Quick manual for MATLAB tool

MATLAB
SIMULINK

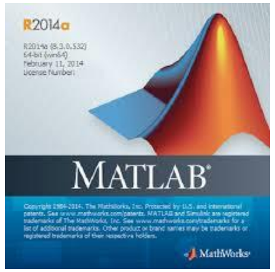
Reference files:

https://granite.phys.s.u-tokyo.ac.jp/svn/LCGT/trunk/VIS/SuspensionControlModel/
http://gwdoc.icrr.u-tokyo.ac.jp/cgi-bin/private/DocDB/ShowDocument?docid=6284

→ *Experience is the best teacher.*
You can run demo files from examples.zip.

Preparation

1. Install MATLAB (2014)

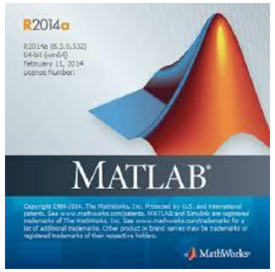


Procedure

2-1. make a virtual experimental system (.slx file).
2-2. make servo filters (.m file).
2-3. run measurement codes.

Preparation

1. Install MATLAB (2014)

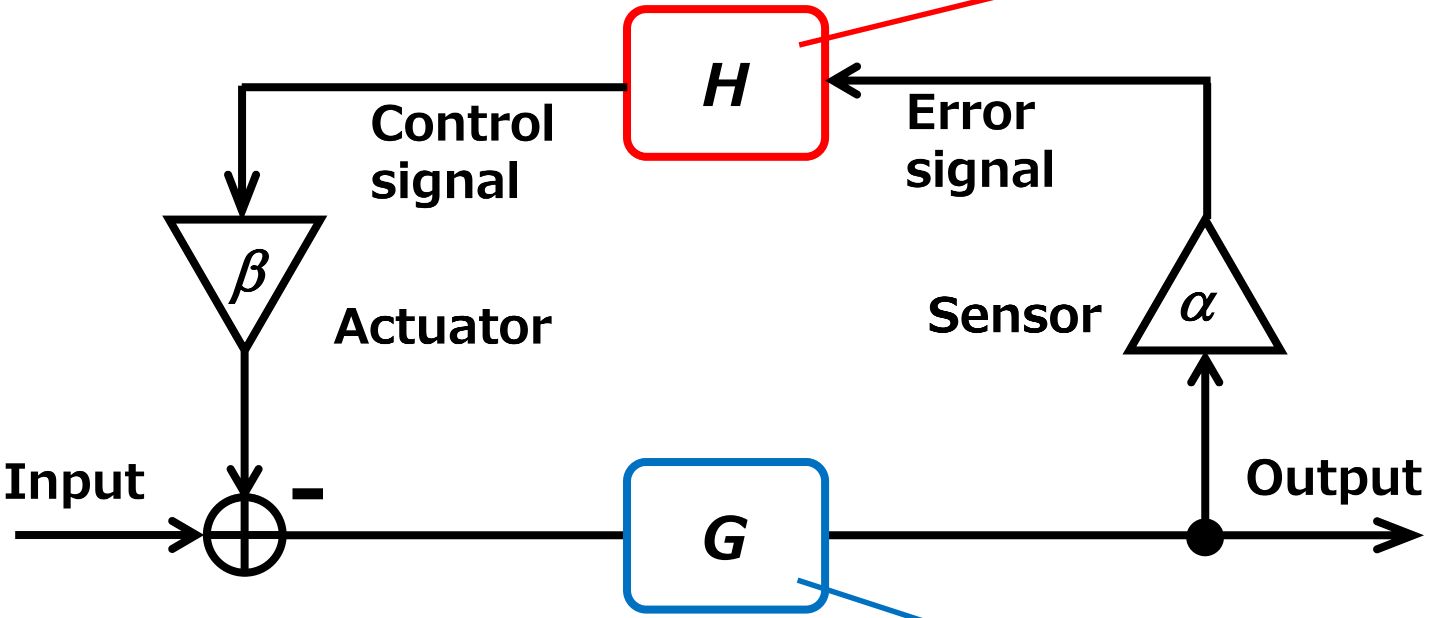


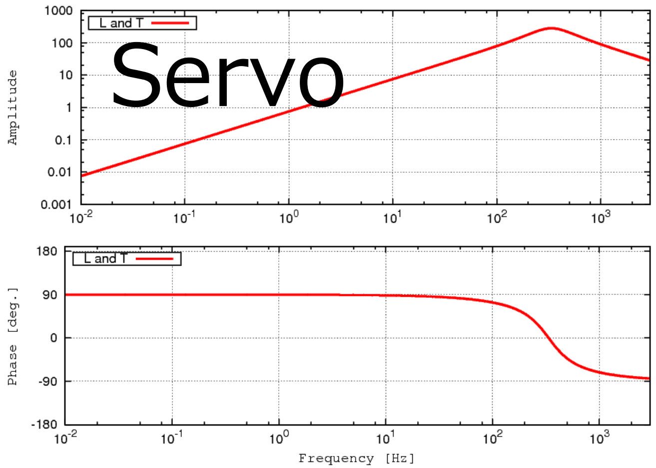
Procedure

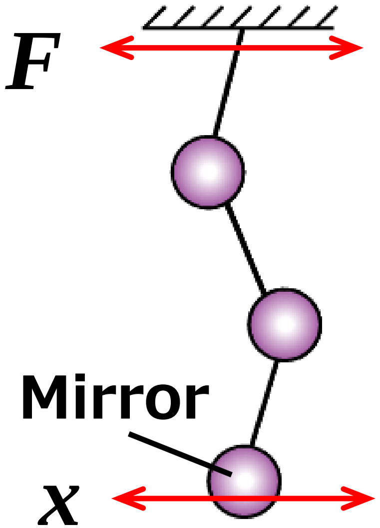
2-1. make a virtual experimental system (.slx file).
2-2. make servo filters (.m file).
2-3. run measurement codes.

2-1. make experimental system

In principle,





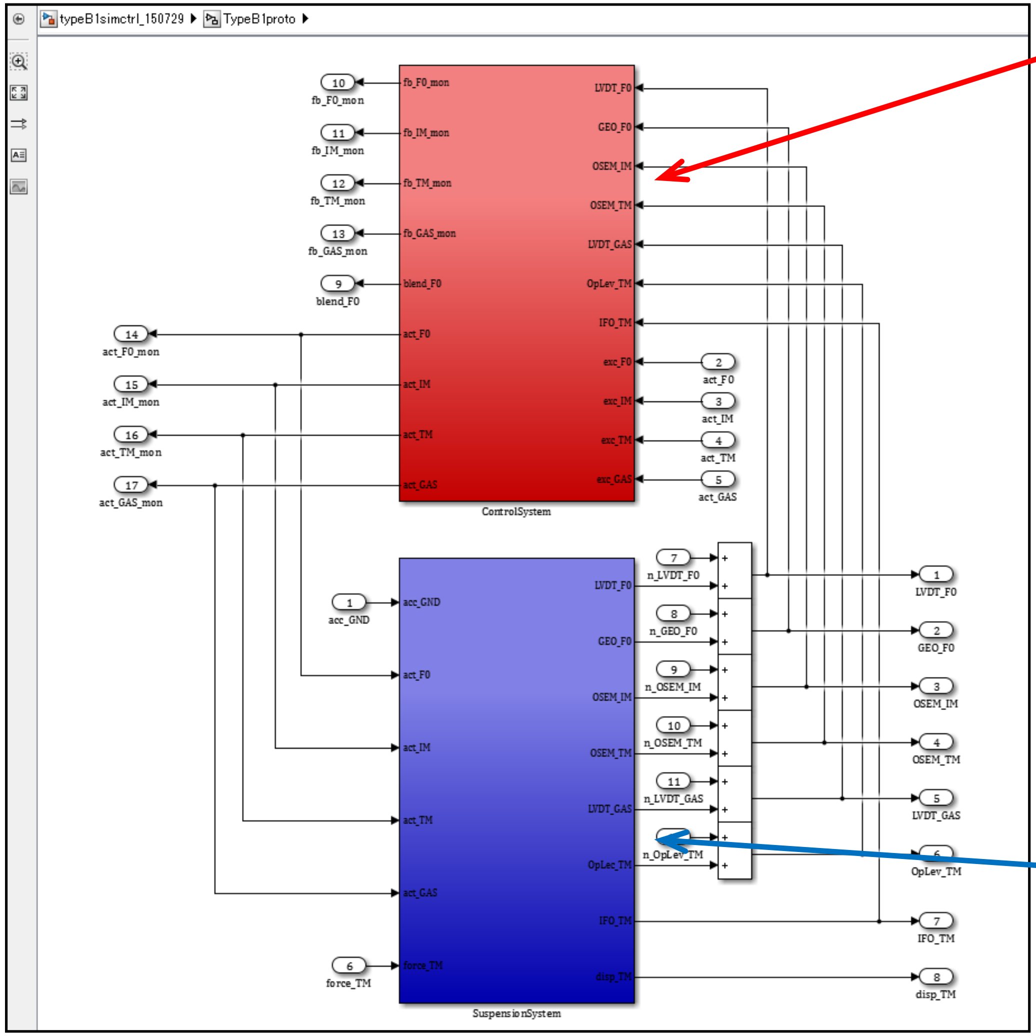


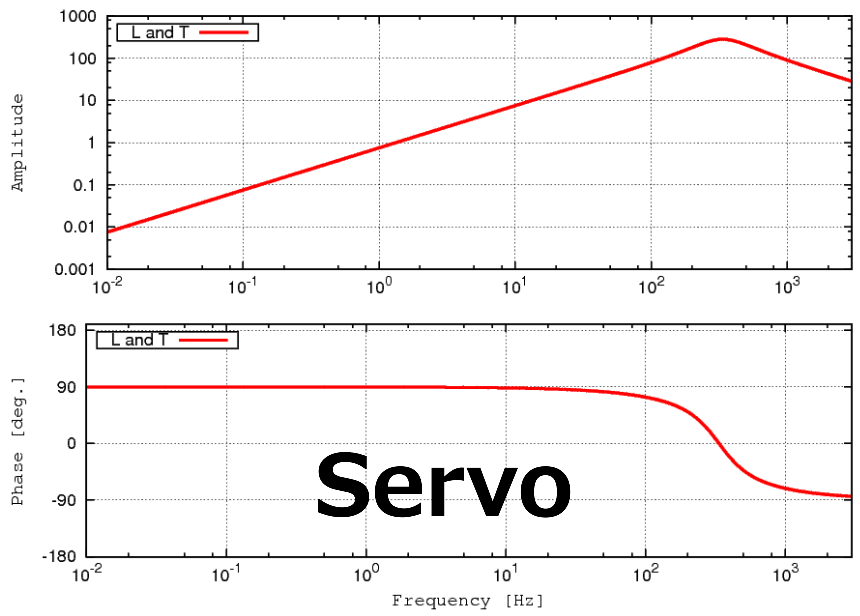
Suspension mechanical model

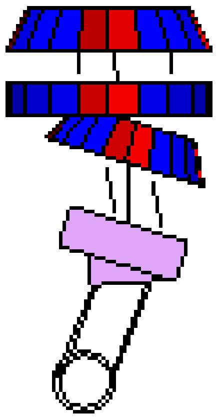
First, one has to construct a virtual experimental system.

2-1. make experimental system

In our calculation,



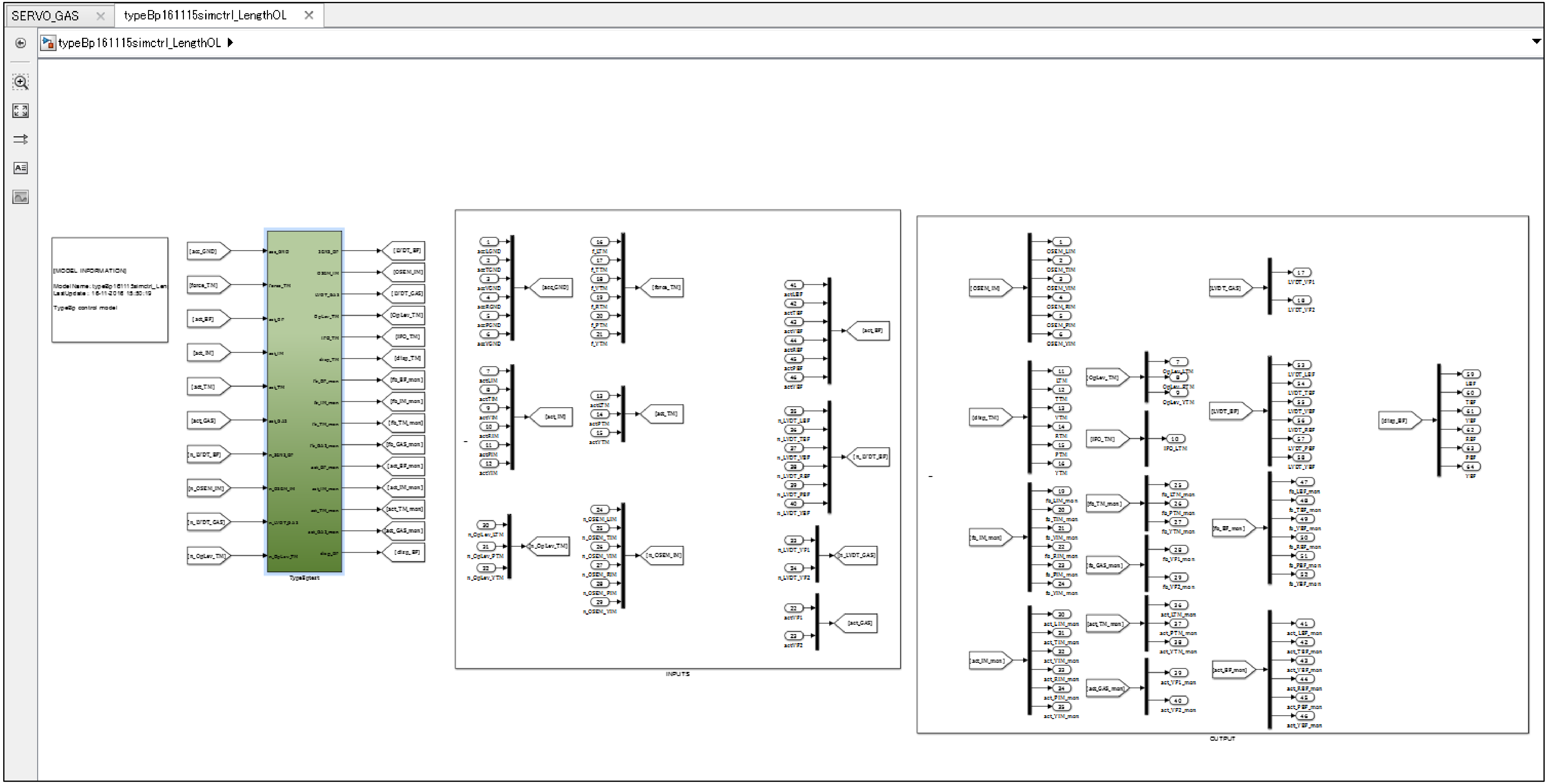




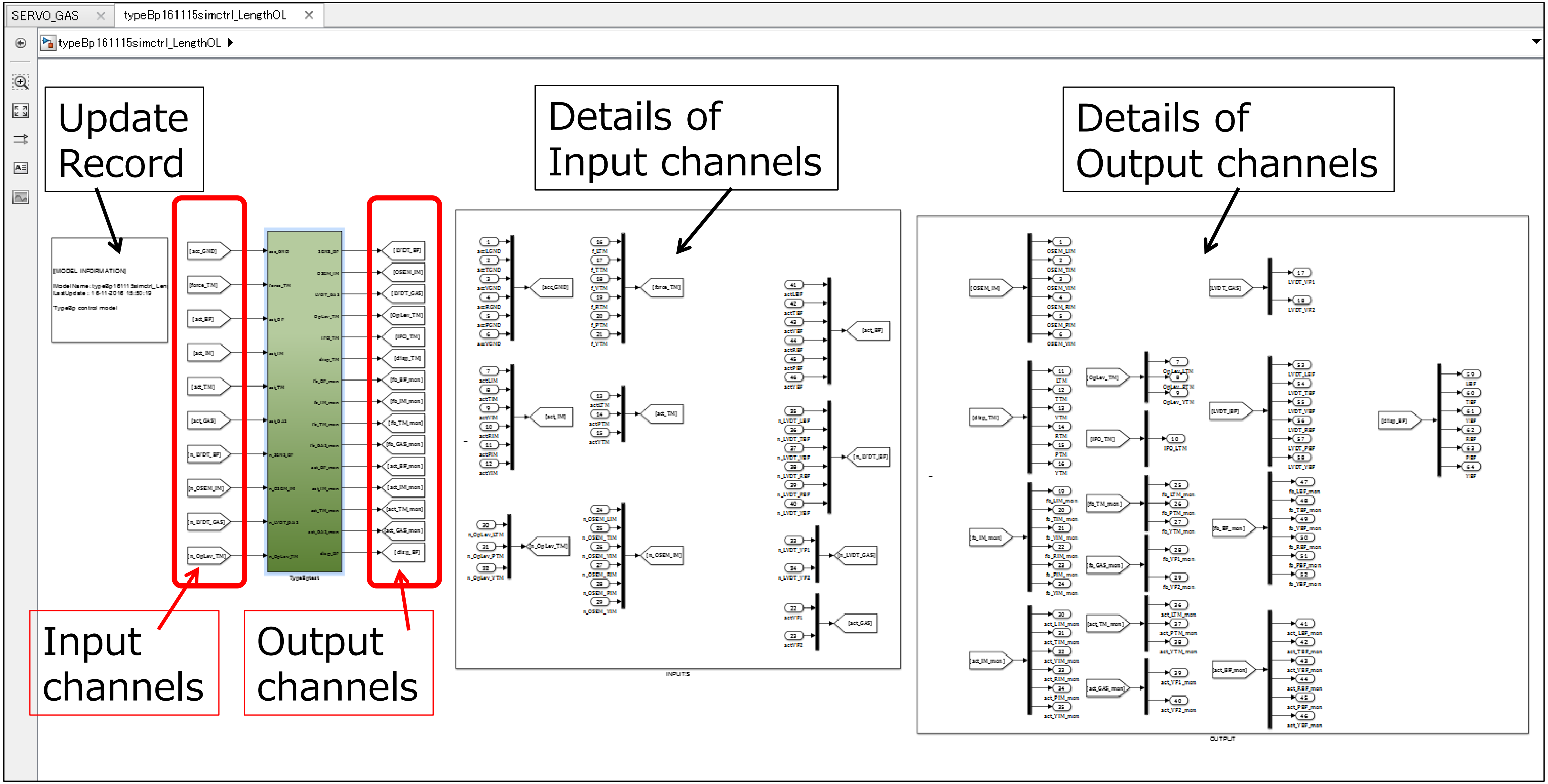
Suspension mechanical model

experimental system → .slx file

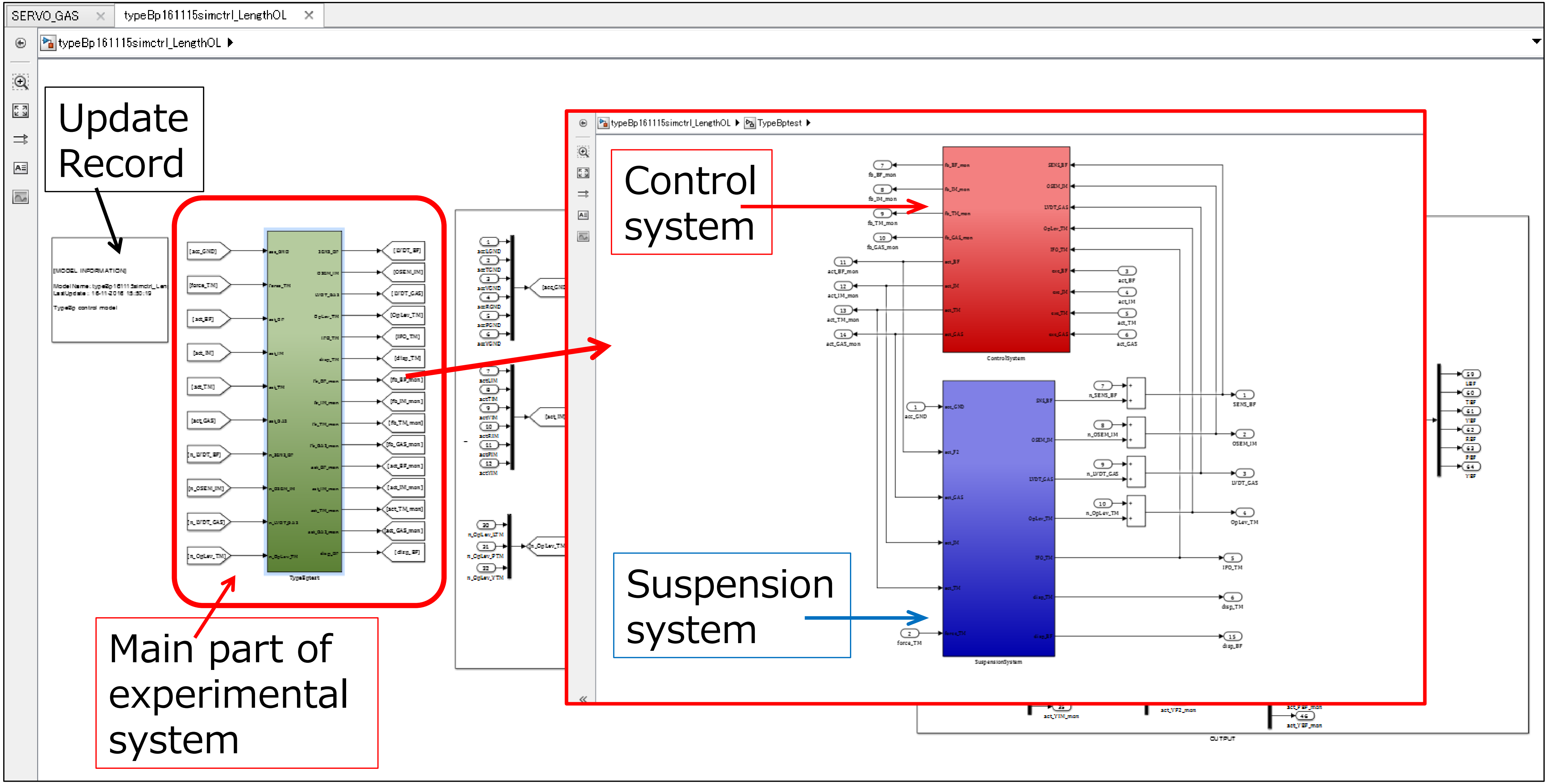
2-1-1. experimental-system-file structure (what's .slx?) 1



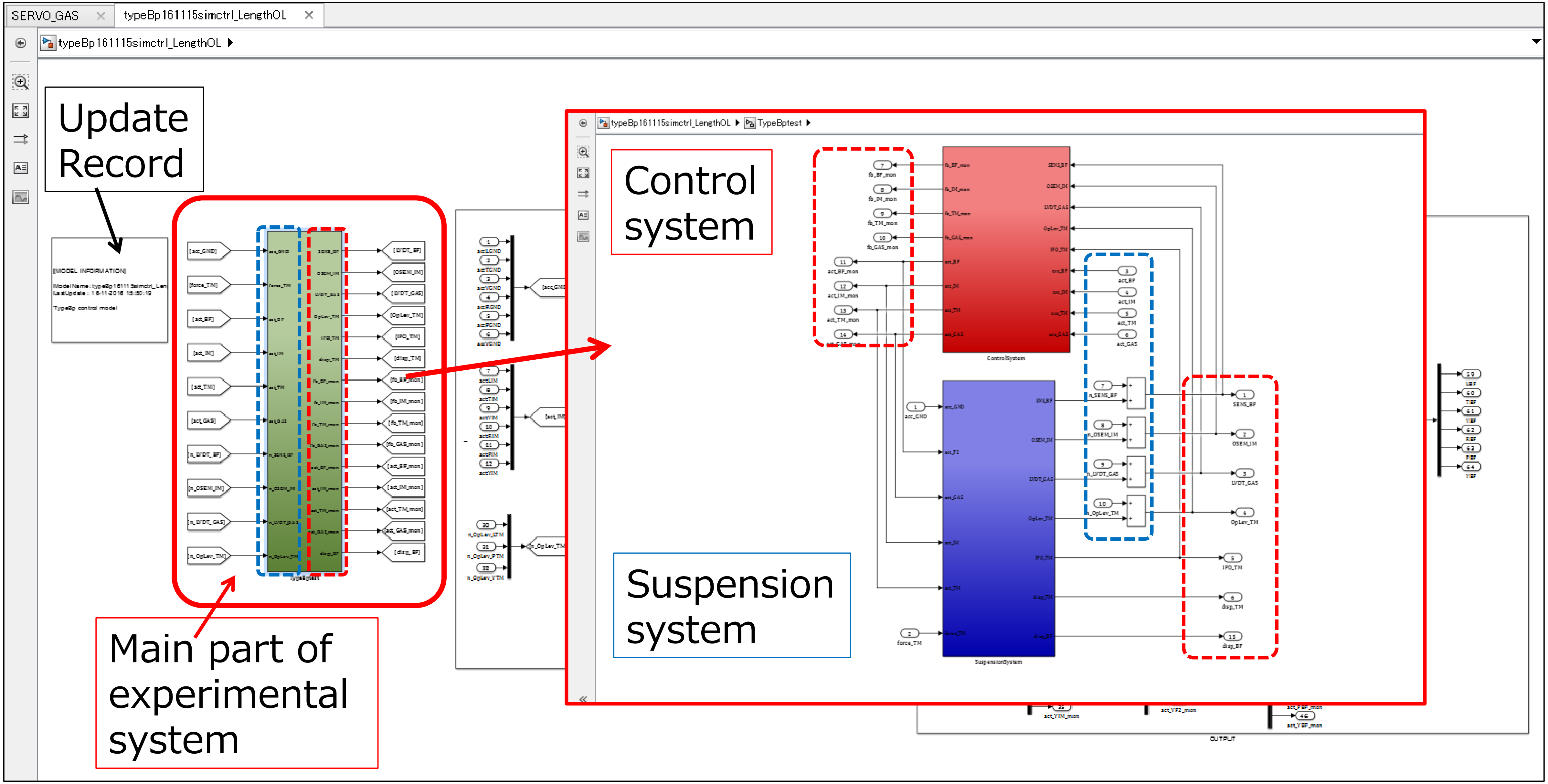
2-1-1. experimental-system-file structure (what's .slx?) 2



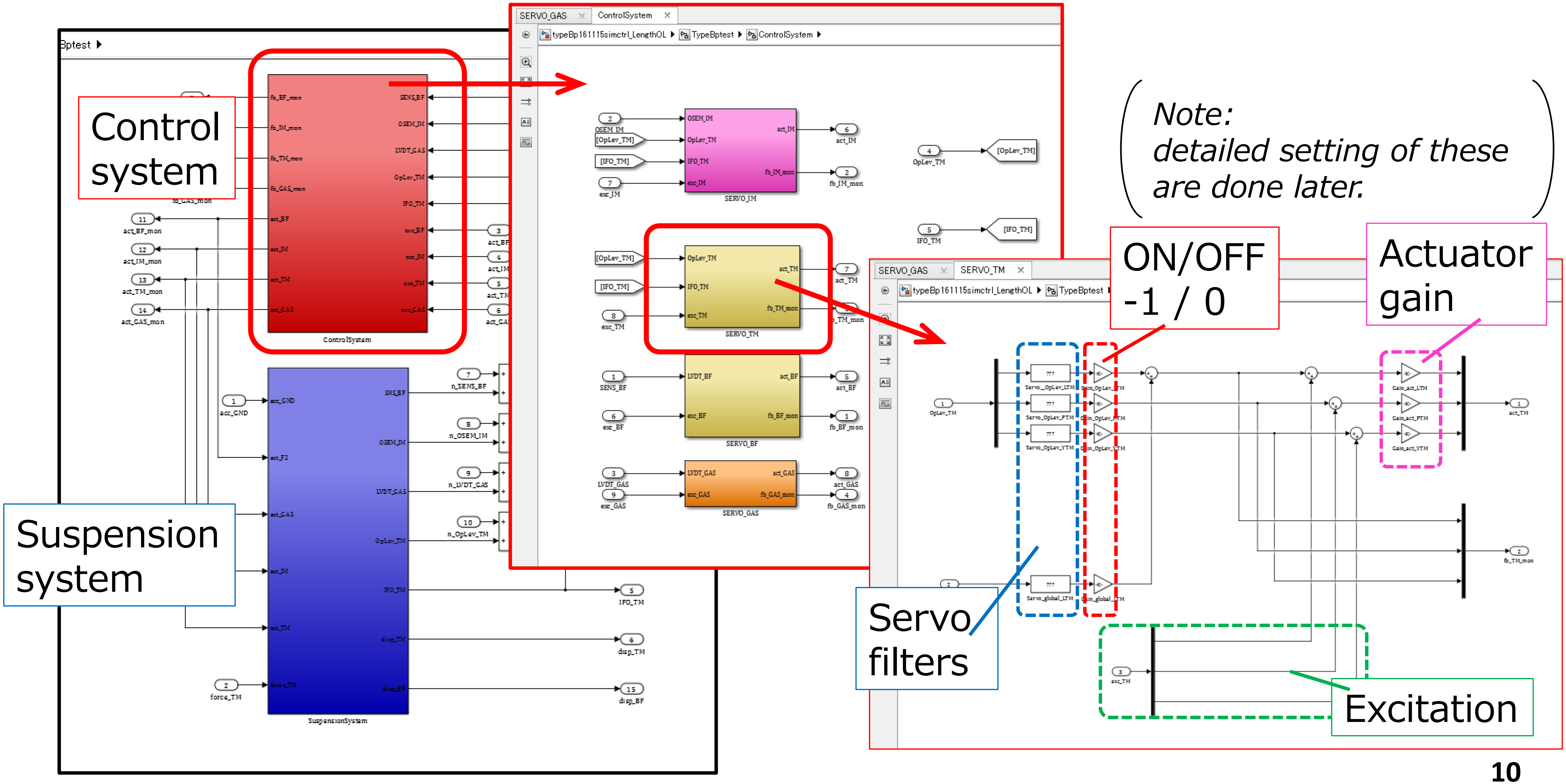
2-1-1. experimental-system-file structure (what's .slx?) 3



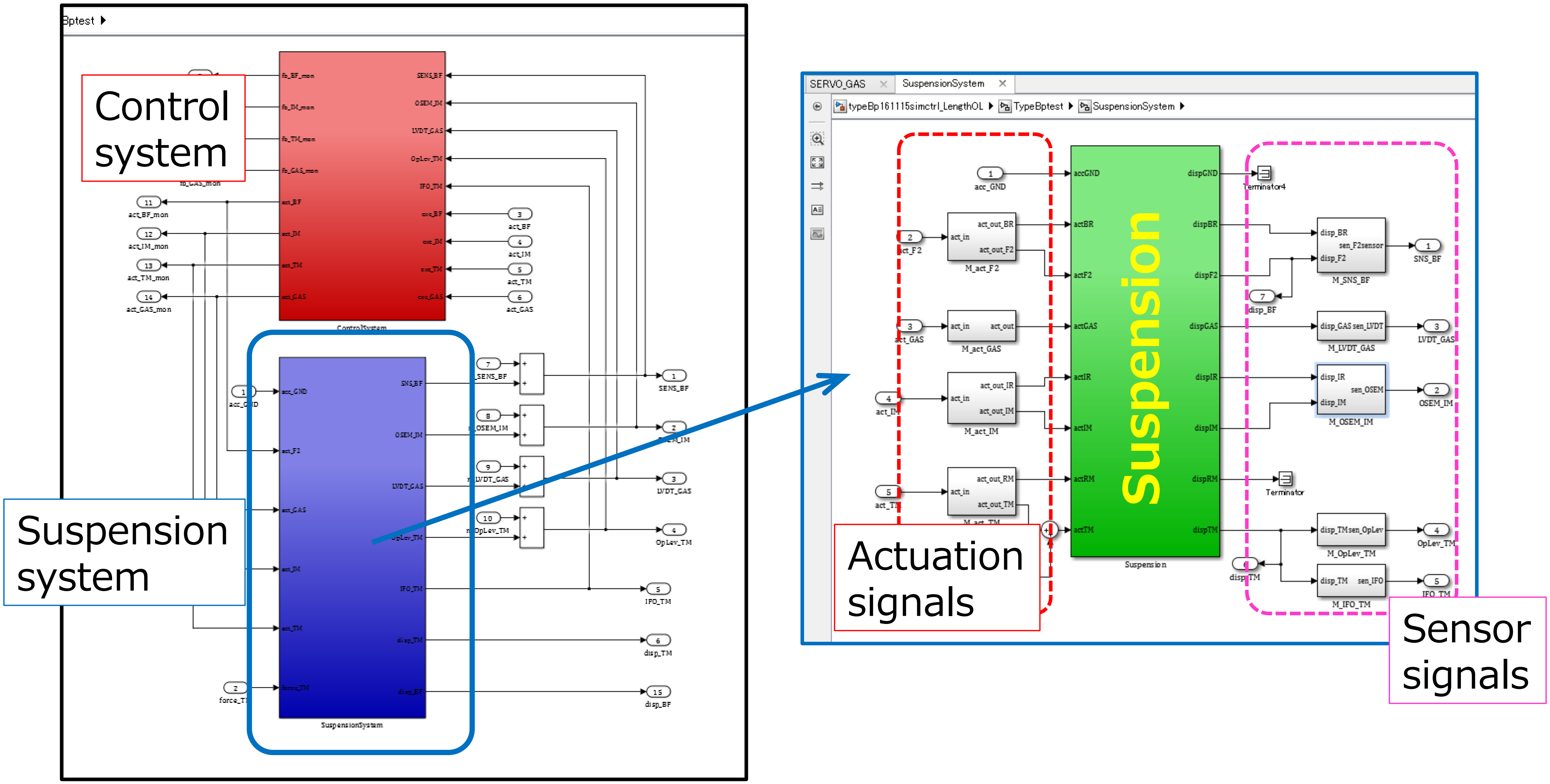
2-1-1. experimental-system-file structure (what's .slx?) 4



2-1-1. experimental-system-file structure (what's .slx?) 5



2-1-1. experimental-system-file structure (what's .slx?) 6



2-1-2. Steps for constructing .slx file

1. make control system part by mimicking some existing files.

2. make suspension system part (*mdl.slx) by using:
   - 2-1. *param.m & *mdlconst.m files

     or
   - 2-2. sumcon2controlsim.m(*1)

(*1) this file is available from:
https://granite.phys.s.u-tokyo.ac.jp/svn/LCGT/trunk/VIS/SuspensionControlModel/utility/sumucon2controlsim.m

2-1-2. *param.m & *mdlconst.m files

param.m should include information of:

1. physical parameters (Young's modulus, Poisson ratio etc.)
2. mass & moment of inertia of each component
3. suspension conditions (Wire length, diameter, etc.)
4. inverted pendulum, magnetic damper.

mdlconst.m should include information of:

1. path to the *param.m
2. registration of suspension component parameters
3. input & output variables
4. suspension model name

→ *In order to get *mdl.m file, execute "*mdlconst.m" on MATLAB.*

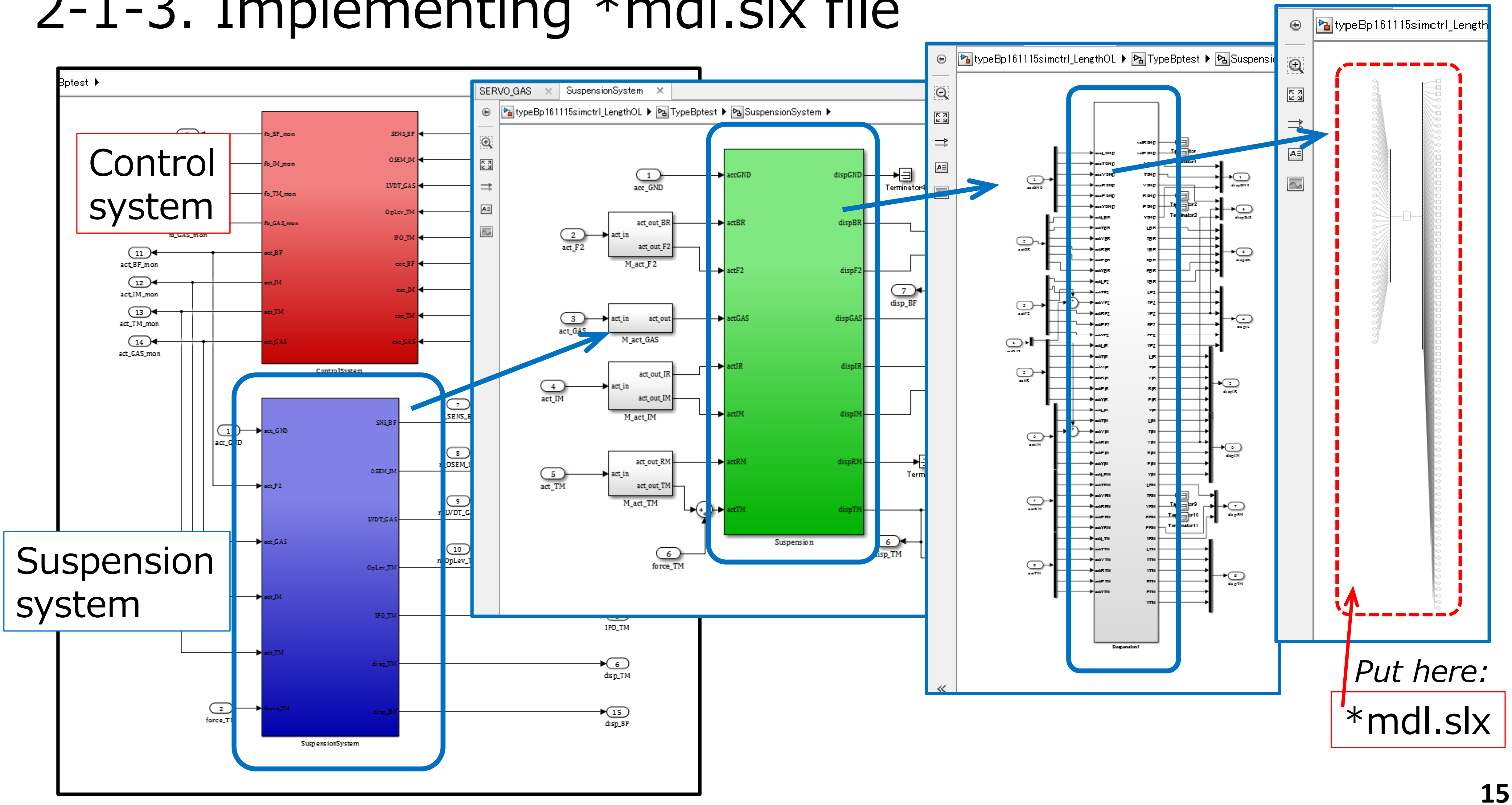
2-1-2. sumcon2controlsim.m

→ *In order to get *mdl.m file (suspension part)*

1. Execute “sumcon2controlsim.m” on MATLAB

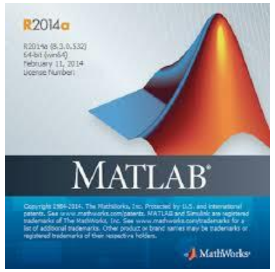
2. Follow its guide

2-1-3. Implementing *mdl.slx file



Preparation

1. Install MATLAB (2014)

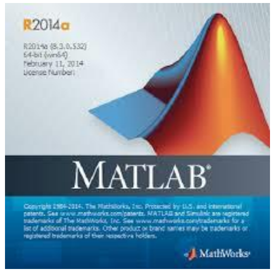


Procedure

2-1. make a virtual experimental system (.slx file).
2-2. make servo filters (.m file).
2-3. run measurement codes.

Preparation

1. Install MATLAB (2014)



Procedure

2-1. make a virtual experimental system (.slx file).
2-2. make servo filters (.m file).
2-3. run measurement codes.

2-3. Run measurement codes

ex. Transfer function plot (without control):

typeBp161115_nocontrol_TF.m

```
%% ABOUT THIS FILE
% ---------------------------------------------------------------------
% Type-B1 prototype for KAGRA
% Coded by Y. Fujii on 2016/11/16
% ---------------------------------------------------------------------

%% PRELIMINARY
clear all;  % Clear workspace
close all;  % Close plot windows
addpath('../../utility');  % Add path to utilities
g = 9.81;

%% IMPORT SUSPENSION MODEL
matfile='typeBp161115mdl';
load([matfile,'.mat']);

%% TUNING DAMPER
% This part compensates the failure in converting the structural damping to
% viscous damping.
% REDUCE DAMPING ON RIM
sys1.a(34,34)=sys1.a(34,34)/30; % RIM
sys1.a(40,40)=sys1.a(40,40)/30; % RRM
%sys1.a(46,46)=sys1.a(46,46)/30; % RTM
% INCREASE DAMPING ON YF0
%sys1.a(15,15)=sys1.a(15,15)*3000; % YF1
%sys1.a(15,15)=sys1.a(15,15)*1000; % YF1
% INCREASE DAMPING ON LF1
%sys1.a(13,13)=sys1.a(13,13)*3000; % LF1
% INCREASE DAMPING ON TF1
%sys1.a(14,14)=sys1.a(14,14)*3000; % TF1
% INCREASE DAMPING ON YRM, YTM
sys1.a(42,42)=sys1.a(42,42)*30;   % YRM
sys1.a(48,48)=sys1.a(48,48)*300;  % YTM
% INCREASE DAMPING ON VTM
sys1.a(45,45)=sys1.a(45,45)*100;  % VTM

%% IMPORT SERVO FILTERS
addpath('servofilter');         % Add path to servo
typeBp161115_no_control;  % NO CONTROL MODE
rmpath ('servofilter');         % Remove path to servo

%% IMPORT SIMULINK MODEL
mdlfile='typeBp161115simctrl_LengthOL'; % typeBp_bKAGRA ver.161115
st     =linmod(mdlfile);
invl   =strrep(st.InputName, [mdlfile,'/'],'');
outvl  =strrep(st.OutputName,[mdlfile,'/'],'');
sysc0  =ss(st.a,st.b,st.c,st.d,'inputname',invl,'outputname',outvl);
```

Line 153, Column 1   Spaces: 4   Objective-C

(to be Continued)

Path to utility

Import suspension model

Tune mechanical Q factors
for each degree of freedom
*by changing viscous damping strength.(*2)*

Path to servo filters

2-3. Run measurement codes

ex. Transfer function plot (without control):

typeBp161115_nocontrol_TF.m

(Continued)

```
%% IMPORT SERVO FILTERS
addpath('servofilter');          % Add path to servo
typeBp161115_no_control;  % NO CONTROL MODE
rmpath ('servofilter');          % Remove path to servo

%% IMPORT SIMULINK MODEL
mdlfile='typeBp161115simctrl_LengthOL'; % typeBp_bKAGRA ver.161115
st     =linmod(mdlfile);
invl   =strrep(st.InputName, [mdlfile,'/'],'');
outvl  =strrep(st.OutputName,[mdlfile,'/'],'');
sysc0  =ss(st.a,st.b,st.c,st.d,'inputname',invl,'outputname',outvl);

%% FREQUENCY
freq1=logspace(-2,2,1001);
freq2=logspace(-2,2,1001);

%% TRANSFER FUNCTION BF
bodesusplotcmpopt_myfunc(sysc0,{'actLBF','LVDT_LBF'},freq1,'legend',{'L'},...
    'ylim',[1e-4,1e2],'calibration',{gain_act_LBF},'unit1','1');
export_fig('figures/TF/typebp161115_forceTFcomp_LBF.pdf')

bodesusplotcmpopt_myfunc(sysc0,{'actTBF','LVDT_TBF'},freq1,'legend',{'T'},...
    'ylim',[1e-4,1e2],'calibration',{gain_act_TBF},'unit1','1',...
    'title','Transfer function from actTBF to TBF');
export_fig('figures/TF/typebp161115_forceTFcomp_TBF.pdf')

bodesusplotcmpopt_myfunc(sysc0,{'actVBF','LVDT_VBF'},freq1,'legend',{'V'},...
    'ylim',[1e-4,1e2],'calibration',{gain_act_VBF},'unit1','1',...
    'title','Transfer function from actVBF to VBF');
export_fig('figures/TF/typebp161115_forceTFcomp_VBF.pdf')

bodesusplotcmpopt_myfunc(sysc0,{'actRBF','LVDT_RBF'},freq1,'legend',{'R'},...
    'ylim',[1e-4,1e2],'calibration',{gain_act_RBF},'unit1','1',...
    'title','Transfer function from actRBF to RBF');
export_fig('figures/TF/typebp161115_forceTFcomp_RBF.pdf')

bodesusplotcmpopt_myfunc(sysc0,{'actPBF','LVDT_PBF'},freq1,'legend',{'P'},...
    'ylim',[1e-4,1e2],'calibration',{gain_act_PBF},'unit1','1',...
    'title','Transfer function from actPBF to PBF');
export_fig('figures/TF/typebp161115_forceTFcomp_PBF.pdf')

bodesusplotcmpopt_myfunc(sysc0,{'actYBF','LVDT_YBF'},freq1,'legend',{'Y'},...
    'ylim',[1e-4,1e2],'calibration',{gain_act_YBF},'unit1','1',...
    'title','Transfer function from actYBF to YBF');
export_fig('figures/TF/typebp161115_forceTFcomp_YBF.pdf')
```

Line 52, Column 27   Spaces: 4   Objective-C

Import Simulink model

Set frequency

Plot transfer functions of suspensions (from actuator to sensor)

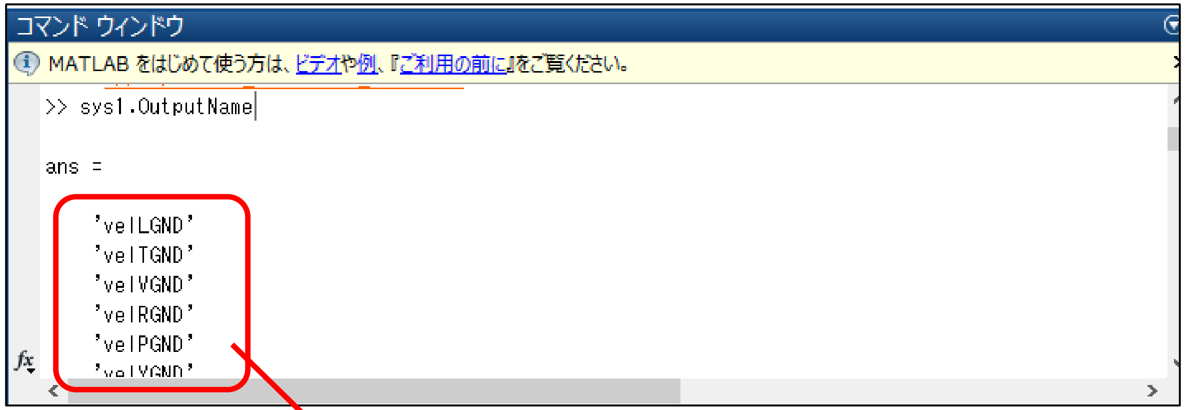
2-3. Run measurement codes (Note)

*(*2) Tuning of mechanical Q factors:*

```
%% TUNING DAMPER
% This part compensates the failure in converting the structural damping to
% viscous damping.
% REDUCE DAMPING ON RIM
sys1.a(34,34)=sys1.a(34,34)/30; % RIM
sys1.a(40,40)=sys1.a(40,40)/30; % RRM
%sys1.a(46,46)=sys1.a(46,46)/30; % RTM
% INCREASE DAMPING ON YF0
%sys1.a(15,15)=sys1.a(15,15)*3000; % YF1
%sys1.a(15,15)=sys1.a(15,15)*1000; % YF1
% INCREASE DAMPING ON LF1
%sys1.a(13,13)=sys1.a(13,13)*3000; % LF1
% INCREASE DAMPING ON TF1
%sys1.a(14,14)=sys1.a(14,14)*3000; % TF1
% INCREASE DAMPING ON YRM, YTM
sys1.a(42,42)=sys1.a(42,42)*30;    % YRM
sys1.a(48,48)=sys1.a(48,48)*300;   % YTM
% INCREASE DAMPING ON VTM
sys1.a(45,45)=sys1.a(45,45)*100    % VTM
```

1. How to check these numbers:

→ execute "sys1.OutputName" on MATALB.



For instance,
if you want to tune
Q factor of TM motion
in Yaw direction (YTM),
you can input:
Sys1.a(48,48).

Vel/disp	L	T	V	R	P	Y	Vel/disp	L	T	V	R	P	Y
Vel GND	1	2	3	4	5	6	Vel IM	31	32	33	34	35	36
(Disp)GND	7	8	9	10	11	12	Vel RM	37	38	39	40	41	42
Vel BR	13	14	15	16	17	18	Vel TM	43	44	45	46	47	48
Vel F2	19	20	21	22	23	24	(disp)BR	49	50	51	52	53	54
Vel IR	25	26	27	28	29	30	…	…					