Development of a DLL-Based MARS-KS/MATLAB Coupling Framework for Integrated Load-Following Analysis of a PWR Small Modular Reactor

Gyudong Kim [a], Takhyun Chun [a], Jeong Ik Lee [a *]
[a]*Department of Nuclear and Quantum Engineering, KAIST 373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea*
[*]*Corresponding author: jeongiklee@kaist.ac.kr*

Keywords : MARS-KS, MATLAB, DLL coupling, load-following, small modular reactor, Rankine cycle

1. Introduction

The increasing penetration of variable renewable energy and the diversification of future electricity systems are strengthening the demand for flexible low carbon generation. Nuclear power plants have traditionally been operated primarily as baseload units; however, regular load-following operation becomes increasingly important in systems with high nuclear shares or rapidly varying renewable generation. Since a load maneuver changes reactor power, coolant temperatures, component thermal conditions, and control-system responses simultaneously, the achievable load-change rate and the associated operating margins of a small modular reactor (SMR) should be evaluated through dynamic simulation before the plant and its control strategy are finalized [1].

This requirement is particularly important for soluble-boron-free pressurized-water SMRs. Soluble-boron-free core concepts use advanced burnable absorbers and lattice optimization to control excess reactivity while maintaining a sufficiently negative moderator temperature coefficient (MTC) [2],[3]. Under this condition, a change in secondary-side steam-generator heat removal modifies the primary coolant inlet and average temperatures, thereby producing a reactivity feedback and a corresponding change in core power. Previous studies have shown that this thermal feedback can support passive or autonomous power maneuvering without soluble-boron concentration adjustment [2],[4].

The governing feedback loop spans both the reactor coolant system and the balance-of-plant (BOP). Secondary side mass flow rate, pressure, and temperature determine the steam generator heat removal rate and thus affect primary coolant temperature and core reactivity. Conversely, reactor thermal power and primary side thermal-hydraulic conditions determine the steam state and available heat input to the power conversion system. Prescribing the secondary boundary independently of the reactor response can therefore break the closed interaction among steam-generator heat transfer, core average temperature, reactivity, reactor power, and turbine-generator output. An integrated primary/secondary transient calculation is required to evaluate both load-following capability and coupled control stability.

MARS-KS provides a suitable basis for the reactor-side calculation since it combines one-dimensional thermal-hydraulic models with reactor kinetics [5]. Its built-in turbine and regenerative-cycle representation, however, is not intended to reproduce a detailed off-design steam Rankine cycle. A previous MARS-KS secondary-side study reported an inconsistency in the original turbine-work formulation for compressible steam, for which density changes substantially across the turbine. The study consequently introduced a pre-generated turbine map and an additional model to determine pressure, temperature, flow rate, and efficiency [6]. It also identified difficulties in matching design pressure and flow since MARS-KS explicitly accumulates losses in pipes and volumes, branch-flow errors can become significant in the low-pressure turbine system, and feedwater heating or reheating may require surrogate heat structures when dedicated power-cycle components are unavailable. Differences of up to approximately 15% were attributed to these secondary-system modeling limitations [6].

A practical alternative is to retain the reactor-system capabilities of MARS-KS while assigning the detailed power conversion calculation to an external solver. Previous studies have coupled independently developed analysis codes to MARS while preserving their original calculation sequences and input/output structures, including MARS-KS/FRAPTRAN coupling and MARS/containment-code coupling using a dynamic-link library (DLL) [7],[8]. The present study therefore develops a two-way DLL-based coupling framework between Fortran-based MARS-KS 2.0 and a MATLAB transient Rankine cycle solver. The external model represents a grid connected and droop controlled turbine-generator system, turbine-valve protection logic, feedwater pump dynamics, and condenser pressure control. The objective is to provide consistent primary/secondary boundary exchange and accepted-time-step synchronization while retaining detailed off-design power cycle behavior.

2. MARS-KS and MATLAB Transient Analysis Models

2.1 MARS-KS nodalization and MATLAB transient analysis models

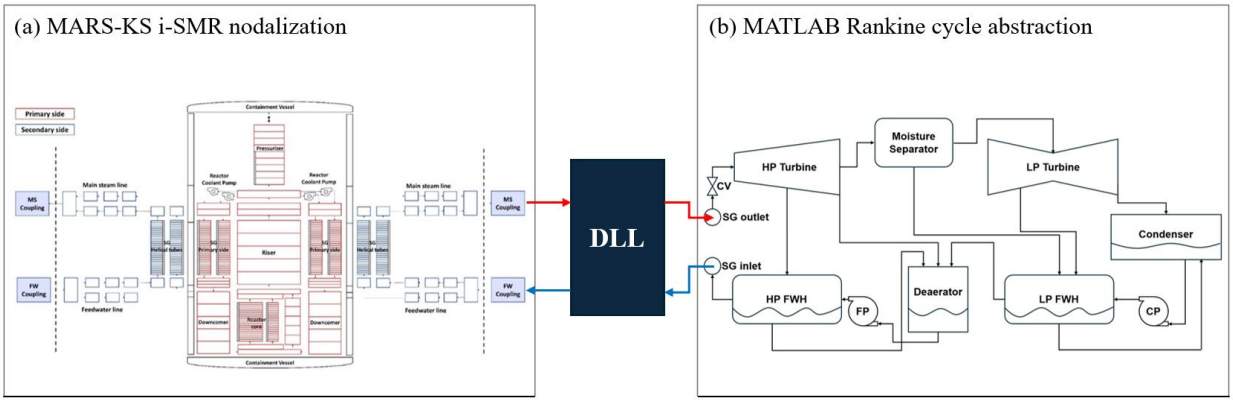

Fig. 1 System models used in the coupled analysis. (a) i-SMR MARS-KS nodalization adapted from Joo and Lee [11], (b) intentionally simplified MATLAB Rankine cycle configuration

Fig.1(a) summarizes the two system models and their coupling boundaries. The MARS-KS primary side model uses the same i-SMR input deck and nodalization as Joo and Lee [11]. The reactor coolant system is represented by the reactor vessel, pressurizer, reactor coolant pumps, and steam-generator primary channels, while one-dimensional steam generator, main steam, and feedwater paths represent the secondary side. Time-dependent volumes and junctions define the secondary interfaces [11]. The schematic diagram in Fig. 1(a) is used to explain the coupling locations; detailed component-specific dimensions and proprietary design information are omitted from the presentation, while the computational input retains the original nodalization used for the analysis.

The external MATLAB model represents the power conversion system as a steam Rankine cycle containing a high pressure turbine, a moisture separator, a low-pressure turbine, extraction and drain paths, a deaerator, high- and low pressure feedwater heaters, a condenser, and two pumps. The arrangement shown in Fig. 1(b) is intentionally simplified to emphasize the thermodynamic and control connections relevant to this study and should not be interpreted as the final detailed BOP layout of the i-SMR.

2.2 Lumped-parameter transient Rankine model

The MATLAB transient solver adopts a lumped-parameter control-volume formulation. Components with relevant hold-up, including the condenser, deaerator, and feedwater heaters, are represented by mass and energy storage equations, whereas turbines, pumps, valves, and heat-exchanger performance are evaluated through algebraic relations. The generic control-volume balances used in the model are given by Eqs. (1) and (2).

$$\frac{dM}{dt} = \sum \dot{m}_{in} - \sum \dot{m}_{out} \quad (1),$$

$$\frac{dU}{dt} = \sum \dot{m}_{in} h_{in} - \sum \dot{m}_{out} h_{out} \quad (2).$$

At each right-hand-side evaluation, the current states and actuator positions are passed to the steady/off-design cycle network, which evaluates thermodynamic properties and algebraic component relations. These relations follow the Rankine cycle modeling approach of Kim et al. [16]. The derivatives are returned to ode15s, giving a semi-explicit ODE/DAE-style formulation in which REFPROP and component-performance relations act as algebraic constraints. ode15s was selected for the stiff differential and differential-algebraic system [13].

For each coupling interval, the MATLAB calculation reads the requested net-output command, generates a smooth load reference, evaluates the current cycle network and properties, calculates the grid-connected governor demand, applies the protection branches and low-value gate, evaluates the control-valve servo and turbine/pump power, and integrates the condenser, deaerator, feedwater-heater, feedwater-pump, and actuator states to the MARS-KS trial time.

2.3 Grid-connected droop governor and turbine-valve control

Only grid-connected operation is considered in this study. Grid frequency fixes turbine-generator speed, so the shaft-speed equation is not integrated and the control valves regulate load rather than speed. The speed/load-changer reference follows the demanded net output, and the droop governor provides the normal load-control branch [14],[15]. In Eqs. (3)-(5), L_{pu} is normalized net-output demand, $N_{ref,pu}$ and $N_{grid,pu}$ are the per-unit reference and grid speeds, CV_{gov} is governor valve demand, and R is permanent droop; R = 0.05 was used.

$$L_{pu} = \frac{P_{net,demand}}{P_{net,design}} \quad (3)$$

$$N_{ref,pu} = 1 + R \cdot L_{pu} \quad (4)$$

$$CV_{gov} = \frac{N_{ref,pu} - N_{grid,pu}}{R} \quad (5)$$

The final control-valve command is the smallest demand from the droop governor, load limit, steam-pressure regulator, and condenser-vacuum runback branches, as expressed by the low-value gate in Eq. (6) [12]. Eq. (7) converts this command to valve motion using a first-order hydraulic servo with opening and closing rate limits.

$$CV_{cmd} = min\left(CV_{gov}, CV_{load-limit}, CV_{pressure}, CV_{cond,vacuum}\right) \quad (6)$$

$$\frac{dCV_{open}}{dt} = \frac{CV_{cmd} - CV_{open}}{\tau_{CV}} \quad (7)$$

2.4 Auxiliary BOP controls and dynamic load

The auxiliary controls include a PI control loop for regulating the steam-generator inlet temperature through the top feedwater heater, together with a feedwater-pump actuator. A portion of the superheated steam is bypassed to the hot side of the top feedwater heater, and the bypass steam flow is modulated to maintain the steam-generator inlet temperature at its target value. The commanded bypass steam mass flow follows Eq. (8). In addition, the feedwater-pump mass-flow command is set proportional to the turbine steam-flow demand, while the actual pump flow follows the first-order response given by Eq. (9).

$$\dot{m}_{TFWH,cmd} = \dot{m}_{TFWH} + K_p\left(T_{target} - T_{actual}\right) + K_i I_{TFWH} \quad (8)$$

$$\frac{d\dot{m}_{FP}}{dt} = \frac{\dot{m}_{FP,cmd} - \dot{m}_{FP}}{\tau_{FP}} \quad (9)$$

Condenser pressure is calculated from a fixed-volume saturated-mixture inventory. Cooling-water flow is pressure-controlled, and marginal-vacuum and trip thresholds can restrict the turbine valves through the vacuum branch of the low-value gate. The mass, energy, and epsilon-NTU relations are given by Eqs. (10)-(12).

$$\frac{d\dot{m}_{cond}}{dt} = \frac{\dot{m}_{cond,cmd} - \dot{m}_{HPT1}}{\tau_{cond}} \quad (10)$$

$$\frac{dU_{cond}}{dt} = \frac{U_{cond,cmd} - U_{HPT1}}{\tau_{cond}} \quad (11)$$

$$Q_{cond} = \dot{m}_{cw} c_p \left(T_{sat}(P_{cond}) - T_{cw,in}\right)\left(1 - exp\left(\frac{-UA}{\dot{m}_{cw} c_p}\right)\right) \quad (12)$$

3. DLL-Based MARS-KS/MATLAB Coupling Methodology

3.1 Coupling architecture and software layers

The implemented coupling framework uses a run-time DLL architecture in which MARS-KS remains the master solver and controls the global transient time. The principal software elements and exported interface routines are summarized in Table I.

At run time, the Fortran code loads matcouple.dll and resolves MatInit, MatPredict, MatFeedback, MatCommit, and MatFinalize using LoadLibrary and GetProcAddress. The C++ wrapper initializes the MATLAB Runtime and calls marsmatcore.dll, a shared library generated using MATLAB Compiler SDK. This avoids static linkage and follows standard Windows run-time dynamic linking [8]-[10]. Figure 2 summarizes the software hierarchy and bidirectional data paths. Rankine cycle parameters from the accepted state are reused as the initial estimate of the next Rankine cycle calculation rather than being reinitialized at every call.

Table. 1: Roles of the principal coupling modules and run-time interface routines

Element	Stage/layer	Primary role
LINK_MATLAB	Fortran data	Stores coupling mode, TMDPVOL/TMDPJUN and feedback maps, DLL handles, procedure pointers, trial metadata, and C-interoperable buffers.
MatlabCoupleM	Fortran control	Parses 230xxxxx cards, loads the DLL, resolves exports, and dispatches prediction, feedback, commit, and finalization.
MatInit	Initialization	Initializes the external backend and registers volume/junction mapping data during MARS-KS input processing.
MatPredict	Pre-HydroSolve	Evaluates the MATLAB trial state at t(n+1) and returns TMDPVOL/TMDPJUN boundary values.
MatFeedback	Post-HydroSolve	Transfers accepted MARS-KS junction flow and volume pressure/temperature to the external solver.
MatCommit	Accepted step	Promotes the trial MATLAB state to the accepted state and records history, skipped for rejected or repeated steps.
MatFinalize	Termination	Terminates the MATLAB backend and releases run-time resources.
MATLAB gateway/ state/solver	External solver	-mars_boundary_law.m routes actions -mars_rankine_state.m manages trial/accepted states -iSMR_MARS_coupling_core.m aggregates feedback and solves the Rankine cycle

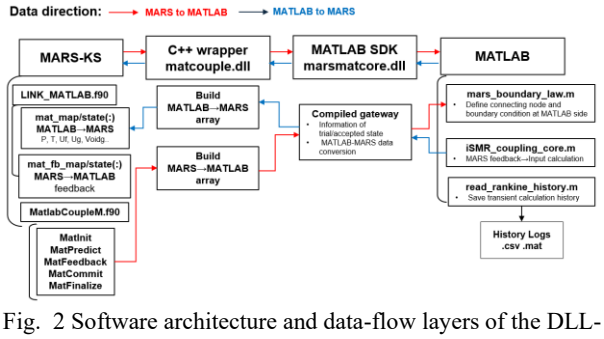

Fig. 2 Software architecture and data-flow layers of the DLL-based MARS-KS/MATLAB coupling framework.

3.2 Two-way data exchange and accepted-time-step synchronization

Fig. 3(a) summarizes the physical interfaces. Main-steam junction flows between the steam-generator

outlets and headers are summed, while corresponding header vapor pressures and temperatures are averaged, to define the control-valve-inlet mass flow, pressure, and temperature supplied to MATLAB. Pressure is converted from Pa to kPa, vapor temperature is used for the steam state, and flow signs follow the main-steam direction.

Conversely, Rankine cycle feedwater-heater outlet mass flow, pressure, and temperature are supplied to the MARS-KS steam generator feedwater inlet header, as shown in Fig. 3(a). Pressure and temperature override the mapped TMDPVOL state, and mass flow overrides the TMDPJUN state. Physical interface names are used so the method is independent of nodal identifiers.

Fig. 3(b) shows accepted-time-step synchronization. MatlabPredict generates the t(n+1) trial boundary. SetBoundaryProperty first evaluates the native time-dependent boundary and then applies MATLAB TMDPVOL/TMDPJUN overrides before property updating. HydroSolve advances the native hydrodynamic conservation equations and updates pressures, phasic velocities, and volume/junction properties under MARS-KS time-step and convergence control [17]. For an accepted step, MatlabCollectFeedback reads the new-time state, MatFeedback transfers it, and MatCommit promotes the trial MATLAB state. Rejected or repeated steps are not committed.

The implementation uses weak two-way coupling. The MARS-KS state accepted at $t(n)$ determines the MATLAB boundary at $t(n+1)$, without an inner fixed-point iteration. This minimizes modifications to the native MARS-KS convergence sequence. A strong-coupling iteration can be introduced if the one-step lag becomes significant during rapid transients.

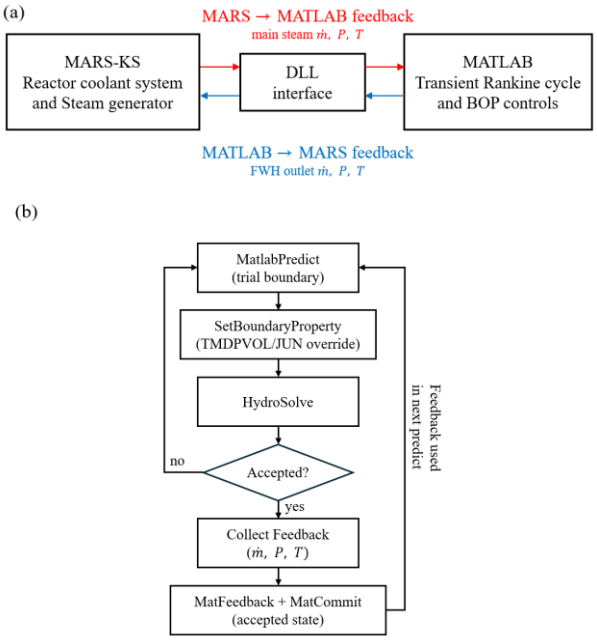

Fig. 3. Two-way coupling implementation: (a) physical boundary and feedback interfaces; (b) accepted-time-step sequence of the DLL-based weak coupling.

4. Results and Discussion

4.1 Verification of Bidirectional Data Transfer

Before applying the coupling framework to the complete i-SMR and Rankine cycle models, a reduced smoke-test nodalization was constructed to verify the bidirectional data-transfer functions independently of the plant models. As shown in Fig. 4, the MARS-KS test model consisted of a source time-dependent volume, TMDPVOL 100; a MATLAB-controlled time-dependent junction, TMDPJUN 101; an intermediate single volume, SNGLVOL 200; an outlet single junction, SNGLJUN 201; and a sink time-dependent volume, TMDPVOL 300. The source and sink temperatures were maintained at 300 K, while the sink pressure was fixed at approximately 101 kPa.

The MATLAB test function generated sinusoidal pressure and mass-flow signals as

$$P = 150 + 2\sin(8\pi t)\ [kPa] \quad (13)$$

$$\dot{m} = 0.06 + 0.02\sin(8\pi t)\ [kg/s]. \quad (14)$$

The calculated pressure was transferred through the DLL layer and applied to TMDPVOL 100, while the calculated mass flow was applied to TMDPJUN 101. After each accepted MARS-KS hydraulic step, the pressure calculated in SNGLVOL 200 and the mass flow calculated at SNGLJUN 201 were collected from the new-time MARS-KS state and returned to the MATLAB layer. MATLAB then stored these values for comparison with the corresponding MARS-KS outputs. This configuration therefore tested MATLAB to MARS transfer through the inlet boundary and MARS to MATLAB transfer through the downstream volume and junction within a single calculation.

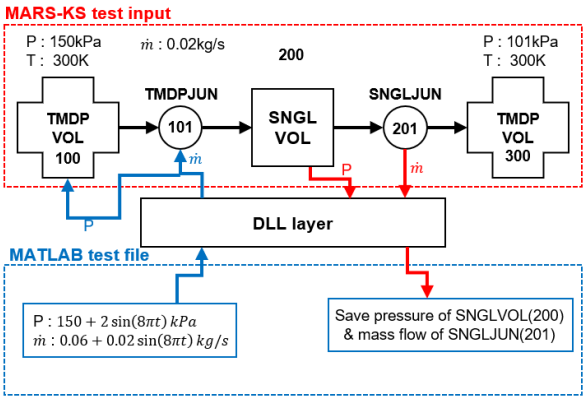

Fig. 4 Test configuration for the bidirectional MARS-KS/MATLAB coupling smoke test.

Figs. 5(a) and 5(b) show the pressure applied to TMDPVOL 100 and the mass flow applied to TMDPJUN 101, respectively. The MARS-KS boundary values reproduced the prescribed MATLAB signals

without observable attenuation, phase shift, or waveform distortion. The MATLAB-generated and MARS-KS-applied values were nearly identical and consequently overlap in the plots; therefore, only the red curves representing the MATLAB-to-MARS transfer are visibly presented.

Figs. 5(c) and 5(d) present the MARS-KS pressure response in SNGLVOL 200 and the mass flow response at SNGLJUN 201, respectively. The initial pressure and flow excursions correspond to the hydraulic adjustment of the intermediate volume from its initial condition to the imposed upstream boundary and downstream sink conditions. After this short initialization period, the pressure reached a periodic response and the outlet mass flow closely followed the imposed inlet-flow waveform. These variations represent the physical response calculated by MARS-KS and should not be interpreted as data-transfer errors. The values recorded in MATLAB overlapped the corresponding MARS-KS results; therefore, only the blue MARS-to-MATLAB curves are shown.

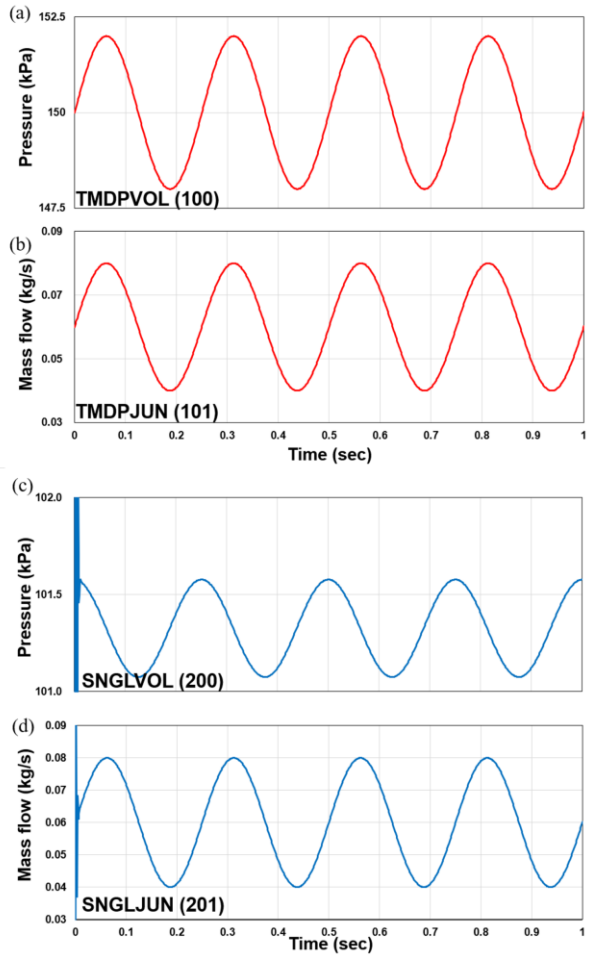

Fig. 5 Bidirectional data-transfer results of the MARS-KS/MATLAB smoke test: (a) pressure applied to TMDPVOL 100, (b) mass flow applied to TMDPJUN 101, (c) pressure returned from SNGLVOL 200, and (d) mass flow returned from SNGLJUN 201.

Table 2 summarizes the maximum relative transfer errors obtained during the smoke test. For the MATLAB-to-MARS direction, the maximum errors were 0.00034% for the pressure of TMDPVOL 100 and 0.000497% for the mass flow of TMDPJUN 101. For the MARS-to-MATLAB direction, no difference was detected, at the reported numerical precision, between the MARS-KS values and the values stored by MATLAB for the pressure of SNGLVOL 200 and the mass flow of SNGLJUN 201. The small residual errors in the MATLAB-to-MARS direction were less than 5×10^{-4} and were negligible relative to the amplitudes of the imposed signals.

The smoke-test results confirm that the developed interface correctly maps and transfers pressure and mass-flow data in both directions and that the transferred values are applied to or collected from the intended MARS-KS components. The test also confirms that the DLL communication and boundary-override procedures do not introduce an appreciable numerical distortion into the transferred signals. This verification is limited to the coupling and data-transfer implementation; the thermodynamic accuracy and load-following performance of the complete MARS-KS/MATLAB model are evaluated separately using the integrated plant calculation.

Table. 2 Maximum relative errors in the bidirectional data transfer

Transfer direction	MATLAB to MARS		MARS to MATLAB	
ID	100	101	200	201
Property	Pressure [kPa]	Mass flow [kg/s]	Pressure [kPa]	Mass flow [kg/s]
Maximum transfer error(%)	0.00034	0.000497	0	0

4.2 Preliminary Coupled Load-Following Response

Fig. 6 presents the integrated response of the coupled MARS-KS/MATLAB model during a representative 100–50–100% load-following maneuver. To protect proprietary design information, the variables are reported in normalized form using their respective reference values. The requested net output was maintained at 100% until 500 s, decreased linearly to 50% between 500 and 1,100 s, maintained at 50% until 1,600 s, and then increased to 100% between 1,600 and 2,200 s. The corresponding load-change rate was approximately 5%/min during both the load-reduction and load-recovery periods.

Fig. 6(a) compares the requested net output with the MATLAB-calculated net output and the MARS-KS reactor power. During the load-reduction period, both calculated responses followed the overall trend of the requested output. The reactor power approached

approximately 50% near the end of the downward ramp and then increased slightly during the low-load holding period. The MATLAB net output temporarily decreased below the requested value, reaching approximately 46%, before gradually recovering toward the target. This difference reflects the additional influence of turbine-valve dynamics, component inventories, feedwater-system behavior, and transient power-conversion efficiency on the electrical output.

During the load-recovery period, the reactor power increased smoothly toward the requested value, whereas the MATLAB net output exhibited a more pronounced delay near the upper end of the ramp. Both quantities eventually returned close to their high-load reference values. No discontinuity or progressively growing oscillation was observed during either load reversal, suggesting that the accepted-time-step synchronization operated without observable numerical instability during the analyzed maneuver.

Fig. 6(b) shows the normalized steam-generator inlet and outlet mass-flow rates. The two histories remained nearly coincident throughout the transient and followed the imposed load variation. During the downward ramp, both flow rates decreased to approximately 48%, with only a small temporary difference near the completion of the ramp. During load recovery, the two flow rates increased together, slightly exceeded their high-load reference values, and subsequently returned toward the original condition. The absence of a sustained divergence between the inlet- and outlet-flow histories indicates that the feedwater boundary calculated by MATLAB and the main-steam response calculated by MARS-KS remained mutually consistent throughout the coupled calculation.

The primary-side coolant-temperature response is shown in Fig. 6(c). As the load decreased, the core inlet temperature increased from approximately 89% to 94% of its reference value, and the core-average temperature increased from approximately 94.5% to 96.5%. In contrast, the core outlet temperature decreased only slightly and remained close to its reference value. During load recovery, the inlet and average temperatures decreased toward their initial levels, while the outlet temperature recovered. This inverse relationship between reactor power and the core inlet and average temperatures is consistent with the negative-MTC feedback mechanism of a soluble-boron-free PWR-SMR. An increase in moderator temperature introduces negative reactivity and supports passive power reduction, whereas a decrease in moderator temperature during load recovery provides a positive reactivity contribution [2],[4].

Fig. 6(d) presents the steam-generator inlet temperature and top feedwater-heater bypass mass flow calculated by the MATLAB model. The steam-generator inlet temperature decreased with the load reduction, remained near its low-load value during the holding period, and returned toward the high-load condition during the upward ramp. The top feedwater-heater bypass remained inactive during the initial load-reduction period and began to increase after the low-load condition was reached. It reached its maximum value during the load-recovery period and subsequently decreased during post-ramp settling. This response indicates that the auxiliary feedwater-heating control participates mainly in recovering and regulating the steam-generator inlet condition after the low-load transition. The delayed and gradual bypass response also contributes to the difference between the reactor-power and net-output trajectories shown in Fig. 6(a).

Fig. 6(e) shows the reactor-coolant-system mass flow, primary-system pressurizer pressure, and secondary-side steam-generator outlet pressure. The reactor-coolant-system mass flow and pressurizer pressure remained close to their reference values, with deviations generally limited to approximately 1% over the analyzed transient. In contrast, the steam-generator outlet pressure exhibited a larger and repeatable variation, increasing during load reduction and decreasing again during load recovery. This behavior reflects the secondary-side pressure response to changes in steam-flow demand and turbine-valve operation, while the principal primary-side hydraulic variables remained comparatively well regulated.

The corresponding turbine responses are presented in Fig. 6(f). The high-pressure-turbine (HPT) and low-pressure-turbine (LPT) work were individually normalized by their respective high-load reference values. Both decreased as the requested load was reduced; however, the low-pressure-turbine work reached a lower normalized value of approximately 46%, while the HPT work decreased to approximately 49% and then gradually increased during the low-load holding period. This difference reflects the distinct inlet states, pressure ratios, extraction flows, moisture-separator conditions, and stage mass flows experienced by the two turbine sections in the MATLAB Rankine-cycle model.

During load recovery, the LPT work increased relatively smoothly toward its reference value, whereas the HPT work exhibited a more nonlinear response. This difference is primarily associated with the governing-stage response of the HPT to control-valve (CV) motion. As the load is restored, the governor requires relatively large CV adjustments to recover the steam admission and downstream turbine conditions. Since the CV directly regulates the steam admitted to the HPT governing stage, these changes in valve opening alter the throttling loss, admission condition, and governing-stage efficiency [12]. Consequently, the HPT work exhibits a nonlinear response during the load-recovery and post-ramp periods. In contrast, the LPT is not directly governed by the CV and receives steam after expansion through the HPT and intermediate cycle components, resulting in a comparatively smoother work response. The pronounced HPT governing-stage response therefore contributes to the delayed and nonuniform recovery of the MATLAB-calculated net output shown in Fig. 6(a).

Taken together, the results demonstrate the need for an integrated primary- and secondary-system transient

calculation. Reactor thermal power is governed primarily by steam-generator heat removal and moderator-temperature feedback, whereas net electrical output is additionally affected by turbine-section dynamics, power-cycle inventories, feedwater heating, valve response, and component time constants. The temporary mismatch between reactor power and net output, the core-temperature feedback in Fig. 6(c), the feedwater-heater bypass response in Fig. 6(d), and the different HPT and LPT work trajectories in Fig. 6(f) would not be reproduced if either reactor power or the secondary-side boundary were prescribed independently.

The present results should be interpreted primarily as a verification of the developed coupling framework and a preliminary demonstration of integrated load-following analysis, rather than as a quantitative prediction of the final i-SMR transient performance. In the current MATLAB model, several component inventories and hold-up parameters, including those associated with the condenser, deaerator, and feedwater heaters, were assigned representative values since detailed design information was unavailable. Since these quantities directly affect mass and energy accumulation and, consequently, the characteristic response time of the power conversion system, the predicted transient delay, overshoot, and settling behavior may contain corresponding uncertainties. In addition, the present weak-coupling implementation introduces a one-step delay between the accepted MARS-KS state and the subsequent MATLAB boundary prediction. Further validation of the component inventory parameters, together with coupling-interval and time-step sensitivity studies, is therefore required before the framework is used to quantify limiting load-change rates or final i-SMR operating margins.

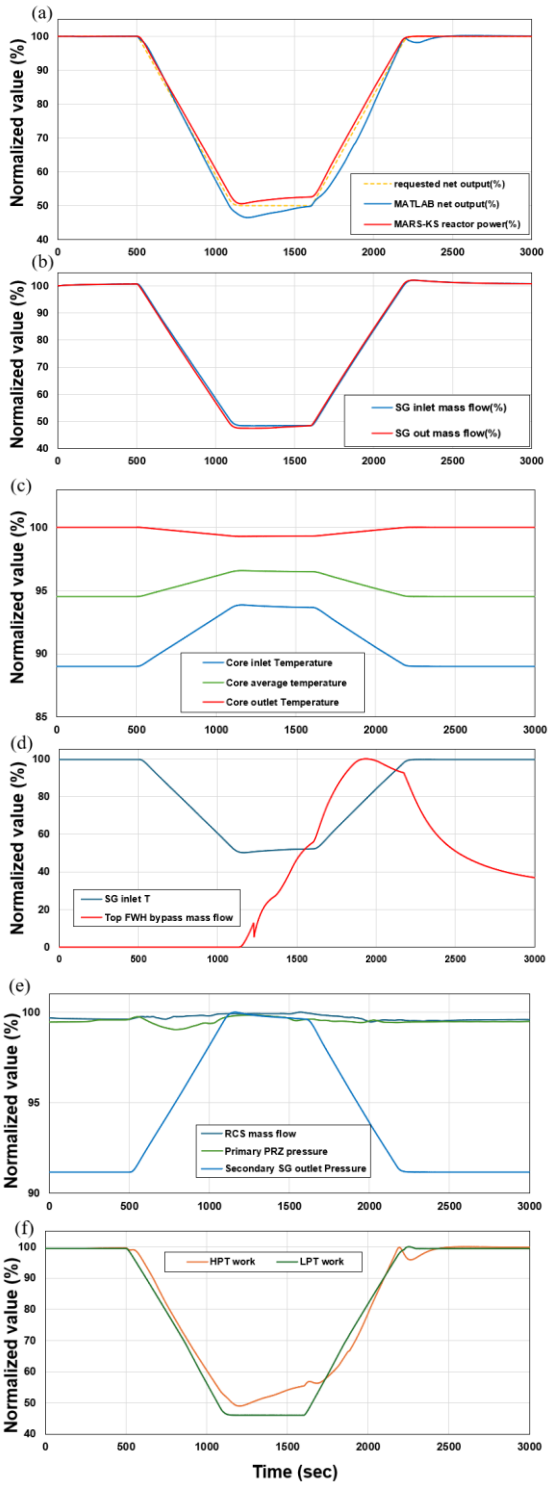

Fig. 6. Normalized primary-system, secondary-system, and power-conversion-system responses of the coupled MARS-KS/MATLAB model during the 100–50–100% load-following maneuver: (a) requested net output, MATLAB-calculated net output, and MARS-KS reactor power; (b) steam-generator inlet and outlet mass-flow rates; (c) core inlet, average, and outlet temperatures; (d) steam-generator inlet temperature and top feedwater-heater bypass mass flow; (e) reactor-coolant-system mass flow, pressurizer pressure, and steam-generator outlet pressure; and (f) high-pressure- and low-pressure-turbine work.

5. Conclusions

A two-way DLL-based coupling framework was developed between MARS-KS 2.0 and a MATLAB transient steam Rankine-cycle solver for integrated PWR-SMR load-following analysis. MARS-KS retains control of the global transient time, hydraulic solution, and time-step acceptance, while the MATLAB solver evaluates the off-design power-conversion system and balance-of-plant controls. The trial/commit strategy advances the MATLAB state only after the corresponding MARS-KS time step has been accepted.

The smoke-test calculation confirmed that pressure and mass-flow data were transferred in both directions and applied at the intended physical interfaces. The maximum MATLAB-to-MARS transfer errors were 0.00034% for pressure and 0.000497% for mass flow, while no difference was detected in the MARS-to-MATLAB direction at the reported numerical precision. These results demonstrate that the DLL communication and boundary-override procedures introduce negligible distortion into the exchanged signals.

The integrated 100–50–100% load-following calculation demonstrated that the developed framework can represent the closed interaction among turbine-generator load demand, feedwater and main-steam conditions, steam-generator heat removal, primary coolant temperature, reactor power, and net electrical output. The reactor power and MATLAB-calculated net output followed the requested load trajectory, although the electrical output showed a larger delay and undershoot because of the additional dynamics of the turbine-generator and BOP systems. The steam-generator inlet and outlet mass-flow rates remained closely matched, while the core inlet and average temperatures varied in the direction expected from negative-MTC feedback.

The reactor-coolant-system mass flow and pressurizer pressure remained close to their reference values, whereas the steam-generator outlet pressure and top feedwater-heater bypass flow responded more substantially to the secondary-side load change. The HPT and LPT work also exhibited distinct part-load and load-recovery responses. In particular, the stronger nonlinearity of the HPT work during load recovery is primarily associated with the control-valve response at the HPT governing stage. As the load is restored, relatively large CV movements are required to recover the steam admission and stabilize the downstream turbine operating conditions. These valve-position changes modify the throttling loss, governing-stage admission condition, and consequently the HPT efficiency and work [12]. In contrast, the LPT is not directly subjected to governing-stage control-valve throttling and therefore exhibits a comparatively smoother recovery. The temporary mismatch between reactor thermal power and net electrical output is thus influenced not only by reactor-side thermal feedback but also by the nonlinear HPT governing-stage response induced by CV motion.

The present study establishes the software architecture, bidirectional data-transfer capability, and functional feasibility of the integrated MARS-KS/MATLAB calculation. It does not yet define the final load-following capability or operating margins of the i-SMR because several component inventories and dynamic parameters were based on assumed or representative values, and the present weak coupling retains a one-step lag. Future work will validate the MATLAB power-cycle model and component inventories against available design or experimental data, quantify mass- and energy-balance errors, assess time-step and coupling-interval sensitivities, reduce the computational cost associated with thermodynamic-property and diagnostic calculations, and evaluate whether an inner strong-coupling iteration is required for more rapid transients.

Acknowledgements

This work was supported by the Innovative Small Modular Reactor Development Agency grant funded by the Korea Government MCEE (No. RS-2024-00400615)

REFERENCES

[1] OECD Nuclear Energy Agency, Technical and Economic Aspects of Load Following with Nuclear Power Plants, OECD Publishing, Paris, 2021, doi:10.1787/29e7df00-en.
[2] A. A. E. Abdelhameed, X. H. Nguyen, J. I. Lee, and Y. Kim, "Feasibility of passive autonomous frequency control operation in a Soluble-Boron-Free small PWR," Annals of Nuclear Energy, Vol. 116, pp. 319-333, 2018, doi:10.1016/j.anucene.2018.02.036.
[3] X. H. Nguyen, S. Jang, and Y. Kim, "Truly-optimized PWR lattice for innovative soluble-boron-free small modular reactor," Scientific Reports, Vol. 11, Article 12891, 2021, doi:10.1038/s41598-021-92350-5.
[4] Y. Jeong, D. Choi, T. Oh, and Y. Kim, "Load-follow operation capability of soluble boron-free small modular reactor ATOM," Frontiers in Energy Research, Vol. 13, Article 1639569, 2025, doi:10.3389/fenrg.2025.1639569.
[5] MARS Development Team, MARS Code Manual, Volume I: Code Structure, System Models, and Solution Methods, KAERI/TR-2812/2004, Korea Atomic Energy Research Institute, 2004.
[6] D. H. Kim, J. J. Lee, and J. I. Lee, "Development of SMR Secondary side MARS-KS input for Artificial Intelligence Operational Scheme," Transactions of the Korean Nuclear Society Autumn Meeting, Goyang, Korea, Oct. 24-25, 2019.
[7] H. Kim, C. Shin, Y. Yang, and T. Kim, "Development of fully coupled FRAPTRAN with MARS-KS code system for calculation of fuel behavior during LOCA," Proc. TopFuel 2018, Prague, Czech Republic, Sep. 30-Oct. 4, 2018.
[8] B. D. Chung, J. J. Jeong, and W. J. Lee, "MARS 1.3 System Analysis Code Coupling with CONTEMPT4/MOD5/PCCS Containment Analysis Code using Dynamic Link Library," Proceedings of the Korean Nuclear Society Autumn Meeting, Seoul, Korea, Oct. 1998.
[9] MathWorks, "mcc - Compile MATLAB functions for deployment," MATLAB Compiler SDK Documentation, accessed Aug. 19, 2026.

[10] Microsoft, "Using Run-Time Dynamic Linking," Win32 API Documentation, accessed Aug. 19, 2026.
[11] S. Joo and J. I. Lee, "Machine Learning-Based Prediction of Pressurizer Control Performance in i-SMR under Load-Following Conditions," Transactions of the Korean Nuclear Society Autumn Meeting, Changwon, Korea, Oct. 30-31, 2025.
[12] Electric Power Research Institute, Steam Turbine Mechanical Hydraulic Control System - Operation, Inspection, Setup, Troubleshooting, and Maintenance Guide, Revision 1, EPRI Report 1019313, Palo Alto, CA, 2009.
[13] MathWorks, "ode15s - Solve stiff differential equations and DAEs," MATLAB Documentation, accessed Aug. 19, 2026.
[14] Woodward, Governing Fundamentals and Power Management, Manual 26260, 2004.
[15] North American Electric Reliability Corporation, Reliability Guideline: Primary Frequency Control, revised Sep. 20, 2023.
[16] G. Kim, Y. J. Chae, T. Chun, and J. I. Lee, "Small modular reactors (SMRs): Comparison of saturated versus superheated steam Rankine cycle under equal thermodynamic constraints," Nuclear Engineering and Technology, Vol. 58, No. 5, Article 104161, 2026, doi:10.1016/j.net.2026.104161.
[17] B. D. Jeong, et al., MARS Code Manual, Volume III: Programmer's Manual, KAERI/TR-3150/2006, Korea Atomic Energy Research Institute, 2009.