# User Guide

## Purpose

The NOAH RTC Tool is used to find the optimal setting in an RTC setup.

The tool uses a complete SWMM model of a drainage system and evaluates the best setting of a setup with a sensor located downstream of an actuator (an orifice) that is controlled by the setting of the sensor.

## Installation

To use the NOAH RTC Tool the following is needed:

* Installation of Anaconda3 (including Python, Spyder and Jupyter Notebook).
* Installation of SWMM
* A cloned version of the repository from Gitlab (<https://gitlab.gbar.dtu.dk/jowi/NOAH.git>) from branch *magnus*. (Use command: *git branch --set-uptream=origin/magnus* to pull from the correct branch.)

The directory structure needs to be similar to the one in the *\NOAH\magnus* directory on gitlab. If a clone of the repository is done this should be sufficient.

Create a Conda environment named NOAH. See *readme.md* for what this should include.   
This contains all the packages necessary to run the scripts needed.

In order to run the tool, execute the *pyswmm\_GUI.py* script from Spyder. Other Python IDE’s might work as well but the software used for the tests was Spyder.

## Input

The RTC Tool requires a complete and executable SWMM model of a drainage system. This, including any external files that is used, should be in the *\model* directory. External files (such as rain series) should also be in the *\lib\interface* directory.   
The model does not need any control setup in the SWMM file, but it requires that an orifice is created at the locations that are to be tested.

Also, a configuration file with the same name as the SMWW model is required in the *\config\saved\_configs* directory. This is created automatically by the tool if the “Overwrite existing configuration file” is checked.

## Output

When the “Run” button is clicked a directory with the timestamp (in format date\_time e.g. *2019-11-20\_10-16-34*) is created in the *\output* directory that contains all the results from the simulation. This includes the following:

* An .out file that is like the one that is created by SWMM. This can be opened using python (see below).
* An .rpt file that is like the one that is created by SWMM. This can be opened by any text editor.

If optimization is used the following is also created:

* A text file containing the optimal solution found by the optimizer.
* A .pickle file that contains the results of the first part of the optimization. This contains a Pandas DataFrame with the starting points and objective values of the simulations. This can be opened using Python (see below).

The .out and the .rpt file that are created are the ones that come from the last simulation. If enough simulations are performed this is also the optimal RTC setup, however this is not guaranteed if the maximum number of iterations has been exceeded.

To open the two files that requires Python, a Jupyter Notebook (“*Open\_result\_files.ipynb*”) that shows how this is done is found in *\lib* directory.

A *\saved\_output* directory is provided for the user to save outputs that should be saved for later.

## Interface

In the user interface the following elements:

* Selection of SWMM model
* Four tabs with different input parameters.
  + RTC setup – Specify the location of the sensor and actuator (gate) and setting if only one simulation is run.
  + Rain series – empty
  + Control objective – specify the objective of the optimization and up to three nodes where CSO should be computed (and minimized).
  + RTC Optimization – Activate/deactivate optimization and specify parameters that are needed for the optimization.
* Checkmark to tell whether a new config file should be written or the existing one is to be used.   
  All fields must be filled if a new configuration file is written (except if less than 3 CSO’s are used, then leave those blank)
* Buttons
  + Run: Run one simulation or optimization depending on the settings.
  + Save config: Save the configuration file without running the simulation.
  + Write to SWMM file: Saves a SWMM file with the settings specified in the “RTC setup” tab.
  + Exit: Closes the program.

## Optimization

The optimization is calculated in two steps:

* An initial “screening” where the parameter range specified in the configuration file is divided into even intervals. The objective values are evaluated for each simulation.
* The point with the lowest objective value is used as the starting point for the actual optimization that runs until it has found the minimum value or until the maximum number of iterations have been met.

The optimization can aim to minimize either the volume or the number of CSO events from the specified nodes. Both are read from the .out files.   
Note that depending on whether the SWMM model has CSO’s as simply flooding from certain nodes or as outlets some code needs to be changed. (See script *pyswmm\_Simulation.py* in functions *count\_CSO\_volume* and *count\_CSO\_events*)

The .pickle file that is saved in the *\output\timestamp* directory can be used to inspect the results of the first step of the optimization. It allows the user to get an overview of the objective values over the entire parameter range and thus give a qualitative estimation of whether the optimized result is correct.

Total CSO volume is computed by multiplying the flow be the reporting time step (default 5 seconds). This can be changed if the reporting step of the SWMM file is changed.

CSO events are computed by counting every timestep where there is a CSO. If these are within a certain range (default 24 hours) these are counted as one CSO event.   
If an event lasts longer than a maximum allowed duration (default 24 hours), this event is counted as the duration divided by the allowed duration. This is done to ensure that unrealistic setups are avoided if e.g. a gate is closed all the time thus resulting in one constant CSO over the entire simulation time.

## Other remarks

At the moment the GUI says “not responding” while the simulation is running. The simulation is still running in the background so be patient. And better user feedback during the simulations is to be implemented.

This is the very first version (a draft) of the actual tool as well as this manual. Many improvements are to be done but please give feedback on what works well and what can be improved.