# MPC Trials

## IBPSA Project 1 Expert Meeting (Virtual)

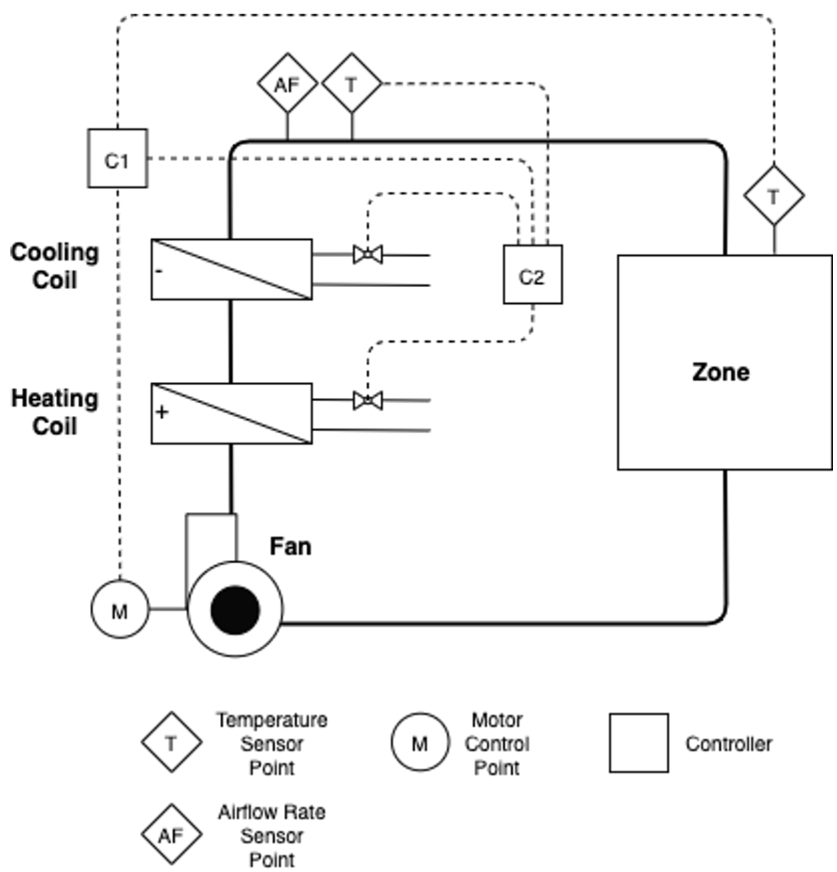
## WP1.2 Day 1 Session 1

5/7/2021

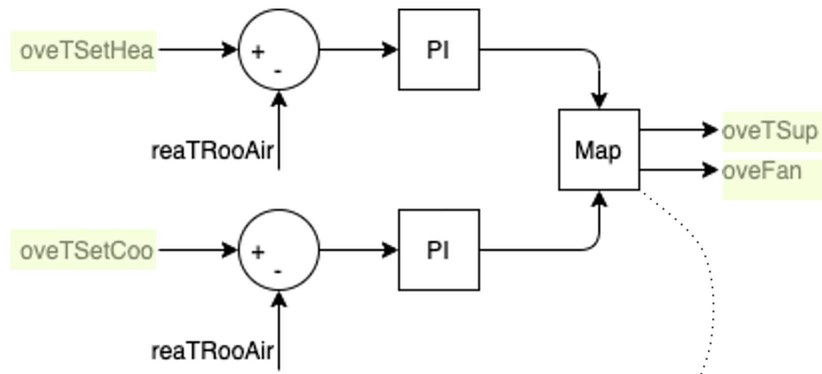
**David Blum**
Building Technology and Urban Systems
Lawrence Berkeley National Laboratory
dhblum@lbl.gov

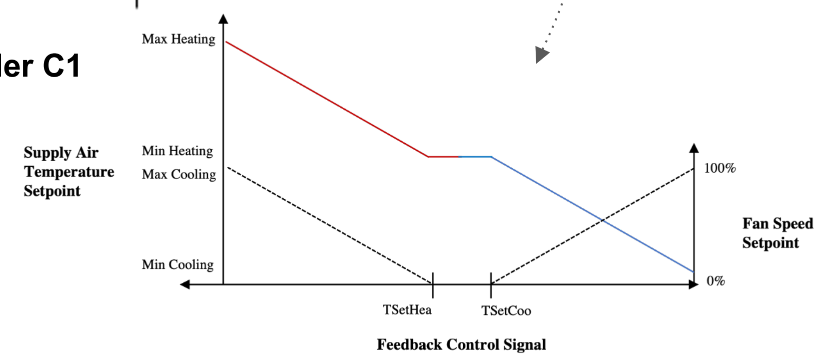
# Test Case

**bestest_air:** (Developer LBNL, Reviewer KU Leuven)
BESTEST Case 900 room model with idealized fan coil unit.

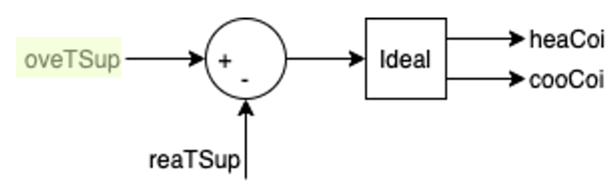



System Schematic




Controller C1




Controller C2

# MPC

## Demo MPC Test on BESTEST Air Building Type Test Case

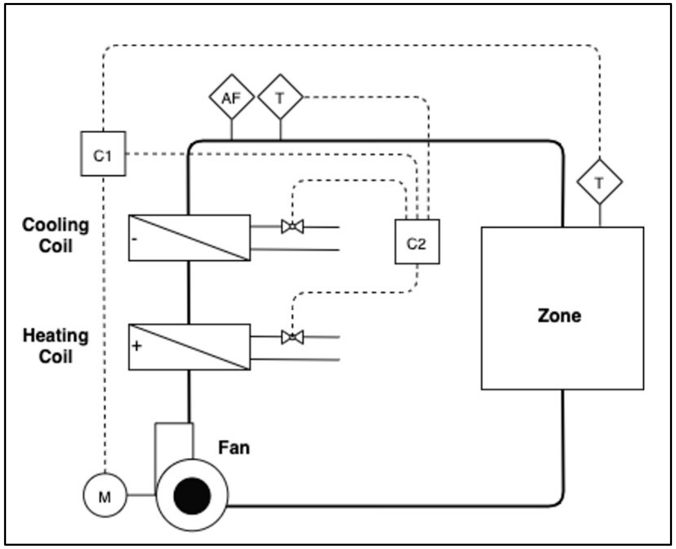
- Controller Implementation: MPCPy (Modelica + JModelica)
- Models:
  - Envelope: R3C3
  - Fan: $P_{fan} = a*V^3 + b*V^2 + c*V + d$
  - Heating: $P_{hea} = Q_{hea}/eff_{hea}$
  - Cooling: $P_{coo} = Q_{coo}/eff_{coo}$
- Parameter Estimation: Least squares optimization
- State Estimation: Simple moving 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

# Interface

```python
#%% SETUP TEST CASE
# ----------------
# Set URL for testcase
url = 'http://localhost:5000'
# Set simulation parameters
use_mpc = True
scenario = {'time_period':'typical_cool_day',
            'electricity_price': 'constant'}
com_step = 600
test_warmup_period=7*24*3600
# ----------------
#%% TEST CONTROLLER IMPORT
# ------------------------
mpc = controller.controller()
mpc_horizon=6*3600
mpc_warmup_period=8*3600
measurement_points = ['zon_reaPPlu_y','zon_reaPLig_y','fcu_reaTSup_y',
                      'fcu_reaFanSet_y','zon_weaSta_reaWeaTDryBul_y',
                      'zon_weaSta_reaWeaHGloHor_y', 'zon_reaTRooAir_y']
# ------------------------
#%% RUN TEST CASE
# ----------------
# 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 = {}
# Initialize boptest
y = requests.put('{0}/scenario'.format(url), data=scenario).json()['time_period']
# Record test start time
start_time = y['time']
# Simulation Loop
while y:
    # Use MPC control after sufficient warmup
    if (y['time'] >= mpc_warmup_period + start_time) and use_mpc:
        # Update results in controller database
        res_his = dict()
        current = y['time']
        start = current - mpc_warmup_period
        for point in measurement_points:
            res_his[point] = requests.put('{0}/results'.format(url),
                                          data={'point_name': point, 'start_time':start,'final_time':current}).json()[point]
            if 'time' not in res_his:
                res_his['time'] = requests.put('{0}/results'.format(url),
                                          data={'point_name': point, 'start_time':start,'final_time':current}).json()['time']
        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=current)+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()
# --------------
#%% Get KPIs
# ----------
kpis = requests.get('{0}/kpi'.format(url)).json()
print(kpis)
```

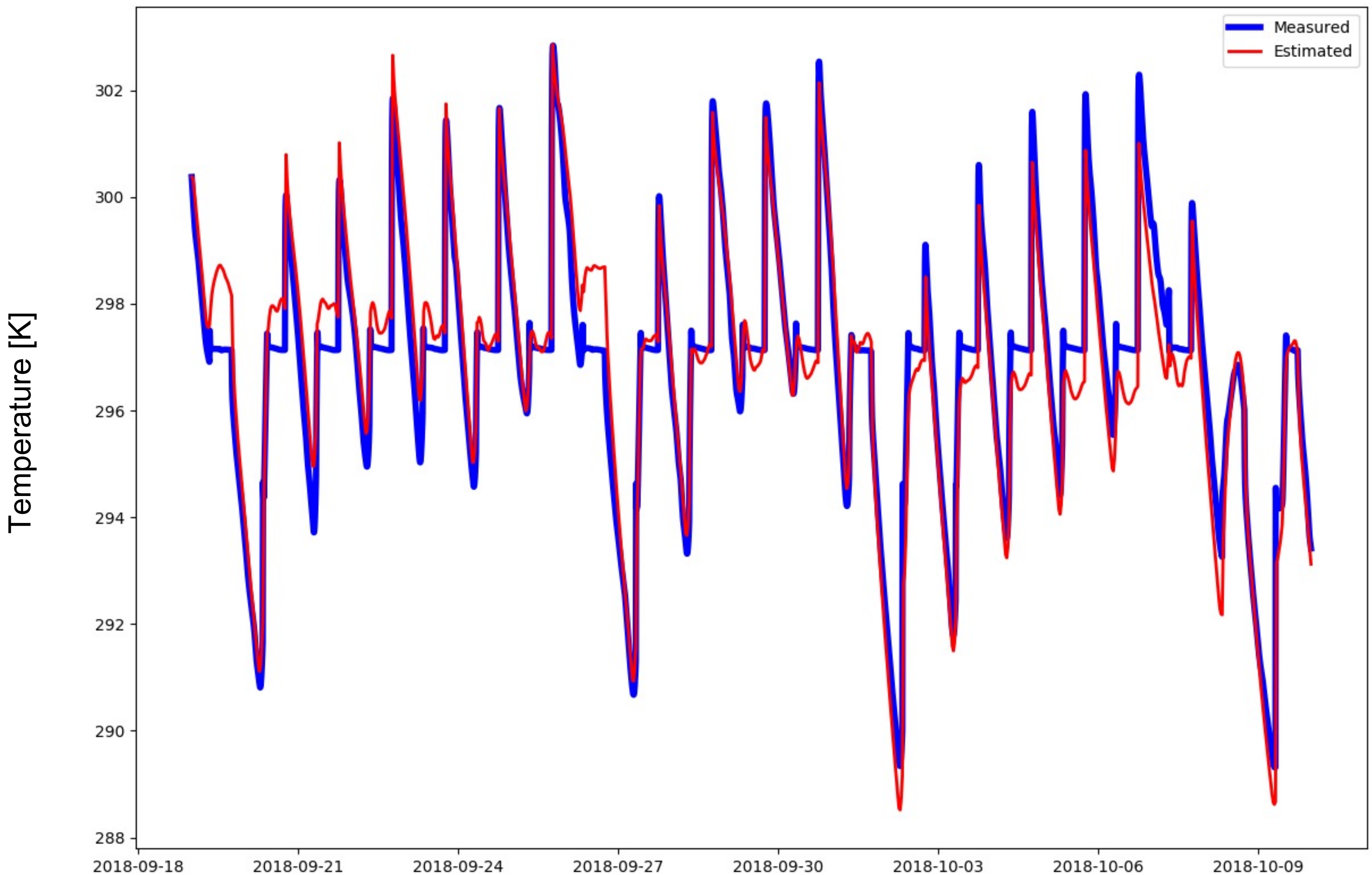
- Controller Implementation: MPCPy (Modelica + JModelica)
- Models:
  - Envelope: R3C3
  - Fan: $P_{fan} = a*V^3 + b*V^2 + c*V + d$
  - Heating: $P_{hea} = Q_{hea}/eff_{hea}$
  - Cooling: $P_{coo} = Q_{coo}/eff_{coo}$
- Parameter Estimation: Least squares optimization
- State Estimation: Simple moving horizon
- Control Optimization: Minimize ($E_{fan} + E_{hea} + E_{coo}$)
  s.t. $T_{Min} \le T_{Zon} \le T_{Max}$
  $Q_{Min} \le (Q_{coo}$ or $Q_{hea}) \le Q_{Max}$
- Horizon: 6 hours
- Control step: 10 minutes
- System control signals: Fan Speed, SAT Setpoint





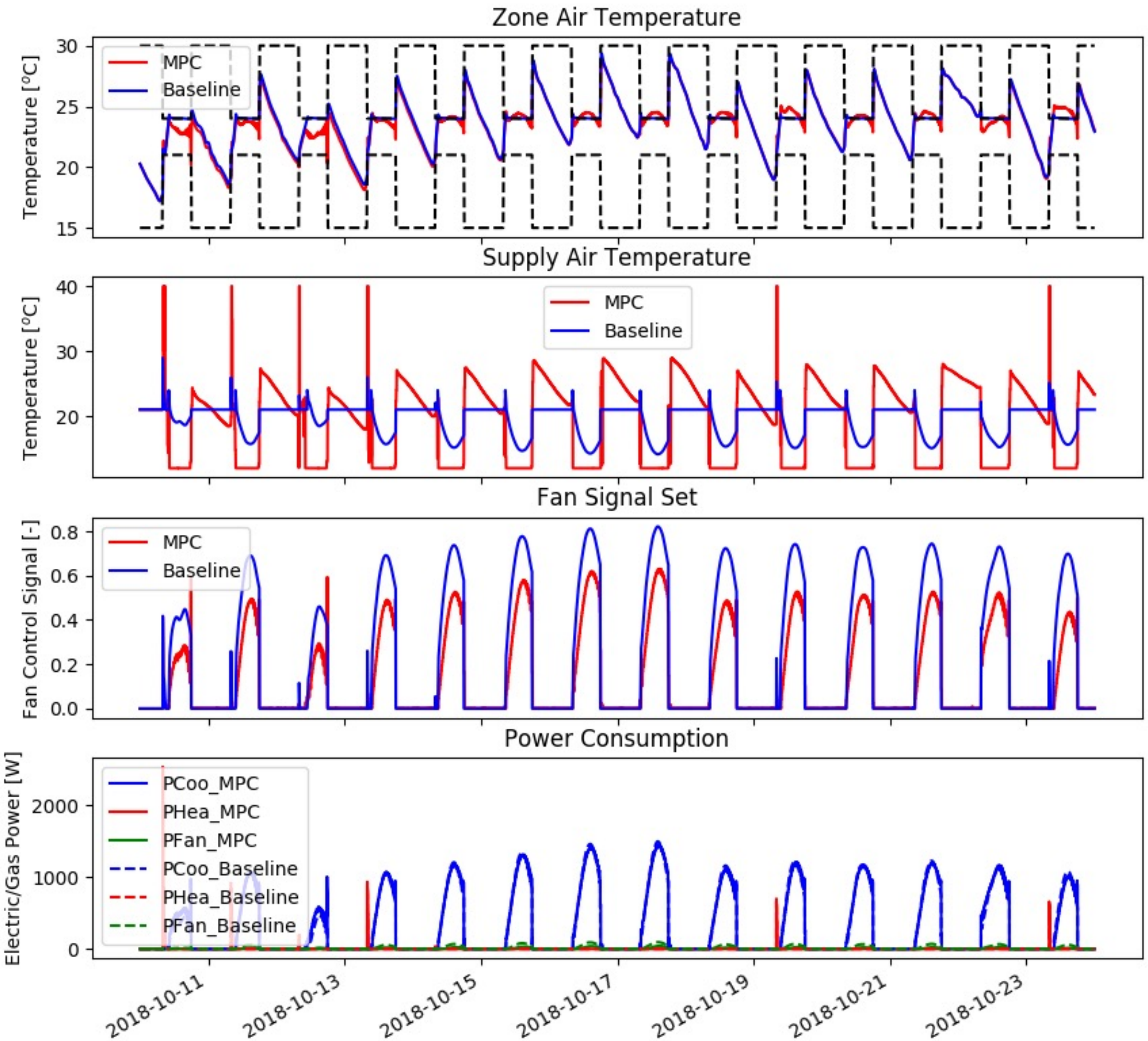
# Results – Peak Cool Day Scenario

## Parameter Estimation for Envelope Model



# Results – Peak Cool Day Scenario

## Control




| KPI | Baseline | MPC |
| --- | --- | --- |
| Energy [kWh] | 106.8 | 105.3 |
| Thermal Discomfort [Kh] | 7.7 | 24.3 |
| Computational Time Ratio | 0.00065 | 0.0066 |

See also “BOPTREE”: https://docs.google.com/spreadsheets/d/1E-5wR7nasW8h6kEtrXnUzcRrEA6f7TIxjbtSdB1Cz3w/edit?usp=sharing