Table of Contents

close all

```
clear all
```

parameters setting (Choose one of the 3 cases)

```
set_common_params

% set_position_position_controller
% set_position_force_controller
% set_force_position_controller
set_force_force_controller
```

Controller

```
master_controller = @(x_m, xd_m, xdd_m, f_m, x_s, xd_s, xdd_s, f_s) ((K_mpm * x_m
        (K_mps * x_s + K_mps_d * xd_s + K_mps_dd * xdd_s + K_mfs * f_s));
slave_controller = @(x_m, xd_m, xdd_m, f_m, x_s, xd_s, xdd_s, f_s) ((K_spm * x_m +
        (K_sps * x_s + K_sps_d * xd_s + K_sps_dd * xdd_s + K_sfs * f_s));
```

initial condition

```
x_m = 0;
xd_m = 0;
xdd_m = 0;
x_s = 0;
xd_s = 0;
xdd_s = 0;

tau_op = 0;
f_m = 0;
tau_m = 0;
f_s = 0;
tau_s = 0;
```

operator input function

```
input_force = @(t) ( 5-5*cos(4*pi*t));
% input_force = @(t) ( 5-5*cos(1*pi*t));
% input_force = @(t) (1);
```

init simulation

```
dt = 0.001;
sim_time = 10;
t = linspace(0, sim_time, sim_time/dt);

x_m_log = zeros(size(t));
x_s_log = zeros(size(t));
f_m_log = zeros(size(t));
f_s_log = zeros(size(t));
```

simulation start

```
for i = 1:length(t)
    % operator input force
    tau_op = input_force(t(i));

% -----------------master dynamics, operator impedance-----------------
    % operator impedance => master dynamics works at every cases
    % master dynamics => operator impedance works except 1st case.

    % operator impedance model
    f_m = tau_op - (m_op*xdd_m + b_op*xd_m + c_op*x_m);

    % master dynamics
    xdd_m = (tau_m + f_m - b_m*xd_m) / m_m;
    xd_m = xd_m + xdd_m * dt;
    x_m = x_m + xd_m * dt;
%-------------------------------------------------------------------------

    % slave dynamics
    xdd_s = (tau_s - f_s - b_s * xd_s) / m_s;
    xd_s = xd_s + xdd_s * dt;
    x_s = x_s + xd_s * dt;

    % object impedance model
    f_s = m_w * xdd_s + b_w * xd_s + c_w * x_s;

    % master controller
    tau_m = master_controller(x_m, xd_m, xdd_m, f_m, x_s, xd_s, xdd_s, f_s);
    % slave controller
    tau_s = slave_controller(x_m, xd_m, xdd_m, f_m, x_s, xd_s, xdd_s, f_s);

    % logging
    x_m_log(i) = x_m;
    x_s_log(i) = x_s;
```

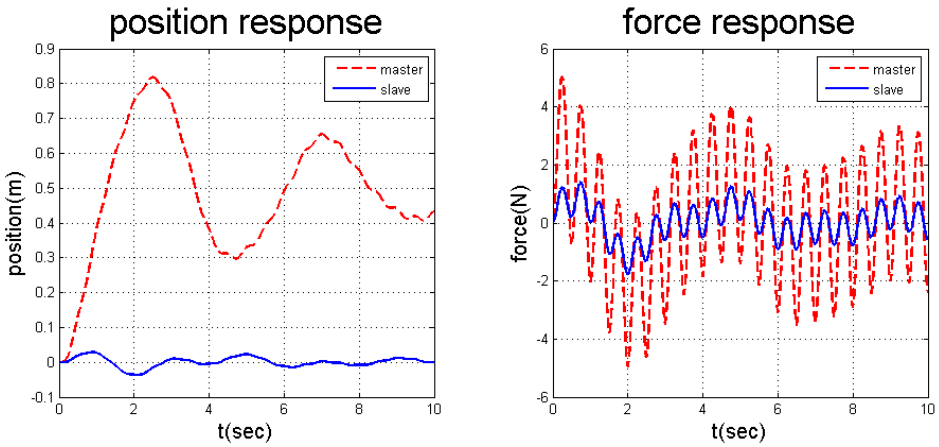
```
    f_m_log(i) = f_m;
    f_s_log(i) = f_s;
end
```

plotting

```
figure(1);
subplot(1,2,1);
plot(t, x_m_log, 'r--','linewidth',2);
hold on; grid on;
plot(t, x_s_log, 'b','linewidth',2);
hold off;
xlabel('t(sec)','fontsize',15); ylabel('position(m)','fontsize',15)
legend('master', 'slave');
title('position response','fontsize',25);


subplot(1,2,2);
plot(t, f_m_log, 'r--','linewidth',2);
hold on; grid on;
plot(t, f_s_log, 'b','linewidth',2);
hold off;
xlabel('t(sec)','fontsize',15); ylabel('force(N)','fontsize',15)
legend('master', 'slave');
title('force response','fontsize',25);

% autoArrangeFigures(1,2)
```



Published with MATLAB® R2013b