OAK RIDGE National Laboratory

Development of Multi-Purpose Dynamic Nuclear Thermal Rocket System Models

Jordan D. Rader PhD

Tom P. Norby

NETS 2021

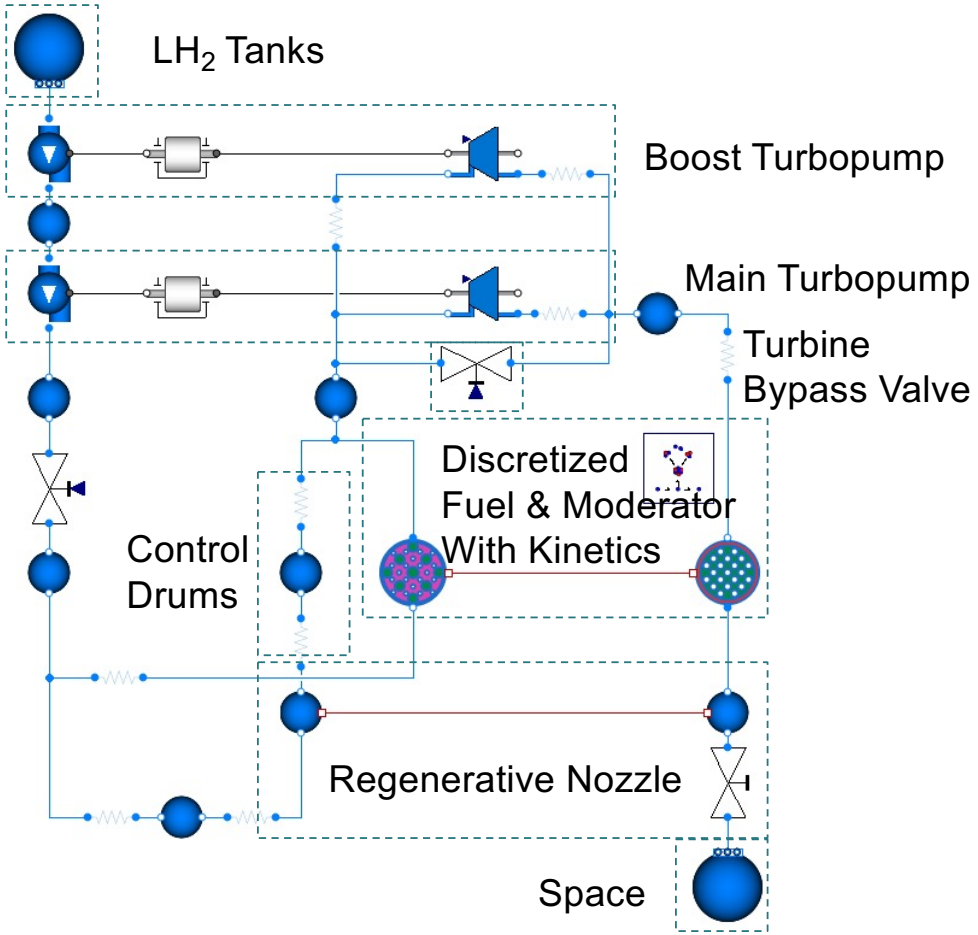
ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Project Background

- Since 2017, ORNL has been developing dynamic models of nuclear thermal propulsion (NTP) systems in support of the NASA space nuclear propulsion (SNP) program
- ORNL works with NASA, industry, and universities to develop the NTP instrumentation and controls (I&C) strategy
  - BWXT, Aerojet Rocketdyne, Univ. Tenn. Knoxville, and others
- To support integration with other aspects of the I&C program, the ORNL model is being parameterized and modified for export out of the model development environment
  - A side benefit being additional access to the model by others due to removal of export-control restricted features

Example NTP Model



OAK RIDGE National Laboratory

Motivation for Current Work

- **Given:**
  - Model written in Modelica
  - Utilizing the ORNL-developed, open-source TRANSFORM library of components
- **However:**
  - Development occurs in the Dymola environment
    - A commercial integrated development environment (IDE) ($$)
  - There are many potential users of a dynamic model with various backgrounds and proficiencies and…
  - Users may be unfamiliar, or uninterested, with the details of the model
    - Or only interested in certain outputs or the effects of certain inputs
    - Or unwilling/unable to deal with the commercial license or learn a new software/language
    - There are >10,000 values calculated each time step which can be overwhelming/confusing
- **Therefore:**
  1. To best utilize the model, it should be **parameterized**, to the most practical extent
  2. And made available to interested users via **export** outside of the IDE
  3. And **executable** with free (or already obtained) software

Step 1: Parameterization

- Parameterization is necessary for <u>useful</u> model export (step 2)
- **Parameter**s are determined by the initialization problem
  - They may be specified or calculated from other parameters
- What types of parameterization are supported?
  - Fluid boundary conditions: LH_2 tank pressure and temperature or nozzle back pressure (if not functions of time)
  - Reactor power shape and nominal reactor power
  - Nuclear heating factors
    - Appreciable nuclear heat is deposited in regions outside the active fuel
  - Initial conditions
  - Many other things inasmuch as they do not violate the compilation requirement (below)
- What is not supported?
  - Anything that affects the compilation of the model
    - Removal or addition of components
    - Nodalization of discretized components
    - Statements that affect the number of equations to be solved
      - Typically, some forms of Boolean-based coding
      - Behavior may not be as expected

Useful in the context of simulation outside of IDE for a variety of purposes

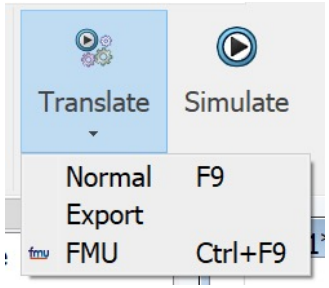
parameter - a Modelica keyword that denotes time invariance and determination at initialization

When fully parameterized, the model is largely generic

A generic model is quite powerful

Step 2: Model Export

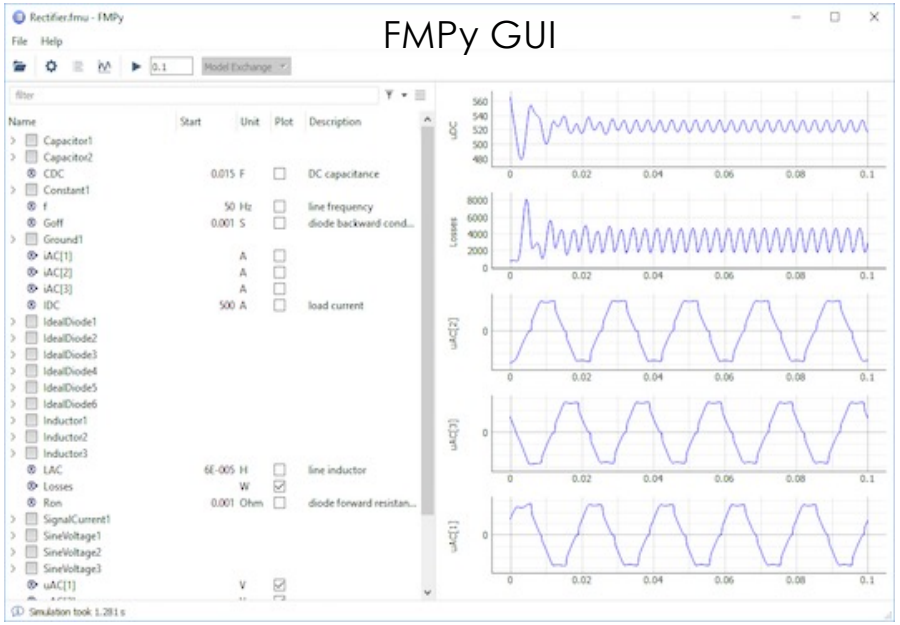
- Dymola supports model export to a Functional Mockup Unit (FMU)
  - The Functional Mockup Interface (FMI) defines a container (the FMU) and interface for exchanging dynamic model information
- Relevant simulation parameters:
  - FMI 2.0
  - Co-simulation (Dymola solver embedded in FMU)
  - Sdirkhw34 solver (supports stiff equation sets)
  - 64-bit



Step 3: Model Execution

- Model tested using the FMPy python module from CATIA
  - Available on GitHub or via `pip install`
  - Can be used in regular python environment
  - Or as self-contained GUI
- Many other options available
- Example (next slide) made using jupyter notebook
- Need to confirm that Dymola simulation is the same as FMU

FMPy GUI

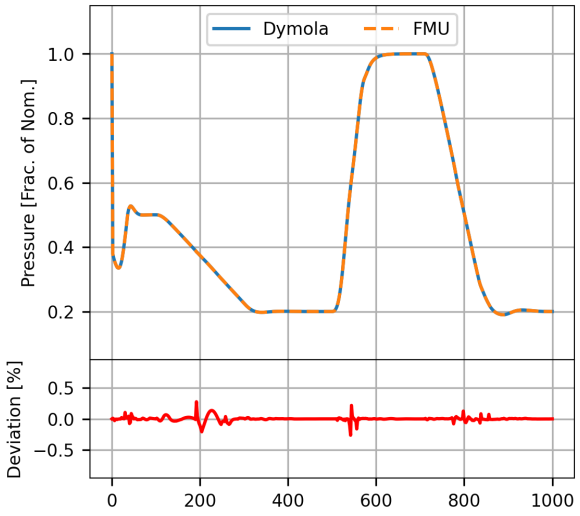


Comparison of FMU and Dymola Calculations

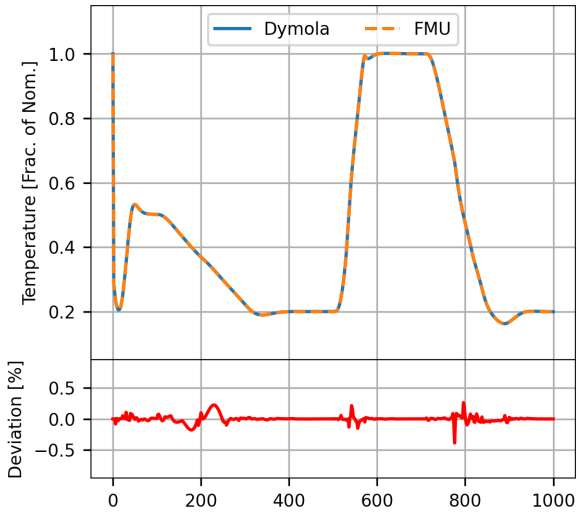
- Establish steady state at 20%
- Thrust-up in 60 seconds, hold 150 seconds, thrust-down in 120 seconds, hold at 20%

Time [s]	Pressure [%]	Temperature [%]
0	50	50
100	50	50
300	20	20
500	20	20
560	100	100
710	100	100
830	20	20
1,000	20	20

Chamber Pressure



Chamber Temperature



Less than 0.5% difference

OAK RIDGE National Laboratory

How will this be used for I&C development?

- Control system studies
  - I&C designers can use their language or program of choice with confidence that the exported model behaves like the baseline model
- Engine vehicle integration
  - The engine needs to be started and stopped multiple times (perhaps 8)
  - The same model can be used for analysis of each burn just by changing the parameters
- Interfacing with experiments via hardware-in-the-loop (HIL)
  - The exported model runs faster than real time

Summary & Next Steps

- Developed an analysis framework to support I&C activities
  - Assemble basic design information
  - Create generic, parameterized model
  - Export model for external users
  - Apply specific design information
  - Execute model(s)
- Continuing to update the generic model to support any changes to engine and vehicle architecture
- Working with SNP team members to explore use of FMU in I&C and related applications