eurac research

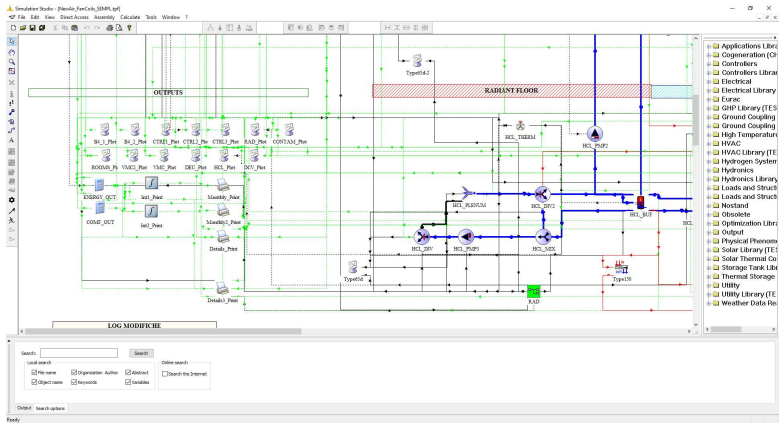
Meta models for load prediction toward optimisation of HVAC systems

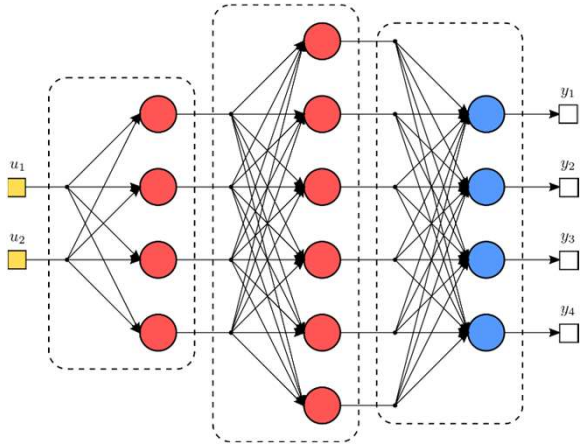
Anton Soppelsa

HP SIM workshop, Innsbruck, 30/06/2022

SHC jargon: models and meta-models

- Models are relatively detailed, first principles, white box models of the building envelope, TES, energy generators and energy distribution system
  - We use TRNSYS to develop models
- Meta-models are much less detailed models of the same objects developed for (predictive) control purposes
  - We are currently working on meta-models based on a neural network, (linear) state-space system or even (linear) IO mappings





Requirements of meta-models

- (Predictive) Accuracy
  - They need to have a certain accuracy when used to predict outcomes in what-if scenarios
- **Identifiability**
  - They need to be constructed from measured data
- **Adaptability**
  - They need to adapt to changing environment
- Agility
  - They need to be run (simulated) quickly even on hardware with limited memory and computational resources

To be useful in predictive controllers

To let the controller to create its own model on the field

To maintain the performances in face of changing environment

This is to target BEMS hardware

Development/Deployment process

- We start selecting one out of a number of reference buildings (envelope),HVAC systems (normally HP-based) and geographical location
- We generate waveform datasets selecting a small subset of the available variables
- We develop/apply the identification procedures on the dataset to obtain the meta-model
- The meta-model is embedded in the model predictive controller

$$\begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \end{cases}$$

Models structure

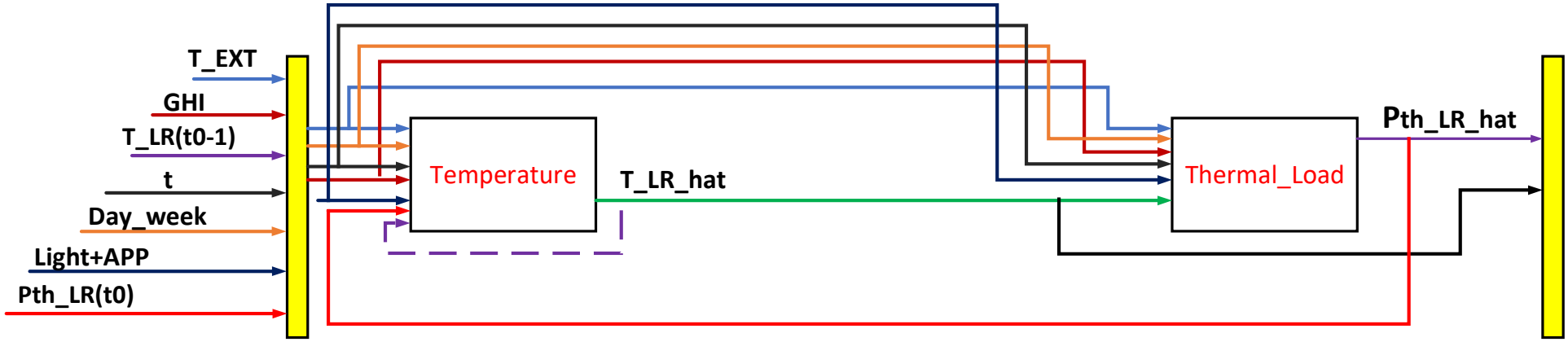
- Models are developed as functional blocks (FB) which is also python interface
- Roughly speaking FB is whatever can be cast in this form

$$\begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \end{cases}$$

- The FB class is quite broad and encompasses discrete LTI, non-linear and time-variant systems
- Non-linear system are implemented with an ANNModel object (which is itself a FB) using tensorflow.keras library tomake use of ANN features

Temperature and load model (I)

(thanks to our colleague M.H. Fouladfar)

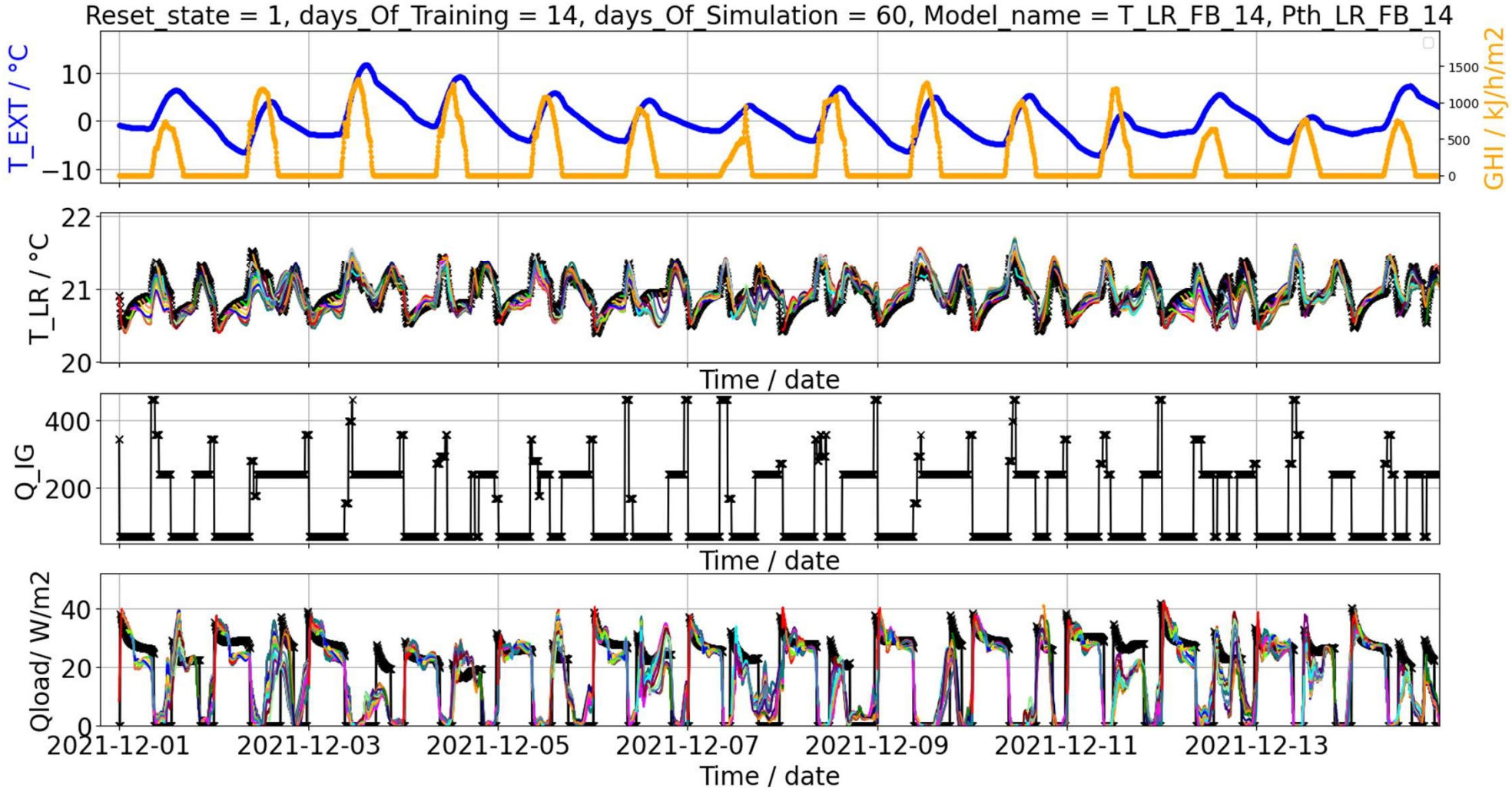


structure	trainingFraction	Reset state	For.Factor	D.Sampling	O.W.	Neurons	Activation	Training	Pred	Normalization
nlayers	last	1	0.25	3	64	16	relu	14 days	1st Dec to 31st January	checked (10%)

T_EXT, GHI, T_LR(t-1), t , day_week, Light+APP, RAD_QF_LR_hat
T_EXT, GHI, Light+APP, T_LR_hat, t, Day_week (Filter1 Neg-->0)

Valid	***Training***

Temperature and load model (II)



Temperature and load model (III)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2021-12-01	12.0	10.8	10.2	10.7	11.5	10.2	8.5	6.8	5.0	3.2	4.8	4.8	4.7	4.0	8.5	14.9	24.1	33.6	28.8	23.4	17.1	15.6	16.9	20.1
2021-12-02	12.4	10.7	10.1	10.7	12.0	12.1	11.0	9.6	8.3	5.7	6.9	9.6	8.2	5.5	3.6	6.3	6.5	6.5	3.8	-0.4	-2.3	-4.0	-1.4	-4.4
2021-12-03	-6.2	-5.8	-6.5	-6.2	-5.8	-6.6	-7.8	-8.4	-8.5	-11.0	-10.6	-7.2	-5.9	-5.6	-2.0	-1.2	0.1	0.5	2.6	3.4	5.3	10.7	12.3	11.9
2021-12-04	-0.6	-1.8	-1.9	-1.9	-1.3	-0.9	-0.3	0.4	1.0	2.3	2.0	4.7	6.3	5.7	4.3	2.8	3.9	3.0	0.6	4.4	5.0	5.0	5.5	6.0
2021-12-05	20.2	20.7	20.2	20.7	21.6	19.7	16.5	13.8	11.8	9.7	10.5	9.7	5.5	5.7	3.2	1.3	-0.6	-1.3	0.5	3.3	3.0	2.3	0.8	-0.6
2021-12-06	-1.3	-1.9	-1.9	-2.0	-1.4	-1.2	-1.7	-2.1	-2.4	-3.6	-0.8	-1.0	-5.7	-6.2	-7.1	-8.7	-11.2	-13.3	-15.9	-14.1	-9.2	-6.1	-2.8	0.0
2021-12-07	-6.2	-8.4	-9.1	-8.6	-8.0	-8.5	-9.1	-9.6	-9.8	-9.1	-6.1	-2.7	-2.4	-7.2	-7.6	-6.4	-8.1	-12.1	-11.9	-10.1	-8.2	-8.6	-8.4	-7.9
2021-12-08	-18.3	-21.8	-20.7	-19.2	-18.0	-17.6	-17.5	-17.2	-16.5	-16.2	-13.2	-11.6	-8.9	-7.2	-5.9	-4.7	-4.5	-1.0	2.9	3.0	-3.6	-8.8	-11.9	-12.5
2021-12-09	-3.1	-4.1	-4.4	-4.2	-4.2	-4.7	-4.8	-4.7	-4.4	-5.8	-2.5	-0.7	-2.0	-1.7	-2.3	-2.5	-2.9	-7.1	-7.9	-8.3	-9.1	-8.5	-6.5	-6.2
2021-12-10	-7.6	-11.1	-10.5	-9.4	-8.7	-9.4	-9.9	-10.0	-10.0	-12.3	-13.9	-11.4	-11.4	-15.6	-16.5	-19.5	-21.6	-21.6	-22.9	-24.3	-25.2	-24.6	-23.8	-24.2
2021-12-11	-15.2	-16.0	-15.3	-15.1	-14.4	-14.9	-15.8	-16.5	-17.0	-16.0	-14.6	-13.3	-9.9	-9.5	-9.9	-9.7	-10.8	-12.9	-13.2	-10.1	-9.4	-7.0	-5.6	-6.3
2021-12-12	6.5	5.5	4.6	5.0	5.2	4.4	3.2	2.1	1.2	0.3	-1.5	-1.9	-2.4	-3.7	-4.6	-7.9	-10.1	-11.4	-11.3	-11.1	-12.1	-12.0	-12.7	-13.2
2021-12-13	-9.0	-9.8	-9.7	-9.1	-9.3	-9.5	-9.6	-9.3	-9.0	-8.1	-8.1	-7.6	-4.1	-8.8	-10.7	-10.9	-9.9	-7.6	-6.1	-5.5	-6.4	-9.0	-8.3	-8.1
2021-12-14	-9.4	-10.1	-11.9	-12.1	-11.2	-10.8	-10.8	-10.6	-10.1	-8.0	-9.2	-8.3	-5.6	-8.1	-9.8	-9.9	-10.2	-9.9	-10.6	-8.2	-8.7	-6.3	-6.2	-6.2
2021-12-15	-12.9	-13.2	-14.6	-13.6	-11.2	-8.9	-7.5	-5.7	-3.2	0.7	1.8	2.8	5.2	6.5	8.0	9.1	11.2	11.7	9.3	10.5	15.9	16.0	15.3	15.9
2021-12-16	12.4	7.5	2.9	3.1	5.6	6.8	6.9	7.5	8.5	10.6	11.3	10.2	10.5	10.7	15.2	9.9	11.6	19.7	19.4	15.5	12.6	0.5	-10.9	-12.8
2021-12-17	-5.9	-4.1	-6.4	-6.3	-5.6	-4.4	-4.0	-2.9	-1.5	-0.3	2.5	2.8	2.9	-1.0	-2.1	-0.8	0.1	1.6	1.2	3.1	3.5	3.1	5.7	5.4
2021-12-18	-4.5	-6.8	-6.5	-4.1	-1.7	-0.0	1.0	1.8	2.7	3.5	5.3	5.8	8.2	9.8	12.6	5.4	5.5	5.8	3.3	-0.0	-7.8	-14.8	-16.2	-16.7
2021-12-19	-28.9	-32.1	-30.3	-28.8	-26.4	-24.1	-22.7	-20.9	-18.9	-14.8	-10.8	-9.7	-8.3	-4.9	-1.6	-0.8	0.6	-4.1	-3.4	-0.3	1.0	-1.3	-1.5	-1.3
2021-12-20	3.1	-1.7	-1.8	-0.4	1.2	1.8	1.2	0.1	-0.7	-2.6	-2.9	-2.7	-2.1	-1.4	-5.8	-11.8	-14.1	-13.1	-13.6	-15.8	-19.6	-25.9	-26.1	-25.1
2021-12-21	-12.8	-15.0	-14.8	-15.4	-16.7	-18.2	-19.2	-19.9	-20.4	-21.2	-22.8	-22.1	-21.7	-23.5	-23.0	-22.7	-23.5	-24.5	-24.0	-21.6	-19.6	-21.1	-20.8	-20.8
2021-12-22	-16.7	-20.6	-20.5	-19.9	-18.4	-17.3	-17.1	-17.0	-16.9	-14.1	-11.9	-9.4	-7.2	-8.3	-8.2	-7.9	-5.8	-9.7	-9.4	-5.9	-2.4	-4.0	-4.1	-3.9
2021-12-23	-21.0	-24.2	-23.5	-23.5	-24.2	-24.5	-24.6	-24.6	-24.8	-25.1	-25.7	-24.8	-25.3	-25.4	-25.4	-26.2	-28.0	-27.9	-27.0	-30.6	-28.0	-24.7	-22.4	-26.7
2021-12-24	-13.8	-16.1	-15.1	-13.8	-13.4	-12.5	-11.7	-10.2	-8.8	-8.9	-11.3	-9.4	-9.4	-12.6	-16.0	-17.6	-17.5	-17.7	-16.4	-15.7	-15.3	-15.1	-14.2	-13.1
2021-12-25	-3.7	-6.4	-5.7	-4.9	-3.0	-0.4	1.2	2.5	3.2	4.1	1.8	-2.9	-5.7	-5.4	-2.2	-2.7	-4.1	-4.7	-3.5	-2.0	0.7	1.9	2.4	1.7
2021-12-26	14.9	10.4	10.1	10.6	8.7	4.8	1.4	-1.2	-3.3	-3.5	-4.9	-5.3	-7.0	-8.4	-10.5	-12.6	-13.7	-14.2	-15.0	-13.5	-13.1	-13.1	-13.2	-13.9
2021-12-27	-11.8	-14.8	-14.9	-13.5	-12.4	-12.3	-12.4	-12.9	-13.5	-12.7	-11.2	-8.1	-9.5	-6.6	-3.0	-1.1	-0.7	-1.7	-1.8	-1.6	-1.3	-8.3	-10.8	-6.5
2021-12-28	-1.7	-3.8	-6.1	-5.6	-5.0	-4.8	-4.8	-5.0	-5.3	-6.8	-7.3	-6.0	-8.4	-9.9	-10.2	-11.5	-12.2	-12.4	-9.2	-6.4	-6.9	-15.2	-14.1	-13.8
2021-12-29	-2.6	-3.0	-5.2	-5.3	-4.3	-4.1	-4.3	-4.7	-4.9	-4.1	-2.2	-3.1	-3.4	-6.3	-5.9	-2.0	-1.2	-1.9	-3.3	2.1	0.6	0.1	-7.3	-11.0
2021-12-30	0.3	4.9	4.1	3.4	4.5	4.9	4.7	4.6	4.3	3.9	3.3	8.4	7.8	8.0	7.1	8.9	13.4	15.2	14.0	11.9	16.9	23.0	23.4	29.4
2021-12-31	33.9	31.0	29.4	28.0	28.4	28.9	29.0	29.7	29.2	28.5	28.4	31.6	33.6	33.7	30.3	26.1	26.2	31.6	40.4	50.8	53.8	53.7	50.0	49.8
2022-01-01	52.9	52.7	53.8	57.7	60.6	63.2	65.9	68.3	70.5	72.1	69.8	75.6	73.3	74.2	61.8	50.6	42.6	36.0	32.7	27.5	25.5	24.1	28.6	23.1
2022-01-02	-9.2	-13.5	-14.6	-14.8	-14.2	-13.1	-12.0	-10.6	-9.2	-8.0	-9.0	-7.5	-5.2	-6.7	-6.5	-4.6	-2.3	-1.7	-0.2	1.8	3.0	-0.9	-1.6	-1.7
2022-01-03	8.3	8.0	7.9	7.2	6.2	5.4	4.5	3.7	3.1	2.1	2.7	5.3	15.9	16.0	8.2	6.0	7.5	6.7	5.1	3.4	3.3	2.5	2.4	2.0
2022-01-04	2.2	-0.8	-0.5	-0.0	-0.6	-0.8	-0.8	-0.6	-0.5	0.5	-4.8	-2.6	-1.3	3.1	1.4	0.2	0.1	0.5	0.2	-1.7	-1.9	-3.8	-5.3	-6.9
2022-01-05	-10.0	-11.3	-11.3	-11.3	-11.4	-11.5	-11.9	-12.6	-13.2	-10.9	-9.8	-8.0	-6.6	-8.8	-11.7	-12.2	-10.0	-8.1	-8.1	-9.4	-12.2	-11.9	-12.1	-12.6
2022-01-06	6.7	7.1	6.3	6.2	6.0	5.6	5.3	4.7	4.0	4.1	3.3	6.0	3.6	3.1	4.4	5.3	3.1	8.1	12.6	8.6	-0.2	-3.3	-3.1	-1.8
2022-01-07	-9.7	-11.6	-13.5	-13.7	-13.4	-13.3	-13.2	-12.8	-12.3	-11.7	-13.1	-12.8	-11.8	-12.0	-12.4	-9.9	-12.8	-7.6	-8.0	-11.8	-8.7	-6.7	-8.3	-14.1
2022-01-08	-12.1	-12.5	-12.9	-12.3	-11.9	-11.6	-11.6	-11.4	-11.4	-11.8	-14.2	-13.3	-12.4	-14.6	-19.4	-18.3	-17.5	-13.4	-12.3	-15.2	-17.0	-13.6	-10.4	-10.1
2022-01-09	17.0	14.4	13.3	14.1	13.2	12.0	11.4	10.5	9.6	10.8	9.9	10.4	14.3	10.5	8.3	7.5	6.2	8.6	7.9	7.2	7.7	4.0	3.1	2.4
2022-01-10	-0.2	-0.5	-0.8	-1.2	-0.9	-0.8	-1.0	-1.3	-1.8	2.4	1.3	1.5	4.0	5.9	8.0	10.1	10.4	11.4	14.1	15.3	10.9	8.6	7.7	6.8
2022-01-11	20.8	16.1	16.3	17.6	17.2	16.3	16.1	16.8	17.9	16.6	17.4	17.4	19.3	15.3	12.1	9.1	9.4	9.3	11.8	15.4	13.4	14.1	12.9	15.6
2022-01-12	19.7	18.2	16.6	16.0	15.8	15.4	15.7	15.9	15.8	15.8	18.0	19.3	18.2	17.6	16.1	14.4	13.2	16.2	21.1	23.4	21.4	14.9	9.1	10.3
2022-01-13	3.8	0.3	-0.3	0.0	-0.1	-0.3	-0.9	-2.3	-3.6	-2.7	-2.4	-1.9	-1.8	-0.2	-2.2	0.7	-0.9	-6.9	-1.9	-0.2	0.2	-2.9	-0.1	-0.1
2022-01-14	-26.5	-29.6	-29.2	-28.2	-28.2	-28.3	-28.1	-28.1	-27.7	-27.2	-28.7	-28.7	-28.2	-28.0	-27.4	-24.3	-25.7	-27.4	-26.1	-24.7	-25.0	-26.1	-25.3	-25.0
2022-01-15	-2.4	-5.4	-5.0	-4.1	-4.2	-4.2	-3.6	-3.2	-3.1	-4.0	-4.2	-3.9	-3.7	-2.3	-1.9	-5.0	-6.7	-5.4	-2.5	-5.3	-7.5	-7.9	-9.4	-10.0
2022-01-16	-13.0	-14.5	-12.4	-12.5	-12.7	-13.0	-12.7	-12.6	-12.2	-11.4	-12.4	-12.9	-10.6	-11.1	-14.1	-14.3	-10.9	-10.7	-11.3	-11.9	-11.5	-12.1	-11.8	-13.2
2022-01-17	1.8	4.7	4.5	4.7	4.7	4.8	5.1	4.9	4.5	4.7	6.0	5.8	8.8	7.3	7.0	7.3	7.0	4.1	3.0	3.2	4.0	3.3	2.8	2.5
2022-01-18	-2.4	-2.4	-3.1	-3.0	-2.8	-2.7	-2.3	-2.1	-1.8	-1.5	-0.5	-2.5	0.5	0.7	1.1	1.6	4.0	5.9	4.1	4.5	4.9	5.9	5.2	3.8
2022-01-19	-5.5	-8.5	-8.9	-8.1	-7.3	-6.5	-5.7	-5.0	-4.1	-2.3	0.0	0.4	1.8	4.5	1.9	-2.6	-4.7	-5.8	-4.8	-3.8	-8.6	-12.6	-12.8	-13.0
2022-01-20	0.3	-0.4	-0.4	-0.1	-0.6	-0.8	-0.3	0.2	0.6	0.0	3.2	7.1	8.2	6.7	6.3	7.6	6.6	3.7	7.4	12.8	18.2	22.0	22.9	24.5
2022-01-21	22.2	16.4	15.9	16.8	17.8	18.2	18.7	18.9	18.4	14.4	14.6	14.0	15.4	16.4	17.6	19.1	22.7	20.7	15.0	14.6	15.8	18.6	25.3	19.2
2022-01-22	-5.0	-9.1	-6.8	-6.2	-5.3	-3.9	-2.5	-1.2	-0.8	-3.2	-6.0	-6.1	-5.9	-5.5	-3.9	-5.5	-4.9	0.3	0.5	1.5	-0.3	-1.1	-0.8	-0.8
2022-01-23	10.0	12.3	12.3	11.5	10.4	9.7	9.6	9.1	8.9	11.0	11.7	12.3	5.5	2.6	-0.1	1.6	0.8	-0.7	0.3	-0.1	1.1	0.9	0.6	0.9
2022-01-24	-4.4	-6.3	-6.9	-7.0	-7.1	-7.1	-6.8	-6.8	-6.7	-3.4	-2.9	1.1	0.3	3.5	0.9	0.7	0.0	-0.5	-0.5	1.4	-3.4	-5.8	-6.1	-6.6
2022-01-25	17.9	14.8	14.4	14.6	14.0	13.1	12.7	11.8	10.4	10.2	8.6	7.8	9.4	9.0	8.2	7.3	7.5	8.0	8.6	6.6	3.5	3.0	3.3	2.5
2022-01-26	-14.0	-20.6	-19.7	-17.3	-16.7	-16.2	-14.7	-14.7	-14.6	-17.8	-17.3	-16.4	-13.1	-13.4	-15.1	-15.7	-15.9	-13.7	-9.8	-8.0	-8.6	-13.3	-13.6	-13.5
2022-01-27	0.1	-5.6	-4.8	-3.5	-3.4	-3.1	-2.0	-1.7	-1.4	-1.3	0.9	1.4	0.8	1.1	-1.5	-2.0	0.3	3.4	3.5	5.1	6.3	5.6	4.9	5.3
2022-01-28	-13.6	-20.2	-19.3	-18.3	-18.4	-18.4	-17.9	-17.7	-17.5	-17.8	-14.2	-12.0	-12.8	-11.8	-13.5	-12.6	-11.1	-11.1	-12.3	-13.3	-20.8	-26.0	-28.4	-27.0

Embedding in the optimal contol problem

- Possible optimization mechanisms:
  - Using Particle Swarm Optimization
    - This has the advantage of being applicable even if the models have not been developed in a symbolic framework
    - The only requirement is that to model outputs needs to simulate **really fast**
      - Because the PSO will need to do several thousands of simulations to find the optimal trajectory
  - Using a "canonical" formulation of the optimization problem

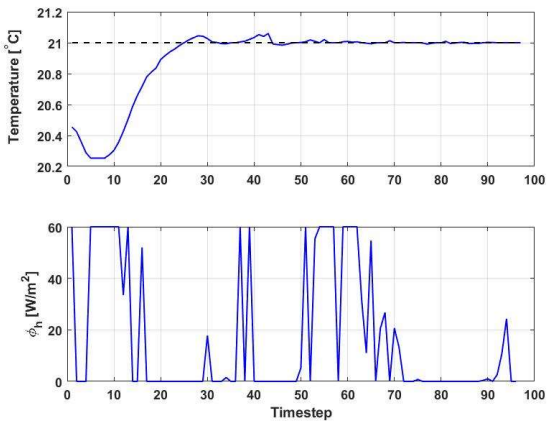
$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{t=0}^{N-1} L(\boldsymbol{x}(i), \boldsymbol{u}(i), \boldsymbol{y}(i))$$

$$\text{s.t.} \begin{cases} \boldsymbol{x}(t+1) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \end{cases}$$

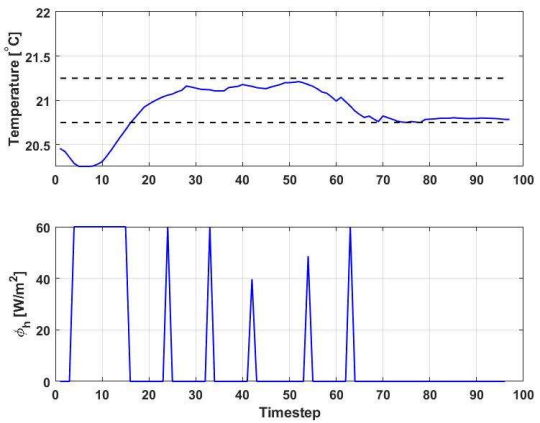
Optimal control using canonical formulation

- The ANN-based functional block has been translated manually in a symbolic model within the Gekko framework (thanks to our colleague H. Nagpal)

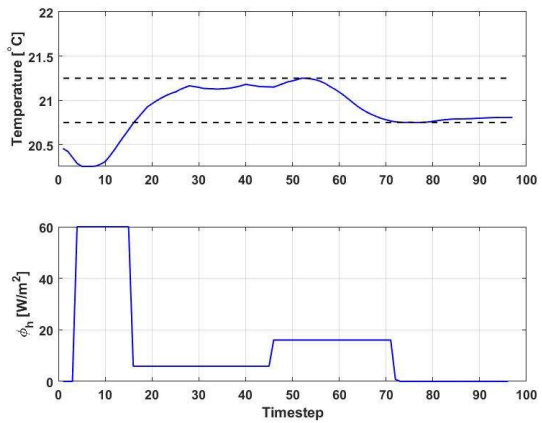
The controller aims to follow the given reference set point (21 °C).



The controller tries to maintain the temperature between 20.75 and 21.25 °C.



With a penalization on rate of the thermal power input, producing smoother results.



Tools

- TRNSYS is our modelling language for white box models
- Python is our development language for control design/optimization
  - Toolboxes we currently use include:
    - cvxpy, pyomo, Gekko and CasADi for optimization
    - TensorFlow for ANN
    - Self-developed library managing compound functional block models

Current activities

- Still to be decided if it is better to go for the PSO or symbolic optimization
- For symbolic optimization we need to be sure that the translation can be done and that can be done automatically
- Trying again to develop MIMO models using ANN, which proved difficult at the beginning of the project
- Trying to settle the framework for optimization (suggestions?)

Thank you for your attention

Anton Soppelsa

Institute for Renewable Energy
Anton.soppelsa@eurac.edu

www.eurac.edu
www.noi.bz.it

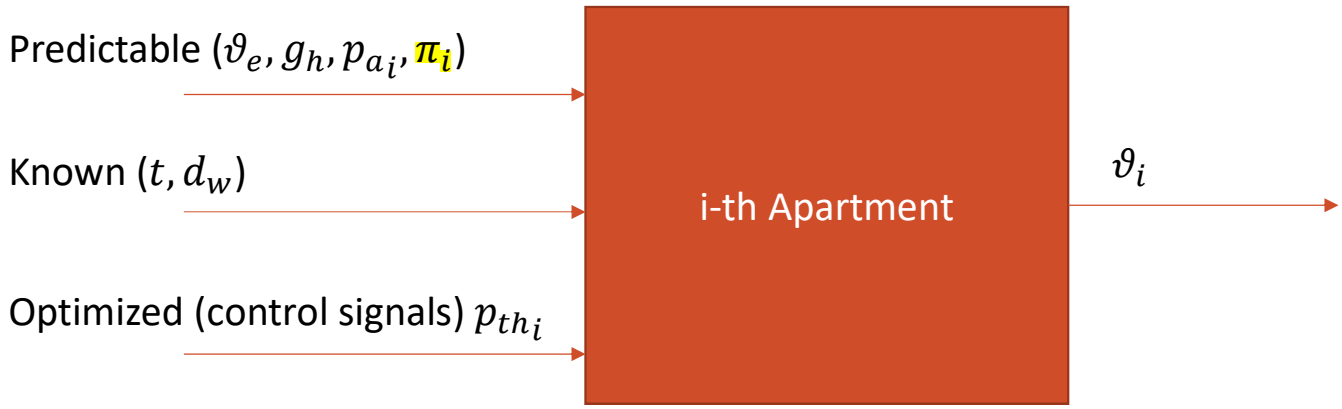
```
562 class FunctionalBlock(Verbosable):
563     """
564     FunctionalBlock is an interface (or abstract class) to implement dynamic
565     models.
566
567     This interface supports standard discrete-time state space models, that is
568     those in the form:
569
570         x(t + 1) = f(x(t), u(t))
571         y(t)     = g(x(t), u(t))
572
573     where m is the size of the input vector (u(t)) or number of input variables,
574     n is the size of the state vector (x(t)) or the number of state variables and
575     p is the size of the output vector (y(t)) or the number of output variables.
576
577     Additionally, the interface also supports models where an extended state
578     is constructed. Given two positive or at least equal to zero integeres qx >= 0
579     and qu >= 0 the extended state csi at time t is defined to be:
580
581         csi(t) = [x(t - qx), x(t - qx + 1), ..., x(t - 1), x(t), u(t - qu), ...; u(t -
582
583     where the operator [a, b] performs the juxtaposition of vectors a and b,
584     for example the vectors a = [0., 1., 2.] and b = [0.5, 0.5] becomes
585     [a, b] = [0., 1., 2., 0.5, 0.5]
586
587     Note that the size of the extended state is (qx + 1)*n + qu*m. Note also
588     that components of the extended state vector from index 0 to index
589     (qx + 1)*n - 1 pertain to the so called state variables and their past
590     values while the components from (qx + 1)*n to (qx + 1)*n + qu*m - 1
591     pertain to the past values of the so called input variables.
592     The current state variables are immersed in the extended state vector and
593     are found at from index qx*n to index (qx + 1)*n - 1.
594
595     This allows to implement systems in slightly different form, for example
596
597         x(t + 1) = f(x(t), x(t - 1), ..., x(t - qx), u(t), ... u(t - qu))
598         y(t)     = g(x(t), x(t - 1), ..., x(t - qx), u(t), ... u(t - qu))
599
600     or other more specific forms. This has been foreseen for the application
601     of ANN models or other models derived using machine learning.
602
603     Nearly complete freedom in the construction of the parameters on which
```

Source code of the functional block interface

ROM of an apartment (not including generators or storages) ROM-A

It is a dynamic system with the following inputs and outputs

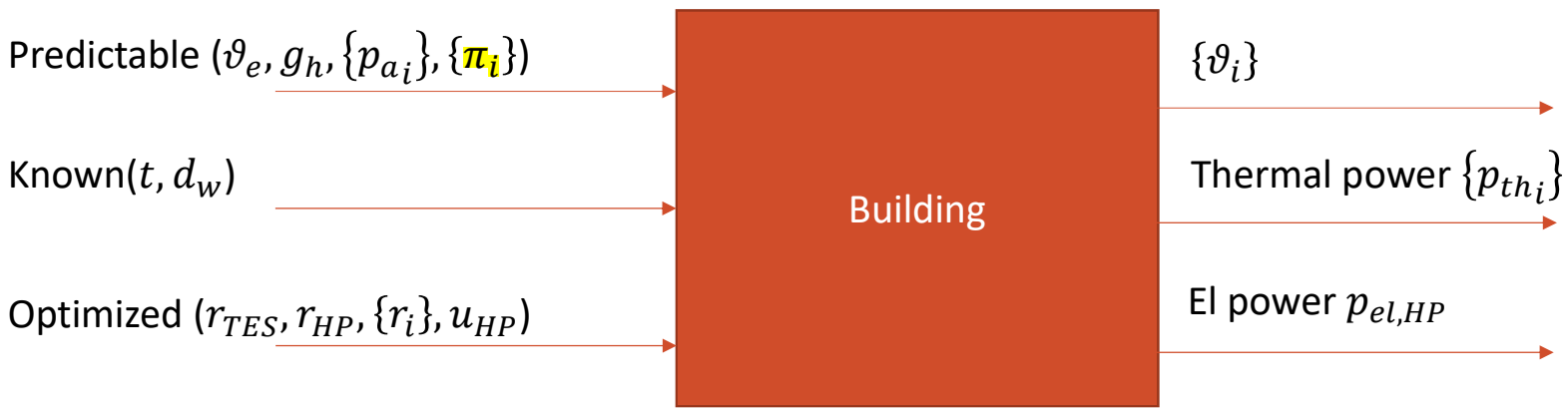
- Inputs:
  - Predictable: outdoor temperature, global horizontal irradiation, appliances power, presence of people? (likely not measurable on a demo)
  - Known: seconds of the day and day of the week (if those are really needed it means that the system is time-variant)
  - Optimization variables: **input** thermal power
- Output:
  - temperature in the apartment



ROM of a building including generators, storages and apartments

It is a dynamic system with the following inputs and outputs

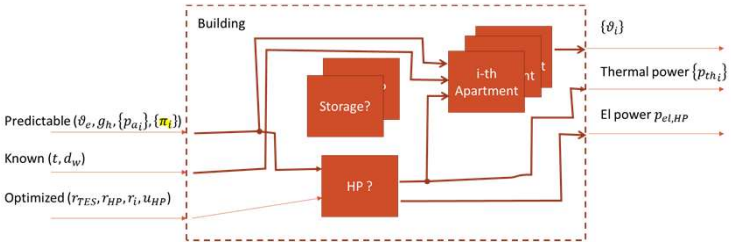
- Inputs:
  - Predictable : outdoor temperature, global horizontal irradiation, appliances power, presence of people? (likely not measurable on a demo)
  - Known : seconds of the day and day of the week
  - Optimization variables: temperature reference signal (TES reference **OR** HP reference **OR** apartments reference) **OR** HP on/off



How can we build the building ROM?

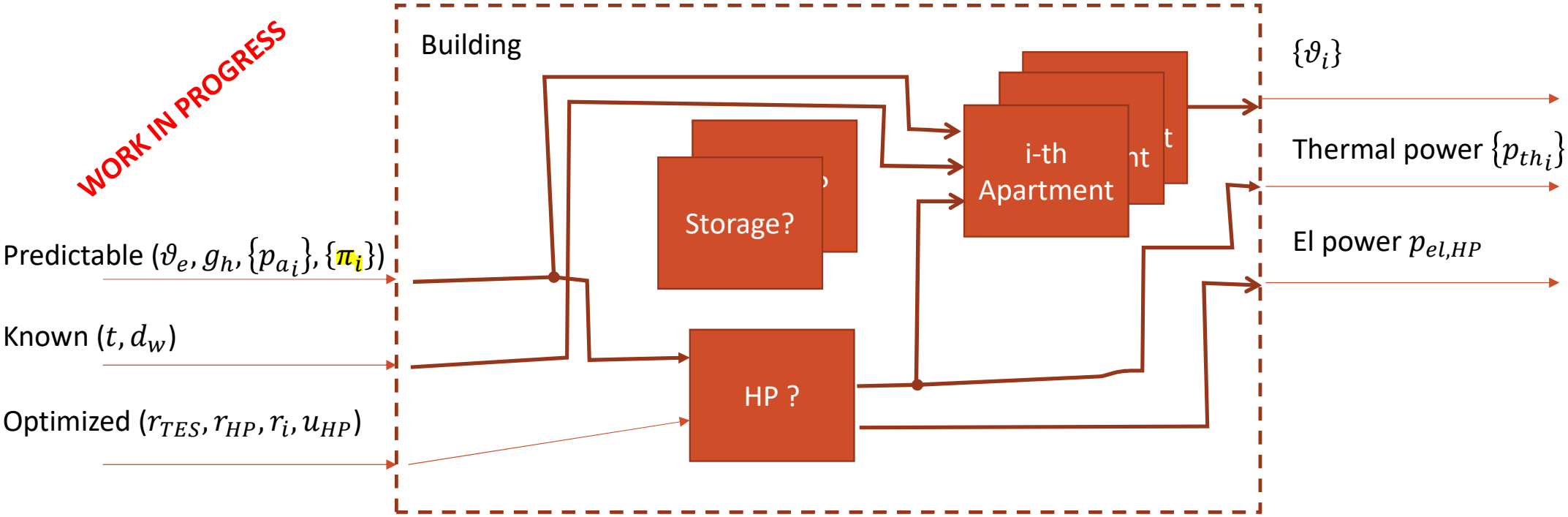
- One option could be training a big neural network with multiple outputs
  - We could call this approach top-down (or holistic)
  - I think our results shows that this is difficult/may require lots of data

- Another option is building a compound model trying to take advantage of the models developed so far
  - We could call it bottom-up approach (or reductionistic)



Top-down ROM model of the building

- Not discussed

Bottom-up ROM model of the building

- Note that here by building I mean the structure, the envelope and the energy systems (alike a body is made by the skeleton, skin and internal organs)