Short-cut adsorption chiller model
how to generate the figures Semester Project

presented by

Hibiki Kimura

StudentID no. 21-908-090

Zurich, April 1, 2022

Contents

Chapter 1

Manual of calculation

1.1 Make Dynamic simulation

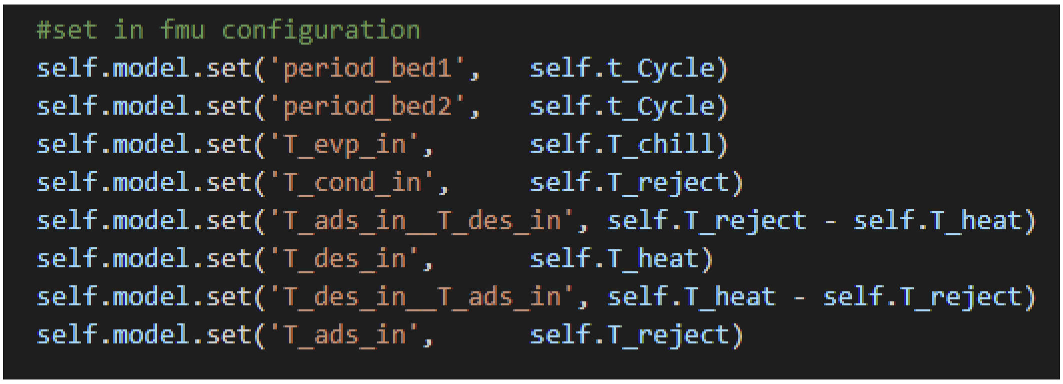
Install the dynamic simulation package, Sorplib from Chair of Technical Thermodynamics. You can download from here:
`https://git.rwth-aachen.de/ltt/SorpLib/-/tree/SorpLib_v2`

Note: Branch 'Sorplib v2' should be used.
You need Modelica license for the simulation.

How to open the Sorplib.

1. Drop the .mo file of this package into DYMOLA.
2. Open 'Applications' -> 'Adsorption Chiller' -> 'ModelicaMediaLibrary'.
3. Copy the two bed system adsorption chiller model and paste on your new file.

How to propagate FMU file.
You must be able to edit the parameters of FMU file by python command. You need to mark the parameters that you want to edit later by naming in DYMOLA. You click the icon of the component and find some parameters of it. You right click the icon and propagate. You can name the parameter as you want. I named the parameters as you can see below.



```
#set in fmu configuration
self.model.set('period_bed1',   self.t_Cycle)
self.model.set('period_bed2',   self.t_Cycle)
self.model.set('T_evp_in',      self.T_chill)
self.model.set('T_cond_in',     self.T_reject)
self.model.set('T_ads_in__T_des_in', self.T_reject - self.T_heat)
self.model.set('T_des_in',      self.T_heat)
self.model.set('T_des_in__T_ads_in', self.T_heat - self.T_reject)
self.model.set('T_ads_in',      self.T_reject)
```

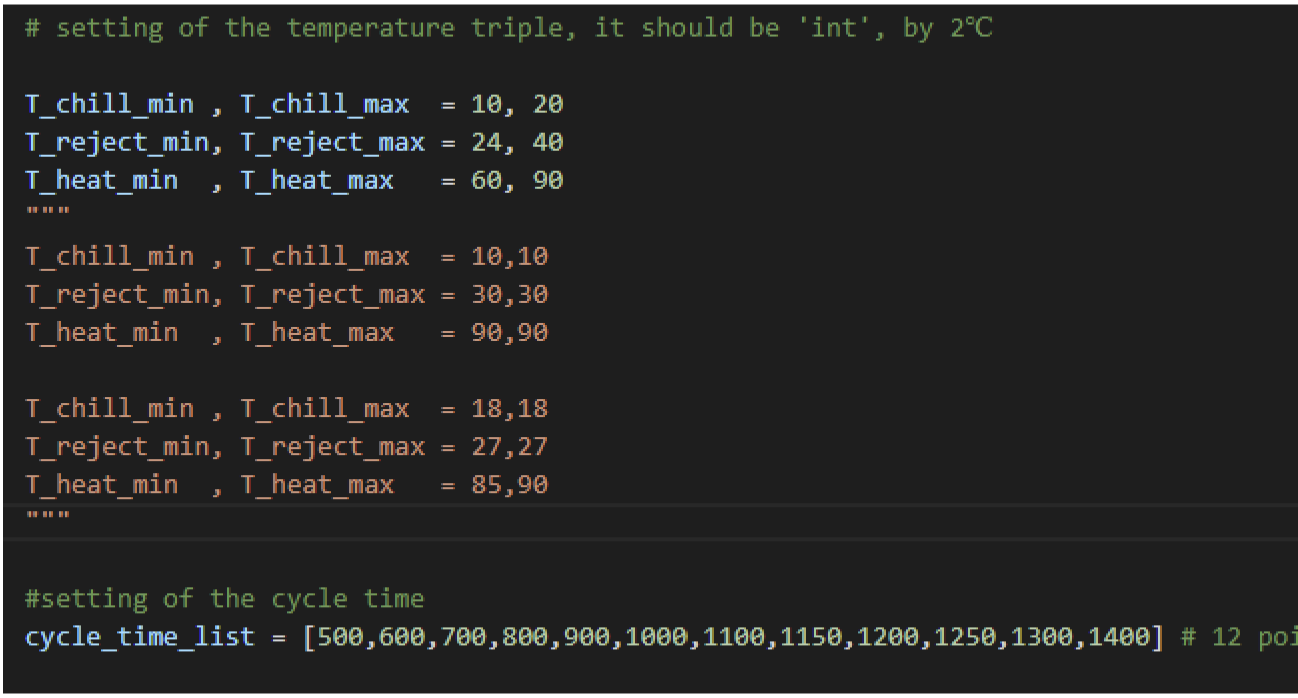
Figure 1.1: Naming of the parameters. (./FMU/run_fmu.py)

First two parameters are the cycle time of the chiller. You can edit form the temperature swing box. The third and forth parameters are the temperature of the evaporator and the condenser. The last four parameters are the amplitude of the temperature swing at adsorption beds. Note that the temperature swing is defined from the bottom temperature and the variation. One of them has positive variation and the other has negative variation.

After you confirm the simulation can be done without error, you can generate the fmu file from a function, 'Translate'. You can find the FMU files at './FMU/FMU/Silica123_water.fmu' and './FMU/FMU/AQSOA_water.fmu'.

1.2 Run Dynamic simulation

You bring the fmu file that you made into './FMU/FMU'. The temperature range and cycle time can be edited in the main() function of 'dynamic_ac.py'. You can define the minimum and maximum temperature of each component. Here, you have to adjust the cycle time so that the operating simulation is Pareto frontier.



```
# setting of the temperature triple, it should be 'int', by 2℃

T_chill_min , T_chill_max  = 10, 20
T_reject_min, T_reject_max = 24, 40
T_heat_min  , T_heat_max   = 60, 90
"""
T_chill_min , T_chill_max  = 10,10
T_reject_min, T_reject_max = 30,30
T_heat_min  , T_heat_max   = 90,90

T_chill_min , T_chill_max  = 18,18
T_reject_min, T_reject_max = 27,27
T_heat_min  , T_heat_max   = 85,90
"""

#setting of the cycle time
cycle_time_list = [500,600,700,800,900,1000,1100,1150,1200,1250,1300,1400] # 12 poi
```

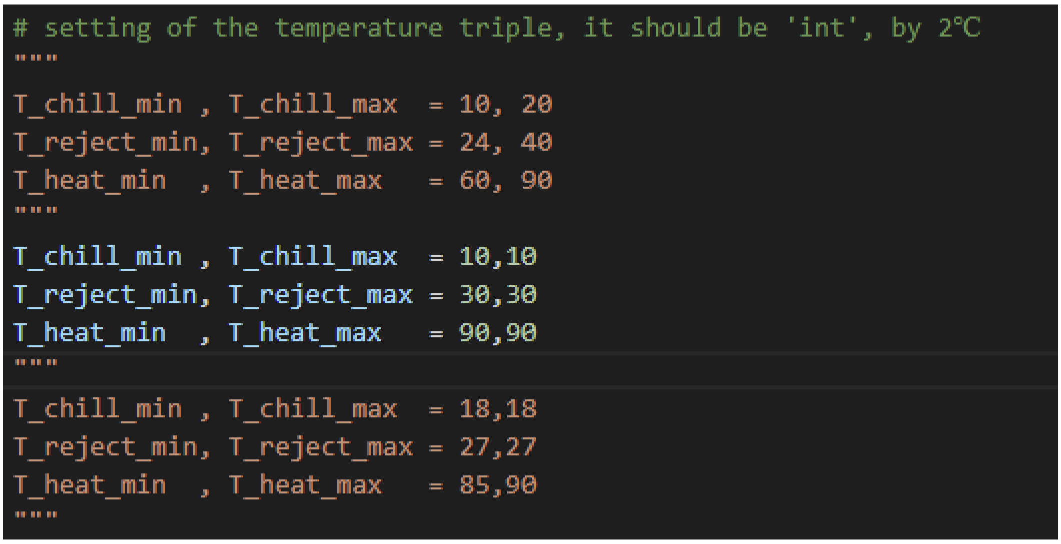
Figure 1.2: Temperature and cycle time range of the dynamic simulation

You set the name of the fmu file. You can also adjust how many raps you want to simulate. In my experience, you need more than 40 raps to get equilibrium of the COP. You can check it on DYMOLA software.

You can find the results in Results folder in the form of csv and pickle. If the simulation stop in the middle (it sometimes happen due to the license problem, make sure that you connect to the server while you’re running DYMOLA python simulation), you can restart from there. You can check how many simulation you’ve finished by seeing the result file. You edit the parameter ’restart’ in main(), which is the number of the cases that you finished so far. The name of the results file is "Results/saving"+self.file_name.replace(".fmu","").replace("./FMU/","")+".csv".

1.3 Extract dynamic simulation of two temperature triples

Make the two more lines of the dynamic simulation as a sample data for fitting. You edit the temperature setting in the main() in ’dynamic_ac.py’ and run the simulation again. You can get the simulation results with ’./Results/ .pickle’



```
# setting of the temperature triple, it should be 'int', by 2℃
"""
T_chill_min , T_chill_max  = 10, 20
T_reject_min, T_reject_max = 24, 40
T_heat_min  , T_heat_max   = 60, 90
"""
T_chill_min , T_chill_max  = 10,10
T_reject_min, T_reject_max = 30,30
T_heat_min  , T_heat_max   = 90,90
"""
T_chill_min , T_chill_max  = 18,18
T_reject_min, T_reject_max = 27,27
T_heat_min  , T_heat_max   = 85,90
"""
```

Figure 1.3: temp setting for lines

It's better to choose temperature triple to make the performance results of two simulation separate. One of the temperature setting should have high COP and Qflow, and the other should have lower performance.

1.4 Fitting of short-cut model

Now, you have a sample data of dynamic simulation and ready for the fitting.

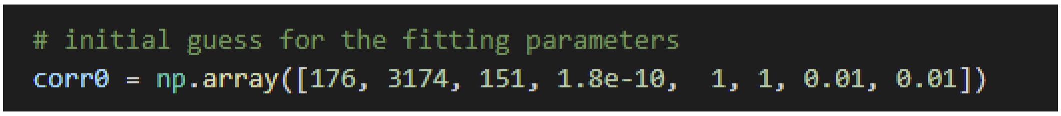
1.4.1 One temperature triple fitting

Here, you make the fitting performance map of the one temperature fitting. Make sure that the dynamic file that you're using is the one that you made in the last section.

```
python fitModel_one_temp.py
```

You need to run this fitting several time with changing the initial guess. I recommend you to mainly adjust the $C_{sor}andC_{HX}$ because they are the original coefficient for the short-cut model. When you run the second fitting, you should stick on to the original heat conductance rate and diffusion coefficient (the same value as dynamic simulation).

Finally you'll get these parameters and diagram of fitting. The performance map will be stored in the form of "Fig/fitting_silica_one_temp.eps".



```
# initial guess for the fitting parameters
corr0 = np.array([176, 3174, 151, 1.8e-10,  1, 1, 0.01, 0.01])
```

Figure 1.4: initial guess for the fitting.

Table 1.1: Fitting parameters of one temperature triple.

	fitting parameter	dyn parameter
$A\alpha_{\text{evp}}$ in W/K	69.24	176
$A\alpha_{\text{cond}}$ in W/K	2854.02	3174
$A\alpha_{\text{ads}}$ in W/K	424.69	151
D_{eff} in m^2/s	6.550 e-10	1.8e-10
C_1^{sor}	0.597	–
C_2^{sor}	3.15e-03	–
C_1^{HX}	0.285	–
C_2^{HX}	5.54e-02	–

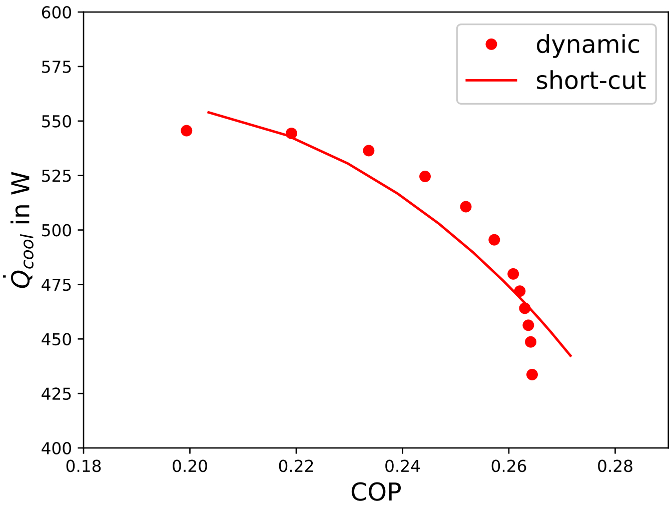
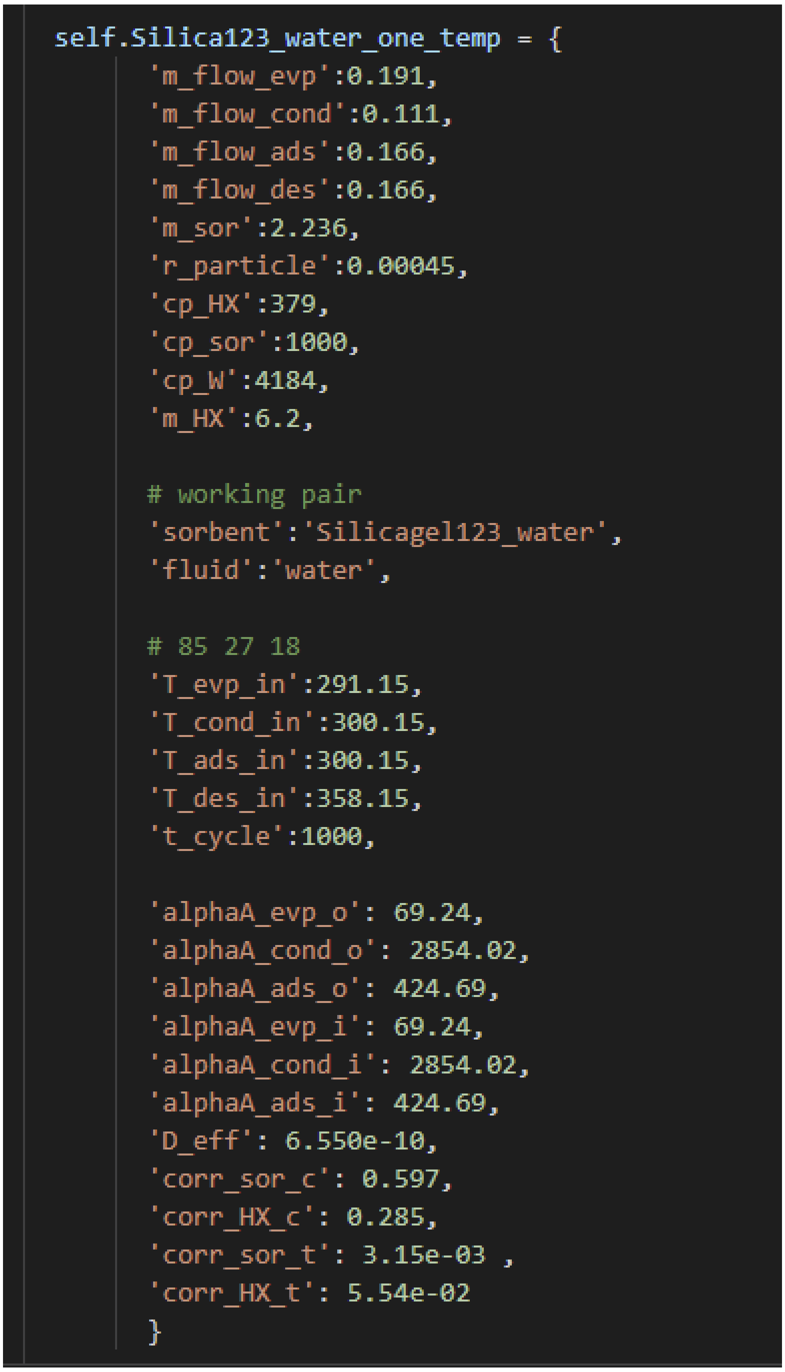


Figure 1.5: The fitting of the performance map of a temperature triple

The fitting parameters are stored in 'param_database.py'. You need to use this data at line 70 of 'generate_multiopint.py'.

You run the short-cut model simulation and you'll get the results in the form of 'Results/Silica_water_stat.csv'.

```
python generate_multipoint.py
```



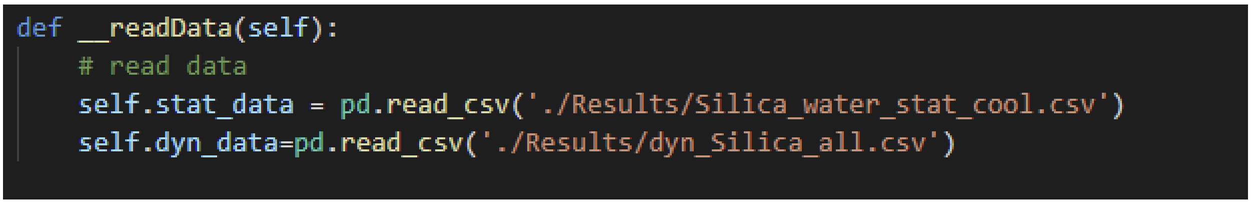
```
self.Silica123_water_one_temp = {
        'm_flow_evp':0.191,
        'm_flow_cond':0.111,
        'm_flow_ads':0.166,
        'm_flow_des':0.166,
        'm_sor':2.236,
        'r_particle':0.00045,
        'cp_HX':379,
        'cp_sor':1000,
        'cp_W':4184,
        'm_HX':6.2,

        # working pair
        'sorbent':'Silicagel123_water',
        'fluid':'water',

        # 85 27 18
        'T_evp_in':291.15,
        'T_cond_in':300.15,
        'T_ads_in':300.15,
        'T_des_in':358.15,
        't_cycle':1000,

        'alphaA_evp_o': 69.24,
        'alphaA_cond_o': 2854.02,
        'alphaA_ads_o': 424.69,
        'alphaA_evp_i': 69.24,
        'alphaA_cond_i': 2854.02,
        'alphaA_ads_i': 424.69,
        'D_eff': 6.550e-10,
        'corr_sor_c': 0.597,
        'corr_HX_c': 0.285,
        'corr_sor_t': 3.15e-03 ,
        'corr_HX_t': 5.54e-02
        }
```

Figure 1.6: One temp parameters

Now, you have both data, short-cut model and dynamic model. Then, you can compare the results with this command.

```
python validation.py
```

Make sure that the files that you're using are the one that you made.



```
def __readData(self):
    # read data
    self.stat_data = pd.read_csv('./Results/Silica_water_stat_cool.csv')
    self.dyn_data=pd.read_csv('./Results/dyn_Silica_all.csv')
```

Figure 1.7: Caption

Then, you'll get these graphs.

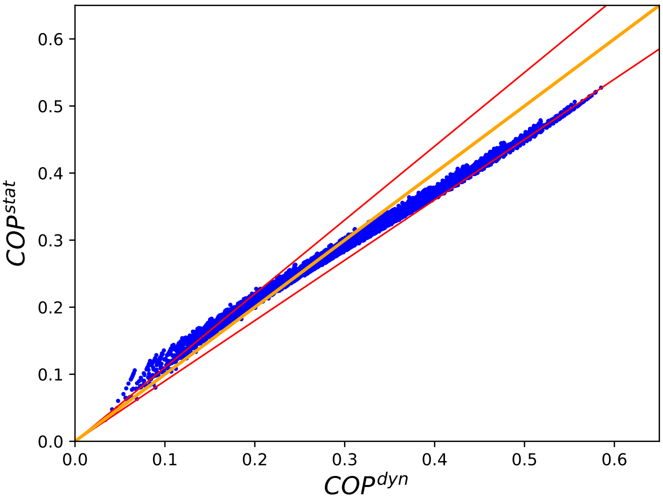


Figure 1.8: The COP calculated in the short-cut model and dynamic model. The short-cut model has fitting parameters of one temperature triple.

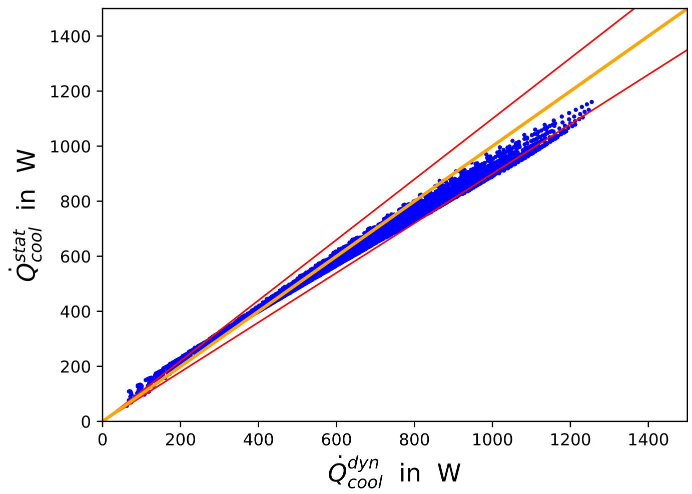


Figure 1.9: Cooling capacity calculated in the short-cut model and dynamic model. The short-cut model has fitting parameters of one temperature triple.

1.4.2 two temperature fitting

It's same procedure as above. Use fitModel.py for the fitting.

1.4.3 without time dependent

It's same procedure as above. Use fitModel_without_dt.py for the fitting.