Reporting of First Results

IBPSA Project 1 Expert Meeting Montreal (Virtual)

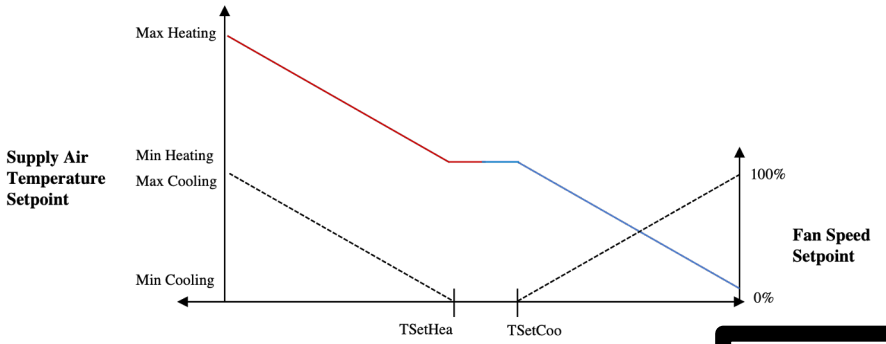
10/13/2020

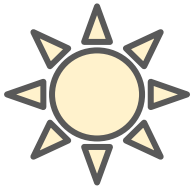
David Blum
Building Technology and Urban Systems
Lawrence Berkeley National Laboratory
dhblum@lbl.gov

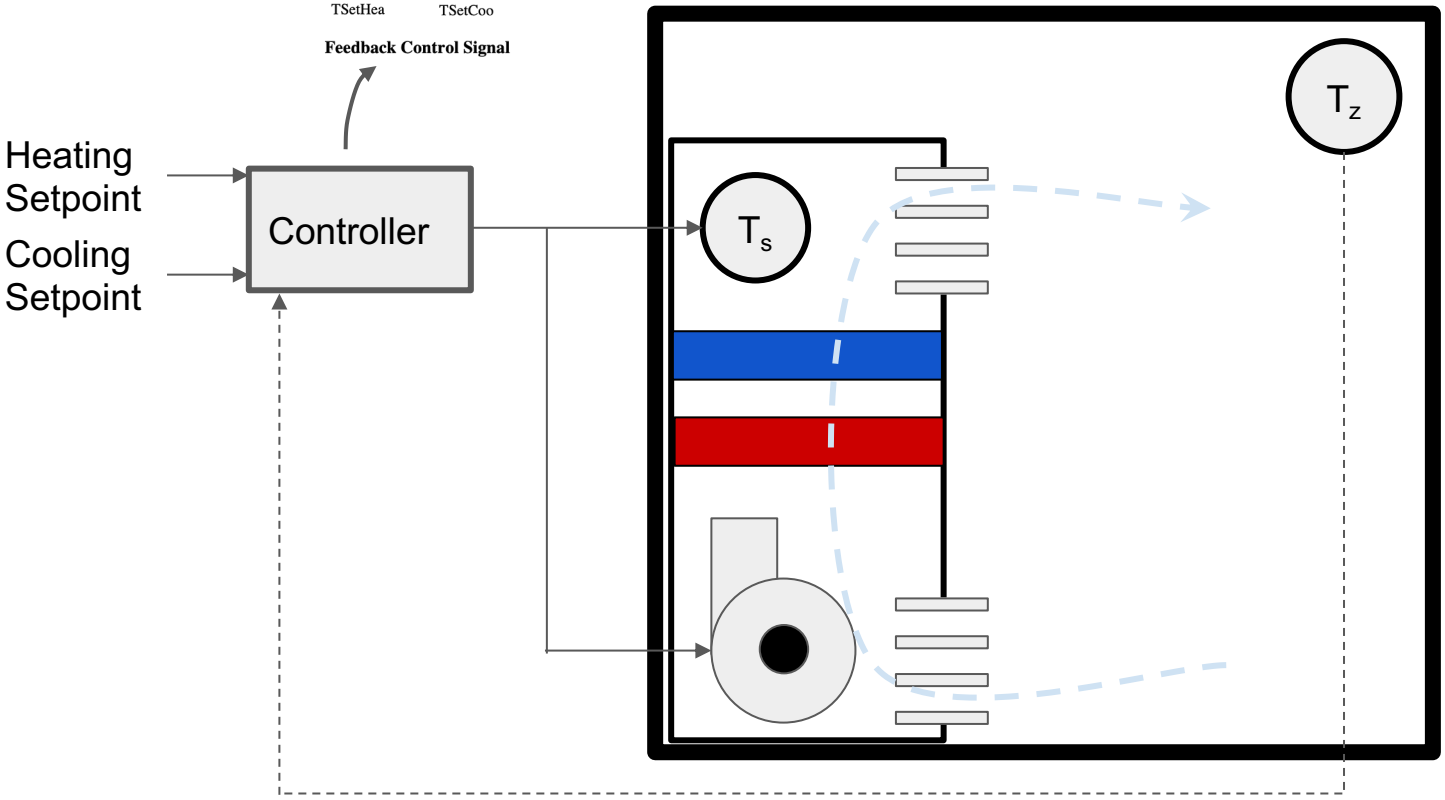
Test Case

BESTEST Air

- BESTEST envelope + ideal FCU heating and cooling







MPC Summary

MPC Controller Overview

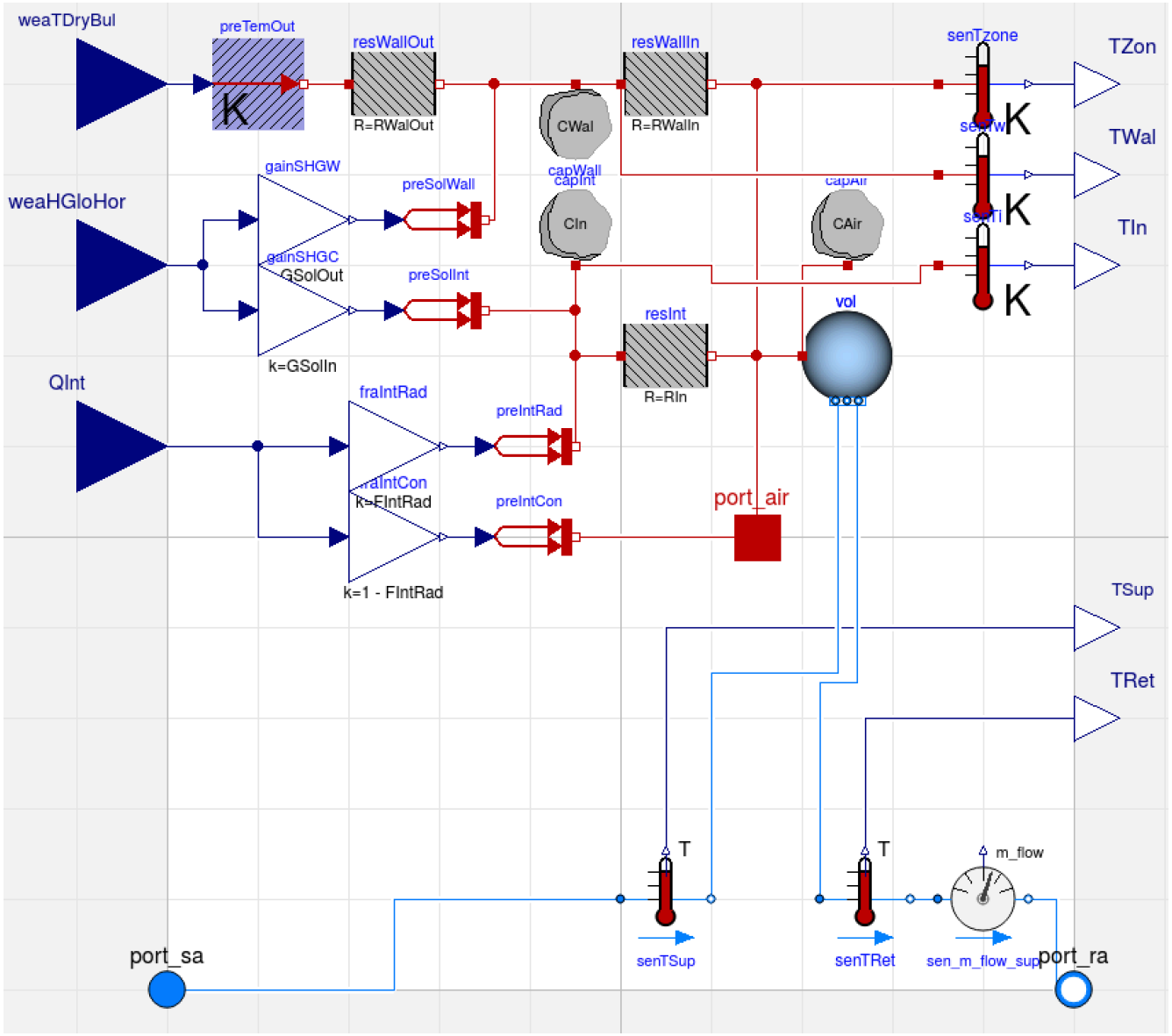
- Controller Implementation: MPCPy (Blum and Wetter 2017)
- Models:
  - Envelope: R3C3
  - Fan: $P_{fan} = a*V^3 + b*V^2 + c*V + d$
  - Heating , Cooling: $P_{hea} = Q_{hea}/eff$, $P_{coo} = Q_{coo}/COP$
- Parameter Estimation: Least squares optimization
- State Estimation: Moving horizon
- Control Optimization: Minimize $(E_{fan} + E_{hea} + E_{coo})$

  s.t.

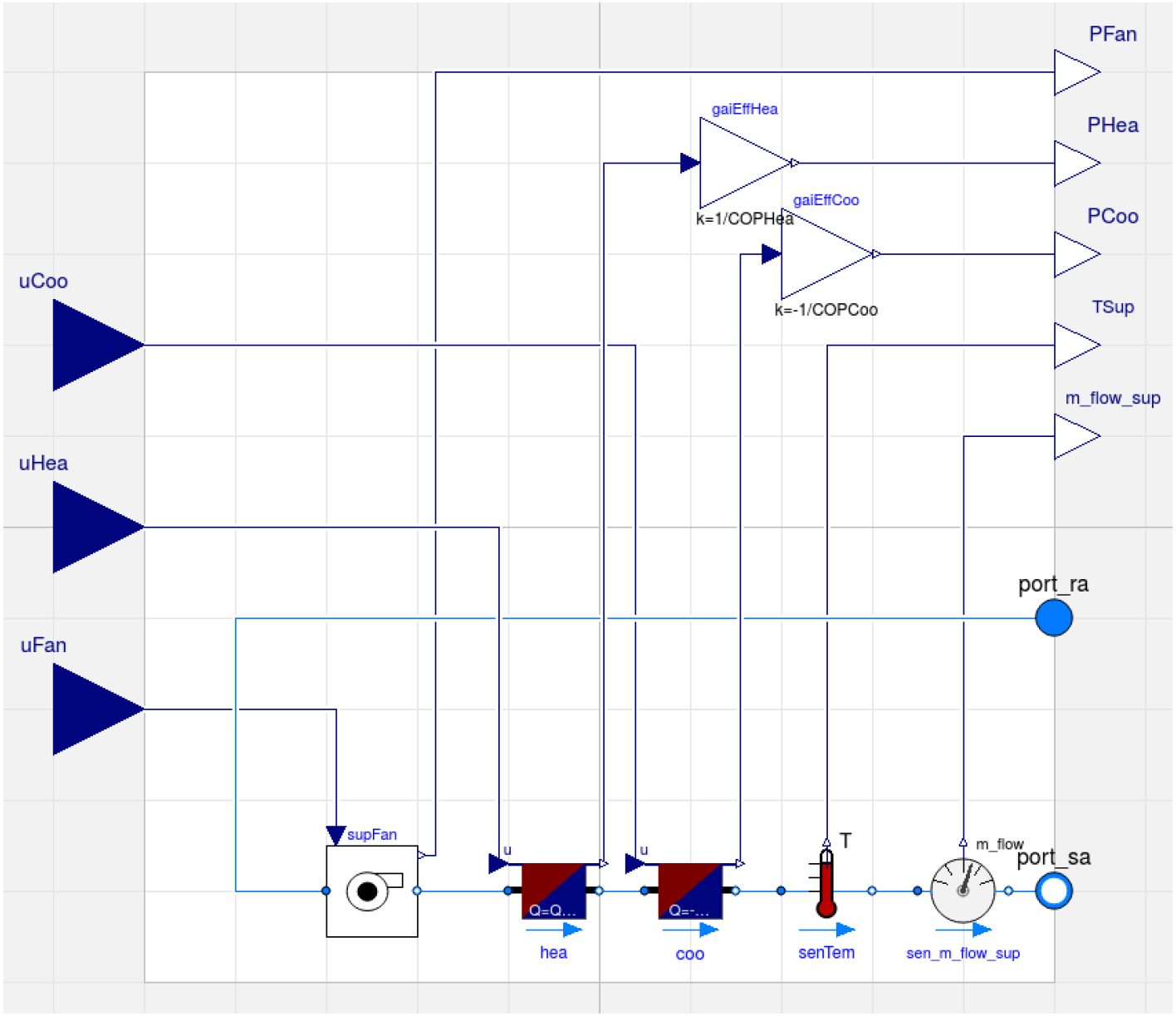
  $T_{Min} \leq T_{Zon} \leq T_{Max}$

  $Q_{Min} \leq (Q_{coo} \text{ or } Q_{hea}) \leq Q_{Max}$
- Horizon: 6 hours
- Control step: 10 minutes
- System control signals: Fan Speed, SAT Setpoint

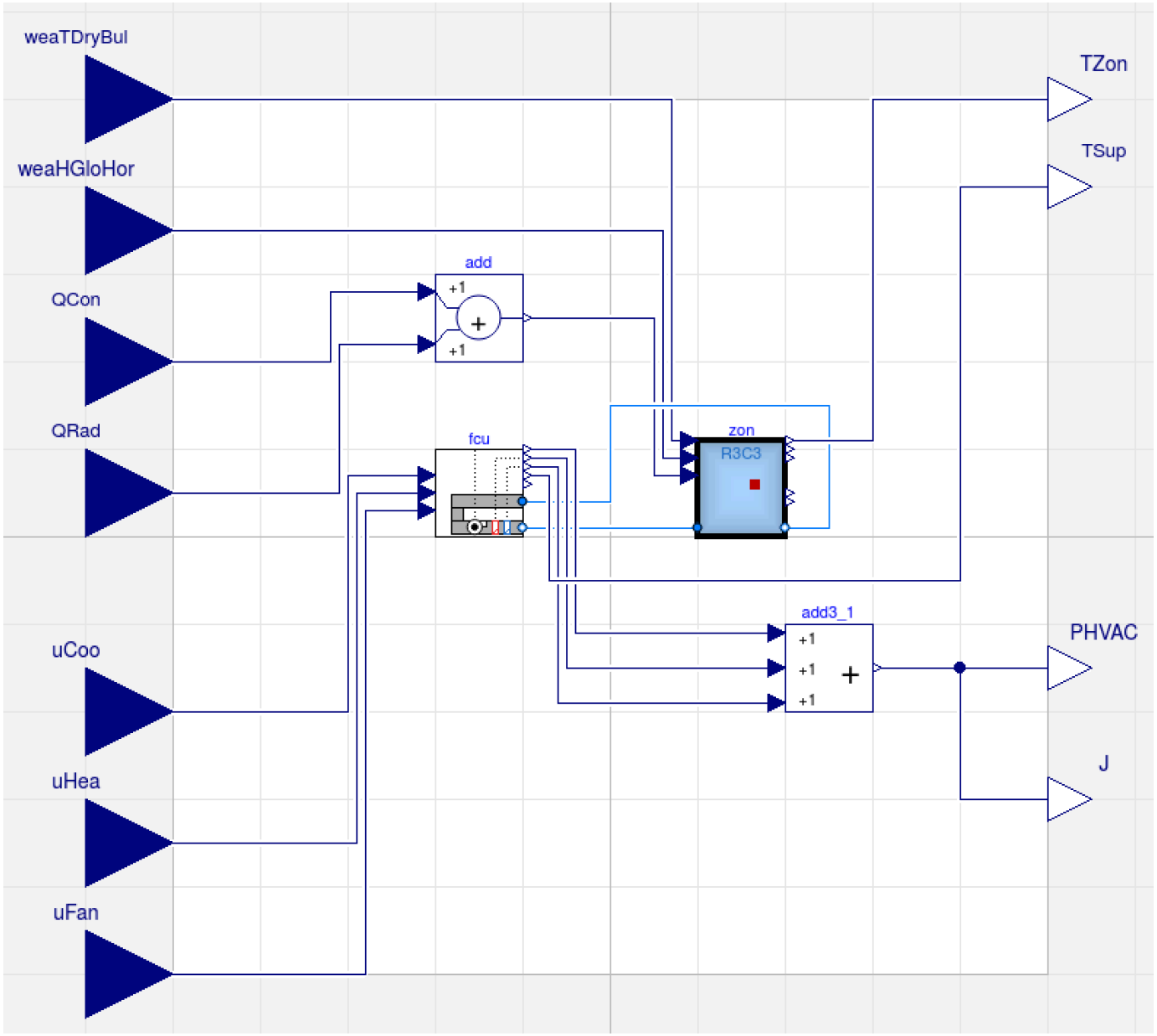
MPC Model: Envelope



MPC Model: FCU



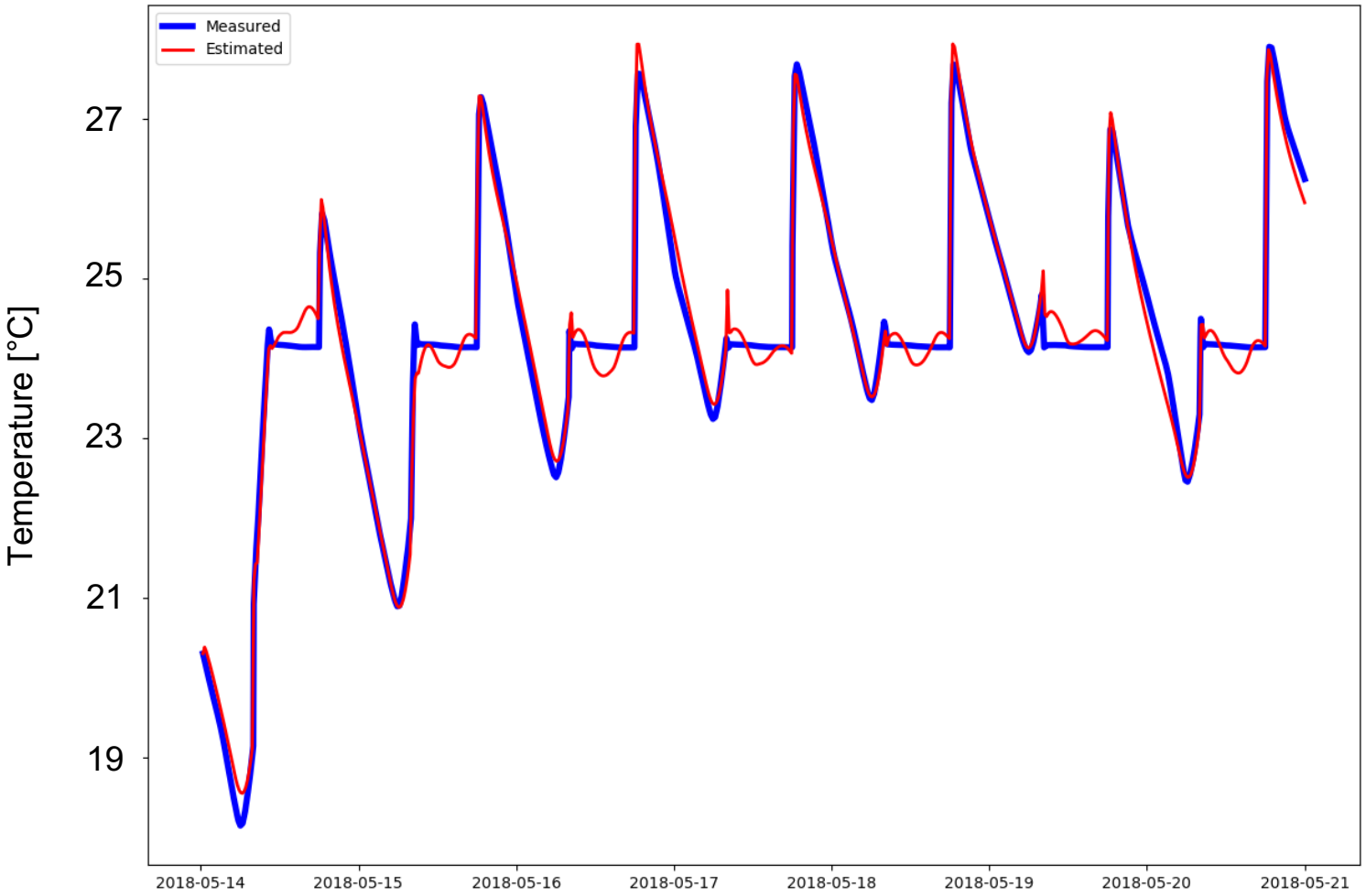
MPC Model: System



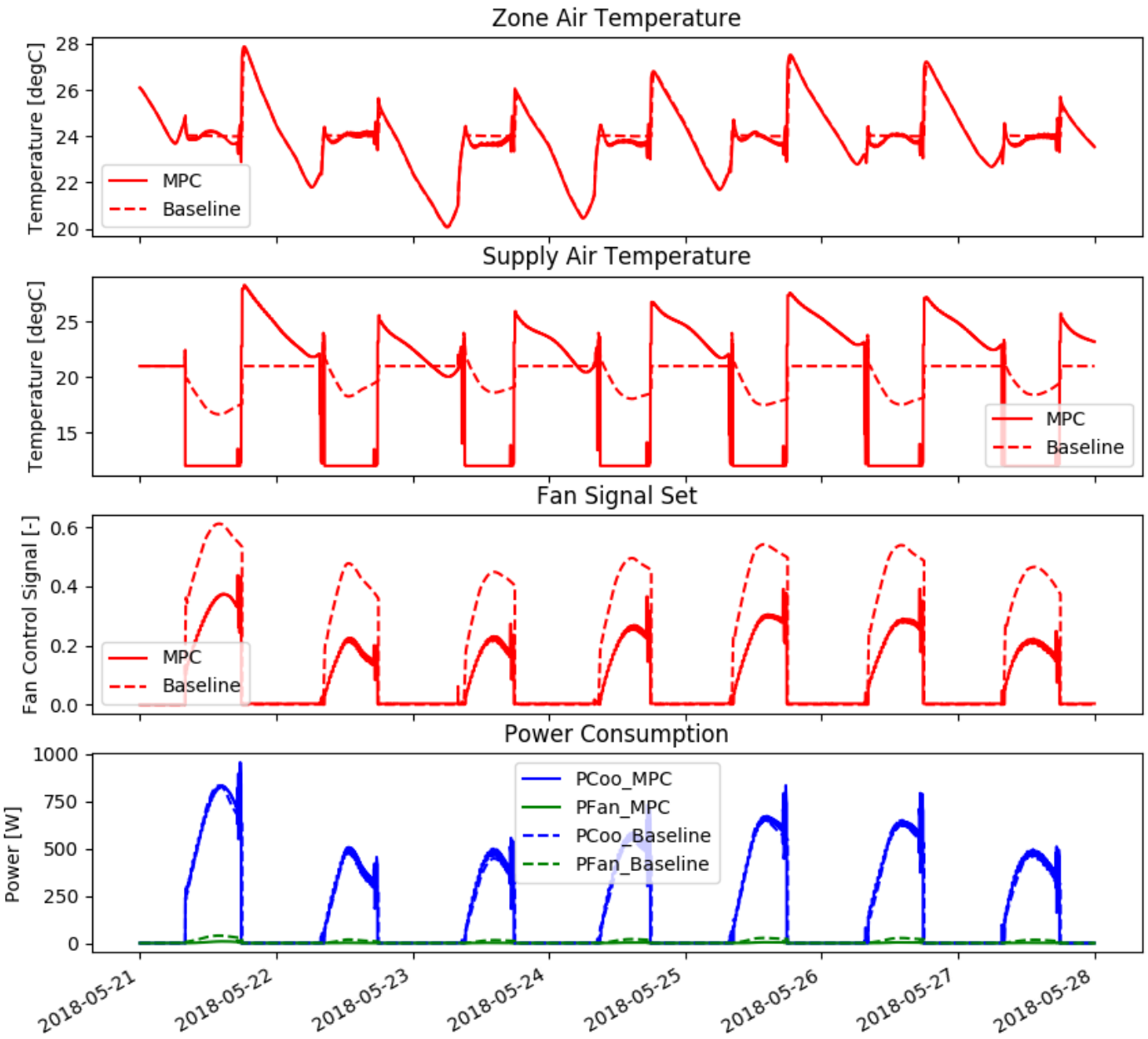
Interface for BOPTEST

```python
#%% GENERAL PACKAGE IMPORT
# ------------------------
import requests
import pandas as pd
import controller
# ------------------------

#%% SETUP TEST CASE
# -----------------
# Set URL for testcase
url = 'http://localhost:5000'
# Set simulation parameters
start_time = 20*7*24*3600
length = 7*24*3600
com_step = 600
test_warmup_period=7*24*3600
# -----------------

#%% TEST CONTROLLER IMPORT
# ------------------------
mpc = controller.controller()
mpc_horizon=6*3600
mpc_warmup_period=8*3600
# ------------------------

#%% RUN TEST CASE
# --------------
# Initialize
res = requests.put('{0}/initialize'.format(url), data={'start_time':start_time,'warmup_period':test_warmup_period})
# Set communication step
res = requests.put('{0}/step'.format(url), data={'step':com_step})
# Set forecast parameters
res = requests.put('{0}/forecast_parameters'.format(url), data={'horizon':mpc_horizon, 'interval':com_step})
# Run test case
print('\nRunning test case...')
# Initialize u
u = {}
# Simulation Loop
for i in range(int(length/com_step)):
    # Use MPC control after sufficient warmup
    if (i*com_step >= mpc_warmup_period):
        # Update results in controller database
        res_his = requests.get('{0}/results'.format(url)).json()
        mpc.update_database(res_his, 'historic')
        # Update forecast in controller database
        res_for = requests.get('{0}/forecast'.format(url)).json()
        mpc.update_database(res_for, 'forecast')
        # Compute optimal control
        start_mpc = pd.Timedelta(seconds=start_time+i*com_step)+pd.to_datetime('1/1/2018')
        final_mpc = start_mpc + pd.Timedelta(seconds=mpc_horizon)
        start_historic_mpc = start_mpc - pd.Timedelta(seconds=mpc_warmup_period)
        mpc.optimize_control(start_mpc, final_mpc, start_historic_mpc)
        # Set control signals
        u = mpc.get_control_setpoints(start_mpc)
    # Advance simulation with input
    y = requests.post('{0}/advance'.format(url), data=u).json()
# --------------
```

Parameter Estimation



Control Results: Timeseries



Control Results: KPIs

KPI	Baseline	MPC	% Difference
Energy [kWh]	31.30	30.60	-2.25
Cost [$]	1.71	1.67	-2.29
Thermal Discomfort [Kh]	1.17	4.81	312.36
IAQ [ppmh]	598.51	597.29	-0.20
Emissions [kg CO2]	20.72	20.24	-2.28
Time Ratio [-]	8.82-06	0.02	173100.17

BOPTEST Development

- Weather station (#234, Modelica-IBPSA #1402)
- Dynamic cost KPI API (#244)
- Data input type checking
- Reset time ratio KPI on initialization of test case