

Matlab / Simulink

Starting a Simulink job from a Matlab Script

Suppose you have a model in Simulink. And suppose that parameters of this model are calculated in a Matlab script.

In that case it could be handy that if you run the script in Matlab it also executes the simulink model and gives you the results. (instead of running the matlab script, switching to Simulink, start a simulation there and then clicking and autoscaling the plotwindow from Simulink).

I have got some options.

Option1

Suppose the simulink model is saved under the name BlueBox_ClosedLoop.mdl

```
% Start and simulate simulink model (StructureWithTime)
Model = 'BlueBox_ClosedLoop';
load_system(Model)
% Copy current settings associated with model
Cs = getActiveConfigSet(Model);
Model_cs = Cs.copy;
% and adapt them
set_param(Model_cs, 'AbsTol', '1e-5', ...
    'SaveState', 'on', ...
    'SaveOutput', 'on', ...
    'SaveFormat', 'StructureWithTime');
% Run simulation
SimOut = sim(Model, Model_cs);
% Extract and plot data
SimOutScopeData = SimOut.get('ScopeData')
figure(1);
plot(SimOutScopeData.time, SimOutScopeData.signals.values);
title(sprintf('Simulink Output for %s \n', SimOutScopeData.blockName));
grid on;
```

Option 2

Replace the lines below "% Extract and plot data" with the following:

Matlab / Simulink

```
% Create a run in the Simulation Data Inspector
runID = Simulink.sdi.createRun('My Run','base',{'SimOut'});
% See the results in Simulation Data Inspector
Simulink.sdi.view;
```

Option 3

Below a function you could call like

```
simulink_run('Model Name')
```

to create a figure for each scope in your Simulink model. Save this m_script somewhere in your Matlab Path

```
%void SimulinkRun(string Model)
function simulink_run(Model)
%% Start and simulate simulink model (make a figure for each scope)
%
load_system(Model);

% Copy current settings associated with model
Cs = getActiveConfigSet(Model);
Model_cs = Cs.copy;
% and adapt them
set_param(Model_cs,'AbsTol','1e-5',...
          'SaveOutput','on',...
          'SaveFormat','StructureWithTime');

% Find all scope blocks in the model
ModelScopes = find_system(Model,'LookUnderMasks','on','IncludeCommented',...
                          'on',...
                          'AllBlocks','on','BlockType','Scope','DefaultConfigurationName',...
                          'Simulink.scopes.TimeScopeBlockCfg');
% Set all Scope parameters
% Note use {} instead of () to access cell arrays
for i=1:length(ModelScopes)
    ScopeConfig = get_param(ModelScopes{i},'ScopeConfiguration');
    ScopeConfig.DataLogging=true;
    ScopeConfig.DataLoggingVariableName=['ScopeData' num2str(i)];
    ScopeConfig.DataLoggingSaveFormat='StructureWithTime';
end

% Run simulation
SimOut = sim(Model, Model_cs);

% Extract and plot data
for i=1:length(ModelScopes)
```

Matlab / Simulink

```
SimOutScopeData = SimOut.get(['ScopeData' num2str(i)])
figure(i);
plot(SimOutScopeData.time,SimOutScopeData.signals.values);
title(sprintf('Simulink Output for %s',SimOutScopeData.blockName),
...
      'interpreter', 'none');
grid on;
end
end % Function
```

Before Matlab 2016 the function `simplot` was still supported and then it was implemented as follows.

Suppose the simulink model is saved under the name `Pr_E4_Deadbeat.mdl`

```
%% Start and simulate simulink model
% Open simulink model
Pr_E4_Deadbeat;
% command gcs refers to the current system model
% command gcb refers to the current block model
set_param(gcs, 'SaveOutput', 'on');
set_param(gcs, 'SaveFormat', 'StructureWithTime');
%Met 'Scope' de naam van de simulink scope die je wilt zien
set_param([gcs '/Scope'], 'SaveToWorkspace', 'on');
set_param([gcs '/Scope'], 'SaveName', 'ScopeData');
sim(gcs);
simplot(ScopeData);
```

The 2 lines after opening the model instruct the simulink simulation to enable saving of the simulationresult to the workspace of Matlab. 'sim(gcs)' starts the simulation and the next line plots the result. ScopeData is the default name that simulink uses in a plotwindow.

Unieke FAQ ID: #1025

Auteur: Harry van der Pol

Laatst bijgewerkt:2017-07-03 17:33