ANL/NSE-23/29

Extension of Plant Dynamics Code Capabilities for Simulation of TerraPower Pascal Reactor

Nuclear Science and Engineering Division

ANL/NSE-23/29

Extension of Plant Dynamics Code Capabilities for Simulation of TerraPower Pascal Reactor

Anton Moisseytsev
Nuclear Science and Engineering Division
Argonne National Laboratory

May 17, 2023

ABSTRACT

Argonne National Laboratory has been developing the Plant Dynamics Code (PDC) for design and transient analysis of supercritical carbon dioxide (sCO_2) Brayton cycles. In previous analyses with PDC, only indirect sCO_2 cycles, where the heat is being added through a heat exchanger, such as sodium-to-CO_2 HX, were analyzed. Under the U.S. Department of Energy Technology Commercialization Fund (TCF), Argonne cooperated with TerraPower to bring the Plant Dynamics Code to commercial market. The main focus of the TCF project is to extend the application base and the code usability by developing the capabilities to be able to simulate reactor systems with direct sCO_2 cycles.

These new PDC capabilities have been developed in application to the TerraPower's Pascal reactor concept. This report documents the Pascal reactor modeling with the PDC, including simulation of the Pascal split-expansion cycle, the development of the reactor module in the PDC, modeling of the Pascal's shutdown heat removal system in PDC, and other updates to the code. The report also describes the results of the steady state and transient demonstration of the newly developed code features.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 Introduction and Project Goals

Argonne National Laboratory (Argonne) has been developing the Plant Dynamics Code (PDC) [1] for design and transient analysis of supercritical carbon dioxide (sCO_2) Brayton cycles. The philosophy for PDC creation and development has always been a requirement to address and accurately calculate the specific features of sCO_2 cycles, such as CO_2 properties variations close to the critical point, and the effect of those properties' variations on the performance of the cycle components, such as compressors and coolers, as well as on the integrated performance of the entire cycle. PDC has been used extensively for analysis of sCO_2 cycles, mostly in application to nuclear reactors, such as sodium-cooled fast reactors [2,3,4]. The code has also been extensively validated using experimental data from integral loops [e.g., 5] and individual component testing.

Under the U.S. Department of Energy Technology Commercialization Fund (TCF), Argonne cooperated with TerraPower to bring the Plant Dynamics Code to commercial market. The TCF project is funded by U.S. DOE with 50% cost share from TerraPower. The main focus of the TCF project was to extend the application base and the code usability by developing the capabilities to be able to simulate reactor systems with direct sCO_2 cycles. In previous analyses with PDC, only indirect sCO_2 cycles, where the heat is being added through a heat exchanger, such as sodium-to-CO_2 HX, were analyzed. Adding the possibility to analyze direct cycles would significantly increase the applicability range of the code and remove one of the main barriers in adopting the code by the industry. These new capabilities are being developed in application to the TerraPower's Pascal reactor concept.

1.1 Pascal Reactor

Pascal is the name of a heavy water gas turbine reactor (HWGTR) developed by TerraPower. It features a direct-cycle architecture in which sCO_2 used for reactor cooling also serves as the working fluid in an sCO_2 power cycle. As illustrated in Figure 1, the reactor employs vertical pressure tubes and heavy water moderation, which provides a reliable heat sink in case primary cooling is lost. Addition information about the HWGTR and its power cycle can be found in [6].

The Pascal HWGTR features two unique elements not typically found in analyses of sCO_2 power cycles. First is direct coupling to the reactor core, which necessitates modeling of heat transfer between the sCO_2, fuel, pressure tubes, and moderator fluid. Next is the use of a split-expansion cycle, illustrated in Figure 2 and described in [7], in which a turbine upstream of the reactor reduces reactor operating pressure below the maximum pressure of the cycle. This reduces the strength requirements for reactor components and broadens reactor design space. Table 1 provides the design conditions for the Pascal sCO_2 cycle, with cycle point numbers corresponding to those in Figure 2.

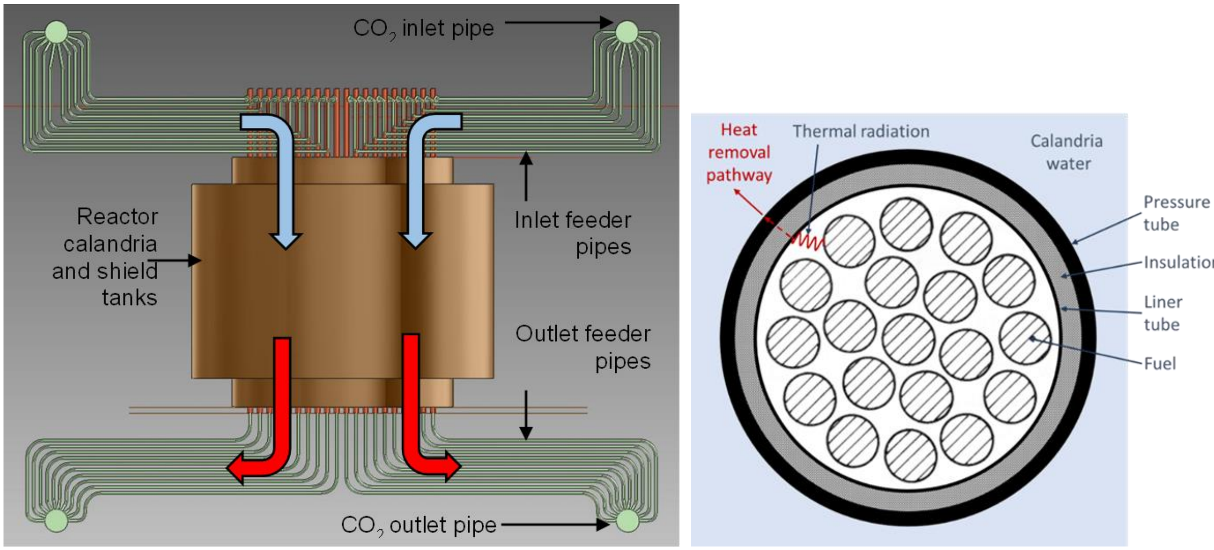


Figure 1. Pascal Reactor and Channel Structure.

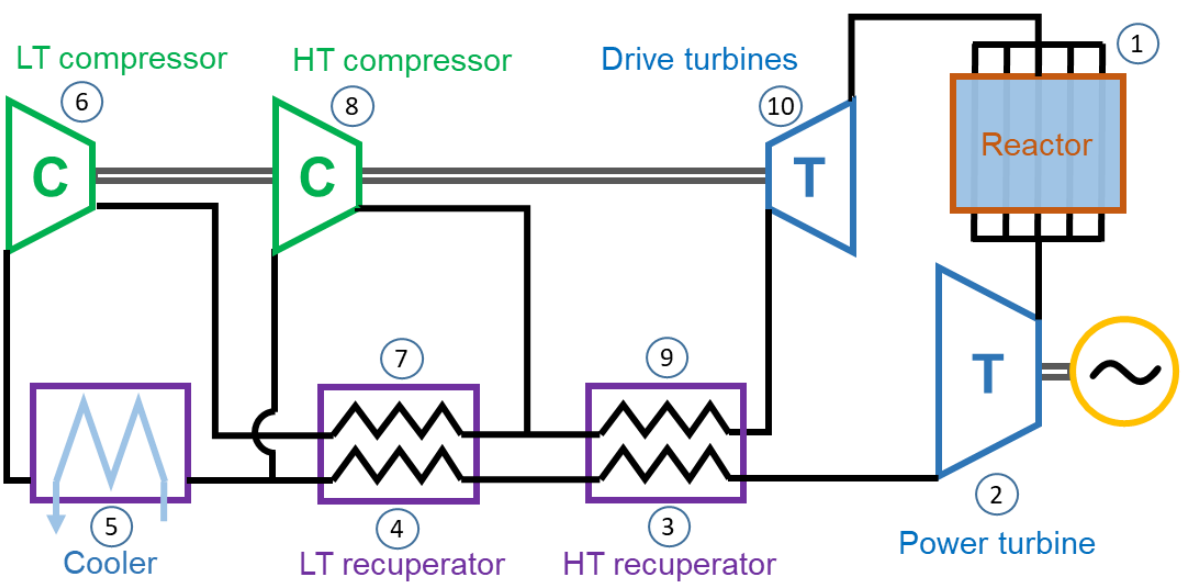


Figure 2. Pascal Split Expansion Cycle.

Table 1. Steady State Cycle Parameters

		Inlet		Outlet	
Index	Component	Pressure (MPa)	Temp. (°C)	Pressure (MPa)	Temp. (°C)
1	Reactor	15.00	400.9	14.08	550.0
2	Low-pressure turbine	13.65	549.7	7.97	483.6
3	HT recuperator (hot stream)	7.91	483.5	7.81	173.3
4	LT recuperator (hot stream)	7.77	173.1	7.67	65.0
5	Cooler	7.65	64.9	7.55	29.4
6	LT (main) compressor	7.50	29.3	22.35	55.3
7	LT recuperator (cold stream)	22.22	55.2	21.91	169.9
8	HT (re-) compressor	7.65	64.8	21.99	163.6
9	HT recuperator (cold stream)	21.81	167.0	21.70	440.6
10	High-pressure turbines	21.22	440.2	15.07	401.0

1.2 Project Workscope

The following tasks were identified for the TCF project to be able to model the Pascal reactor with PDC:

Task 1: Providing design information on Pascal

TerraPower will provide design information for the Pascal reactor sufficient for modeling with PDC. TerraPower will also provide general requirements for the desirable new code features.

Task 2: Establish baseline PDC model for Pascal

Argonne, with assistance from TerraPower, will use PDC to create a model of the sCO_2 cycle for Pascal reactor energy conversion system. The code predictions will be compared against previous Pascal calculations and design goals. The model will be used as a base for further code development in the project.

Task 3: Modeling and implementation of reactor components

Modeling of in-core structures and heat transfer of a direct-cycle application, in which CO_2 passes directly through the core, needs a sufficiently detailed modeling treatment of the core. Planned new code features include the ability to model temperatures of fuel, moderator, and additional structures in the core, as well as heat transfer between these structures and the coolant. Additionally, modeling of parallel channels (e.g., an average channel and a peak channel) based upon categorization by power-to-flow ratio as well as radially and axially varying power generation will help capture limiting behavior in the core such as peak temperatures. Pascal incorporates a moderator with significant heat capacity and heat transfer to the moderator needs to be modeled and included.

Task 4: Modeling and implementation of shutdown cooling system

The project will incorporate modeling of auxiliary equipment. A direct-cycle application will need to have additional auxiliary systems for safety that need to be modeled in PDC. These include a shutdown cooling system, resembling an LWR residual heat removal system, which would consist of a low-pressure-ratio CO_2 blower, heat exchangers, and valves. Additional equipment that may require more specialized modeling may include containment isolation valves.

Task 5: Select design and transient simulation

Argonne, with guidance from TerraPower, will select a set of design conditions and sample transients to investigate and test the new code on. Argonne will simulate those steady-state and transient conditions with the improved PDC and provide the results to TerraPower for evaluation.

The requirements for Task 1 were satisfied with information on the Pascal reactor presented above, along with some more detailed information of the Pascal reactor and cycle components needed for implementation of other tasks.

For Tasks 2-4, Argonne will modify PDC to include the modeling of new components and features and will test the newly developed code section for both steady-state and transient conditions. TerraPower will serve as a code reviewer for newly implemented features to help ensure that they meet TerraPower's needs. Implementation of Tasks 2 through 5 is described in the rest of this report, with the chapter number corresponding to the Task number.

It is also important to note for the contents of this report that anything outside the tasks described above was not included in the project. For example, even though the PDC has capabilities to improve cycle and component designs by doing parametric studies, such design improvements were intentionally not included in the work for this project. Likewise, even though some control investigation will be carried out in the last task, designing and optimizing a control strategy for Pascal is not the goal of this project and thus such analyses were not carried out.

2 Pascal Cycle Modeling

Task 2 in the TCF project was simulation of the Pascal sCO_2 split expansion cycle in PDC to establish a baseline for future transient simulations. This initial simulation was carried out in two steps. First, the cycle was modeled with the component performance (efficiencies) provided by TerraPower. Next, the heat exchanger and turbomachinery design was incorporated into the PDC model to compare their performance with the Pascal design calculations.

2.1 Given Efficiencies

The results of this simulation are presented in Figure 3. The cycle was modeled with two turbomachinery shafts, a shaft for a drive turbine (DTurb) and compressors (C) – low temperature (LT) and high temperature (HT), – and a shaft with a power turbine (PTurb) and a generator. Given by the Pascal configuration, the simulated cycle includes two recuperators, high- and low-temperature (HTR and LTR, respectively), a cooler (Cool), and a reactor (Rx) located between the drive and power turbines. The input for the PDC simulation also included the boundary cycle conditions, such as 550 °C reactor-outlet temperature, 22.3 MPa maximum cycle pressure, and low temperature compressor inlet conditions (inputs are highlighted in green in Figure 3).

For this first stage of the Pascal sCO_2 cycle simulation, the component efficiencies and pressure drops were specified in the PDC input as following:

- *Reactor*: given outlet temperature, power, and pressure drop,
- *Turbines and compressors*: given efficiencies (derived from inlet/outlet conditions),
- *Recuperators*: given effectiveness (from inlet/outlet temperatures) and pressure drop,
- *Cooler*: given pressure drop (integrated into the outlet pipe),
- *Piping*: given pressure drop (matched by changing the pipe inner diameter)
- *Valves*: no effect,
- *Flow split*: given (42%).

The results in Figure 3 agree very well with the Pascal design conditions in Table 1. All pressures are matched at worst *within 0.01 MPa*; all temperatures – *within 0.1 °C*. The net cycle efficiency of 39.4% also agrees very well with the 38.9% provided by TerraPower.

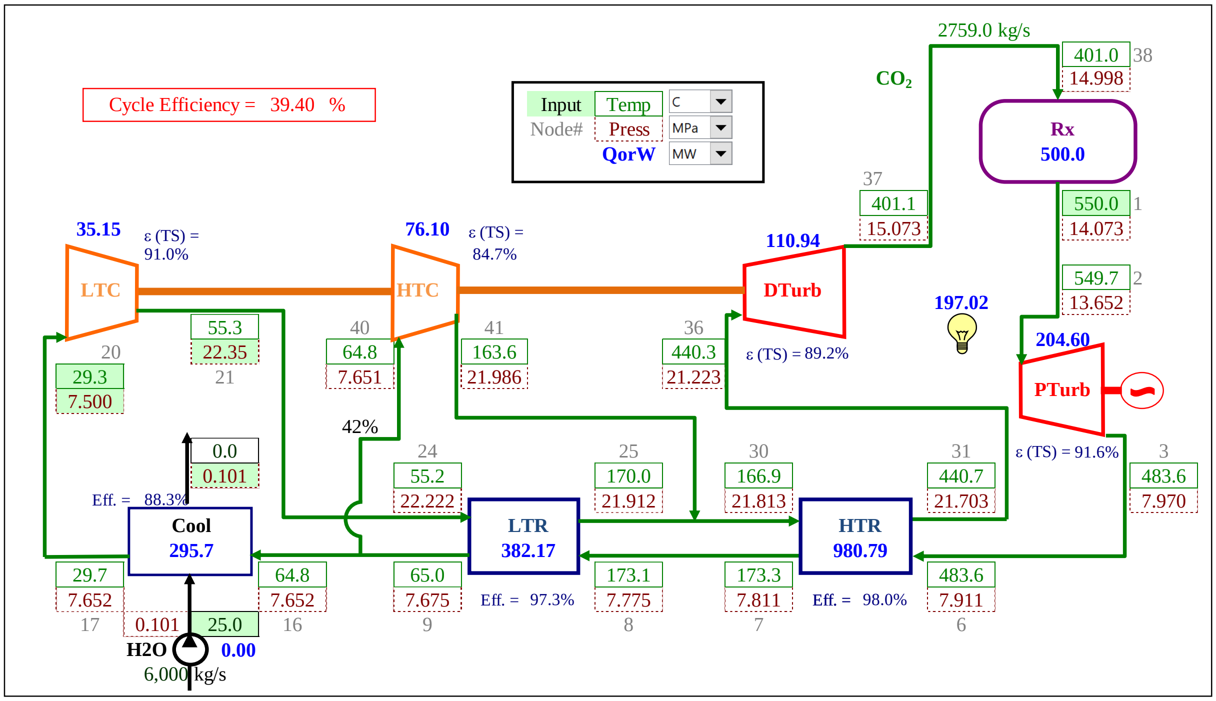


Figure 3. PDC Results for Pascal sCO_2 Cycle with Given Efficiencies.

2.2 *Cycle with Component Design*

After the overall Pascal cycle model was established in PDC, it was gradually extended to include the actual component design, such as for the heat exchangers and turbomachinery. That detailed component modeling is required in PDC for transient simulation. For example, the exact heat exchanger wall mass is an important parameter to characterize the transient response of the heat exchanger and can only be obtained from the detailed design information (i.e., efficiency and pressure drop used in previous step are not sufficient for transient simulation). Likewise, detailed turbomachinery design is required to calculate the maps for off-design performance of turbines and compressors in transients.

2.2.1 *Heat Exchangers Design*

For the next stage of cycle modeling, the PDC model was extended to include the heat exchangers (cooler and recuperators) designs provided by TerraPower. It was realized, though, that the Pascal cycle (and component) design is still in preliminary phase and not all the heat exchanger design information required for the PDC simulation is available. An example of such information includes a zigzag angle for the printed circuit heat exchangers (PCHE) for both recuperators and the cooler. Therefore, this parameter, along with some other inputs, was guessed and adjusted in the PDC model to match the performance of the heat exchangers as close as possible.

The PDC results with integrated heat exchanger designs are shown in Figure 4. The performance of these compact diffusion-bonded heat exchangers predicted by PDC is close to the TerraPower design calculations. For example, the recuperator effectiveness is around 95% for both units. The cooler effectiveness is more than 98%, as a close approach at the cold end is required in this cycle to achieve the specified compressor-inlet conditions. The cycle efficiency decreased slightly form 39.4% to 39.0%, but it is still close to the Pascal design value of 38.9%.

One of the interesting results from this PDC simulation is the temperature profiles inside the cooler shown in Figure 5. In the Pascal sCO_2 cycle, while the minimum pressure is above the critical value, the minimum temperature goes below critical, meaning that there would be a pseudo-critical transition somewhere in the cooler. Due to CO_2 properties variations, the results in Figure 5 shown a double-pinch-point characteristic of the cooler temperature profiles. The first pinch point, at around 0.6 m, is calculated when CO_2 goes through a pseudo-critical point transition and experiences a condensation-like behavior at almost constant temperature. The second pinch point is calculated at the CO_2 outlet (z=0), where CO_2 is being cooled below its pseudo-critical temperature and approaches the water inlet temperature.

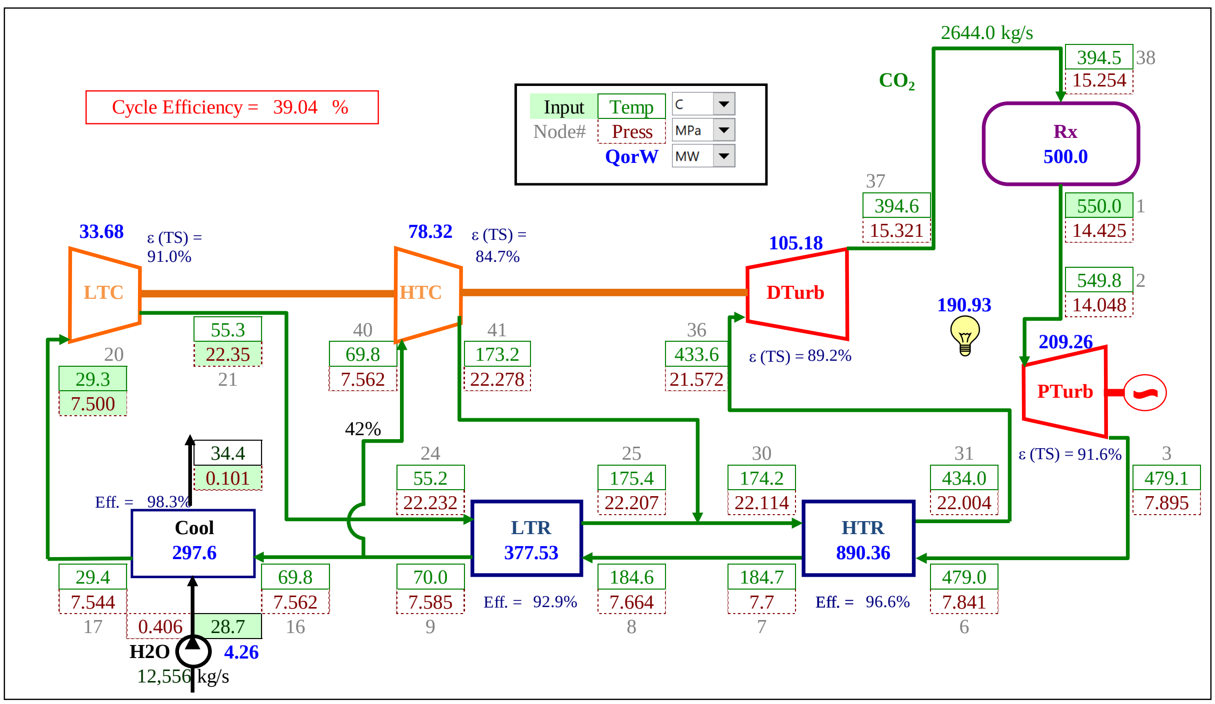


Figure 4. PDC Results for Pascal sCO_2 Cycle with Heat Exchanger Designs.

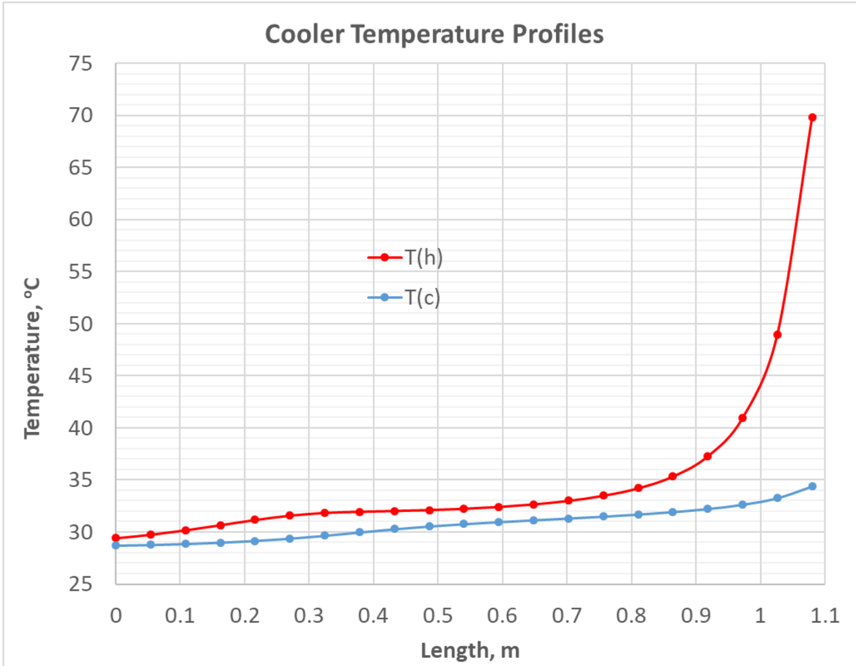


Figure 5. PDC Results for Temperature Profiles in Pascal sCO_2 Cycle Cooler.

2.2.2 *Piping and Valves*

The more detailed sCO_2 cycle piping information, such as length, diameter, number of bends, and bend radii have been provided by TerraPower and integrated into the PDC model at this stage. In addition, the throttling valves, located at the inlet of each turbine and outlet of each compressor, were included into the PDC model with valve pressure drops at design (fully open) conditions also modeled in the PDC. The PDC results with the included piping and valves information are provided in Figure 6. Overall, these results are very similar to those in the previous figure.

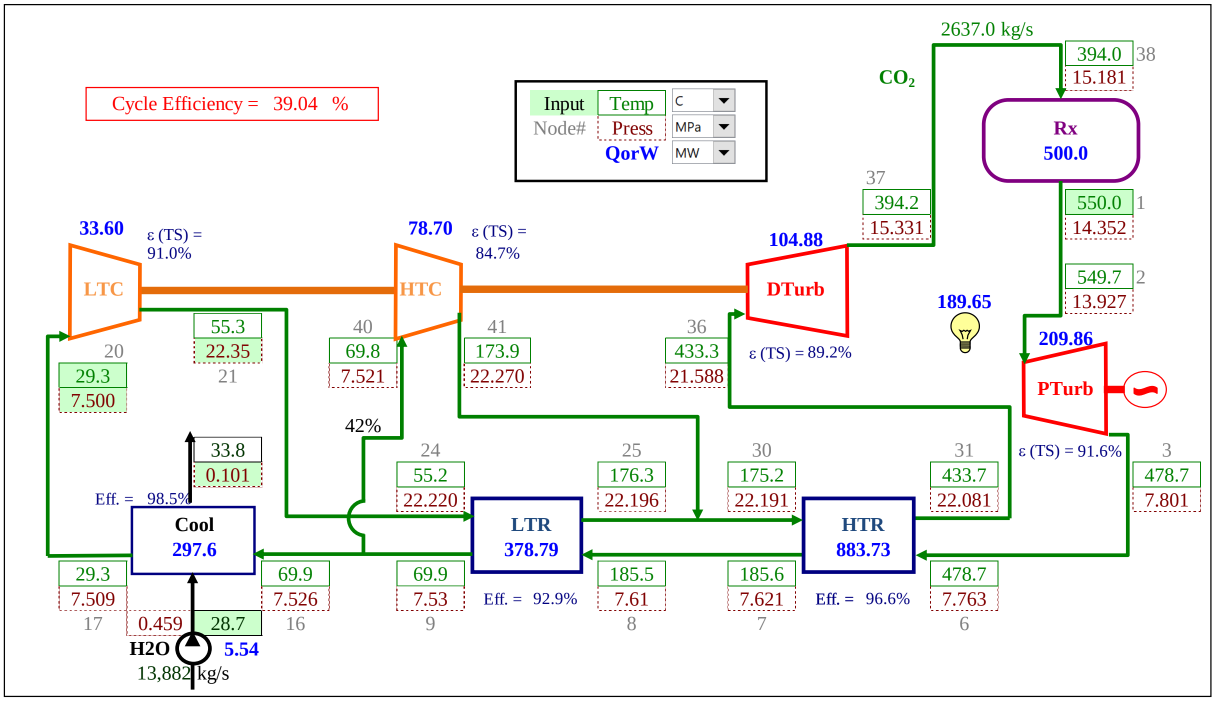


Figure 6. PDC Results for Pascal sCO₂ Cycle with Piping and Valves.

2.2.3 *Three Shaft Configuration*

In PDC simulations carried out so far, the turbomachinery shaft configuration from Figure 2 was assumed, where the drive turbine (DTurb) drives the two compressors (LTC and HTC). At this point, though, TerraPower has indicated that this configuration is not necessarily fixed and there exists another option where each of the two compressors is driven by its own drive turbine. Therefore, this configuration would have three turbomachinery shafts: two "drive" shafts with compressor and drive turbine, and one "power" shaft with the power turbine and a generator. The drive shafts would be disconnected from the grid and therefore may operate at different speeds from the power turbine. Also, the configuration with three shafts would allow operating the compressors (and corresponding drive turbines) at different speeds which could be beneficial for conditions where one compressor operates close to the CO_2 critical point.

In order to assess the PDC capability to simulate multiple turbomachinery shafts, both in steady-state and in transients, the PDC Pascal model has been updated to include three shafts. This PDC model is shown in Figure 7, with all the cycle components, connecting piping, and control valves. Figure 7 also shows the PDC cycle nodes (inlet and outlet for each pipe, in green numbers) for the pressure and temperature calculations. The drive turbine is split between the two: drive turbine for high temperature compressor (DT_HT) and drive turbine for low temperature compressor (DT_LT). Note that the two drive turbines operate in parallel and thus have similar inlet pressures and temperatures and outlet pressure (the small differences could result from inlet pipes and throttle valves). The flow split between the drive turbines is specified as 30%/70% between low and high temperature shafts. Until turbomachinery design is implemented (in the next section), the same efficiency for the two drive turbines, 89.2%, is assumed.

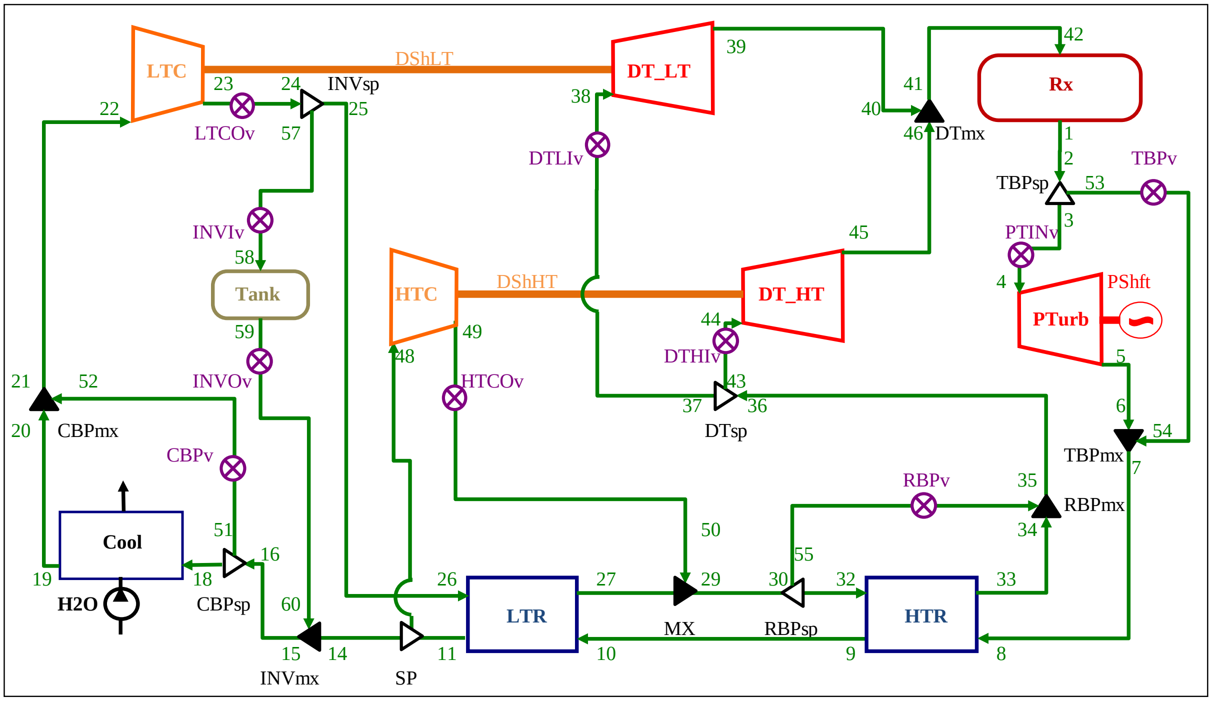


Figure 7. PDC Pascal sCO_2 Cycle Model with Three Turbomachinery Shafts.

The PDC steady-state results with the three shafts are shown in Figure 8. The cycle efficiency (38.9%), as well as all other conditions are close to those in previous figures. It is important to note however that the results in Figure 8 have not been refined yet to achieve perfect power balance for the two drive shafts. For each of these shafts, the calculated turbine power is less than the compressor power, and thus is not sufficient to drive the compressors. This is because in the current steady-state model in PDC, a perfect shaft balance is not a requirement. A (small) motor-generator was added to each shaft in the PDC model (it is a required component for turbomachinery shafts in PDC). Thus, for steady-state calculations, PDC calculates the actual shaft power balance, assuming that the net difference will be compensated by that motor-generator, with negative value corresponding to required external power input for the motor. In other words, there is no difference in the PDC steady state model between the shafts connected or not connected to the grid. That difference is only implemented for the shaft speed equations and treatment in dynamics. Therefore, the current results for the shaft power balance will need to be refined before the transient calculations could proceed. That refinement, however, would depend on actual turbine and compressor performances (rather than using given efficiencies), which is implemented in the next section. The shaft power balance will be dealt with later in Section 2.2.5 of this report.

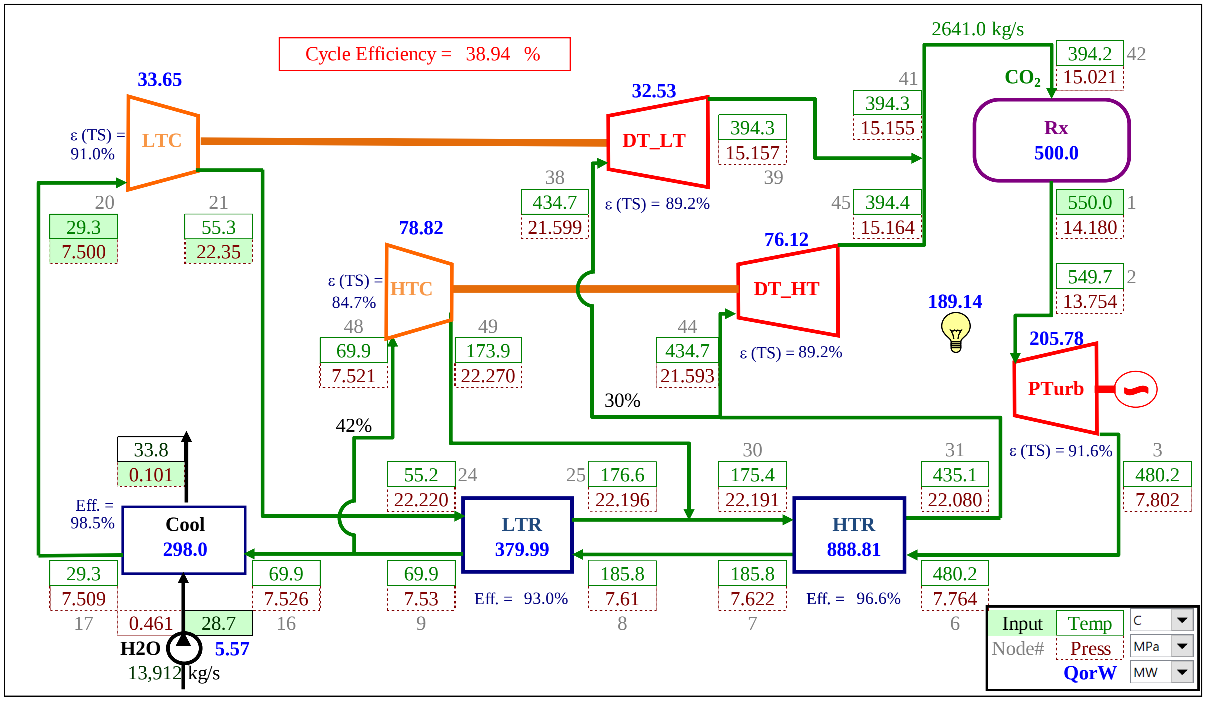


Figure 8. PDC Results for Pascal sCO_2 Cycle with Three Turbomachinery Shafts.

2.2.4 *Turbomachinery Design*

Compared to the heat exchangers described in Section 2.2.1, less details on the turbomachinery design is currently available for the Pascal reactor. Aside from the inlet/outlet conditions in Table 1, only efficiencies *targets* currently exist for turbines and compressors. No detailed design, like number of stages and stage geometries, could be provided for this project to be simulated in PDC. The only available information is that turbines should be axial, and the compressors are centrifugal. Also, shaft rotational speeds are not known, except for the power turbine that operates at synchronous speed of 60 Hz (3600 rpm).

As described in Introduction, the design of the Pascal reactor and the cycle components is not the goal for this project. Therefore, the following approach was selected for the turbomachinery design. Some reasonable assumptions on the design goals (like rotational speed and number of stages) will be made, under which the PDC turbomachinery design subroutines will be used to come up with the component designs and performance. As long as target efficiencies can be achieved with those reasonable assumptions and corresponding PDC designs, no further effort will be made to optimize the turbine or compressor designs. Therefore, it is expected that the resulting turbomachinery designs would not be optimal, but as long as all PDC design requirements are satisfied and the performance is acceptable, the designs will be considered "good enough" for this project.

The resulting design parameters and performance characteristics for the Pascal cycle turbomachinery obtained with PDC are provided in Table 2. The first few rows in the table, from "Speed" to "Number of stages" are the input to the PDC calculations, while the rest of the information is the result from the turbomachinery design subroutines. It was found that both the power turbine and low-temperature compressor and drive turbine could be designed for 60 rps shaft speed, while better designs for the high temperature compressor and its drive turbine are obtained at 120 rps speed. The number of stages was selected as the minimum value that satisfy the efficiency targets. Note that in the PDC calculations, the total-to-static (T-S) efficiency is used for the power balance (work) for each component, while the turbine or compressor outlet temperature for cycle calculations are obtained using static-to-static (S-S) efficiency. As Table 2 demonstrates, there could be a significant difference between those two definitions, which is a common feature for sCO_2 cycles. Since the turbine or compressor power is more important for cycle efficiency, the T-S efficiencies from PDC calculations are used to satisfy the efficiency targets for Pascal reactor.

The resulting cycle performance with these designs is shown in Figure 9. The cycle efficiency changed slightly to 39.8%. Likewise, all other cycle conditions changed slightly but remain close to previous results and original Pascal conditions.

Table 2. Pascal Turbomachinery Design with PDC

Shaft	*Power*	*LT Drive*		*HT Drive*	
Speed, rps	60	60		120	
Component	**PTurb**	**DT_LT**	**LTC**	**DT_HT**	**HTC**
Turbine/Compressor	Turbine	Turbine	Comp	Turbine	Comp
Type	Axial	Axial	Centr.	Axial	Centr.
Blades	Shrouded	Shrouded	Unshrouded	Shrouded	Unshrouded
Number of stages	3	3	2	1	2
Min hub radius, m	0.30	0.28	0.11	0.22	0.16
Min/max blade height, cm	15.8/25.7	5.1/6.9	3.3/7.5	7.7/10.0	4.2/6.7
Max wheel diameter, m	1.11	0.69	0.8	0.64	0.66
Max diameter, m	1.11	0.69	1.29	0.64	1.53
Total length, m	1.66	0.49	0.44*	0.55	0.49*
Max Mach number	0.416	0.316	0.260	0.552	0.713
Exit speed, m/s	46.0	33.4	42.1	57.6	64.2
Efficiency, S-S, %	93.6	91.6	94.0	93.7	91.7
Efficiency, T-S, %	**92.4**	**90.4**	**90.2**	**90.1**	**88.9**
Power, MW	**203.89**	**33.74**	**32.96**	**73.15**	**71.56**

*Centrifugal compressor length is an estimate for wheel axial length only

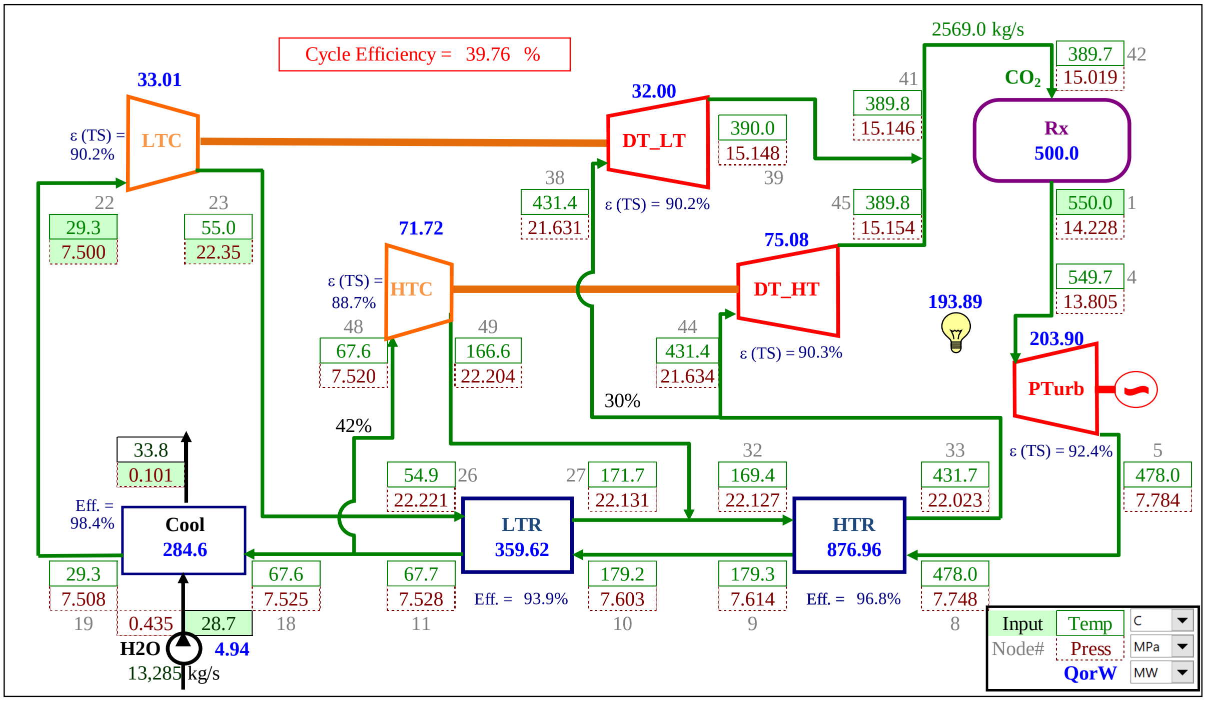


Figure 9. PDC Results for Pascal sCO₂ Cycle with Detailed Turbomachinery Designs.

2.2.5 *Drive Turbines Shaft Power Balance*

With the turbomachinery designs implemented, the drive shafts power balances can now be refined. In PDC, the generator (shaft) power balance is calculated as:

$$W_{gen_j} = \left[\sum_{i=1}^{nT_j} W_{turb_i} \cdot (1 - l_{mech}) - \sum_{i=1}^{nC_j} W_{comp_i} \cdot \frac{1}{1 - l_{mech}} \right] \varepsilon_{gen_j}$$

where:

W_{turb}, W_{comp} = turbine and compressor power (work),
l_{mech} - mechanical (frictional) losses,
ε_{gen} - generator efficiency.

For the drive shafts, $l_{mech} = 0.01\ (1\%)$ is assumed similar to other shafts. $\varepsilon_{gen} = 1.0\ (100\%)$, since these shafts should include a generator at design conditions and thus there should not be an associated generator loss.

With these inputs, PDC calculates shaft balances of -1.68 MW and +1.88 MW for the low-temperature and high-temperature shafts in Figure 9. Because the shaft power balances are of opposite signs, both can be improved by changing (slightly) flow split between the drive turbines. It was found that changing the flow split from 70%/30% to 68.8%/31.2% would result in much improved power balance in both shafts. The results with adjusted flow split are provided in Figure 10. For both shafts, the net power is reduced to approximately 0.1 MW. More refinements to this balance will be made later, after the reactor model is implemented.

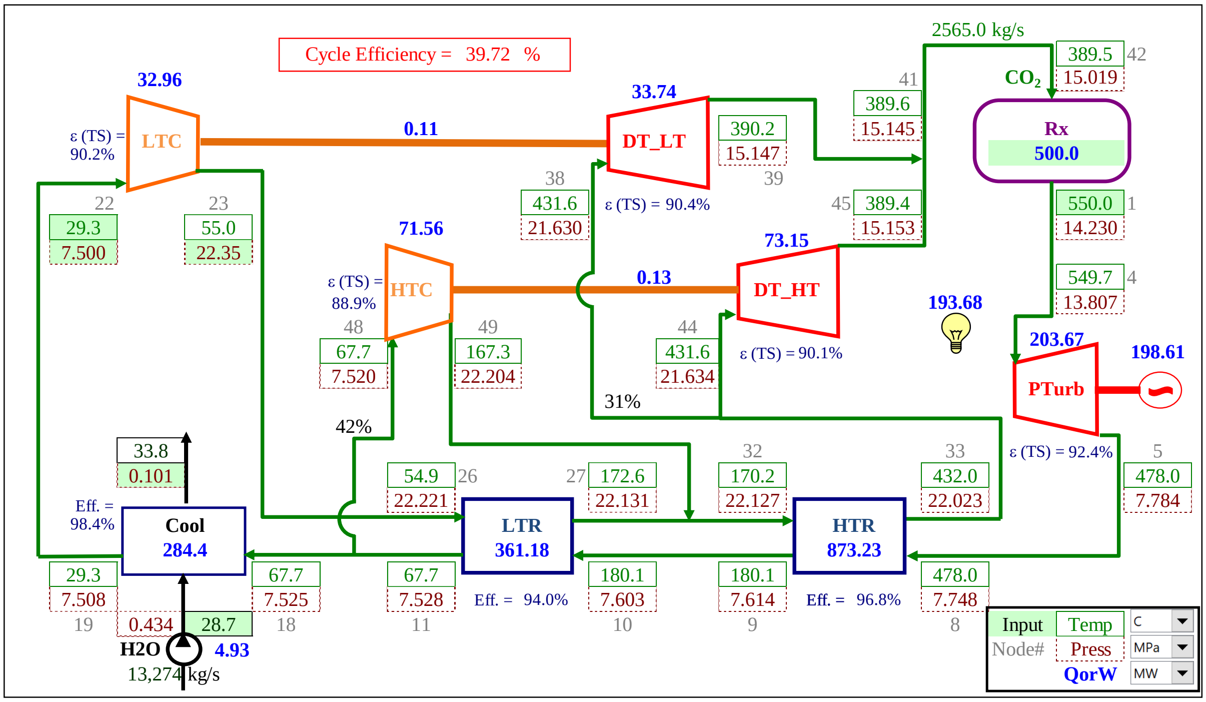


Figure 10. PDC Results for Pascal sCO_2 Cycle with Shaft Power Balance.

The current cycle conditions in Figure 9 are those obtained with detailed designs for all cycle components and piping, and thus will be used as the reference cycle conditions for the reactor modeling described in the next chapter.

3 Reactor Module

Simulation of the Pascal reactor component in PDC was the major topic of this project. Not only adding such a component would allow simulation of the entire Pascal plant, but also extending PDC capabilities to include a reactor module in general would allow modeling of an entire class of direct sCO_2 cycles, where the reactor coolant is the same as the one used in the energy conversion system. Addition of a reactor module to PDC would allow direct simulation of the entire plant with PDC, without a need to couple to a dedicated reactor code.

In previous work, the PDC was primary developed for simulation of indirect sCO_2 cycles, where heat is added to the cycle in a Heat Addition Heat Exchanger (HAHX) (also called Reactor Heat Exchanger, RHX). For liquid-metal cooled reactors, that would be sodium-to-CO_2 HX for sodium-cooled fast reactors, or lead-to-CO_2 HX for lead-cooled fast reactors.

At the same time, for the validation work with the data from small-scale loops, an electrical heater component was added to PDC [8,9]. The PDC treatment of an electrical heater was very similar to a shell-and-tube heat exchanger, except there was no primary-side fluid; rather, heat was directly added to the HX tubes to simulate electrically heated rods.

In the most recent work prior to this project, the electrical heater component option was extended to allow for simulation of a prismatic helium-cooled gas reactor [10,11]. For this reactor type, a matrix material was added between the coolant tubes and the fuel channels, to represent the graphite structure and heat transfer from fuel to coolant through that structure. However, other than this change, the reactor behavior was simulated very similarly to the previous model of an electrical heater.

Because of these recent developments, the main goals of this project were identified to enhance the PDC capabilities to be able to model the Pascal reactor geometry (Figure 1) and also model the features important to the Pascal reactor steady-state and transient analysis, including:

- ***Channel-type structure***, where coolant and fuel channels are located within the channel tubes surrounded by a moderator,
- Simulate ***heat transfer*** not only from fuel to coolant, but also ***between coolant and the moderator***, which is especially important in transients,
- Add provisions to simulate ***heat loss*** (from the moderator),
- Update ***pressure drop and heat transfer equations and correlations*** to be able to model Pascal specifics, such as grid plates and other channel tube form losses,
- Add a treatment of unheated (no fuel) ***inlet and outlet sections*** of coolant channels,
- Provide capability to specify an arbitrary ***axial power profile*** in fuel (electrical heater always assumes uniform linear power),
- Provide a capability to calculate ***peak temperatures*** in the reactor by allowing of simulation of ***more than one coolant channels*** (all previous calculations were done on average basis),
- Update material properties formulations in PDC to ***include reactor-specific materials***, such as uranium oxide.

In addition to these features important for the Pascal design, other improvements were introduced to the PDC reactor module to better simulate aspects important to any reactor design. Examples of such features include introduction of form pressure losses, multiple-radial-node treatment in the fuel, and others. These improvements, general to all reactor types, are also described in this section.

The implementation of all these modifications is described in the following sub-sections of this chapter. The reactor module is still developed based on previous electrical heater module. However, the modifications are done in a way to allow using the module for a wide range of designs, starting from a previously modeled electrical heater to various reactor designs, including pin-type geometry, channel-type geometry for Pascal, and prismatic VHTR-type geometry for gas -cooled reactors.

3.1 Reactor Types

Figure 11 shows the three reactor types that are simulated in PDC. The Prismatic Type has already been implemented in previous work [10]. In this case, “matrix” represents the graphite matrix. “Tube” is either the coolant liner (tube) on a graphite matrix (if a design includes one) or a part of graphite matrix surrounding the coolant channel (in this case, the tube will be of same material as the matrix). One of the recent inputs required for this type was the number of fuel channels per one coolant channel. That input will also be useful for the other types described below.

The Pin Type is essentially the same geometry that has been assumed in previous simulations for electrical heaters. For this design, “tube” is a fuel element (or electrical rod) cladding. The only modification for this type implemented in this work as an addition of the reactor shell, which is simulated as “matrix” to preserve the same nomenclature as for the other reactor types. Modeling of the shell “matrix” will be very similar to the channel tube for the Channel Type described below. For this reactor type, the PDC will still simulate one coolant channel which surround one fuel pin. Therefore, for this type, it is always one-to-one relationship between number of fuel pins and the number of coolant channels.

The Channel Type will be a new reactor type added specifically for simulation of the Pascal reactor. For the most part, modeling is similar to the Pin Type configuration, but there are significant differences which required modifications to the code to simulate this geometry. These differences include:

- “Matrix” will represent the channel tube,
- Coolant channel represents all coolant within the channel tube. Therefore, the input of number of fuel pins per coolant channel will be used. For Pascal reactor, this input is equal to 21,
- The moderator will be modeled as a constant temperature heat sink as described in Section 3.2.

Because development of the Channel Type reactor module is the primary focus of this project, this report will be focusing on implementation of this reactor type. Any additional considerations for other reactor types will still be mentioned, but only when necessary.

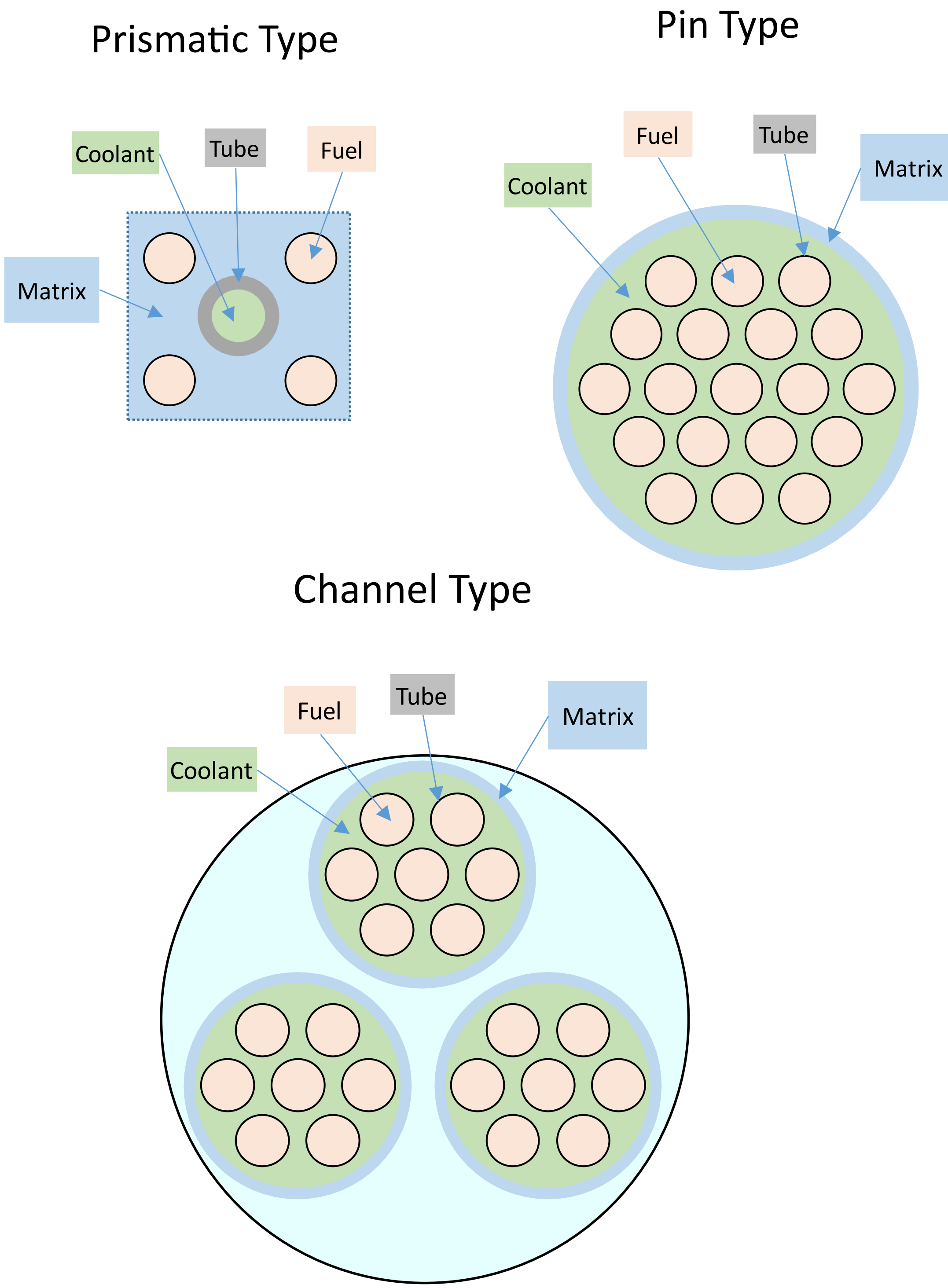


Figure 11. Reactor Types in PDC.

Other than special treatment of peak channels described in Section 3.7, the calculations for all types are still carried out for an average coolant channel, under an assumption that all fuel pins and coolant channels are identical. For the Channel Type reactor, it is assumed that all channels are identical.

Aside from the geometry considerations and input, one of the most significant differences between reactor types in Figure 11 is the heat flow path. For the Prismatic Type, the heat from fuel goes to the matrix, then to the coolant type and to the coolant. For the other two types, the heat flow is from fuel to the coolant through the "tube" (cladding), and then from the coolant to the "matrix" tube. Therefore, the heat transfer equations will be formulated differently for the Prismatic Type. It will also affect the simulation of heat loss described in Section 3.2. However, for any geometry, the total heat will be conserved (in steady state), and the amount of heat removal by coolant will be less than the heat generated in fuel by the amount of the heat loss from the reactor (matrix).

To simplify understanding of the new reactor module features, implementation of these new features is described further in this report as they are applied in the PDC steady-state models. The special consideration for the transient part of the code are provided later in Section 3.9.

3.2 Channel Type

A new type of reactor component for Channel Type was introduced in PDC. Therefore, there is currently three types of the reactor geometry supported by the code (refer to Figure 11):

1. Pin-type or electrical heater,
2. Channel type,
3. Prismatic type.

The input file for the reactor component was split into three sections to provide the input for each type, if needed. In addition, there is a general section of the input file where input common for all types is provided. The general arrangement of the new input file is shown in Figure 12. The input file is described in Section 3.10.

```
<General input>
Reactor type [1=Pins, 2=Channel, 3=Prismatic]
2
-------------------- Pin type / El. Heater ----------------------
<Input>
------------------------ Channel type ---------------------------
<Input>
----------------------- Prismatic type  -------------------------
<Input>
```

Figure 12. Reactor Component Input File Structure.

The following subsections describe the code modifications needed to introduce the Channel Type and model its specific features.

3.2.1 *Channel Tube*

The channel tube is now simulated in PDC using the existing "matrix" component of the reactor, introduced previously for the prismatic type. The new input are provided for the channel tube, including the inner diameter and thickness.

The material input was already introduced for prismatic type, and thus no changes to the code or input file were required. However, the Pascal channel tube has a complex structure with multiple material layers (see Figure 1). In the implementation of the tube modeling in PDC it was decided that the additional complexity of multi-layered tube is not warranted for the plant simulation purposes. Instead, a new equivalent material, "PCT" for Pascal Channel Tube, was introduced into the PDC material properties database. The thermal conductivity, density, and heat capacity of this new material were calculated outside the code to preserve the corresponding cumulative properties of the multi-layered structure, while preserving the tube thickness. These new properties are still formulated and implemented in PDC as temperature depended.

The heat transfer equations were extended to include heat transfer from coolant to the channel tube. The thermal resistance of the channel tube is calculated using standard resistance for a cylindrical structure, in the same way as it is calculated for cladding or heat exchanger tubes. One radial node treatment is applied to channel tube, consistent with all other components. The heat transfer (heat loss) from channel tube to the moderator is discussed in Section 3.3. It is noted though that unless heat loss is simulated, the tube temperature will be equal to the coolant temperature, at each axial node, in steady-state calculations.

The treatment of the shell in the Pin Type geometry (Figure 11) is implemented very similarly to the channel tube described here. The only significant difference is that for the Pin Type, coolant channel calculations are done for the coolant surrounding one fuel pin. Therefore, the shell heat transfer perimeter is split between all coolant channels (fuel pins) for the heat transfer calculations from the coolant to the shell. Again, the assumption here is that all coolant channels (fuel pins) are identical in the one-dimensional treatment in the PDC.

3.2.2 *Structure Rods*

The Pascal channel design includes a structural central rod in each channel, as shown in Figure 13. The provisions for simulating these rods have been added to PDC, but only to include the effect of the reduction in the coolant flow area and increase in wetted perimeter for pressure drop. The structure rods do not generate heat. It is assumed that these rods do not participate in heat transfer at all, i.e., their heat capacity effects are neglected in transients (in steady-state, rod temperatures will be equal to that of the coolant).

For general use, the number of rods is added as new input, along with the rod outer diameter. (If multiple rods are present, they are assumed to be of the same diameter.) A case without structure rods (N=0) is also supported since their area and perimeter will automatically be set to zero, with no effect on the coolant channel. The structure rods are only supported for the Pin and Channel Types. For the Channel type, the input is for the number of rods per coolant channel (=1 for Pascal). For the Pin Type, it's the total number of structure rods in the reactor.

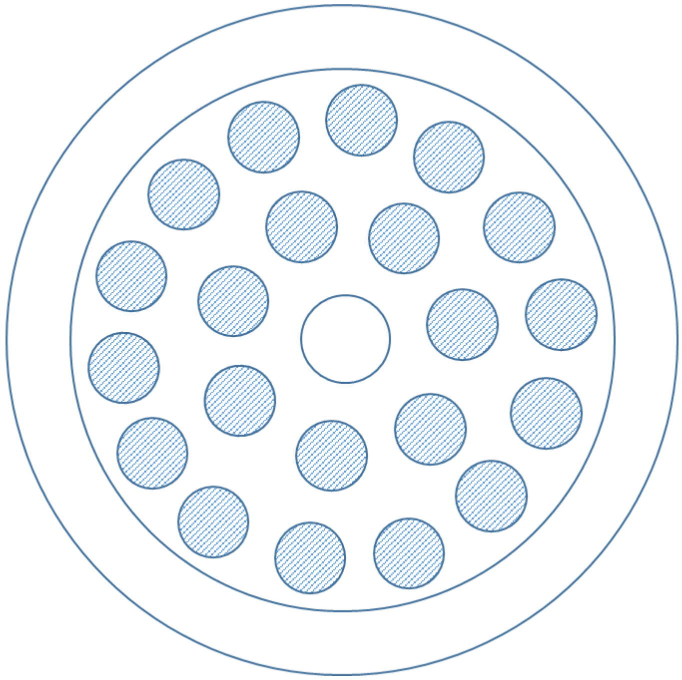


Figure 13. Pascal Channel with Structure Rod in the Middle.

3.2.3 *Form Pressure Losses*

In previous analysis, only frictional pressure drop in coolant was included either for the electrical heater or the prismatic reactor. To simulate the Pascal features such as grid plates and geometry changes, form pressure drop has been added. The form loss coefficient is a new input. The total pressure drop is now calculated as:

$$\Delta p_{total} = \Delta p_{fric} + \Delta p_{form} = 2f\rho V^2 L/D_h + 1/2K\rho V^2 \quad \textbf{(1)}$$

where

Δp_{total}, Δp_{fric}, Δp_{form} = total, frictional, and form pressure drops,

f = friction factor,

ρ = coolant density,

V = coolant velocity,

L = channel length (discussed below),

D_h = hydraulic diameter,

K = form loss coefficient.

Since the coolant conditions in the PDC reactor module are calculated on an axial mesh with a number of nodes, Equation (1) is applied to each node. Length is then the node length. The form loss coefficient is also distributed along the reactor length. This approach is implemented because it is assumed that majority of the form loss (at least in Pascal design) will come from grid spacers, which are distributed along the channel length. Under this assumption, the form loss in Equation (1) is used in PDC in the following way for each axial node *i*:

$$K_i = \frac{\Delta L_i}{L} K_{core} \tag{2}$$

where,

ΔL_i = axial node length,

L = total channel (reactor) length,

K_{core} = total core loss coefficient (user input).

3.2.1 *Non-Uniform Axial Mesh*

In previous calculations a uniform axial mesh was used, with the only input provided by the user was the number of axial (i.e., along the channel) nodes. For more flexibility, a new input was introduced to allow users to specify axial node lengths. The input is provided for the array of axial node lengths. That array is re-normalized in the code to preserve total reactor (channel) length. Therefore, this input allow either to specify the axial lengths directly or to provide fractional relationship between the node lengths. A uniform axial mesh can still be used if the user provides the same values for all nodes, which will then be normalized to have the same axial lengths for all nodes.

3.2.2 *Heat Transfer Perimeter*

In all previous calculations, the only structure that participated in heat transfer with the coolant was the tube, either the fuel cladding for the pin type or coolant channel tube for prismatic type. With introduction of other structures, like channel tube and structure rods, a distinction had to be made between the wetted perimeter and heat transfer perimeter. The former is used for the hydraulic diameter and pressure drop calculations and includes all surfaces in contact with the coolant. The latter is used for heat transfer to the coolant and include only surfaces that participate in the heat transfer. For the Pin and Channel types, the heat transfer perimeter is split between the cladding and the channel tube/shell surfaces for the corresponding heat transfer calculations.

3.3 Heat Loss

One of the distinct features of the Pascal reactor is the moderator surrounding the channel tubes (shown in light blue in Figure 11). The moderator is included for neutronics purposes (to moderate neutrons), but it also provide a significant heat sink during transients. Therefore, it is important for the Pascal reactor simulation to account for the heat transfer from the channel tube to the moderator. In the project workscope it was stated that due to sufficiently large mass of the moderator, the moderator temperature is not expected to change much in transients, and, therefore, it is sufficient to model this heat transfer as *a constant temperature heat sink*. Since it is the channel tube that is in contact with the moderator, this heat path is now modeled in the PDC as heat loss from the channel tube or "matrix" in PDC nomenclature in Figure 11.

The heat loss from the matrix to the constant temperature heat sink is modeled, at each axial node i, as:

$$\Delta Q_i = HTC \cdot \Delta S_i \cdot (T_{mo,i} - T_{HS}) \quad \textbf{(3)}$$

where,

ΔQ_i = heat loss in axial node *í*,

HTC = heat transfer coefficient for the heat loss (user input),

$\Delta S_i = P \cdot \Delta L_i$ = surface area for heat loss in axial node *i*

P = channel tube outer perimeter,

$T_{mo,i}$ = matrix (channel tube) outside surface temperature in axial node *i*.

T_{HS} = heat sink temperature (user input).

The *HTC* and heat sink temperatures are the new user inputs. In the calculations presented below, these inputs were adjusted to match the Pascal design heat loss of 13 MW (out of 500 MW total power).

The matrix temperature equations were modified to include the heat loss from Equation (3). This is in addition to the heat transfer between the coolant and the matrix (channel tube). In steady state, the heat transfer rate from the coolant to the channel tube is equal to the heat loss from channel tube to the moderator, and the heat loss can be formulate based on the coolant temperature:

$$\Delta Q_i = \frac{T_{av,i} - T_{HS}}{\frac{1}{h_c \cdot \pi D_{in}} + \frac{\ln{(D_{out}/D_{in})}}{2\pi k_m} + \frac{1}{HTC \cdot \pi D_{out}}} \Delta L_i \quad \textbf{(4)}$$

where,

$T_{av,i}$ = average coolant temperature for node *i*,

h_c = coolant heat transfer coefficient,

D_{in}, D_{out} = channel tube inner and outer diameters, respectively,

k_m = channel tube (matrix) thermal conductivity.

In the steady-state solver, the coolant temperatures are calculated based on heat addition rate to the coolant. Then, the heat loss at each node is calculated using Equation (4). After that, the mid-wall (half-resistance) matrix (channel tube) temperature is calculated as:

$$T_{m,i} = T_{av,i} - \frac{\Delta Q_i}{\Delta L_i}\left(\frac{1}{h_c \cdot \pi D_{in}} + \frac{1}{2}\frac{\ln\left(\frac{D_{out}}{D_{in}}\right)}{2\pi k_m}\right) \quad \textbf{(5)}$$

Because matrix thermal conductivity is a function of matrix temperature, iterations on the matrix temperature in Equations (4) and (5) are added to the code.

Due to the addition of the heat loss, the heat generated in the fuel will now be greater than the heat removed by the coolant. Since the primary goal of the PDC remains the cycle analysis, the primary meaning of the "reactor power" in the PDC input and reporting remains the amount of heat supplied to the cycle, i.e., net heat added to the coolant. This is what is provided in the input file and will be 500 MW_t for Pascal design. The heat generation in the fuel will be calculated by the code based on the heat addition to the coolant and the heat loss (513 MW for Pascal at design conditions). In transient output, the latter will be referred to as heat generated in reactor (Q_Rx) and heat supplied to the Brayton cycle (Q_Rx_BC).

For the Pin Type (or an electrical heater), the heat loss will represent the heat loss from the reactor shell, if non-zero value is provided for the *HTC* input. For the prismatic type, the heat loss is from the graphite matrix. In all cases, the heat loss will be from "matrix" material, so the Equation (3) is applicable to all reactor types. Equations (4) and (5) are formulated slightly different for the Prismatic Type since the primary heat flow there is from fuel to the matrix. Then, heat is split between the heat addition to the coolant (with thermal resistance between the matrix, tube, and coolant) and heat loss from the matrix. Also, matrix resistance in Equations (4) and (5) is formulated differently from the cylindrical geometry and is documented in Reference [10].

3.4 Fuel Temperatures

For reactor applications, it is important to calculate fuel temperatures accurately, both in steady-state and in transients. The peak fuel temperature is one of the most important safety considerations. In previous simulations (mostly for electrical heaters), a rather simplified single-node treatment of "fuel" temperatures was implemented. Therefore, the fuel temperature modeling was refined in PDC to include multi-node treatment in radial direction (to calculate fuel centerline temperature with temperature-dependent thermal conductivity), include the heat conductance in fuel-cladding gap, and add (UO_2) fuel properties to the PDC material database.

3.4.1 *Multi-Node Treatment*

For more accurate calculations of peak (centerline) fuel temperatures, a number of radial nodes in fuel is simulated. The number of radial nodes in fuel is a new input for PDC. The code uses a uniform radial mesh based on radius, i.e.:

$$\Delta r_i = \frac{r_f}{N}, for\ all\ i = 1, \dots, N \tag{6}$$

where r_f is the fuel radius and *N* is number of radial nodes (both are user inputs).

It will also be assumed that the heat generation in the fuel is radially uniform, with volumetric energy generation $q''' = \frac{q'}{\pi r_f^2}$, and $q' = Q/L$ based on total fuel heat generation.

Since the fuel surface temperature is calculated from the coolant temperatures and temperature rise across the cladding, the temperatures for all fuel nodes can be calculated starting from the outer node. The solution of the heat conduction equation in cylindrical geometry with internal heat generation is:

$$T(r) = T_s + \frac{q'''}{4k}(R^2 - r^2) \tag{7}$$

where,

T_s = surface temperature at radius *R*,

k = fuel thermal conductivity.

In the multi-node fuel treatment, it is assumed that the fuel thermal conductivity is fixed for the entire node (but could change between the nodes).

Equation (7) can be re-arranged using the relationship between volumetric and linear heat fluxes to obtain temperature at the border of each radial node:

$$T_i = T_{i+1} + \frac{q'}{4\pi k r_f^2}(r_{i+1}^2 - r_i^2) \tag{8}$$

Defining r_1=0, r_{N+1}=r_f, and using the definition of thermal resistance for the radial node *i* in the way similar to other PDC treatments $q' = \frac{T_i - T_{i+1}}{res_i}$: the fuel resistance, by node, is:

$$res_i = \frac{r_{i+1}^2 - r_i^2}{4\pi k_i r_f^2} \tag{9}$$

For the inner most node *i*=1, r_1=0:

$$res_1 = \frac{1}{4\pi k_1}\frac{r_2^2}{r_f^2} \quad \textbf{(10)}$$

Note that for a single radial node (N=1 and r_2=r_f), Equations (9) and (10) reduce to: $res = \frac{1}{4\pi k}$, which is the same as was used before for the single-node treatment.

Equations (9) and (10) are implemented in the PDC to calculated fuel temperatures for all radial nodes. Since thermal conductivity in general is temperature-dependent, iterations on fuel properties and temperatures are implemented. The solution was verified by comparing the results for multiple nodes with single node treatment, both with the fixed and changing thermal conductivity (with fixed conductivity, the solutions for single and multiple nodes were identical).

3.4.2 *Gap Conductance*

The fuel temperature calculations were also extended to include thermal resistance of the fuel-cladding gap. The gap treatment is a simple fixed-conductance approach with the gap conductivity, in W/m^2-K, provided in the input file. Because of the gap thermal resistance, a distinction is now made, in steady-state calculations, between cladding inner surface and fuel outer surface temperatures. In transient calculations, the gap thermal resistance is added to the resistance between the fuel outer node and cladding.

3.4.3 *Fuel Properties*

Properties for uranium oxide (UO_2) fuel have been added based on formulations in the open literature [12]. By comparing the values for the UO_2 density, specific heat, and thermal conductivity between Reference [12] with those used for Pascal design, a very good agreement was found if a 5% porosity is assumed. Therefore, the PDC properties database was updated with the UO_2 properties with 5% porosity.

3.5 Inlet and Outlet Sections

As shown in Figure 1, the Pascal reactor design involves relatively long coolant pipes before and after the reactor core, to connect to the inlet and outlet collectors. In order to include the effect of these pipes on the reactor pressure drop, as well as on thermal inertia and delays in transients, the provision of the inlet and outlet sections have been added to the reactor component in PDC.

The inlet/outlet sections represent the *unheated* (out-of-the-core) parts of the reactor channels. As such, there is no heat transfer from the fuel. Also, for the Pascal reactor, the inlet and outlet sections are outside the reactor vessel, so there should be no heat loss to the moderator. For these reasons, it is assumed that there is no heat transfer in the inlet and outlet sections (in steady-state calculations), and the only effect is on pressure drop. In transient, only heat exchange between the coolant and channel tube will be modeled, with adiabatic conditions on the channel tube outer surface. It is assumed for the inlet and outlet sections that the channel tube dimensions (diameters and thickness) remain the same as in the core, as is the channel tube material.

Because the only heat transfer simulated in the inlet and outlet sections, - between the coolant and the tube wall, - is expected to be relatively small, a single-node treatment is applied to each of these sections. Meaning that a single tube wall temperature is calculated and for coolant only inlet and outlet temperatures and pressures are obtained.

The same treatment of inlet and outlet sections will be applied for the Pin and Prismatic types. Meaning, no heat transfer to the fuel or heat loss from the matrix.

The new inputs are introduced to the PDC to define inlet and outlet sections:

- Inlet and outlet section lengths, and
- Inlet and outlet section form loss coefficients.

In addition, for the Channel Type only, there is an option to simulate either empty (no internal structure) channel tubes or to preserve the core geometry. The latter option is added to allow treatment of the situations where, for example, fuel rods are inserted from the top of the reactor and their (unfueled) length extends beyond the reactor core. Again, there will be no heat transfer simulated between these structures and the coolant in the inlet and/or outlet sections, the presence of the structure will only affect the coolant flow area, wetted perimeter, and hydraulic diameter, for calculations of pressure drop and heat transfer to the channel tube wall. This option is defined by the “Inlet and outlet geometry flags” input and is provided for the two sections as “0=core geometry” and “1=empty channel tube”. The geometry option is applied to the entire length of the corresponding inlet or outlet section. It was decided, however, for the Pascal simulation presented below not to preserve the core geometry in either the inlet or outlet section, because the channel part with fuel pins is relatively short in this design. Instead, the effect of the fuel pins is included in the core inlet form loss coefficient. Still, this option is retained in the code for future uses.

The equations for the pressure drop and the heat transfer (between coolant and channel tube) in the inlet and outlet section are the same as for core sections. The only differences are related to how the coefficients for these equations are calculated, including flow area, wetted perimeter, hydraulic diameter, and so on.

Introduction of the inlet/outlet sections and the form loss pressure drop (in Section 3.2.3) make it sufficient to simulate a wide range of pressure losses outside the reactor core. Therefore, previous input of heat transfer section-to-total pressure drop ratio is not needed anymore and is deleted from the reactor modeling.

3.5.1 *Headers Treatment*

The Pascal reactor includes the inlet and outlet headers, from which all channel tubes originate and terminate (see “CO_2 Inlet Pipe” and “CO_2 Outlet Pipe” in Figure 1). These headers will not be modeled specifically in the PDC. The working fluid volume in the inlet and outlet headers will be included in the already existing PDC input for the reactor inlet and outlet volumes (these volumes are only used in transient calculations and have no effect on the steady-state results). The pressure drop in each header (50 kPa in Pascal) will be included as the inlet form loss for each inlet and outlet section.

3.6 Axial Power Profile

A feature to specify axial power profile in the core has been added to the code. In previous calculations (mostly for an electrical heater), a uniform axial distribution of power has been assumed. Now, the new input was introduced to define the relative power in each axial node in the core (there is no heat generation in the inlet and outlet sections). The power fractions are re-normalized in the code to preserve total power.

To demonstrate this new feature, a non-uniform power profile was simulated for the Pascal reactor. At this point, though, the exact power profile is not available for Pascal, so a cosine profile with specified power peaking (1.304) was simulated. The PDC results in terms of the heat flux and fuel and coolant temperatures for this profile are shown in Figure 14. The fuel temperature results include multi-mode treatment discussed in Section 3.4. The results are as expected, with sine shape for the coolant temperature and the peaking in the fuel temperature close to the middle of the core. It was also confirmed that for the heat flux in Figure 14, the peak-to-average ration is equal to 1.304. Also note that in Figure 14, consistent with the PDC internal calculations, the core length is measured from outlet to inlet.

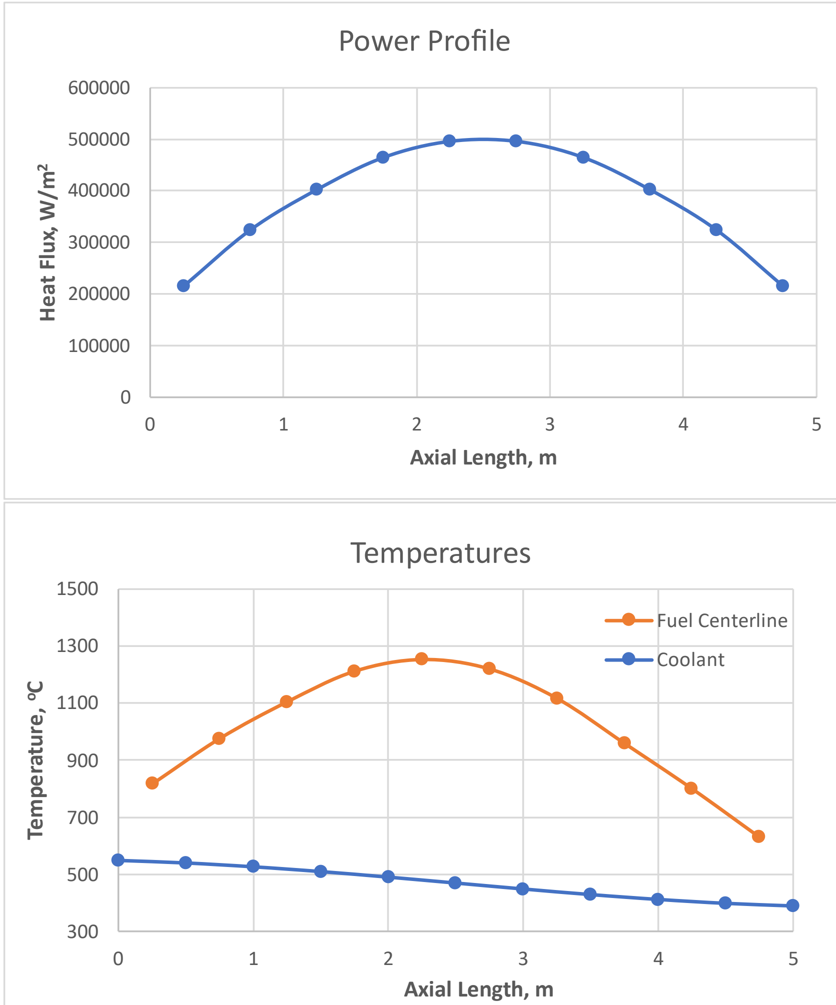


Figure 14. Axial Power Profile and Temperature.

3.7 Peak Channels

Until this point, PDC has been developed exclusively as a system-level code. That is, the performance of each component (heat exchanger, turbine, etc.,) was calculated only to characterize its effect on the entire cycle or plant. For these reasons, an average-channel approach has been implemented for all components. For example, heat transfer in a heat exchanger is calculated for that in a single tube (or channel), under an assumption that all tubes behave identically. This treatment has also been applied so far to the reactor component in order to calculate the effect from the reactor (heat addition and pressure drop) on the cycle. At the same time, the average-channel approach could not be used to characterizing variations within the component, which is often important for a reactor to calculate the peak temperatures.

In order to provide a possibility to calculate peak temperatures, the concept of multiple channels has been introduced to PDC for the reactor component. The concept has been implemented in a rather simplistic way, where the user provides the *peaking factors for each channel for power and flow*. It is then assumed that these peaking factors are fixed, i.e., they would not change in a transient. For generosity, these channels are called "peak channels" in this report, although they do not necessarily exhibit peak (i.e., higher than average) temperatures.

A user provides input of how many additional channels (up to 100 is currently supported) will be simulated (the first channel is always the average channel and is not included in this number). Then, for each of the additional channels, power and flow multiplication factors are provided. This multiplication factors can be >1 , =1, or <1, to the user's choice, but they need to be positive. These factors are the only inputs required for the additional channels. It is assumed that the geometry and materials in all channels are the same, and only power and flow are different.

The calculations still start with an average channel. The code calculates the coolant temperature for all axial nodes in this channel. Then, all other temperatures (fuel, cladding, channel tube) are also calculated for the average channel. These calculations are not different from the average channel approach implemented before.

Once the calculations for the average channel are completed, the calculations for each additional channel proceed. The coolant inlet temperature and pressure are the same for all channels. The heat addition to coolant at each axial node is calculated by multiplying that for the average channel by the power factor for the current channel. Likewise, the coolant flow rate in a channel is obtained using the average value, multiplied by the flow factor. From those, the calculations for this channel proceed the same way as for the average channel. First, coolant temperatures and pressures are obtained. Then, temperatures for other structures are calculated from heat loss and heat transfer between coolant at these structures. No other provisions for the additional channels are needed, except for the pressure equilibration discussed below.

With the flow rates for the peak channels, in general, being different from the average channel, the pressure drop in these channels would be different from that of the average channel (and between the peak channels). In order to preserve the pressure change across the reactor component, the following procedure is applied. First, the pressure drop across the average channel is calculated "as-is", with the user input. That is, the input for parameters like form pressure loss coefficients is applied to the average channel. This pressure drop in the average channel will be the total pressure drop across the reactor component and is not affected by the peak channels. In order to match the pressure drop across all channels, the inlet section form loss coefficient is adjusted for all peak channels. That adjustment is reported to the user in the reactor output file, for each peak channel (no adjustment is made to the average channel). Note that this procedure may, theoretically, result in negative net form loss for a peak channel. That negative form loss factor will not affect the PDC calculations in any way (as the flow rate in peak channels is derived from that in the average channel). So, it is left to the user to monitor for this situation and adjust the average channel form loss input, if desired.

The new peak channel feature was tested for Pascal reactor with several combinations of average and peak channels. An example of one peak channel is discussed below. It was specified that the radial (across the channel tubes) peaking factor for this reactor is 1.25, so the peak channel was simulated with power factor P=1.25. At this point, orificing for Pascal reactor has not been finalized, so for these calculations, a perfect orificing is assumed, where flow factor in the peak channel is equal to the power factor, i.e., F=1.25 as well. With this input, the coolant temperatures should be the same in both the average and peak channels, but the cladding and fuel temperatures should be higher in the peak channel.

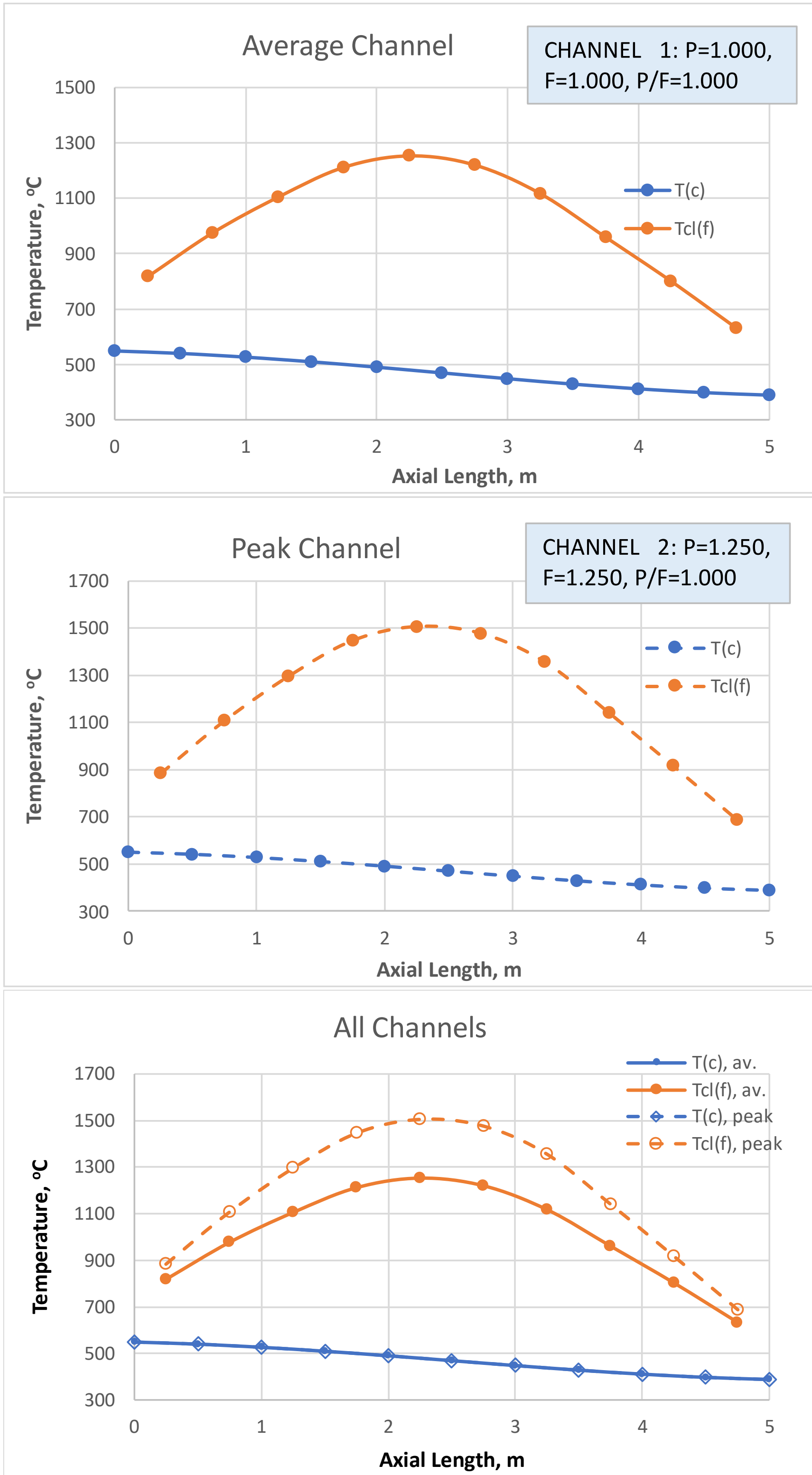


Figure 15. Temperature Profiles with Peak Channels.

The PDC steady-state results with two channels, one default average with P=1.0 and F=1.0, and one peak with P=1.25 and F=1.25, are shown in Figure 15 for the coolant, T(c), and fuel centerline, Tcl(f), temperatures. The first two plots show the results for the average and peak channels, respectively. The last plot compares the results for both channels (on that plot, the coolant temperatures are identical). The results in Figure 15 demonstrates that the new PDC feature works as expected with fuel temperatures being higher in the peak channel. Also, the temperature rise from coolant to fuel centerline, for example at z=2 m mark, is about 1000 °C for the peak channel and about 750 °C for the average channel – a ratio of 1.3, which is close to the power peaking factor of 1.25 (the ΔT ratio is not exactly the same as power ratio, due to temperature dependency of properties like thermal conductivity).

3.8 Reactor Steady State Results

Figure 16 shows reactor temperatures (other than fuel centerline shown in Figure 15) calculated in PDC at steady state, including inner and outer channel tube surfaces, cladding outer surface and fuel outer surface. The results are again shown for the average and peak (P=F=1.25) channels. Because the coolant temperatures are the same for these two channels (with same P/F), the channel tube temperatures are also the same. The results in Figure 16 are shown in the core only. Outside the core, only coolant and channel tube temperatures are calculated. Since there is no heat transfer (from fuel) and no heat loss to the moderator outside the core, both coolant and tube wall temperatures are equal to the core inlet and core outlet temperatures in the inlet and outlet sections, respectively.

Table 3 shows how the total reactor pressure drop is simulated and matched in PDC. The first row in Table 3 are the Pascal design parameters. The second row is the PDC target for each section, where the pressure drop in headers is included as the target for the inlet and outlet sections. Note that the Pascal design specifications is for the peak channel, with 125% flow rate, so the PDC targets (and results) are also for the peak channel (see Section 3.7). The next row represent the form losses input for each section, in order to match the pressure drops. The resulting pressure drops calculated by PDC are shown in the last row, demonstrating good matching of PDC targets for each section, as well as the total reactor pressure drop. Note that while the pressure drop matching in Table 3 is based on the peak channel, the total reactor pressure drop is still the same for all channels, as discussed in Section 3.7.

Finally, Figure 17 shows the PDC results of the steady state cycle calculations with the reactor module completed and integrated into the code and the cycle analysis. Overall, the results are close to those obtained previously, as the heat addition from the reactor remains the same and the reactor pressure drop has been matched as shown in Table 3.

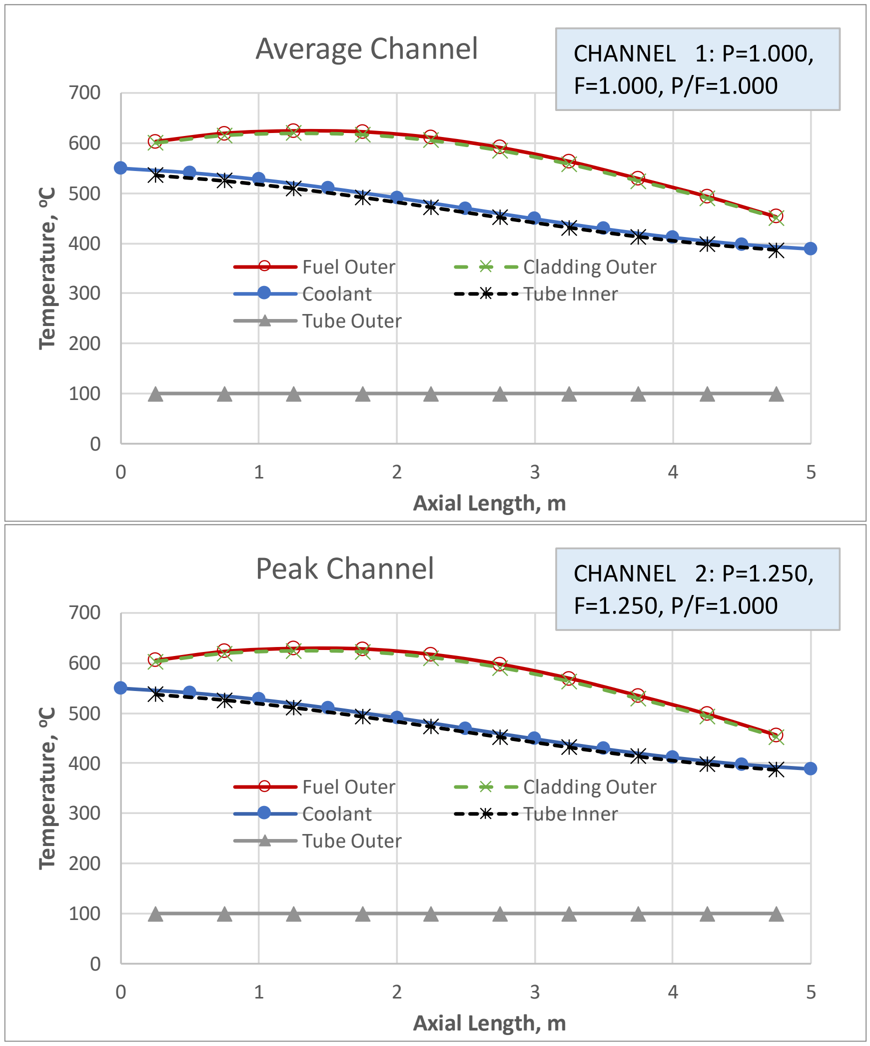


Figure 16. Profiles for Other Calculated Temperatures.

Table 3. Pascal Reactor Total Pressure Drop

Pressure drops, kPa	Inlet Header	Inlet Section	Core	Outlet Section		Outlet Header	Total
Design	50	344	280	198		50	922
PDC - target	-	394	280	248		-	922
Form loss input	-	100.0*	1.55	18.8		-	
PDC – final	**-**	**393.3**	**280.4**	**247.6**		**-**	**921.3**

*Adjusted by the code to 42.8 for the peak channel

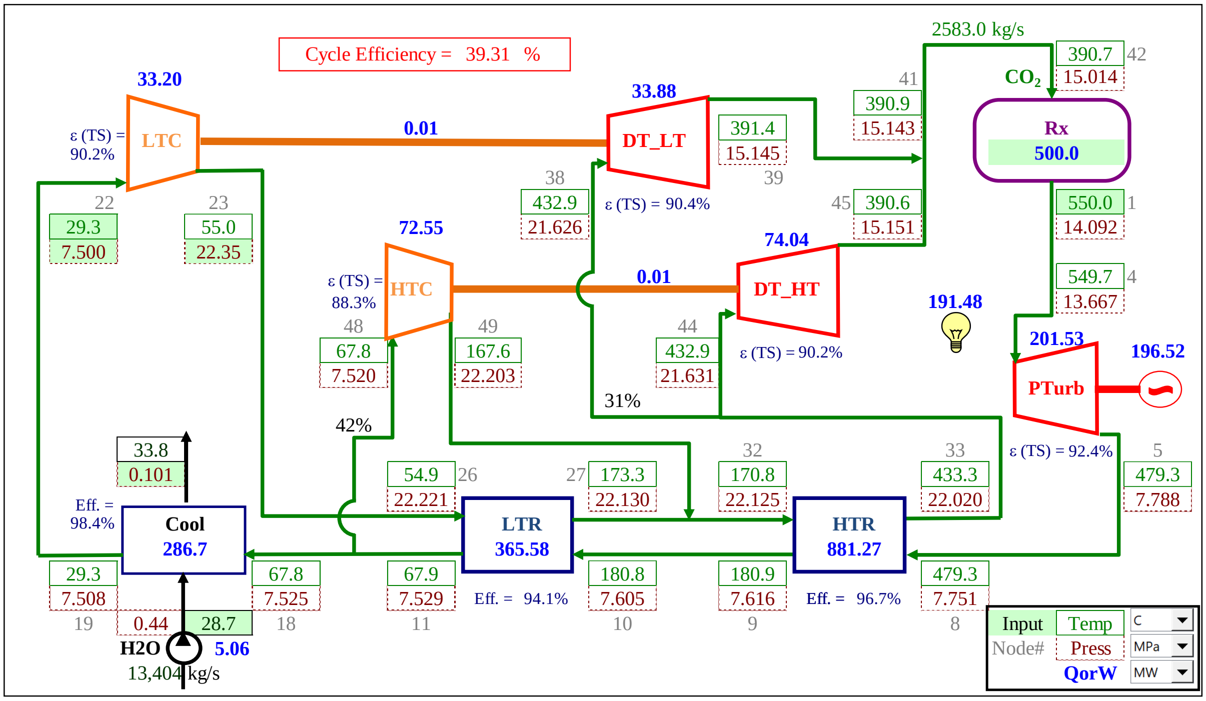


Figure 17. Pascal Cycle Results with Reactor Module Fully Integrated.

3.9 Dynamic Equations

The transient part of the PDC reactor module was updated in a similar fashion to the steady-state modifications described above. These modifications are described below in this section. The reactor component is still based on the previous electrical heater model. Similar to the reporting on the steady-state modifications, the emphasis in this report is done on the implementation of the Channel Type reactor in dynamic part of PDC, with modifications for other types discussed here only when needed.

Also, this report only focuses on the changes that required modifications in the dynamics part of the code. Some of the Pascal features discussed above either do not need code modifications in the transient part or such modifications were trivial and thus are not discussed here. An example of such new features is an axial power profile and non-uniform axial mesh. Since the PDC transient equations are formulated for each axial node, all that required to implement these new features in PDC dynamic equation was to transfer the arrays of axial lengths and power profile from the steady state to dynamics part.

3.9.1 *Reactor Types*

The dynamics equations in PDC for the reactor components were modified to support the three reactor types discussed in Section 3.1 and in Figure 11. Where the equations are different for the different types, such as for heat transfer from matrix or channel tube, code branching was implemented to use the correct equations for the corresponding reactor type. In general, though, the transient code modifications to simulate different reactor types were minimal since the transient part in PDC uses many parameters already calculated in the steady-state part, such as coolant flow area, heat transfer perimeters, etc. In addition, transient calculations in PDC always start from the steady-state conditions, such that the initial temperatures for all reactor structures for dynamic part are already calculated in the steady state part as well.

3.9.2 *Channel Type, Channel Tube and Heat Loss*

The most significant modification for the transient equations in the PDC reactor module were implemented for the treatment of the new structure in the Channel Type of the channel tube. The transient equations for the channel tube are formulated similarly to other tubular structures in the PDC, like fuel cladding and heat exchanger tubes documented in Reference [1]. The most significant difference for the channel tube is that it participates in heat transfer with the coolant on one (inner) side and employs heat loss to the moderator on the other (outer) side. As discussed in Section 3.3, the heat loss to the moderator is implemented in PDC as a heat transfer with constant temperature heat sink. Therefore, the transient equations in PDC for the channel tube is implemented in the following form. Note that the channel tube structure is still simulated as "matrix" in PDC nomenclature, thus a subscript "*m*" is used for the PDC transient equations for this structure:

$$M_{m,i} C_{pm,i} \frac{\partial T_{m,i}}{\partial t} = \Delta x_i \left[\frac{\frac{T_i + T_{i+1}}{2} - T_{m,i}}{res_{wf} + res_m/2} + \frac{T_{HS} - T_{m,i}}{res_m/2 + res_{HL}} \right] \quad \textbf{(11)}$$

where,

$M_{m,i}$ = channel tube mass in node i,

$C_{pm,i}$ = channel tube heat capacity in node i,

t = time,

Δx_i = axial node length,

$T_i,\ T_{i+1}$ = coolant temperatures,

T_{HS} = heat sink temperature (fixed),

$res_{wf} = \frac{1}{HTC_{wf} \cdot \pi \cdot d_i}$ = thermal resistance of the working fluid boundary layer,

HTC_{wf}= working fluid heat transfer coefficient,

$res_m = \frac{ln\frac{d_o}{d_i}}{2\pi k_m}$ = thermal resistance of the tube,

k_m = tube thermal conductivity,

$res_{HL} = \frac{1}{p_{HL} h_{HL}}$ = thermal resistance for heat loss,

p_{HL}, h_{HL} = perimeter and heat transfer coefficient for heat loss (user inputs).

The dynamic equations for the coolant (working fluid) are modified to include new heat transfer from coolant to the channel tube, in addition to already modeled heat transfer to the cladding and convective heat transfer from previous node. The addition to the coolant equation is similar to the first term in the right-hand side of Equation (11).

Equation (11) is applied to both Channel and Pin Types. For the Prismatic Type, the existing equation for matrix, which included heat transfer to coolant and fuel [10], has been modified to include the heat loss from the matrix. This modification is the same as the second term in the right-hand side of Equation (11).

3.9.3 *Fuel Temperature*

The fuel equations have been formulated previously in PDC for both the electrical heaters and prismatic reactor type. The equations are similar to other cylindrical structures, except they include an external heat source, either from electricity or from nuclear reaction. In this work, the only significant change for the fuel equations in transients is the multi-node treatment in radial direction. For the internal node (i.e., not the first or last node), the fuel temperature equation is formulated to include heat transfer to the neighboring nodes:

$$M_{f,i,j} C_{pf} \frac{\partial T_{f,i,j}}{\partial t} = \Delta x_i \left[\frac{T_{f,i,j-1} - T_{f,i,j}}{res_{f,j-1}/2 + res_{f,j}/2} + \frac{T_{f,i,j+1} - T_{f,i,j}}{res_{f,j+1}/2 + res_{f,j}/2} \right] + \Delta Q_{i,j} \quad \textbf{(12)}$$

where the meaning of the variables is similar to those in Equation (11) and j refers to radial node.

The heat addition term $\Delta Q_{i,j}$ in Equation (12) is calculated from the axial power profile distribution and the cross-sectional area fraction of node j in the total fuel crocs-section, under an assumption that heat generation in fuel is radially uniform. The heat addition is treated as external heat source in the PDC dynamic equations and is assumed to be fixed during a time step.

For the first (inner-most) radial node, the formulation is similar to Equation (12), except the first term in the right-hand side is omitted.

For the last node, the heat transfer from the cladding ("tube" in PDC nomenclature) through the fuel-cladding gap replaces heat transfer to the next node:

$$M_{f,i,N} C_{pf} \frac{\partial T_{f,i,N}}{\partial t} = \Delta x_i \left[\frac{T_{f,i,N-1} - T_{f,i,N}}{\frac{res_{f,N-1}}{2} + \frac{res_{f,N}}{2}} + \frac{T_{t,i} - T_{f,i,N}}{\frac{res_t}{2} + res_{gap} + \frac{res_{f,N}}{2}} \right] + \Delta Q_{i,N} \quad \textbf{(13)}$$

where res_{gap} = gap thermal resistance (user input).

All equations are formulated in the PDC dynamic part to calculate structure mid-wall temperatures. The surface temperatures are not calculated because, in general, heat fluxes on surfaces are not conserved in transients. For the same reasons, the fuel centerline temperature could not be obtained in the transient calculations. Instead, the mid-wall temperature of the fuel first radial node is used to represent the centerline temperature.

3.9.4 *Form Loss*

The coolant flow (momentum) equations have been modified to include the form losses. This modification is similar to that provided in Equation (1) for steady state where the form loss is added to the friction loss. Similar to the steady-state treatment, the form loss in the core is distributed among all axial nodes.

3.9.5 *Inlet and Outlet Sections*

Inlet and outlet sections were added to the dynamic equations as the first and last node in axial discretization along the length. The coolant dynamic equations are formulated for the inlet and outlet sections in exactly the same way as for any other core node. The special treatment for other structures in the inlet and outlet sections include:

- There is no fuel – so all coefficients for the "fuel" temperature equations are set to zero and fuel calculations are bypassed.
- There is no heat addition (generation) in these sections. This is taken care of automatically since there are no fuel temperature equations.
- For Empty channels option (see Section 3.5), there is no cladding. So, all coefficients for cladding (tube) temperature are set to zero in this case.
- There is no heat loss from channel tube (matrix) to the constant temperature heat sink moderator. So, the coefficients for this heat loss in Equation (11) are set to zero.

3.9.6 *Peak Power and Flow Channels*

The following modifications were introduced in the PDC dynamic equations to enable treatment of the peak channels:

- An extra dimension was added to arrays for all temperature variables, including fuel, cladding, coolant, and channel tube (matrix). Similarly, an extra dimension was added to all coefficients in dynamic equations.
- For the average channel, no modification in the equations was needed.
- For the additional (peak) channels, the following provisions are implemented:
    - No equations are needed for the coolant flow rate. The flow rate is obtained from the solution of flow equation for the average channel and multiplied by the flow factor for each peak channel.
    - Equations for all temperatures are formulated and solved in exactly the same way as for the main channel.

- The fuel equations are formulated and solved in the same way as for the average channel. The only difference is that the heat generation term in each axial and radial node is obtained with corresponding power peaking factor for that channel.
- The equations for coolant density are not solved. Instead, a non-compressible treatment is applied where coolant pressures at each axial node are calculated using flow rate, inlet pressure and pressure drops. These calculations are done at the beginning of each step and the peak channel pressures are assumed to be fixed during the time step (for the property calculations only, as the momentum equation is not solved for the peak channels).
- The correction to the inlet orifice coefficient calculated at steady state for each peak channel is fixed for the duration of a transient.

3.10 User Interface Update

To accommodate all additional input and now supported three reactor types, the PDC input file for the reactor component has been modified as shown in Figure 12 of Section 3.1. Accordingly, the reactor input data form of the PDC Graphical User Interface has been updated to accommodate the recent changes in the input file. The form now provides the choice of the reactor type and asks for input specific to the selected type.

Figure 18 shows the new PDC GUI form for the reactor input. The form has a general input section, which is common for all reactor types, including power, outlet temperature, axial mesh, and peak P/F channels. Then, a reactor type is selected and the input specific to that reactor type is requested. The example in Figure 18 is shown for the Channel Type of the Pascal reactor. Figure 19 shows the section of the reactor input form for the Pin Type, while Figure 20 shows the same section for the Prismatic Type.

As for all other PDC GUI forms, the reactor input form not only used to provide the input to the reactor component, but also serves as a user guide to provide explanation on each input with tips and figures.

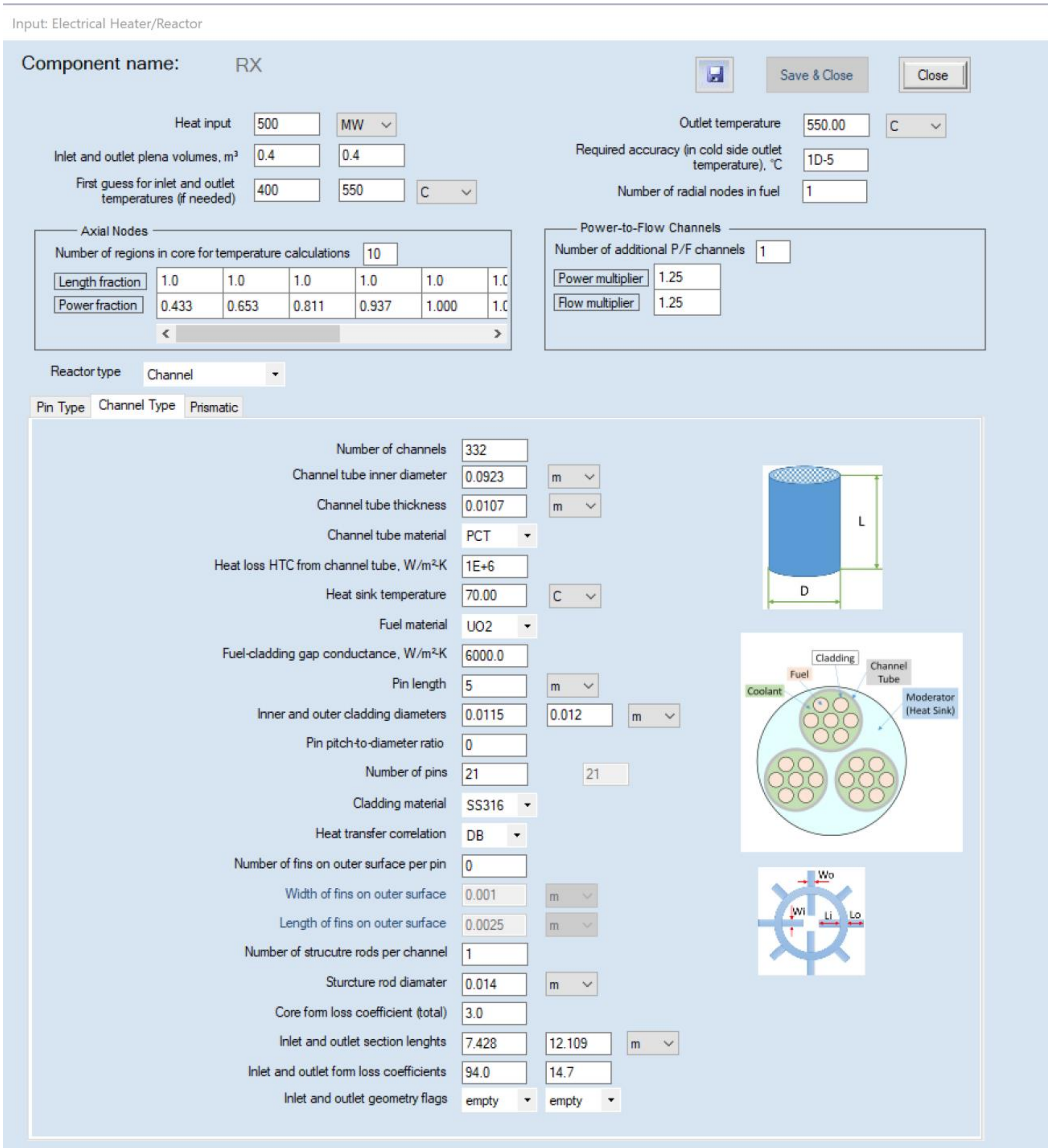


Figure 18. PDC GUI Form for Reactor with Channel Type.

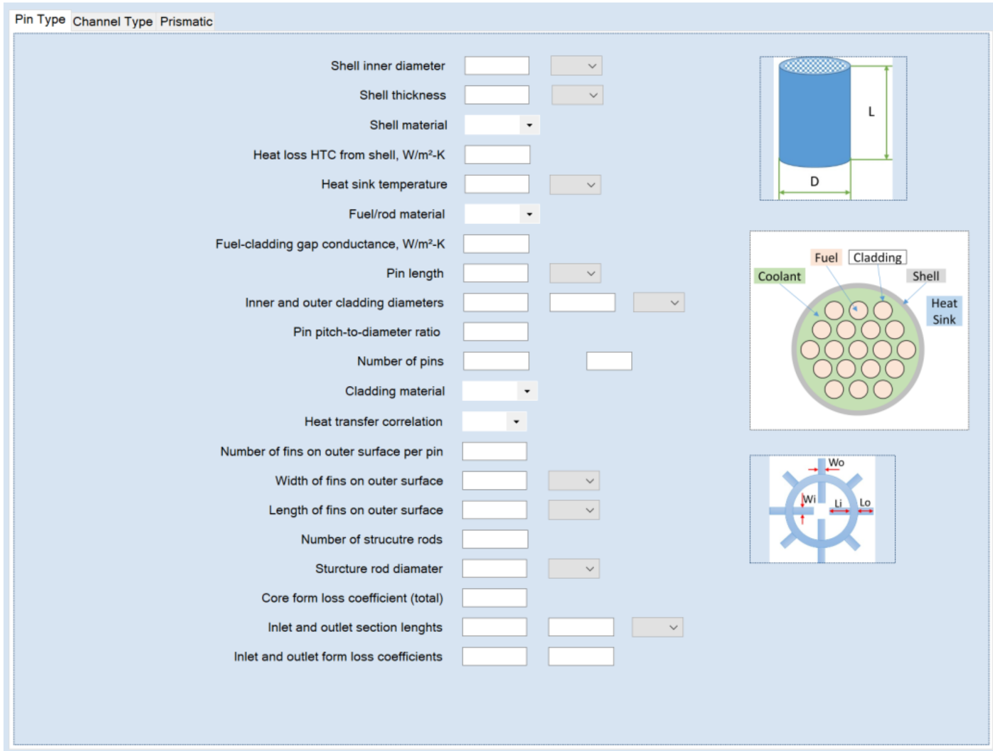


Figure 19. PDC GUI Form Section for Pin Type Reactor.

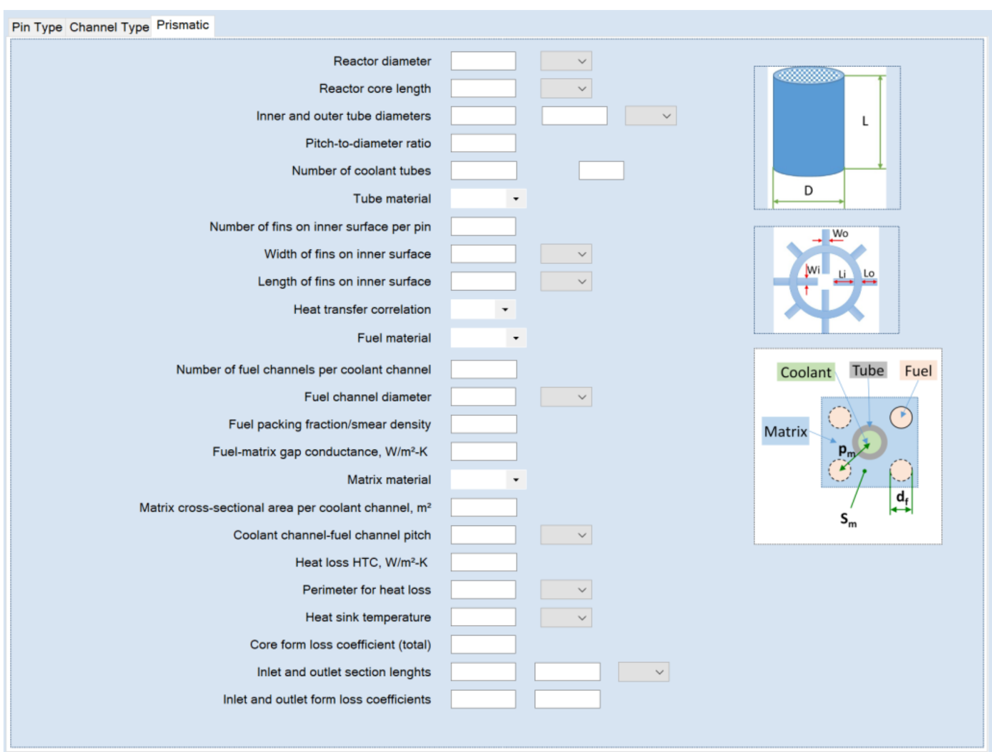


Figure 20. PDC GUI Form Section for Prismatic Type Reactor.

4 Shutdown Heat Removal System

For the direct sCO_2-cooled Pascal reactor, a shutdown heat removal system (SHRS) employs a dedicated sCO_2 loop in parallel to the main power cycle. The SHRS loop (Figure 21) consists of the SHR heat exchanger with water cooling and a sCO_2 pump. The loop includes two isolation valves which are open only when the system operation is needed. In addition, there are two isolation valves on the main cycle, which are normally fully open and are closed only when SHRS is activated. With the main cycle isolation valves closed and the SHRS valves open, the SHRS pump would circulate the CO_2 flow through the reactor and to the cooler, where the reactor decay heat will be removed by water.

Since operation and performance of the SHRS is important for the reactor safety, it was necessary to model this system in PDC to simulate the reactor operation in the decay heat removal mode, as well as during the transition from the normal operation to the decay heat removal.

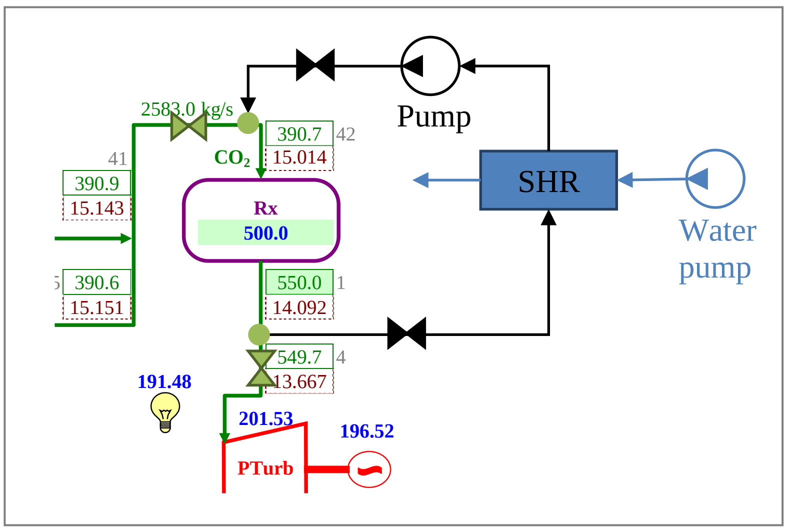


Figure 21. Schematics of Pascal Shutdown Heat Removal System.

4.1 SHRS Loop in PDC Model

The addition of the Pascal SHRS to the PDC model did not require any code modification since the code was already able to simulate the flow branching, valves, a water cooler, and a compressor (pump). The updated PDC model with the SHRS included is shown in Figure 22 and includes (see top-right corner for SHRS branch to the right of the reactor):

- Additional pipes,
- Flow split, SHRsp, and flow merge, SHRmx, branching,
- A shutdown heat removal heat exchanger (SHRHX) with water cooling and water pump,

- SHRS pump (SPump),
- Isolation valves, SHRv at inlet and SHRvo at outlet.

In addition, main loop isolation valves, INSIv and INSOv, were added to the main cycle at the reactor inlet and outlet. Since these valves are fully open at the design conditions (and SHRS valves are fully closed), these model modifications did not affect the steady-state results for the main cycle and the reactor.

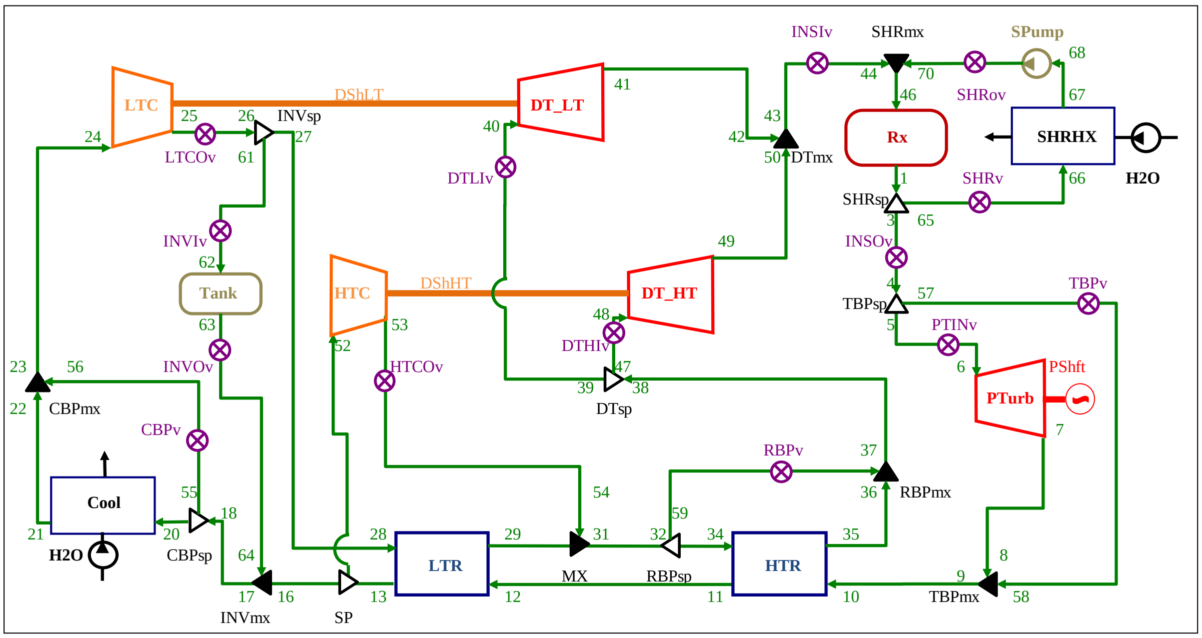


Figure 22. PDC Model of Pascal with Shutdown Heat Removal System.

Although inclusion of the SHRS loop in the PDC model was not an issue, there were still some specific aspects of simulation of the SHRS in PDC, both at design conditions and in transient, that required special attention and were addressed during the work described further in this chapter.

4.1.1 *SHRS at Full Power*

In Pascal design, the SHRS is fully isolated at the full-power (design) conditions, with no flow and no heat removal in the SHRS. Although PDC has provisions of simulating pipe branches without any flow at steady state or in transient (for example, the fully closed turbine bypass line with TBPv valve in Figure 22) that special treatment is only applicable to pipes and not to components. For example, in PDC a heat exchanger component always requires the heat exchange to occur in order to solve the steady state equations. Similarly, the compressor (pump) subroutines required a non-zero flow rate to properly solve their equations.

In order to avoid these limitations, it was decided to simulate Pascal SHRS with some, but very small, flow even at the full-power design conditions. Further analysis have shown that a stable solution can be obtained with 0.01% flow (of the reactor flow) through the SHRS. At the same time, this flow rate is proven to be small enough to not affect the main cycle results in any noticeable way. Therefore, in the PDC simulation of the Pascal SHRS with PDC, the normal operation is simulated with 0.01% flow. Consequently, the flow through the main cycle is 99.99% of the reactor flow.

4.1.2 *SHR Heat Exchanger (Cooler)*

At this point, the SHRS HX has not been designed for Pascal reactor. Therefore, the HX design has been carried out with PDC. This design procedure is discussed in Section 4.2 along with some considerations (limitations) of heat exchanger operation at low flow rate.

4.1.3 *SHR Pump*

For the purpose of the SHRS simulation, it is only required to know the CO_2 flow through the system when it is activated. The exact performance of the SHRS circulation pump is not needed as long as it can still provide the required flow rate. For these reasons and because the Pascal SHRS pump has not been designed yet, the default PDC treatment of the CO_2 compressors with the design and full performance maps is not required for the purposes of the SHRS simulation. Rather, a simplistic treatment of general pump is sufficient for this simulation.

A general pump option has already existed in PDC where the user provides the input for the pump efficiency. The pump inlet temperature is also provided in the input and is also used for the SHRS cooler outlet temperature and design target. For this general pump, the design subroutine is bypassed, and outlet temperature is calculated from the given efficiency, with the inlet and outlet pressures known from the cycle conditions. However, the general pump option in PDC was only developed for the steady-state model and thus needed to be extended for transient simulation. That model development as well as other considerations for the SHRS pump are discussed in Section 4.3.

4.1.4 *Turbomachinery Shaft*

In PDC, each turbomachinery component, i.e., turbine or compressor (pump) is required to be assigned to a turbomachinery shaft. For the steady state solver, the shaft input is required to specify the turbine or compressor rotational speed. Even though the speed input is not needed for the general pump used for the SHRS simulation, the code still checks the input files for completeness and generates an error message if a turbomachinery component exists without a connected shaft. In order to avoid these errors, a simple turbomachinery shaft with the SHRS pump and a motor (a required component for the shaft input) was created and included in the PDC model. However, no input for this shaft will actually be used in the calculations and thus this shaft will have no effect on the calculation results.

4.2 SHRHX (Cooler) Design

Because the design of the SHRS components has not been completed for Pascal yet, this design has been carried out in PDC as described below. However, it needs to be pointed out that the cooler design presented in this report was done only to the extent sufficient to enable simulation of the SHRS operation in PDC. A comprehensive design of the Pascal components was beyond the scope of the work described in this report. In particular, the designs obtained here should be viewed as for demonstration purposes only and is not necessarily representative of the final HX design.

The Pascal SHRS is intended to remove 5% of the total reactor power (or 25 MW). It is also assumed that in the design mode, the SHRS will maintain the same reactor inlet and outlet temperatures, 390 °C and 550 °C, as at the full-power state. With 5% heat removal and the same temperature, it translates to, approximately, 5% flow rate. To simulate the conditions at the SHRS design operation mode, the PDC model in Figure 22 was run with 5% flow split fraction to the SHRS branch. The steady state results of the PDC simulation with this flow are shown in Figure 24 (the conditions on water side of the SHRHX should be ignored for now – they will be calculated later in the HX design).

Note that the conditions in Figure 24 do not exactly simulate the reactor operation at decay heat removal mode. First, the analysis here assumes 95% flow going through the cycle, which would not be the case for the SHRS operation. Second, the flow rate through the reactor in Figure 24 is still 100%, so the reactor pressure drop is not representative of what it would be with 5% flow in decay heat removal mode. However, for the purpose of the SHRHX sizing, the conditions in the SHRS loop and in the HX are sufficiently close to those that would be experienced in the decay heat removal mode.

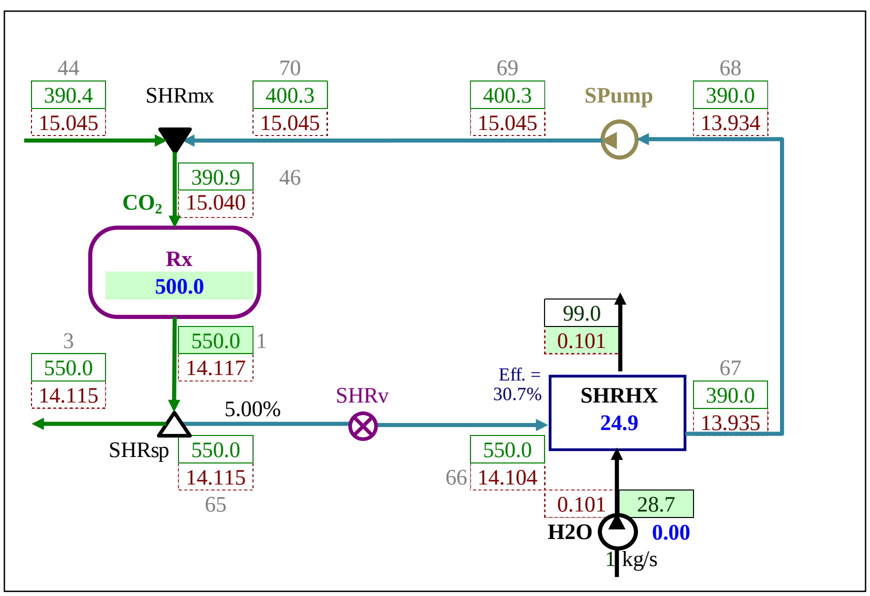


Figure 23. PDC Model of Pascal SHRS with 5% Flow.

For the SHRHX, two design options were considered: PCHE and shell-and-tube heat exchanger. These options are discussed below. For both options, the water inlet temperature of the same value (28.7 °C) as for the main cycle cooler is assumed. The water is assumed to be at atmospheric pressure.

4.2.1 *PCHE Option*

At the first approximation, the SHRHX design was obtained by scaling of the main Pascal cooler with the 1/20 sCO_2 flow factor. However, it was quickly realized that the conditions on the hot (CO_2) side of these heat exchangers are significantly different – for the SHRHX the CO_2 temperatures are between 390 °C and 550 °C, resulting in much larger ΔT between two fluids and much improved heat transfer. Consequently, the main cooler HX designed with 1/20 scaling in size required very little water flow (in PDC calculations) and larger temperature rise on the water side, leading to water boiling. Since treatment of two-phase flow conditions is beyond the PDC intended use, the HX had to be re-designed to avoid water boiling.

Additionally, higher CO_2 temperatures and pressures (compared to the main cooler) would require larger wall thicknesses to satisfy the allowable stress requirements. This was accommodated by increasing the channel pitch-to-diameter ratio as well as the PCHE plate thickness.

Yet another complication with the PCHE design for this application was encountered. When the HX was made smaller to compensate for temperature difference between the hot and cold sides, the HX became too compact even for 5% CO_2 flow rate, resulting in large pressure drop on the CO_2 side. At the end, the PCHE design had to be made very short in order to satisfy the requirements for the pressure drop and heat transfer while avoiding water boiling. The final PCHE design for the SHRHX application is a single unit with width and height both equal to 0.4 m while the (active core) length is 0.12 m. The SHRS performance with this design at 5% flow is shown in Figure 24. The performance of this heat exchanger at 0.01% flow will be evaluated later.

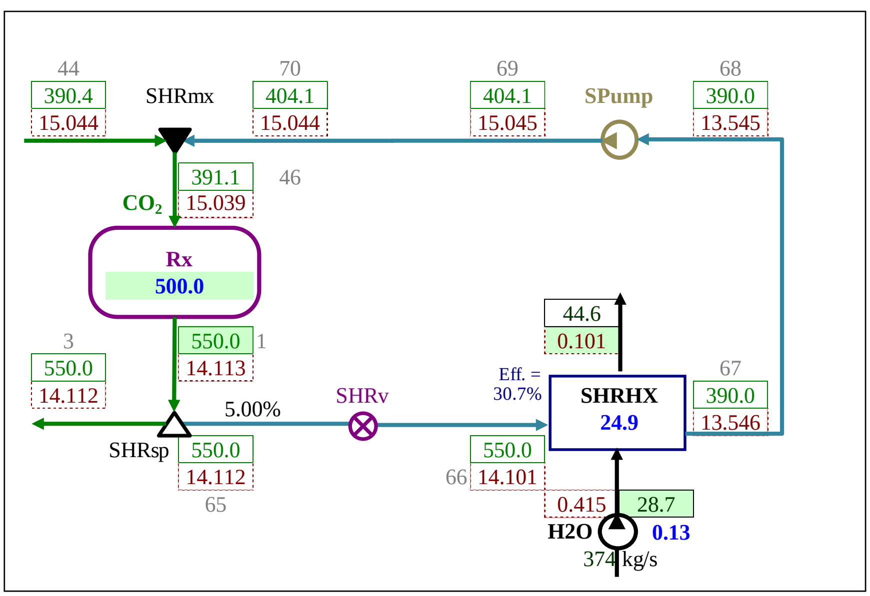


Figure 24. PDC Model of Pascal SHRS with 5% Flow and PCHE SHRHX.

4.2.2 *Shell-and-Tube HX Option*

The analysis in previous section with PCHE design showed that because thermal performance is not an issue with very larger ΔT between CO_2 and water sides, compact HX design may not be required for this heat exchanger. Therefore, the shell-and-tube (S&T) HX might also work for this application. The benefits of S&T HX is that the CO_2 flow channels might be made as large as needed, eliminating (or at least reducing significantly) pressure drop concerns.

There are two main options for the S&T HX: CO_2 on tube or shell side. Shell side CO_2 is better for the pressure drop, since the flow area can be as large as needed by increasing the tube spacing (pitch). At the same time, shell side CO_2 would require a (very) thick shell. For this reason, a design with CO_2 on the tube side was selected for the analysis, as a first choice. As a starting point, 1 in. (2.54 mm) tubes were selected. To accommodate high CO_2 pressure, the tube thickness was selected to be about 1/10 of the tube diameter, or 2 mm. The tube pitch-to-diameter ratio was selected to be 1.4. Lastly, the water flow rate was fixed at 374 kg/s, as for the PCHE design in Figure 24. With the fixed heat duty of 25 WM and this water flow rate, the water outlet temperature is also fixed at 44.6 °C, providing a significant margin to water boiling limit. Any of these designed choices could be revisited later if the selected design does not work.

With the tube dimensions fixed (except for the length), the main design trade-off for the S&T HX is the relationship between the tube length and the shell diameter (or number of tubes). Figure 25 shows the PDC results for the other calculated parameters (tube length, HX volume, and CO_2 side pressure drop) as a function of shell diameter. The trends in Figure 25 demonstrate that there is no clear optimal design in this range. Therefore, a HX design is selected based on the CO_2 side pressure drop considerations. Since the reactor pressure drop at the full flow is 922 kPa, it is estimated that at 5% flow, the reactor pressure drop would be $922kPa*0.05^2=2.3$ kPa. It is reasonable to select a heat exchanger design that would provide approximately the same pressure drop. For the results in Figure 25, for D=1 m and L=1.139 m, CO_2 side pressure drop is 1.8 kPa, which is close to that of the reactor estimate. Therefore, that design has been selected for further calculations.

The SHRS conditions with selected S&T cooler HX are shown in Figure 26. Overall, these results are close to those for PCHE in Figure 24, except for the CO_2 side pressure drop in cooler, which is significantly lower for the S&T design. For that reason, and since the S&T HX design is more conventional (tube length of 1 m as opposed to the PCHE length of 0.12 m), the shell-and-tube design is retained as a reference for further analysis.

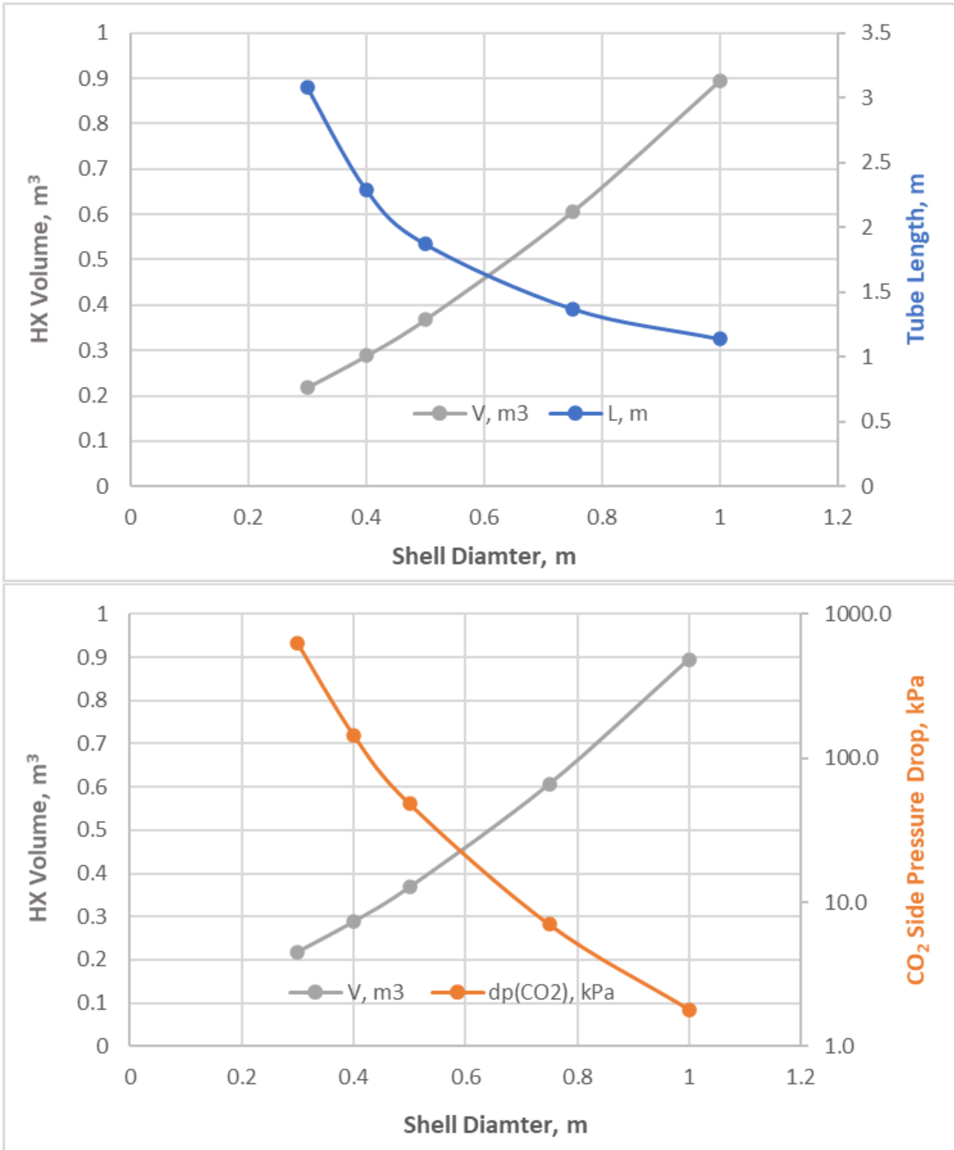


Figure 25. Shell-and-Tube SHRHX Design Options.

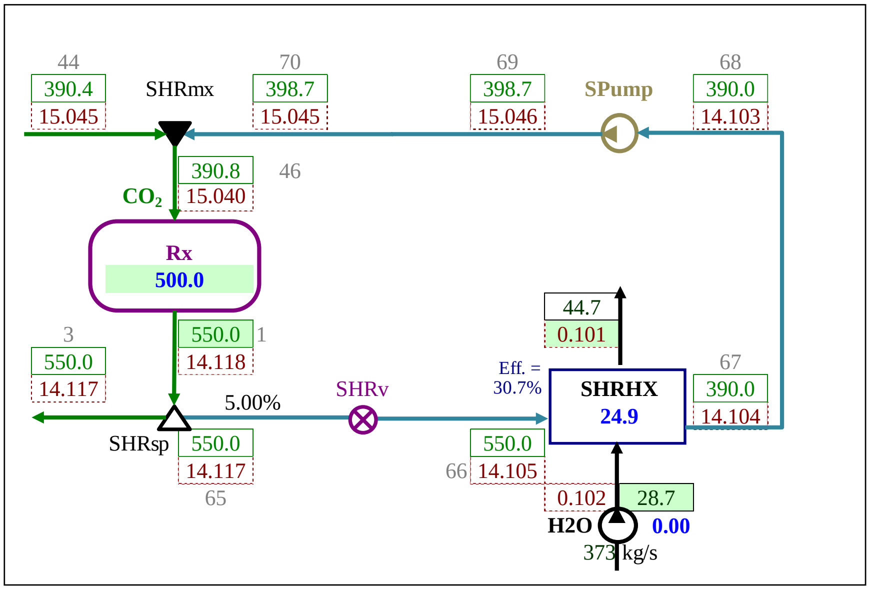


Figure 26. PDC Model of Pascal SHRS with 5% Flow and S&T SHRHX.

4.2.3 *Performance at Nominal (Steady State) Conditions*

The SHRHX performance was checked at the nominal (0.01% flow) conditions. The same water boiling issue was encountered because this HX becomes significantly overdesigned for such lower flow. To compensate for this overdesign, the code selects a very small water flow rate, that results in large temperature change on the water side leading to water boiling. Rather than trying to find a SHRHX design that works both at the design (5% flow) and nominal (0.01% flow), the SHRHX CO_2 outlet temperature was adjusted just enough to avoid boiling. This adjustment is acceptable because there is no requirement to maintain the same CO_2 temperature at low flow conditions (remember that these conditions are only simulated in PDC to approximate no-flow conditions in SHRS at full reactor power). The results of the calculations have shown (Figure 28) that with CO_2-outlet temperature reduced from 390 °C to 140 °C the water boiling can be avoided even with very low flow rates. As shown in Figure 28, the heat removal in SHRHX is 0.13 MW, which is still very small compared to the reactor power. It was also confirmed (Figure 28) that this change (and the addition of the SHRS system) does not affect the conditions in the main cycle, and the cycle efficiency remains at 39.3%.

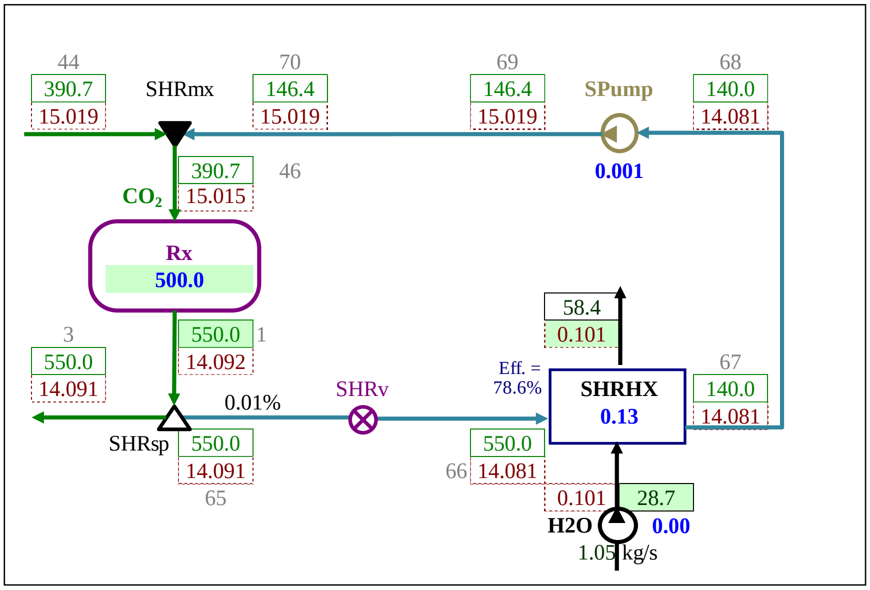


Figure 27. PDC Model of Pascal SHRS with Low Flow and S&T SHRHX.

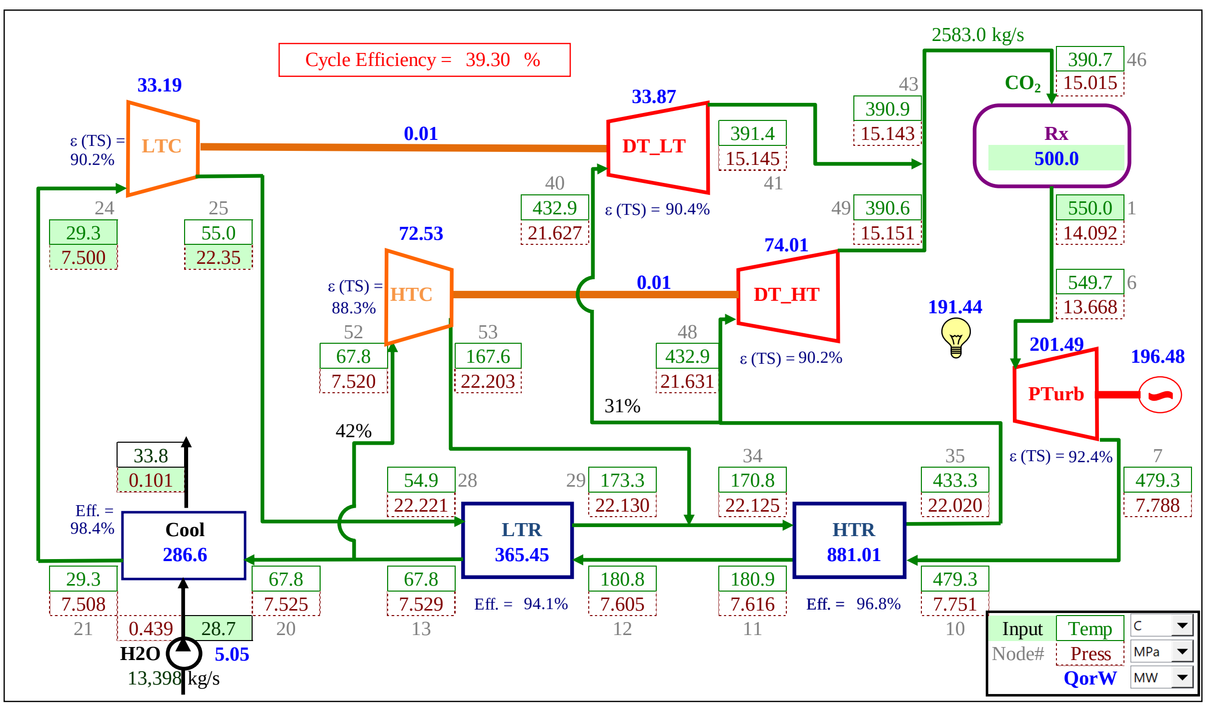


Figure 28. Updated PDC Results for Main Cycle with SHRS Included (not shown on this picture).

4.3 SHRS Pump

As discussed above, a "general type" of compressor/pump is used in this simulation. The goal of this type is to bypass design calculations and only provide enough information to simulate the pump performance. Because the design is skipped, there is no difference between a compressor or a pump, and thus these two terms are used interchangeably in this report for the SHRS pump. Treatment of the general pump is different in the steady-state and dynamic calculations, as described below.

4.3.1 *Steady-State*

In steady state, pump isentropic efficiency is provided by the user in the input file. The isentropic efficiency is the ratio of the enthalpy change in the ideal (isentropic) process to the enthalpy change in the real process (Figure 29). For the general pump, all flow velocities are ignored (assumed to be 0), such that there is no difference between total and static conditions (h=H). Consequently, the definitions of the total-to-static, total-to-total, and static-to-static efficiencies are identical in this case, and thus in the code and in this report the term "efficiency" is used.

The user input efficiency is used in the code to calculate the outlet temperature from the pump following Figure 29. The inlet pressure and temperature are provided from the cycle, as is the outlet pressure. The fluid properties subroutines are used to calculate enthalpy change in an isentropic (constant entropy) compression from inlet to outlet pressure. Then, using the efficiency definition, the outlet enthalpy is calculated. Finally, the outlet enthalpy and pressure are used to obtain the outlet temperature. Once the outlet temperature (enthalpy) is known, the power requirement is calculated from the flow rate and the enthalpy change.

All these calculations and general pump treatment have been implemented in PDC in the past, and thus no code changes were needed to simulate the SHRS pump at steady state.

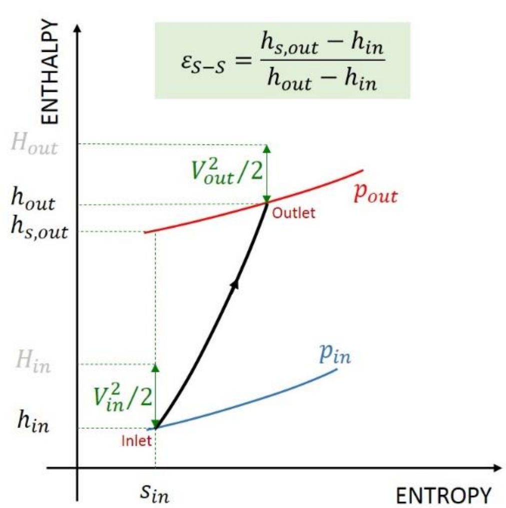


Figure 29. Pump Isentropic Efficiency.

4.3.2 *Dynamics*

In all previous calculations, the general pump option in PDC was limited to the steady state modeling only, and no dynamic treatment for this type existed in the code. Therefore, the code had to be extended to include the general pump option in transient calculations.

The main reason why a general pump was not previously included in the dynamic calculations of PDC is because the dynamic treatment of a compressor (pump) in PDC is based on the concept of the performance maps [1]. These maps are calculated prior to a transient using the compressor performance subroutines and the compressor design. Since no design is obtained for the general pump at steady state, the PDC compressor performance subroutines could not be used to generate the performance maps for this compressor/pump type.

For the Pascal SHRS, though, the detailed simulation of the transient response of the pump is not needed. As discussed above, the modeling only needs to be sufficient to calculate the flow rate through the pump in the transient. Based on this, two simplifications in the compressor treatment in the PDC dynamic equations are adopted for the general pump.

First, the change in the pump performance (efficiency) between the design and off-design conditions is ignored. The efficiency is assumed to be fixed at the steady state value provided in the user input. Therefore, the same procedure to obtain the pump outlet temperature described for the steady state above is used in dynamic calculations. For simplicity, the pump outlet temperature is calculated at the beginning of each time step and is assumed to be constant during a time step. The fluid mass in the pump outlet volume (this volume is still an input for the dynamic calculations even for a general pump) is used to damp any sharp changes in the pump-outlet temperature between the time steps to obtain a smooth change in the overall outlet temperature for the cycle calculations.

The second simplification is that the pump performance in terms of the provided flow rate is calculated directly, either as a user input or from the automatic control. To allow some flexibility, two options were implemented in the PDC dynamic equations: either to calculate flow or to calculate the pump pressure raise (pressure head). The choice of which option to use, as well as the choice of manual or automatic control, is provided to the user in the input file. These options and the corresponding treatment for the general pump in PDC dynamic equations are described below. To select which option a user wants to use, two inputs are provided for the general pump dynamic treatment: the first input flag selects between flow and head options, and the second input flag selects between manual and automatic control. The user can use any combination of these two flags.

4.3.2.1 *Flow or Head Control Options*

If the flow option is selected, the code will calculate the mass flow rate through the pump (either from the user input or from the automatic control). Then, the same compressibility equations, as for all other cycle nodes, are applied to the pump inlet and outlet nodes to calculate density change at those nodes. The pump inlet temperature is calculated from the inlet pipe. The pump outlet temperature is calculated from the pump efficiency definition. The inlet and outlet temperatures and densities calculated this way are sufficient to calculate all other properties (pressure and enthalpy) at the pump inlet and outlet for cycle dynamic calculations.

For the pump head option, the code will calculate pressure rise in the pump (again either from the user input or the automatic control). That pressure rise, together with the input pressure is used to calculate the outlet pressure. Again, the outlet enthalpy is calculated from the pump efficiency definition. The outlet pressure and enthalpy are used to calculate other properties (temperature and density) at the pump outlet. The pump outlet pressure is used to calculate the flow rate in the pipe connected to the pump outlet using the same compressibility equations as for the rest of the cycle. The pump flow rate is then assumed to be equal to that in the outlet pipe. And this flow rate, as well as the flow rate in the inlet pipe, are used to calculate density change at the pump inlet node. That density change is converted to the change in pump inlet pressure in the same way as for the other cycle nodes.

Although both the flow and head options for the general pump treatment were implemented in the code, it was found in further calculations that the flow option provides a more stable solution (less oscillations with a larger time step) for the simulation of the Pascal SHRS operation. Therefore, this flow control option will be used in simulations described in Chapter 6 of this report. Still, both options are maintained in the code for future use.

4.3.2.2 *Manual Control*

For manual control, the user directly provides the table of the desired value versus time. For the flow control option, this would be the flow rate through the pump versus time. For the pressure head option, the table is the head versus time. These manual control tables are very similar to the manual control implemented in PDC for the control valves, with the similar input format and treatment in the code. Linear interpolation is used between the table points, and the first and last table values are extended backwards and forwards, respectively, if the table does not cover the entire transient. For both the inputs of the flow and head, the values provided in the table are the normalized values to the steady-state conditions. Note that for the specific case of the SHRS pump simulation for Pascal, the steady-state values are those obtained at the low-flow (0.01%) conditions as described above, and not the SHRS design conditions.

4.3.2.3 *Automatic Control*

Automatic control for the general pump is implemented similarly to other automatic controls for the Brayton cycle. It is based on the PID (proportional, integral, and differential) control logic and a principle of maintaining a target temperature at a specified cycle node. Therefore, the following inputs are created for the pump automatic control in PDC:

- Cycle temperature node index for target temperature,
- Target temperature table, as temperature versus time,
- PID coefficients for the pump control.

In addition, an input is provided for limits on the maximum and minimum pump head or flow (in %), as well as for the maximum head or flow change rate (in %/s).

For the Pascal SHRS simulation, the pump control is set up to maintain the reactor outlet temperature at the design value of 550 °C. Therefore, the cycle node is set to 1 (reactor outlet) and the automatic control table has only one column with 550 °C for target temperature. The PID coefficients were optimized to obtain the desired control response in transients simulated in Chapter 6.

4.3.2.4 *User Interface Update*

The only change to the steady state input file for a compressor was an explicit declaration of general type (see the last input in the example below). In previous calculations, general type was triggered in steady state calculations by providing input =0 for number of compressor stages. The example below shows the beginning of the input for the SPump for Pascal SHRS simulation. The rest of the input file is common for all compressors but is not used (although it is still read) if the compressor type is set to 0 for general compressor/pump.

```
*************************** Compressor Input ************************
Efficiency (for first guess, static-to-static), %
90
Desired pressure ratio
1.1
Min. temperature control flag (0=no, 1=min, 2=inlet) and value, C
2 140
Min. pressure control flag (0=no, 1=min, 2=inlet) and value, MPa
0 7.4D0
Outlet pressure control flag and value, MPa (used only if flag=1)
0 20D0
Accuracy on exit pressure
1.D-8
Inlet nozzle efficiency, %
90
Volume at exit, m3
1.0
Compressor type (0-Pump/general, 1-axial, 2-centrifugal)
0
```

To provide the required input for the newly implemented general pump treatment in PDC, a new input file is now required for the general pump for dynamics calculations, The file should be located in the DY/Input folder and is named "Control_XXXXX_dat.txt", where "XXXXX" corresponds to the pump name in the PDC model (for SPump name used in the Pascal simulation, the file name is "Control_SPump_dat.txt").

An example of the SPump input file for dynamic calculations is shown below. Note that this particular file is set up for the manual pump control (Number of points in the manual control table is >1) and to maintain 100% flow. The pump could be switched to the automatic control by setting the Number of points input for manual control to 1. This flexibility to switch between control modes was used together with the PDC restart capability in the SHRS simulation in Chapter 6.

```
******** Input data for General Pump control ****************
Dynamic control mode: head (0) or flow (1)
1
HFr: Pump rated head (Pa) or rated flow (kg/s) (0=use SS value)
0
----------- Manual control -----------------
Number of points in the manual control table (1 = no manual action)
2
Pump control table (Time; Head/HFr or Flow/HFr,%)
0    3
100  100
----------- Automatic control -----------------
Cycle temperature node index for pump control
1
Number of points in the temperature control table (0 = no control)
1
Temperature control table (Time, s; Temp, C)
0
550.0
Coefficients for pump head or flow control (P I D)
200 0 100
Pump head or flow maximum change (%HFr/s)
37500
Min/max normalized (to HFr) head or flow for pump
60 1240
```

The PDC GUI form has been updated to provide the input listed above for the dynamic calculations. That form opens instead of the compressor maps input if general type is encountered.

5 Cycle Control Extension

Although modification to the PDC Brayton cycle control logic was not originally included into this project, during the work described in the previous chapters as well as the transient simulations presented in the next chapter, it was realized that some modifications to the control implementation in PDC was needed to simulate Pascal transients. The main driver for these modifications was the necessity to simulate multiple controls of the same type. For example, the Pascal sCO_2 cycle (see Figure 22 for example) employs three throttling valves upstream of each of the three turbines. Moreover, roles and operation of these valves is expected to be significantly different as the drive turbine throttling valves are to be used for the drive shaft speed control, while the power turbine throttling is usually used for turbine power control. In previous PDC development, only one control of each type (including, for example, turbine throttling) was simulated and supported. Therefore, modifications to the Brayton cycle control logic were needed to allow flexibility in simulating multiple controls of the same type.

In the code itself, such flexibility is relatively straightforward as the control inputs and actions variables were simply extended to become arrays. Other than that, the control logic in PDC remains the same, with the PID approach. At the same time, it was realized that the existing structure of the Brayton cycle control input file (BCcontrol_dat.txt) does not allow simple implementation of multiple controls. For example, each control includes a table for the controlled parameter, and providing multiple tables at the same location could be problematic. Therefore, the entire structure of the Brayton cycle control input file was revised in this work, as described below. Also, to simplify working with the tables, a concept of the automatic control tables was developed and introduced. That concept is described below as well.

In addition to the input file modification, other changes to the PDC control logic were introduced for Pascal reactor. Those include the concept of the isolation valves and modification to the reactor control input to simulate decay heat operation. These changes are also described below.

5.1 New Brayton Cycle Control Input File Structure

The Brayton cycle control input for dynamic calculations in PDC (BCcontrol_dat.txt) has been modified to include the following sections:

- Turbomachinery shaft input
- Isolation valves (described in Section 5.2)
- Manual valve control
- Automatic control tables
- Automatic control, for the following control mechanisms:
  - Turbine bypass
  - Inventory
  - Turbine inlet (throttling)

- Compressor surge
- Cooler bypass
- Water/air flow control
- Recuperator/HX bypass

Not all the sections of the input file have been modified. For example, the input for the compressor surge control remains the same. Below is provided the description of the changed sections.

5.1.1 *Turbomachinery Shaft Input*

The turbomachinery shaft input section remains largely unchanged from previous implementation. For each shaft (identified by its name), a table of the shaft speed versus time is provided. That shaft speed is either the given speed for synchronously connected shafts or the target shaft speed for asynchronous shafts or shafts disconnected from the grid (shaft operating mode is provided in the shaft steady state input file). The shaft speed is specified in percent of the nominal (steady state) shaft speed.

The input file structure for the shafts is flexible in that as many shaft speed tables can be provided as needed. However, the code will check if all shafts speed are defined and will generate an error message if input is missing.

5.1.2 *Isolation Valves*

A new section on isolation valves was added to the Brayton cycle input file. This new feature is described in Section 5.2 below.

5.1.3 *Manual Valve Control*

Previously, a manual control was grouped together with the control type. For example, the manual control for the turbine bypass valve was specified in the turbine bypass control section. To provide more flexibility in defining the manual control action, all manual control tables are now moved to the Manual Valve Control section. For each valve, for which manual action is desired, a valve is identified by its name and a table of valve position (in % open) versus time is provided.

This section is flexible too in that as many manual control tables as needed can be specified. The manual control in PDC supersedes any automatic control. If a manual control table (with more than two entries) is provided for a valve, then this table is used in the code for the valve potion in transient. Any automatic control action will be ignored. Therefore, in order to trigger an automatic control for a valve, either manual control table should not be specified for this valve, or the manual control table should have only one entry (column).

Because the manual control is now independent of the automatic control type, simplifications in the code were possible, where the manual control action is now treated with a single array for valve action and position. (In previous modeling, manual control was calculated for each control type: i.e., turbine bypass, turbine throttling, and so on). This modification also allows to specify manual controls for the valve which do not belong to any of the automatic control type, for example, for a compressor outlet valve.

If neither a manual nor automatic control is defined for a valve, its position (% open) will remain unchanged in a transient (valve position is calculated at steady state based on pressure drop input and/or flow rate).

5.1.4 *Automatic Control Tables*

To simplify implementation of multiple controls (described in the next section), the control tables for automatic control have been moved to a new section of the Brayton cycle control input file. Here, a user can define a table that will be used in automatic control. For example, a table of target compressor inlet temperature will be defined in this section. But the compressor inlet (or cooler water flow control) will simply refer to this table for the target value.

A number of automatic controls can be defined as desired by a user (up to 25 tables are currently supported). For each table, the following input is provided:

- *Table index and description*. The index is used by the code and other inputs to refer to the table and should be unique. The description is only provided for the user and is not used in the code.
- *First row flag option*. There are two options supported in PDC: time (in seconds) and load (in %). For load following analysis such as that presented in Section 6.3, it is more convenient to define control tables as a function of load (grid demand). In some other cases, it may be more suitable to define tables as a function of time.
- *Table itself*, as value versus time or load. The meaning of value of each table is defined by how the table is used. For example, for the compressor-inlet temperature, it would be temperature (in °C).

The code was modified to allow storing an array of control tables. This modification is also similar to that implemented to the manual valve control array described above. In the previous version, tables were referred to by name for a control type (e.g., inventory control table). Since the meaning of a table is now defined by how it is used (and referred to in input), there is no need to store tables by names, but rather in arrays (matrixes).

At least one control table is required, to refer to in the next section input, even if all automatic controls are disabled. There is no requirement, however, that a control table needs to be referred to in other input.

5.1.1 *Automatic Control*

Two major updates have been implemented to the automatic control input for PDC: allow for multiple entries and use references to the control tables. These changes are described below. Also, some changes to the water/air flow control and turbine throttle control were made, as also described here.

5.1.1.1 *Multiple Entries for Control Types*

To facilitate using multiple controls of the same time, some controls in the automatic control section have been extended to allow multiple entries for all inputs. This change is implemented for certain control types where multiple inputs are expected. For example, inventory control is not envisioned to require multiple parallel inventory control systems to remove working fluid from multiple points in the cycle. Similarly, only one plant power regulator through a single turbine bypass control is likely to be needed, so multiple controls for turbine bypass are not needed (at least, at the moment). Therefore, the multiple control options is currently implemented only for the turbine inlet (throttling) control (for possible multiple turbines) and the water/air flow control (for possible multiple coolers).

The implementation of multiple controls of the same type is rather simple. All variables (including inputs and control actions) are converted to arrays. In the input file, first, the user specifies how many controls of that type is to be simulated. Then, for each input, multiple entries are provided according to that number of controls. The control actions in the PDC transient part are independent, i.e., multiple controls of the same time do not share any input or calculations.

5.1.1.2 *Control Tables*

With implementation of the tables for automatic control (Section 5.1.4), it is now sufficient to simply refer to the table index for each automatic control. For example, if a control table was defined for the target compressor inlet temperature (either as a function of time or load), the water flow rate control just need an input for that table to define the target table.

The most benefit from this arrangement is realized with the multiple entries for a control type discussed above. With the table index system, multiple controls of the same type now only requires an array of control table indexes. Note that these indices could be different or the same. For example, in some simulations described in Section 6.3, the turbine throttling valve for the power table uses one control table, while the two drive turbine throttling valves share a common table (for the shaft speed target). The only requirement is that the table with the index defined for the control should be provided in the Automatic Control Tables section of the input.

5.1.1.3 *Air/Water Flow Rate*

In the previous implementation of the Brayton cycle control logic, the air or water flow rate control in the cooler was the secondary control to the cooler bypass to assist in controlling the conditions at the cooler outlet and compressor inlet. This was done for the analysis of sCO_2 cycles, where the working fluid at the cooler outlet/compressor inlet is at the closest approach to the CO_2 critical point. In was found in previous work [3,4,13] that a very accurate control of the CO_2 conditions is needed at this point. Therefore, a combination of the cooler bypass (CBP) control and the water/air flow rate control in the cooler was developed and implemented in PDC. In some sCO_2 cycle designs, however, including that of the Pascal cycle, the cooler bypass control is not utilized. Instead, the conditions close to the critical point are maintained by the cooling fluid (water) flow rate control in the cooler alone. Such situation was still supported in previous implementation but required manual disabling of the CBP control in order to trigger water flow rate control for compressor-inlet conditions.

In order to simplify implementation of the control options for the user input, and at the same time provide more flexibility for the cooler control, the code modifications were introduced to enable treatment of the water/air flow rate control as an independent (from CBP) control. That treatment was partially facilitated by implementation of the automatic control tables described above. Now, the water/air flow control can refer to the table of the target compressor-inlet temperature, again independent of the cooler bypass control. The following options are now supported for the water/air flow control in the coolers:

- Option flag = 0: CBP assist – this is the same logic as before, where the primary control is cooler bypass, and the water/air flow control is only used as a control assists (effectively, just to prevent CBP valve from full open or full close),
- Option flag = 1: target temperature for the water/air control is the first stage impeller inlet of the specified compressor (the impeller inlet temperature is usually the lowest temperature in the sCO_2 cycle and thus has the closest approach to the critical point),
- Option flag = 2: target temperature for the water/air control is the inlet temperature of the specified compressor,
- Option flag = 3: target temperature for the water/air control is the cooler outlet temperature,
- Option flag = 4: target temperature for the water/air control is the temperature of the corresponding node in the cycle.

The index for the corresponding CBP valve, compressor, cooler, or cycle node is provided in the next input.

The other modification implemented for the water/air flow control in the coolers is the support of multiple control mechanisms as discussed in Section 5.1.1.1. For the Pascal design, that feature is used for the independent control of the main cycle cooler and the SHRHX in the shutdown heat removal loop. Independent input is provided for each desired control mechanism. This includes the input for the control options listed above, allowing multiple control mechanisms to be set and operate independently of each other. For the simulations presented in Chapter 6, the main cooler flow control is setup to maintain the low-temperature compressor (LTC) inlet temperature, while the SHRHX control is setup to maintain the SHRHX outlet temperature.

5.1.1.4 *Turbine Inlet Control Options*

With introduction of the multiple controller options for the turbine inlet (throttling) control, the Pascal control for the drive turbines and power turbine inlet valves can be simulated independently of each other. Each valve can now have its own control option and its own target table. In particular, the drive turbine inlet valves can be set up to regulate the turbine speeds, while the power turbine inlet valve could remain inactive (power regulation for that shaft will be done by turbine bypass control).

The following control options are now supported for the turbine inlet control:

- Option flag = 1: target is the pressure drop across the valve, in MPa,
- Option flag = 2: target is the turbine bypass flow fraction (this is to use inlet valve in assistance with turbine bypass),
- Option flag = 3: target is the cycle output, in % - this option is to use turbine throttling instead of turbine bypass for power control,
- Option flag = 4: target is the shaft power (in W) or shaft speed (in %), depending on the shaft connection mode. For this option, the next input is the shaft name to be controlled.

The first three options already existed in the PDC. Only Option 4 was added in this work to allow the drive turbine shaft control with throttling valves. The implementation of this option is similar to other existing options, except the valve action is based on the difference between the current value of the shaft power (or speed) and the target value from the control table. If shaft is synchronously connected to the grid (when the shaft speed is dictated by the grid), then the turbine inlet valve is used to control the shaft power. Otherwise, the shaft speed is allowed to change according to the shaft power balance, and the turbine inlet control is used to control that speed.

5.2 Isolation Valves

Operation of the Pascal shutdown heat removal system, described in Chapter 4 above, requires insulation of the main cycle. As PDC has been developed for the transient analysis of the sCO_2 power cycles, simulation of conditions where the cycle is isolated has not been originally envisioned for the code. Therefore, in order to simulate the Pascal SHRS, several changes to the code were implemented with regards to the branches and loop isolation.

In the previous work, the PDC flow and other equations have been formulated to allow isolation of a single pipe. For example, the turbine bypass line can be open or closed, with either some flow rate in it or no flow rate at all. This applies to both the steady state and transient equation formulation. The isolation feature of a single pipe was tested several times in transients where the turbine bypass line was activated and then closed with a control valve action. Similar situations were simulated in other lines, such as the inventory control inlet and outlet lines, which are normally turned off, either for both lines or at least for one line.

At the same time, the PDC feature to handle no-flow conditions was only developed for the pipes, and not the components. For example, the turbomachinery performance can only be obtained for a turbine or compressor if there is a flow in that component. Operation of the compressor or turbine without a flow was, and still is, beyond the purposes of the PDC analysis. A similar restriction remains in heat exchangers that still require flow rate to solve the transient equations correctly. Therefore, in order to make simulation of the cycle isolation conditions possible, the following changes were implemented in PDC.

First, it is assumed that when a branch (or entire cycle) is isolated, it has no effect on the results in another (active) branch. For example, if the Pascal main cycle is isolated, the performance of the cycle components like turbines and heat exchangers in this branch does not affect in any way the performance of the reactor and the SHRS branch. Therefore, the transient calculations of an isolated branch can be skipped (or bypassed). That applies to all pipes in that isolated branch, but also to all the components in the branch. For this reason, a check was added to each component (heat exchanger or turbomachinery) to verify that there is still a flow rate in either the inlet or the outlet pipe for that component. If there is, then the transient calculations proceed as before. If both the inlet and outlet flows are zero, then all the transient equations for this component will be skipped.

Second, special treatment for flow isolation was included in PDC. Although it was found in previous work that PDC equations could handle valve opening and closing, it was still found in the simulations described in Chapter 6 that the flow equations could become unstable "near isolation", leading to a requirement for a very small time step. "Near isolation" refers in this context to the conditions in the pipes that do not have the valve in them but are connected to the pipes that are isolated. The problem was especially pronounced for the pipes located away from the isolated valves. For example, when both the inlet and outlet valves around the reactor were shut down, the flow rate in the entire loop decreased. However, as the flow approached zero in all pipes, the solution became unstable in some pipes. In the pipes with the isolation valve, the flow was forced to go to zero when the valves were fully closed. However, in connecting pipes, and especially ones located far from isolation, such as around the compressor, the flow was not forced to be zero and remained at very small number (or oscillating around zero). Even though physically such small flow rates did not affect the transient results significantly, mathematically the converged solution could not be obtained even with a very small time step (10^{-7} s or less).

In order to avoid these convergence problems at very low flow rates, the following approach was adopted in PDC. The concept of global isolation valves was introduced. Any valve can be designated as an isolation valve. Once this designation is made, a list of the pipes that this valve isolates is provided in the user input. That list is the only input for the isolation valve. The main idea here is that once the valve is fully closed, the conditions of the fully closed valve and zero flow are applied not only to the pipe on which this valve is located, but at the same time the same condition is applied to all the pipes in the isolation list for that valve. For example, either of the main insolation valves in Figure 22, INSOv or INSVIv, can be designated as the main cycle insolation valve, with the all the pipes in the main cycle (from 2 to 27 in Figure 22) provided in the isolation list for the valve. Then, when this valve is fully closed (as it was done in simulations in Section 6.2 below) flow in all cycle pipes will be forced to stop. Combined with the conditions described above for the component bypass, this setup is sufficient to effectively bypass all cycle calculations once the isolation valve is closed. This approach proved to be sufficiently easy to implement and also proved to provide stable solution for the transient without the need for a very small time step. Note also that designation of a valve as an isolation valve only extend the action of this valve on more than one pipe. The valve can still be controlled either from a manual input or by the automatic control, like any other valve. Also, the action only applies when the valve is fully closed, until then the isolation valve has no effect on pipes other than that on which it is located.

To define the isolation valves and to provide the input for these valves, a section on Isolation Valves was added to the Brayton cycle control input file. This section is added after the Shafts and before the Manual Control section. Any number of valves can be assigned as isolation valves (at least one isolation valve is required, that could be any valve with just one pipe on which it is located – this way there would be no effect from this new feature). A valve name is provided followed by the list of the pipes the valve isolates.

5.3 GUI Update

The user interface for PDC was updated to reflect the changes in the Brayton cycle control input file described above. These changes only apply to the Cycle Control section of the dynamic input. The different new sections of this form are shown in Figures 30 through 34, according to the new input structure described in Section 5.1. Most of these figures only show some examples of the input available from those sections. For instance, the automatic control Figure 34 only shows an example of the input for the turbine inlet control; other controls are also included in the input and the form. The new forms (like all other PDC GUI forms) are interactive and allow the user to modify the input as well as provide guidance for each input.

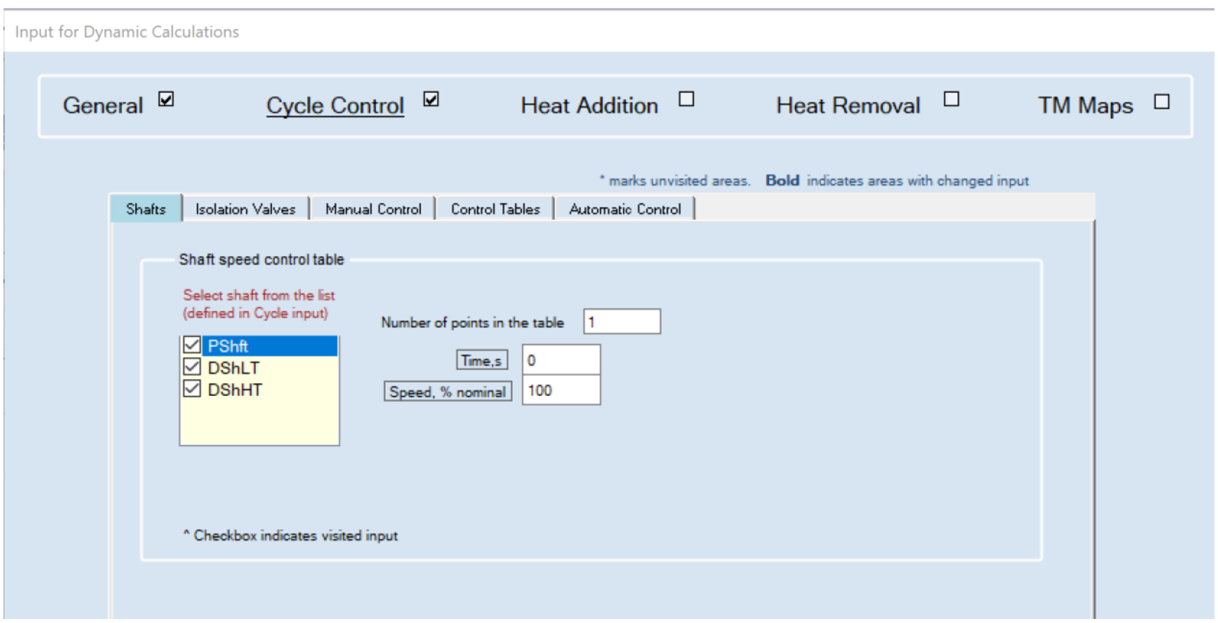


Figure 30. PDC GUI for Cycle Control: Shafts.

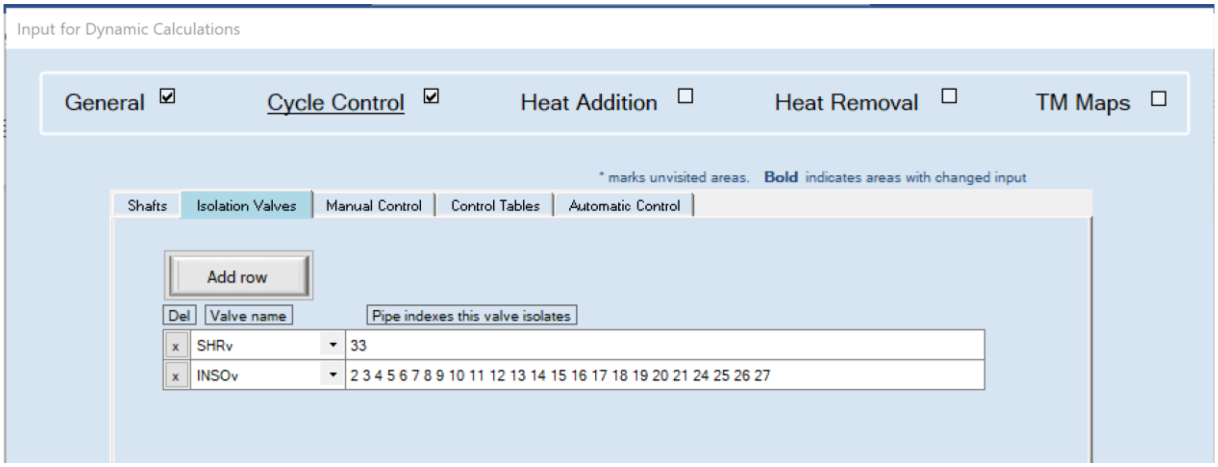


Figure 31. PDC GUI for Cycle Control: Isolation Valves.

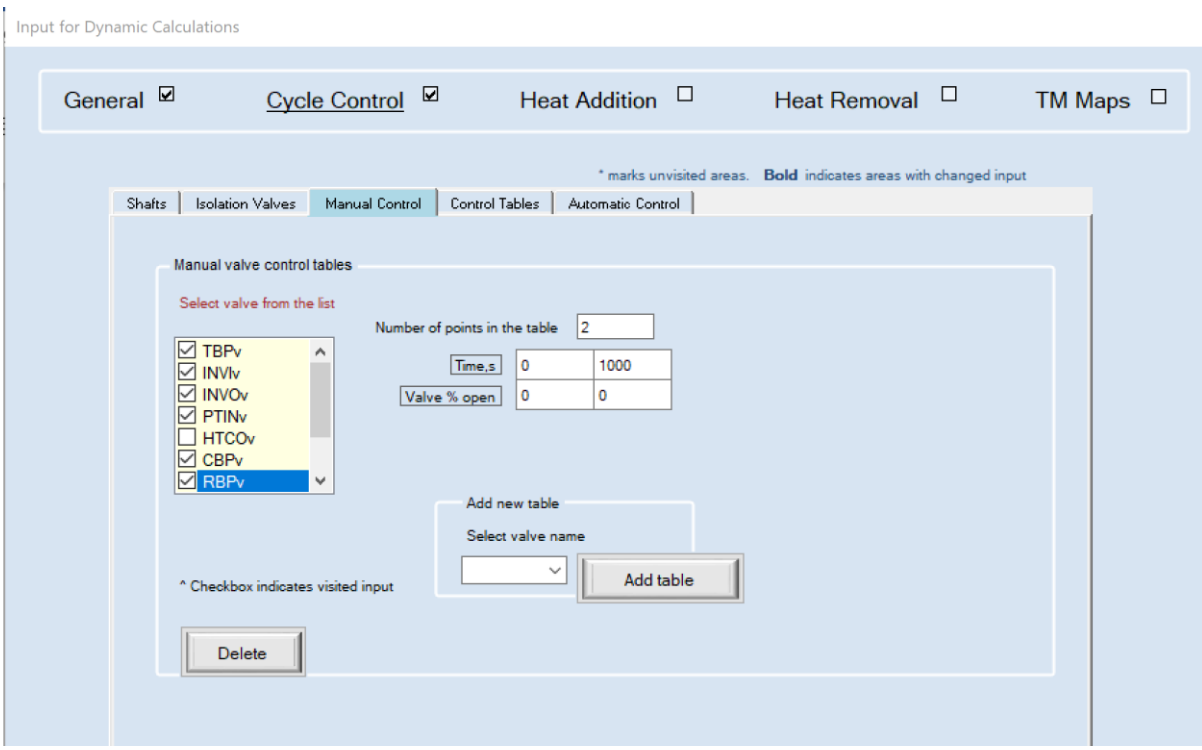


Figure 32. PDC GUI for Cycle Control: Manual Control.

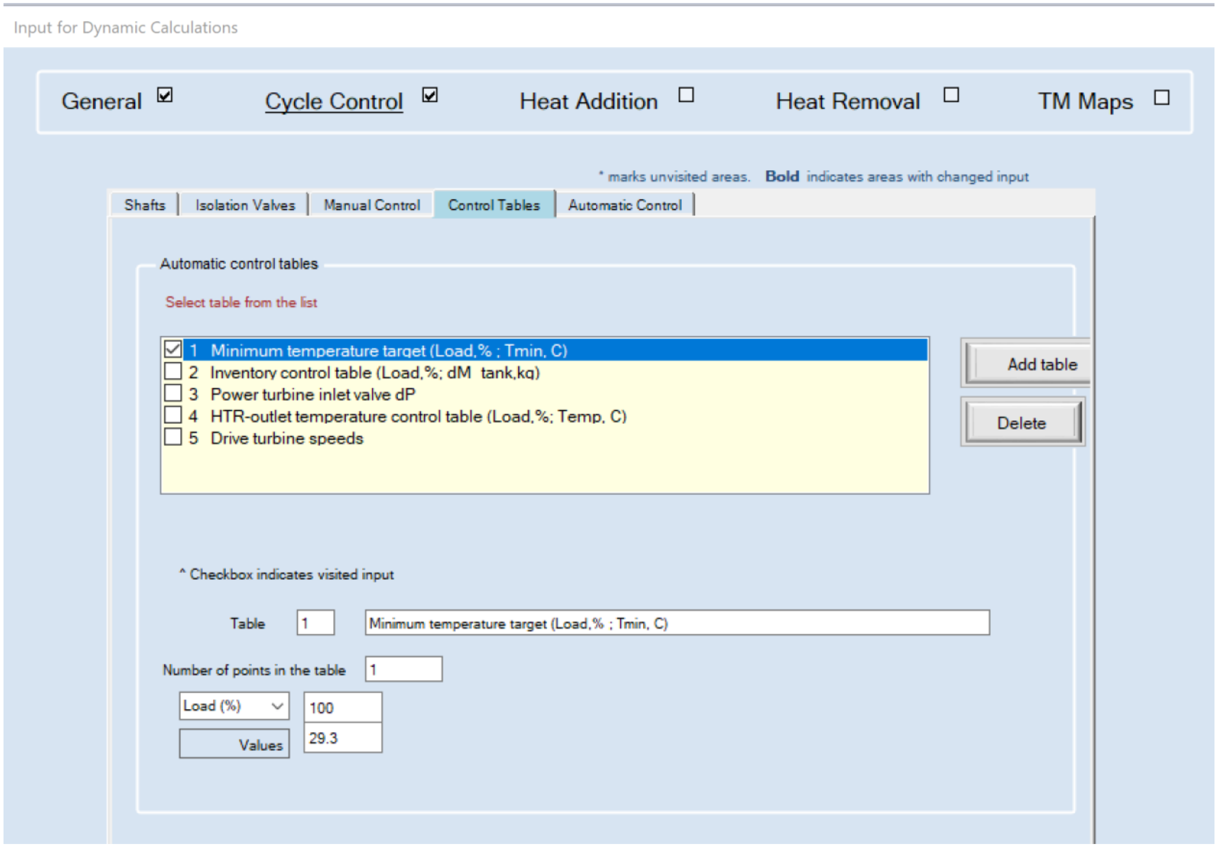


Figure 33. PDC GUI for Cycle Control: Control Tables.

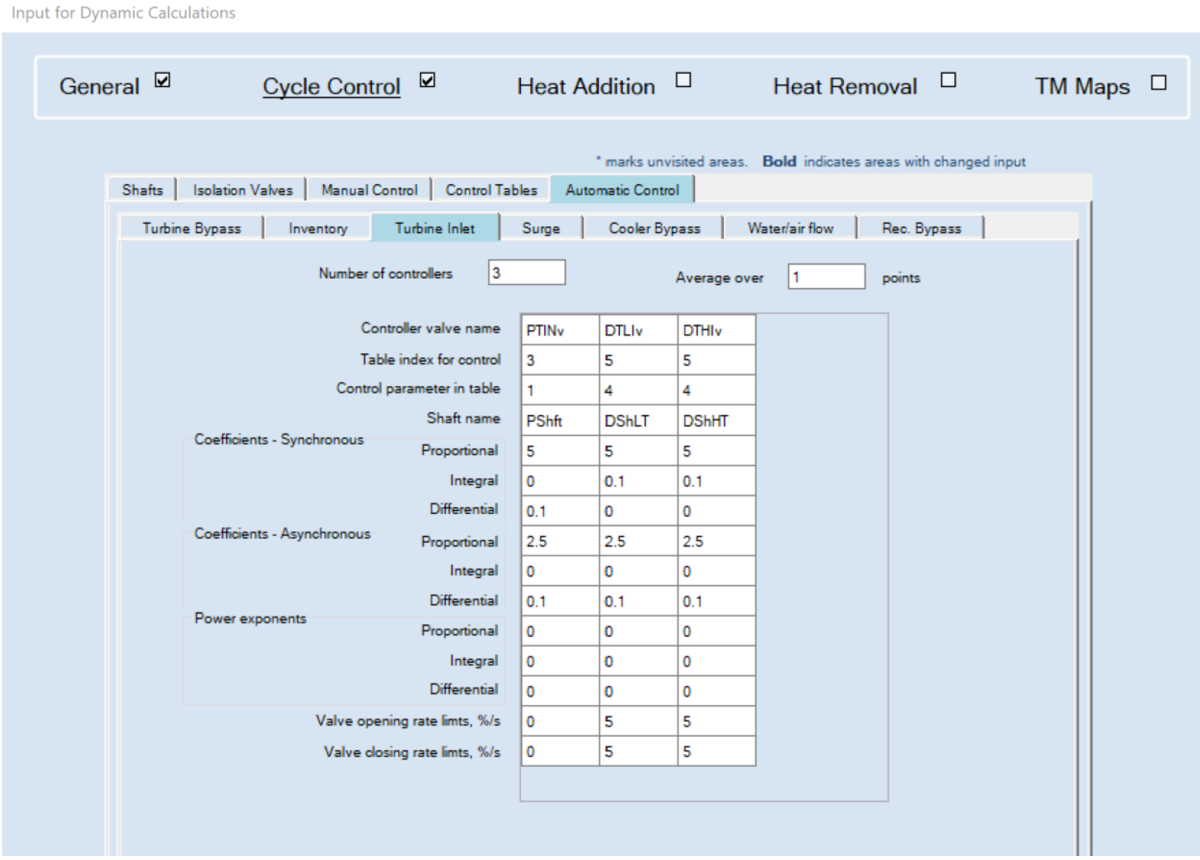


Figure 34. PDC GUI for Cycle Control: Automatic Control.

5.4 Decay Heat Input

There was another change to the PDC input for dynamic calculations. Although it is not directly related to the Brayton cycle control, it is still described in this section for completeness. This change was made to allow more detailed input for the reactor power, for example to more accurate simulation of the decay heat mode. The change is made to the Heat Addition section of the dynamic input.

Previously, the input for the reactor power was provided in the standard PDC control table format (see for example Figure 32). That table format has a horizontal format (two rows, up to 100 columns). This format works fine for relatively short tables like for the cycle control. At the same time, the reactor power history after a scram can be rather complicated and may require all 100 entries (as it did in some simulations described in Chapter 6). For such long tables, the horizontal format make it problematic to provide the input and read the table.

To avoid the limitations of the standard table format, an option was added for the reactor power input to provide it in a separate file. The reactor transient input file (Heat Addition input in HAcontrol_dat.txt) only specifies the file name. Then, the table is stored in that file, in a vertical format of values versus time pair in each line. The value in this case is the reactor power, in % of the steady state value. This option is triggered by providing “0” for the number of entries in the standard table (if the input is not zero, the standard table is read and the file name for the external input is ignored).

Figure 35 shows an example of using this new option with the external file for the reactor power input. In this GUI form, the external file is read and is provided for the user to view. The file can be edited in GUI by clicking on the Edit button (or it can be edited in any text editor).

The change described here is only provided for the user convenience and only defines how the table for the reactor power is read by PDC. No changes in the code were needed for how the table is used in the code.

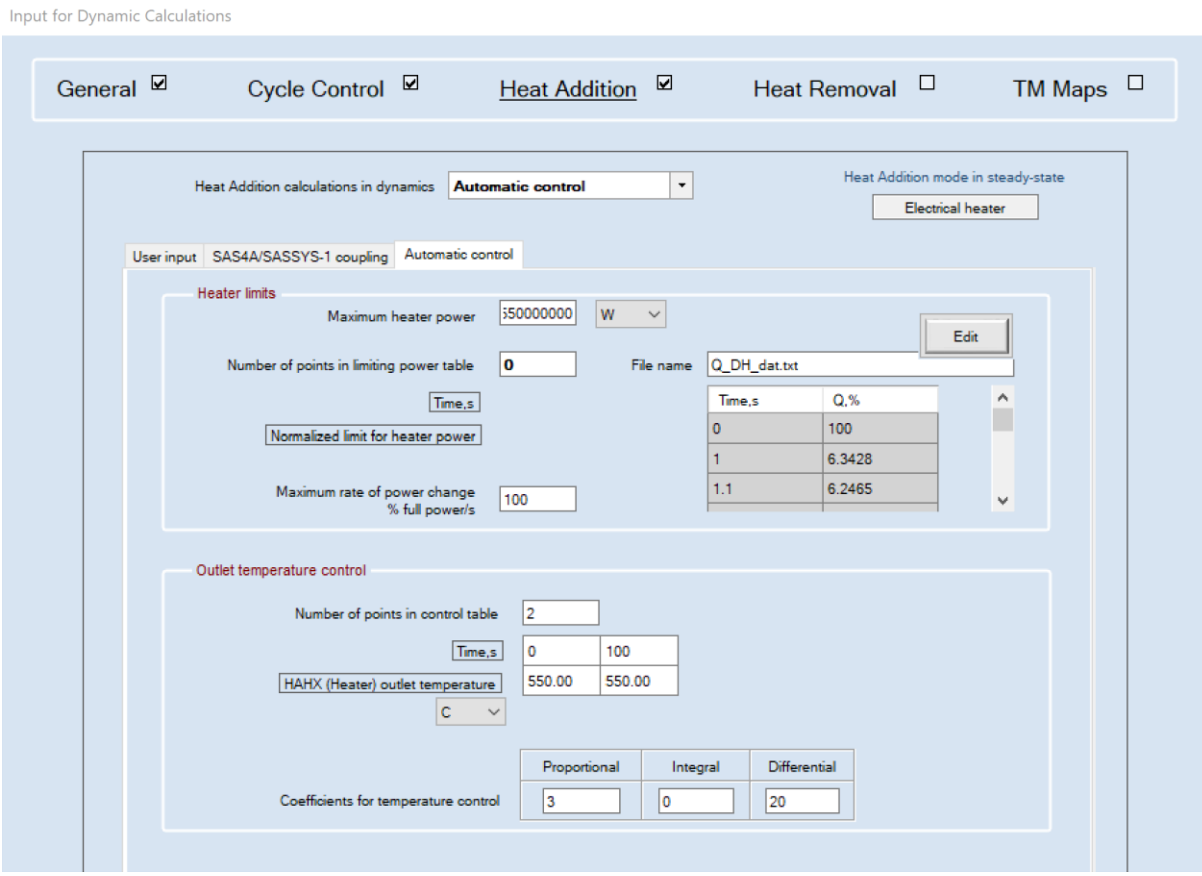


Figure 35. PDC GUI for Heat Addition Input.

6 Transient Simulation

In this chapter, transient simulation of the Pascal reactor is described. The primary purpose of this simulation was to demonstrate the new code features implemented in this project. Two series of transients have been run: shutdown heat removal system activation and load following. These two transients are described in Sections 6.2 and 6.3, respectively. Before the transients were run, however, the standard PDC transient preparation procedures were implemented. These procedures are briefly described in Section 6.1

It is important to note that the transient simulations presented in this chapter were carried out with the primary goal of demonstrating newly developed PDC featured and testing those features in transients. The transient simulation presented in this chapter is not intended to analyze in detail the dynamic behavior of the Pascal reactor. For example, the load following analysis is Section 6.3 is simulated with the main purpose of investigating the control of the two drive turbomachinery shafts and is not focused on the demonstrating (or optimizing) the load following capabilities of the Pascal reactor.

6.1 Transient Preparation

In PDC, all transients are run from the steady-state conditions. For the Pascal system, those conditions are provided in Figure 28 for the main cycle and in Figure 27 for the SHRS.

The turbine and compressor transient treatment in the PDC is based on turbomachinery maps. These maps were generated for each compressor and turbine. For the power turbine, only synchronous (fixed speed) maps were generated, as this component is connected to the grid and operates at the gid frequency. For the drive turbines and compressors, both synchronous and general (changing speed) maps were generated. The synchronous maps are used in simulation where the shaft speed in not expected to change. The advantage of these maps is that they allow more map points (higher resolution) on all other inputs (inlet temperature, inlet and outlet pressure) for the fixed file size. The general maps are used when the shaft speed is changing in the transient.

All transient calculations in PDC are preceded by the steady state initialization phase. At this phase, the dynamic equations are solved, but no transient initiator is introduced. The primary purpose of this phase in PDC is to smoothen the transition from the steady state solution (which is often obtained with fixed convergence) to the dynamic solution (which is obtained with its own convergence and approaches that could be different from those involved in steady state). The control action during this transient initiation phase is usually not applied, except for the maintaining boundary conditions, such as maximum and minimum cycle temperatures. The goal of this simulation is to obtain a stable transient solution before the actual transient begins. The checks are also made that the transient solution does not deviate too much from that obtained in steady state.

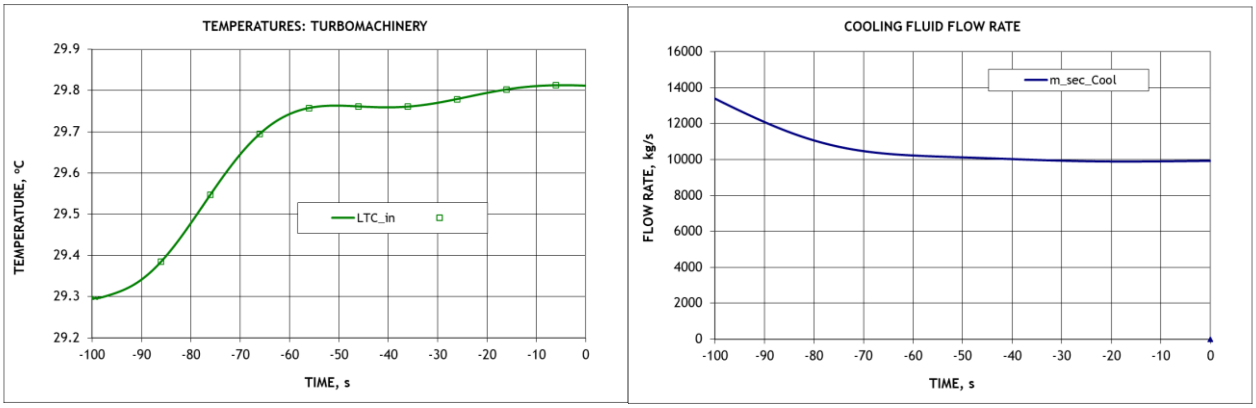
The next task in the PDC transient preparation is an optimization of some of the control mechanisms. In this context, optimization means selecting the PID control coefficients that provide stable but fast response to the control action request. Although most of the cycle control will be optimized for the transients where they are activated (for example, shaft speed control for drive turbine in load following), some controls are general for all transients and are therefore optimized before the transients are simulated. Usually, those controls are related to maintaining boundary conditions in terms of the maximum and minimum temperatures. For the Pascal reactor, these include:

- Reactor power control to maintain the reactor-outlet temperature,
- Cooler water flow rate control to maintain the LTC inlet temperature,
- SHRHX water flow rate control to maintain the SHRHX CO_2 outlet temperature.

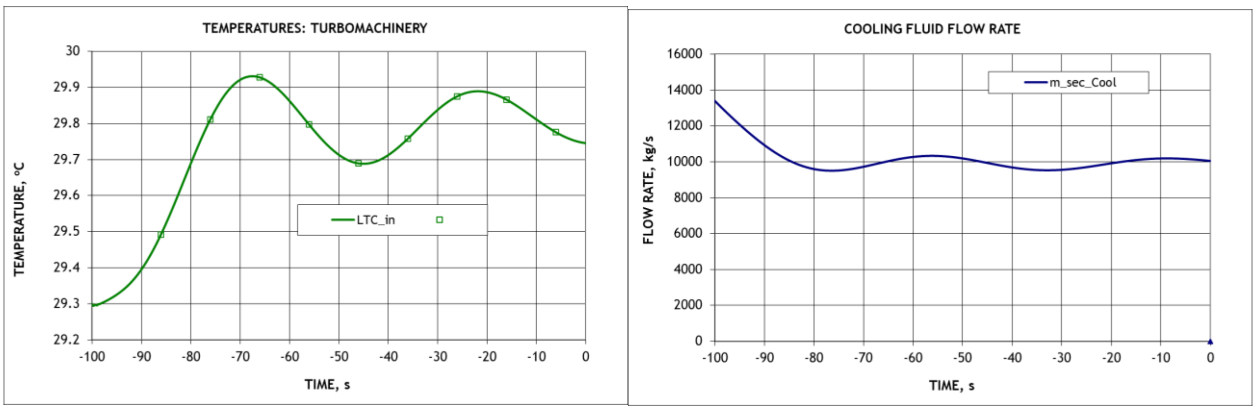
These controls were optimized during the pre-transient simulation. The target temperatures were subject to step change and the PID coefficients were selected to provide fast but stable response to achieve the new targets, as well as to maintain stable boundary conditions in the long term. Figure 36 shows the example of the compressor-inlet temperature control optimization, where the target LTC-inlet temperature was subjected to +0.5 °C step change (from 29.3 °C to 29.8 °C). Figure 36 demonstrates how the control PID coefficients were selected to obtain a desirable control response (this figure shows only few steps, but more variations of PID coefficients were tried). Note that LTC-inlet temperature control is one of the most sensitive controls in Pascal, since it maintains the CO_2 conditions very close to the critical point (CO_2 critical temperature is 30.98 °C). Still, even in these circumstances, PDC is able to provide stable control. Similar optimization was carried out for the other two external controls listed above.

The final step in the PDC transient preparation was to check the PDC restart capability. That feature provides means to offer more flexibility, in particular, for control during a transient. It will be used in the SHRS activation transient, which includes several distinct stages (such as valve closing, pump startup, etc.) that requires different approaches to control action. The details of control implementation for this transient are provided in the next section; for the restart feature check only the code capability to stop and then continue transient calculations was verified.

100 0 0



200 0 0



100 10 0

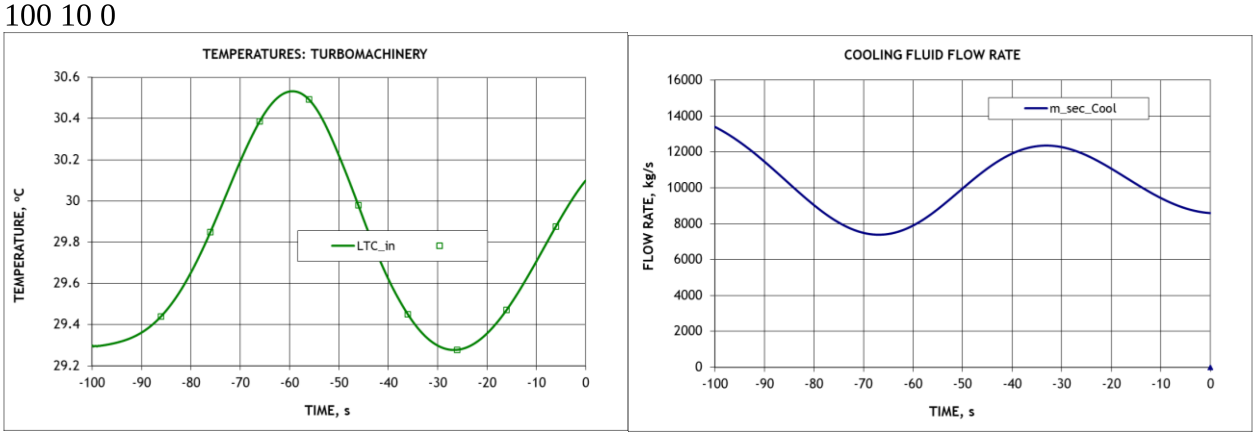
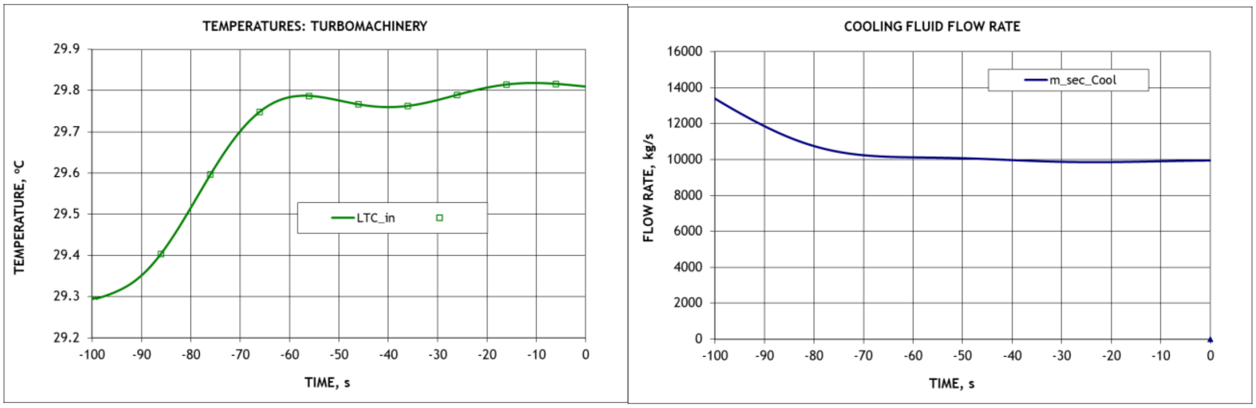


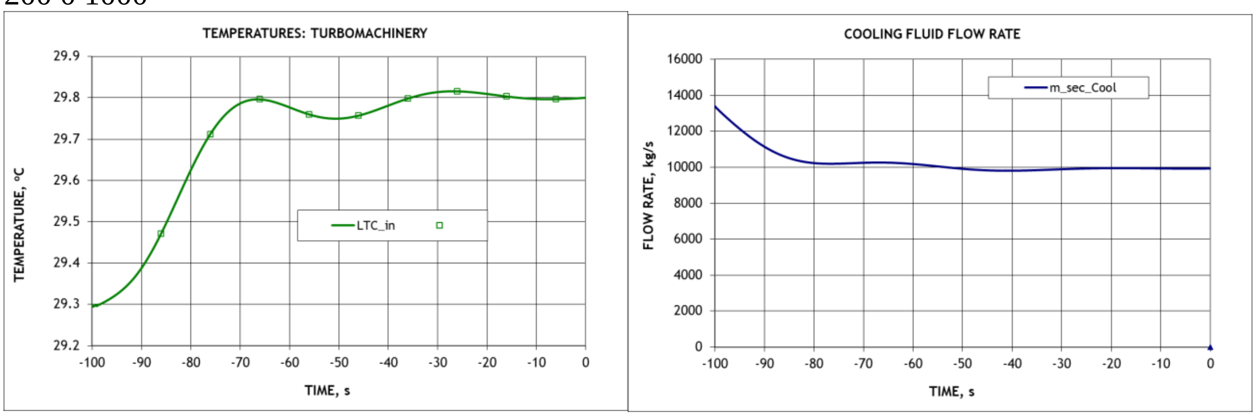
Figure 36. Optimization of Compressor-Inlet Temperature Control.

(continued on next page)

120 0 100



200 0 1000



175 0 1000

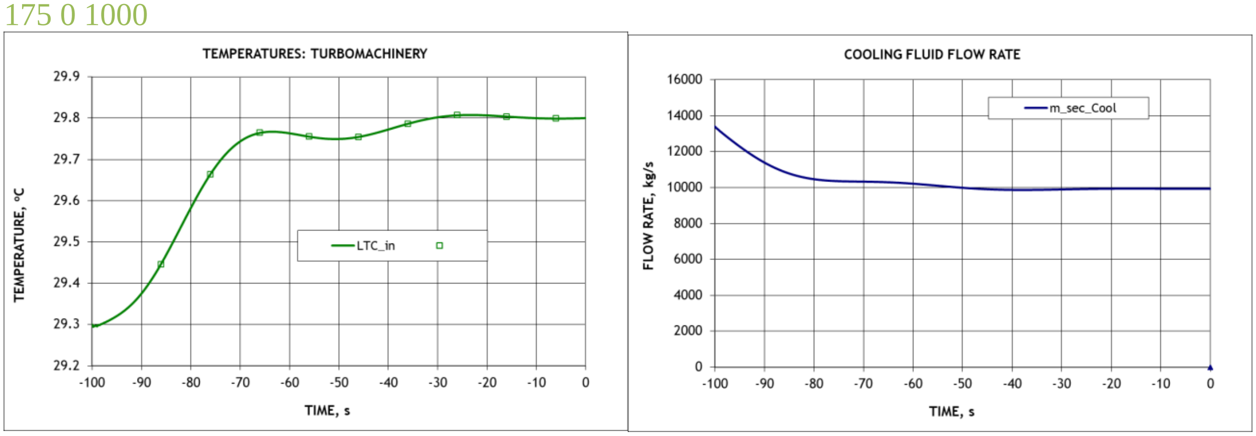


Figure 36. Optimization of Compressor-Inlet Temperature Control. (Continued)

6.2 SHRS Activation Transient

This transient simulates activation of the Shutdown Heat Removal System and transition from normal plant operation to decay heat removal mode by the SHRS. Figure 37 shows the PDC model of the SHRS, components, and valves relevant to this transient as well as the main valve control action as discussed below.

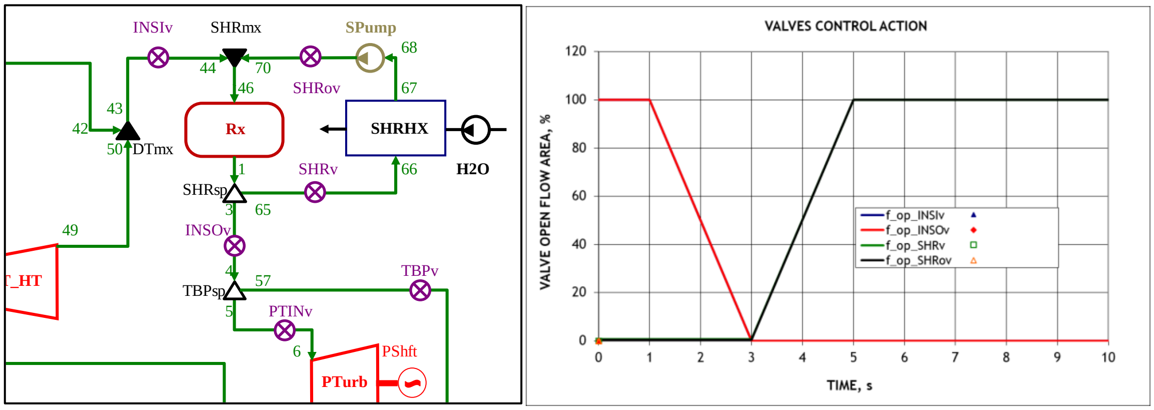


Figure 37. Pascal SHRS Model in PDC and Valve Control Action.

The following transient sequence has been specified by TerraPower for the Pascal reactor:

- At t=0 transient start. The reactor scram is initiated.
- At t=1 s reactor scram is complete. The reactor power reaches the decay heat level (~6%). The reactor power history after the scram has been provided by TerraPower. It is simulated in the PDC as a manual control for the reactor power (see Section 5.4). Main loop insulation valves (INSIv and INSOv in Figure 37) closing is initiated.
- At t=3 s, main loop insulation valves closing is complete (Figure 37 shows the valve action, with INSIv and INSOv lines identical). At this point, the main cycle is fully insulated, and the main cycle calculations are bypassed (as described in Section 5.1.2 Isolation Valves). Opening of the SHRS valves is initiated. Note that at the instance of t=3 s both main loop and the SHRS are fully closed. There is essentially no coolant flow in the reactor. Also at that time, water flow control for SHRHX is activated.
- At t=5 s, SHRS valves are fully open. SHRS pump is transitioned to the automatic flow control mode.

Similar to other results in this Chapter, the PDC simulation and results are provided for the code demonstration purposes. For this transient, the sequence specified above was provided for the PDC simulation. It was not the purpose of this simulation to optimize the Pascal operation during the decay heat removal mode in any way.

Figure 38 shows some of the PDC transient results during the main loop insulation valve closure stage. The reactor fuel power (Q_RHX_Rx curve) is defined by the simulated scram input and is reduced to 6% in the first second. The heat removal from the reactor by the CO_2 (Q_RHX_BC curve) is defined by the coolant flow rate and inlet/outlet conditions. The flow reduction does not start until t=1 s, and then it decreases until t=3 s, when closure of the insulation valves virtually stops the flow (heat removal after 3 s is related to the SHRS activation and will be discussed below). As the valves are being closed, the flow rates in the turbines and compressors (and the rest of the main loop) decreases to zero at t=3. Note that the flow decrease is not linear, as the valve resistance is not linear with the uniform valve closure. The CO_2 cycle loop pressures are also significantly affected by the valve closure action. The power produced by all turbines is reduced, and at some point before 3 s, cycle stops providing output to the grid (W_2_grid curve) and starts to draw power from the grid. Once the main loop is isolated at t=3 s, the calculations for this part are bypassed (the PDC results are maintained at the previous value). The compressor-inlet temperature is maintained during the first few seconds, while the reactor temperatures are affected.

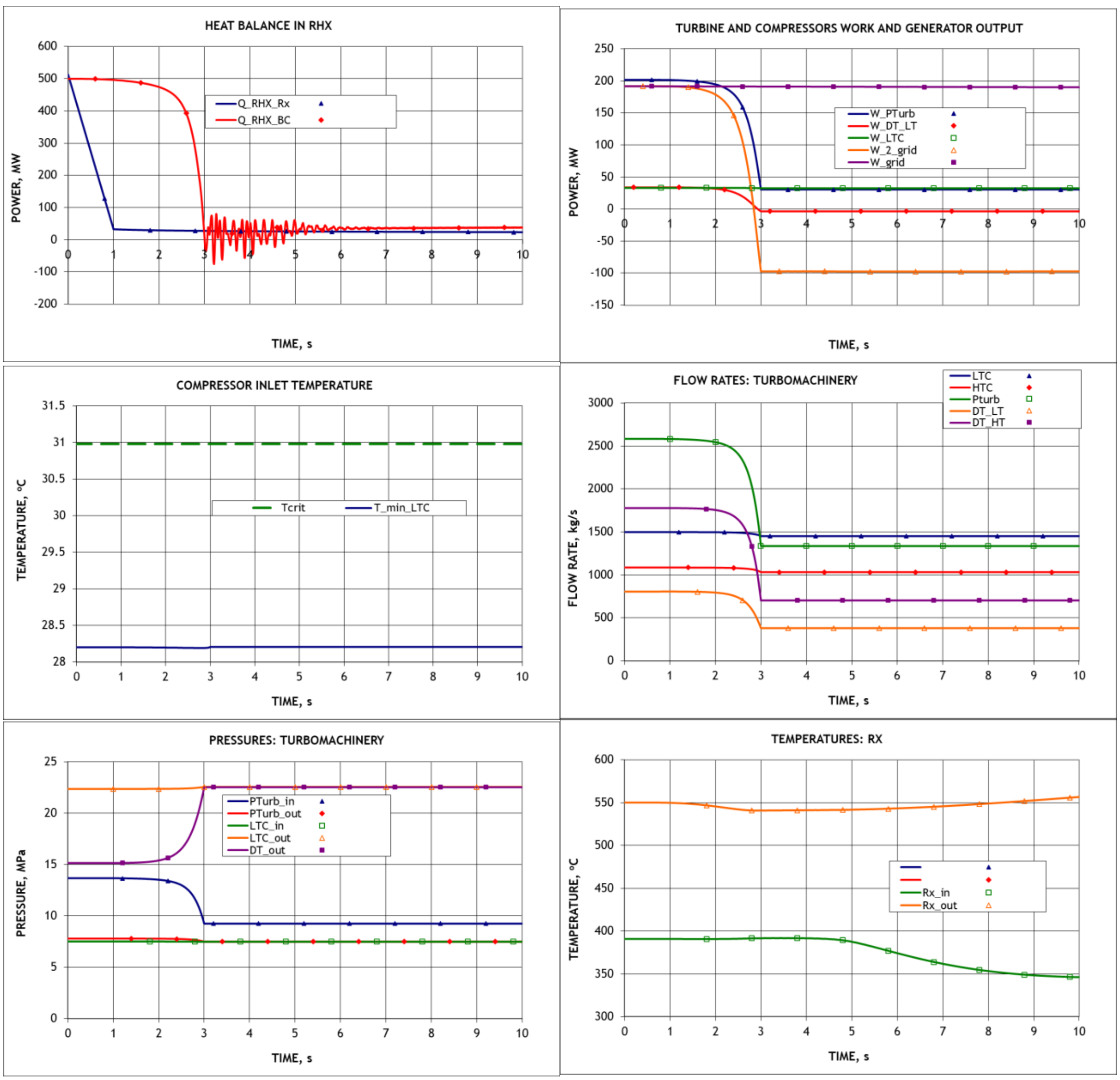


Figure 38. PDC Results: Insulation Valve Closure Stage.

Figure 39 shows some PDC results relevant to the SHRS activation phase of the transient. The startup of the SHRS pump is modeled with the manual control option with increase in the flow rate from zero to the design 5% flow in 2 seconds. The first plot in Figure 39 also shows flow rates in other SHRS pipes. These flow rates experience some oscillations during the valve opening phase, but rather quickly, in about 5 seconds, converge to the flow rate provided by the pump. The pump power also reaches the design value shortly after the pump is turned on. The SHRS pressures quickly collapse to almost a single value once the reactor is fully insulated, but then pressures and pressure drops are established as the flow starts to circulate through the SHRS. The SHRHX water flow is initiated after 3 s and the control is switched to automatic to maintain the SHRHX-outlet temperature of 390 °C, which is established by 7 s into the transient.

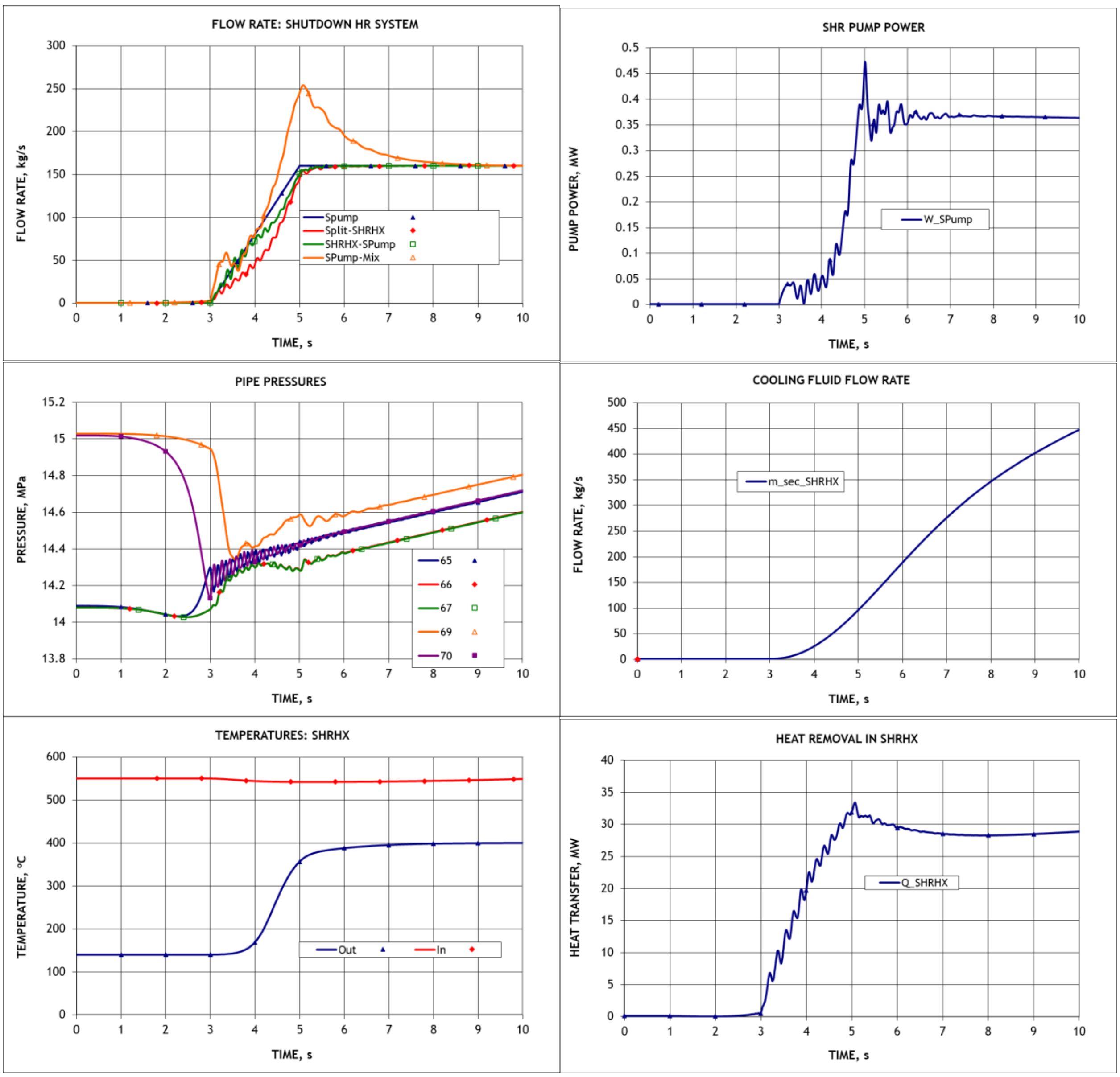


Figure 39. PDC Results: SHRS Activation.

Figure 40 shows the reactor results early in the transient. The flow rate in all nodes is reduced to zero as the main insulation valves are closed. Then, the flows oscillate around zero as the SHRS pump is activated and some flow through the reactor is established. The coolant temperatures reduce slightly in the first few seconds when the reactor is shutdown, then start to increase during the low-flow conditions. The reactor pressures are similar to those in the SHRS in Figure 39: once the flow is stopped, the pressures collapse to a single value, then they increase as reactor is being heated up. The channel wall temperatures are largely unaffected on this time scale. Cladding temperatures show a similar trend as the coolant temperature, while the fuel temperatures decrease rapidly with the reactor scram, then stabilize at some level (on this time scale).

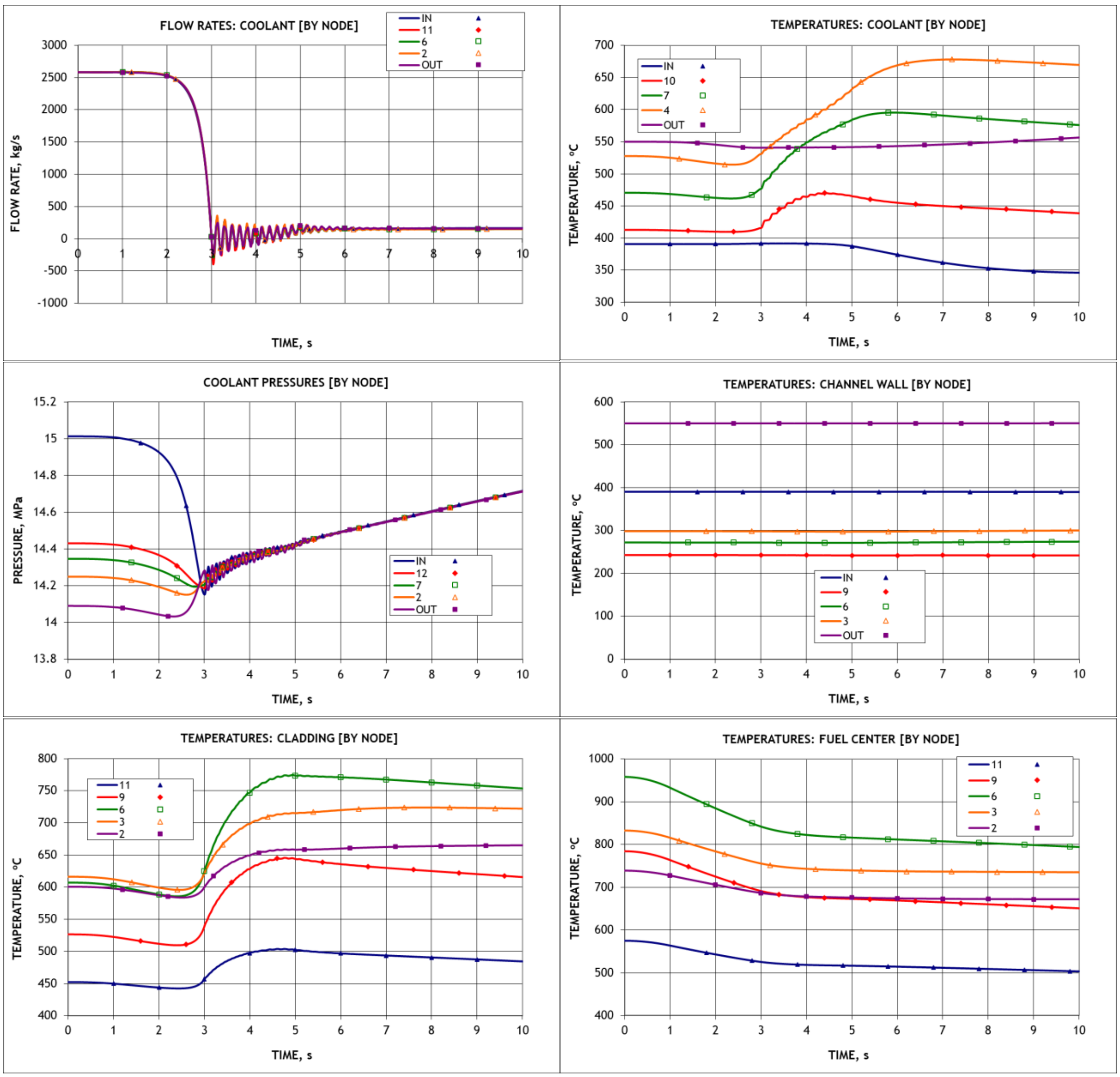


Figure 40. PDC Results: Reactor Early, Average Channel.

Figure 41 compares the early reactor results for the average and peak channels. The trends in all temperatures are similar, but the peak channel results show higher cladding and fuel temperatures. The coolant inlet and outlet temperatures remain similar between the two channels, as power-to-flow ratio remains the same for both channels. However, coolant temperature in internal nodes start to deviate due to heat capacity effects from the fuel. The peak cladding temperature of almost 850 °C is calculated in the peak channel at 5 seconds.

Average Channel Peak Channel

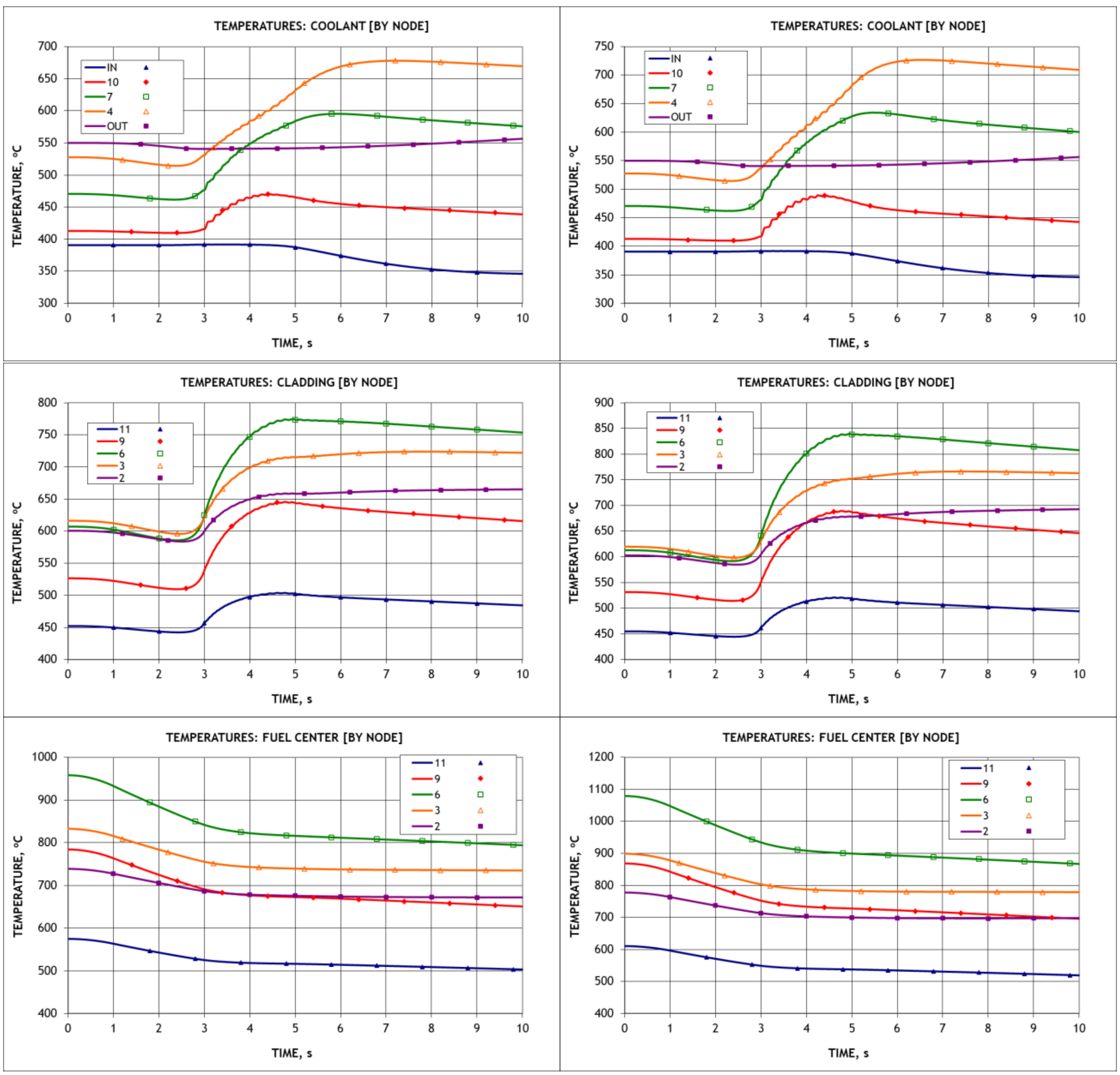


Figure 41. PDC Results: Reactor Early, Average vs Peak Channels.

Figure 42 shows the PDC results in the mid and long term, up to 1 hour of simulated transient. Following the short term results in Figure 38, the heat removal rate in the reactor is below the heat generation rate early in the transient. To compensate for that, the SHRS automatic pump control increases the flow rate through the pump. In this simulation, a limit of 200% relative to the pump design flow (5% nominal) is imposed. That limit is reached at around 50 s and pump operates at this flow rate for short time. Then, at around 100 s, the decay heat generation rate in the reactor drops below the heat loss to the heat sink. That heat loss was initially around 13 MW and remains at about this level as the channel tube temperatures does not change significantly in this transient. Once the heat generation drops below the heat loss, there is no need for the SHRS flow and the automatic control on the pump drops the flow rate to the minimum allowed flow (set to 10% of the design flow rate). The reactor outlet temperature, after short term increase, is maintained close to the design value of 550 °C, but start to decrease in the long term, along with other temperatures, when decay heat generation drops below the heat loss. Similar behavior is observed with the SHRHX water flow control, which is set to maintain the SHRHX-outlet temperature at 390 °C. Early in the transient, the demand for high heat removal rate leads to an increase of the water flow rate to the limit. After a short-term increase, the SHRHX-outlet temperature is maintained at the target level for about half an hour. But after the reactor starts to cool down from the heat loss, the SHRHX temperatures start to decrease, without the need for water flow rate, which is reduced to the minimum level as well. The system pressures, in general, follow the same trend as temperatures. There is a short-term increase, then the pressures are stabilized, and eventually start to decrease as the system cools down.

Figure 43 demonstrate the reactor results in the mid and long terms. Again, the trends here are similar to those discussed above, with a quick short-term increase, some stabilization in the first half of the hour, and then a slow decrease in the long term. The reactor results show that the second peaks in coolant, cladding in fuel temperatures do not exceed the peaks early in the transient observed in the previous figures. For example, the peak cladding temperature in the mid-term increases to 700 °C, significantly below the 850 °C level observed early in the transient.

Overall, the PDC results presented in this section demonstrate the PDC capabilities to simulate not only operation of the SHRS system, but also transition from the normal operation to the decay heat removal mode. The code was shown to be capable of calculating the transient with complete insulation valve closing and opening in another branch, as well as through the insulation of the main loop. Startup and operation of controls in the SHRS is demonstrated, although in this transient such operation was further complicated by the presence of the heat loss in the reactor, which exceeds the decay heat generation in the long term.

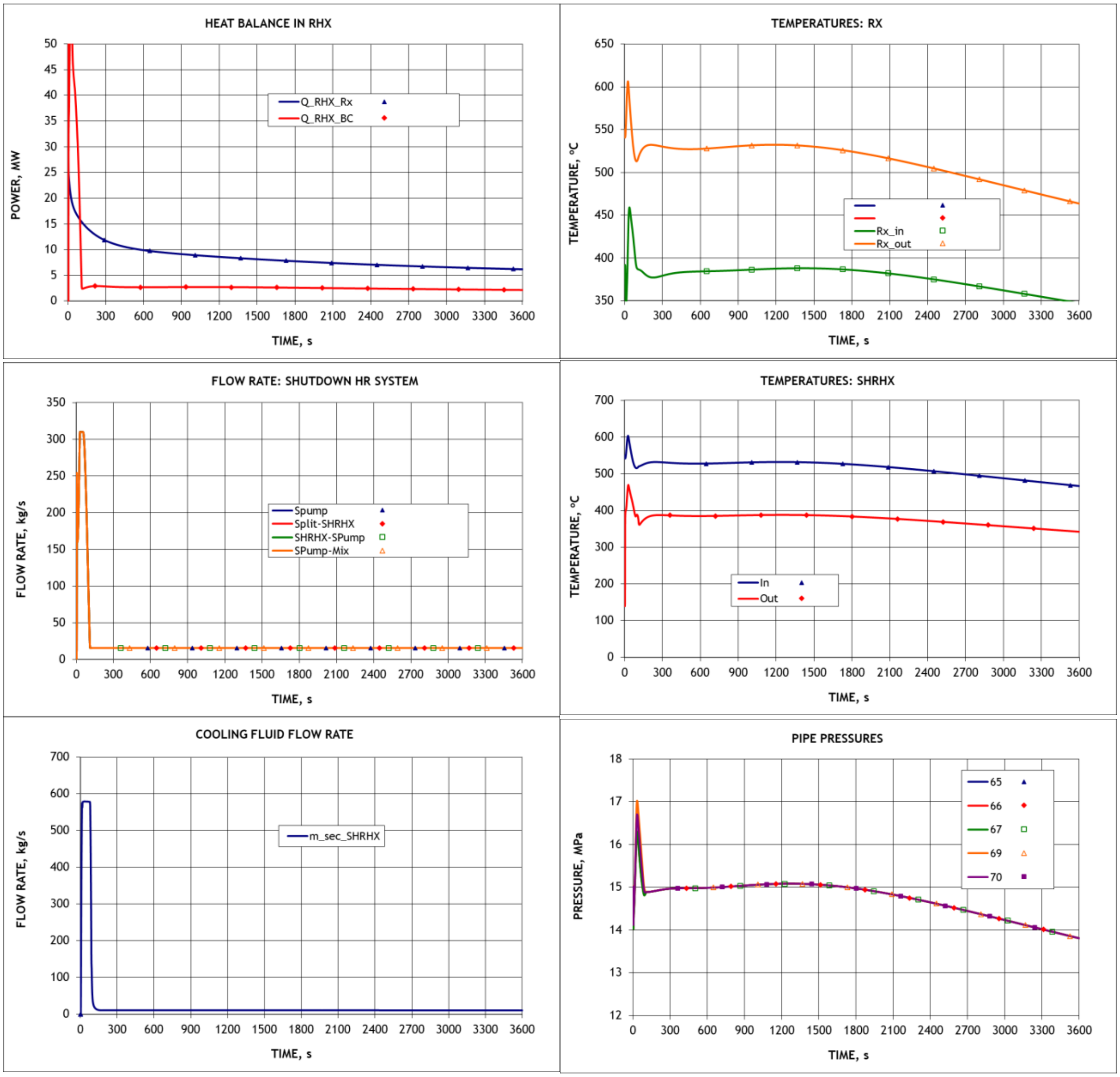


Figure 42. PDC Results: Mid and Long Term.

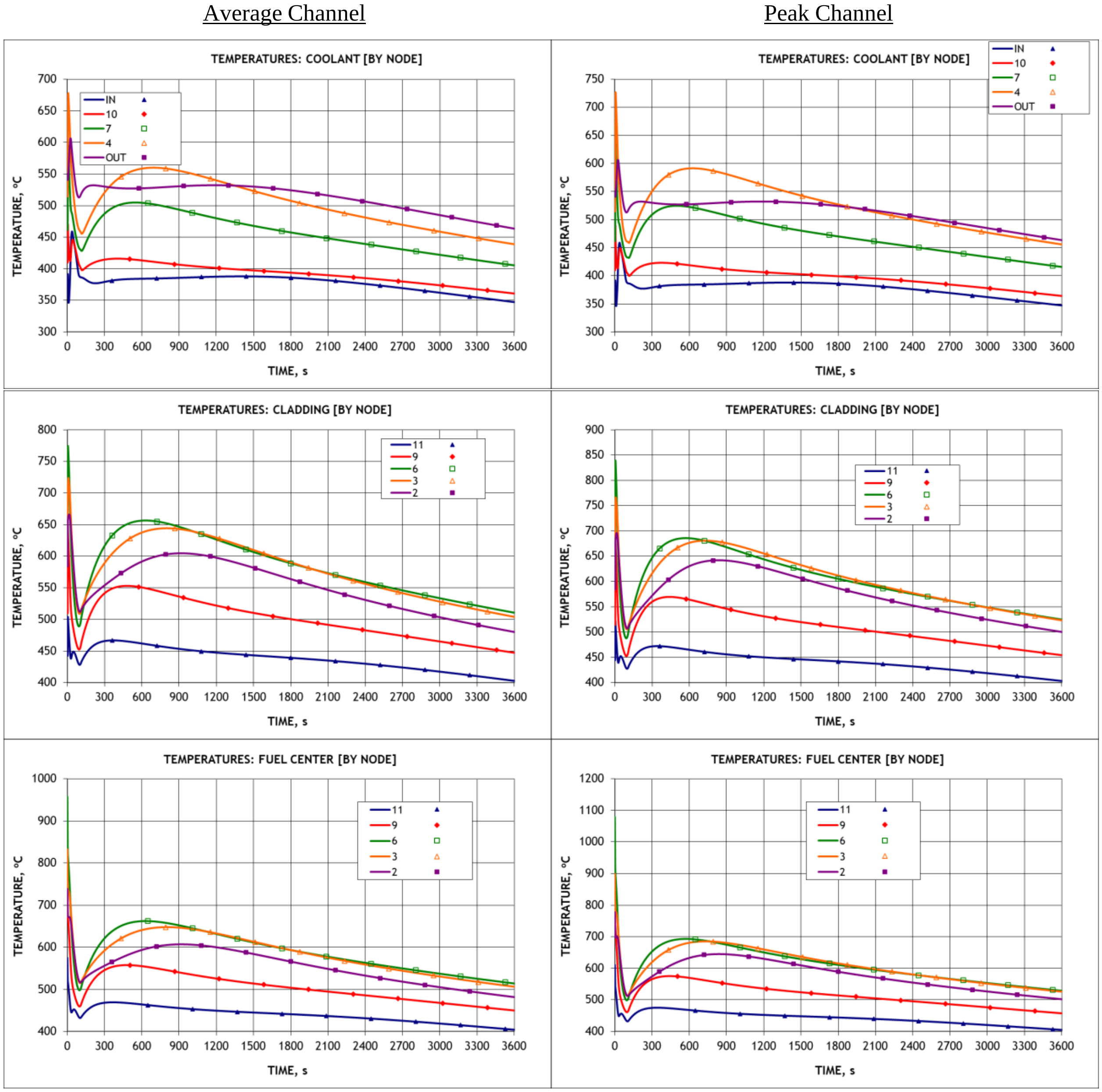


Figure 43. PDC Results: Reactor Mid and Long Term, Average vs Peak Channels.

6.3 Control Transients

Another set of transients was simulated in PDC, this time to demonstrate operation of the main plant controls, including those for the reactor and the main cycle. For this demonstration, the load following transients were simulated. The load following is defined as the normal reactor plant operating mode where the plant electrical output is varied to match the imposed changing demand from the electrical grid. For this simulation, a linear change in the grid demand at 5%/min was postulated. The demand is set to decrease from nominal 100% all the way to 0%, although the actual operating range of the reactor may be limited and will be investigated in this simulation.

The load following transients were selected for this demonstration since these transients tends to characterize the behavior of the entire plant and all components, including the reactor, turbomachinery, and heat exchangers. They also engage many control mechanisms, often in combination with each other.

It is again needs to be pointed out that the transient simulation presented in this section was carried out with the sole purpose of demonstrating the newly developed code features, specifically the redefined plant control logic and the transient operation and control of the new reactor module. The results presented here are not meant to represent a comprehensive load following analysis of the Pascal reactor. It was not the goal of this analysis to improve the load following performance of this design, if and when non-optimal performance is encountered. At the same time, the results presented here will certainly provide insights into Pascal reactor operation and control during load following for future comprehensive control analysis of this reactor design.

Figure 44 shows again the PDC model of the Pascal reactor used in this simulation. This time, the figure highlights the main cycle control mechanisms which will be simulated in the load following analysis, including:

- Inventory control, consisting of the inventory Tank and inlet (INVIv) and outlet (INVOv) valves,
- Turbine bypass with TBPv valve,
- Turbine throttling: drive turbine throttling valves (DTHIv and DHLIv) for drive shaft (DShHT and DShLT) speed control for low and high temperature compressors, respectively, and power turbine inlet valve PTINv (this control will not be simulated in this analysis).

In addition to these cycle controls which will be used directly for plant power maneuvering, another group of controls to define boundary conditions is also simulated in the Pascal PDC model and includes:

- Reactor power control, which is set to maintain reactor-outlet temperature of 550 °C in all transients analyzed here. The control acts on power produced in the reactor fuel.
- Cooler water flow rate control, which is set to maintain the compressor inlet temperature at the design level of 29.3 °C.

As described in Section 6.1, before the full transient simulation was carried out, the external controls for the reactor power and the water flow rate were optimized to achieve a desired control action. Even though the optimization in Section 6.1 was carried out for a postulated step change in target temperatures, the optimized controls demonstrated favorable behavior in all control transients presented here, such as no re-optimization of these controls was necessary.

Figure 44 also shows other controls included in the PDC Pascal model, which will not be used in the simulations presented here. These controls include the cooler and recuperator bypass lines and valves (CBPv and RBPv), compressor outlet throttling valves (LTCOv and HTCOv), and the main cycle isolation valves (INSIv and INSOv). These valves will remain either fully closed (for bypass controls) or fully open (for compressor throttling and isolation valves) during the transients. Likewise, although Figure 44 includes the SHRS loop of the Pascal reactor, this loop is not activated in the control transients and will remain fully isolated.

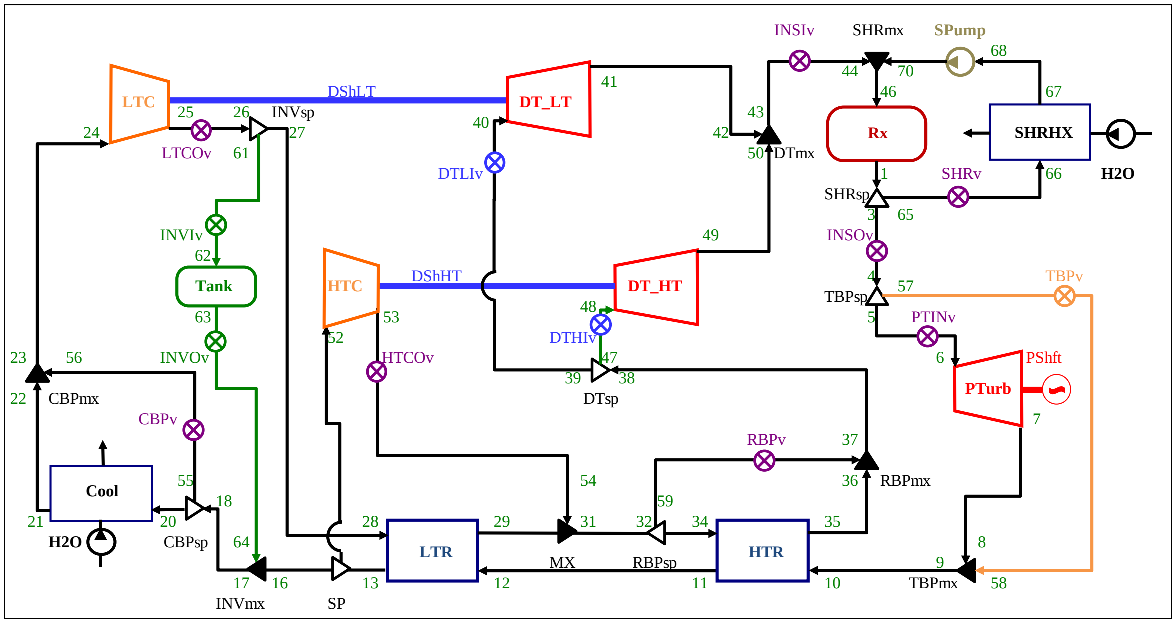


Figure 44. Pascal Model in PDC and Cycle Control Mechanisms.

In the simulations presented in this section, a few specifics of the Pascal design related to the plant control and load following were identified. First, as was already discussed above, the design employs three turbomachinery shafts, with two drive shafts where the turbines drive two compressors. For these drive shafts, two grid connection modes where simulated. In the synchronous mode, the shaft rotational speeds are fixed, and they deliver or draw power to or from the grid as needed. The second mode is disconnected mode, where the drive shafts are not connected to the grid and their speeds are allowed to change, depending on the power balance between the turbine and the compressor. Because the disconnected operation mode is more challenging to simulate, it will be assumed in all control calculations described above. For control purposes, the target shaft speeds will be fixed at 100% nominal (steady state) values, unless shaft speed is actively varied as a power control mechanism.

The second Pascal specific is the approach to the critical point in the Low Temperature Compressor. Figure 45 demonstrates the LTC design conditions on the CO_2 temperature-entropy diagram. The compressor inlet conditions (green marker) is above the critical pressure but below the critical temperature, and thus is located in the supercritical liquid region. The first stage inlet conditions (red symbol), which is obtained after the flow accelerate to the impeller flow area, is below both the critical pressure and the critical temperature, although it is still above the saturation line (subcritical liquid). That close approach to the critical point and the saturation line proved to be challenging in both generating the LTC performance maps (where inlet temperature and pressure had to be varied independently over a specified range) and during the transients. In particular, the issues were encountered in transients where the compressor-inlet temperature was maintained, but compressor-inlet pressure was reduced, such as during the inventory control. When pressure is reduced at fixed temperature the conditions shift to the right on the T-S diagram. As shown in Figure 45, for the Pascal LTC design, this shift is very limited before two-phase flow conditions are encountered. Therefore, special attention was paid to generating performance maps for the LTC, as well as for the temperature and pressure control during the load following transients.

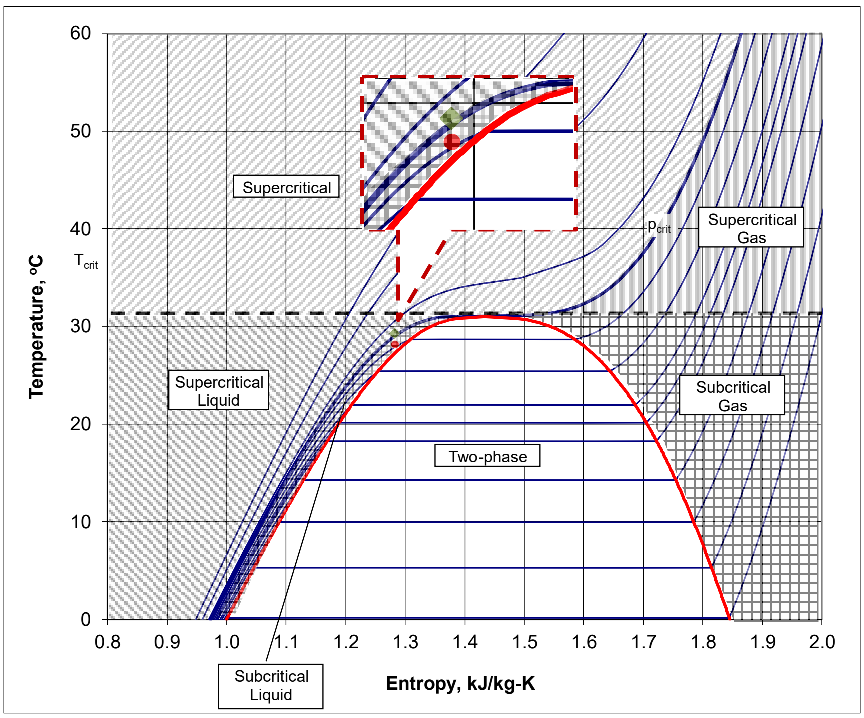


Figure 45. Pascal LTC Inlet (Green) and First Stage Inlet (Red) Conditions.

The final goal of the load following simulations with PDC was to run the transients in an automatic mode, where the user only provides the load table (grid demand versus time) and the code calculates the required control action and the plant transient response. However, some control mechanisms require the control tables to be fully automatic. For example, the shaft speed control needs a table of how much the speed needs to be changed depending on the grid demand. Likewise, the inventory control needs a table of CO_2 mass removal from the cycle. To generate such tables, the control transients were run in two stages. At the first stage, a manual and slow control action was implemented. For example, the drive turbine shaft speed was slowly reduced from 100% down over 10,000 seconds. Then, the net generator output was recorded as a function (table) of the shaft speed. On the second stage, that table was provided as an input, and the load following transient was run in automatic mode, without any manual action. This process was repeated for all transients described below. In this report, only the final, automatic, stage results are presented.

6.3.1 *Turbine Bypass Control*

The turbine bypass control (Figure 44) acts by sending some of the CO_2 flow around the power turbine, thus reducing the turbine work and therefore the generator output. This is usually the simplest control to implement, as it requires only one valve activation. It is also one of the fastest controls, as it act on pressure difference and thus the control action occurs at the speed of sound. At the same time, the turbine bypass is usually the least efficient control, since its action is based on not extracting a portion of useful work in the turbine.

In the turbine bypass control simulation, the drive turbine shaft speeds were assumed to be maintained at 100% speed. The simulation is still done in the disconnected (from the grid) mode for these shafts, so the PDC also calculates the required drive turbine throttling valve actions to maintain that 100% speed.

The results for the turbine bypass control are shown in Figure 46. Similar to other results in this section, the first plot shows the power balance in the system. The two curves on this plot important for the load following purposes are: W_grid is the user-specified grid demand and W_2_grid is the net generator output from the plant. To successfully achieve the load following, these two curves should be on top of each other, which is the case for the results in Figure 46. The resulting valve control action, as the percent open for each valve relevant to this transient, is shown in the second plot in Figure 46.

Although the plan was to investigate load following down to 0% load, the results in Figure 46 show that only load variation in the 100%-50% range can be achieved by this control. The PDC results show that the control range is limited by the stall margin limit on the high temperature compressor. Since the main goal of the present analysis was the demonstration of the PDC new features and not the load following optimization of the Pascal reactor, no effort was made to redesign the HTC in order to extend its operating range. Therefore, this simulation was limited to the load variation from 100% to 50% and then back to 100%. This limit on the HTC stall was found to be common for other control simulations presented in this section.

Other results in Figure 46 show that the rest of the goals of the control simulation with turbine bypass are successfully achieved. For example, the reactor outlet temperature is maintained very close to the target value of 550 °C throughout the entire transient.

Figure 47 shows the transient results during the turbine bypass control on the reactor side. All these results are as expected, without anything standing out. As the result of the control action on the cycle, the CO_2 flow rate in the reactor is reduced slightly. This reduction is followed by the power adjustment, such that coolant temperatures at the reactor outlet are maintained (while the reactor inlet temperature increases in the middle of the transient). The fuel temperatures, in both the average and the peak channels, are decreased as a result of lower reactor power. The cladding temperatures increase slightly at the inlet but decrease slightly for the outlet nodes.

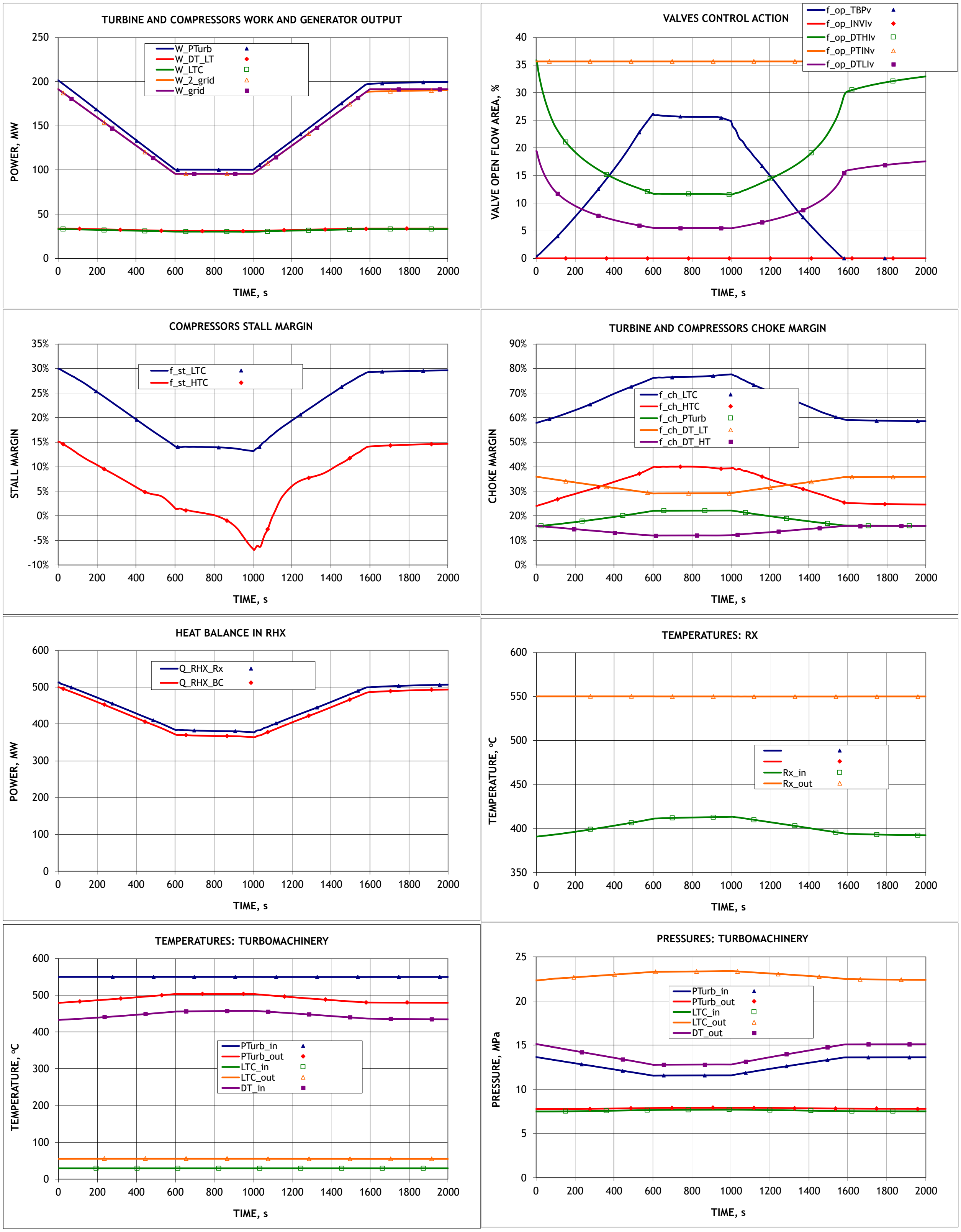


Figure 46. Load Following Results: Turbine Bypass Control.

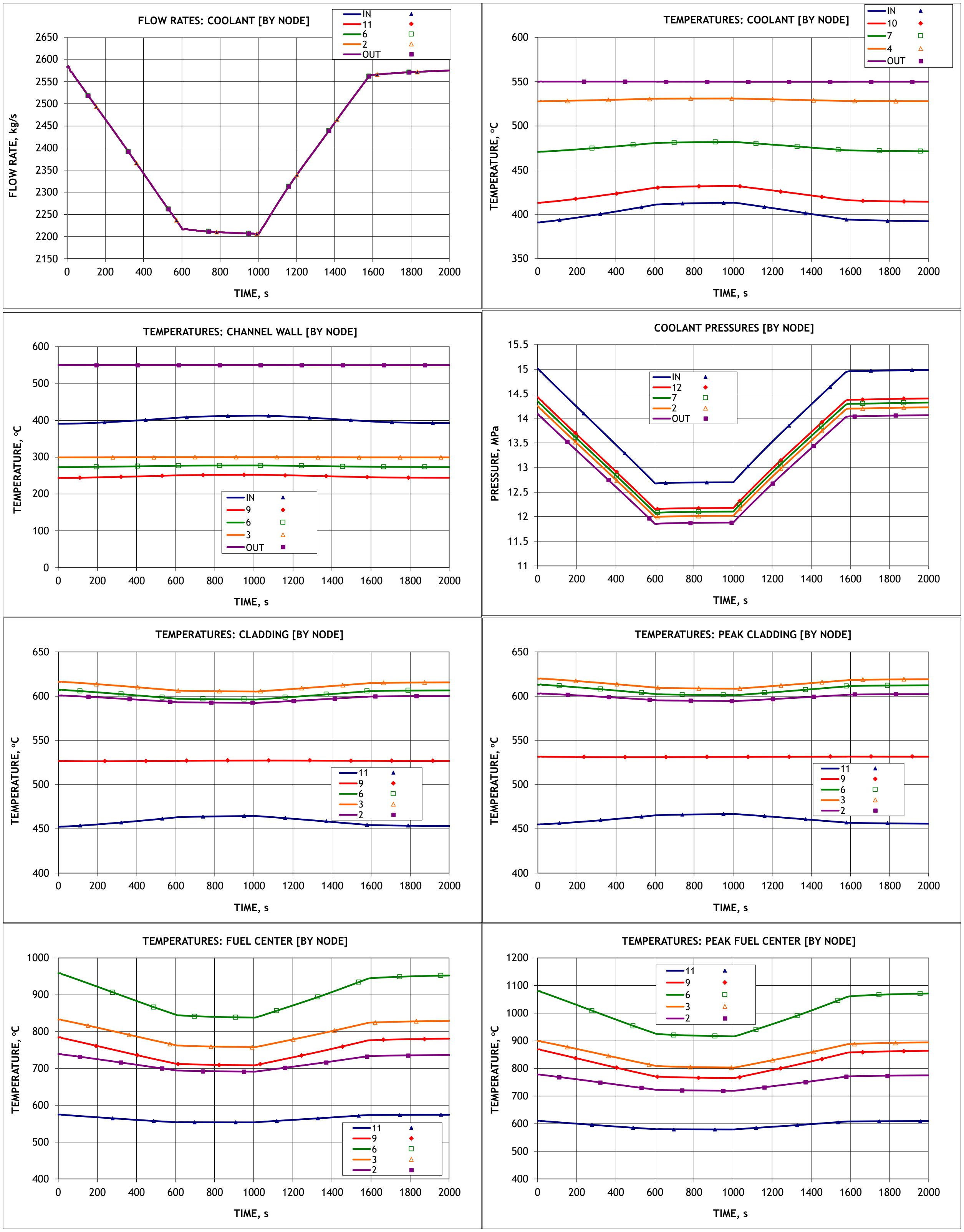


Figure 47. Load Following Results: Turbine Bypass Control, Reactor Results.

6.3.2 *Shaft Speed Control*

The Pascal sCO_2 cycle configuration with three turbomachinery shafts allows for independent variation of the compressor speeds, while the power turbine stays connected to the grid at synchronous speed. A change in the compressor rotational speed results in the change in the pressure ratio delivered by the compressor and thus affects the flow rate through this compressor, and consequently the flow rate in the entire cycle. The Pascal configuration allows for an independent speed and flow rate control for the HTC and LTC; however, in the simulation presented here, a symmetrical speed change was assumed where the shaft speed for both compressors was varied to the same degree. This uniform change in speed of two compressors is still obtained by independent control of two drive turbine throttling valves.

Simulation of the compressor speed load following with PDC was done in two stages. First, the compressor speed was varied manually slow enough that conditions close to steady state are maintained. From this simulation, a table of generator output (in %) versus the drive shaft speed (in %) was created. Then, that table was used as a target compressor speed versus grid demand in an automatic control mode.

The results of the Pascal reactor load following simulation with the compressor speed control are shown in Figure 48. Similar to the turbine bypass results presented above, the load following was limited to the 100%-50% range by the HTC stall limit. The results in Figure 48 also show that the compressors operation and/or their controls become unstable in the middle of the transient, - this is expected as operating of a compressor below the stall/surge limit can be intrinsically unstable. In any case, refining of the compressor design or control in this regime was not the goal of this work. Other than those oscillations, the rest of the plant's response to the load variation is favorable, as the load is closely matched, and other cycle and the reactor conditions are maintained. The results in Figure 48 show that relatively small change in the compressor speed, less than 10%, is sufficient to vary the plant output by 50%. In order to achieve these variations in the shaft speed, both the drive turbine inlet valves are throttled to about 1/3 of their original open fraction (see f_op_DTHIv and f_op_DTLIv curves).

The results on the reactor side for the compressor speed load following are shown Figure 49, for the average and peak channel. Overall, the reactor results are similar to those obtained for the turbine bypass control. The change in CO_2 flow rate (this time imposed by compressors slowing down) is compensated by the reactor power adjustment, such that the coolant temperatures at the core outlet are maintained. The fuel temperature, again, are reduced due to lower reactor power at lower grid demand.

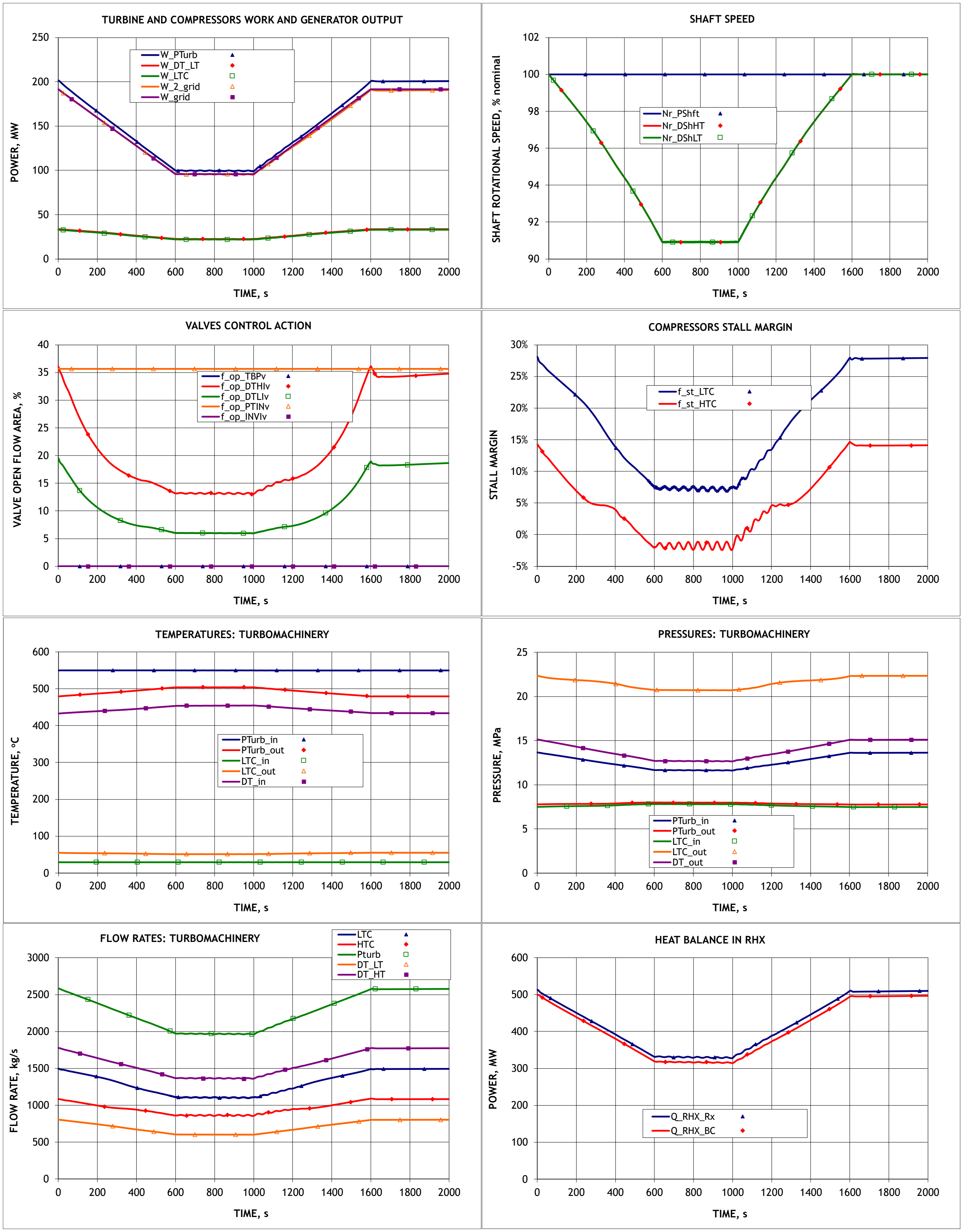


Figure 48. Load Following Results: Compressor Speed Control.

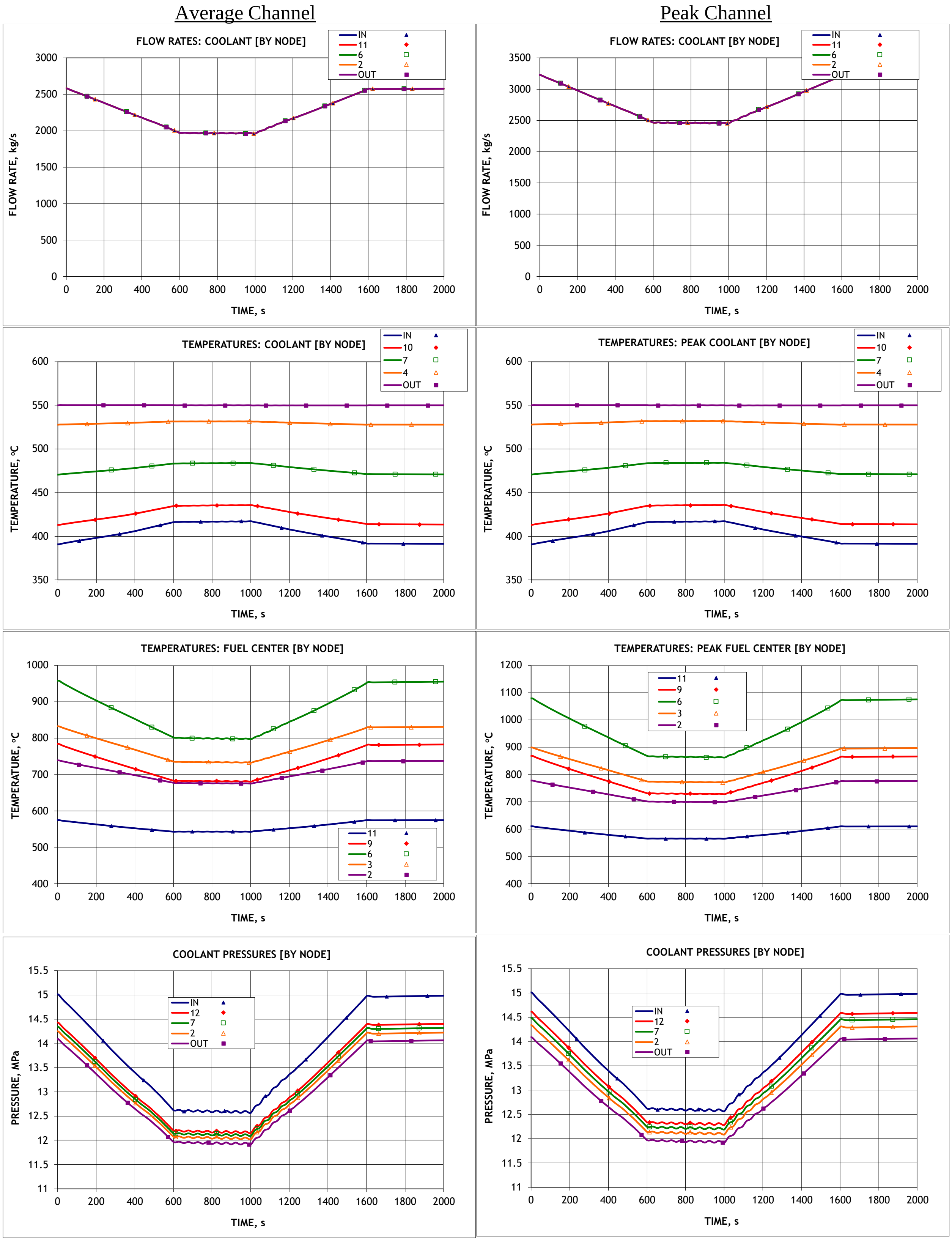


Figure 49. Load Following Results: Compressor Speed Control, Reactor Results.

6.3.3 *Other Control Transients*

In addition to the turbine bypass control and the compressor speed control, other load following transients were simulated for Pascal design with the improved PDC code. As the two transients presented above have already demonstrated the new code features for the reactor transient simulation and the enhanced control logic, these other transients are only briefly mentioned here in this report.

6.3.3.1 *Inventory Control*

An attempt was made to simulate load following by Pascal reactor with the inventory control. This control acts by removing some of the CO_2 mass from the cycle (and adding it back when needed) and storing it in the external inventory control tank (see Figure 44). When mass is removed, CO_2 density is decreased everywhere the cycle. Since the compressors tend to operate at relatively constant volumetric flow rate (and velocity triangles), reduction in density leads to proportional decrease in the flow rate delivered by the compressors. This control action is usually beneficial to the Brayton cycles, as change in density does not affect the cycle temperatures much (other than from reduced turbomachinery efficiency at off-design conditions), such that the cycle efficiency close to the design value is maintained at partial loads.

However, as discussed at the beginning of Section 6.3, the implementation of the inventory control proved problematic for the Pascal design. As CO_2 mass is removed from the cycle and cycle pressures are decreasing, the low temperature compressor inlet conditions move to the two-phase dome. As PDC is not designed to properly handle two-phase flows, the range of the inventory control was very limited in the current simulations. Load reduction from 100% to only 85% could be accommodated before issues were encountered with the LTC performance calculations and generating of the compressor maps. Therefore, full potential of the inventory control could not be realized for the Pascal concept. However, a combination of the inventory control with other controls was still investigated.

6.3.3.2 *Combination of Control Mechanisms*

As discussed above, the range of the inventory control is limited for the Pascal reactor due to close approach to the two-phase dome at the LTC inlet. This approach effectively prohibits the LTC inlet pressure reduction in the load following transients. At the same time, the results of the two control mechanisms for which the full load following simulation was completed, the turbine bypass control and the compressor shaft speed control, show that these controls act by *increasing* the LTC inlet pressure. Therefore, theoretically, there could be combinations of these controls and the inventory control which would maintain the LTC inlet pressure. Together with the already implemented the LTC inlet temperature control to keep this temperature fixed during load following, this could result in maintaining both LTC inlet temperature *and* pressure, thus avoiding the issues with the LTC operation in the two-phase domain. At the same time, this combination would allow to at least partially realize the benefits of the inventory control for the cycle efficiency at partial loads.

To demonstrate the feasibility and features of control combinations for Pascal load following, two new transients were simulated. In the first transient, the turbine bypass control was combined with the inventory control. In the second transient, the compressor shaft speed control is simulated together with the inventory control. In both simulations, the goal was to find such a combination of the two controls that, on one hand, maintains the LTC compressor inlet pressure at the design level (7.5 MPa and thus above the crucial pressure) and, on the other hand, still follows the specified load schedule at 5%/min rate. The load reduction was simulated from 100% to 50%, consistent with the results obtained in previous simulations in this section.

The results of this simulation of combination of controls are shown in Figure 50. This time, the main results are shown as a function of the generator output or load, even though each line is still obtained in the full transient simulation. For each transient, except for the inventory control, the results in Figure 50 are obtained with the load following at 5%/min range. (For the inventory control, these load following could not be obtained as discussed above).

The results in Figure 50 demonstrate that while turbine bypass and compressor speed control by themselves increase the LTC inlet pressure, a combination of these controls with the inventory control allows to maintain the LTC inlet pressure in the transient (the LTC inlet pressure lines for INV+TBP and Speed+INV are identical at 7.5 MPa). At the same time, the results in Figure 50 show that addition of the inventory control is beneficial, since higher cycle efficiency is maintained at partial loads. Specifically, *the highest cycle efficiency at loads below 100% is obtained for the combination of the compressor speed and inventory controls*. And since no other issues were recorded for these results, this control combination is recommended to be considered as the optimal control approach for Pascal reactor load following. Again, these recommendations are only preliminary as the results presented in this section were not intended as a comprehensive control analysis of the Pascal reactor design.

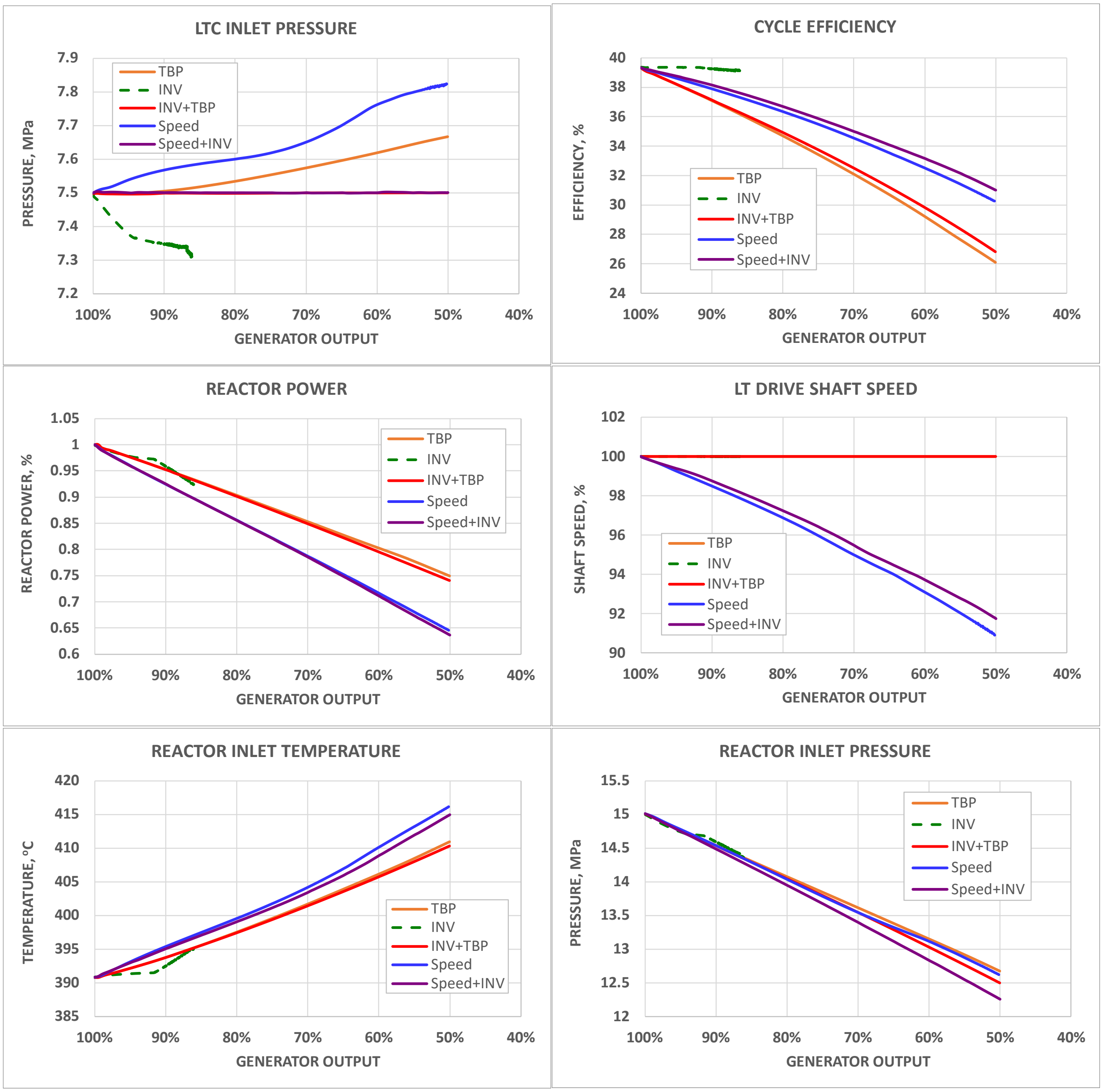


Figure 50. Control Combinations and Comparison of Control Approaches.

7 PDC Update on Model Selection

Although it was not originally planned for this project and the PDC modification presented here was not specific to the Pascal reactor, one more important code modification was introduced in this work and is described in this report. This code modification is related to the defining of the model for the calculations. In PDC, multiple input (and output) files are stored in the Data directory, which further branches into the steady state (SS) and dynamic (DY) sub directories. In previous calculations with the PDC, a copy of the Data directory was kept for each simulated model (like AFR-100, Pascal, etc.) and then replaced in the code folder as needed.

A code change was introduced to keep all models together and allow the user to select which model to utilize in the PDC calculations. This was done by changing the code's Data directory structure and by the introduction of a new input file.

The PDC Data directory now has first-level directories for each model. For example, the input and output files for the Pascal reactor developed in this work are stored in the Data\Pascal directory. That directory further branches into the SS and DY subdirectories as before. The input and output files for other PDC models are stored in their own subdirectories of the Data folder.

A new input file, *Model_dat.txt*, was introduced to allow the user to select which model to work with. The file is located in the Data directory of the code. The file has only one input (in the first line) which specifies which model subdirectory of the Data folder to be used. This is now the first input file that is read by the code and defines where the input files are located and are read by the code. The rest of the PDC calculations are the same as before. The output files are now saved to the subdirectory of the Data folder as specified in this new input.

An update to the Graphical User Interface main PDC form was introduced to allow the selection of the PDC model. This form will then save this selection to the Model_dat.txt input file to be read by the code.

The new code feature of the model selection was tested on the Pascal input, as well as the inputs for all previous models still maintained for PDC. This test also, importantly, verified that the code changes introduced in this work are applicable to the previous models and the results consistent with previous calculations are still obtained for these old models. The input files for the previous PDC model were modified to be consistent with the changes introduced to the code in the work described in this report. For example, all BCcontrol_dat.txt input files for the cycle control were changed to reflect the new input structure described in Chapter 5. None of these changes were needed for the Pascal model as these changes were already implemented during the work on this project.

Summary

The Plant Dynamics Code (PDC) has been previously developed at Argonne National Laboratory for simulation of supercritical carbon dioxide (sCO_2) Brayton cycles. The code was used extensively in the past for design, steady state, and transient analysis of sCO_2 cycles coupled to various reactor concepts.

Under a new Technology Commercialization Fund provided by US Department of Energy with cost share by TerraPower, the capabilities of the PDC were significantly extended to allow simulation of the direct sCO_2 cycles where the reactor coolant is the same working fluid used in the power conversion cycle. An example of such direct sCO_2 systems is the TerraPower's Pascal reactor.

To establish the baseline for further code development, the Pascal sCO_2 cycle has been modeled first using the existing PDC capabilities. The Pascal sCO_2 cycle features a unique split-expansion configuration where the main heat addition to the cycle in the reactor occurs between the turbine stages in order to limit the CO_2 pressures in the reactor. This cycle configuration was successfully modeled in the PDC, except for the reactor module which was represented by an electrical heater. This also includes modeling of the special features of the Pascal's sCO_2 cycle, such as three-shaft configurations where a power turbine is separated from the two drive turbines, which provide power only to drive the two cycle compressors, as well as low temperature compressor operation at supercritical inlet pressure but a subcritical inlet temperature. The design of each cycle component, including turbines, compressors, and heat exchangers was implemented in the PDC at the level required for steady state and transient simulation. The performance of each component and the entire cycle was verified by comparing with the Pascal design information.

The primary focus of the PDC code modifications has been an introduction of the reactor component. Pascal reactor uses a channel-type configuration where a number of coolant (channel) tubes contain several fuel pins and are surrounded by moderator. The PDC modifications were introduced to simulate this channel-type geometry, as well as important features of this concepts, such as heat transfer from fuel to the coolant, as well as from the coolant to the channel tubes and eventually to the moderator. This channel-type geometry is now one of the three design options supported by the PDC reactor model; the other two being the pin-type structure (inherited form previous shell-and-tube heat exchanger and electrical heater models) and the prismatic-type graphite-based geometry (implemented previously for simulation of gas-cooled reactor cycles).

Further development of the new PDC reactor model was implemented to facilitate the simulation of the following aspects and features important for nuclear reactors in general and to the Pascal concept in particular:

- Heat loss from the channel tube to the moderator was simulated as constant-temperature heat sink. This heat transfer path is especially important during low power operation, such as under decay heat removal mode.
- Calculations of the fuel temperatures were extended to provide an option of multi-node fuel temperature treatment in radial direction. This approach not only allows for capture

of temperature dependence of fuel thermal conductivity, but also provides for more accurate calculations of fuel centerline and peak temperatures, both in steady state and during a transient.

- Uranium oxide fuel properties were added to the PDC properties database.
- The fuel-cladding gap thermal resistance was simulated as a fixed gap conductance provided by the user.
- Treatment of unheated (no fuel) inlet and outlet sections was included into the reactor module. For the channel-type geometry, an option is provided to simulate these inlet and outlet sections either with empty channel tubes or with the same geometry as in the core.
- An option was introduced to provide an arbitrary axial (i.e., along the flow direction) mesh, along with a user-specified axial power profile in the fuel.
- A form loss coefficient was introduced into the pressure drop calculations. An input for the core and the inlet and outlet sections is provided by the user.
- A concept of multiple power-and-flow channels was introduced to track peak reactor temperatures, in addition to the reactor-average values. A number of such channels can now be specified by the user, and the power and flow factors for these channels, relative to the average channel, are provided in the input. These power and flow factors are assumed to be fixed during a transient.

The PDC steady state and dynamic equations were modified to include the new reactor geometry and to simulate the new features listed here. The steady-state solution was verified by comparing the results with the Pascal design specifications.

The second significant modification of the PDC for simulation of the direct Pascal reactor concept was an introduction of the reactor shutdown heat removal system (SHRS). Because Pascal uses direct sCO_2 cycle for normal heat removal, heat removal for the decay power is also implemented using an sCO_2 loop. The Pascal SHRS loop consist of a water-cooled SHR heat exchanger (SHRHX), CO_2 circulating pump, isolating valves, and connecting piping. This SHRS loop is parallel to the main power cycle and is usually isolated during a normal operation. In decay heat removal mode, the main cycle is isolated instead, and the SHRS loop and its components are activated. As a CO_2 water cooler is already implemented in the PDC, no code modifications were needed to simulate this component. Several design options for SHRHX were investigated and simulated in the PDC. For the SHRS pump, the general pump model was extended in PDC from the steady state calculations to the transients. This general pump model bypasses the design and performance subroutines. Instead, a user-specified efficiency is used to calculate the pump performance in both steady state and now during transients. For the pump flow control options in transients, the general pump control model was introduced to PDC. This control model allows for two options: direct flow control or pump driving head control. Both options can be used in either manual or automatic mode. For automatic regime, the same PID approach used for other PDC controls is implemented for the general pump model. In the SHRS simulation, the pump control was set up for automatic flow control to maintain the reactor-outlet temperature during the decay heat removal mode. The final code modification for the

SHRS implementation was an introduction of a concept of isolation valves. Any valve can be designated as an isolation valve, and for this valve a list of cycle pipes which this valve isolates is provided in the input. Then, when the isolation valve is fully closed, the flow is assumed to stop in all pipes in the valve's list, and further transient calculations for these pipes and cycle components connected to these pipes are bypassed. The concept of isolation valves was used in simulation of the SHRS activation transients, where the main cycle is isolated and the PDC transient calculations for the cycle components, such as turbine and compressors, is no longer needed.

Although it was not originally planned, during the work on simulating the Pascal concept several additional needs for the PDC modifications were realized. These are mostly related to changing the approach in the PDC implementation of Brayton cycle control logic to reflect specifics of the Pascal concept. The first change was the possibility to define multiple control mechanisms for some types of PDC controls. For example, the multi-shaft turbomachinery arrangement required three independent turbine throttling control mechanisms, one for the power turbine, and one for each of the two drive turbines. Likewise, the presence of the main cycle cooler along with the SHRS heat exchanger necessitated two independent controls for water flow rate for these coolers. Therefore, the control logic, including the input file, was modified to allow arrays of inputs for multiple controls in the same category. To facilitate this implementation, the concept of control tables was also introduced. A user can now define a control table (for example, target shaft speed versus time or load) independently of the control or type of control, and then a table can be used for one or multiple control mechanisms. Similarly, the manual valve control action was decoupled from the control types, and such manual control action can now be specified for any valve used in the model, in a separate section of the Brayton cycle control input file. The last change in the cycle control input was incorporation of the isolation valve concept implemented for simulation of the SHRS system.

In addition to these changes, several other code improvements were introduced in this work. These include the ability to provide a reactor power table in a separate file and provision for a user's selection of the PDC model to use in calculations.

The PDC graphical user interface was also modified to reflect all changes implemented in the code and the input files, including a new reactor input form, a new option for a general compressor type, modifications in the Brayton cycle control logic, and the model selection.

To demonstrate the newly implemented code features, two sets of transient simulations were carried out. The first transient simulated the activation of the SHRS and transition from the main cycle to the SHRS operation mode. The main cycle isolation valves were closed and the SHRS valves were opened. Both the SHRS pump and SHRHX water pump were started up and then transitioned to the automatic control operating mode according to the Pascal SHRS setup. The code showed the ability to handle such significant changes in the system operation, including a brief short-term period where neither the main nor SHRS loop provided coolant circulation in the reactor. In the long term, the heat loss to the moderator in the reactor has been demonstrated to show significant effect on the SHRS operation after the decay heat generation dropped below the heat loss level.

The second set of the demonstration transients included simulation of normal operation during load following. The electrical grid demand was postulated to change at 5%/min rate, and different control actions were simulated and implemented to match the changing grid demand.

The load following by the turbine bypass and compressor speed controls were successfully demonstrated. Due to specifics of the Pascal design, the implementation of the inventory control was shown to be problematic, as the two-phase conditions at the low temperature compressor inlet could not be avoided. However, combinations of either the turbine bypass or compressor speed controls with the inventory control were demonstrated to evade this problem by maintaining the compressor inlet pressure above the critical value. These combinations also showed benefits over the base controls by increasing plant efficiency at partial loads. In all simulated transients, the dynamic response of the newly developed reactor module was as expected without any issues or problems.

Acknowledgements

Argonne National Laboratory's work was supported by the U. S. Department of Energy under Prime Contract No. DE-AC02-06CH11357 between the U.S. Department of Energy and UChicago Argonne, LLC. The work presented in this report was funded by the US DOE Technology Commercialization Fund with 50% cost share by TerraPower. The author is grateful to Robert Pertoski and the entire TerraPower Pascal team for their support and guidance during the project, as well as for providing the Pascal design information.

References

1. Moisseytsev, A., Sienicki, J. J., "PDC: Plant Dynamics Code for Design and Transient Analysis of Supercritical Brayton Cycles," ANL-ART-154, Argonne National Laboratory, September 30, 2018.
2. Moisseytsev, A. and Sienicki, J. J., "Analysis of Thermal Transients for sCO_2 Brayton Cycle Heat Exchangers," GT2019-90374, ASME Turbo Expo 2019, GT2019, Phoenix, AZ, USA, June 17-21, 2019.
3. Moisseytsev, A. and Sienicki, J. J., "Dynamic Control Analysis of the AFR-100 SMR SFR with a Supercritical CO_2 Cycle and Dry Air Cooling Part I: Plant Control Optimization", Proceedings of 26th International Conference on Nuclear Engineering, ICONE-26, London, England, July 22-26, 2018.
4. Moisseytsev, A. and Sienicki, J. J., "Dynamic Control Analysis of the AFR-100 SMR SFR with a Supercritical CO_2 Cycle and Dry Air Cooling Part II: Plant Control under Varying Ambient Conditions", Proceedings of 26th International Conference on Nuclear Engineering, ICONE-26, London, England, July 22-26, 2018.
5. Moisseytsev, A. and Sienicki, J. J., "Simulation of IST Turbomachinery Power-Neutral Tests with the ANL Plant Dynamics Code," 6th International Symposium - Supercritical CO_2 Power Cycles, Pittsburgh, PA, March 27-29, 2018.
6. Petroski, R., et al., "Design of a direct-cycle supercritical CO_2 nuclear reactor with heavy water moderation," Nuclear Engineering and Technology, 2021, ISSN 1738-5733, https://doi.org/10.1016/j.net.2021.09.030.
7. Muto, Y., Kato, Y., "Optimal Cycle Scheme of Direct Cycle Supercritical CO_2 Gas Turbine for Nuclear Power Generation Systems," International Conference on Power Engineering, Hangzhou, China, 2007.
8. Moisseytsev, A. and Sienicki, J. J., "Lessons Learned and Improvements in ANL Plant Dynamics Code Simulation of Experimental S-CO2 Loops," PowerEnergy2015-49182, Proceedings of ASME Power & Energy 2015, San Diego, CA, June 28-July 2, 2015.
9. Moisseytsev, A., "Challenges and Results of Validating of the ANL Plant Dynamics Code with the SNL S-CO_2 Loop Experimental Data," Presentation at ASME 2014 Verification and Validation Symposium, Las Vegas, NV, May 7-9, 2014.
10. Moisseytsev, A., "PDC Modifications for Analysis of Gas-Cooled Reactors with Direct Helium Brayton Cycle," ANL/NSE-21/31, Argonne National Laboratory, May 9, 2022.
11. Moisseytsev, A., and Filippone, C., "Load Following Analysis of the Holos-Quad 10MWe Micro-Reactor with Plant Dynamics Code," ANL/NSE-21/32, Argonne National Laboratory, May 9, 2022.
12. Carbajo, J.J., Yoder, L.Y., Popov, S. G., Ivanov, V.K., "A Review of the Thermophysical Properties of MOX and UO_2 Fuels," Journal of Nuclear Materials, 299 (2001), pp. 181-198.

13. Moisseytsev, A. and Sienicki, J. J., "Controllability of the Supercritical Carbon Dioxide Brayton Cycle Near the Critical Point," 2008 International Congress on Advances in Nuclear Power Plants (ICAPP 2008), Anaheim, CA, June 8-13, 2008, paper 8203.

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC