Engineering tool integration for complex system simulation and optimization

Szymon Podlasek[1*], Urszula Jachymczyk[1]
AGH University of Krakow, Poland[1]
podlasek@agh.edu.pl, delhi@student.agh.edu.pl

Abstract: *The integration of engineering support tools is essential for the efficient modeling, simulation, and optimization of complex technical systems. This paper presents a dynamic model of a micro-combined heat and power (mCHP) system, developed to validate the feasibility of integrating various computational environments. The approach leverages modular architectures, enabling seamless data exchange between distinct software platforms, thus supporting both detailed thermodynamic analysis and real-time performance optimization. The flexibility of this approach allows for the inclusion of diverse analytical frameworks, including neural network-based optimization, data-driven control strategies, and alternative programming languages, without being limited to a single computational tool. This adaptability makes the proposed architecture particularly suitable for evolving engineering applications, where rapid prototyping and iterativ e refinement are critical. The study highlights the potential of such integrated environments to enhance the design and operational efficiency of energy systems, providing a scalable foundation for future expansions.*
Keywords*: ENGINEERING TOOL, INTEGRATION ENERGY SYSTEMS, DATA EXCHANGE, SIMULATION, OPTIMIZATION*

1. Introduction

In response to the increasing complexity of energy systems, effective modeling and optimization require the cooperation of various computational tools and simulation environments. Modern engineering approaches increasingly rely on the integration of multiple platforms - from programming languages such as Python to engineering environments like MATLAB, Simulink, or ANSYS, and data analysis tools such as Excel [1-3]. A key aspect of this integration is the ability to exchange data flexibly between system components while maintaining consistency in simulation and control algorithms.

The integration of different computational environments enables the exchange of variables and supports further data analysis, visualization, or optimization. This architecture allows users to benefits of various tools within a single, unified workflow, without being limited to a specific piece of software (Fig.1).

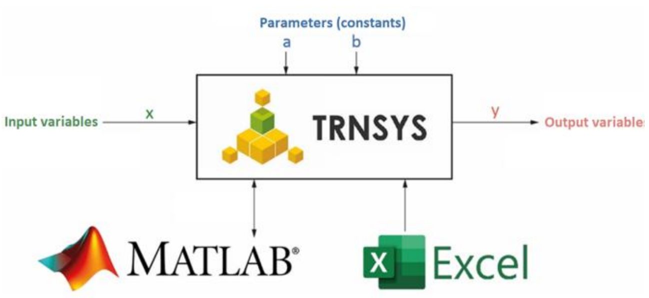

Fig. 1 *Idea diagram showing the possibilities of data exchange between different engineering tools.*

A common approach involves using one tool to simulate the dynamic behavior of the physical system, while a separate environment is responsible for generating control signals based on the simulation results. This setup enables the use of advanced algorithms, including optimization methods and control strategies, to dynamically adjust the system's behavior. Increasingly, neural algorithms such as convolutional neural networks (CNNs) or reinforcement learning (RL) methods are applied for this purpose, successfully supporting event detection and enabling process optimization [4-6].

These methods require real-time access to system data and the ability to quickly iterate and validate results. A modular integration framework also facilitates future system expansion and adaptation to changing operational and technological requirements [7-8]. The use of inter-environment communication supports coupled simulations and the implementation of hybrid optimization strategies, which are widely discussed in recent literature on advanced energy systems [9-11].

2. Model of the Analyzed Object

The dynamic model of the steam generator in the micro-combined heat and power (mCHP) system enables the calculation of thermodynamic parameters of the working fluid in the form of water vapor and its condensate within the Rankine cycle, as well as global system parameters [12, 13]. The system is based on the combustion of heating oil, which provides the energy required to generate steam in the steam generator. The working fluid is first preheated in a tank that also functions as a deaerator, and then further heated by hot flue gases to reach the boiling point at a specified pressure. In the system, the generated steam is first dehydrated and then directed to a steam engine connected to an electric generator. Subsequently, the expanded steam is passed to a condenser and a condensate tank in order to restore the initial parameters of the working fluid at the steam generator inlet and to remove gases dissolved in the condensed water.

The modeling and simulation of the mCHP system were conducted using both built-in components from the software library and user-defined components. The components available in the software library were experimentally validated and/or based on actual operational data, ensuring high reliability of the simulation results. The system model developed in TRNSYS provides dynamic data related to system operation. Temperature profiles, flow rates, power levels, and control functions for all system components are calculated through the simulation.

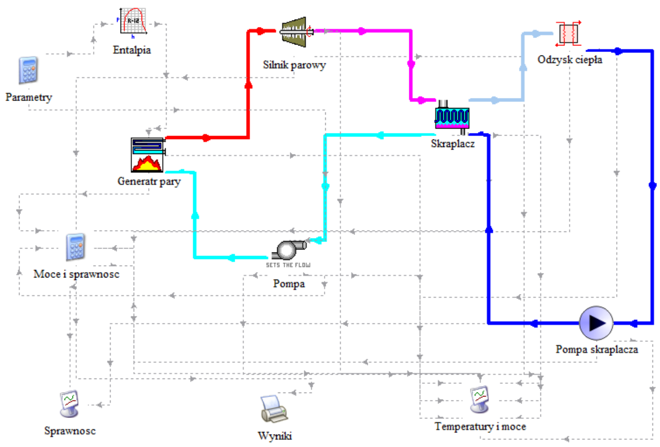

Fig. 2 *A model of mCHP installation in TRNSYS software.*

The developed model of the mCHP installation, based on a fire-tube steam generator and discussed in the earlier sections of this work, consists of the following components (Fig.2):

- Steam generator, producing steam at a specified pressure using heating oil combustion: component Type 638;
- Steam engine with generator, generating electrical energy by expanding steam in pistons: component Type 592a;
- Condenser, removing heat from the expanded working fluid in order to restore the steam parameters at the steam generator inlet: component Type 598;
- Pump, pressurizing the condensed water to the desired pressure level: component Type 597;
- Condenser circulation pump, driving the flow in the condenser loop: component Type 3d;
- Heat recovery, simulating a heat exchanger that extracts thermal energy from the condenser's cooling stream: component Type 5b.

In addition, the mCHP installation model includes supplementary components used to carry out the system simulation and to generate output data. These elements include calculators for data input and processing, as well as printers and plotters. Specifically, two calculators were employed to input the main parameters into the model, perform unit conversions, and calculate efficiency. The model parameters will be presented later in terms of kilowatts (kW), while the equation implemented in the model is defined by the function *eta_ele_cycle = P_TURB_kW/max(Q_BOIL_kW,1)* which corresponds to equation (e1), where the turbine power output N_{sp} is related to the thermal power of the steam generator Q_{wt}:

$$\eta_{kog} = \frac{N_{sp}}{Q_{wt}} \quad \text{(e1)}$$

Additionally, the model includes a component used to calculate thermodynamic parameters based on the specified steam conditions at the outlet of the steam generator. This component, referred to as "Enthalpy," is implemented using Type 58. It is worth noting that the calculation of system efficiency (*eta_ele_cycle*) does not account for the power consumption of the pumps in the steam and condenser loops.

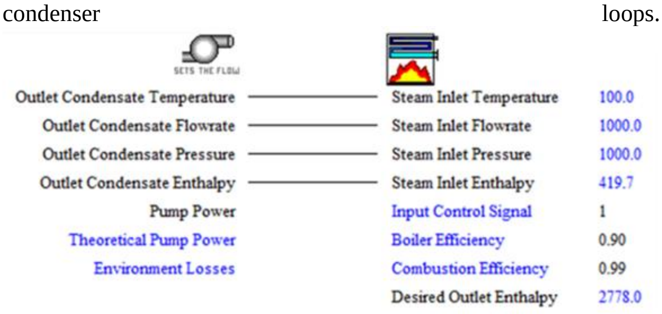

Fig. 3 *Structure of connection and transmission of parameters between blocks in Trnsys environment.*

Figure 3 shows an example of the connection structure used to develop the mCHP system model for the pump and steam generator. Dynamic parameter exchange between model components is observed during the simulation process.

3. Model Parameters and Preliminary Simulation

The system was simulated by setting the duration to 1.0 hour with a time step of 0.01 hour. Under such conditions, the dynamic model was assumed to operate in a static mode to determine the operating conditions at the set conditions. The parameters used for the developed model are shown in Table 1.

The developed model was used to evaluate the performance and efficiency of the presented Rankine cycle that can be achieved under nominal operating conditions. In particular, the model made it possible to determine the power and efficiency of the system depending on the main operating parameters, such as the mass flow of steam and the maximum pressure of the steam system.

To enable data exchange, the model was extended with the Type155 block. The connection scheme be presented in Section 4.

Table 1: *Operating Parameters of the Modeled Cogeneration System.*

Main Parameters of the Steam System Maximum pressure: 10 bar Maximum temperature: 180.0 °C	
Steam Generator	Thermal power: 100.0 kW Generator efficiency: 0.90 Combustion efficiency: 0.99
Condensate Pump	Nominal mass flow rate: 2000 kg/h Power: 0.5 kW Power conversion factor: 0.05
Steam Engine	Nominal power: 15.0 kW Steam outlet pressure: 1.0 bar Isentropic efficiency: 0.75
Heat Recovery	Working fluid: water Inlet temperature: 25.0 °C Flow rate: 1500 kg/h
Condenser	Pinch point temperature difference: 15.0 °C Minimum pressure: 0.5 bar Cooling degree: 3.0 °C
Main Pump	Nominal mass flow rate: 150 kg/h Overall efficiency: 0.60 Motor efficiency: 0.90

The results of the simulation are shown in Figure 4, where the relationship between the flow rate and the system's output and efficiency is depicted. It can be seen from the graphs that maximum power generation is achieved for a given value of flow (150 kg/h), and with a decrease in this value, electrical power decreases sharply. In this range, the power generated is proportional to the mass flow, since the steam temperature at the steam turbine inlet is constant at 185°C. On the other hand, with a flow value greater than 150 kg/h, the temperature at the turbine inlet drops significantly, which affects the generated power, as well as the efficiency of the system.

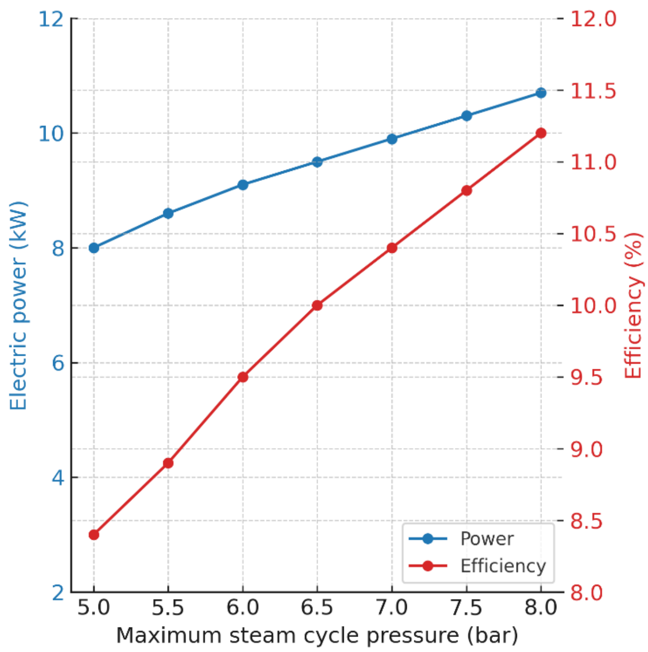

Fig. 4 *Analysis of the effect of changing the maximum pressure on the power and efficiency of the system.*

The effect of maximum cycle pressure on performance is shown in Figure 4. Trends in both power and efficiency highlight that the system performs better at the highest possible pressure (taking into account the temperature of the heat source). Increasing the nominal pressure from 5.0 to 10.0 bar allows a 27.8% increase in power output. Simplified simulations developed using TRNSYS software have shown that the developed mCHP system based on a flame steam generator, operating at a pressure of 10.0 bar and a maximum steam temperature of 185°C, can achieve power and efficiency of more than 10kW and 10%, respectively, with optimized steam flow values. The developed model allows to carry out dynamic simulations of the mCHP system under selected operating parameters, but further work needs to be expanded and the constraints and boundary conditions need to be made more precise. The model does not take into account factors that can affect the final power and efficiency of the system. It provides a basis for further work on the design of a trigeneration system, where the heat can also be used for cooling purposes.

4. Data Exchange Structure

In the context of control system research, a key component is the Type155 block, which enables data exchange between the TRNSYS environment and MATLAB. This functionality was implemented to integrate custom control algorithms into the simulation environment modeling the operation of a trigeneration installation. As a result, the TRNSYS simulation can be controlled using optimization algorithms developed in MATLAB. A graphical representation of the data exchange mechanism and the structure of parameter mapping is shown in Figure 5. In addition to interaction between software environments, it is also possible to import input data from auxiliary files, such as weather data or consumption profiles. All constant parameters, input variables, and external data are processed, and the results of the computational operations are provided as output variables.

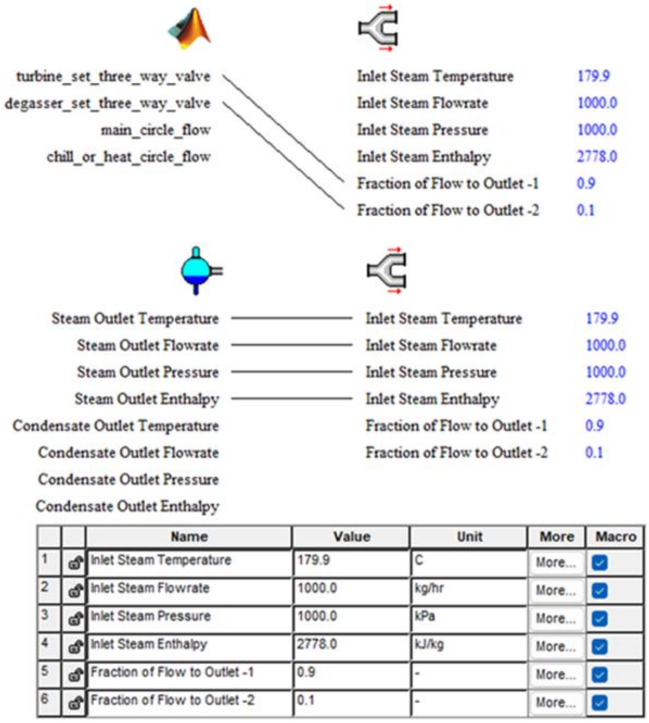

Fig. 5 *Data transfer structure between environments using the Type155.*

To enable training and simulation of a reinforcement learning (RL)-based algorithm, additional scripts were developed to facilitate data exchange between decision-making programs. In this configuration, the program invoked by the Type155 block serves solely as a bridge between the neural network training environment and the simulated operation of the cogeneration unit. On the MATLAB side, data transmission was established using a 'localhost' connection, which is used to set up network communication within the same device. In this case, it allows two separate MATLAB programs to communicate with each other.

The communication was implemented using the tcp_client and tcp_server tools, which enabled the transfer of data from the TRNSYS simulation and the return of computed results. This setup allowed the RL network to learn based on observations from the dynamic simulation of the trigeneration system and to generate actions in response. The performance of these actions was then rewarded or penalized. The internal structure and logic of the RL algorithms implemented in the MATLAB environment are described in [14, 15].

The main component of the reward function (R_1) is based on the difference between the power demand ($powerDemand$) and the power generated ($powerGenerated$) (e2).

$$R_1 = -\sqrt{abs(powerDemand - powerGenerated)} \qquad (e2)$$

The use of the absolute value allows penalizing the agent's actions in cases of both power surplus and deficit. Applying the square root to the difference intensifies the impact of deviations, both when the mismatch is large and when approaching zero error. An important aspect is also the inclusion of a penalty in situations where no variation is detected in the decision variables relative to the moving average over the last five iterations. This penalty is applied only if the deviation exceeds 0.1 kW. This threshold was defined as the acceptable deviation from the network's target performance. It helps prevent excessive oscillations near the target value and reduces the risk of misleading the agent during training—especially in scenarios where a zero error is achieved, but the agent is still required to take action through its decision variables.

In the case of fixed settings, the trajectory of this parameter follows a straight line, while in the learning scenario, each decision variable independently affects power generation. The ranges of the decision variables were defined within 55–95% to avoid operation in undesired regions. The variable demand defined in the model falls within the range of 5 to 10 kWe, while control of the decision variables allows the system to generate power within the range of 4 to 10 kWe. The remaining simulation parameters result from the decision variable settings and are recalculated during each program iteration (every 3 minutes in a 1000-hour simulation). The main objective of the algorithm is to match the generated power to the consumer's demand over time.

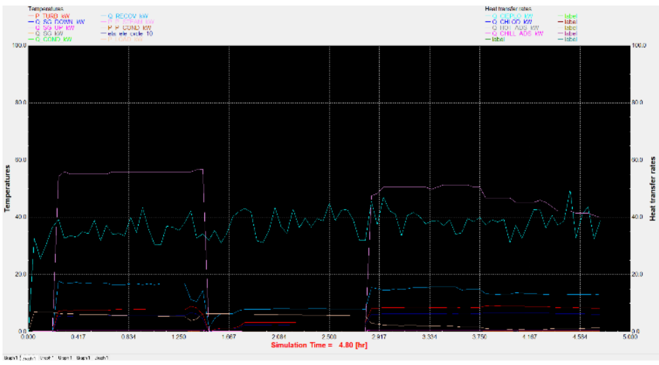

Fig. 6 *Data transfer structure between environments using the Type155.*

Figure 6 presents selected curves illustrating the operational parameters of the simulated trigeneration system over time. The startup process of the installation is clearly visible, along with the initial interactions of the neural network, reflected in the fluctuations of the generated power (P_TURB).

5. Conclusion

The developed simulation model of the trigeneration energy system enables full integration of external control algorithms with the dynamic structure of the thermodynamic system. A key role is played by the Type155 component, which facilitates bidirectional real-time data exchange between the TRNSYS and MATLAB environments. This makes it possible to incorporate a decision-making layer based on reinforcement learning, which dynamically adjusts control signals to match the variable demand profile. The model allows for the observation of how control decisions affect the physical parameters of the system, such as turbine power and efficiency, creating a foundation for the analysis and development of control strategies in a virtual environment. Importantly, the tool fulfills all key aspects of proper optimization: it attempts to achieve the objective function by maximizing the reward and responds by adjusting parameters within the allowed ranges.

The proposed architecture confirms the relevance of using multiple engineering tools for comprehensive energy system design. The learning process, based on both simulation and experimental data, allows for safe testing of algorithms without the risk of destabilizing a real installation. The model can serve as a research tool for the implementation and evaluation of adaptive strategies in the energy sector.

Funding

This research was funded by the Excellence Initiative – Research University, Action. 12 – Integration of educational process with scientific research – Support for student research clubs – Rector's grants, 5rd edition of the competition (ID 8980).

6. References

1. Nichols, Daniel. "Arduino-based data acquisition into Excel, LabVIEW, and MATLAB." The Physics Teacher 55.4 (2017)
2. Papkov, Vyacheslav, Nikita Shadymov, and Dmitry Pashchenko. "CFD-modeling of fluid flow in Ansys Fluent using Python-based code for automation of repeating calculations." International Journal of Modern Physics C 34.09 (2023)
3. Elomari, Youssef, et al. "A hybrid data-driven Co-simulation approach for enhanced integrations of renewables and thermal storage in building district energy systems." Journal of Building Engineering 104 (2025)
4. Knap, Paweł, Krzysztof Lalik, and Patryk Bałazy. "Boosted convolutional neural network algorithm for the classification of the bearing fault form 1-d raw sensor data." *Sensors* 23.9 (2023)
5. Balazy, Patryk, Kamil Pieprzycki, and Paweł Knap. "Robust Reinforcement Learning For Overhead Crane Control With Variable Load Conditions." *2024 25th International Carpathian Control Conference (ICCC)*. IEEE, 2024
6. Lalik, K., et al. "Q-Learning neural controller for steam generator station in micro cogeneration systems. Energies 14, 5334 (2021)
7. Meiers, Josef, and Georg Frey. "Interfacing TRNSYS with MATLAB for Building Energy System Optimization." *Energies* 18.2 (2025)\
8. Adesanya, Misbaudeen Aderemi, et al. "Deep reinforcement learning for PID parameter tuning in greenhouse HVAC system energy Optimization: A TRNSYS-Python cosimulation approach." *Expert Systems with Applications* 252 (2024)
9. Podlasek, Szymon, et al. "Application of ANN control algorithm for optimizing performance of a hybrid ORC power plant." *Energy* 306 (2024)
10. Bakeer, Abualkasim, et al. "Effect of mission profile resolution on photovoltaic energy yield prediction in Python and MATLAB." *2021 IEEE 15th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*. IEEE, 2021
11. Sudhakar, K., et al. "Modelling of a solar desiccant cooling system using a TRNSYS-MATLAB co-simulator: A review." *Journal of Building Engineering* 24 (2019)
12. Żołądek, Maciej, Rafał Figaj, and Krzysztof Sornek. "Energy analysis of a micro-scale biomass cogeneration system." *Energy Conversion and Management* 236 (2021)
13. Imran, Muhammad, et al. "Dynamic modeling and control strategies of organic Rankine cycle systems: Methods and challenges." *Applied energy* 276 (2020)
14. https://www.mathworks.com/products/reinforcement-learning.html
15. Ciaburro, Giuseppe. *MATLAB for machine learning*. Packt Publishing Ltd, 2017.