INL/EXP-22-68275-Revision-0

Digital Twin Development for Real Time Optimization

August 2022

Changing the World's Energy Future

Junyung Kim, Daniel Garrett

INL/EXP-22-68275-Revision-0

Digital Twin Development for Real Time Optimization

Junyung Kim, Daniel Garrett

August 2022

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Digital Twin Development for Real Time Optimization

Junyung Kim*, Daniel Garrett**

* Intern, Rensselaer Polytechnic Institute, ** Mentor, Digital Reactor Technology & Development (C160), INL

Background

The integrated energy system (IES) is a comprehensive system integrating several energy sources (e.g., electricity and heat). System integration, analysis, exploration and optimization should be completed in the digital space. Digital Twin (DT) is especially suitable for IES which requires integration of multiple subsystems and high reliability. One application of the DT concept is to conduct real-time optimization of the control and operations of a cyber-physical system (CPS) with data-driven simulations.

Objective

Idaho National Laboratory (INL) is developing DT for real time optimization (RTO). Two key elements for the success of RTO are

[1] Smooth interface between digital model and optimization

[2] Fast optimization time

The digital twin development of IES is taken as the research object, and the development concept, workflow and technical characteristics of DT are discussed with simple case study.

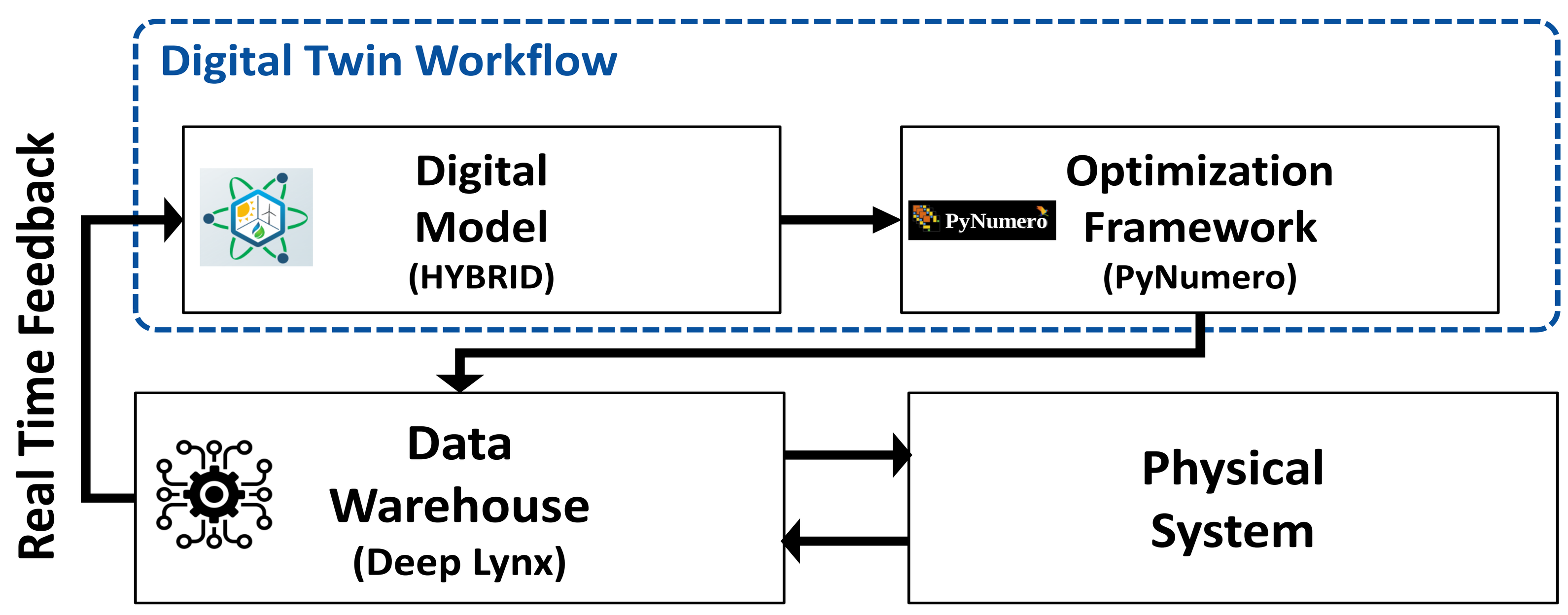


Fig. 1: Conceptual schematic of real time optimization workflow diagram.

Framework

To develop the digital twin of a physical system, system-level simulation model in Modelica language and RAVEN (Risk Analysis and Virtual Control Environment) have been mobilized to create a reduced order model (ROM) of the physical system. ROM works as DT of the physical system, and the Python-based optimization framework (PyNumero) receives the ROM as an external model and provides optimal inputs maximizing/minimizing objective functions. The external grey box model in PyNumero enhances the interface between digital model and optimization framework: one does not need to provide inputs, constraints, and outputs explicitly as algebraic expressions to the optimization framework, but simply plug in ROM to external grey box block in the framework.

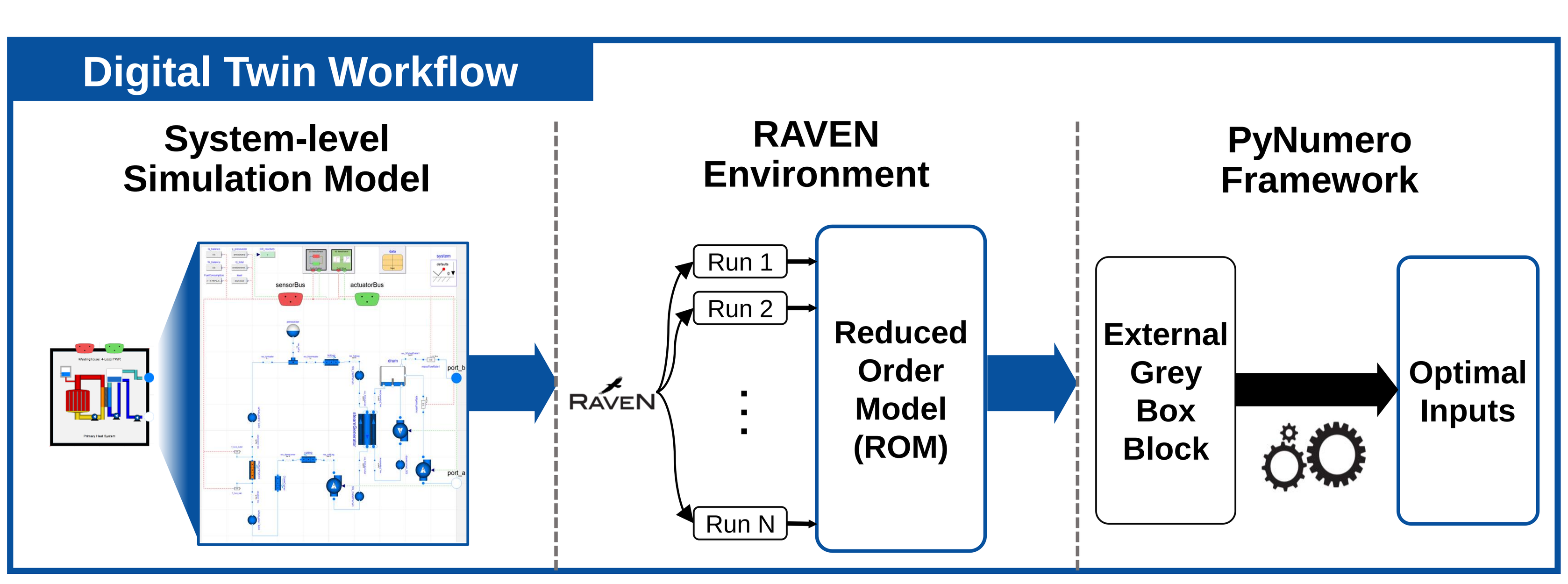
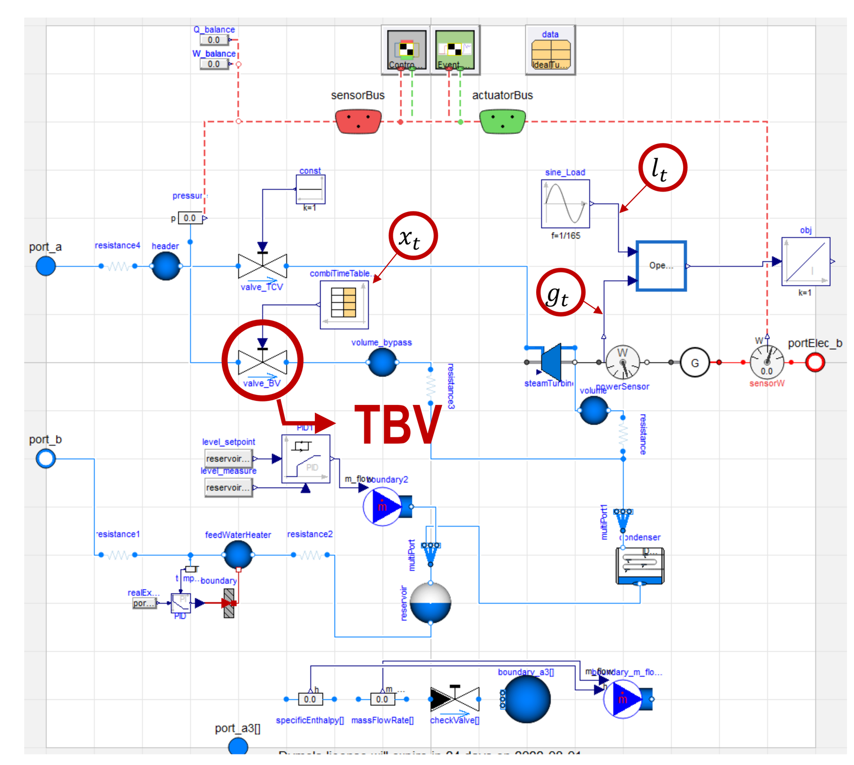


Fig. 2: Digital twin workflow diagram.

Case Study

Assume that plant operators want load following operation: matching electrical power generated as close as to the load demand by changing turbine bypass valve (TBV) opening area over time. To find the optimal inputs (TBV opening area at different time steps), DYMOLA simulation model and the objective function have been developed.

System-level Simulation Model Development



Objective Function:

$$f(x_t; t \in \{30, 40, \dots, 130\}) = \int_{t=30}^{130} \left(\frac{l_t - g_t(x_t)}{1e8} \right)^2 dt$$

l_t = load demanded at time t

x_t = TBV opening area at time t

$g_t(x_t)$ = electricity generated area at time t

Fig. 3: Rankine Cycle Balance of Plant (BOP) DYMOLA model.

Optimization output

```
solver = pyo.SolverFactory('cyipopt')
solver.config.options['hessian_approximation'] = 'limited-memory'
results = solver.solve(concreteModel)
✓ 48.3s
```

Optimal inputs found

```
2 Var Declarations
    inputs : Size=10, Index=egb._input_names_set
        Key                : Lower           : Value           : Upper           : Fixed : Stale : Domain
        valve_area_t100_110 :  0.100022703595 :  0.15219250539 : 0.199890470109 : False : False :  Reals
        valve_area_t110_120 :  0.200052037649 : 0.219024168197 :  0.29997138076 : False : False :  Reals
        valve_area_t120_130 :  0.250003071874 : 0.332950145461 : 0.349971767492 : False : False :  Reals
          valve_area_t30_40 :  0.100033251173 : 0.154657874271 : 0.199831875903 : False : False :  Reals
          valve_area_t40_50 :  0.100000581797 : 0.120968757709 : 0.149975259404 : False : False :  Reals
          valve_area_t50_60 : 0.0500093263225 : 0.0855123411679 : 0.0999924619682 : False : False :  Reals
          valve_area_t60_70 : 0.0500200772192 : 0.0597456433164 : 0.099973034102 : False : False :  Reals
          valve_area_t70_80 : 0.0500092598493 : 0.0525886075982 : 0.0999875780544 : False : False :  Reals
          valve_area_t80_90 : 0.0500134698581 : 0.0783750020986 : 0.149955769791 : False : False :  Reals
         valve_area_t90_100 :  0.100011876156 : 0.103754730728 : 0.149985381879 : False : False :  Reals
    outputs : Size=1, Index=egb._output_names_set
        Key       : Lower          : Value          : Upper         : Fixed : Stale : Domain
        BOP.obj.y : 0.730359256268 : 0.740140140055 : 7.51498651505 : False : False :  Reals
```

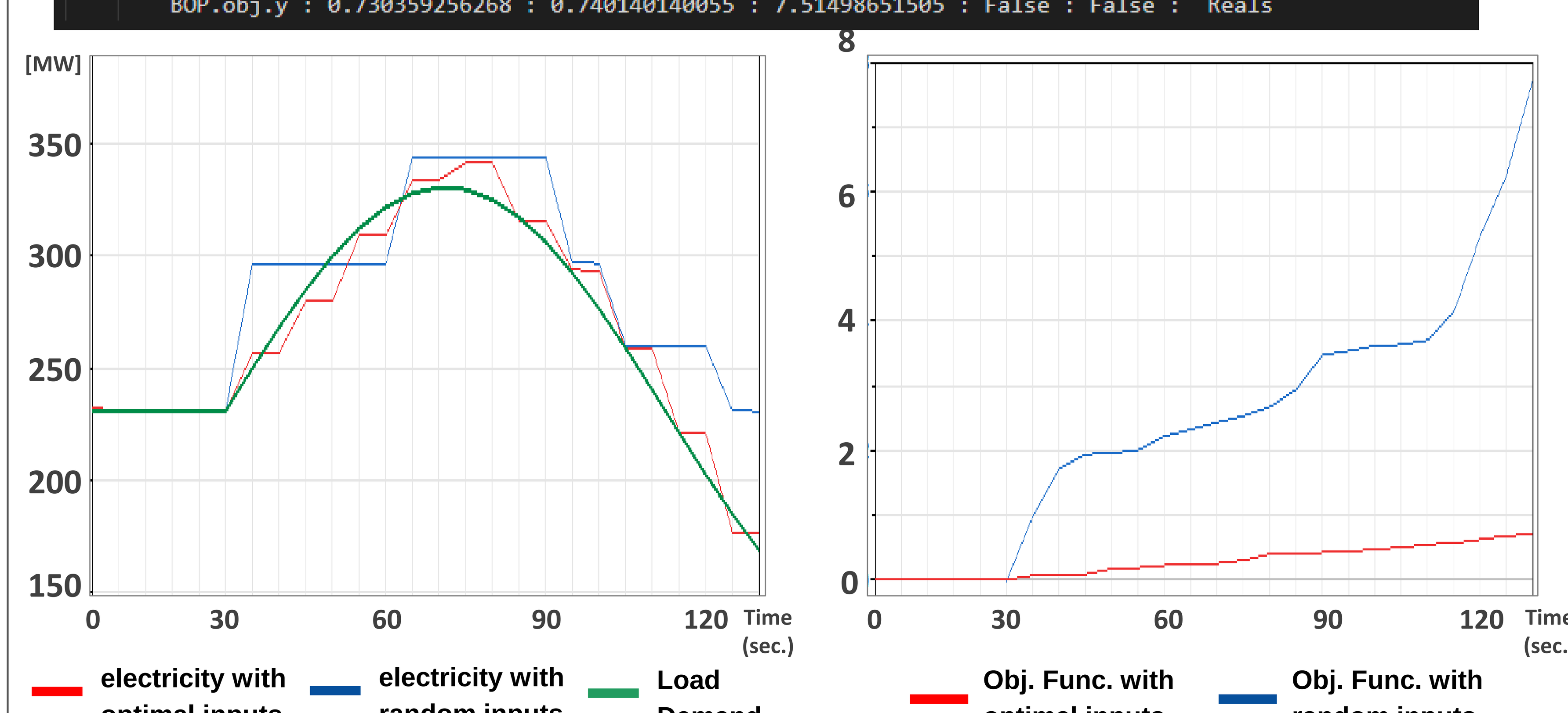


Fig. 4: Python code snippet (PyNumero) and the optimization results

Conclusions & Future Works

In this research, DT for RTO has been developed and demonstrated using system-level simulation model, RAVEN, and PyNumero. It is worth noting that optimization time is dependent on which ROM type is used, and how big the optimization model is. As a future work, we will demonstrate whole-scale real time optimization including the interface with data warehouse and physical system.

Acknowledgements

The authors would like to thank and acknowledge the Department of Energy (DOE) a national laboratory Integrated Energy Systems (IES) Program.

Idaho National Laboratory