June 1, 2021

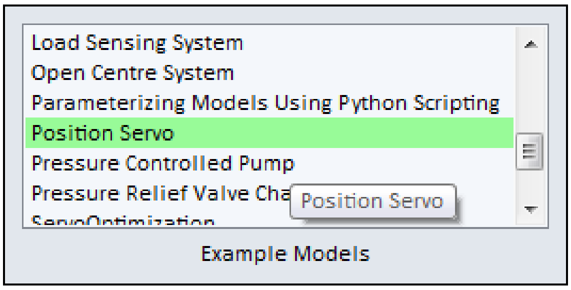
# HOPSAN

# Tutorial - Running an Optimization

## Using the optimization dialog

Numerical optimization is an important tool for automatically finding a desired parameter set for a model. Hopsan contains an optimization dialog from where optimizations can be controlled. This guide shows how to optimize control parameters for a simple hydraulic position servo using the Complex-RF algorithm. Optimization in Hopsan can either be performed from a dialog or by loading previously saved (or manually created) script files. This tutorial demonstrates both methods.

1. **Open the model**
   The *Welcome* tab has a box with example models. Open the *Position Servo* example model.




   This model describes of a hydraulic position servo, where a PI-controller is used to control the position. This consists of a proportional gain ($K_p$) and an integrator gain ($K_i$). The values of these gains will decide the speed and stability of the servo. It is thus desirable to find as good controller parameters as possible.

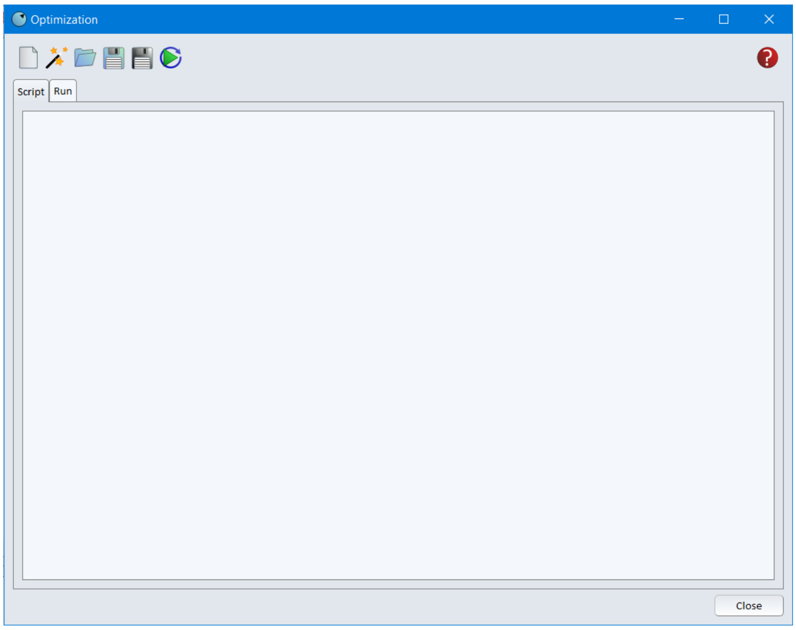
2. **Simulate**
   Simulate the model by clicking the simulate icon.

   ***Simulate current project (Ctrl-Shift-S)***

   Now double click on the "Scope" component on top of the model. Two curves will be plotted, the reference signal and the actual position. The reference signal is a step, and the position shows the corresponding step response. As you can see, it is quite oscillative. We want to adjust the control parameters to reduce its rise time, overshoot and settling time by using the *optimization* feature.

3. **Open the dialog**
   Click the optimization icon to open the dialog.

   ***Open optimization dialog (Ctrl-Shift-Z)***

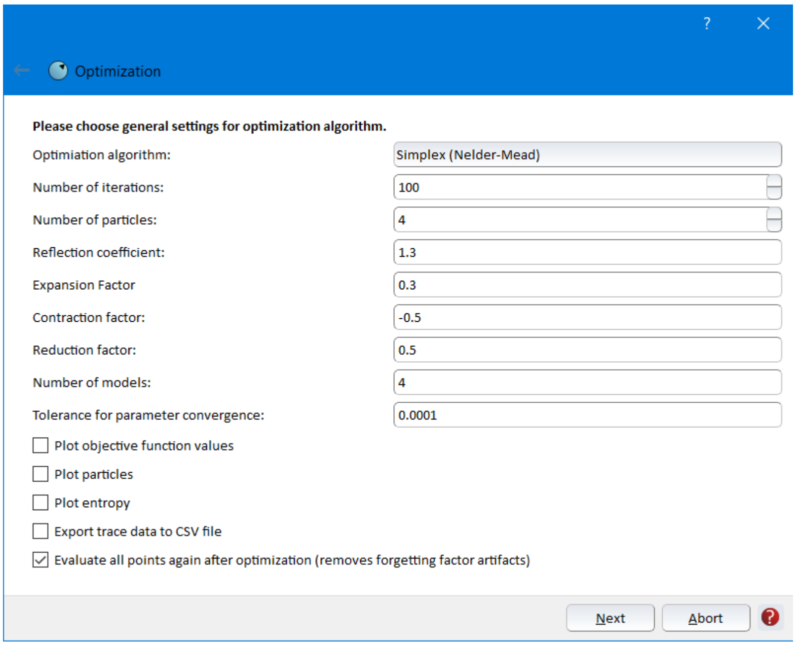



The first page should look like above. You can either write an optimization script by hand, load a script file or start the script wizard. Let us begin with the latter. Click on the wizard icon.

***Script Wizard***

## 4. Optimization settings

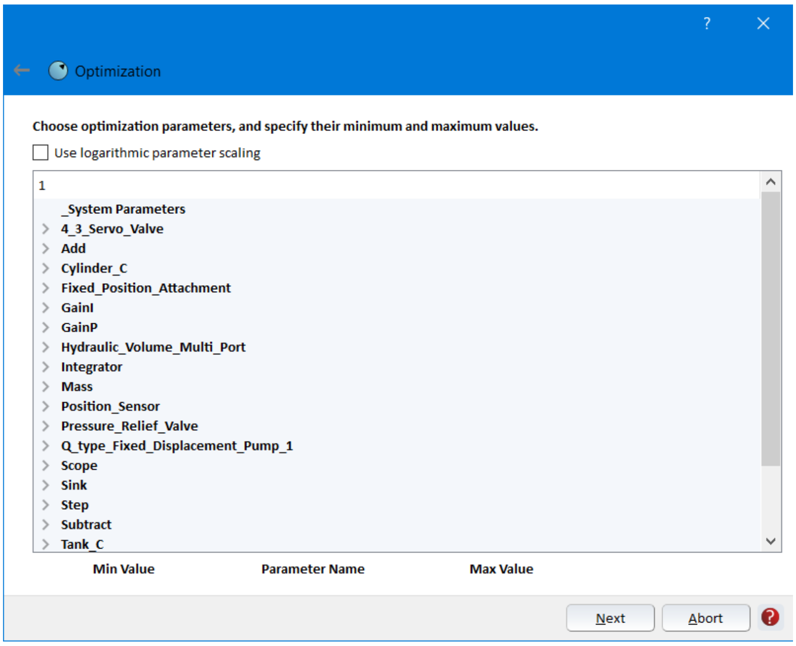
The first wizard page will open. From here you can choose the general optimization settings.




- Set *Algorithm* to *Complex-RF*
- Set *Number of iterations* to 1000
- Set *Number of points* to 4

Leave all other settings unchanged for now.

5. **Choose optimization parameters**
   Next page controls the parameters in the model that shall be optimized.




   Check the following parameters in the list:

   - GainI → k#Value
   - GainP → k#Value

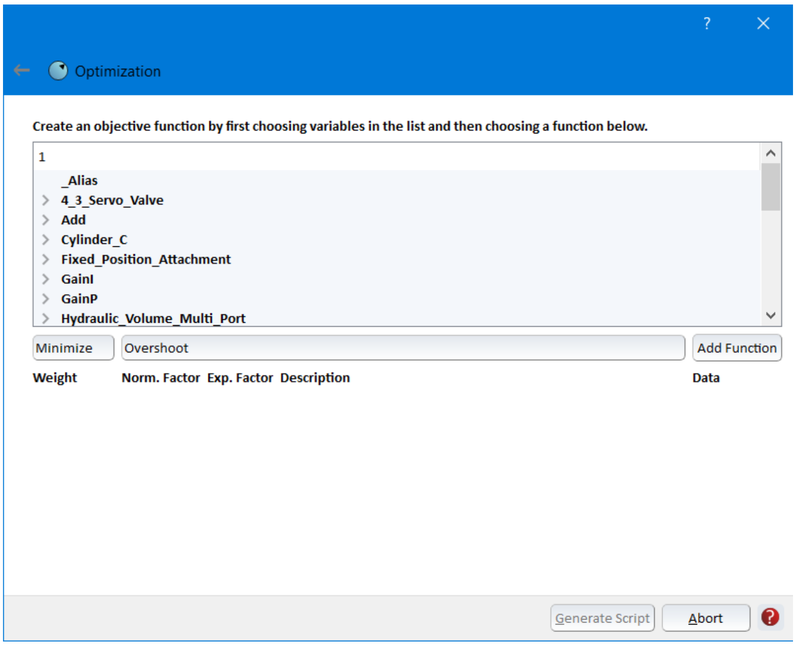
   Then give them the following min and max values:

| **Parameter** | **Min** | **Max** |
|---|---|---|
| GainI, k#Value | 0 | 0.001 |
| GainP, k#Value | 0 | 0.04 |

6. **Choose objective functions**
   The optimization algorithms use objective functions to calculate how good a certain point in the parameter space is. From the third page this can be chosen. In this model we want the actual position to be as close as possible to the reference signal. To do this, check the following variables in the list:

   - Position_Sensor → **out** → Value
   - Step → **out** → Value

   Then choose "Minimize" and "Absolute Average Difference" in the boxes. Click "Add Function".

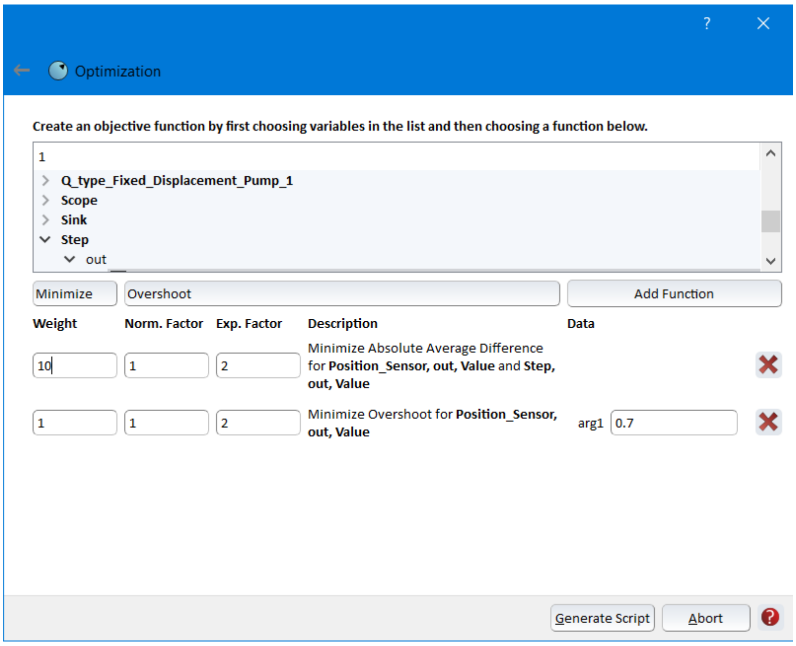



Apart from minimizing the average difference, we also want to minimize the overshoot of the step response.

- Uncheck **Step** → **out** → Value

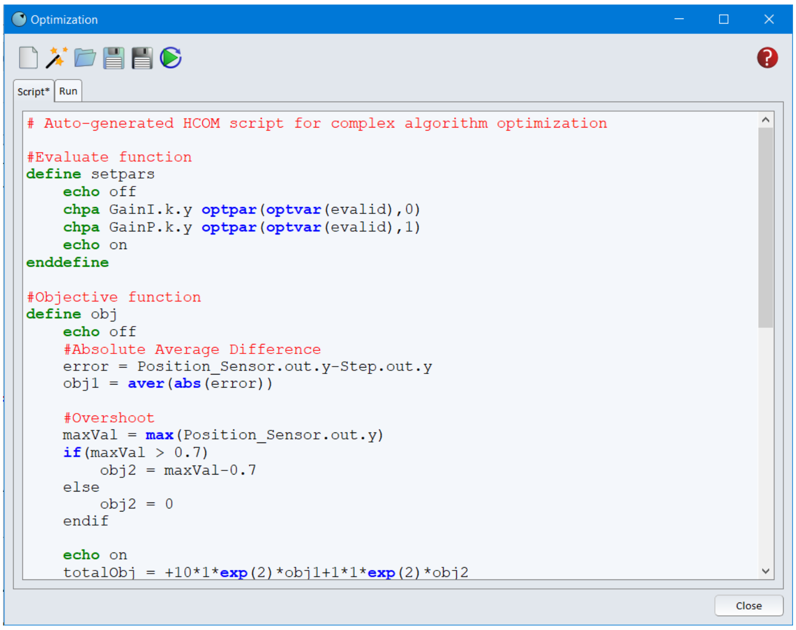
Now choose "Minimize" and "Overshoot", and then click "Add Function". In the box called "arg1", change the value to 0.7.

Now we have added a second objective function. The total objective function will the sum of these. As they may not be equally important or equally scaled, it is possible to weight the individual functions. Change "Weight" for the first function to 10. Now the page should look like this:

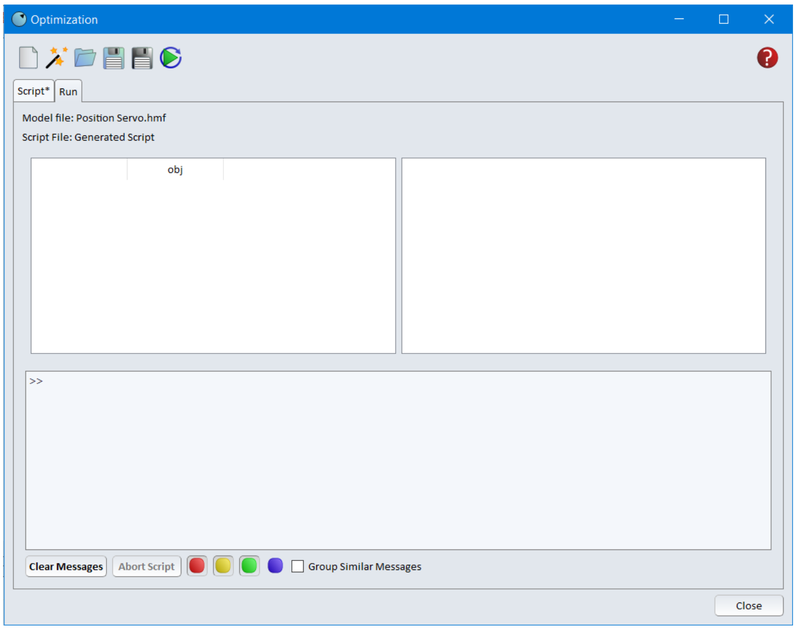


### 7. Start the optimization

Next page shows the generated script code. Modifications can be made if necessary, but leave it as it is for now.



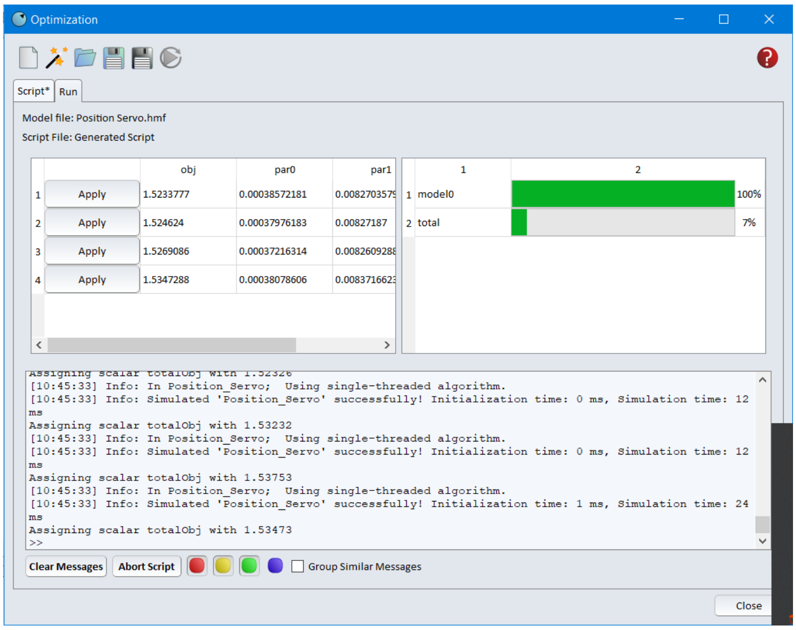

Now change to the "run" tab. From here the execution of the optimization can be monitored:




Click on "Start Optimization".

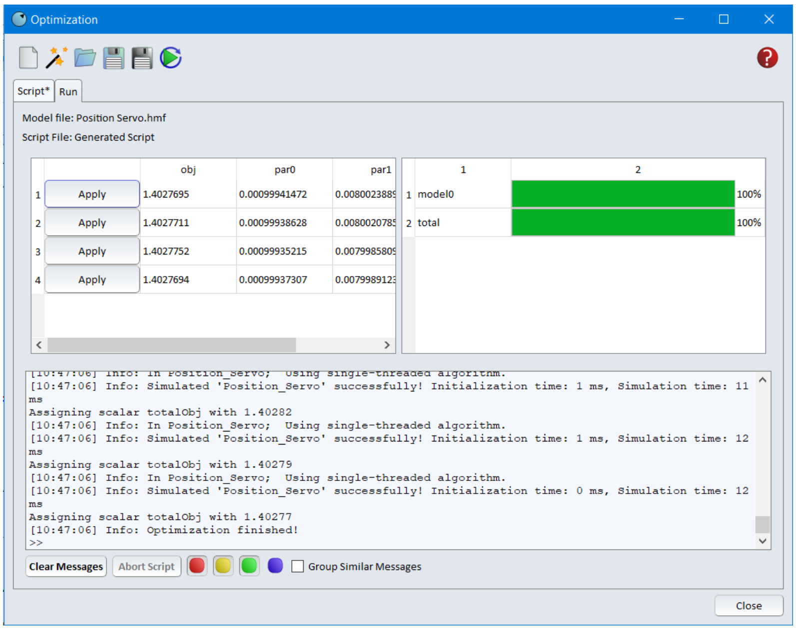
***Start Optimization***

During the optimization the objective function values and the values of the optimization parameters are shown for the different points. The rows are sorted from best to worst.




## 8. Evaluate the results

Once the optimization has finished, we can evaluate the results. Parameters from a point can be written back to the original model by using the "Apply" buttons.




Click the "Apply" button for the point with green text (the best point). Then go back to the original model and click the Simulate button again.

***Simulate current project (Ctrl-Shift-S)***

Now plot the step response by double clicking the scope component on top of the model. If a desirable result has not been found, it can be necessary to change objective functions or parameters and re-run the optimization.

## Optimizing from script files

It is often necessary to re-use the settings from the optimization more than once. This can be done by saving the optimization scripts to a file, and loading them again later.

1. **Follow steps 1-7 above**
2. **Saving a script file**
   Click on Save script file in the dialog.
3. **Loading a script file**
   To load a previously saved script file, click on Load script file in the dialog. You will be taken directly to the code tab. Go to the "run" tab and start the optimization as before. Note that the settings on the first three pages are not updated by the script. Running the wizard again will re-generate the script and overwrite the loaded code.
4. **Loading a script file from command terminal**
   It is also possible to load a script file from the command terminal. Just write “exec” followed by the path to the file. Hopsan will automatically understand that this is an optimization script and start the dialog. You can use “pwd” to show current directory and “cd” to change directory if necessary. Example: `>> exec ..\folder\myscript.hcom`