INL/MIS-23-71698



FORCE – Transient Physical Modeling Workshop

HYBRID Overview April 4, 2023

Presented by: Dr. Daniel Mikkelson

where while the second state of the second state of

Session Agenda

1. HYBRID and FORCE (15 min)

- a) How HYBRID fits
- b) What is HYBRID
- c) What is in HYBRID
- 2. Features of Modelica (15 min)
 - a) Inherent advantages
 - b) Replaceable modeling
- 3. How Models are Constructed (30 min)
 - a) Model integration
 - b) Parameterizing and information passing
 - c) Initialization and RAVEN interface



How HYBRID Fits Within FORCE

- INL tools enable IES modeling analysis
 - Physical process, integration modeling
 - Long-term technoeconomic analysis
 - Capacity, dispatch optimization
 - Stochastic analysis, multiple commodities
 - Energy storage, varied markets
 - Real-time optimization





HYBRID – What Is It?

- The HYBRID repository is a collection of physical models written to characterize:
 - Thermal and electrical integration of different processes
 - Ramp speed
 - Evaluation of novel control schemes
 - Off-design system states
 - Dispatch feasibility
 - Safety limit approaches, considering control system effects

https://github.com/idaholab/HYBRID



[X] HYBRID and FORCE

New HYBRID Structure



Modelica Dynamic Models
Primary Heat Systems
Energy Manifold
Balance of Plant
Industrial Process
Energy Storage
Secondary Energy Source
Primary Heat Switch Yard
Electrical Grid
Control System Center
Experimental Systems

Steady-State	Transient	ROMs	Cost Information
Tech 1	Tech 1	Tech 1	Tech 1
Tech 2	Tech 2	Tech 2	Tech 2
Tech 3	Tech 3	Tech 3	Tech 3
Tech 4	Tech 4	Tech 4	Tech 4
Aspen, Mathcad, Excel, etc.	non-Modelica or "Save-total" Modelica	Trained on other models	HERON-readable format



[X] HYBRID and FORCE

V&V Matrix (1/4)

Subsystem Name	V&V	Example Type	ROM generated	Steady- state model	Published documents	Reference documents	Nominal Conditions, Notable Limitations
4-loop PWR	Face	Integrated			INL/EXT-19-55395	Systems Summary of a Westinghouse PWR Nuclear Power Plant 1984 "PWR Description", Jacopo Buongiorno	3400 <u>MWt</u> , Steam: 1750 kg/s, 69 bar, 285°C
Small modular IPWR	Data: steady state	Integrated			INL/CON-16-39032	https://aris.iaea.org/PDF/NuScale.pdf doj: 10.1016/j.desal.2014.02.023	160 <u>MWt</u> , Steam: 35 bar, 300°C, 75 kg/s
Small modular natural circulation IPWR	Data: Steady state	Integrated			INL/EXT-19-55520 doj: 10.1080/00295450.2020.1781497 doj: 10.1016/j.apenergy.2022.118800 INL/RPT-22-69214	NuScale Standard Plant Design Certification Application	200 MWt. Steam: 35 bar, 310°C, 84 kg/s
HTGR	Data: Transient	Integrated			doi: 10.2172/1890160 INL/RPT-22-68222 INL/RPT-22-66941 INL/RPT-22-69214	doi: 10.1016/j.nucengdes.2017.11.041	130 <u>MWt</u> , Steam: 140 bar, 540°C, 50 kg/s
SFR	Physics	Individual			INL/RPT-22-68222		BOP under construction



[X] HYBRID and FORCE

V&V Matrix (2/4)

Subsystem Name	V&V	Example Type	ROM generated	Steady- state model	Published documents	Reference documents	Nominal Conditions, Notable Limitations
Solid Media TES	Face	Integrated			doj; 10.1016/j.est.2022.104387 doj; 10.1016/j.apenergy.2022.118800 INL/EXT-21-61985	doj; 10.1063/1.4984432	Nominally concrete, requires steam
2-tank TES	Face	Integrated			INL/EXT-18-45505 INL/RPT-22-66941 INL/RPT-22-69214		Molten salt
Thermocline TES	Physics, some data	Integrated			INL/EXT-20-59195 INL/EXT-21-64408 INL/EXT-21-61985		Thermal oil
Latent heat TES	Physics, some data	Individual			INL/EXT-21-61985		
Battery storage	Physics	Integrated	Х		INL/MIS-20-60624		
Compressed air	Physics	Individual			INL/RPT-22-66941		Single-mode operation



[X] HYBRID and FORCE

V&V Matrix (3/4)

Subsystem Name	V&V	Example Type	ROM generated	Steady- state model	Published documents	Reference documents	Nominal Conditions, Notable Limitations
Reverse osmosis desalination	Data: <u>steady-</u> <u>state</u>	Integrated			INL/EXT-18-45505 INL/EXT-15-36451		
High temperature steam electrolysis	Data: <u>steady-</u> <u>state</u>	Integrated			INL/EXT-16-40305 INL/EXT-19-55395	doj: 10.1016/j.jpowsour.2006.12.081 doj: 10.1016/j.ijhydene.2012.12.086 doj: 10.2172/1513461	
HTSE 'experimental'	Face	Individual			INL/EXT-22-02188		
Single-stage balance of plant	Face	Integrated			Many		
Two-stage balance of plant	Face	Integrated			INL/RPT-22-69214		
Stage-by- stage balance of plant	Physics	Integrated			doj: 10.1016/j.apenergy.2022.118800		



V&V Matrix (4/4)

Subsystem Name	V&V	Example Type	ROM generated	Steady- state model	Published documents	Reference documents	Nominal Conditions, Notable Limitations
TEDS loop	Physics, some data	Individual, Integrated			INL/EXT-20-59195 INL/EXT-21-64408		
MAGNET loop	Physics	Individual, Integrated			INL/EXT-22-02188		
Subsystem Name	V&V	Example Type	ROM generated	Steady- state model	Published documents	Reference documents	Nominal Conditions, Notable Limitations
Steam manifold	Physics	Integrated			Many		
Switchyard	Physics	Integrated			Many		
Electric grid	Physics	Integrated			Many		
Natural gas turbine	Face	Integrated	Х		INL/EXT-16-40305		



HYBRID Dynamic Modeling

- HYBRID evaluates the feasibility of systems developed within FORCE and provides constraint data necessary for broader system evaluations
 - An ideal intermediary for determining:
 - Integration design
 - Control methods
 - Ramp rate feasibility
 - Determination of off-design **behaviors**



[X] HYBRID and FORCE

Where FORCE Interacts?

- Inputs are system sizing
 - Values taken from RAVEN/HERON in optimization workflow
- Control strategies desired
 - Each subsystem has its own control strategy
- Planned coupling methodologies
 - Supervisory control
 - Minimum electrical and heat rates for each subsystem
- Thermal and electrical demands for each subsystem through time.
 - Total demand an input from balancing authority routine





Example Fluid Coupling Points

11

Why Modelica?

- Rapid Development
 - Fidelity level controlled by user
 - Fast feedback from development environment
- Collaborative
 - Model repositories can be open-source like HYBRID with standardized connections
 - FMI/FMU allows for "black-box" sharing
- Flexible, Adaptable
 - Modeling across multiple physical domains
 - Models modifiable for existing and new users



Modelica Features

Equation Based (acausal)

Order of computations is not decided at modeling time

- Equations do not specify input/outputs
 - $x + y = z^x + yz$
- Solutions direction adapts to data flow



Built for Dynamic Problems

- Time integration handled by solver
 - der(v) = a + bx(t)



Example from the Modelica Standard Library



HYBRID and FORCE

[X] Modeling Features

13

Dynamic Simulation

- Time dependent aspects of a system
- Concerned with concepts of:
 - States: Attributes described at a point in time
 - Events: Occurrences that trigger a state transition
 - Transitions: A change in the state of an object
 - Actions: Instantaneous operation that results due to an event
 - Activities: Ongoing operations upon the state of an object Example of a dynamic problem



HYBRID and FORCE

[X] Modeling Features

14

Replaceable Modeling





HYBRID and FORCE

[X] Modeling Features

15

Design Capability

- Physical models are focused on process system bases
 - A few coupled subsystems (nuclear plant + gas turbine + thermal storage + grid + ancillary process)
 - Focus within Hybrid has been single energy park systems





Model Construction

Design Capability

- Figures of merit
 - Demand missed
 - System stability
 - System pressure, temperature, thermal gradients, valve positioning, etc.
 - Control strategy effects
 on each subsystem
 - Carbon accounting





Model Construction

Interconnectability

- Create self-contained process models
- These models calculate on- and offdesign behavior
- Coupling occurs with other Modelica models or process models built via FMI/FMU

HYBRID and FORCE



Interconnectability

- Models are configured using interchangeable base classes for ease of use and adaptability of models into different configurations
- Can be exported in the FMI/FMU standard to allow robust interoperability with industry



HYBRID and FORCE

Energy Arbitrage IES

- NuScale-style SMR
- High-fidelity balance of plant
- Integrated-concrete thermal-energy storage system (dual network model)
- Week-long-scaled dispatchable demand profile calculated and input



Energy Arbitrage IES



HYBRID and FORCE

[X] Modeling Features

PHS– Westinghouse (WH) Style: 4-Loop (PWR)





Model Construction

HYBRID and FORCE

Energy Manifold







| HYBRID and FORCE

[X] Modeling Features

ES – Sensible Thermal Energy Storage (TES)







| HYBRID and FORCE

[X] Modeling Features

24

High-Temperature Steam Electrolysis (HTSE)





HYBRID and FORCE

X Modeling Features

Model Construction

L I

Reverse Osmosis (RO) Desalination



| HYBRID and FORCE

[X] Modeling Features

Model Construction

Integrated Energy Systems

Natural Gas Fired Turbine







HYBRID and FORCE

[X] Modeling Features

27

DETAIL Model





| HYBRID and FORCE

28

Integration of Existing Models

- Drag and drop of models is the most common method of building top-level systems
 - Example: IES, Reactor model
 - Prebuilt models combined in unique ways for simulation setup
 - Primary simulation difficulties are system-wide initialization and proper calibration of controls
- Subcomponents can be combined to make usable components
 - Example: Shell and tube heat exchanger
 - Configured models allow for standardized components for full system builds
 - Primary difficulty is to ensure appropriate parameter pass-through



Integration of Existing Models

- Using existing models takes advantage of object building within Modelica
- The same components can be used repeatedly
- Subsystems have been tested and verified
- Ports impose consistent communication between components



Example: IES



Example: IES – Reactor Model

- Sixteen different drag-and-drop components make up reactor model
 - Includes pipes, sensors, feedwater pump, primary heat exchanger, nuclear core model, and control signals
 - Some of these models have drag-and-drop subcomponents
- Subsystem level is self-contained, only needing feed flow and steam produced connections.



Example: IES – Turbogenerator

- Turbogenerator system demonstrates five connection types
 - Fluid (blue)
 - Heat (red solid)
 - Mechanical (gray)
 - Electrical (red solid)
 - Control (red & green dashed)



Integration of Existing Models

- Construction using pre-existing models creates instantiations of the objects within current level model
- Typically, ports and connectors are used to communicate information between objects
- Assuming the building block models exist, the construction process can happen quite intuitively
 - Example: Shell & tube heat exchanger
- Parameter passing must be handled at every level



Example: Shell & Tube Heat Exchanger

- What do we need to make a STHX?
 - Shell fluid flow path

Tube fluid flow path



 Pipe model to establish conductivity



Possibly a vectorization reversing unit to allow for counter-flow OR concurrent flow

 External fluid connectors





X Model Construction

HYBRID and FORCE

Example: Shell & Tube Heat Exchanger

- Finished product thermally connects two fluid streams
- One final question: how do we properly pass parameters to next-level modeling?
 - Each component in the figure on the right has its own parameters
 - For example: what is the diameter of the tube in the tube model?





Integration of Existing Models: Passing Parameters

- Typically, parameters must be re-declared at every level
 - Default values can be put in, as the highest modeling level will be distributed down
- "Replaceable" keyword allows for all potential values matching the type of that parameter to be selected via drop-down menu
 - For example: two-phase media types
- Parameters can be grouped into data structures for easier pass-through

STHX in NHES.Systems	.PrimaryHeatSystem.SMR_Generic.Components.SMR_Taveprogram	?
General Shell Initializ	tion Tube Initialization Wall Initialization Advanced Visualization Add modifiers Attributes	
Component		Icon
News		
Name STRA		
Comment		9
1odel		
Path TRANSFORM	HeatExchangers.GenericDistributed_HX	GenericDistrib.
Comment A (i.e., no inl	et/outlet plenum considerations, etc.) generic heat exchanger with discritized fluid and wall volumes where concurrent/counter flow is specified mass flow direction.	
Paramotore		
a u u		
nParallel	TRANCCORM Elvid Channel Blatikute Wedels Distributed/clumes 1D LiesEndersees Challes Trube IV /D is also and The Committee IV	
Medium, chell	TRANSFORM. Huld. Closure Relations. Geometry _Models. Distributed Volume_LD. Healt. Xolarget _ site land to be the Xolarget	
Medium_snen	Modelica Media Water StandardWater × III > Tube cide medium	
Material tubeWall	TRANSFORM Media Solids SS304 V	
counterCurrent	the construction of the co	nd flux vector order
ressure Loss		
FlowModel_shell	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec 🗸 🔢 🛌 Shell side flow models (i.e., momentum, pressure	loss, wall friction)
FlowModel_tube	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec 🗸 🏢 🗾 Tube side flow models (i.e., momentum, pressure	e loss, wall friction)
leat Transfer		
HeatTransfer_shell	TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Nus_SinglePhase_2Region 🗸 🏢 🔸 Shell side coeffici	ent of heat transfer
HeatTransfer_tube	TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Alphas_TwoPhase_3Region 🗸 💷 + Tube side coeffici	ent of heat transfe
InternalHeatGen_tube	GenericHeatGeneration 🗸 🧾 🕨	
InternalHeatGen_shell	GenericHeatGeneration 🗡 鼲 🔸	
race Mass Transfer ——		
T		11 12 12 12

Parameter interface seen above. Interface method depends on type of parameter (single value, package selection, set of values, etc)

X Model Construction



HYBRID and FORCE

Passing Parameters

eneral Shell Initializ	zation Tube Initialization Wall Initialization Advanced Visualization Add modifiers Attributes	
omponent		Icon
Name STHX		
Comment		AA
del		j j
Path TRANSFORM	I HeatFychanners GenericDistributed HX	GenericDistrib
Comment A (i.e., no in	Interestioning is some representation of the second secon second second sec	
arameters		
Parallel	for the state of identical parallel HXs	
Geometry	TRANSFORM.Fluid.ClosureRelations.Geometry.Models.DistributedVolume_1D.HeatExchanger.ShellAndTubeHX (D_i_st v III Seometry	
Medium_shell	Modelica.Media.Water.StandardWate 🗸 🛙 Edit 🖥 Shell side medium	
Medium_tube	Modelica.Media.Water.StandardWater 💙 📧 🗡 Tube side medium	
4aterial_tubeWall	TRANSFORM.Media.Solids.SS304 V III + Tube wall material	
counterCurrent	true VI Swap shell side temperature ar	ad flux voctor order
counter current		iu nux vector order
ressure Loss		
ressure Loss	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec >>) III • Shell side flow models (i.e., momentum, pressure	e loss, wall friction)
ressure Loss FlowModel_shell FlowModel_tube	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developer III Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developer III Tube side flow models (i.e., momentum, pressure	e loss, wall friction) e loss, wall friction)
essure Loss FlowModel_shell FlowModel_tube eat Transfer	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec >> III • Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec >> III • Tube side flow models (i.e., momentum, pressure	e loss, wall friction) e loss, wall friction)
essure can bit essure Loss lowModel_shell lowModel_tube eat Transfer HeatTransfer	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec > III Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec > III Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region >> TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region >> III Shell side coefficient	e loss, wall friction) e loss, wall friction) ent of heat transfer
essure can the essure Loss NowModel_shell NowModel_tube eat Transfer HeatTransfer_shell HeatTransfer_tube	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III · Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region >> III · Shell side coefficit TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region >> III · Shell side coefficit TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region >> III · Tube side coefficit	e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer
essure Loss FlowModel_shell FlowModel_tube eat Transfer HeatTransfer_shell HeatTransfer_tube InternalHeatGen_tube	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III · Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III · Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_ZRegion >> III · Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region >> III · Tube side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region >> III · Tube side coeffici	e loss, wall friction) e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer
essure Loss FlowModel_shell FlowModel_tube eat Transfer HeatTransfer_shell HeatTransfer_tube internalHeatGen_tube internalHeatGen_shell	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developer III Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developer III Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region III Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region III Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region III Tube side coeffici GeneridHeatGeneration III GeneridHeatGeneration III	e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer
essure Loss FlowModel_shell FlowModel_tube eat Transfer HeatTransfer_shell HeatTransfer_tube internalHeatGen_tube internalHeatGen_shell ace Mass Transfer_	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III Shell side flow models (i.e., momentum, pressure transfer.Models.DistributedPipe_1D_SinglePhase_Developec > III Tube side flow models (i.e., momentum, pressure transfer.Models.DistributedPipe_1D_MultiTransferSurface.Nus_SinglePhase_2Region >> III Shell side coefficit TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Nus_SinglePhase_2Region >> III Shell side coefficit TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Alphas_TwoPhase_3Region >> III Tube side coefficit GenericHeatGeneration >> III GenericHeatGeneration >> III	e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer
essure can the essure Loss FlowModel_shell FlowModel_tube eat Transfer	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec > III Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec > III Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region > III Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region > III Shell side coeffici CenericHeatGeneration > III GenericHeatGeneration > III CenericHeatGeneration > GenericHeatGeneration > III GenericHeatGeneration > III CenericHeatGeneration >	e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer
essure Loss FlowModel_shell FlowModel_tube eat Transfer HeatTransfer HeatTransferbe InternalHeatGen_shell race Mass Transfer InternalTraceGen_tube InternalTraceGen_shell	TRANSFORM. Fluid. ClosureRelations. PressureLoss. Models. DistributedPipe_1D. SinglePhase_Developer > III • Shell side flow models (i.e., momentum, pressure (TRANSFORM. Fluid. ClosureRelations. PressureLoss. Models. DistributedPipe_1D. SinglePhase_Developer > III • Tube side flow models (i.e., momentum, pressure (TRANSFORM. Fluid. ClosureRelations. HeatTransfer. Models. DistributedPipe_1D. MultiTransferSurface. Nus_SinglePhase_2Region > TRANSFORM. Fluid. ClosureRelations. HeatTransfer. Models. DistributedPipe_1D_MultiTransferSurface. Nus_SinglePhase_2Region > III • Shell side coeffici TRANSFORM. Fluid. ClosureRelations. HeatTransfer. Models. DistributedPipe_1D_MultiTransferSurface. Alphas_TwoPhase_3Region > III • Shell side coeffici GenericHeatGeneration > III • GenericHeatGeneration > III • Shell side coeffici GenericHeatGeneration > III • GenericHeatGeneration > III • Shell side coeffici	e loss, wall friction) e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer ent of heat transfer
ressure carbin ressure Loss FlowModel_shell FlowModel_tube eat Transfer_ HeatTransfer_shell HeatTransfer_tube InternalHeatGen_tube InternalHeatGen_shell race Mass Transfer	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec >> III • Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec >> III • Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultiTransferSurface.Nus_SinglePhase_2Region >> III • Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Alphas_TwoPhase_3Region >> III • Tube side coeffici CenericHeatGeneration >> III • GenericHeatGeneration >> III •	e loss, wall friction) e loss, wall friction, ent of heat transfer ent of heat transfer
essure Loss essure Loss FlowModel_shell FlowModel_tube at Transfer teatTransfer_tube internalHeatGen_tube internalHeatGen_shell ace Mass Transfer nternalTraceGen_shell	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III • Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III • Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultTransferSurface.Nus_SinglePhase_2Region > III • Shell side coefficit TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region > III • Tube side coefficit TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultTransferSurface.Alphas_TwoPhase_3Region > III • Tube side coefficit GenericHeatGeneration > III • GenericHeatGeneration > III • GenericHeatGeneration > GenericHeatGeneration > III • GenericTraceGeneration > III • GenericTraceGeneration >	e loss, wall friction) e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer ent of heat transfer etton \sim [I] , etton \sim [I] ,
essure Loss FlowModel_shell FlowModel_tube eat Transfer HeatTransfer_tube InternalHeatGen_tube InternalHeatGen_shell ace Mass Transfer internalTraceGen_tube internalTraceGen_shell	TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D.SinglePhase_Developec > III • Shell side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_SinglePhase_Developec > III • Tube side flow models (i.e., momentum, pressure TRANSFORM.Fluid.ClosureRelations.PressureLoss.Models.DistributedPipe_1D_MultiTransferSurface.Nus_SinglePhase_2Region >> III • Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Alphas_TwoPhase_3Region >> III • Shell side coeffici TRANSFORM.Fluid.ClosureRelations.HeatTransfer.Models.DistributedPipe_1D_MultiTransferSurface.Alphas_TwoPhase_3Region >> III • Tube side coeffici GenericHeatGeneration >> III • Shell side coeffici GenericHeatGeneration >> III • GenericHeatGeneration >> III • GenericTraceGeneration >> III • GenericTraceGeneration >> III • CenericTraceGeneration >> III • GenericTraceGeneration >> III • GenericTraceGeneration >> III • CenericTraceGeneration >> III • GenericTraceGeneration >> III • GenericTraceGeneration >> III • CenericTraceGeneration >> III • GenericTraceGeneration >> III • GenericTraceGeneration >> II	e loss, wall friction) e loss, wall friction) ent of heat transfer ent of heat transfer ent of heat transfer ent of heat transfer

redeclare STHX.Ge	eometry in NHES.Systems.PrimaryHeatSystem.SMR_Generic.Components.SMR_Taveprogram ? X
General Shell Sid	de Tube Side Add modifiers Attributes
nTubes nR nSurfaces_tube	data.nTubes_steamGenerator # of tubes per heat exchanger 2 Number of radial nodes in wall (r-direction) 1 Number of transfer (heat/mass) surfaces
height_a_tube height_b_tube angle_tube dheight_tube	0 m Elevation at port_a: Reference value only. No impact on calculations. height_a_tube + sum(dheights_tube) m Elevation at port_b: Reference value only. No impact on calculations. 0 • Vertical angle from the horizontal (-pi/2 < x <= pi/2) length_tube*sin(angle_tube) m Height(port_b) - Height(port_a) distributed by flow segment
Inputs: Tube Wall —	
drs	fill(th_wall/nR, nR, nV)
th_wall	data.th_steamGenerator_tube
Inputs	
dimension_tube	data.d_steamGenerator_tube_inner • m Characteristic dimension (e.g., hydraulic diameter)
crossArea_tube	0.25*pi*dimension_tube*dimension_tube • m ² Cross-sectional flow areas
perimeter_tube	4*crossArea_tube/dimension_tube • m Wetted perimeters
length_tube	data.length_steamGenerator_tube • m Pipe length
roughness_tube	2.5e-5 m Average heights of surface asperities
surfaceArea_tube	{if i == 1 then perimeter_tube*length_tube else 0 for i in] 💷 • m ² Inner surface area
	OK Cancel Info



] HYBRID and FORCE

Modeling Features

38

Integration of Existing Models

- Using existing models takes advantage of object building within Modelica
- The same components can be used repeatedly
- Ports impose consistent communication between components
- When building sub-models and subsystems, make sure that relevant parameter passing methods are set up



Parameter Sweeping

- Dymola has internal parameter sweeping method
- Allows for output space generation across single altered parameter at a time
- Auto generates separate output files to keep post simulation
- Auto generates plotting set of output values desired by user



Modeling Features

tegrated Energy Systems





] HYBRID and FORCE

Modeling Features





HYBRID and FORCE

Modeling Features







HYBRID and FORCE

Modeling Features

Manual Parameter Sweep

- In the case that a model is sensitive to initial conditions, it is possible to manually alter parameters via the dsfinal.txt file to effectively manually parameter sweep
- Combined with script generation, this process can be automated so that there is less user attention required



- Model initialization is key to obtaining results, especially if a simulation stabilization time frame can be avoided
- One method of creating an initial state is to use a robust outside result to create an initial conditions table
 - ASPEN HYSYS is often used

General A	dd modifiers Attri	butes			
Component —			Ico	n	
Name d	lataInitial				
Comment			6	DataInitial_	NS
					٦
4odel N	HES.Systems.PrimaryH	leatSystem.SMR_Generic.Components.Data.DataInitial_NS		Generic	J
Comment					
Daramotore					
		[/	-		
d_start_core	_coolantSubchannel	{0.72999456787,0.70768652344,0.68389465332,0.65836		• g/cm³	
p_start_core	_coolantSubchannel	{12903247.0,12898190.0,12893307.0,12888614.0}		• Pa	
h_start_core	_coolantSubchannel	{540.2313,558.79304,576.1813,580.9483}	1.0	· K	
d start bott	_coordinationariter	(0.6592090057.0.65920466046)	100	J/Kg	
n start hot	eg eg	{1.08 E1756 1.06 10010}	100	g/un™ ► har	
T start bott	~9 ea	{324 70516357 324 676477051		• • • • • • • • • • • • • • • • • • • •	
h start hot	~9 en	{1488030 125 1488878 6251		► 1/kg	
d start cold	~9 e0	{0.75105032617\		 a/mg 	
n start cold	lea	{129.36737}		yon ≻har	
T start cold	Lea	(285 19472656)		• °C	
h start cold	Lea	{1260434,125}		• 1/ka	
d start STH	X tube	{0.80191125488.0.11777983093.0.06753452301.0.04695		▶ a/cm³	
p_start_STH	 X_tube	{39.281535,39.2524275,39.1300875,38.91694,38.610355		 bar 	
T_start_STH	 X_tube	{248.02053223,249.22127686,249.04030762,248.721887		• °C	
h_start_STH	X_tube	,2374421.2,2749133.0,2919464.0,2980839.5,3004198.5}		▶ J/kg	
d_start_STH	X_shell	{0.66202679443,0.66976708984,0.68217218018,0.69354		▶ g/cm³	
p_start_STH	X_shell	{128.10909,128.16239,128.19835,128.23497,128.27221,		• bar	
T_start_STH	X_shell	01654,547.5736,541.4915,535.5601,527.6039,516.3904}		۰к	
h_start_STH	X_shell	{1480592.25,1463564.625,1435439.875,1408754.25,138	П	J/kg	
d_start_inlet	Plenum	0.751	037	▶ g/cm³	
p_start_inlet	Plenum	129.	207	• bar	
T_start_inlet	Plenum	285.	193	• °C	
h_start_inlet	Plenum	1.26043e	+06	• J/kg	
d_start_outle	etPlenum	0.658	335	∙ g/cm³	
p_start_outle	etPlenum	128.	884	• bar	
T_start_outle	etPlenum	324.	724	• °C	
h_start_outle	etPlenum	1.48899e	+06	 J/kg 	
Ts_start_cor	e_fuelModel_region_1	326, 753.8575; 620.802, 676.6141, 692.713, 665.06805]	Ш	•	
Ts_start_cor	e_fuelModel_region_2	629.82477; 547.5703, 569.67633, 586.9591, 594.09796]		•	
Ts_start_cor	e_fuelModel_region_3	5, 590.4041; 540.2313, 558.79364, 576.1813, 586.9483]		•	
T_start_STH	X_tubeWall	, 554.01654, 560.9657, 576.07135, 583.2033, 585.6959]		• К	
p_start_pres	surizer	12807	852	• Pa	
level_start_p	ressurizer	1.18	567	• m	
h_start_pres	surizer	1.47822e	+06	• J/kg	
d_start_pres	surizer_tee	0.658	202	∙ g/cm³	
p_start_pres	surizer_tee	12807	852	• Pa	
T start pres	surizer tee	586.90	674	• к	



HYBRID and FORCE

- Inherent method within Dymola to save within a model the initial conditions
- When used, the output space is saved within the model directly as adjustments to the attributes

Save Start Values in Model		?	×
Source for start values	 		
 Current Variable Browser content Initialize the model and save the results 			
Store options			
 Store values in current model Store values in new model 			
Name:			
SMR_IES_CTES			
Description:			
Extends:			
NHES.Systems.Examples.SMR_IES_CTES			
Insert in package:			
NHES.Systems.Examples		✓ ¹⁰ / ₁₀	
Open new class in:			
This tab			\sim
Advanced options for storing start guesses			
Save changes in parameters and in initial values of states			
Additionally, save changes in the start attributes of:			
Iteration variables Iteration variables and torn variables			
Outputs, auxiliary variables, and states			
Only save start guesses for additional variables at start time. Other useage may cause unwanted changes in the model parametrization. These advanced options are only intended for saving start guesses and must not be used to continue simulations from times later than the start time.			
Advanced <<	OK	Can	cel



```
DUP
 CS (
  BV_openingNominal(k(start=0.001)),
  PID_BV_opening(
    I(k(start=0.1)),
                                               actuatorBus(opening BV(start=0.001), opening TCV(start=0.5)),
    addP(
     k1(start=1.0),
                                               boundary(port(T(start=421.1691093331664))),
     u1(start=0.030000001485),
     u2(start=0.030000001485)),
                                               boundary2 (medium (
    gainPID(k(start=-1.0)),
    gainTrack(k(start=-1.1111111111111111))),
    gain_u_m(k(start=5E-10)),
                                                      T(start=298.16763607879363),
    gain_u_s(k(start=5E-10)),
    limiter (uMax(start=0.999), uMin(start=-0.0009)),
                                                      T degC(start=25.017636078793657),
    null bias(k(start=0.0)),
    vMax(start=0.999),
                                                      d(start=998.544943058541),
    yMin(start=-0.0009)),
  PID TCV opening(
                                                      p bar(start=34.47380000000004),
    I(k(start=2.0)),
    addP(
     k1(start=1.0),
                                                      sat(Tsat(start=514.8425665422984)),
     u1(start=0.0086185),
     u2(start=0.0086185)),
                                                      u(start=104571.53362749857)), ports(h outflow(start={108023.9370710939}),
    gainTrack(k(start=1.1111111111111111))),
    gain u m(k(start=2.5E-09)),
                                                         - / - + - - + - ( 2//7200 01) ) )
    gain u s(k(start=2.5E-09)),
    limiter(uMax(start=0.5), uMin(start=-0.4999)),
                                                       Kt(start=0.013324090093760938),
    null bias(k(start=0.0)),
    u s(start=3447400.0),
                                                       Q mech(start=65056859.70577499),
    yMax(start=0.5),
    vMin(start=-0.4999)).
                                                       Q units(start={42261285.79911617,42261285.79911617}),
  TCV openingNominal(k(start=0.5)),
  delavStartBV(start=100.0),
                                                       Q units start(start={42261285.79911617,42261285.79911617}),
  p_Nominal1(k(start=3447400.0)),
   switch P setpoint(y(start=60000000.297)),
                                                       Obs(start={-9732855.946228676,-9732855.946228676}),
  valvedelav(k(start=100.0)),
  valvedelayBV(k(start=100.0))),
                                                       T a start(start=293.15),
 PID(
  I(k(start=2.0)),
                                                       T b start(start=293.15),
  addP(
                                                       T nominal(start=293.15),
    k1(start=1.0),
    u1(start=1.0),
                                                       bubble in(d(start=820.3581983078773), h(start=1013666.6724914373)
    u2(start=1.0)),
  gainPID(k(start=100000000.0)),
                                                       bubble out(d(start=989.8436373961912), h(start=191812.29519356362
  gainTrack(k(start=1.11111111111111112E-08)),
  gain_u_m(k(start=0.002374343174368348)),
                                                       d nominal(start=13.671247252758716),
  gain_u_s(k(start=0.002374343174368348)),
  k m(start=0.002374343174368348),
                                                       dew in(d(start=15.307197090608243), h(start=2803284.170249812)),
   k s(start=0.002374343174368348),
  limiter(
                                                       dew out(d(start=0.06816373081854721), h(start=2583886.8570257137)
    u(start=100000000.0),
    uMax(start=1E+60),
                                                       h a start(start=2997670.0),
    uMin(start=-1E+60)),
  null bias(k(start=100000000.0)),
                                                       h b start(start=2058530.3155751962),
  u_m(start=421.1691093331664),
  yMin(start=-1E+60)),
                                                       h is(start=2070197.5860370956),
 PID1(
  PID(
                                                       h out(start=2209318.4481315315),
    I(k(start=2.0)),
    Nd(start=10.0),
                                                       p a start(start=3337380.0),
    Ni(start=0.9),
                                                       p b start(start=10000.0),
    Td(start=0.1),
    Ti(start=0.5),
                                                       p inlet nominal(start=3337380.0),
    addP(
     k1(start=1.0),
                                                       p outlet nominal(start=10000.0),
     u1(start=1.0),
     u2(start=1.0)),
                                                       p ratio(start=0.0032668200354825697),
    gainPID(k(start=100.0)),
    gainTrack(k(start=0.011111111111111111))),
                                                       portHP(
    a_{2} m = m(k(c_{2}) + m(k_{1}))
                                                                                                                                                                           Integrated Energy Systems
```

| HYBRID and FORCE

Modeling Features

 Another method of importing initial conditions is using default output format



 This method can be used to alter parameters by altering the text file



RAVEN Interfacing

- Default initialization or final status text file is standard RAVEN input method
- Values identified in RAVEN input substituted

-2 1.000000000000001E-01 280 # nuScale Tave enthalpy Pressurizer CR.PID.k 1 -2 5.000000000000000E-01 9.99999999999997E-61 1.000000000000000E+100 280 # nuScale Tave enthalpy Pressurizer CR.PID.Ti 1 -2 1.000000000000001E-01 1.0000000000000000E+100 280 # nuScale_Tave_enthalpy_Pressurizer_CR.PID.Td 1 -2 0 0 280 # nuScale_Tave_enthalpy_Pressurizer_CR.PID.yb 1 5.000000000000000E-01 -2 0 256 # nuScale_Tave_enthalpy_Pressurizer_CR.PID.k_s 6 -2 5.000000000000000E-01 0 256 # nuScale_Tave_enthalpy_Pressurizer_CR.PID.k_m 6 -2 5.66000000000001E+00 0 280 # nuScale_Tave_enthalpy_Pressurizer_CR.PID.yMax 1 -2 0 0 280 # nuScale_Tave_enthalpy_Pressurizer_CR.PID.yMin 1



RAVEN Interface

- Executable made via Dymola and path input into RAVEN
 - User should make sure that "evaluate parameters at translation" option is disabled
- Dymola is subType "Dymola" in the input deck
- Input name type is "Dymolalnitialisation"



Model Analysis

- Scripting allows for manual creation of parameter sweep
 Method is: Translate(), import(), simulate()
- Dymola has internal parameter sweep methods
 - Only one parameter can be changed at once
- RAVEN interface uses standard input to accept new parameter methods
- Reminder from previous: models can use text reading for input, which can read dispatch information generated by another code



HYBRID Expansion

Development of concurrent model structures

- Modelica transient models
- Aspen HYSYS steady-state models
- Reduced order models based on Modelica transient modeling
- Subsystem costing information
- Full FORCE vertical integration
- Continued expansion of modeling capabilities



Why Aspen?

- Industry-standard thermodynamic and chemical analyses tool
- Allows for process changes, flow rearrangements
- Chemical reactors allow for process calculations





Why ROMs?

- Allow for shorter computation times
- Accuracy relative to trained model will be identical
- RAVEN contains many ROM training methods
- DMDc is nominal method of choice, allows for analysis of systems that use controllers



Questions?



HYBRID – What Is It?

- Hybrid is a collection of physical models written to characterize:
 - Thermal and electrical integration of different processes
 - Ramp speed
 - Evaluation of novel control schemes
 - Off-design system states
 - Dispatch feasibility
 - Safety limit approaches, considering control system effects

https://github.com/idaholab/HYBRID

