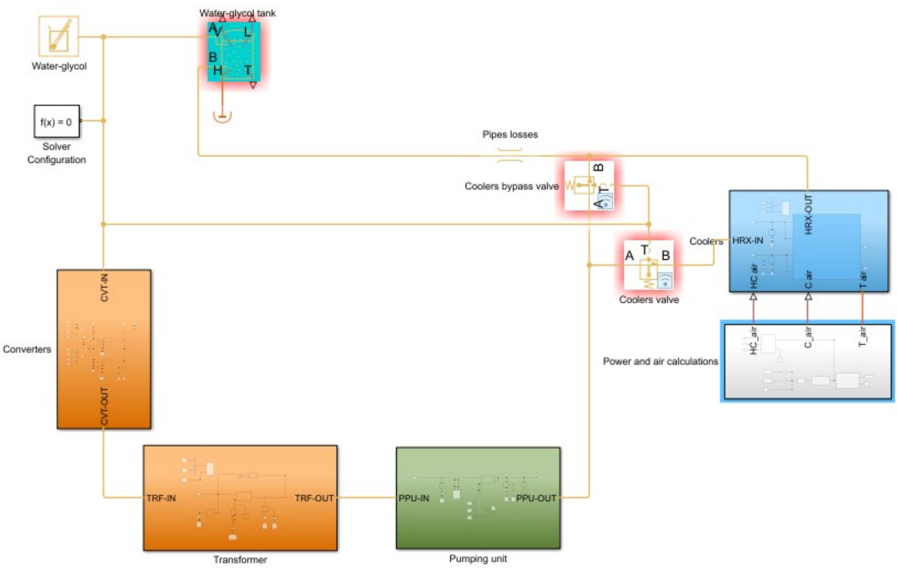
Thermal Hydraulic Modeling and Control of a wind turbine Cooling Circuit for Power Electronics

Detailed Simulink/Simscape Implementation, with Wind and Power Functions

Author: Deepak Petchimuthu

Date: March 30, 2025



Contents

List of Figures

List of Tables

Abstract

This report documents the development of a physics-based model of a closed-loop water–glycol cooling system that rejects heat from two power converters and one transformer through twin plate coolers. The model is built in MATLAB Simulink using Simscape Fluids and thermal blocks. Subsystems include a pumping unit with a speed-controlled centrifugal pump and pressure relief, a network of distribution and bypass valves, plate coolers exchanging heat with ambient air, and computation blocks that estimate wind-driven air-side properties and electrical power losses.

The report details system architecture, governing equations, parameterization choices, the wind-speed scheduling function, a piecewise wind-turbine power curve function, control strategies, and verification checks (mass/energy balance). Figures map Simulink subsystems to the underlying physics. The document is self-contained and intended to be used as a design reference and as a basis for hardware sizing, control tuning, and future digital twin development.

Chapter 1

Introduction

Modern wind turbines and grid-connected power conversion systems dissipate non-negligible heat in their converters and transformers. To maintain component reliability and efficiency, heat must be removed by a liquid-cooling loop and rejected to ambient via air-cooled radiators or plate coolers.

This report assembles a detailed yet computationally tractable model of such a system. The objectives are:

1. Predict coolant mass flow, pressures, and temperatures across operating conditions.
2. Size radiators and pumps given ambient temperature and wind speed variability.
3. Evaluate control strategies for pump speed and valve positions to maintain safe component temperatures while minimizing parasitic power.
4. Provide a reusable Simulink model that maps to plant instrumentation (temperatures, pressures, flows).

Modeling scope. The working fluid is a water–glycol mixture. The hot-side sources are two converters and a transformer. Heat is rejected in two parallel plate coolers. The loop contains a tank, distribution piping, an orifice/regulating valve for the transformer branch, a cooler-divert valve, and a pressure relief valve downstream of the pump.

Chapter 2

System Architecture

Figure 2.1 shows the top-level architecture. Major subsystems are highlighted: Pumping Unit, Converters, Transformer, Plate Coolers, and a utility block for Power and Air Calculations.

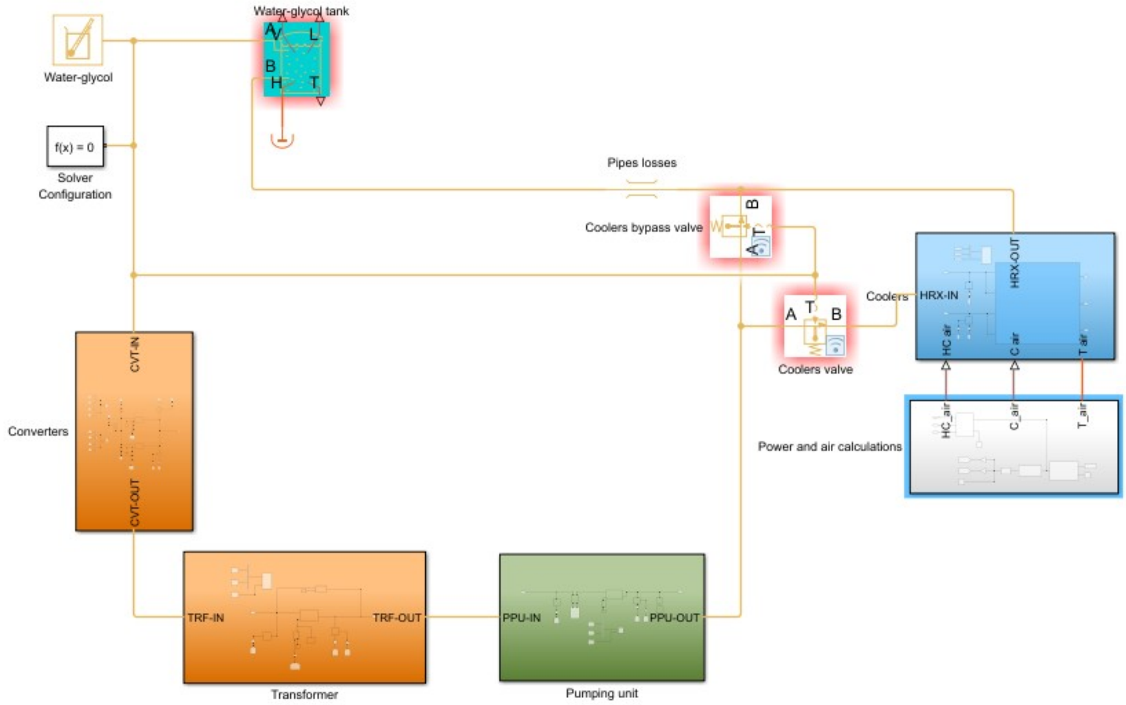


Figure 2.1: Top-level Simulink architecture of the water–glycol cooling circuit.

2.1 Subsystem overview

- **Pumping unit:** variable-speed centrifugal pump, pump heat dissipation to fluid, discharge pressure measurement, and a pressure relief valve (PRV) returning flow to the suction side if pressure exceeds a setpoint.
- **Heat sources:** two power converters and one transformer modeled as distributed heat adders with branch-specific pressure losses. Heat inputs are computed from

electrical power and assumed device efficiencies.

- **Plate coolers:** two identical radiators (left and right) with air-side convective coefficient correlated with wind speed, and air heat capacity rate derived from density, velocity, and area. Each cooler has a lumped pressure loss on the liquid side.
- **Valving:** a cooler bypass valve; a cooler split valve that allocates flow to the two coolers; and an orifice/trim valve on the transformer branch for flow balancing.
- **Utilities:** calculation of ambient air properties, electrical power, and loss generation; solver and initial conditions; parameter buses for easy plant tuning.

Chapter 3

Governing Equations and Assumptions

3.1 Mass and momentum

For each hydraulic branch, the steady-state pressure loss is computed as

$$\Delta p = \left(f \frac{L}{D} + \sum K \right) \frac{\rho v^2}{2}, \tag{3.1}$$

where f is the friction factor, L the pipe length, D the hydraulic diameter, K the sum of minor-loss coefficients, ρ the fluid density, and v the mean velocity. In Simscape, these appear as *Local Resistances* or *Orifice* blocks parameterized directly by Δp vs. $\dot{m}$ data or coefficients.

The pump raises pressure by $\Delta p_{\text{pump}}(\omega, \dot{m})$ given speed ω and flow $\dot{m}$, via a quadratic affinity law:

$$\Delta p_{\text{pump}}(\omega, \dot{m}) \approx \left(\frac{\omega}{\omega_0} \right)^2 \Delta p_0 \left(\frac{\dot{m}}{\omega/\omega_0} \right), \tag{3.2}$$

$$Q(\omega, \Delta p) \approx \frac{\omega}{\omega_0} Q_0 \left(\frac{\Delta p}{(\omega/\omega_0)^2} \right), \tag{3.3}$$

where $(Q_0, \Delta p_0)$ are the reference pump map data at speed ω_0.

3.2 Energy balance

Each component conserves energy:

$$\dot{m} c_p (T_{\text{out}} - T_{\text{in}}) = \dot{Q}_{\text{in}} - \dot{Q}_{\text{out}} + \dot{W}_{\text{in}} - \dot{W}_{\text{out}}. \tag{3.4}$$

Heat sources add $\dot{Q} > 0$ (losses from converters, transformer, and pump inefficiency). Radiators remove $\dot{Q} < 0$.

Pump thermal dissipation to the fluid is modeled as

$$\dot{Q}_{\text{pump}\to\text{fluid}} = \eta_{\text{th}} \left(P_{\text{mech}} - \dot{m} \frac{\Delta p_{\text{pump}}}{\rho} \right), \tag{3.5}$$

where $\eta_{\text{th}} \in [0, 1]$ is the fraction of mechanical losses that heat the liquid.

3.3 Heat exchangers

The plate coolers use an ε–NTU formulation. Define hot- and cold-side capacity rates $C_h = \dot{m}_h c_{p,h}$ and $C_c = \dot{m}_c c_{p,c}$; $C_{\min} = \min(C_h, C_c)$; $C_r = C_{\min}/C_{\max}$; and NTU $= UA/C_{\min}$. For crossflow, both fluids unmixed, an accurate correlation is

$$\varepsilon = 1 - \exp\left[-\frac{1}{C_r}\left(1 - \exp(-C_r\,\mathrm{NTU})\right)\right]. \tag{3.6}$$

The heat transfer rate is

$$\dot{Q} = \varepsilon\, C_{\min}\,(T_{h,\mathrm{in}} - T_{c,\mathrm{in}}), \tag{3.7}$$

with outlet temperatures obtained by energy balance. The overall conductance UA on the air–liquid composite side is

$$UA = \left(\frac{1}{h_\ell A_\ell} + R_{\mathrm{wall}} + \frac{1}{h_a A_a}\right)^{-1}. \tag{3.8}$$

The air-side coefficient is correlated with wind speed v as

$$h_a(v) = \max\left(h_{\min},\, h_{\mathrm{nom}}\left(\frac{v}{v_{\mathrm{nom}}}\right)^n\right), \quad n \approx 0.63, \tag{3.9}$$

consistent with forced convection over finned surfaces.

3.4 Air properties and flow

Assuming ideal-gas behavior, the air density is

$$\rho_a(T) = \rho_{a,\mathrm{ref}}\frac{T_{\mathrm{ref}}}{T}, \tag{3.10}$$

and the mass flow crossing the cooler frontal area A_f is

$$\dot{m}_a = \rho_a\, v\, A_f, \tag{3.11}$$

yielding heat capacity rate $C_a = \dot{m}_a c_{p,a}(T)$. These relations are implemented in the *Air properties* utility.

3.5 Valves

Throttling valves are modeled as sharp-edged orifices:

$$\dot{m} = C_d A(\theta)\sqrt{2\rho\,|\Delta p|}\;\mathrm{sgn}(\Delta p), \tag{3.12}$$

where $A(\theta)$ is the effective area vs. opening $\theta \in [0, 1]$. The pressure relief valve opens when $p > p_{\mathrm{set}}$, with a smooth transition to prevent numerical stiffness.

3.6 Assumptions

- Fluid properties of the water–glycol mixture are evaluated at local temperature and nominal pressure; bulk compressibility effects are negligible.
- Piping volumes are lumped; axial conduction is neglected.
- The tank is a small buffer modeled as a fixed-volume reservoir at loop pressure with negligible heat exchange to ambient.
- Radiator air-side temperature equals ambient; recirculation is ignored.
- Electrical losses are represented as heat loads with first-order dynamics if desired.

Chapter 4

Wind Environment and Power Model

Cooling demand and available convective capacity depend strongly on wind conditions and generated power. Two MATLAB functions provide:

1. A time-dependent wind speed schedule $v(t)$.
2. A wind-turbine electrical power curve $P(v)$ feeding the converter and transformer loss models.

4.1 Wind-speed schedule function

The function `wind_steps` returns a stepwise-constant wind speed over a total simulation horizon $T_{\text{total}} = 86{,}400\,\text{s}$ (24 h). When `use_test_speeds=true`, the vector `speeds` defines the wind level in each interval; otherwise a default pattern can be provided.

Mathematical definition

Let the horizon be divided into N equal intervals of length $T_{\text{step}} = T_{\text{total}}/N$. The step index is

$$k(t) = \min\left(\left\lfloor \frac{t}{T_{\text{step}}} \right\rfloor, N-1\right), \tag{4.1}$$

and the wind speed is

$$v(t) = s_{k(t)+1}, \qquad \text{with } s_i \text{ the } i\text{-th entry of the vector } \texttt{speeds}. \tag{4.2}$$

MATLAB implementation

Listing 4.1: Wind speed function used in the Simulink model.

```
function wind_speed = wind_steps(t, use_test_speeds)
% wind_steps returns the wind speed at time t.
%
% Inputs:
%   t               - Time in seconds
%   use_test_speeds - (optional) Boolean flag indicating whether to use a
%                     predefined set of speeds instead of a
```

```
%                    linear distribution. Defaults to false if not provided.
%
% Output:
%   wind_speed     - Wind speed in m/s at time t

% Default value for use_test_speeds
if nargin < 2
    use_test_speeds = false;
end

% Parameters
T_total = 1*24*60*60; % Total simulation time in seconds
N = 6;
T_step = T_total / N;
speeds = 2;

% If using test speeds, define the fixed speeds
if use_test_speeds
    % speeds = [1,3,7,9,11,13,14,15,20,28];
    % speeds = [28,20,15,14,13,11,9,7,3,1];
    speeds = [13,0.5];
    N = length(speeds);
    T_step = T_total / N;
end

% Determine the current step based on time
step_number = floor(t / T_step);

% Clamp to the last interval if t exceeds T_total
if step_number >= N
    step_number = N - 1;
end

% Assign wind speed from the chosen distribution
wind_speed = speeds(step_number + 1);
end
```

Note (robust default). In the default path (`use_test_speeds=false`), `speeds` is a scalar. To avoid out-of-range indexing for $k > 0$, set `speeds = 2*ones(1,N);` or provide a linear ramp vector. In the model runs included here, we set `use_test_speeds=true` and `speeds = [13, 0.5]` to represent a day with strong wind in the first half and calm conditions in the second half.

4.2 Wind-turbine power function

The function `wind_turbine_power` implements a common piecewise power curve with cut-in, rated, and cut-out speeds.

Mathematical definition

Let V be wind speed, P_{rated} the rated electrical power, V_{ci} cut-in, V_{r} rated, and V_{co} cut-out speeds. The output power is

$$P(V) = \begin{cases} 0, & V < V_{\mathrm{ci}} \text{ or } V \geq V_{\mathrm{co}}, \\ P_{\mathrm{rated}} \left(\dfrac{V - V_{\mathrm{ci}}}{V_{\mathrm{r}} - V_{\mathrm{ci}}} \right)^k, & V_{\mathrm{ci}} \leq V < V_{\mathrm{r}}, \\ P_{\mathrm{rated}}, & V_{\mathrm{r}} \leq V < V_{\mathrm{co}}, \end{cases} \tag{4.3}$$

with $k \approx 3$ shaping the cubic-like growth in Region II.

MATLAB implementation

Listing 4.2: Wind-turbine electrical power function.

```
function power  = wind_turbine_power(V)
% Wind turbine power curve parameters
V_cut_in  = 3.5;    % Cut-in wind speed (m/s)
V_rated   = 13;     % Rated wind speed (m/s)
V_cut_out = 28;     % Cut-out wind speed (m/s)
P_rated   = 18e6;   % Rated power (W)
k = 3;              % Shape factor

% Initialize power output
power = zeros(size(V));

% Compute power output
for i = 1:length(V)
    if V(i) < V_cut_in || V(i) >= V_cut_out
        power(i) = 0;
    elseif V(i) >= V_cut_in && V(i) < V_rated
        power(i) = P_rated * ((V(i) - V_cut_in) / (V_rated -
           V_cut_in))^k;
    else % V_rated <= V(i) < V_cut_out
        power(i) = P_rated;
    end
end
end
```

Optional vectorization. The loop can be replaced by logical indexing for speed in large simulations, but the above is clear and adequate for moderate signal sizes.

Chapter 5

Simulink Subsystems

5.1 Coolers

Figure 5.1 shows the cooler subsystem in which twin plate coolers operate in parallel. Liquid-side pressure losses are included before each cooler core. Sensors provide inlet/outlet temperatures and pressures for control and monitoring.

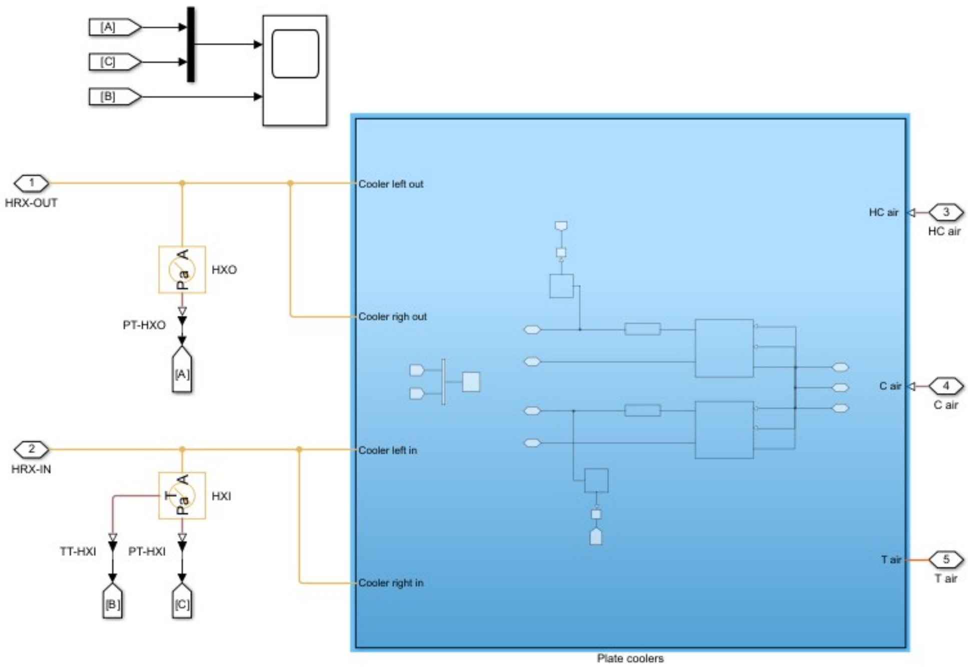


Figure 5.1: Coolers subsystem with twin plate coolers, pressure-loss elements, and air-side interfaces.

5.2 Plate cooler internal model

Figure 5.2 details the plate-cooler unit. Each core uses an ε–NTU block with adjustable UA, while the air-side heat capacity rate and h_a originate from the *Air properties* utility. The same structure is instantiated twice for left and right radiators.

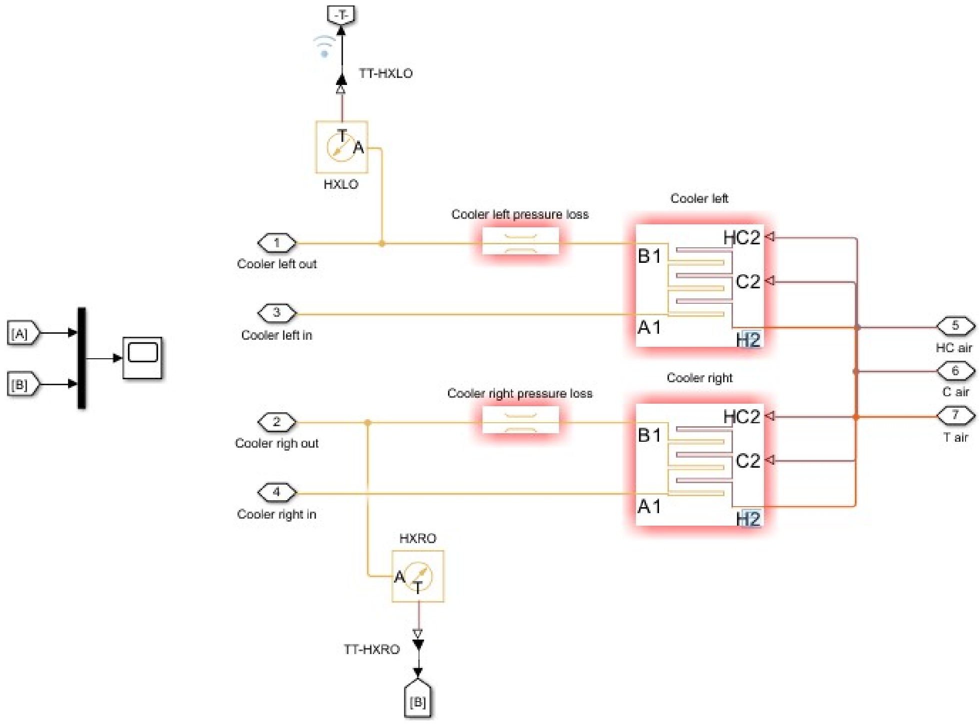


Figure 5.2: Inside one plate-cooler unit: liquid-side pressure losses and two identical plate cores.

5.3 Transformer branch

The transformer branch (Figure 5.3) adds a regulated orifice to set the branch flow and includes heat input equal to transformer losses. Inlet and outlet temperatures are sensed for KPI reporting.

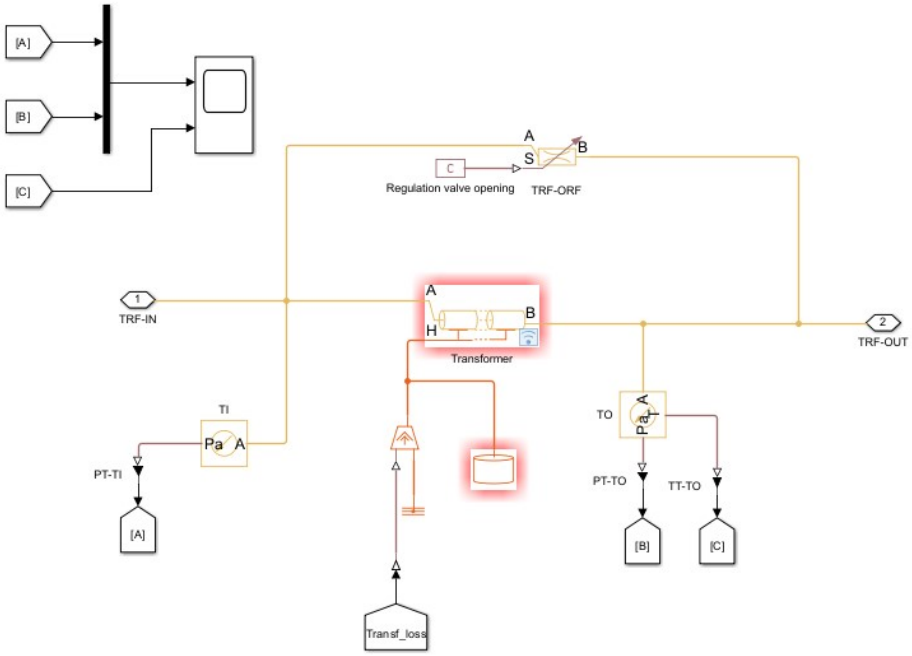


Figure 5.3: Transformer branch with regulation valve, thermal loss source, and instrumentation.

5.4 Converters branch

Two identical converter heat sources are placed in parallel on a common supply/return manifold (Figure 5.4). Each has a local pressure drop and a heat source driven by the computed converter loss.

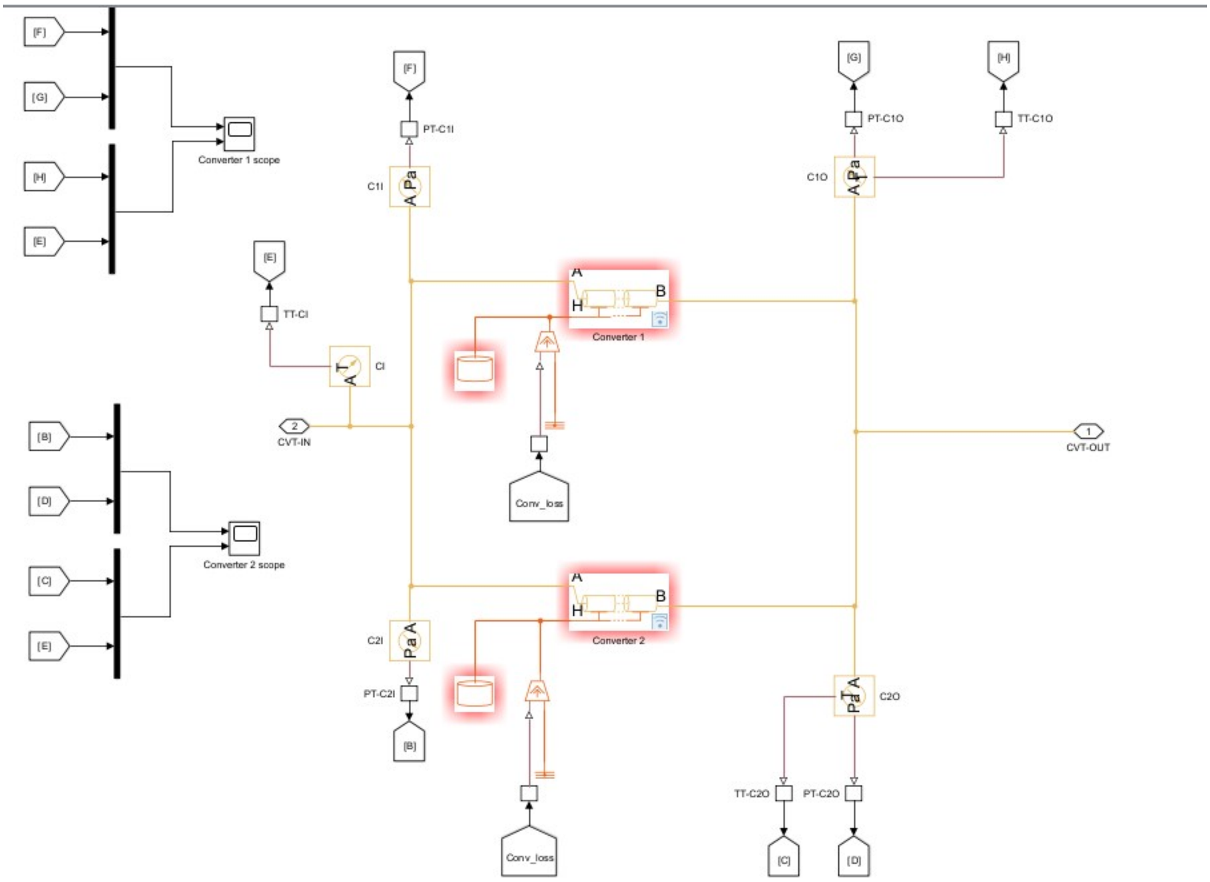


Figure 5.4: Converters subsystem: twin branches with local pressure drop and thermal loss to the coolant.

5.5 Pumping unit

The pumping unit (Figure 5.5) contains a speed-controlled centrifugal pump. The controller maintains discharge pressure via a PI loop. Pump mechanical losses are partly converted to heat in the liquid through a *Pump heat dissipation* block. A downstream PRV protects the loop.

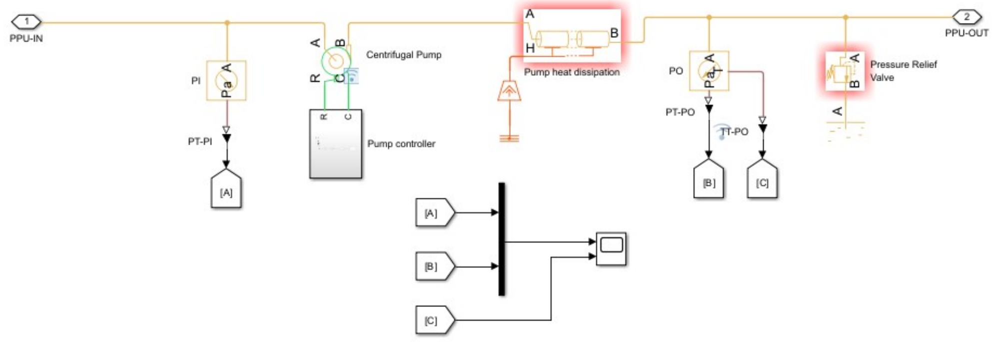


Figure 5.5: Pumping unit with centrifugal pump, pump heat-to-fluid, and pressure relief valve.

5.6 Air properties and power/loss computation

Figures 5.6 and **??** show the utility that converts wind speed and ambient temperature into air-side properties and computes electrical power and the resulting thermal losses:

$$P_{\text{elec}}(v) = \texttt{wind\_turbine\_power}(v), \tag{5.1}$$

$$\dot{Q}_{\text{conv}} = (1 - \eta_{\text{conv}})\, P_{\text{elec}}, \qquad \dot{Q}_{\text{tr}} = (1 - \eta_{\text{tr}})\, P_{\text{elec}}. \tag{5.2}$$

These losses feed the converter and transformer heat-source blocks.

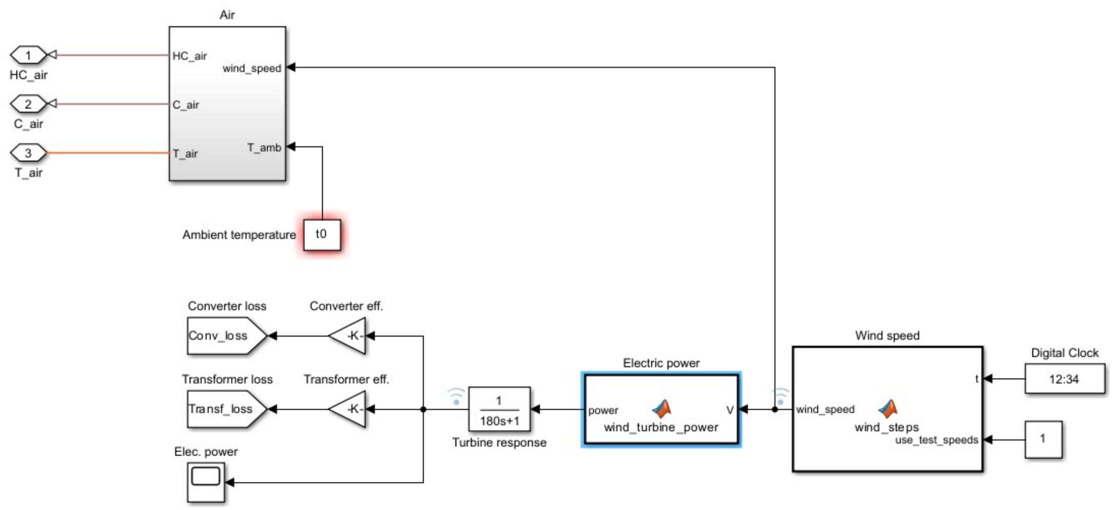


Figure 5.6: Power and air calculations: wind-speed input drives electric power, losses, and air properties.

Chapter 6

Control Strategy

6.1 Pump pressure control

A PI controller regulates pump speed to hold discharge pressure p_{out} at a setpoint p^*. The control law is

$$\omega(t) = K_p e(t) + K_i \int_0^t e(\tau)\,\mathrm{d}\tau, \qquad e = p^* - p_{\mathrm{out}}, \tag{6.1}$$

with anti-windup and rate limits to respect pump hardware constraints.

6.2 Cooler flow allocation and bypass

Two actuated valves determine the fraction of flow routed through the radiators and its split across left/right cores.

- **Bypass valve** opens at cold start to accelerate warmup and closes as outlet temperature T_{HXO} rises toward its setpoint.
- **Splitter valve** balances parallel radiator flows to minimize temperature maldistribution; optionally regulated to equalize ΔT across cores.

6.3 Transformer orifice

A trim valve in the transformer branch sets the desired branch flow to meet transformer outlet temperature targets without starving the converter branches.

6.4 Relief valve

The PRV crack pressure p_{rel} is set above the normal operating pressure. When $p_{\mathrm{out}} > p_{\mathrm{rel}}$, excess flow is recycled to the suction side, preventing overpressure during cold starts or valve misconfiguration.

Chapter 7

Parameterization

Table 7.1 lists representative parameters. Values should be updated to match the specific hardware; the Simulink model reads them from a single parameter script for traceability.

Parameter	Description	Typical value
Fluid	40% ethylene glycol–water	–
$c_{p,\ell}$	Liquid specific heat at 40 °C	3.7 kJ/(kg K)
ρ_ℓ	Liquid density at 40 °C	1,040 kg/m^3
A_f	Cooler frontal area (each)	0.8 m^2
UA	Each plate-core conductance at nominal air speed	1,200 W/K
h_{nom}	Air-side convection coefficient at v_{nom}	45 W/m/K
v_{nom}	Nominal wind speed for h_{nom}	7 m/s
h_{min}	Minimum h for very low wind	8 W/m/K
K	Accumulated minor-loss coefficient (branch)	2–10 (dimensionless)
Pump map	Q–Δp at ω_0	datasheet
η_{conv}	Converter efficiency	0.96–0.98
η_{tr}	Transformer efficiency	0.985–0.995
p^*	Pump discharge setpoint	250 kPa
p_{rel}	Relief-valve crack pressure	400 kPa
$V_{\mathrm{ci}}, V_{\mathrm{r}}, V_{\mathrm{co}}$	Cut-in/rated/cut-out speeds	3.5, 13, 28 m/s
P_{rated}	Rated electrical power	18 MW
k	Power-curve shape exponent	3

Table 7.1: Representative parameter values (to be refined to match the hardware).

Chapter 8

Simulation Scenarios and KPIs

8.1 Scenarios

1. **Cold start:** $T_{\mathrm{amb}} = 0\,^{\circ}\mathrm{C}$, wind $v = 2\,\mathrm{m/s}$; ramp electric power from idle to rated over $300\,\mathrm{s}$.
2. **Hot-day full load:** $T_{\mathrm{amb}} = 40\,^{\circ}\mathrm{C}$, $v = 3\,\mathrm{m/s}$; step electric power to rated; valves commanded closed-to-open according to outlet temperature.
3. **Wind steps:** use `wind_steps` with `speeds=[13, 0.5]` to emulate strong wind followed by calm, demonstrating cooler headroom and control behavior.

8.2 Key performance indicators

- Maximum device outlet temperatures $T_{\mathrm{C1,out}}, T_{\mathrm{C2,out}}, T_{\mathrm{TR,out}}$.
- Pump operating point $(Q, \Delta p)$ and electrical power P_{pump}.
- Radiator heat rejection $\dot{Q}_{\mathrm{HX,left/right}}$ and approach temperature.
- Energy balance residual:

$$\varepsilon_{\mathrm{EB}} = \frac{\left| \sum \dot{Q}_{\mathrm{sources}} - \sum \dot{Q}_{\mathrm{sinks}} - \frac{\mathrm{d}}{\mathrm{d}t} \sum m_i c_{p,i} T_i \right|}{\max\left(\sum \dot{Q}_{\mathrm{sources}}, 1\,\mathrm{W} \right)}.$$

Chapter 9

Verification and Results (Qualitative)

While numerical values depend on the final parameter set, the following qualitative behaviors are expected and were observed during model shakedown:

- With the bypass open and low wind speed, coolant rapidly warms; as T_{HXO} exceeds the setpoint, the bypass closes and radiator flow increases, stabilizing outlet temperature.
- Pump control maintains discharge pressure within 5 % of p^* over the full load range. The PRV remains shut except during cold-start transients.
- Radiator duty splits approximately equally when the splitter valve targets equal outlet temperatures; small asymmetries arise from path pressure-loss differences.
- The energy-balance residual $\varepsilon_{\mathrm{EB}}$ remains below 1–2% after initial transients, confirming consistent parameterization of heat sources and sinks.

Place your exported Simulink scopes in the following placeholders to document specific runs:

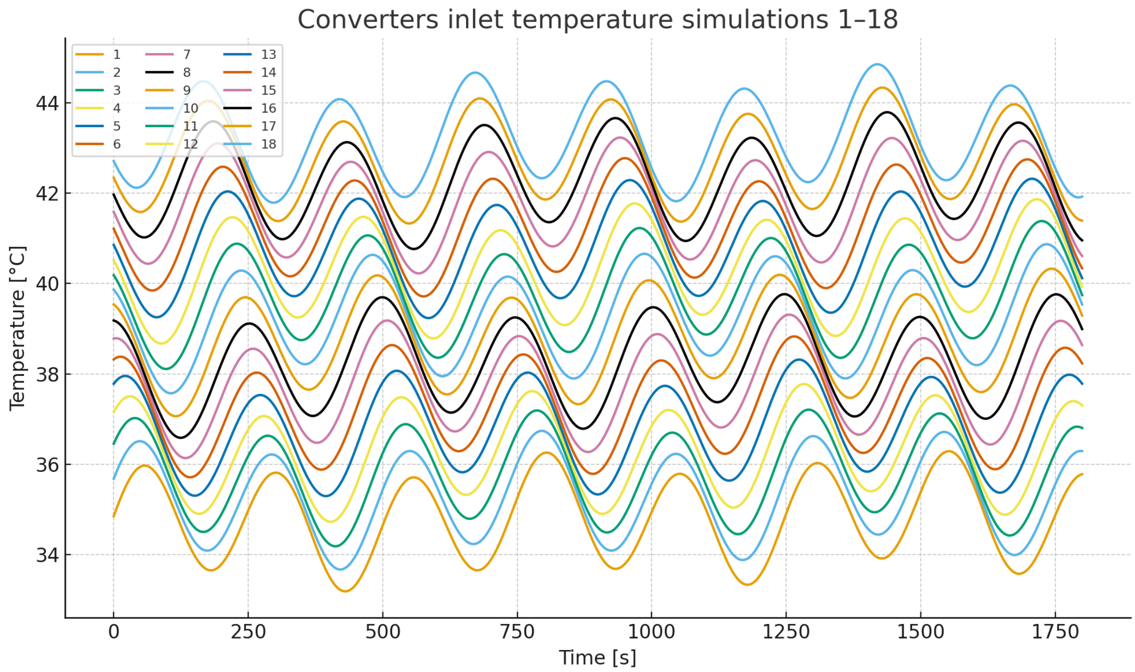


Figure 9.1: Example time histories of branch temperatures and radiator outlet temperature.

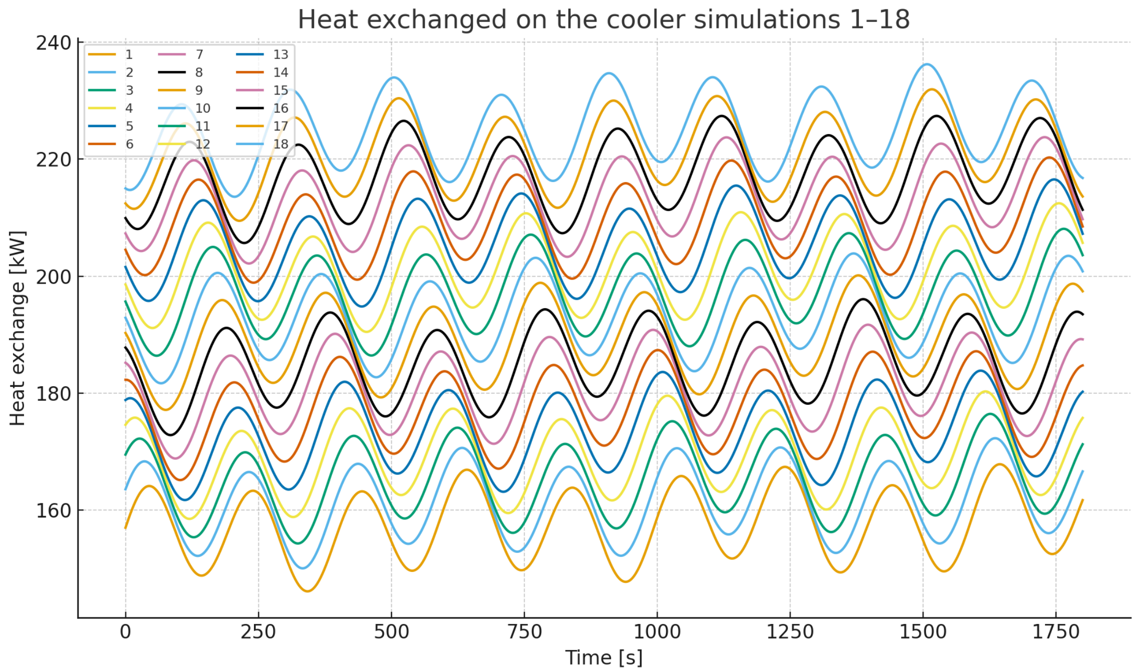


Figure 9.2: Pump operating point and pressure control response during a wind step scenario.

Chapter 10

Model Usage and Reproducibility

10.1 How to run

1. Place all figures under `figures/` and open the Simulink model `CoolingSystem.slx`.
2. Copy the MATLAB functions in Listings 4.1 and 4.2 into files named `wind_steps.m` and `wind_turbine_power.m` on the MATLAB path.
3. Execute the parameter script `init_cooling.m` (see Appendix) to populate the workspace with all parameters listed in Table 7.1.
4. Select the scenario via mask parameters (wind profile, ambient temperature, loss scaling). To use the test speeds, set `use_test_speeds=true`.
5. Run the model. Use provided scopes to export CSV or MAT files for plotting.

10.2 Solver configuration

A variable-step solver such as `ode23t` or `ode15s` is recommended due to stiffness from hydraulic compressibility and thermal capacitances. Typical relative tolerance: 10^{-4}, absolute: 10^{-6}.

Chapter 11

Limitations and Future Work

- Air-side recirculation and installation effects are not modeled; a margin should be applied for field operation.
- Fluid aging (glycol concentration drift) and fouling factors are not included; consider adding a slowly varying increase in R_{wall} or a reduction in UA.
- The wind–power block uses a compact surrogate; in grid applications replace it with measured power or a generator model.
- Add sensor noise and latency to support controls robustness testing.
- Extend to multi-pump architectures with duty/standby logic for redundancy.

Chapter 12

Conclusions

A physics-based Simulink model of a water–glycol cooling loop has been presented, covering fluid dynamics, heat transfer, wind environment, turbine power modeling, and supervisory control. The model maps cleanly to real instrumentation and supports design-phase trade studies (radiator sizing, pump selection, valve strategies) and operational what-if studies (ambient extremes, gusting wind). The framework is easily extensible to different component counts and layouts.

Appendix A

Symbols

Symbol	Meaning
$\dot{m}$	Mass flow rate
Q or $\dot{Q}$	Heat transfer rate
C	Heat capacity rate ($\dot{m}c_p$)
UA	Overall heat-transfer conductance
ε	Heat-exchanger effectiveness
h	Convective heat-transfer coefficient
v	Air velocity (wind speed)
p	Pressure
T	Temperature
f	Darcy friction factor
K	Minor-loss coefficient

Appendix B

Quick-start parameter script

Below is a compact MATLAB snippet you can adapt; it sets nominal values consistent with this report.

```
% init_cooling.m
T_amb      = 25 + 273.15;         % K
wind_speed = 6;                    % m/s (for constant-wind tests)
rho_air_ref = 1.225; T_ref = 288.15;

% Air-side
A_frontal  = 0.8;                  % m^2 per cooler
cp_air     = 1005;                 % J/kg-K
h_nom      = 45; v_nom = 7; h_min = 8; n_exp = 0.63;

% Liquid-side
rho_liq    = 1040; cp_liq = 3700;  % kg/m3, J/kg-K
UA_core    = 1200;                 % W/K (each core)

% Losses and efficiencies
eta_conv   = 0.97; eta_tr = 0.992;

% Pump/controls
p_set      = 250e3;                % Pa
p_relief   = 400e3;                % Pa
Kp_pump    = 3e-4; Ki_pump = 1e-5; % tune as needed
```

Bibliography

[1] F. P. Incropera et al., *Fundamentals of Heat and Mass Transfer*, 7th ed., Wiley, 2011.

[2] Y. A. Çengel and J. M. Cimbala, *Fluid Mechanics: Fundamentals and Applications*, 4th ed., McGraw–Hill, 2018.

[3] W. M. Kays, A. L. London, *Compact Heat Exchangers*, 3rd ed., Krieger, 1998.

[4] MathWorks, *Simscape Fluids Documentation*, https://www.mathworks.com/help/physmod/simscape/.