

Towards Equation-Based Building Energy System Modeling in Julia:
A Modular Prototype and Verification Study

Ruizhi Song, Mingzhe Liu, Zhiyao Yang, Zheng O'Neill*

J. Mike Walker' 66 Department of Mechanical Engineering, Texas A&M University

American Modelica Conference
Modelica & FMI
Atlanta USA 2026

Background

Modeling workflows

- Building energy system models for calibration, optimization, and learning

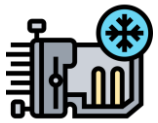
Physical model → **Computational tasks**



Weather



Envelope + zone

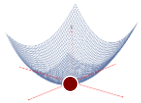


Air + water loops



Calibration

Temperature sensitivity to thermal parameters



Optimization

Temperature / energy sensitivity to control inputs



Learning

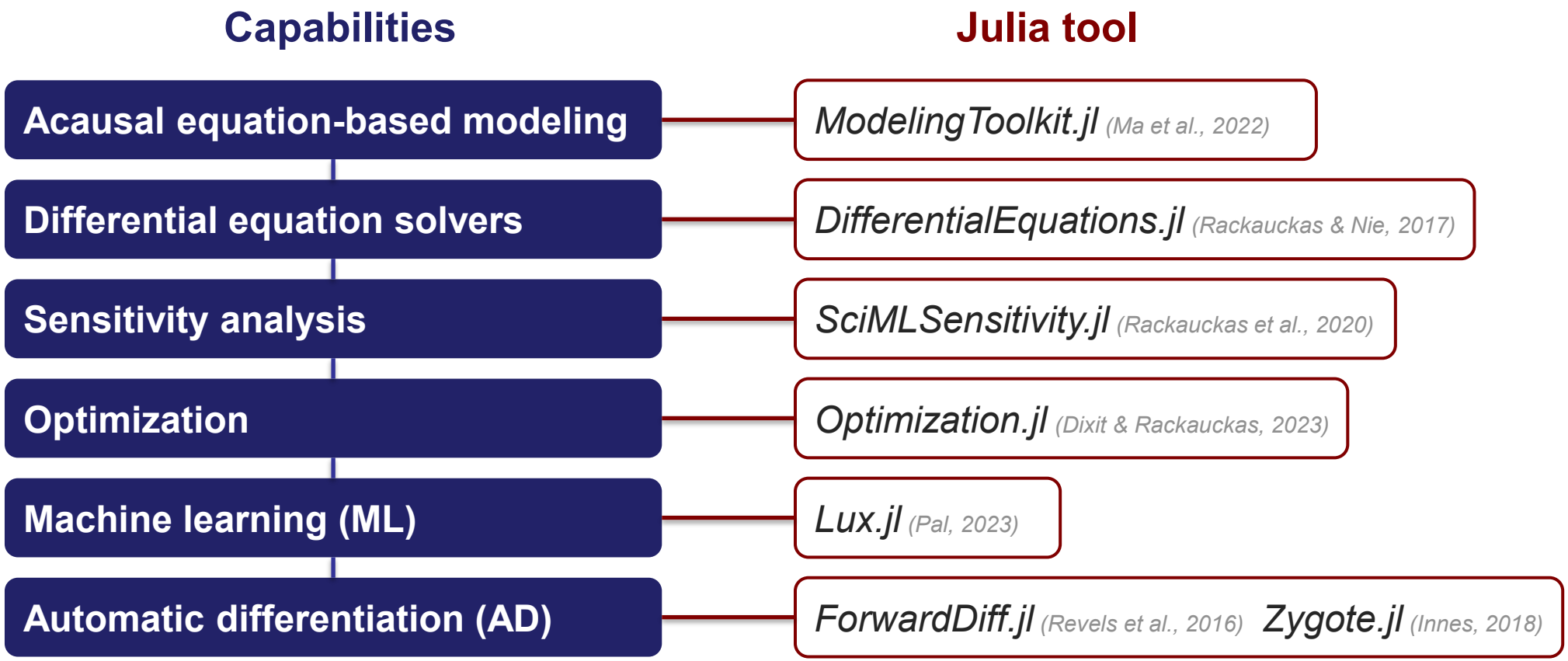
Surrogate training / physics-informed ML

∂ **Efficient derivatives**

Motivation: Single environment

Julia ecosystem

- **Julia**: High-level **scientific computing** with LLVM-based JIT compilation and multiple dispatch *(Bezanson et al., 2017)*



Non-exhaustive

Existing work

➢ Existing work

- **Modelica**: Buildings Library for equation-based building energy system modeling *(Wetter et al., 2014)*
- **Julia**:
  - High-fidelity vapor-compression-cycle:
    - state estimation *(Deshpande & Laughman, 2023)*, sensitivity analysis *(Laughman et al., 2023)*, surrogate modeling *(Rackauckas et al., 2022)*
  - System-specific HVAC simulation with limited modularity and generalizability *(Song et al., 2025)*

➢ Gap

- Reusable modeling support for building domain in Julia, facilitating full SciML pipeline exploration.

➢ Contribution of this work

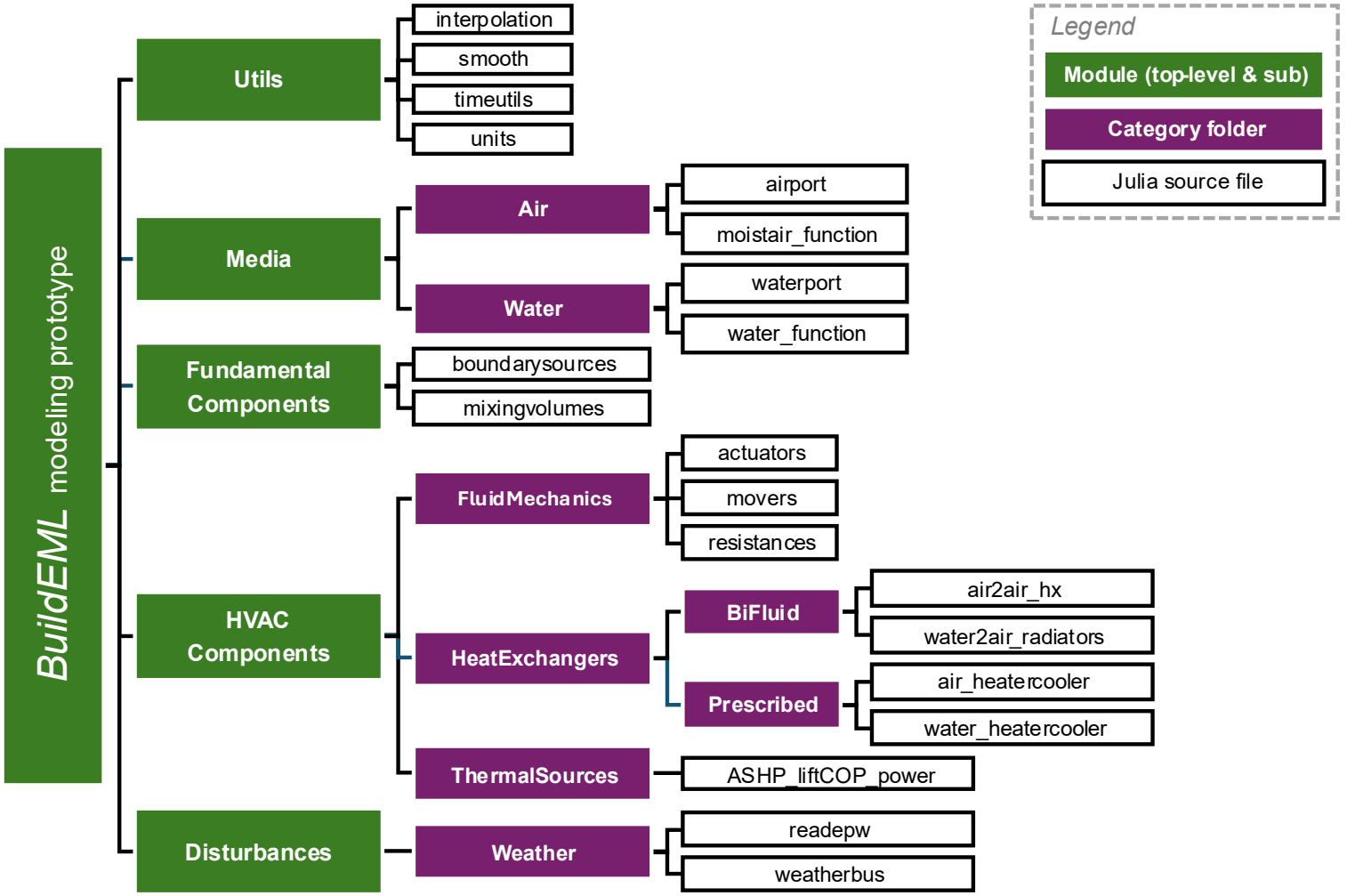
- ❑ An open-source **lightweight** building energy **modeling prototype** BuildEML.
- ❑ Verification against Modelica Buildings Library's **SimpleHouse** cases.

Methods

Prototype design

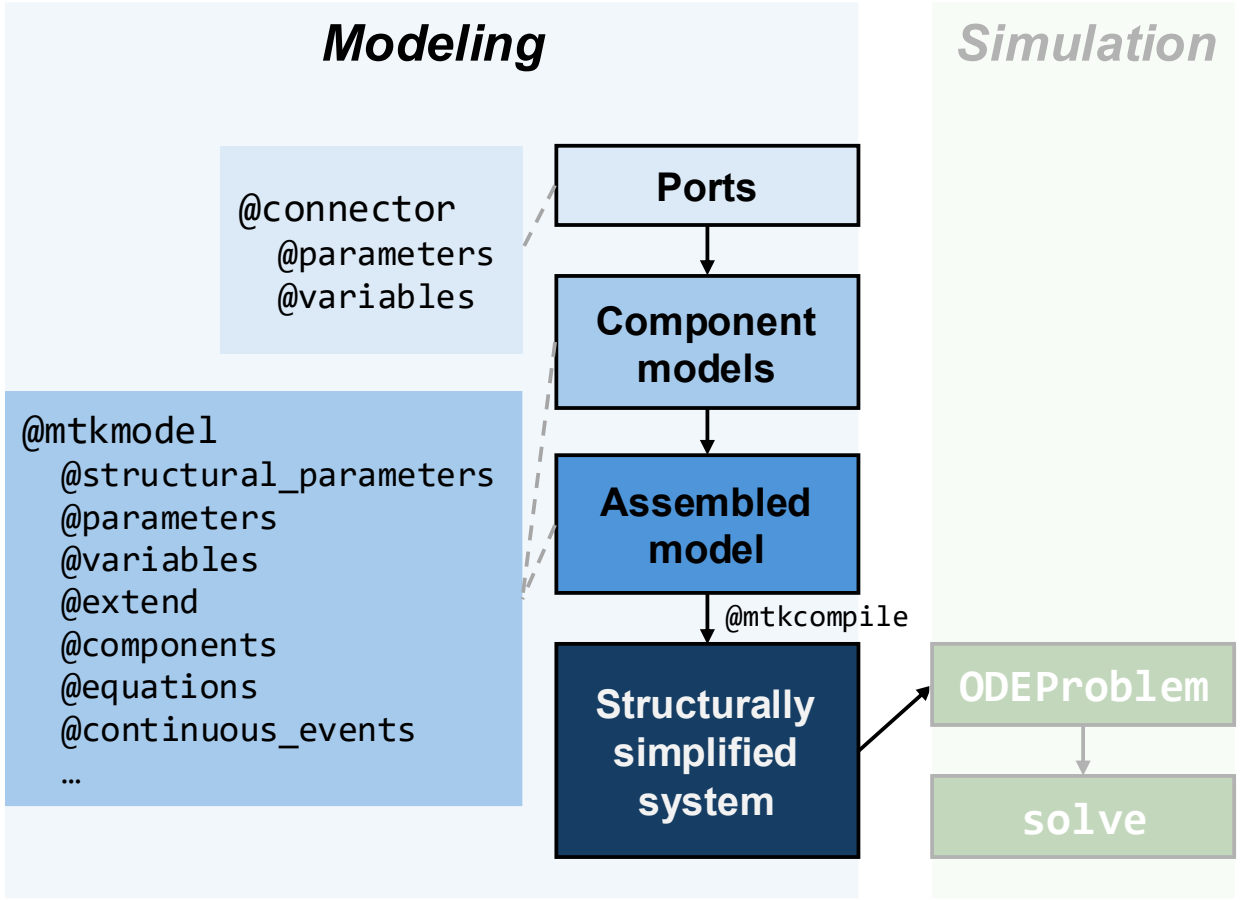
github.com/BE-HVACR/BuildEML.jl/src

- **Submodules**
  - Utils
  - Media
  - FundamentalComponents
  - HVACComponents
  - Disturbances



Modeling and simulation

- Symbolic model construction with *ModelingToolkit.jl* *v10.21*




```
@mtkcompile sys = SimpleHouse3(...)

prob = ODEProblem(sys, u0, tspan)

sol = solve(prob, Rodas5P();
reltol = 1e-6)
```

Ports

- HVAC-oriented connectors with temperature-based stream variables

AirPort (dry air-based)

```
@connector AirPort begin
    @variables begin
        p(t)
        mflow(t), [connect = Flow]
        T_ofo(t), [connect = Stream]
        w_ofo(t), [connect = Stream]
    end
end
```

WaterPort

```
@connector WaterPort begin
    @variables begin
        p(t)
        mflow(t), [connect = Flow]
        T_ofo(t), [connect = Stream]
    end
end
```

Air psychrometric functions: e.g.

```
h_Tw(T,w) = cp_da * (T - 273.15) + w * h_g(T)
```

- Variable-type semantics
  - `Regular`: equalized across connected ports.
  - `Flow`: Σ mflow = 0; positive into a component.
  - `Stream`: upstream values are accessed using `instream()`.

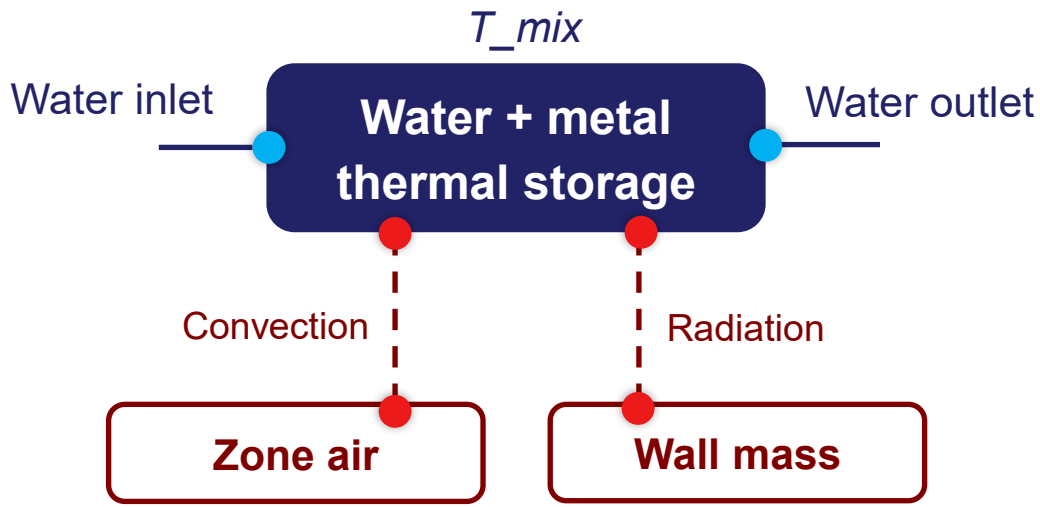
Component + assembled model

- E.g.: A radiator element: reuse a water-volume model and add heat-transfer equations

Component model

```
@mtkmodel RadiatorSingleElement begin
    @extend port1, port2, T_mix, heatport =
        vol = WaterMixingVolumeNode2(V = V_element)

    @equations begin
        Q_con ~ (1 - fraRad_) * UA_element *
            dT_con * smooth_abs(dT_con)^(n_exp_ - 1)
        Q_metal ~
            (M_metal_element * cp_metal_) * D(T_mix)
        heatsource.Q_flow.u ~ -(Q_con + Q_rad + Q_metal)
        heatPortCon.Q_flow ~ -Q_con
        heatPortRad.Q_flow ~ -Q_rad
    end
end
```



System assembly

Heat

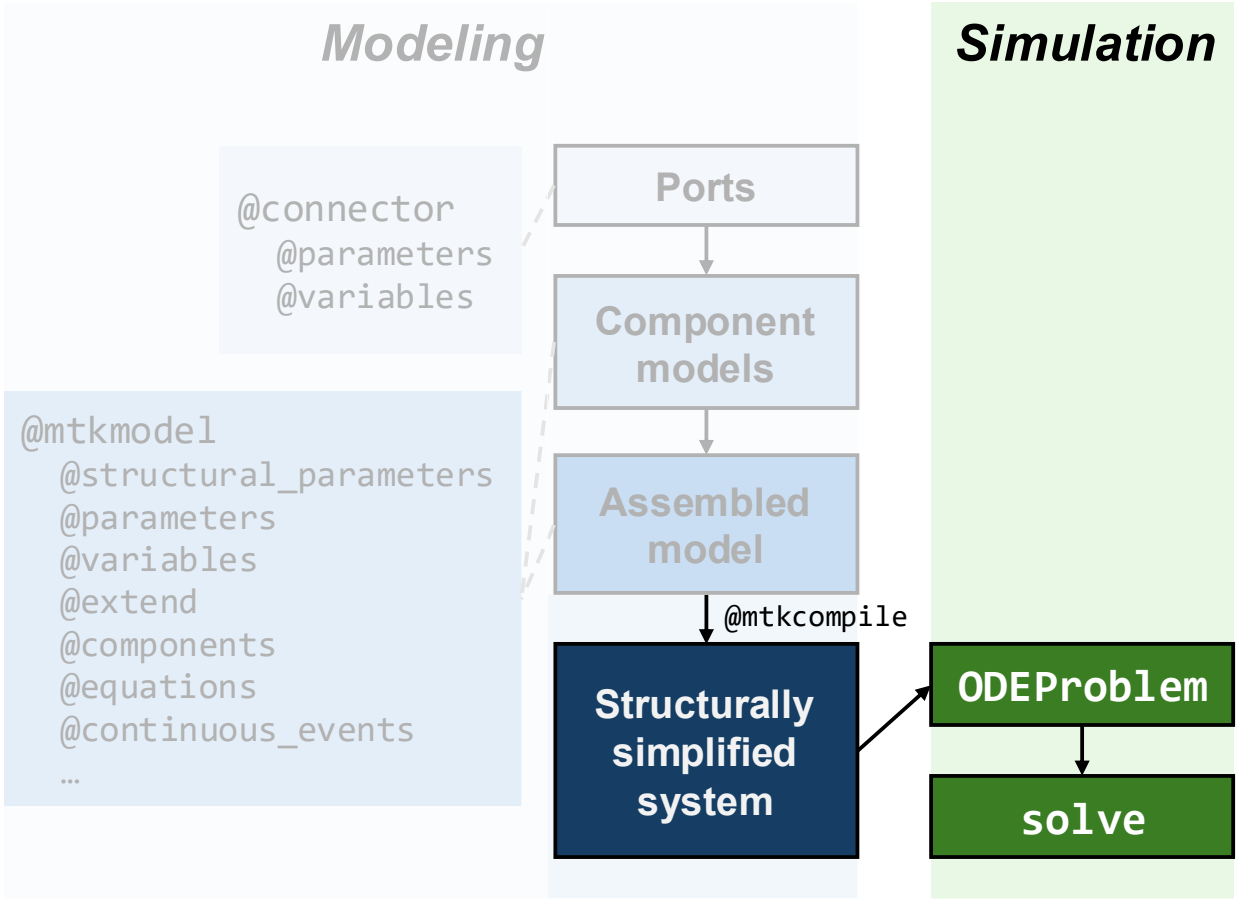
```
connect(rad.heatPortCon, zon.heatport)
connect(rad.heatPortRad, walCap.port)
```

Flow

```
connect(heaWat.port_b, rad.port_a)
connect(rad.port_b, pum.port_a)
```

Modeling and simulation

- Symbolic model construction with *ModelingToolkit.jl v10.21*
- Numerical simulation with *DifferentialEquations.jl v7.17*




```
@mtkcompile sys = SimpleHouse3(...)

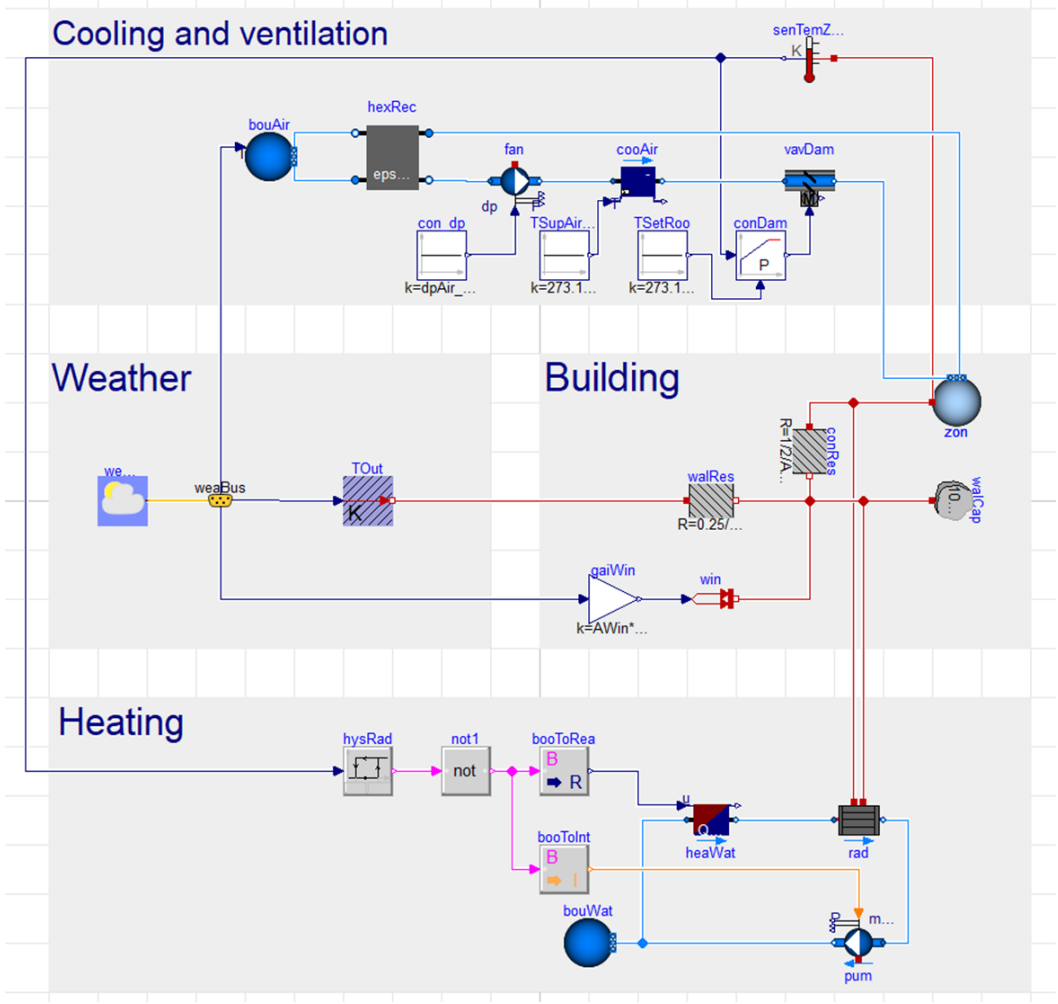
prob = ODEProblem(sys, u0, tspan)

sol = solve(prob, Rodas5P();
reltol = 1e-6)
```

- The compiled system and numerical problem are **Julia objects** accessible to surrounding code.

Verification reference model

- Modelica Buildings Library v11: `Buildings.Examples.Tutorial.SimpleHouse`
  `Buildings.Examples.SimpleHouse`



Weather

Weather-file inputs drive the thermal and HVAC subsystems.

Building

RC envelope, window solar gains, and zone-air thermal storage.

Heating

Water heater, pump, and radiator with hysteresis control.

Cooling and ventilation

Heat recovery, sensible cooling, fan, damper, with PID-control.

Verification reference model

- Four reference cases of increasing system complexity

MBL reference	System scope	Compared outputs
SimpleHouse0	Weather-file reading and interpolation	Outdoor temperature / irradiance
SimpleHouse3	Envelope + window gains + free-floating zone	Zone temperature
SimpleHouse5	Case 3 + hydronic heating / hysteresis (21–23 °C)	Zone temperature
Standalone	Heating (20–22 °C) + ventilation / PID cooling	Zone temperature

Solver

- This work: Rodas5P
- MBL: DASSL, Dymola 2023
  - tolerance of 1e-6
  - output interval of 900 seconds

Common comparison conditions

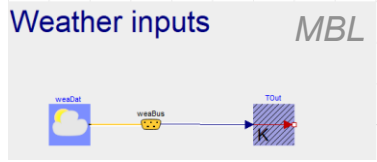
Weather file: Chicago O'Hare TMY3
Evaluation metrics: signed (relative) error, RMSE, MBE

Matched physical settings within each case. Internal heat gains and latent control are excluded.

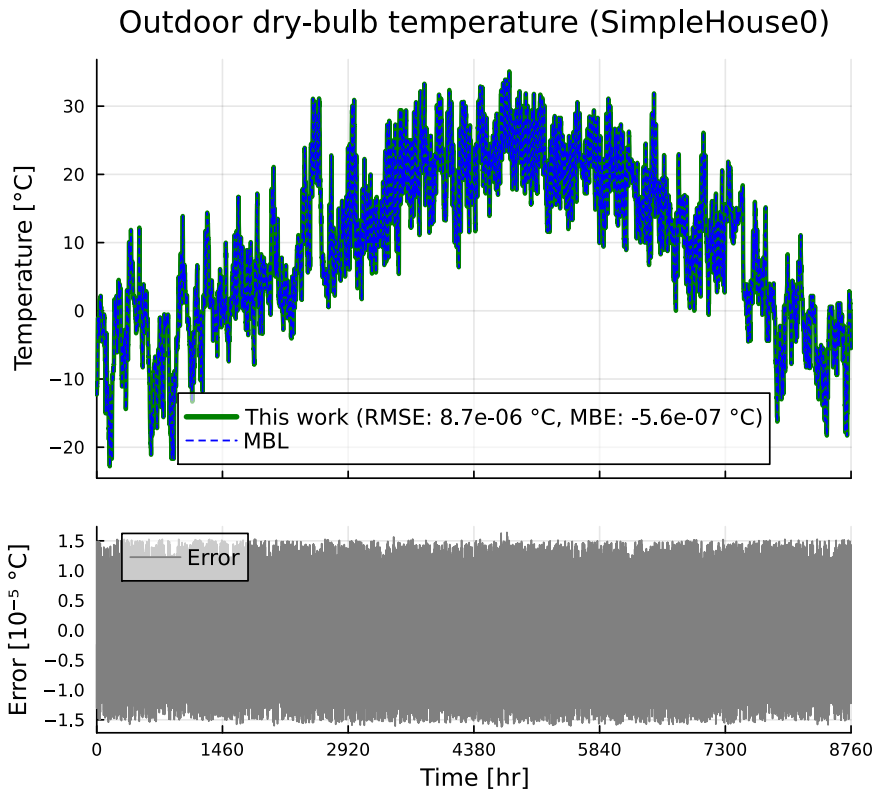
Results

Result: Weather | SimpleHouse0

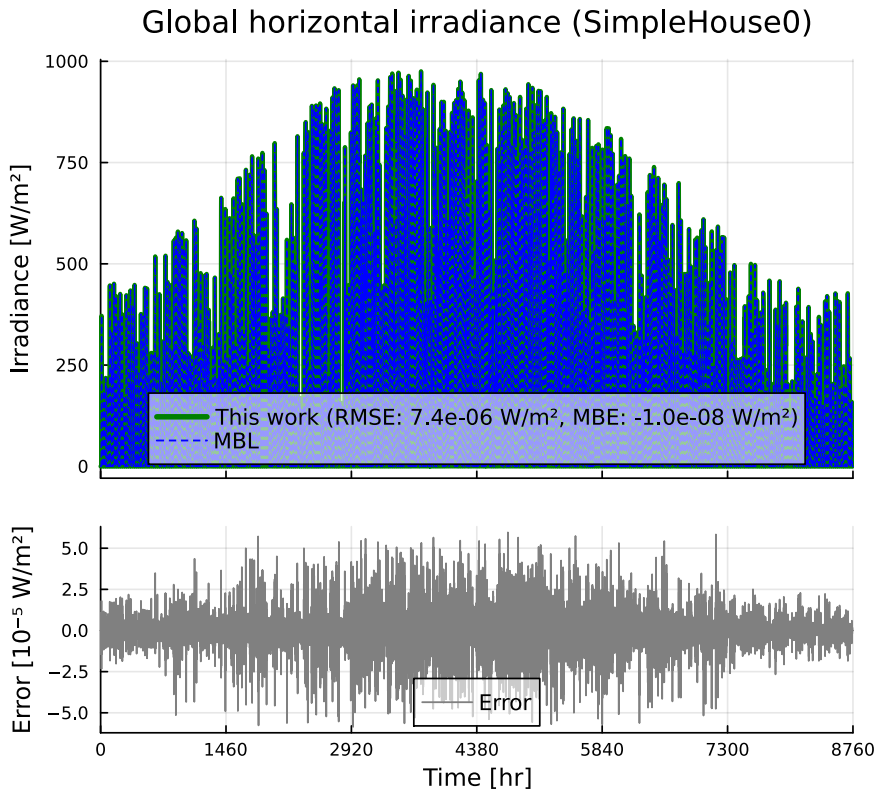
- EPW reading, time alignment, and Akima interpolation



Outdoor dry-bulb temperature



Global horizontal irradiance

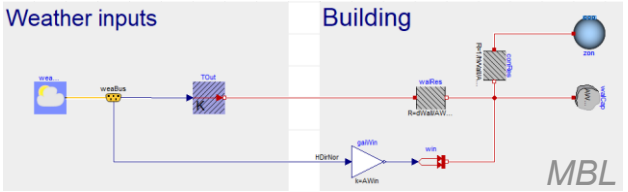


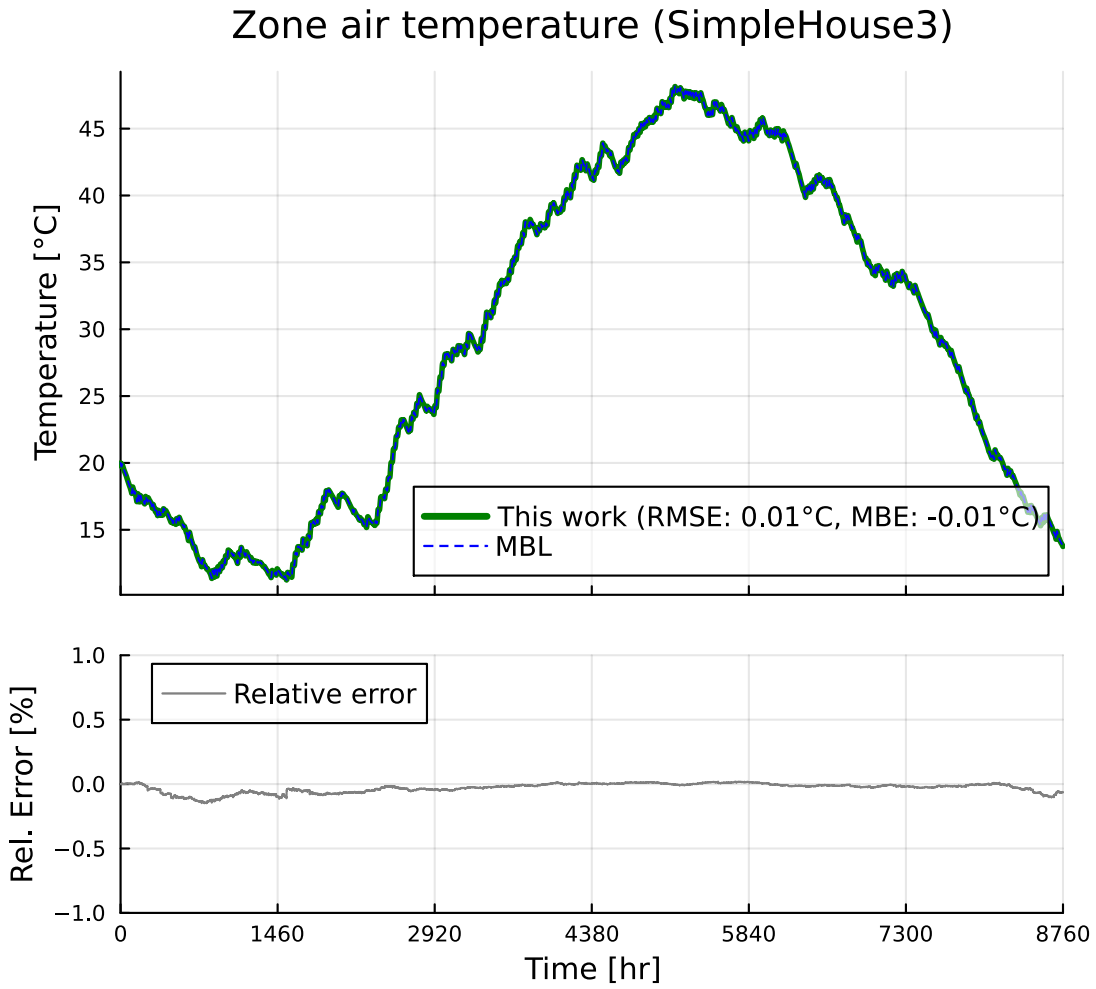
Annual RMSE < 10^{-5} °C

Annual RMSE < 10^{-5} W/m²

Result: Free-floating | SimpleHouse3

- Envelope, window solar gains, and zone-air thermal dynamics





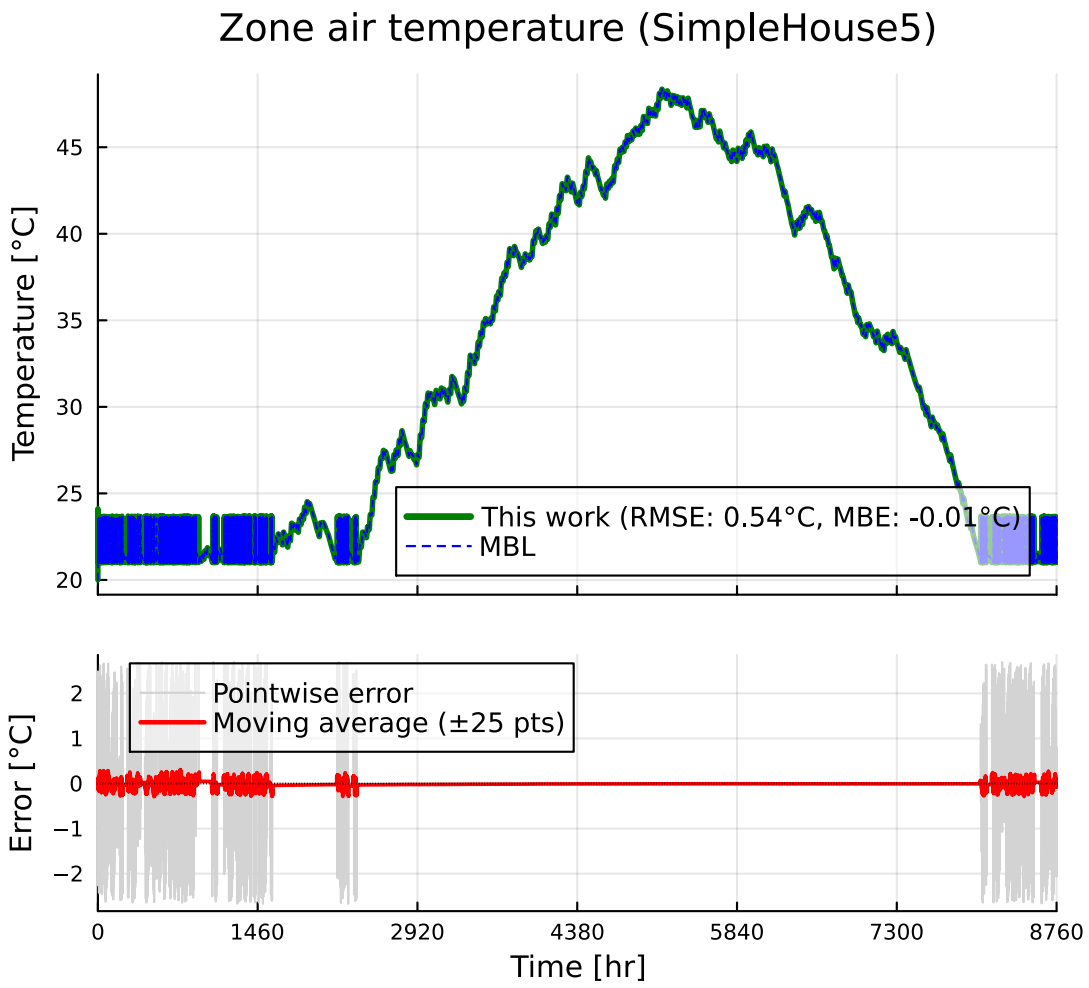
Annual Tzone error

RMSE 0.01 °C
MBE −0.01 °C

Close agreement with reference, supports the envelope and zone-air formulation.

Result: Heating + hysteresis control | SimpleHouse5

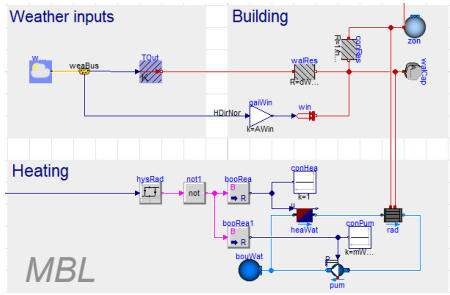
- Band: 21–23°C



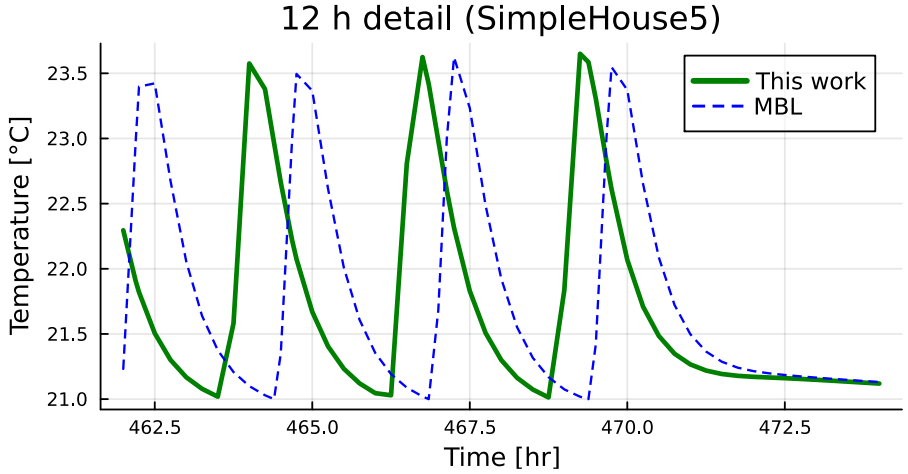
Annual Tzone error

RMSE 0.54 °C

MBE -0.01 °C



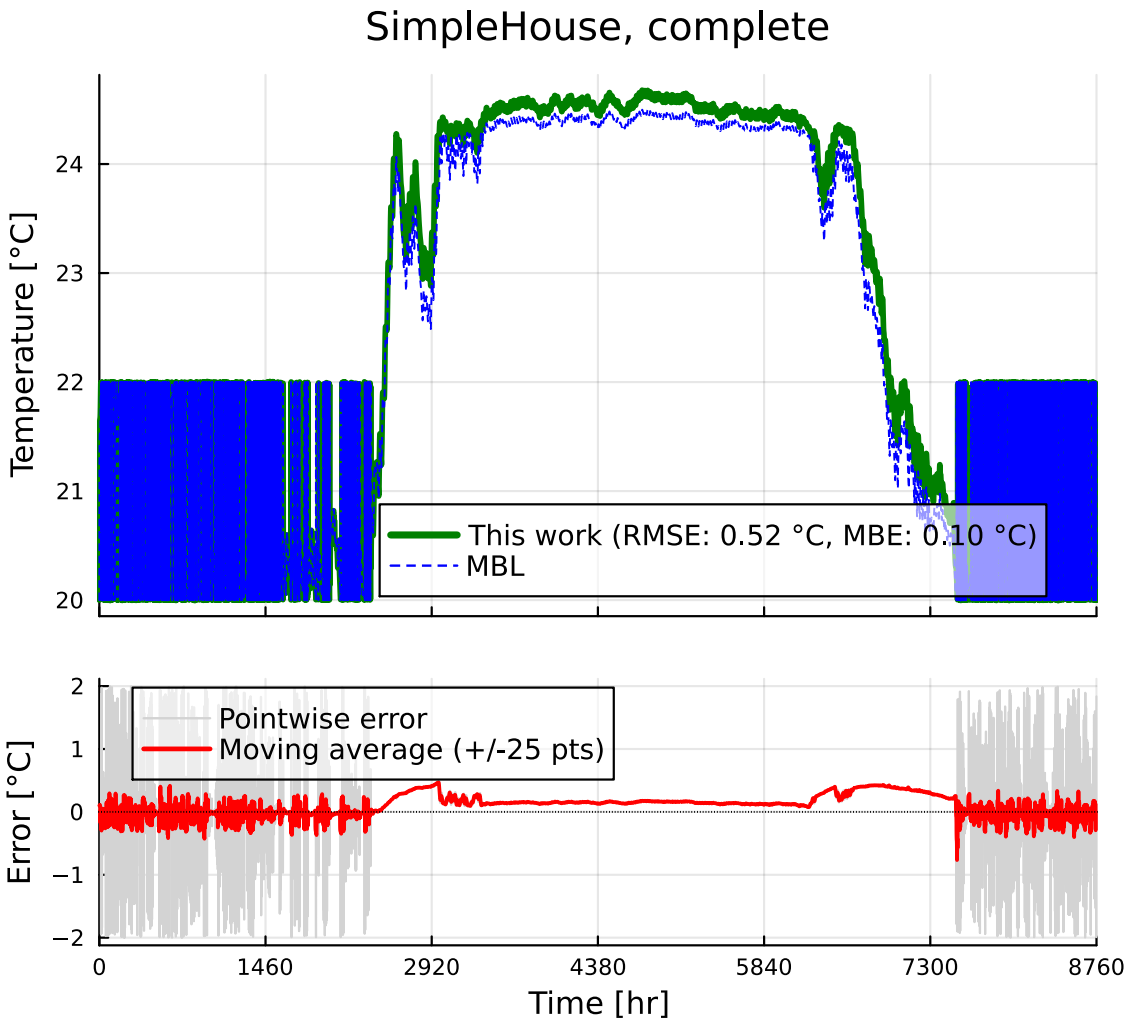
Differences in cycle timing

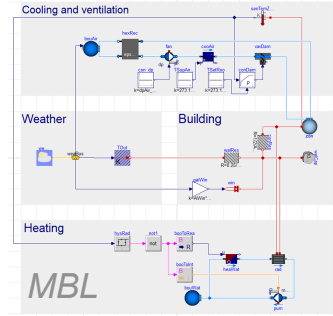


- This work uses a continuous switching-state approximation.
- MBL uses discrete events

Result: Full operation | Standalone SimpleHouse

- Heating band: 20−22 °C. Cooling setpoint: 24°C





Annual Tzone error

RMSE 0.52 °C

MBE +0.10 °C

Heating + hysteresis period

Differences in switching-cycle timing as in SimpleHouse5.

Cooling + PID period

Small positive bias (slightly warmer than reference)

Limitations & Conclusions

Limitations

- **Current scope**: Lightweight prototype for feasibility evaluation.

  - **Media and connectors**

    Fixed property functions. General, replaceable media are not supported.

  - **Flow direction**

    Predefined inlet / outlet roles. Reverse flow is not supported.

  - **Zero-flow behavior**

    Retain non-zero flow (10^{-4} of nominal) to avoid an undetermined outlet temperature.

  - **Hysteresis control**

    Use a continuous first-order approximation instead of event-based switching.

Future work

- **MTK updates**
  - Maintain model compatibility and reproducible examples as MTK evolves.
- **Graphical modeling**
  - Visualize and assemble component networks.
- **Extended verification**
  - Benchmark or more complex reference models.
- **Full SciML workflows**
  - Explore AD, sensitivity analysis, optimization, and ML.

Conclusions

1 Lightweight modular building and HVAC models in Julia

- Weather, thermal, and HVAC components are assembled through connectors.

2 Numerical verification against MBL

- Close agreement for weather and free-floating response.
- Hysteresis control results differ mainly in timing.

3 A foundation for further SciML workflows

- Compiled MTK models can serve future AD-enabled optimization-based control work.

github.com/BE-HVACR/BuildEML.jl

Acknowledgement: This research was partially supported by the National Science Foundation under Grant No. 2318720.

Thank You
For Your Attention!



