System-Level Digital Twin for Cooling Optimization in Formula SAE EV

Arjun Yeleshwarapu
Mechanical Engineering
University of California, Riverside
Riverside, California
Email: ayele002@ucr.edu

***Abstract*—Thermal management is critical for maintaining reliability and performance of power electronics in electric vehicles. This paper presents a physics-based digital twin developed for the cooling system of a Formula SAE electric vehicle inverter. The digital twin integrates real vehicle data including inverter temperature, vehicle speed, and fan PWM signals to simulate thermal behavior and enable rapid design iteration. Results demonstrate that optimized fan control curves reduce peak inverter temperature by 11°C during endurance events without increasing energy consumption compared to traditional bang-bang control. The model also supports exploration of radiator sizing, angle, and fan count, enabling data-driven cooling system design prior to physical prototyping or wind tunnel testing.**

***Index Terms*—digital twin, thermal management, Formula SAE, electric vehicle, inverter cooling, fan control, MATLAB simulation**

I. Introduction

Thermal management of power electronics such as the inverter and motor is critical in electric vehicles (EVs) to ensure efficient operation and prevent overheating that can cause performance degradation or hardware failure. Effective cooling systems must maintain temperatures within safe limits while minimizing energy consumption.

Traditional bang-bang fan control logic turns fans on only after the inverter temperature crosses a threshold, often too late to prevent thermal buildup. This on/off approach causes large power spikes and oscillations, cycling the fans at full load or off, which is inefficient and can stress components.

Thermal derating is commonly implemented in inverter and motor controllers to protect hardware from damage as temperatures approach maximum safe limits. When the inverter temperature nears its cutoff (e.g., around 78 °C with an 80 °C max), power output is typically reduced by limiting torque or current, resulting in decreased performance such as slower acceleration or lower maximum speed. Maintaining a conservative temperature margin (e.g., 65 °C to 75 °C) helps avoid triggering derating, ensuring reliable and consistent system operation.

This project leverages a physics-informed digital twin — a digital replica of the physical cooling system that ingests real-world logged data — to enable system-level simulation and optimization. Beyond tuning fan curves, the digital twin provides a flexible platform for rapid prototyping of radiator sizing, fan configurations, control strategies, and other cooling system parameters without the need for costly physical builds or wind tunnel testing. This data-driven approach supports informed design decisions, reduces guesswork, and accelerates development cycles for the Formula SAE EV cooling system.

II. System Architecture

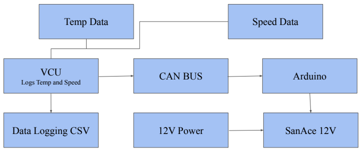


Fig. 1: System architecture for fan control. VCU sends temperature and speed data via CAN to the Arduino, which outputs PWM to the fans.

The cooling system consists of:

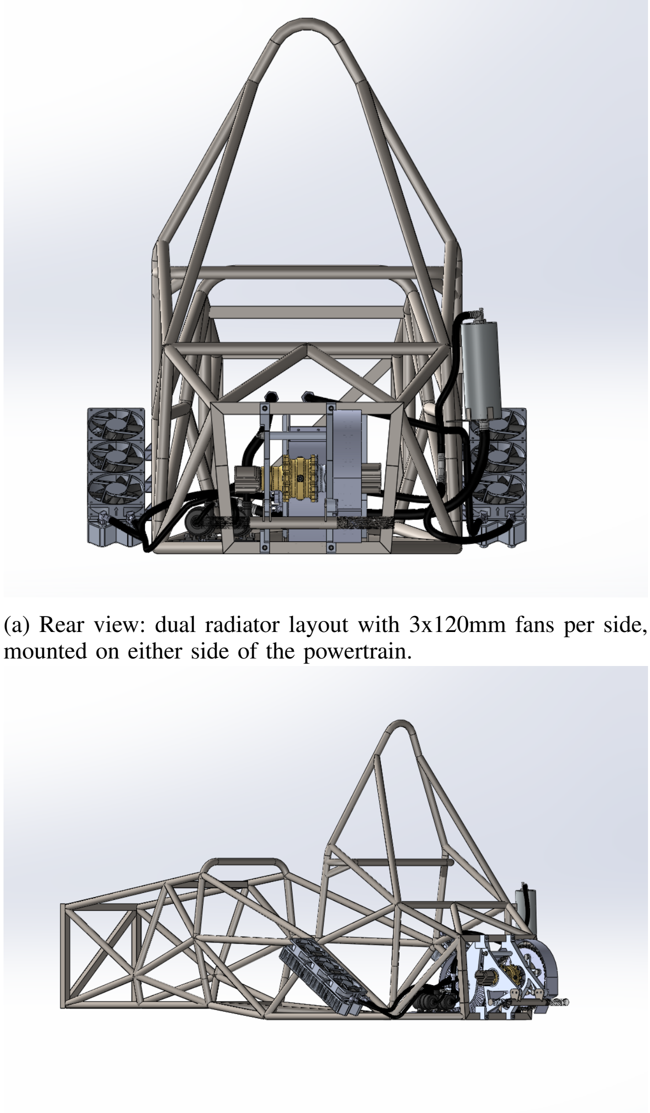
- **Fans:** Sanyo Denki San Ace 12V, 3A high-performance fans rated for approximately 120W power consumption.
- **Radiators:** HX-360XC radiators sized and configured to dissipate heat from the inverter effectively.
- **Control board:** An Arduino microcontroller interfaces with the vehicle control unit (VCU) via CAN bus, sending PWM signals to the fans. The firmware is based on the open-source Arduino PWM CAN Fan Controller.
- **Sensors and data acquisition:** Temperature and vehicle speed data are transmitted over CAN and logged by the VCU onto CSV files for post-processing.

Future work includes the development of a dedicated radiator fan control board to provide tighter hardware integration and improved signal fidelity.

III. Simulation and Testing

Simulations compare bang-bang, optimized linear ramp, and hysteresis PWM control strategies. The model runs using logged vehicle data for inverter temperature and speed, validating predicted thermal responses against measured data.

The digital twin supports rapid iteration of radiator sizing, fan count, and control parameters without physical rebuilds or wind tunnel access.



(a) Rear view: dual radiator layout with 3x120mm fans per side, mounted on either side of the powertrain.

(b) Side view: coolant loop routing from front-mounted radiator to rear inverter and pump setup.

Fig. 2: CAD model of cooling system implementation in the 2025 Highlander Racing EV. Radiators and fans are side-mounted with coolant lines routed through the chassis.

A. Fan Curve Optimization Procedure

```
guess = [60, 180];  % CFM at 50\% and 100\%
lb = [20, 100]; ub = [150, 300];
cost_fn = @(p) fan_curve_cost(p, t, v, qin, T0,
    rad_area_m2);
[opt_params, ~] = fmincon(cost_fn, guess, [], [],
    [], [], lb, ub);
```

Listing 1: Fan curve optimization call

This function performs the full fan curve optimization workflow. It prepares endurance lap data, defines radiator configuration, and runs a constrained optimization (fmincon) to find the optimal fan curve parameters minimizing thermal risk and energy use. After optimization, it simulates the inverter temperature response with and without active fans, calculates fan energy consumption, and generates plots to visualize results. This procedure enables data-driven tuning of fan curves using real vehicle operation data.

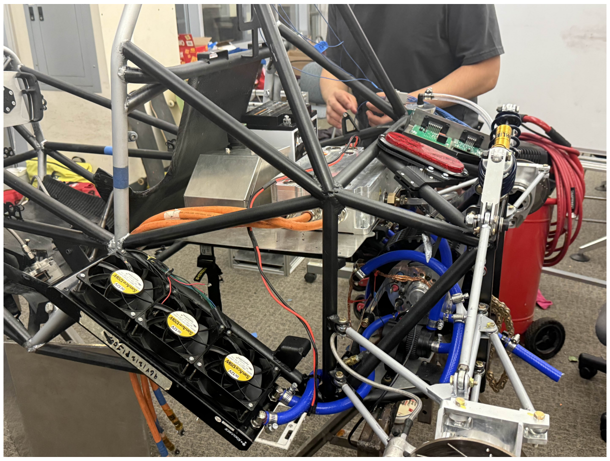


Fig. 3: Real-world implementation of the cooling system on the 2025 Highlander Racing EV. The left-mounted radiator assembly with three 120mm fans is visible, along with the inverter, wiring harness, and coolant routing.

B. Fan Curve Cost Function

```
function cost = fan_curve_cost(params, t, v, qin,
    T0, rad_area_m2)
    [temp, ~, power] = simulate([], t, v, qin, T0,
        params, true, rad_area_m2);

    dt = mean(diff(t));
    energy_wh = sum(power) * dt / 3600;

    % soft penalty above 65\si{\degreeCelsius}
    soft_limit = 65;
    soft_penalty_scale = 0.03;  % weight for going
        above 65\si{\degreeCelsius}
    temp_excess = temp(temp > soft_limit) -
        soft_limit;
    soft_penalty = sum(temp_excess.^2) *
        soft_penalty_scale;

    % hard cutoff if temp ever goes above
        75\si{\degreeCelsius}
    if max(temp) > 75
        cost = 1e6 + max(temp);  % punish
    else
        cost = energy_wh + soft_penalty;
    end
endfunction
```

Listing 2: Fan curve cost function

This cost function evaluates the performance of a candidate fan curve during simulation. It computes inverter temperature and fan power profiles given the input parameters and operating data. A soft penalty is applied for temperatures exceeding a safe threshold (65°C), and a hard cutoff penalty strongly discourages unsafe temperatures above 75°C. The cost combines energy consumption and thermal penalties to guide the optimizer towards efficient and safe fan curves.

C. Digital Twin Simulation Function

The inverter temperature was simulated using a lumped capacitance model that accounts for passive and active airflow, radiator area, fan count, and thermal properties of air and coolant. This simulation was used within the optimizer to evaluate fan curve performance under logged vehicle conditions. Full code is available in Appendix **??** and on GitHub: https://github.com/ayele002/cooling-digital-twin.
This function simulates the inverter thermal behavior over time given vehicle speed, inverter heat input, and fan curve parameters. It models both active fan cooling and passive airflow based on vehicle speed, converts airflow to convective heat transfer, and integrates the lumped thermal model. The simulation outputs arrays of predicted inverter temperatures, PWM control signals, and fan power consumption. This core model enables evaluating the thermal impact of different fan control strategies.

IV. Energy Analysis

The battery pack is configured as 5p16s, with 480 cells totaling approximately 33.6 kg. Assuming nominal cell voltage of 3.7 V and capacity of 4.2 Ah, the pack energy is estimated as:

$$E_{\text{pack}} = 16 \times 3.7\,\text{V} \times 5 \times 4.2\,\text{Ah} = 1243\,\text{Wh}$$

The cooling fans consumed approximately 77.7 Wh during endurance, corresponding to about 6.25% of pack capacity — an acceptable energy overhead for ensuring inverter safety.

V. Results

Results show the optimized fan curve reduced peak inverter temperature by 11 °C without increasing energy use compared to bang-bang logic. Peak fan PWM was reduced by 21.72%, improving efficiency and thermal margin.

Additionally, prolonged idle periods and the ambient heat conditions experienced during testing in Michigan demonstrated that passive cooling alone was insufficient to maintain safe inverter temperatures. This validated the necessity and effectiveness of active fan cooling on the vehicle.

The optimized linear ramp control smoothly adjusts fan power as temperature rises, preventing thermal runaway and reducing peak power consumption. This improves inverter reliability and energy efficiency during endurance.

As shown in Fig. **??**, bang-bang control results in frequent toggling as the system oscillates near the threshold temperature. This behavior leads to inefficient fan usage and increased wear, motivating the adoption of ramp-based PWM logic.

VI. Discussion

The predictive linear ramp control avoided the thermal overshoot seen in bang-bang logic. By reacting early to rising temperature, the fans prevented runaway and enabled the inverter to operate with safer thermal margins. The bang-bang approach stayed pinned at 100% PWM once triggered and failed to cool the system back down.

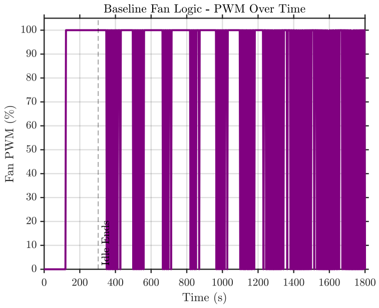


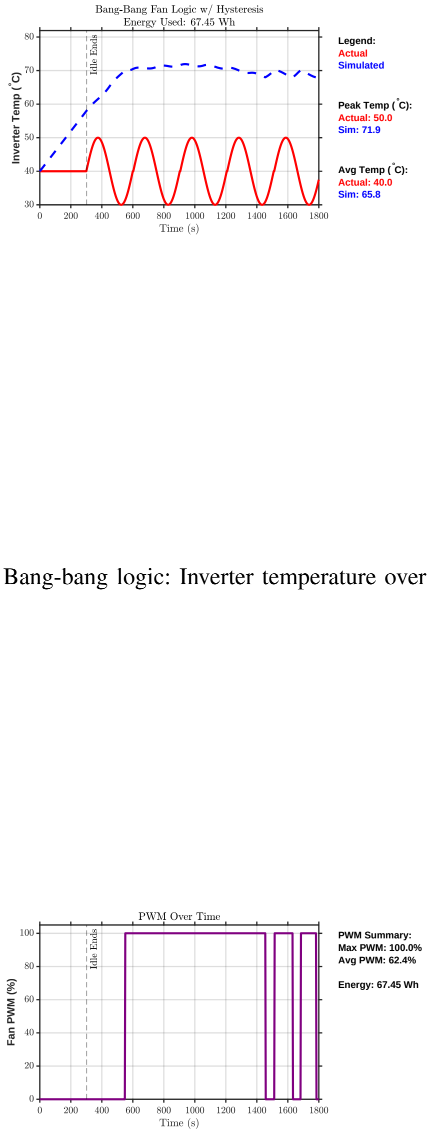
Fig. 4: Fan PWM output using pure bang-bang logic with no hysteresis or smoothing. The rapid toggling behavior is caused by the system temperature hovering near the control threshold, demonstrating the instability of naive control logic.

TABLE I: Comparison of fan control strategies

Metric	**Bang-Bang**	**Ramp**
Peak Temp (°C)	78.7	67.7
Avg Temp (°C)	70.8	63.5
Max PWM (%)	100	79.6
Fan Energy (Wh)	77.2	77.7
Battery Use (%)	6.21	6.25

More importantly, the digital twin serves as a comprehensive system-level simulation tool, enabling the team to rapidly prototype and evaluate various cooling system configurations — including radiator size, fan count, angle, and control logic — using real operational data. This capability removes dependence on physical testing resources such as wind tunnels or track sessions, which are often limited for student teams.

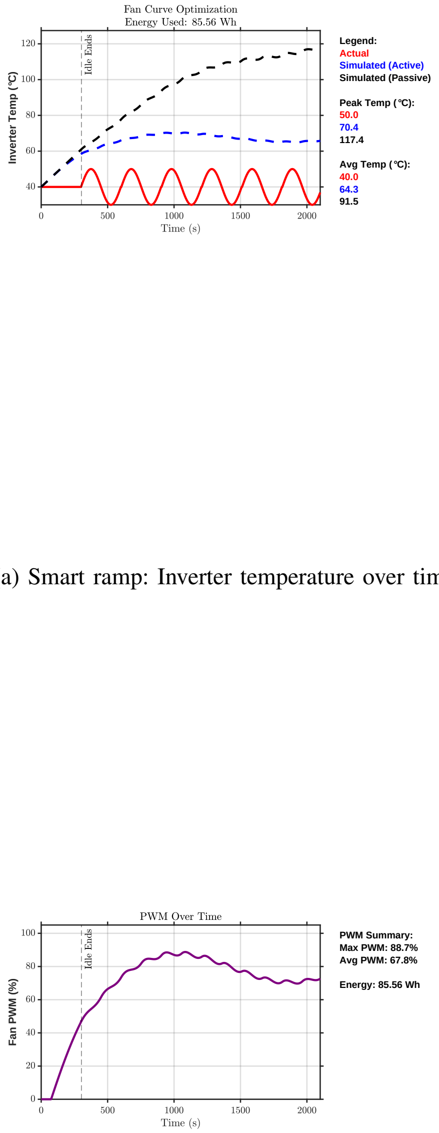
By grounding design iterations in simulated and logged thermal profiles, the team can make data-driven decisions that optimize cooling efficiency, reduce energy consumption, and improve reliability. The fan curve optimization presented here is one key outcome of this broader approach, demonstrating measurable improvements in thermal performance and power usage. However, the true value of the digital twin lies in its flexibility as a holistic cooling system design and validation platform.



(a) Bang-bang logic: Inverter temperature over time.

(b) Bang-bang logic: Fan PWM over time.

Fig. 5: Baseline bang-bang cooling logic results showing thermal runaway and 100% PWM saturation.



(a) Smart ramp: Inverter temperature over time.

(b) Smart ramp: Fan PWM over time.

Fig. 6: Optimized fan curve cooling logic results showing smooth temperature control and reduced fan power.

While developed for student electric racecar design, the digital twin approach demonstrated here is directly applicable to early-stage thermal system design in electric motorcycles, drones, or compact EVs, especially where physical prototyping is expensive or constrained.

VII. Conclusion

A MATLAB-based digital twin enabled us to design and validate smarter fan control logic for our FSAE EV. Our predictive ramp strategy reduced peak inverter temperature by $11\,^{\circ}\mathrm{C}$ while using the same energy as a traditional bang-bang controller. The ability to simulate passive vs. active cooling, tune radiator sizing, and compare fan configurations significantly improved our development speed and system understanding.

While the digital twin offers accurate simulation for fan-based cooling under logged track conditions, it currently assumes fixed ambient temperature and simplified passive airflow modeling. Head loss through the radiator and dynamic fan pressure-flow curves are not yet integrated, and coolant-side modeling is deferred for future work.

VIII. Future Work

While the current digital twin enables effective testing of fan curves and cooling setups, several extensions could improve realism and capability:

- **Model Predictive Control (MPC):** Incorporate a predictive control scheme that dynamically adjusts fan PWM based on future temperature trends.
- **Coolant Pumps:** Modeling the electric coolant pumps and their interaction with pressure head loss across the loop will allow for more realistic flow predictions and potential optimization of pump duty cycles.
- **Head Loss Estimation:** By integrating empirical or simulated head loss calculations, we can account for pressure drops across the radiator, hoses, and fittings. This will help correlate flow rate with inverter cooling capacity.
- **Heat Transfer Coefficient (h) Modeling:** Moving beyond a fixed cooling factor, future versions will dynamically compute h based on Reynolds number and Nusselt correlations, improving the accuracy of convective heat transfer prediction across varying speeds and airflow conditions.
- **Real-Time Twin Integration:** Link the simulation with live telemetry from the car for closed-loop control or live tuning during track testing.
- **Expanded Component Modeling:** Add detailed submodels for battery, motor, and ambient radiation effects to simulate full-vehicle thermal behavior.
- **Parameter Tuning Interface:** Create a GUI or script interface to let users adjust radiator size, fan count, and fan curves interactively.
- **Hardware-in-the-Loop Testing:** Connect the Arduino PWM output into the simulation for testbench-style validation of firmware logic.

Acknowledgment

Thanks to the UCR HR team and CS subteam for support and data access.

Appendix

All code used in this paper, including the simulation function and fan curve optimizer, is available at: https://github.com/ayele002/fsae-cooling-twin

[1]–[3]

References

[1] D. Piromalis and A. Kantaros, "Digital twins in the automotive industry: The road toward physical-digital convergence," *Applied System Innovation*, vol. 5, no. 4, p. 65, 2022. [Online]. Available: https://doi.org/10.3390/asi5040065

[2] F. Zhao, Z. Chen, X. Wang, and Y. Zhang, "Recent advances of digital twin in vehicle and its thermal management: A review," *Journal of Energy Storage*, vol. 64, p. 107108, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X23000774

[3] S. Sharma and P. Singh, "Digital twin technology for thermal management in automotive applications," *Journal of Manufacturing Systems*, vol. 74, pp. 328–342, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214157X24006385