# Coupling physical and machine learning models: case study of a single-family house

Basak Falay[1] Sandra Wilfling[2] Qamar Alfalouji[2] Johannes Exenberger[2] Thomas Schranz[2]
Christian Møldrup Legaard[3] Ingo Leusbrock[1] Gerald Schweiger[2]

[1]AEE-Institue for Sustainable Technologies, Austria b.falay@aee.at
[2]Institute of Software Technology, Technical University of Graz, Austria gerald.schweiger@tugraz.at
[3]DIGIT, Department of Electrical and Computer Engineering, Aarhus University, Denmark, cml@ece.au.dk

## Abstract

The emergence of Cyber-Physical Systems poses new challenges for traditional modelling and simulation techniques. We need to combine white, grey, and black box models as well as different tools developed for specific subsystems and domains. Co-simulation is a promising approach to modeling and simulating such systems. This paper presents a case study where a physical model of a building's heating system implemented in Modelica is co-simulated with a machine learning model of a stratified hot water tank implemented in Python. The Python model is exported as Functional Mock-up Unit using UniFMU.

*Keywords: Co-Simulation, Functional Mock-Up Interface, Modelling, Machine Learning*

## 1 Introduction

Future intelligent and integrated energy systems must have a high degree of flexibility and efficiency to ensure reliable and sustainable operation (Lund et al. 2017). Along with the rapid expansion of renewable energy, this degree of flexibility and efficiency can be achieved by overcoming the clear separation between different sectors and by increasing connectivity and the associated data availability through the integration of sensors and edge/fog computing (Vatanparvar and Faruque 2018). All of these developments drive the transition towards so-called Cyber-Physical Energy Systems (Palensky, Widl, and Elsheikh 2013). Cyber technologies (sensors, edge/fog computing, IoT networks, etc.) can monitor the physical systems, enable communication between different subsystems, and control them. Thus, the emergence of Cyber-Physical Systems poses new challenges for traditional modelling and simulation approaches.

One of these challenges is that models need to combine computational systems and data communication networks with physical systems. Furthermore, recent studies showed that pure white-box models based on first principles deal with drawbacks such as time-consuming development, validation problems or low computational speed (Li and Wen 2014). Consequently, these approaches have limited use for complex systems such as intelligent buildings outside of academia (Schweiger, Nilsson, et al. 2020). Black-box approaches examine the system from the outside using input/output relations. Depending on the approach, they are computationally efficient but compared to white-box approaches they lack in generalizability and extensibility (Thieblemont et al. 2017). Beside white-box and black-box models, grey-box models fall in between (Harish and Kumar 2016). Several papers highlighted the importance of combining white-, grey-, and black-box models for analyzing and optimizing Cyber-Physical Systems (O'Dwyer et al. 2019; Killian and Kozek 2016; Thilker, Madsen, and Jørgensen 2021).

There are two options to simulate the interactions between subsystems; (i) the entire system can be modelled and simulated with a single tool referred to as monolithic, (ii) already established models for the respective subsystems are coupled in co-simulation (Gomes et al. 2018). A recent survey discussed the advantages, disadvantages, and challenges of co-simulation approaches (Schweiger, Engel, et al. 2018). This survey showed that experts consider the Functional Mock-Up Interface (FMI) standard to be the most promising standard for continuous-time, discrete-event, and hybrid co-simulation.

In this paper, the physical parameters of a subsystem (stratified storage tank) are not available. In this situation, model calibration and parameter estimation approaches can be used depending on availability of the measurement data. On the other hand, machine learning models can be as well used to mimic the behavior of the system by construct relationships between input and outputs without being dependent on the components parameters. Artificial Neural Networks was used to model the stratified storage tank in (Géczy-Vig and Farkas 2010). In this work, Random Forest (RF) was used to model the temperatures in each layer of the stratified storage tank. Since the states of the other components influence the state of the stratified storage, we have created a co-simulation workflow where the machine learning and physical models can be coupled. Physical and machine learning models are available at https://github.com/tug-cps/NextHyb2 . Unfortunately, we cannot publish the data due to data privacy policy. Therefore, we have additionally generated a synthetic, open-source data set.

DOI
10.3384/ecp21181335

# 2 Method

## 2.1 Heating System of Single-family House

A single-family house with $180m^2$ floor area, located in Austria with an annual energy consumption of 7500 kWh was analyzed in this work. Figure 1 gives an overview of the main components of the single-family house heating system. The house, equipped with a floor heating system, has three different heat sources: (i) a solar collector with $46m^2$ flat plate area, (ii) a stove which directly heats the house, and the excess heat feeds the storage tank and (iii) an air-to-water heat pump. Additionally, an estimated $3m^3$ storage tank bridges these three heat sources in order to increase the efficiency of the heating system. The house has an indoor pool ($24m^3$), which is heated by the hot water storage tank or directly by the solar collector.

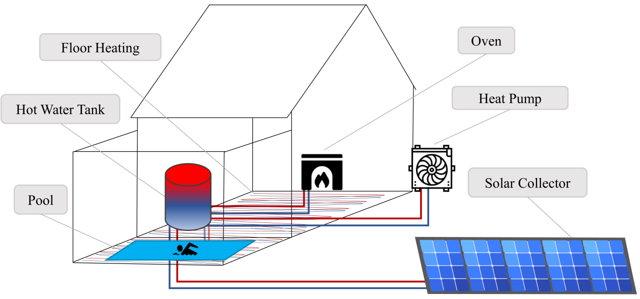

**Figure 1.** Overview of the single-family house heating system. Red line represents the supply and blue line represents the return temperatures.

The following rule-based control strategy of the heating system is given below.

- The priority of the solar collector is to maintain the temperature of top layer of the storage tank at 52°. If this condition is satisfied, then the excess heat from the solar collectors heats the indoor pool to 35°.
- If the solar collector cannot meet the heating demand of the indoor pool and if the top layer temperature of storage tank is higher than 52°, the storage tank heats the pool.
- If these conditions don't satisfy or the temperature of the bottom layer of the storage tank drops below 35°, the heat pump turns on.
- If the temperature of the stove is higher than 40°, the excess heat is fed into the storage tank.

## 2.2 Measurement Data

In Appendix, Figure A.1 shows an overview of the heating system components and the locations of the heat meters. Temperatures are represented in ($T_{component}$, mass flow rate in $dm_{component}$. Four temperature sensors from top to bottom respectively $T_{Storage,1}, T_{Storage,2}, T_{Storage,3}, T_{Storage,4}$ are located at the storage tank. The measured data from the heat meters is between 01.02.2019 and 31.01.2020, with a temporal resolution of 1 minute. Figure 2 gives an overview of the data quality of the measurement data. White lines represent the missing data points and corresponding periods. 4% of the measurement data is missing; 65% of them falls into the period between November 2019 and January 2020, 25% of them falls into September 2019. In addition to missing values, there are wrong measurements between November 2019 and January 2020 due to the failures in the meters.

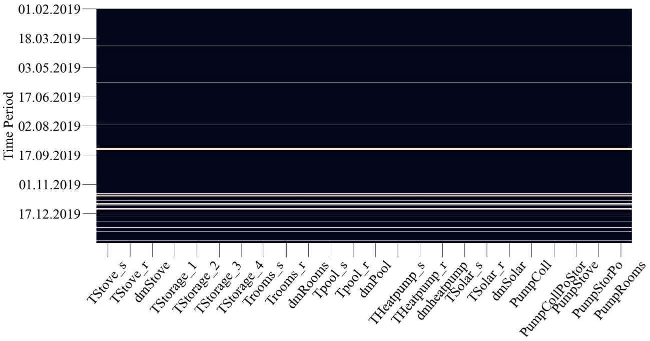

**Figure 2.** Missing data periods for the given measurement data

The missing parts of the data were imputed by taking the profile of the previous day. Figure 3 demonstrates the imputation of missing data points for four days in a row, given in dashed lines. The measurement data was ignored after November 2019 due to the bad quality of data. The whole data set was resampled to 15-minute values to avoid the over fitting the predictions of the ML model. After post-processing, the dataset had 27840 datapoints. The resampled data was later used for training and testing for the ML model.

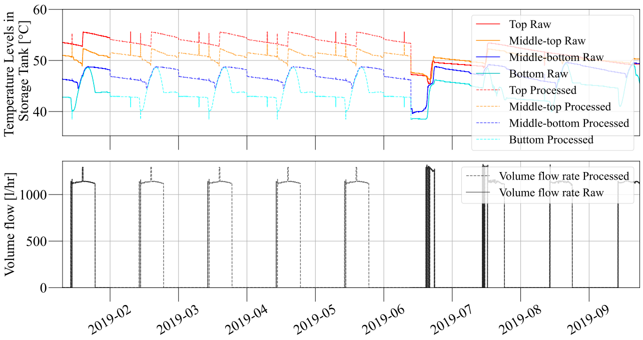

**Figure 3.** Imputation of the missing data

One of the most critical features in the dataset is mass flow rates from each component. Figure 4 shows the sparsity of the mass flow rates from each components. The y-axis represent the total data points (27840) after preprocessing, the x-axis represents the mass flow values of the components. The black points in the figure show the values that are not zero and the gaps between the black points represent the zero values. Mass flow rate in the stove has the highest percentage 99.8% of zero values and the mass flow rate in the *CollectortoPool* has the lowest percentage with 86%.

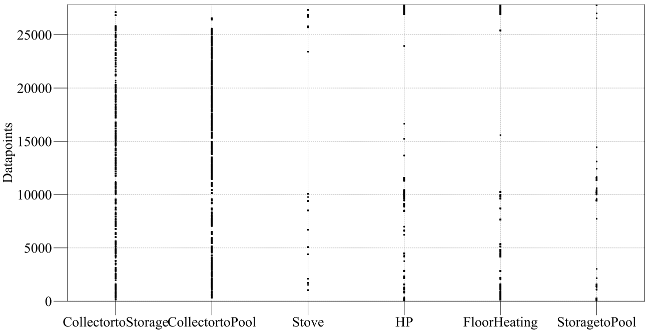

**Figure 4.** Visualization of data sparsity in mass flow rates of each component

## 2.3 Physical Models

The physical models were implemented in the Modelica language (Fritzson and Engelson 1998). Modelica is an open source, a-causal, object-oriented and multi-domain modelling language. A discussion of limitations and promising approaches of the Modelica language can be found (Schweiger, Nilsson, et al. 2020). All the models used in this system are based on the Modelica IBPSA Project 1 library (Wetter, Treeck, et al. 2019) and the Buildings Library (Wetter, Zuo, et al. 2014). Dymola was used to simulate Modelica models (Brück et al. 2002). The following sub - implemented in Modelica are: Solar system (`Buildings.Fluid.SolarCollectors.EN12975`), heat pump (`Buildings.Fluid.HeatPumps.CarnotTCon`) and the indoor pool (`Modelica.Fluid.Vessels.ClosedVolume`). The energy demand of the house and the heat supply profile of the stove were taken from the measurement data instead of modelling these components. Since there was no weather profile acquired within the given data period, Typical Meteorological Year 3 (TMY3) for Austria were generated from Meteonorm.

## 2.4 Machine Learning Model

There was no available information of the system parameters of the storage tank such as the insulation material and the thickness, the wall thickness, the height or the locations of the temperature sensors. Therefore, the storage tank was modelled based on RF. RF is a combination of tree predictors which splits nodes based on a best split of random subsets of the features, thus reducing the variance of the tree model and increasing the overall predictive power of the model.

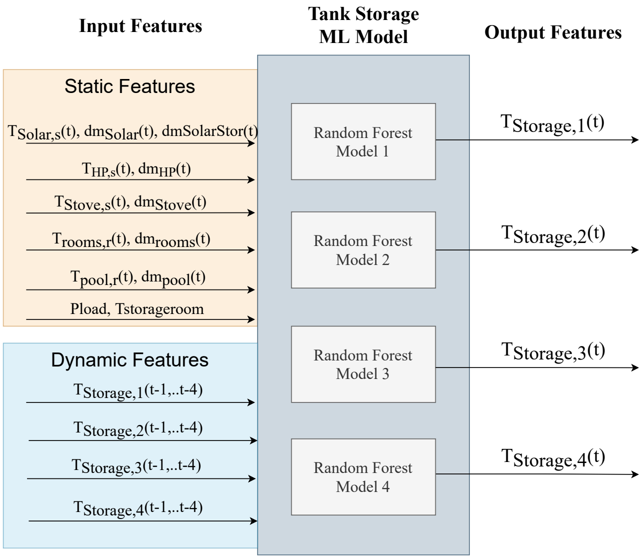
Input Features
Tank Storage ML Model
Output Features
Static Features
$T_{Solar,s}(t)$, $dm_{Solar}(t)$, dmSolarStor(t)
$T_{HP,s}(t)$, $dm_{HP}(t)$
$T_{Stove,s}(t)$, $dm_{Stove}(t)$
$T_{rooms,r}(t)$, $dm_{rooms}(t)$
$T_{pool,r}(t)$, $dm_{pool}(t)$
Pload, Tstorageroom
Dynamic Features
$T_{Storage,1}(t-1,..t-4)$
$T_{Storage,2}(t-1,..t-4)$
$T_{Storage,3}(t-1,..t-4)$
$T_{Storage,4}(t-1,..t-4)$
Random Forest Model 1
Random Forest Model 2
Random Forest Model 3
Random Forest Model 4
$T_{Storage,1}(t)$
$T_{Storage,2}(t)$
$T_{Storage,3}(t)$
$T_{Storage,4}(t)$

**Figure 5.** Input/Output features of the applied machine learning model of the storage tank.

The RF model predicted the four temperature layers of the storage tank. An overview of the input features for the model is given in Figure 5. The static input features are temperatures, $T_i$, and mass flow rates, $dm_i$, from the solar collector, heat pump, floor heating and stove. The dynamic features are the four temperature layers of the storage tank with a 1-hour look-back time with interval 15 minutes and 15 minutes prediction horizon, see Figure 6. The measured data was split randomly into training (80%, 50 epochs) and testing (20%). The model hyperparameters are n_estimators = 100 which represents the number of decision trees that achieves the best trade-off between the accuracy and efficiency; max_depth that has been set to an unlimited value so the nodes can expand automatically; and min_samples_split = 2. The implementation was done using the Python framework presented in (Schranz et al. n.d.) based on Scikit-learn.

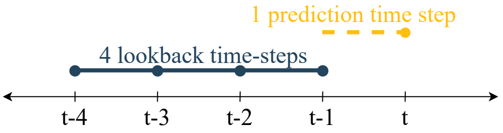

**Figure 6.** At time t, four look-back time-steps are used to predict one time-step in future with each step = 15 minutes.

### 2.4.1 Model Performance Analysis

Two criteria were selected to evaluate the performance of the RF model: the coefficient of variation of the Root mean square error (CVRMSE) and mean absolute percentage error (MAPE) given in Equation 1 and Equation 2.

$$CV(RMSE) = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}}{Y} * 100 \tag{1}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|Y_i - \hat{Y}_i|}{Y_i}\right) * 100 \tag{2}$$

where Y is the true value, $\hat{Y}$ is predicted value, $\bar{Y}$ is the average of the true values over N test samples.

### 2.5 Co-Simulation

To integrate the ML model of the storage tank into the simulation environment, the ML model must be exported as an FMU. The UniFMU tool (Legaard et al. 2021) was used to generate a Python-based FMU. Therefore a template of the FMU was generated using the command `unifmu generate python name`.

To specify the behavior the dummy example in Listing 1, implemented by the generated FMU, is replaced with the components of the ML model. A benefit of this is that the `scikit-learn` code can be reused and integrated into the FMU gradually. This makes sure that no breaking changes occur. A crucial part of the FMUs implementation is the `fmi2DoStep` method which instructs the model to simulate forward in time for an amount of time corresponding to the time step. For the storage-tank FMU, this is equivalent to running one or more inference steps of the trained model.

**Listing 1.** Implementation of `fmi2DoStep` by storage model.

```
from sklearn.ensemble import
    RandomForestRegressor
from sklearn.datasets import
    make_regression
...
def do_step(current_time,step_size,
    no_step_prior):
     self.temp_next=self.forrest(self.
        temp_prevs)
return Fmi2Status.ok
```

## 3 Results and Discussion

### 3.1 Validation of the ML model

Table 1 shows the model performance on the test data set. The ML model is imported as FMU in Dymola. Testing of the FMU-ML model with the measurement data is performed in Dymola environment, see Figure 7. $T_{Storage,4}$ and $T_{Storage,3}$ are the worst predicted target value according to the CVRMSE and MAPE. The discussion of Table 1 is supported with the results of the testing.

**Table 1.** Performance metrics of predicting the four target temperature values: $T_{Storage,1}$, $T_{Storage,2}$, $T_{Storage,3}$ and $T_{Storage,4}$ using random forest models

| | *CVRMSE* | *MAPE* |
|---|---|---|
| $T_{Storage,1}$ | 0.0097 | 2.3056 |
| $T_{Storage,2}$ | 0.011 | 2.2656 |
| $T_{Storage,3}$ | 0.0157 | 4.9064 |
| $T_{Storage,4}$ | 0.0281 | 6.0215 |

Winter, spring and summer periods were chosen, aiming to represent different boundary conditions. The only difference in each test period was the initial values set for the FMU-ML. These initial values of the static and dynamic input features were chosen from the measurement data based on each period starting time. Since the stratified hot water storage tank is a short term storage, the daily predictions are representative. In these three figures, 9 days period for each season was chosen to show the model prediction based on interactions of all heating supplies. In Figure 8, Figure 9 and Figure 10, the first subplot represents the comparison between the true and the predicted temperature values of each 4 layers of the storage tank. The true temperature values of the top layer, middle-top layer, middle bottom layer and the bottom layer are respectively red, dark orange, blue and cyan dashed lines. The predicted temperature values are represented the same color code but in straight lines. In the second subplot, the mass flow rates from different components are given. The mass flow rates stand for when the specific component is turned on/off.

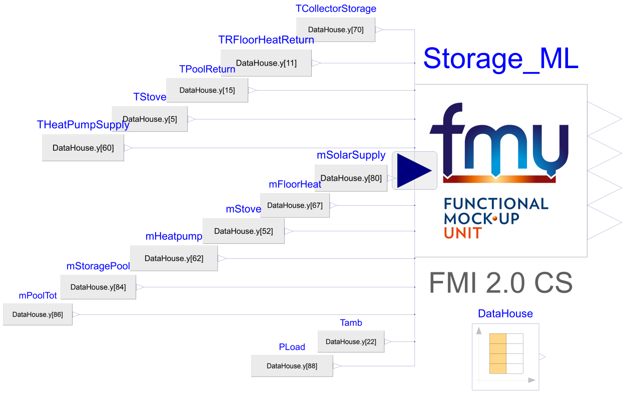

**Figure 7.** Dymola layout of testing FMU-ML storage model

Figure 8 represents the frequently running components; solar collector and pool heating for summer period. Based on these components inputs, the ML model shows good aggrement with the measurement data during the summer period. One of the static input features, "PLoad", which indicates whether at least one component in the system is on, is introduced to capture the cooling behavior of the storage tank. It is observed in the summer period during the night when there is no load, the four temperature values of the storage tank decrease.

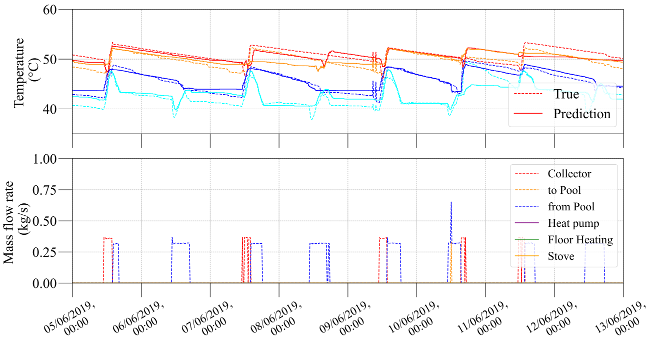

**Figure 8.** Storage temperature levels predicted vs measurement in summer period

The cold return water from the floor heating is fed into the storage tank from the bottom and the middle layer. These temperatures are expected to decrease as in the measurement data. In Figure 9, between 18th and 19th March, when the floor heating starts, the predictions of the temperature of the bottom layer fails. On the 19th March, when the heat pump is on, represented in the purple line, the middle bottom temperature shows an increasing behavior. However, it cannot reach to the values of the measurement data. On 20th March, only two components are on as in the summer period. On this day, the prediction of the temperature values can catch the measurement data.

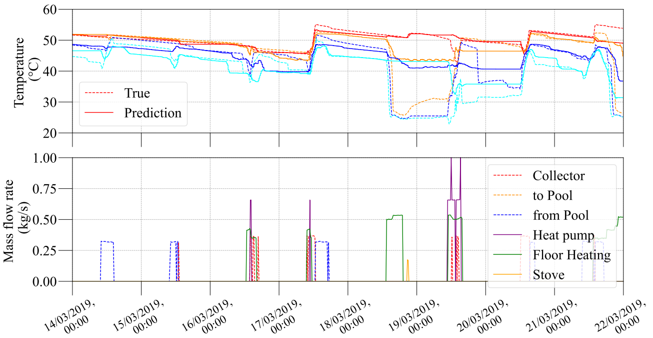

**Figure 9.** Storage temperature levels predicted vs measurement in spring period

The winter period is selected between February and March due to lack of winter representation from the data, explain in Section 2.2. The bottom temperature layer of the storage tank shows a decreasing behavior due to the floor heating as in the measurement data. However, due to the control strategies when the heat pump turns on, the middle bottom temperature of the storage tank doesn't increase as in the measurement data profile.

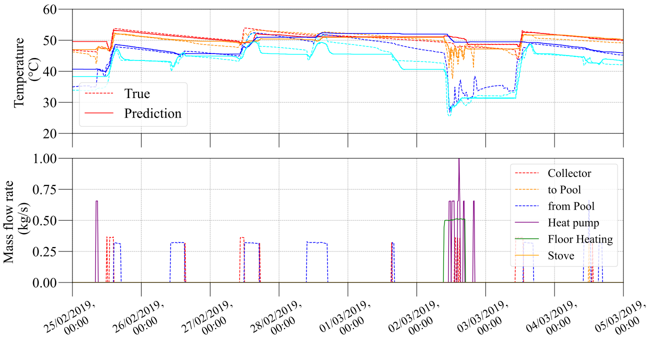

**Figure 10.** Storage temperature levels predicted vs measurement in winter period

From these three testing periods of the FMU-ML model, it is observed that the temperature values of the storage tank is predicted better when there is only collector component is on. In Figure 4, data that represent collector to storage and collector to pool is denser than the other components data. Therefore, these static input features can dominate the predictions more than the other static features which are sparse. Additionally, the dynamic features of the past predicted temperature values of the storage tank are as well input features. Once these values are predicted wrong, the error accumulates to the further time steps. Results from these tests also show the the performance of the $T_{Storage,4}$ and $T_{Storage,3}$ are worse than the other predicted target values.

## 3.2 System simulation

In Appendix, Figure A.2 shows the system implementation in Dymola. All heating components explained in usecase are framed with dashed lines in the figure. The figure is visually simplified by hiding the source/sink component inside of the 'StorageML_FMU' component where the supply or return fluid from each component is fed to storage tank. All the simulations are run in a virtual Ubuntu environment with 188 GB RAM and the Intel Xeon Silver 4215R CPU @ 3.2GHz CPUs. Dymola 2021 FD01 with Dassl solver and 10e-6 tolerance was used during this study. The system simulation with the FMU-ML was run 10 times. The averages of the CPU times for the 32 days simulation with 15 minutes interval is 228 seconds. The CPU-time taken to calculate one grid interval highly depends on how the ML algorithm is implemented, number of inputs features, number of processors.

The translated model statistics are given in Table 2. The originally described system has 1966 non-trivial DAEs, after translation it is reduced to an ODE system with 42 continuous time states.

**Table 2.** Model statistics:Translated model statistics of the single-family house with the FMU-ML storage component.

| | FMU-ML Model |
|---|---|
| Constants | 1733 |
| Parameters depending | 654 |
| Continuous time states | 42 |
| Time varying variables | 777 |
| Alias variables | 1342 |
| Sizes of linear system of equations | {5} |
| Sizes after manipulation of the linear system of equations | {0} |
| Sizes of nonlinear system of equations | {6, 5, 3, 1, 1} |
| Sizes after manipulation of the nonlinear system of equations | {1, 1, 1, 1, 1} |
| Number of numerical Jacobians | 0 |

Figure 11 shows the results of coupling ML and physical models of the single-family house heating system. The first subplot in Figure 11 shows the temperature levels of the tank for a 32 days period. The second subplot shows which component is switched on and the third shows the load condition for the storage tank. Despite the same real-world control strategies implemented into system, weather profile that represents the the measurement data is not available. The TMY3 from Meteonorm is used

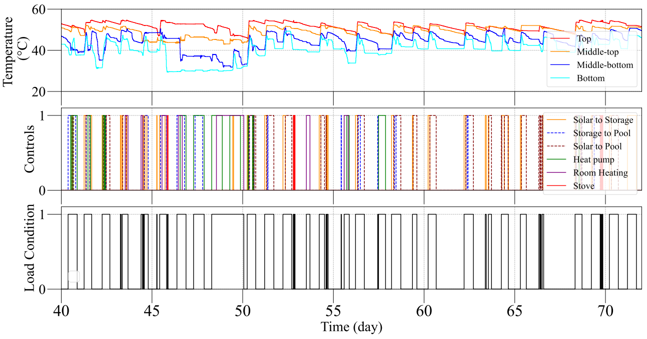

**Figure 11.** Storage temperature levels according to the controls

for the system simulation and solar collector provides different outputs than the measurement data. And all outputs based on the control strategies changes. Also the physical model parameters of the storage tank have not been adjusted to give a good comparison. Therefore, the results from FMU-ML system simulation cannot be compared with the measurement data.

## Acknowledgements

The presented research is part of the project "NextHyb2" (FFG project number 881150) financed by the Austrian Research Promotion Agency FFG. We are grateful to Klima Energie Fonds for sharing the data. We also would like to acknowledge the contributions and the fruitful discussions of Carles Ribas Tugores and Walter Becke in the context of the use case study.

## References

Brück, Dag et al. (2002). "Dymola for multi-engineering modeling and simulation". In: *Proceedings of modelica*. Vol. 2002. Citeseer.

Fritzson, Peter and Vadim Engelson (1998). "Modelica—A unified object-oriented language for system modeling and simulation". In: *European Conference on Object-Oriented Programming*. Springer, pp. 67–90.

Géczy-Vig, P. and I. Farkas (2010). "Neural network modelling of thermal stratification in a solar DHW storage". In: *Solar Energy* 84, pp. 801–806. ISSN: 2261-236X. DOI: 10.1051/matecconf/201822501015.

Gomes, Cláudio et al. (2018). "Co-simulation: a survey". In: 51.3. ISSN: 0360-0300. DOI: 10.1145/3179993.

Harish, VSKV and Arun Kumar (2016). "A review on modeling and simulation of building energy systems". In: *Renewable and sustainable energy reviews* 56, pp. 1272–1292.

Killian, Michaela and Martin Kozek (2016). "Ten questions concerning model predictive control for energy efficient buildings". In: *Building and Environment* 105, pp. 403–412.

Legaard, Christian Møldrup et al. (2021). "A Universal Mechanism for Implementing Functional Mock-up Units". In: *11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. SIMULTECH 2021. Virtual Event, to appear.

Li, Xiwang and Jin Wen (2014). "Review of building energy modeling for control and operation". In: *Renewable and Sustainable Energy Reviews* 37, pp. 517–537.

Lund, Henrik et al. (2017). "Smart energy and smart energy systems". In: *Energy* 137.2, pp. 556–565. DOI: 10.1016/j.energy.2017.05.123.

O'Dwyer, Edward et al. (2019). "Smart energy systems for sustainable smart cities: Current developments, trends and future directions". In: *Applied energy* 237, pp. 581–597.

Palensky, Peter, Edmund Widl, and Atiyah Elsheikh (2013). "Simulating cyber-physical energy systems: Challenges, tools and methods". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.3, pp. 318–326. DOI: 10.1109/TSMCC.2013.2265739.

Schranz, Thomas et al. (n.d.). "Energy Prediction under Changed Demand Conditions:Robust Machine Learning Models and Input Feature Combinations". In: *Building Simulation 2021. International Building Performance Simulation Association*.

Schweiger, Gerald, Georg Engel, et al. (2018). "Co-simulation an empirical survey: applications, recent developments and future challenges". In: *MATHMOD 2018 Extended Abstract Volume*, pp. 125–126.

Schweiger, Gerald, Henrik Nilsson, et al. (2020). "Modeling and simulation of large-scale systems: A systematic comparison of modeling paradigms". In: *Applied Mathematics and Computation* 365, p. 124713.

Thieblemont, Hélène et al. (2017). "Predictive control strategies based on weather forecast in buildings with energy storage system: A review of the state-of-the art". In: *Energy and Buildings* 153, pp. 485–500.

Thilker, Christian Ankerstjerne, Henrik Madsen, and John Bagterp Jørgensen (2021). "Advanced forecasting and disturbance modelling for model predictive control of smart energy systems". In: *Applied Energy* 292, p. 116889.

Vatanparvar, Korosh and Mohammad Abdullah Al Faruque (2018). "Control-as-a-Service in Cyber-Physical Energy Systems over Fog Computing". In: *Fog Computing in the Internet of Things: Intelligence at the Edge*. Ed. by Amir M. Rahmani et al. Cham: Springer International Publishing, pp. 123–144. ISBN: 978-3-319-57639-8. DOI: 10.1007/978-3-319-57639-8_7. URL: https://doi.org/10.1007/978-3-319-57639-8_7.

Wetter, Michael, C van Treeck, et al. (2019-09). "IBPSA Project 1: BIM/GIS and Modelica framework for building and community energy system design and operation – ongoing developments, lessons learned and challenges". In: vol. 323. IOP Publishing, p. 012114. DOI: 10.1088/1755-1315/323/1/012114. URL: https://doi.org/10.1088/1755-1315/323/1/012114.

Wetter, Michael, Wangda Zuo, et al. (2014). "Modelica Buildings library". In: *Journal of Building Performance Simulation* 7.4, pp. 253–270. DOI: 10.1080/19401493.2013.765506. URL: https://doi.org/10.1080/19401493.2013.765506.

# A Appendix

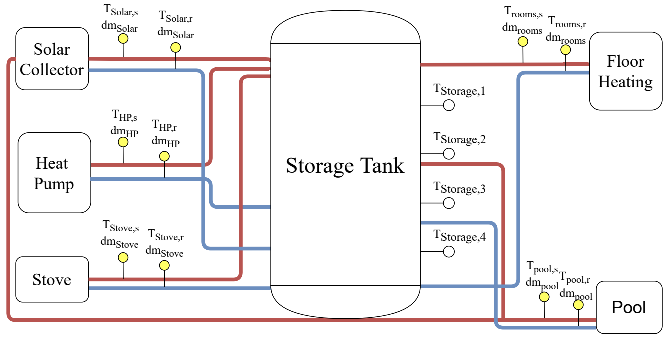

**Figure A.1.** Overview of the system hydraulic flow. The supply pipe is represented in red, return pipe in blue. In each pipe, the temperature and mass flow rates are measured.

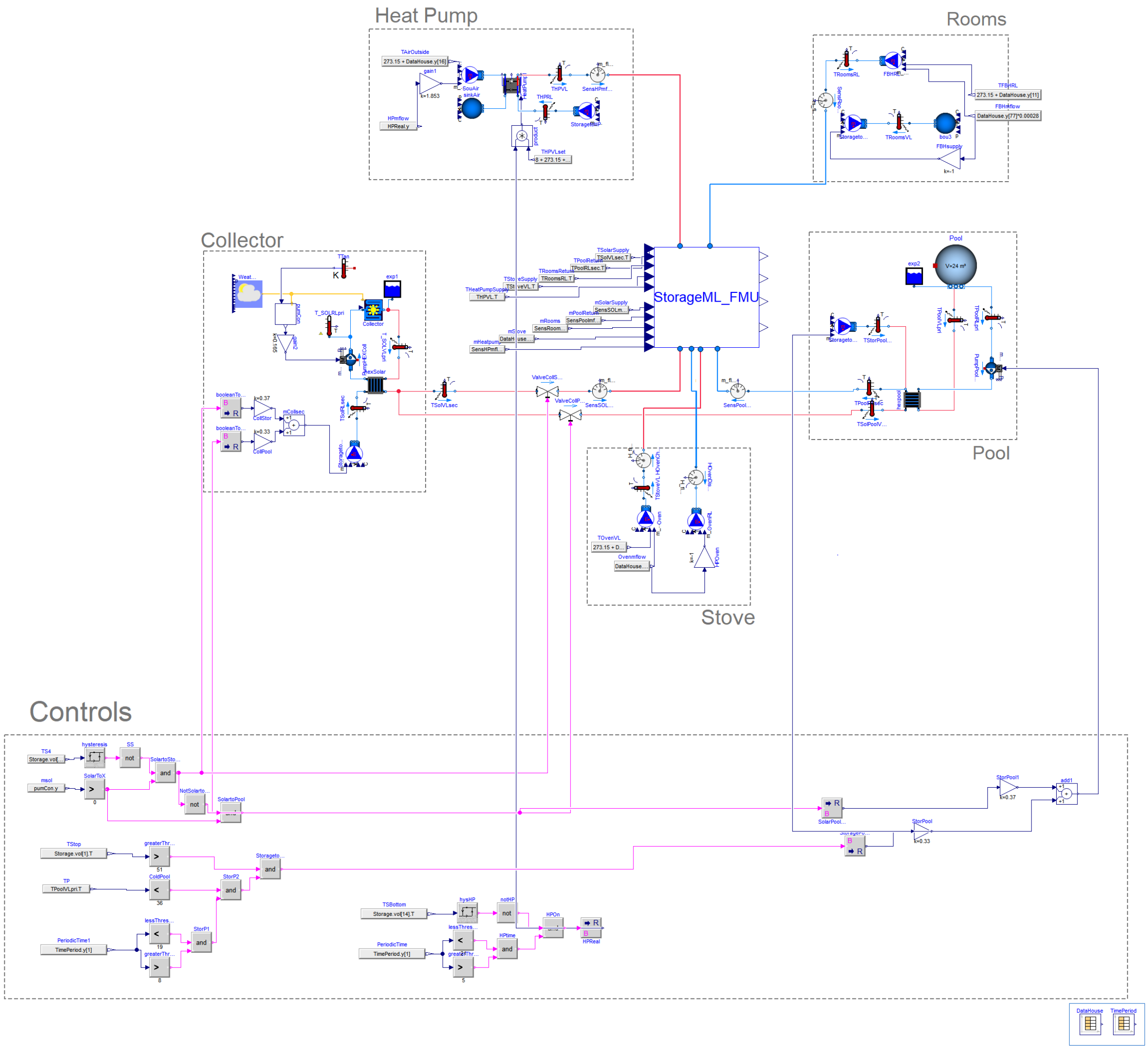

**Figure A.2.** Dymola layout of the single-family house heating system