
Chapter 6

User's Guide to on-line optimization

This chapter is intended as a user's guide to on-line optimization using OCC. The various steps involved during an on-line optimization run has been described in Chapter 2, and thus will not be repeated here.

A short summary of the various steps involved in setting up an on-line optimization run is first presented and detailed descriptions of each step follow. The combined-cycle model as described in Appendix A-1 will be used when necessary for illustration purposes.

6.1 Introduction

The steps involved in preparing for an on-line optimization run can be divided into 2 phases. In the first phase, the files necessary for the run are generated by the user and saved in the appropriate directories. The list comprises of the Simulink model of the process to be optimised and its auxiliary files (Section 6.2), the executable files (".EXE files") for process optimization ("OPT" – Section 6.3) and system identification ("ID" – Section 6.4), as well as the data required for initialising the first cycle (Section 6.5).

The second phase involves setting the parameters prior to the on-line run using the options menu (Section 6.6) and on-line-wizard (Section 6.7).

The steps are summarized in Fig. 6-1.

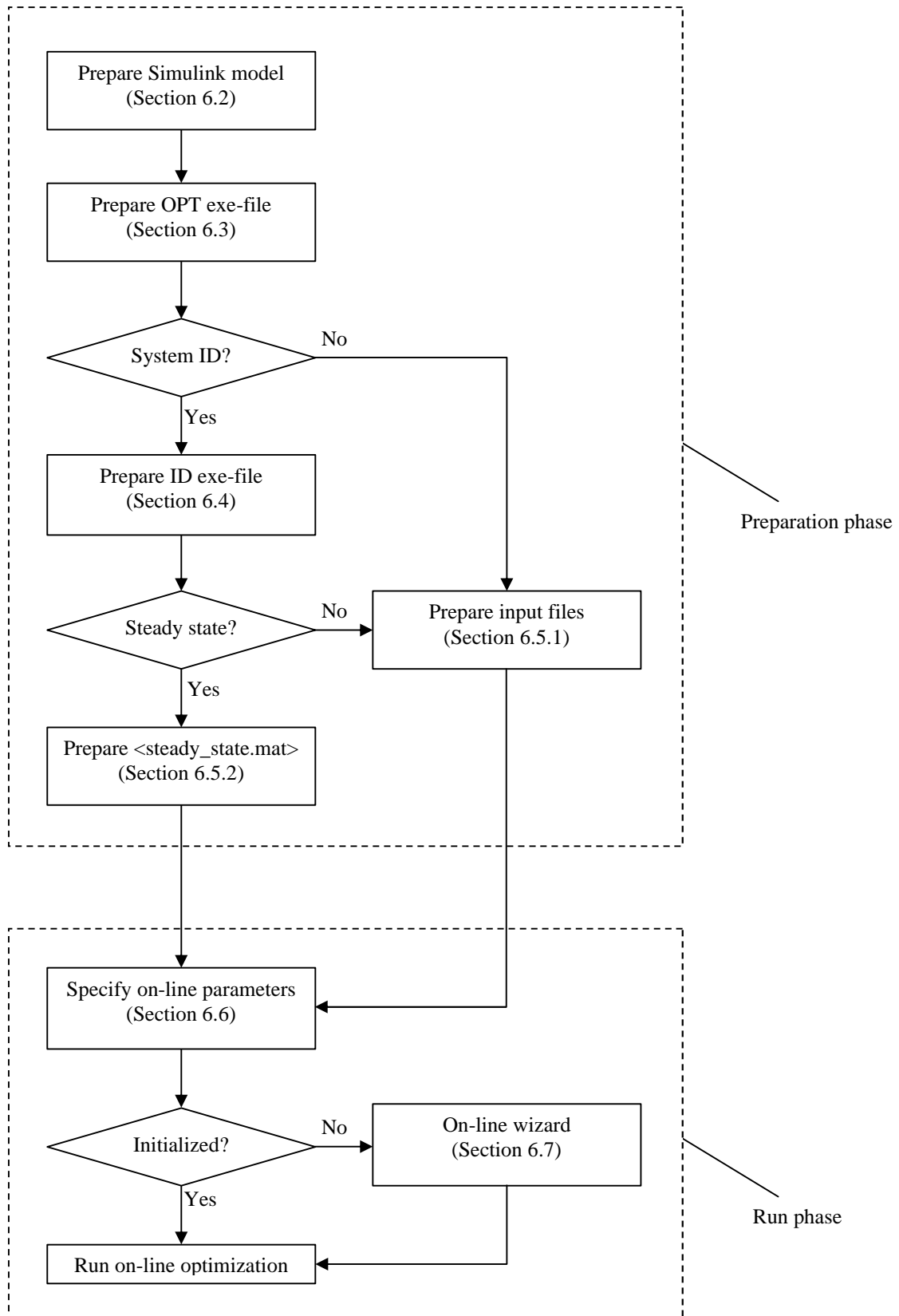


Fig. 6-1: Preparations for an on-line optimization run

6.2 The Simulink model

To simulate the actual physical process during on-line optimization, the user is required to create a Simulink model of the process to be optimized. Control variables, e.g. gas turbine load and mass flow of fuel, obtained from the process optimization are fed to the Simulink model to calculate process variables, e.g. total plant load and exit gas temperature, some of which are then passed back to the optimization cycle as measurements.

Fig. 6-2 shows the Simulink model of the combined-cycle power plant. The various components of the plant, e.g. gas turbine and superheater, have been grouped into subsystems to eliminate screen clutter.

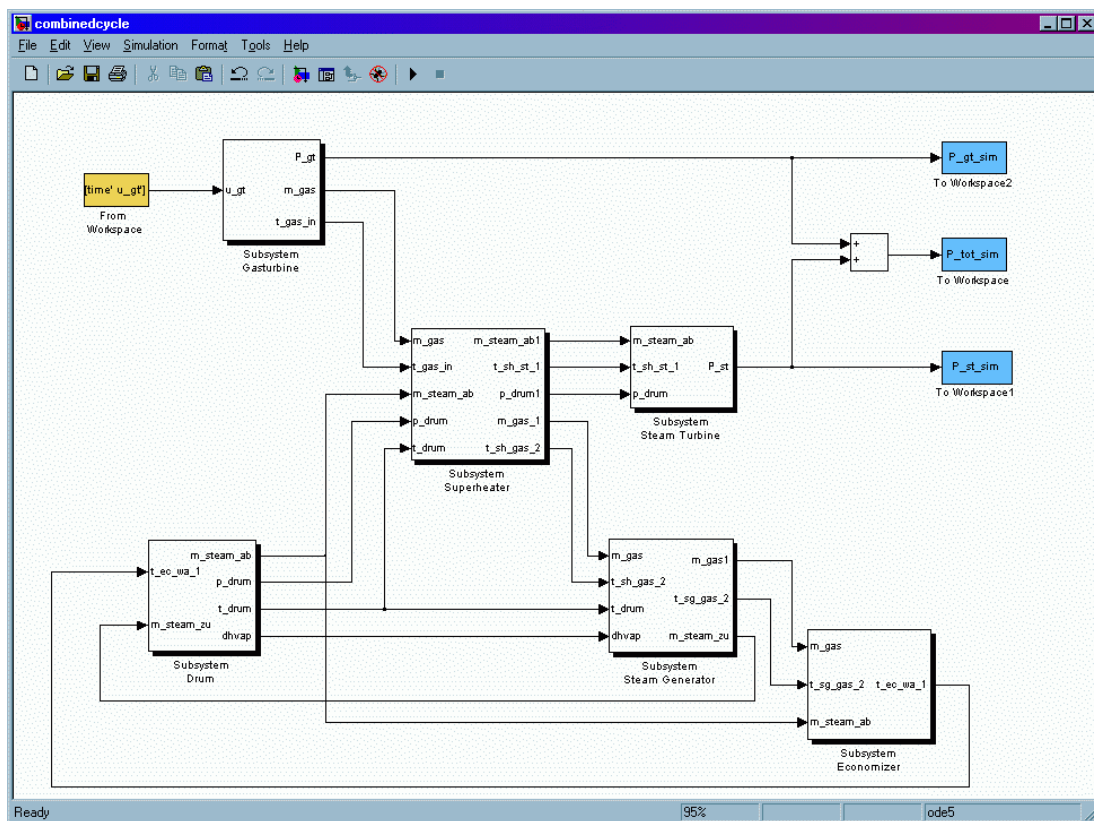


Fig. 6-2: Simulink model for combined-cycle power plant

The following sections explain the requirements that have to be fulfilled when creating the Simulink model.

6.2.1 Process constants

Models for chemical and power engineering processes are often complex and involve many process constants like the universal gas constant R , and reaction constants k . These can either be directly included in the Simulink model, or defined in MATLAB syntax in the M-file <sim_constants.m>, which is executed before the simulation is started. This file

must be saved in the model root directory, e.g. C:/OCC/models/combinedcycle/, and will serve all the various cases of this model.

As the constants have to be present in the MATLAB workspace in which the Simulink model is run, it is important that <sim_constants.m> does not include any function calls. Of course, the file can also be used to call other M-files, which, in turn, would have to obey the above rule.

Constants defined in <sim_constants.m> are read by the Simulink model by using the “Constant” block and setting the “constant value” field to the name of the constant.

An example of the “Constant” block can be found in Fig. 6-3.

6.2.2 Saving process variables from the simulation

The “To Workspace” block can be used to save any variable calculated by Simulink. It is important that variable names end with the extension “_sim”. Only such variables will be saved by OCC to the history file <history.mat>. OCC automatically saves the time vector of the simulation under the variable *time_sim*.

Variables to be used as measurements during the ID process must be sent to the workspace and saved to the history file.

Simulink allows its users to group signals together using the „MUX“ block. However, defining such multidimensional vectors as output to be saved in the history file is not allowed.

It is important that analogous variables have corresponding names. For example, the steel temperature is named *t_steel_sim* and *t_steel* in the Simulink and OCC models respectively. Variables defined in this manner are automatically matched to one another when OCC generates starting guesses from measurements. Also, they would be plotted together on the same axis on the on-line menu when the user chooses to view their results.

An example of the “To Workspace” block can be found in Fig. 6-3.

6.2.3 Initial condition for integrators

All dynamic models would involve the use of “Integrator” blocks. For differential variables where the initial condition changes with each cycle, e.g. steel temperature, the user is required to specify the name of the variable to be used as the initial condition for the integrator. One way of doing this is to set the “initial condition source” option of the integrator to “external”, and connect a “Constant” block to its input. The name of the constant to be used as initial condition must end with the extension “_ini”. The value of

the initial condition is calculated automatically by OCC by interpolating the simulation results from the previous cycle. This requires the variable to be sent to the workspace during each simulation, so that it can be saved later on. The rule of adding the extension “_sim” to the variable name applies (see 6.2.2).

The values of the initial conditions are defined in the M-file <sim_ini.m>. This file is generated and executed automatically during every cycle of the on-line run by OCC before the simulation is started.

Fig. 6-3 shows the configuration of Simulink blocks required to calculate the differential variable t_{steel_sim} . An external source has been used for the “Integrator” block in this case for illustrative purposes. In practice, the initial condition t_{steel_ini} can also be entered as an internal source in the block parameters menu.

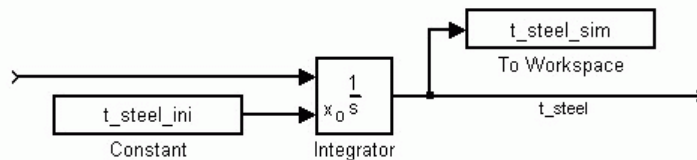


Fig. 6-3: Simulink blocks used to calculate variable t_{steel_sim}

6.2.4 Control variables

The control variables, obtained as solutions from process optimization, are fed to the Simulink model via the “From Workspace” block. Format differences in the entry requirements of this Simulink block and the optimization results found in <optdata.m> mean that the vectors for the control variables have to be transposed.

The “From Workspace” block for control variable u_{gt} is shown in Fig. 6-4. Both vectors $time$ and u_{gt} are transposed by adding an apostrophe.

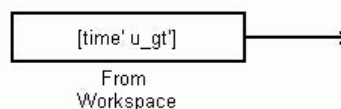


Fig. 6-4: Control variable

6.3 The .EXE file for process optimization

For obvious reasons, the .EXE file for OPT is the single most important file for the on-line optimization. The model for the optimization is defined in the OCC file, which is located in a case directory. For example, the full pathname for the combined-cycle model is <c:/occ/models/combinedcycle/case01/combinedcycle.occ>.

6.3.1 Generating the OPT .EXE file

The OCC file can be created conveniently using the edit menu. The user can make use of the various pop-up menus, edit boxes and checkboxes to specify the OCOMA parameters. These include the method of discretization, solver version and number of timesteps. It goes without saying that the “On-line Optimization” checkbox has to be checked.

The user is basically free to choose the names of variables included in the process model to be optimized. However, these names may not end with “_sim” as this extension is reserved for analogous variables from the Simulink simulation (see 6.2.2). In addition, the variables names have to have at least two letters.

To generate the .EXE file, the user first has to run OCOMA to generate the necessary FORTRAN files. This can be done by clicking “Generate FORTRAN Code” on the off-line menu. Once this has been completed, click “Make Executable” to compile the files; this not only compiles the process model defined in the OCC file, but also links the entire NLP solver that the user has selected. The .EXE file thus generated is therefore complete in itself and can be renamed or moved to another case directory as the user pleases.

The .EXE file receives the name <exe_*.exe>, or <exe_*> for UNIX/LINUX operating systems, where * is the name of the OCC file without the extension “.occ”.

6.3.2 The OPT structure file

An auxiliary M-file is generated automatically when the user activates OCOMA to generate the FORTRAN codes. Similar to the .EXE file, the structure file is named <structure_*.m>, where * is the name of the OCC file without the extension “.occ”.

The structure file is extremely important during on-line optimization as it contains information about the structure of the process model and optimization parameters defined in the OCC file. It is therefore crucial that the .EXE and structure files have corresponding

names¹⁴ and are both found in the same case directory. For example, the file `<combined_cycle.occ>` generates both `<exe_combined_cycle.exe>` (or `<exe_combined_cycle>` in the case of Unix/Linux systems) and `<structure_combined_cycle.m>`.

The data that the structure file provides during on-line optimization are summarized in Table 6-1.

Variable/List	Description	Remarks
<i>time</i>	Vector containing value of all collocation points	
<i>varnames</i>	List of all process variables	
<i>initialnames</i>	List of all differential variables	
<i>measnames</i>	List of all measured variables	Only relevant for ID
<i>paranames</i>	List of all process parameters	
<i>flexibound_mode</i>	0: Fixed bounds for measured variables 1: Flexible bounds	Only relevant for ID
<i>*_LOWER</i>	Lower bound for variable *	Only relevant for ID
<i>*_UPPER</i>	Upper bound for variable *	Only relevant for ID

Table 6-1: Important information contained in structure file

An example of the structure file for combined-cycle power plant is shown below.

```
% Collocation points
time(1) = 12.5000000000;
time(2) = 64.47368421052632;
...
time(19) = 948.0263157894738;
time(20) = 1000.000000000000;

% The measured variables
measnames={
};

% Initial conditions
initialnames={
't_sh_steel_2'
't_sh_steel_1'
't_sg_steel_2'
't_sg_steel_1'
't_ec_steel_2'
't_ec_steel_1'
'p_drum'
'P_st'
};
```

¹⁴ Important especially if the user manually renames and/or moves the .EXE file.

```

% State variables
varnames={
't_gas_in'
'm_steam'
...
'p_drum'
'P_st'
};

% Parameters
paranames={
};

FLEXIBOUND_MODE=0;

```

6.4 The .EXE file for system identification

The procedures for generating the .EXE file for ID are similar to those for OPT (Section 6.3). Except in the case of parallel ID, where the process models can be significantly reduced in size, the main difference between the .EXE files lies in the objective function.

In addition to the requirements mentioned in Sections 6.3.1 and 6.3.2, the following rules and recommendations apply.

6.4.1 Defining the measured variables

The values of certain process variables obtained from the Simulink simulation can be fed to .EXE file through the text file <meas.dat> each time ID is performed. These data serve as measurements from the plant (e.g. temperatures and pressures) and help in identifying other variables that cannot be measured directly.

Variables to be used as measurements are specified in the OCC file using the “add_measured_data” function. For example, the command

```
add_measured_data(t_gas_MEAS);
```

instructs OCC to capture the simulation data of variable t_{gas_sim} and include them in <meas.dat>. The extension “_MEAS” is compulsory to differentiate the measurement t_{gas_MEAS} from the optimization variable t_{gas} .

For OCC to be able to “measure” a variable it is necessary for it to be saved in the history file. This is achieved by using the “To Workspace” block in the SIMULINK model to record the variable, in this case t_{gas_sim} (see Section 6.2.2).

6.4.2 The ID objective function

A commonly used objective function for ID is the least-square function, defined by

$$lstsqr = \sum_{i=1}^n a_i \cdot \left(\frac{x_{id,i} - x_{meas,i}}{x_{id,i}} \right)^2 \quad (6-1)$$

where n : total number of measured variables

a_i : relative weight of variable

x_{id} : ID data

x_{meas} : measurements

The following shows an example of how the least-square objective function can be defined in the OCC model for ID of the combined-cycle power plant. Measurements of variables t_{gas} , t_{drum} , p_{drum} , m_{gas} and m_{steam} are used.

```
add_measured_data(t_gas_MEAS);
add_measured_data(t_drum_MEAS);
add_measured_data(p_drum_MEAS);
add_measured_data(m_gas_MEAS);
add_measured_data(m_steam_MEAS);

add_aeq(delta_t_gas = ((t_gas -t_gas_MEAS )/t_gas )^2);
add_aeq(delta_t_drum = ((t_drum -t_drum_MEAS )/t_drum )^2);
add_aeq(delta_p_drum = ((p_drum -p_drum_MEAS )/p_drum )^2);
add_aeq(delta_m_gas = ((m_gas -m_gas_MEAS )/m_gas )^2);
add_aeq(delta_m_steam = ((m_steam-m_steam_MEAS)/m_steam)^2);

add_aeq(lstsqr = delta_t_gas + delta_t_drum + delta_p_drum
        + delta_m_gas + delta_m_steam);

objective(lstsqr,discrete);

ocomo();
```

In this example, the algebraic variable $lstsqr$ corresponds to Eq. (6-1) for the case where all the relative weights are set to unity. The parameter “discrete” is required to generate a objective function where values at each discrete timestep are considered. Leaving out this parameter generates a final-time objective.

6.4.3 The ID structure file

As with the .EXE file for OPT, a structure file in MATLAB syntax is generated when OCOMA is invoked to generate FORTRAN codes (see Table 6-1). The structure file for the combined-cycle power plant problem is shown below.

```
% Collocation points
time(1) = 2.5000000000;
time(2) = 12.89473684210526;
...
time(19) = 189.6052631578947;
time(20) = 200.0000000000;

% Measured variables
```

```

measnames={
't_gas_MEAS'
't_drum_MEAS'
'p_drum_MEAS'
'm_gas_MEAS'
'm_steam_MEAS'
};

% Initial conditions
initialnames={
'p_drum'
'P_st'
};

% State variables
varnames={
't_gas_in'
'm_steam'
...
'p_drum'
'P_st'
};

% Parameters
paranames={
't_sh_steel_1_IC'
't_sh_steel_2_IC'
't_sg_steel_1_IC'
't_sg_steel_2_IC'
't_ec_steel_1_IC'
't_ec_steel_2_IC'
};

FLEXIBOUND_MODE=0;

```

In this particular example, the initial conditions of the various steel temperatures $t_*_steel_*$ have been defined as parameters to be optimized concurrently. These parameters are listed under the vector *paranames* and contain the extension “_IC” (for “Initial Condition”). At the same time, the number of differential variables listed under the vector *initialnames* has been reduced as compared to the OPT structure file (see Section 6.3.2).

6.5 Initialization files

Before an on-line optimization run can be started, the user has to ensure that certain files required to initiate the first cycle are present.

6.5.1 The input file for process optimization

The user has to provide the input files for the OPT .EXE file in the following two cases:

1. On-line optimization without system identification,

2. On-line optimization with system identification, but not utilizing steady-state data as a starting point for cycle one.

The first input file required is called <initials_opt1.dat>¹⁵ and contains initial conditions of the differential variables defined in the OCC model. The first line is reserved for comments and will be skipped by the .EXE file. Initial conditions are specified first by naming the variable name concerned, followed by the value of the variable on the next line. The user is not required to sort the variables in any particular order. An example of <initials_opt1.dat> is shown here.

```

#-- Process Optimization: <initials.dat>, Cycle 1 --#
P_st
42.889871
p_drum
54.888790
t_ec_steel_1
248.289745
t_ec_steel_2
185.031214
t_sg_steel_1
276.020134
t_sg_steel_2
272.069144
t_sh_steel_1
505.310823
t_sh_steel_2
433.963592

```

The user can create various versions of this file, each with a different set of initial conditions. In the case of the combined-cycle power plant problem, it is thus possible to start the on-line optimization from different loads, e.g. 100 MW, 120 MW and 150 MW. The values of the initial conditions can be obtained through simulation.

The second input file is named <svalues_opt1.dat>. This provides the starting guesses of all process variables and parameters. As above, the first line is reserved for comments. Starting guesses of variables are defined by first naming the variable. On the next line the user then provides the n values corresponding to the n timesteps specified in the .EXE file. Only one value is required for parameters. An example is shown below.

```

#-- Process Optimization: <svalues.dat>, Cycle 1 --#
P_dot_st
0.0000000  0.0000000  0.0000000  ...  0.0000000
P_gt
85.000000  85.000000  85.000000  ...  85.000000
P_st
42.889871  42.889871  42.889871  ...  42.889871

```

¹⁵ The suffix "opt1" indicates that this is the input file for OPT during cycle one.

```

P_tot
127.889868    127.889868    127.889868    ...    127.889868
...
example_parameter
0.5
...

```

Both input files are to be saved in the model root directory. During the run they would be copied into the case directory and renamed automatically to <initials.dat> and <svalues.dat> respectively.

6.5.2 The steady-state MAT-file

If the user desires to perform ID starting from cycle one, a MAT-file named <steady_state.mat> must be provided. This provides data regarding the steady state of the plant before the on-line optimization is started, and serves as a source of information for OCC to generate the input files <initials.dat>, <meas.dat> and <svalues.dat>, which are needed for the first round of ID. The MAT-file must be saved in the model root directory.

The MAT-file contains SIMULINK simulation data and has to be created by the user. Using the combined-cycle problem as an example, the steps are as follows:

1. Clear MATLAB workspace.
2. Change MATLAB working directory to the model root directory.
3. Execute <sim_constants.m> to load the constants into the workspace.
4. Execute <sim_ini.m> to load the initial conditions into the workspace.
5. Define variable *time*, e.g.

```
“time = 0:10:1000;”
```

6. Define control variables, e.g.

```
“u_gt = 0.5 * ones(1:length(time));”
```

This command assigns *u_gt* as an array with the same dimension as *time*. Each element has the value 0.5.

7. Run simulation <combined_cycle.mdl>:

```
“time_sim = sim('combined_cycle.mdl');”
```

8. Save all variables with suffixes “_sim” and “_ini” under <steady_state.mat>:

“save steady_state *_sim *_ini;”

9. Clear MATLAB workspace.

6.6 Runtime parameters for on-line optimization

A number of parameters need to be set before on-line optimization can be started. This is done using the on-line submenu of the OCC options menu shown in Fig. 6-5.

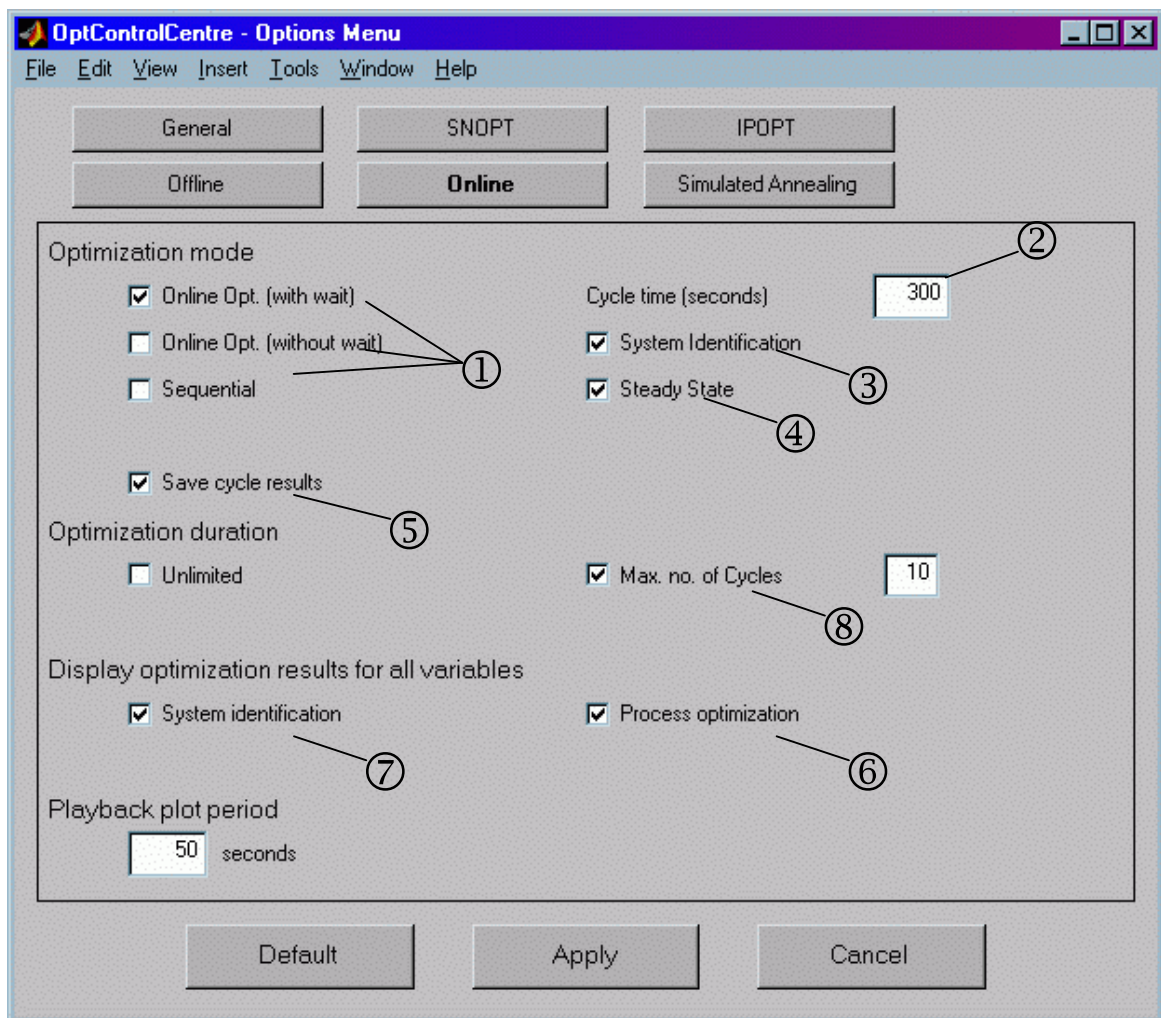


Fig. 6-5: The on-line submenu of the OCC options menu

6.6.1 Optimization mode

Using the on-line options submenu, the user is able to choose between three different general modes of operation for the on-line optimization (item ①).

In mode 1 (“On-line Opt. with wait”) the optimization cycle is repeated at regular intervals (cycle times). This is the normal mode for on-line optimization. The process time during which the n^{th} on-line cycle is started, also known as the cycle launch time, is defined as:

$$t_{launch,n} = t_{launch,n-1} + t_{cycle} \quad (6-2)$$

The “wait” in the name of this particular mode refers to the period of time between completion of a particular cycle and the start of the next. In this study, where the actual plants are only represented by Simulink models, the wait is not implemented in real-time.

Mode 2 (“On-line Opt. without wait”), on the other hand, does not require any cycle time to be specified. The on-line cycle is repeated immediately after all the necessary steps in the current cycle are completed. The launch time of the n^{th} cycle is therefore:

$$t_{launch,n} = t_{launch,n-1} + t_{duration,n-1} \quad , \quad (6-3)$$

where $t_{duration,n-1}$: real-time duration required to perform all steps in the $(n-1)^{\text{th}}$ cycle.

Mode 3 (“Sequential optimization”) is used primarily to generate starting guesses for large problems which would not converge easily otherwise. This new mode has been described in Section 3.5. As with mode 1, a cycle time has to be specified by the user. Unlike modes 1 and 2, mode 3 requires neither ID nor process simulation. The results from the n^{th} cycle are simply appended to those of the previous $(n-1)$ cycles, which are, in turn, truncated at $t_{launch,n}$. The absence of ID means that input files have to be prepared and saved in the model root directory to initialize the first cycle. This mode is not directly relevant for on-line optimization, but instead only makes use of the on-line capabilities of OCC to generate starting guesses for large problems. Instructions to initialization of mode 3 can be found in Section 6.8.

6.6.2 Cycle time

The user has to define the cycle time if on-line optimization mode 1 (“On-line Opt. with wait”) is chosen. The term “cycle time” refers to the constant time interval (in process time, not real-time) that elapses between two on-line cycles. The value can be set in the on-line options submenu (item ②).

The recommended value of the cycle time lies between 30% and 50% of t_{final} of the OPT problem. The following points should be taken into consideration:

1. If cycle time is too small:

The on-line optimization will be repeated too often and therefore the process progresses only very slowly (in process time). Also, a new cycle might be invoked before the old one is completed. This happens when the total real-time duration required to perform ID and OBJ exceeds the cycle time.

2. If cycle time is too large:

The starting guesses for the OPT problem of the new cycle might not be sufficiently good. This is because the bulk of the starting guesses for cycle n is derived from the optimization results from cycle $(n-1)$, with the values at the last timestep being held constant over the future horizon, for which no data is available yet. Setting the cycle time too large would thus mean that the starting guesses for the future horizon could deviate from the actual solution greatly. This could lead to convergence difficulties for large optimization problems.

6.6.3 Saving results from each cycle

By checking the “Save cycle results” option in the on-line options submenu (item ⑤), the following files that are generated during each on-line cycle will be renamed and saved in the case directory.

File generated	New filename ¹⁶	Description
<initials.dat>	<initials_ <i>filename</i> *.dat>	Initial conditions for ID and OPT .EXE files
<meas.dat>	<meas_ <i>filename</i> *.dat>	Measurements for ID .EXE file
<bounds.dat>	<bounds_ <i>filename</i> *.dat>	Variable bounds for ID .EXE file
<svalues.dat>	<svalues_ <i>filename</i> *.dat>	Starting guesses for ID and OPT .EXE files
<sim_ini.m>	<sim_ini*.m>	Initial conditions for Simulink simulation
<simdata.mat>	<simdata*.mat>	Results file from Simulink simulation

Table 6-2: Files generated during on-line optimization

This options allows the user to examine the results from each cycle more closely after the run has been completed.

On the other hand, the optimization results file <optdata.m> is converted, renamed and saved automatically as <optdata_ *filename**.mat> and <optdata_opt*.mat> for ID and OPT respectively¹⁷. Each of these files is essential for the proper running of the on-line optimization and is also required for the various plot functions in OCC during the actual on-line run as well as playback. Unchecking the “Save cycle results” option will have no effects on these files.

¹⁶ * and *filename* denote the cycle number and name of the relevant .EXE file respectively

¹⁷ The M-files are converted to MAT-files to reduce the disk space.

6.6.4 Number of optimization cycles

The user can specify the maximum number of on-line cycles to be carried out using this option (item ⑥). Checking “Unlimited” sets the value to 999.

6.6.5 Display ID and OPT results

Checking items ⑦ and ⑧ in the on-line options submenu allows the optimization results from ID and OPT respectively to be displayed graphically in separate figures. The user can make use of this option to examine the results more closely. This is especially useful during the ID process. Where simulation results are available from the history file <history.mat>, they will be plotted on the same axis as the corresponding variables obtained from ID. For example, variables u_{gt} and u_{gt_sim} will be plotted together. In this way, the user would be able to judge the quality of the ID process rather quickly and conveniently. An example of the plots from the ID of the economizer unit in the combined-cycle problem is shown in Fig. 6-6.

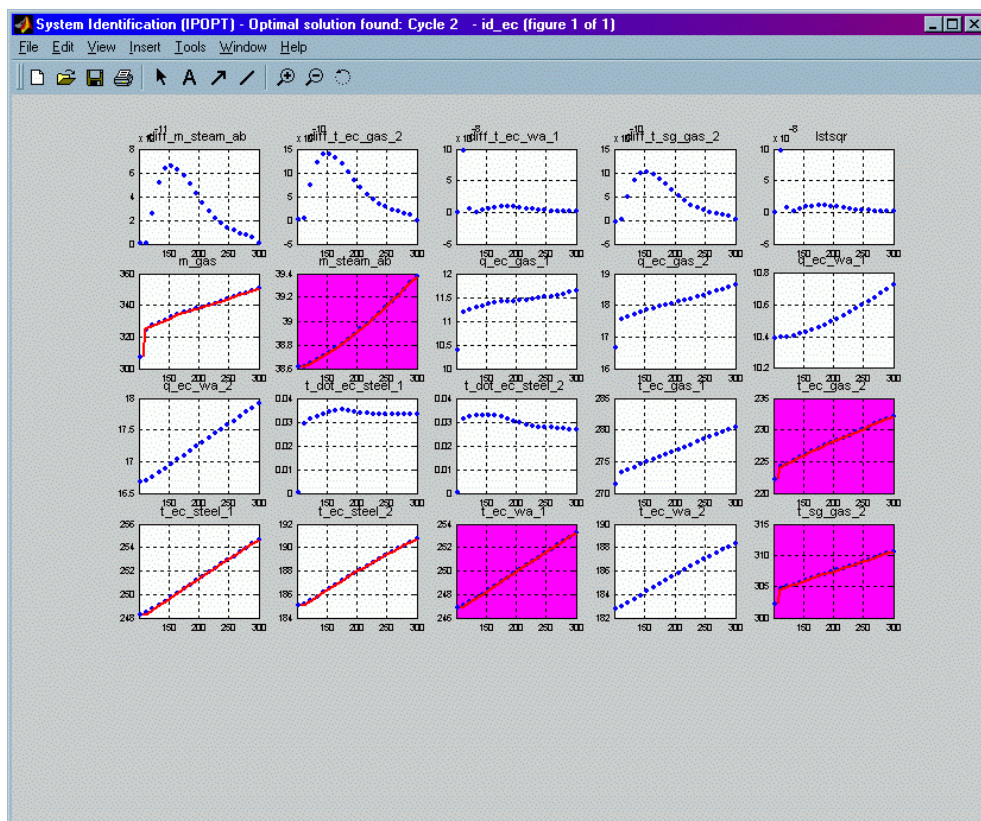


Fig. 6-6: Plot of ID results of the economizer unit

The ID results are plotted with points. The background for the four measurements, in this case variables m_{steam_ab} , $t_{ec_gas_2}$, $t_{ec_wa_1}$ and $t_{sg_gas_2}$, are color-coded for easy recognition. Simulation results have also been saved and plotted in solid lines for variables m_{gas} , $t_{ec_steel_1}$ and $t_{ec_steel_2}$.

6.7 The on-line wizard

After all the necessary on-line OCC parameters the user can invoke the On-line Wizard by clicking “Run” on the on-line menu. This wizard is designed to take the user step-by-step through specifying the run-time options.

1. Specify whether ID is to be carried out.

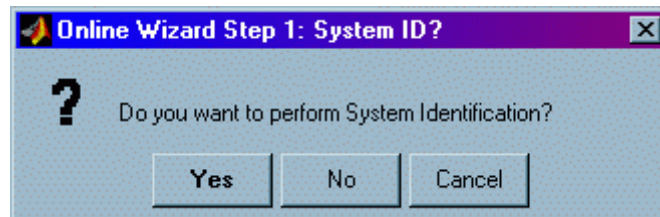


Fig. 6-7: Dialog box for selecting ID

If “Yes”, go to step 2.

If “No”, make sure <initials_opt1.dat> and <svalues_opt1.dat> are present in model root directory. Go to step 3.

2. Specify whether the ID during the first cycle is to be carried out using <steady_state.mat>.

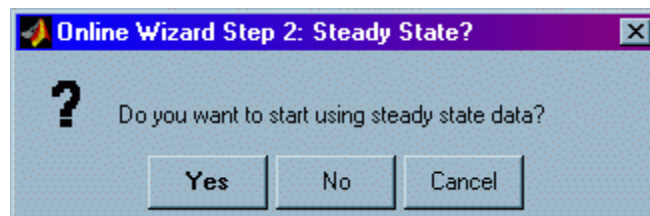


Fig. 6-8: Dialog box for selecting steady state data

If “Yes”, make sure <steady_state.mat> is present in model root directory.

If “No”, make sure <initials_opt1.dat> and <svalues_opt1.dat> are present in model root directory.

3. Specify whether various different OPT .EXE files are to be executed sequentially.

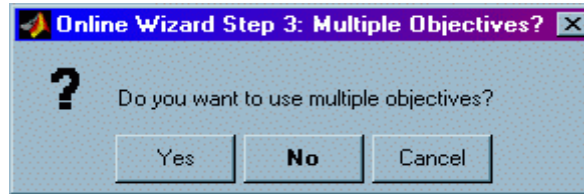


Fig. 6-9: Dialog box for selecting sequential execution of .EXE files

If “No”, go to step 4-1.

If “Yes”, go to step 4-2.

4-1. Specify one OPT .EXE file.

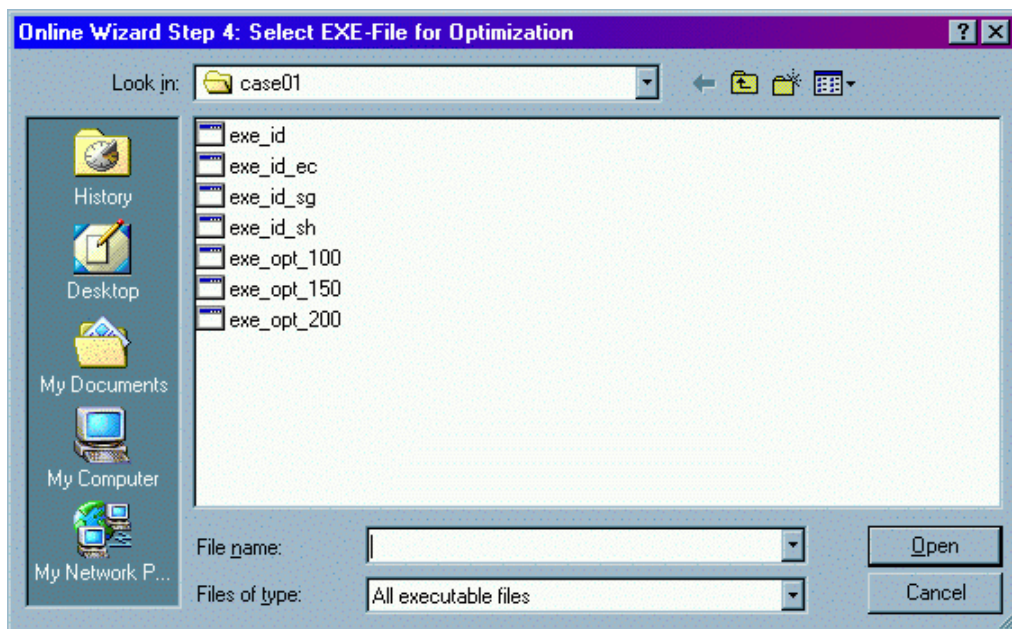


Fig. 6-10: Open file dialog for selecting a single OPT .EXE file

4-2. Specify up to five .EXE files to be carried out sequentially. The user is also required to define the switch-criteria. These criteria must be in MATLAB format, e.g. “time_sim \geq 1000” or “P_tot_sim $>$ 150”, and will be checked at the end of each on-line cycle. Fulfillment of the first criterion causes OCC to switch to the second objective, and so on.

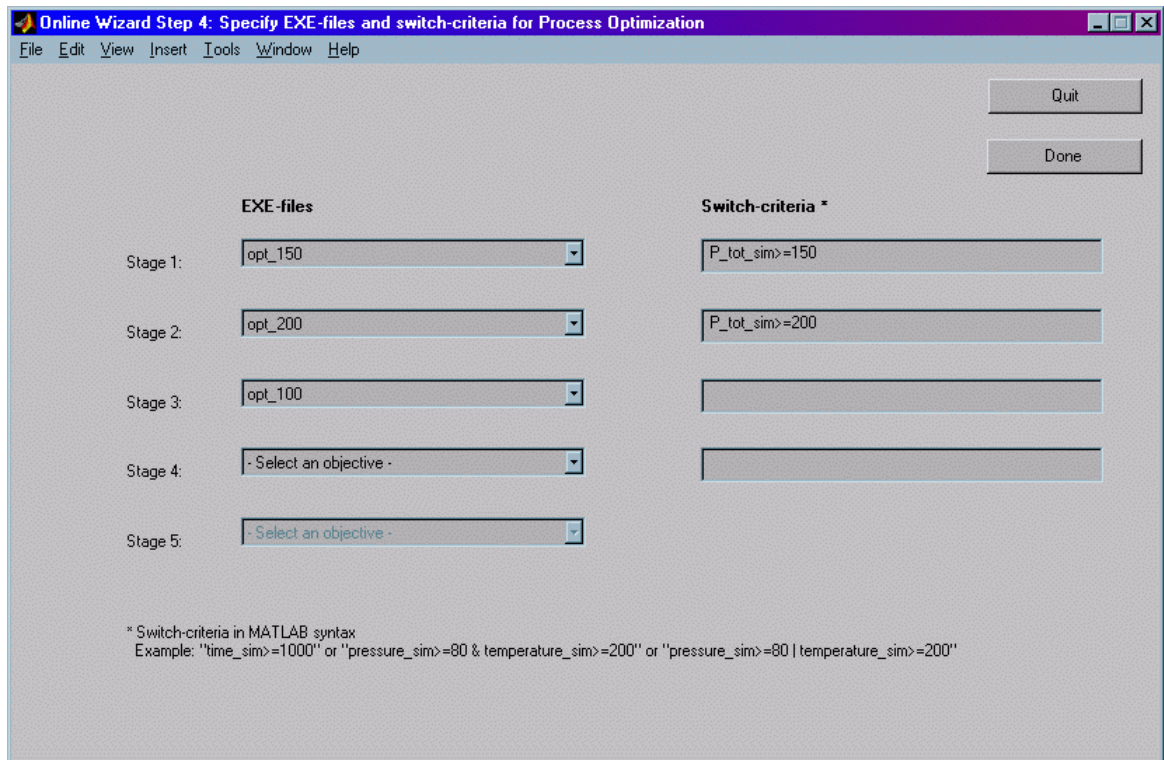


Fig. 6-11: Menu for selecting more than one OPT .EXE files

6. In the case that ID is to be carried out, the user has to select the .EXE files to be used. This is done using the ID selection menu shown in Fig. 6-12. This procedure is repeated for every OPT .EXE file selected. OCC reads the structure file of the OPT .EXE file, which was selected in step four. All the variables present in the optimization problem will be shown on the menu, with differential variables being listed first in bold font.

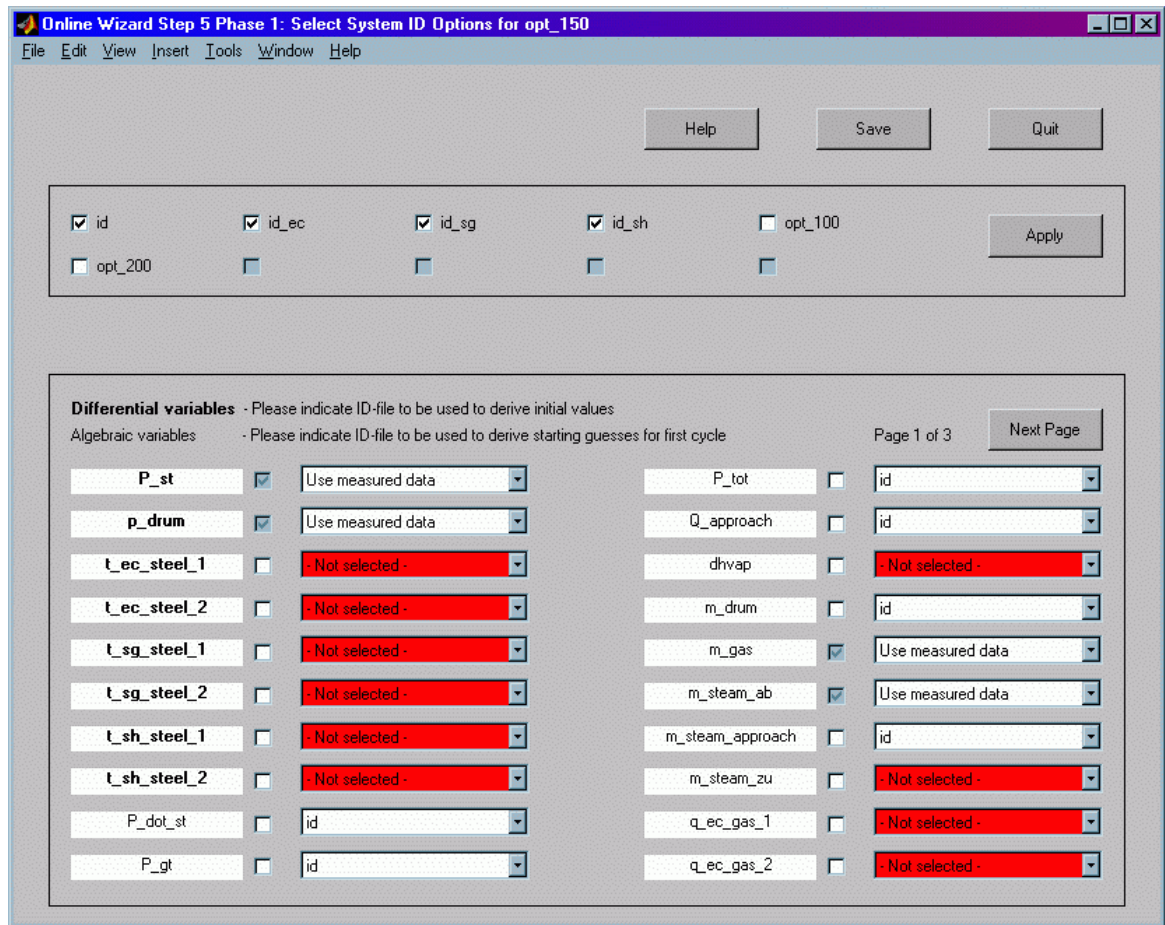


Fig. 6-12: Menu for selecting ID .EXE file(s) and options for starting guesses

Select the ID .EXE files by checking the checkbox next to the file name. Click “Apply” to update the pop-up list beside each variable. Each variable is checked against the variable lists found in the structure files of the selected ID .EXE files.

Using the pop-up lists, the user is able to select the source of information for generating the starting guesses used for the subsequent OPT step. The selection for algebraic variables is only taken into account for the first cycle.

Besides the variables already used as measurements in the ID .EXE files, the user can specify simulation data to be used as data source by checking the checkboxes next to the variable name and selecting “Use measured data” in the adjacent pop-up list.

Click “Save” after data sources for all variables have been selected.

7. Lastly, the user is required to specify the Simulink model to be used.

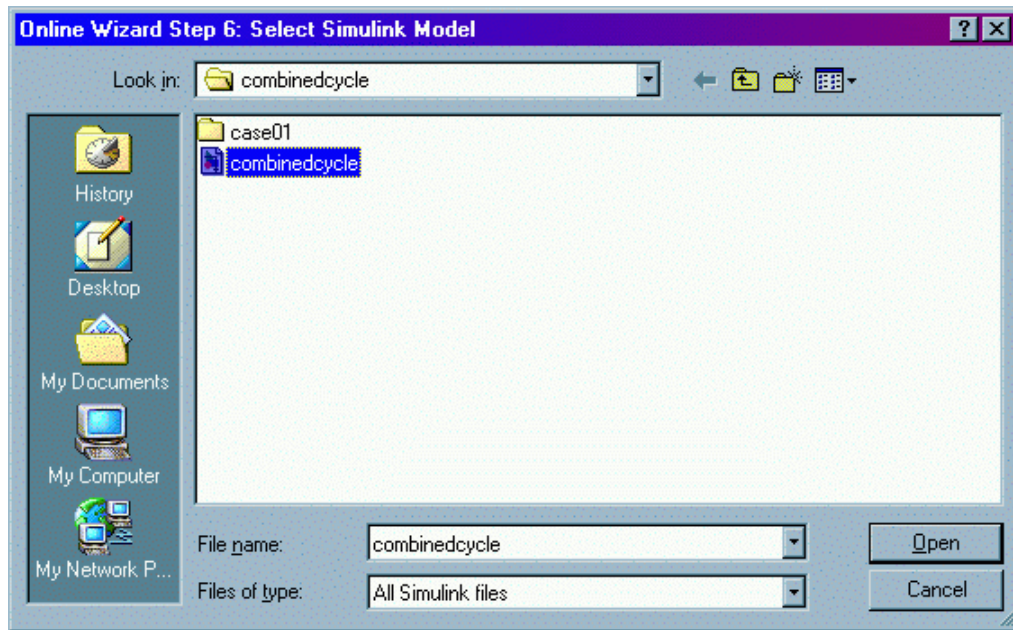


Fig. 6-13: Open file dialog for selecting Simulink model

The information gathered by the On-line Wizard is stored in the initialization files `<on-line_initialise_opt.m>` and `<on-line_initialise_id*.m>`, where * represents the optimization phase(s). These files are saved in the case directory. OCC checks for these files every time before the On-line Wizard is invoked. If they are found, OCC asks the user whether the saved options are to be reused.

6.8 Initialization of on-line mode 3

Initialization of operation mode 3, designed to obtain starting values for complex problems, is relatively simple. It involves the following steps.

1. Prepare the .EXE file for the on-line optimization as described in Section 6.3.
2. Prepare the input files `<initials.dat>` and `<svalues.dat>` for cycle one as described in Section 6.5.1.
3. Select “Sequential optimization” mode in the on-line options submenu (Fig. 6-5 item ①).
4. Specify the cycle time in the options menu submenu (Fig. 6-5 item ②).
5. Click “Run” on the on-line menu and select the prepared .EXE file from the open file dialog box.