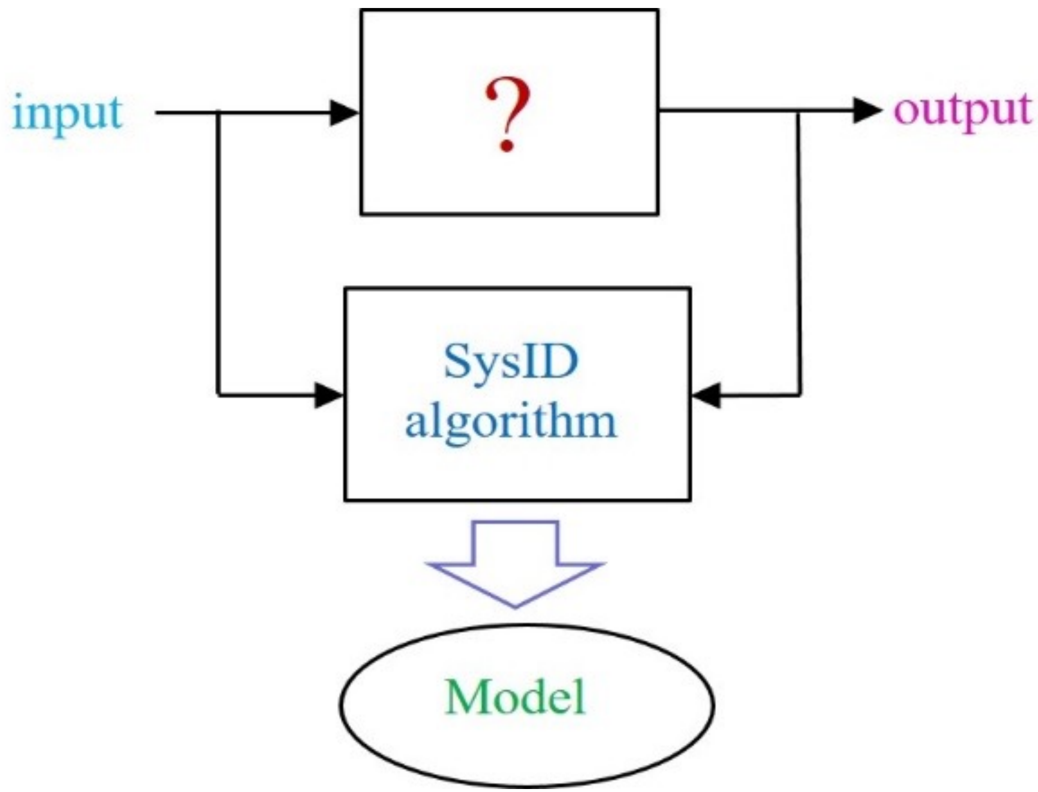
Scilab code

```
ka1 = 0.5; ka2 = 1000; km1 = 0.1471; km2 = 0.26; ke = 0.1996;
jm = 0.000156; bm = 0.001;  kr = 1; tau = 0.0044;
wn=2*%pi*30; z = 0.7;
s=poly(0,'s');
pqvnum = ka1*ka2*km1*km2*(tau*s+1+ka2*km1)*((1-jm*wn^2)*s^2+2*z*wn*s+wn^2);
pqvden = s*((jm*s+bm)*(tau*s+1+ka2*km1)^2+ (tau*s+1)*km1*km2*ke)*(s^2+2*z*wn*s+wn^2);
pqv = syslin('c',pqvnum/pqvden)
```

```
pqv  =                                                     2            3
                 1.006D+08 + 750367.47s - 12843.448s - 0.3822341s
   ----------------------------------------------------------------
                             2            3            4            5            6
   779585.03s + 127410.61s + 932.46886s + 3.4767451s + 0.0002041s + 3.020D-09s
```

Exercise: Find zeros and poles of pqv

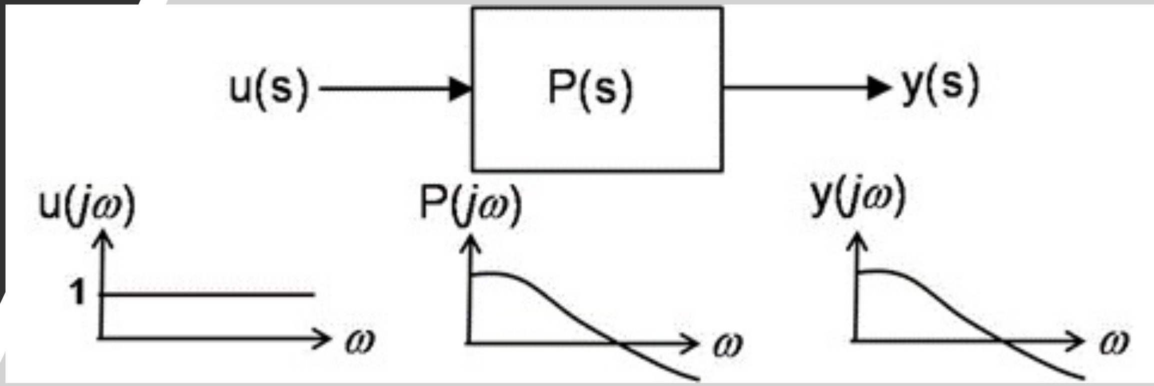
System identification (SysID)



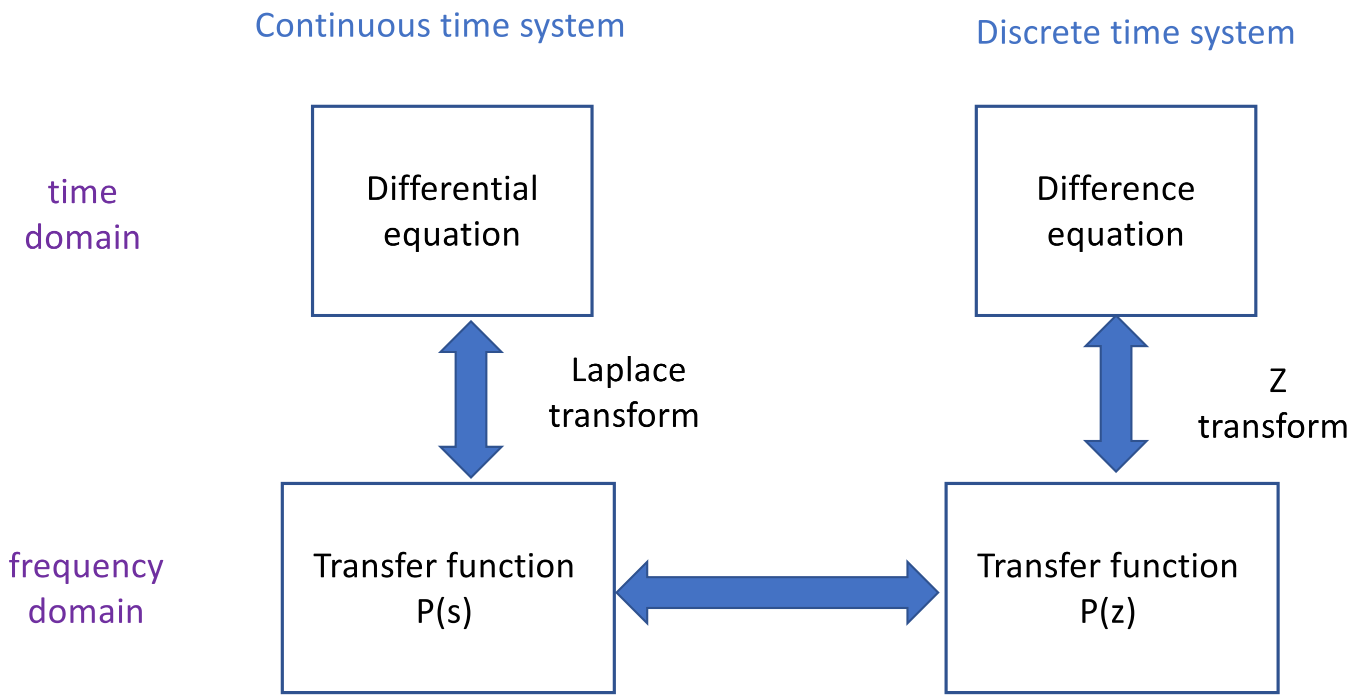
Input choices

- Impulse
- White noise
- Chirp
- Pseudo Random Binary Sequence (PRBS)

$$y(s) = P(s)u(s)$$



System relationships revisited



Parametric SysID (Least-Square estimation)

U(z)

$$G(z) = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \ldots + b_{n-1} z + b_n}{z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n}$$

y(z)

$$\theta = [a_1 \ldots a_n b_1 \ldots b_n]^T$$

Parameter vector to estimate

LS estimation

Form matrix X using input and output samples

$$X = \begin{bmatrix} -y[n-1] & \dots & -y[0] & u[n-1] & \dots & u[0] \\ -y[n] & \dots & -y[1] & u[n] & \dots & u[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y[N_p] & \dots & -y[N_p-n+1] & u[N_p] & \dots & u[N_p-n+1] \end{bmatrix}$$

$$\boxed{\theta_{LS} = (X^T X)^{-1} X^T Y}$$

$X^T X$ is nonsingular when input signal is "persistently exciting."

Robot joint driven by harmonic drive (strain wave gearing)

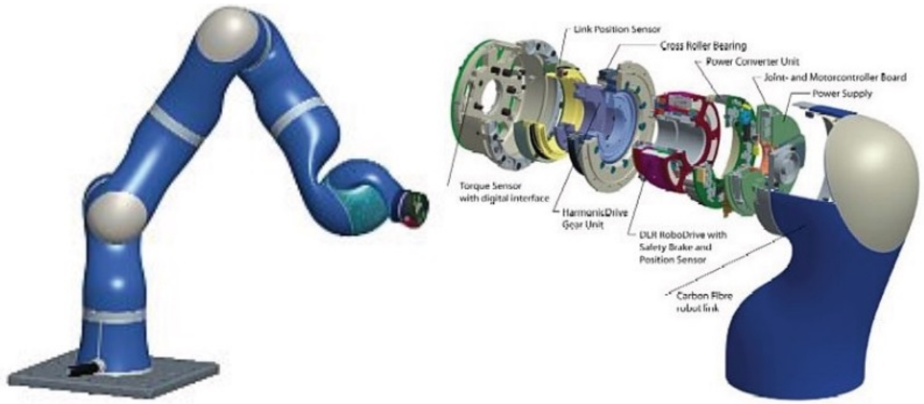
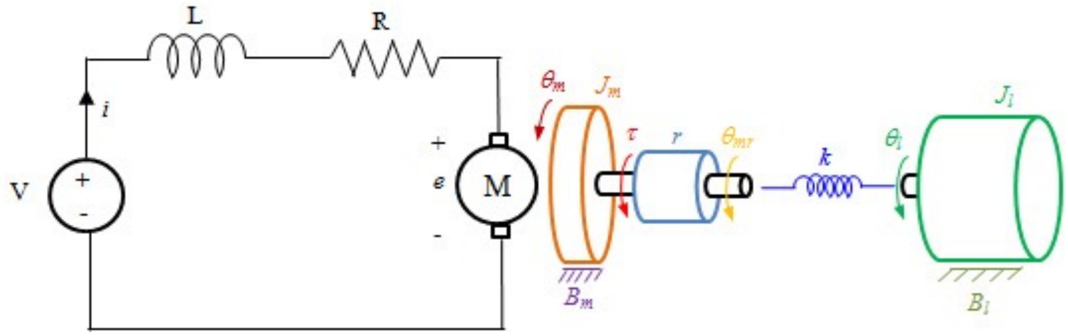


Figure 1

$$\boxed{\frac{\theta_l(s)}{V(s)} = \frac{k_m k}{p_m(s)p_l(s)(Ls+R) - k^2(Ls+R) + rk_m k_b s p_l(s)}}$$



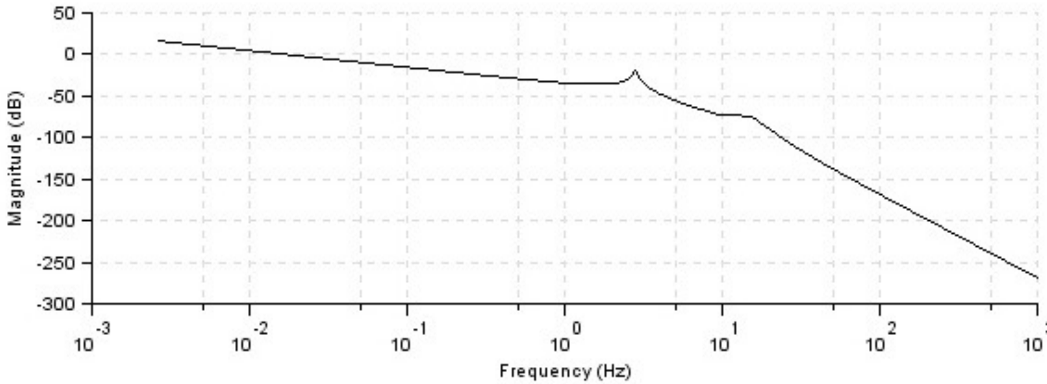
hmdinit.sce

```
-->exec('hdminit.sce',-1);
 P =
                         100000
    --------------------------------------------------
                     2             3         4      5
    1000060s + 4056.0005s + 3400.0801s + 3.008s + 0.3s

 Open-loop poles

    0
  - 4.5241679 + 104.88968i
  - 4.5241679 - 104.88968i
  - 0.4891654 + 17.383778i
  - 0.4891654 - 17.383778i
```

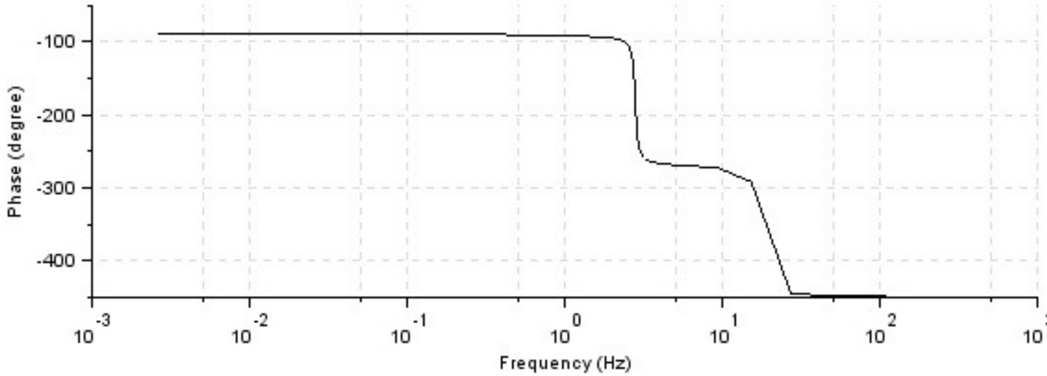
$$P(s) = \frac{100000}{0.3s^5 + 3.008s^4 + 3400.08s^3 + 4056s^2 + 1000060s}$$

```
km = 100;  // torque constant
kb = 1;  // back EMF constant
k = 1000;  // torsional stiffness of harmonic
drive
r = 10;   // gear ratio
L = 0.1  // armature inductance
R = 1;  // armature resistance
Jm = 1;  //  motor inertia
Bm = 0.01;  // motor shaft friction
Jl = 3;  // load inertia
Bl = 0.05;  // load friction
s=poly(0,'s');
pnum = km*k;
pl = Jl*s^2+Bl*s+k;
pm = Jm*s^2+Bm*s+k;
pden = (L*s+R)*pm*pl - k^2*(L*s+R) +
r*km*kb*s*pl;
P = syslin('c',pnum,pden);
disp("P = ")
disp(P)
[A,B,C,D]=abcd(P);
disp("Open-loop poles ");
disp(spec(A))
```

Bode plot

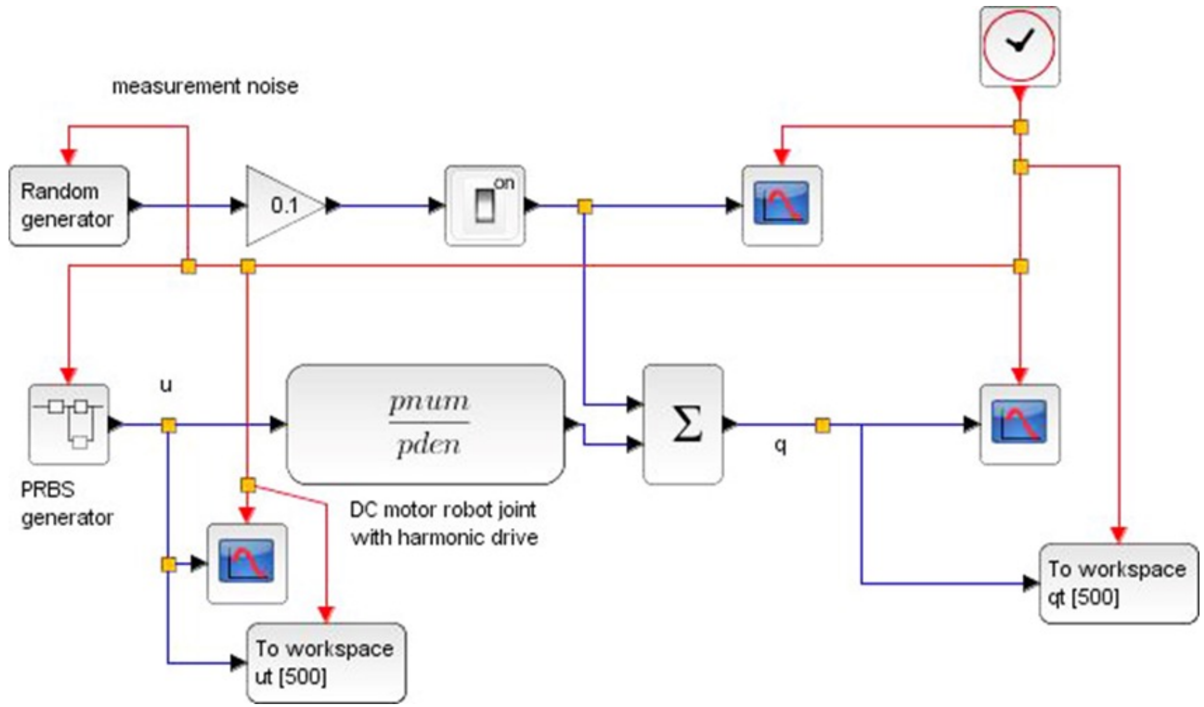


50
0
-50
-100
-150
-200
-250
-300
Magnitude (dB)
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
Frequency (Hz)

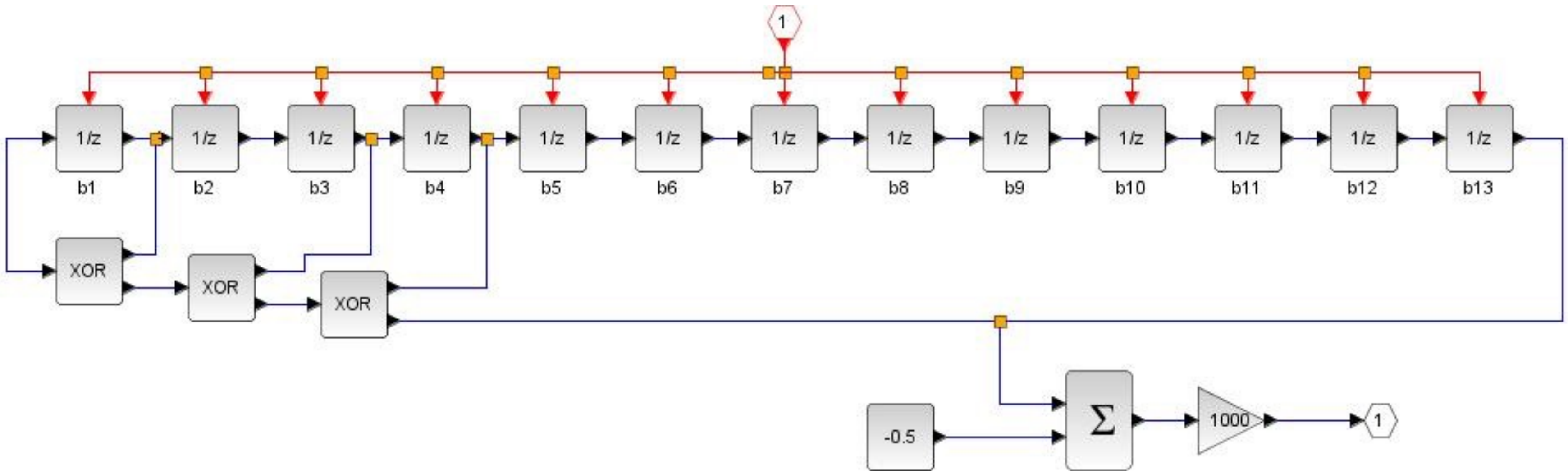


-100
-200
-300
-400
Phase (degree)
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
Frequency (Hz)

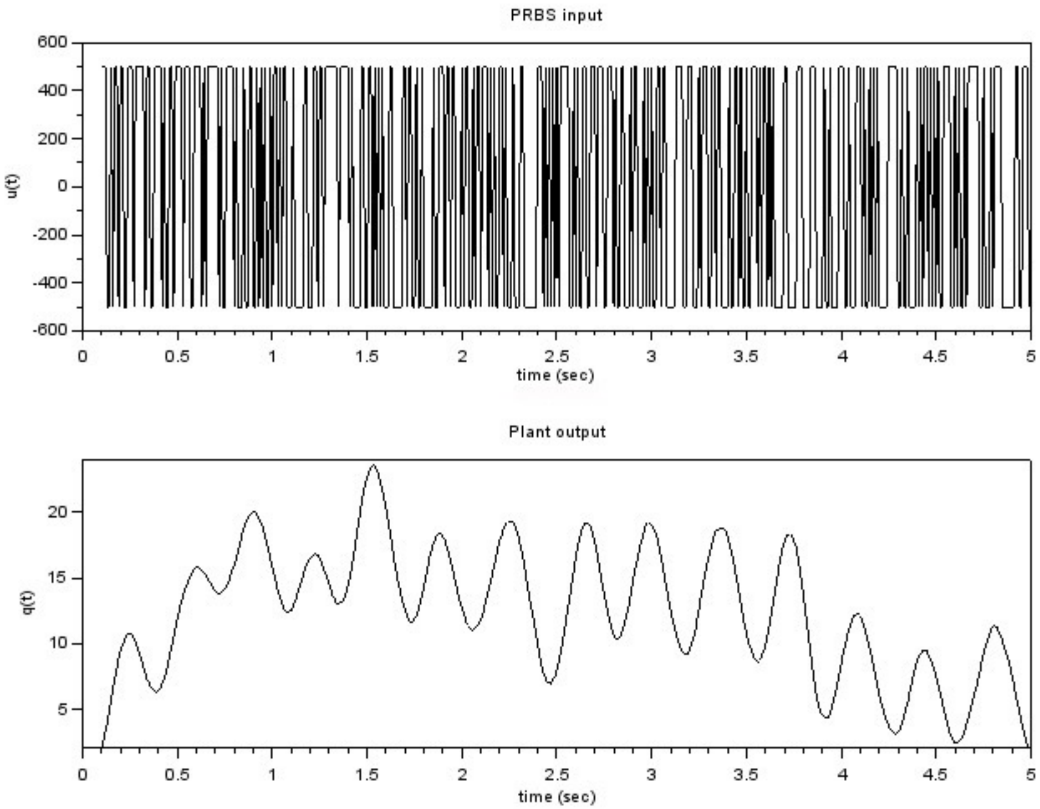
hdm_sysid.zcos



PRBS generator



Input-output data



LS estimated model and parameters

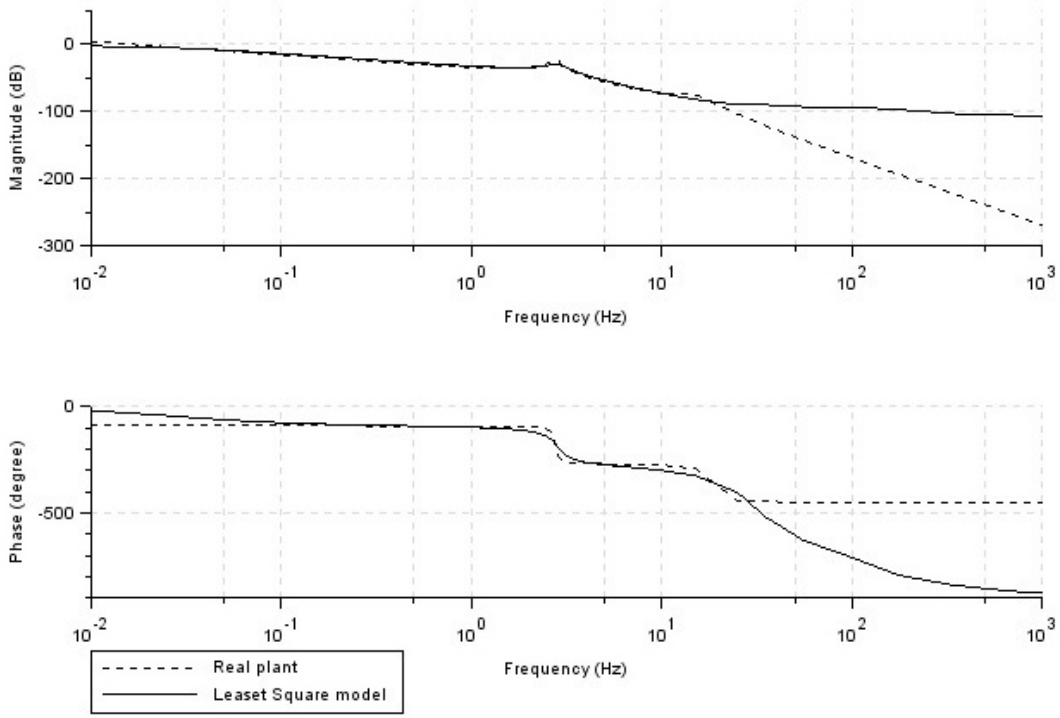
$$P(z) = \frac{b_1 z^6 + b_2 z^5 + b_3 z^4 + b_4 z^3 + b_5 z^2 + b_6 z + b_7}{z^7 + a_1 z^6 + a_2 z^5 + a_3 z^4 + a_4 z^3 + a_5 z^2 + a_6 z + a_7}$$

$$\theta = [a_1, a_2, a_3, a_4, a_5, a_6, a_7, b_1, b_2, b_3, b_4, b_5, b_6, b_7]^T$$

LS estimation of HDM model (Scilab)

- Run hdminit.sce to initialize variables for real plant
- Simulate with hdm_sisid.zcos to acquire input and output data
- Run hdm_sysid.sce to do LS estimation and convert to continuous-time transfer function to compare the Bode plot

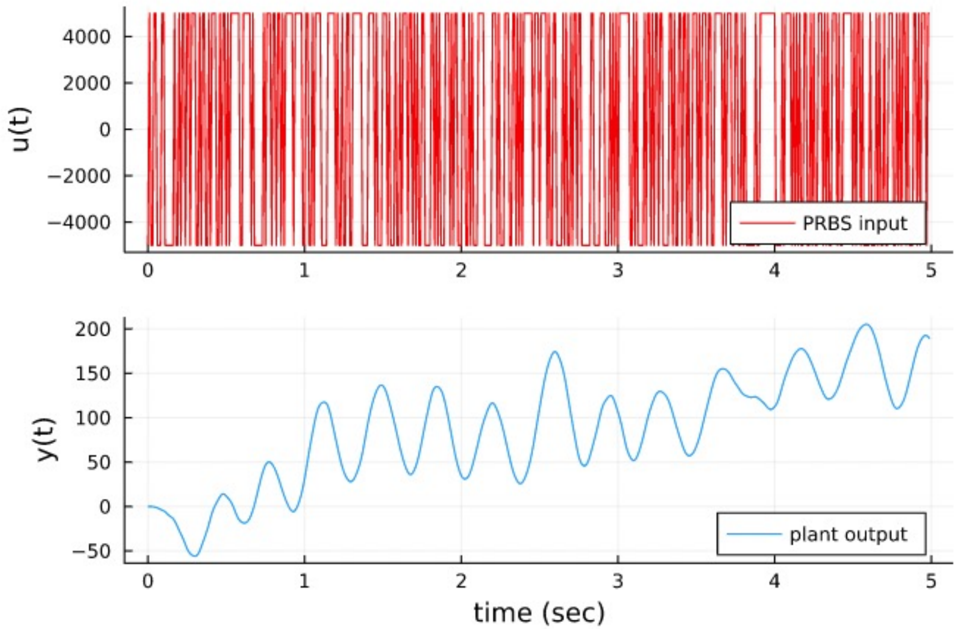
Bode-plot comparison



LS Estimation of HDM model (Julia)

- See Pluto notebook lsid.jl

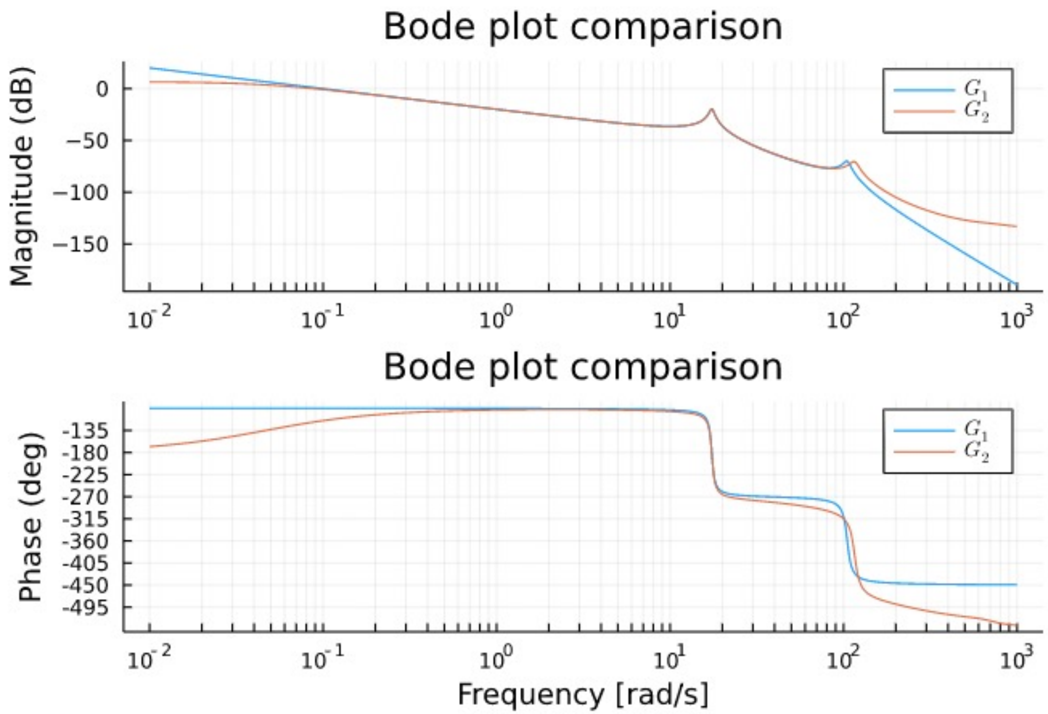
Input-output data




```
begin
    #noise_level = 0.01  # increase this value to add more random noise
    t_sim = collect(0:0.01:4.99)
    num_samples = size(t_sim,1)
    u_data = PRBS_generator(5000,num_samples) # generate PRBSwith amplitude 2000
    y_data, t_data, x_data = lsim(P, u_data, t_sim, method=:zoh)
    ym_data = y_data .+ noise_level*rand(num_samples) # add noise to output
    u_plot = plot(t_data,u_data,label="PRBS
input",ylabel="u(t)",color=:red,legend=:bottomright)
    y_plot = plot(t_data, ym_data, label="plant output",xlabel="time
(sec)",ylabel="y(t)",legend=:bottomright)
        plot(u_plot,y_plot, layout=(2,1))


end
```

Bode-plot comparision

Noise level = 0.01




```
begin
    setPlotScale("dB")
    ω₁ = exp10.(LinRange(-2,3,2000))
    bodeplot([P, P_id],ω₁,title="Bode plot comparison")
end
```

Figure 9 Bode plot comparison between real plant and LSID model

Reference

- L Ljung. System Identification: Theory for the User. 2nd ed. Prentice-Hall, 1999.