Aalto University
School of Electrical
Engineering

INSIDE
Industry Association

SEASHINE

Safe Intelligent Agent to optimize ship energy management

Udayanto Dwi Atmojo (Staff Scientist, Aalto University)

Udayanto.Atmojo@aalto.fi

INSIDE Connect 2025

Funded by
the European Union

ENFIELD Enfield: European Lighthouse to Manifest Trustworthy and Green AI

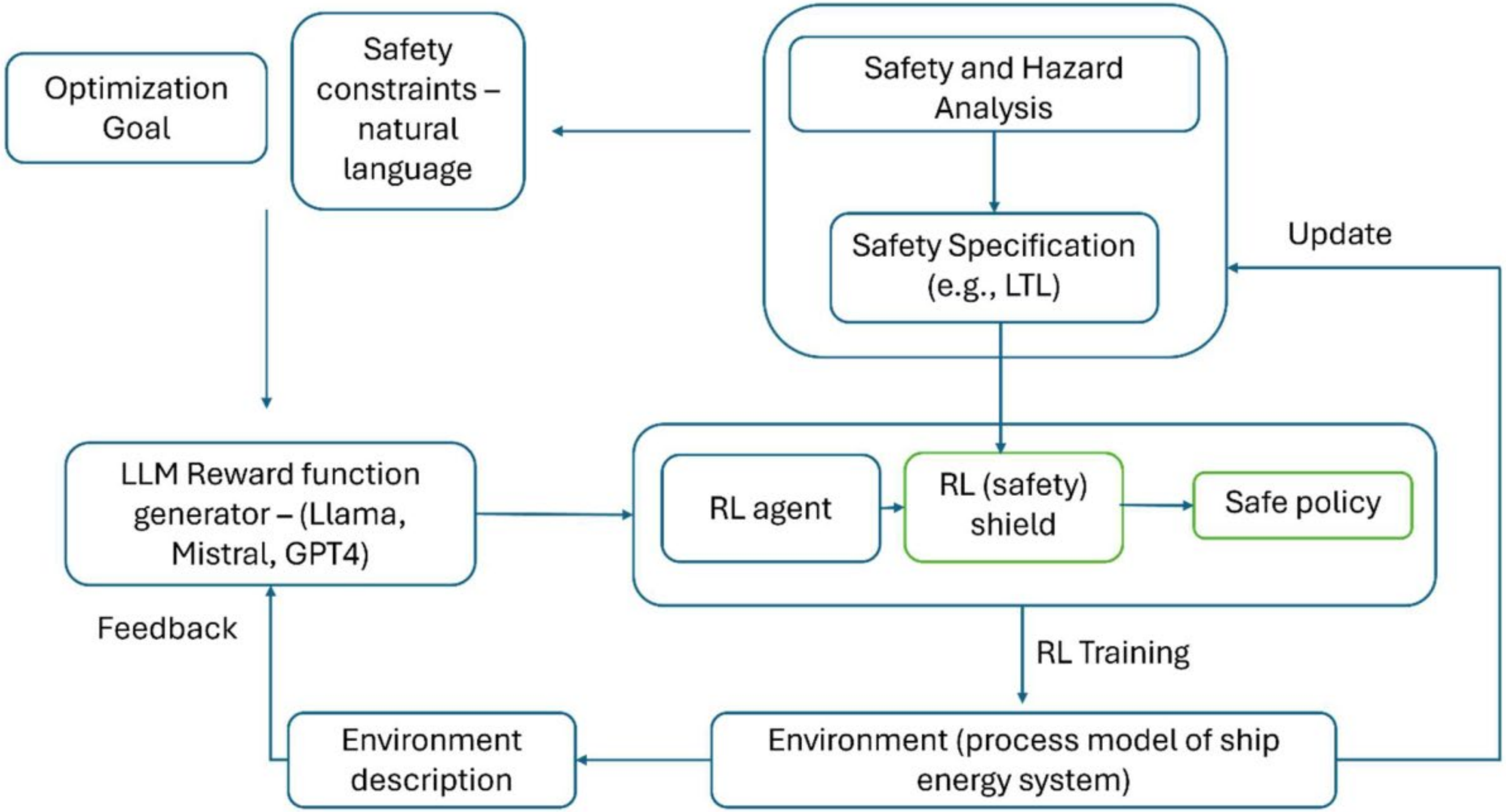
Problems and challenges

- Cruise ship industry needs to adapt to increasingly strict international environmental regulations – International Maritime Organization
- Heating, Ventilation, Air-conditioning systems are one of the most significant consumers of energy on cruise ships, but are hard to optimize
- Reinforcement Learning (RL) is capable of such optimization, however it is inherently hard to ensure its safety

Project, Objectives

- Utilize LLMs for safe and robust RL reward function generation
- Develop a safety “shield” that prevents the RL agent from executing unsafe actions
- Develop and validate safe and robust RL agent for ship energy management

Approach

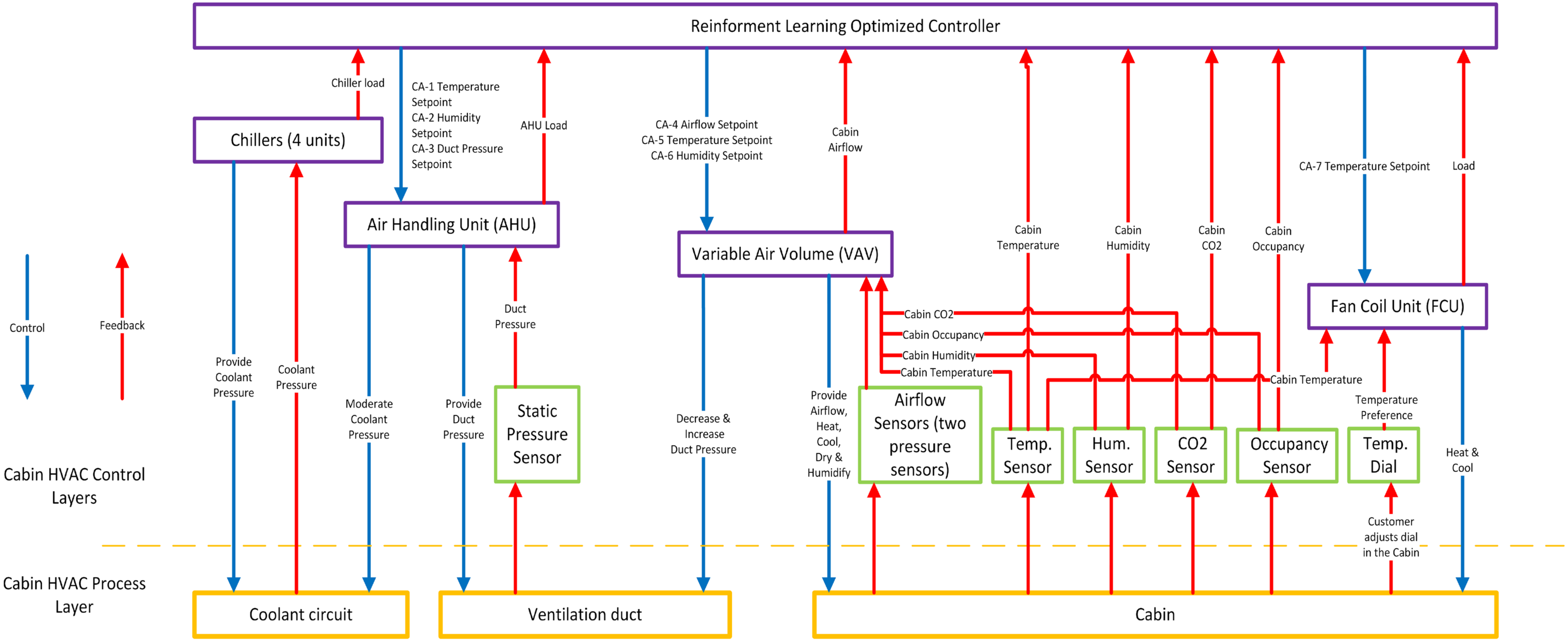

Optimization Goal
Safety constraints – natural language
Safety and Hazard Analysis
Safety Specification (e.g., LTL)
Update
LLM Reward function generator – (Llama, Mistral, GPT4)
RL agent
RL (safety) shield
Safe policy
Feedback
RL Training
Environment description
Environment (process model of ship energy system)

LLM reward function generation results

- LLMs are quite capable of shaping reward function
- Outputted reward function was functionally correct since the first iteration

```python
def reward(self, obs, action=None, prev_action=None):
    energy, sensor_data = self.surrogate_model.exrtact_output(obs, self.timestep)
    true_action = self.surrogate_model.denormalize_data(action.unsqueeze(0))[0]
    true_prev_action = self.surrogate_model.denormalize_data(prev_action.unsqueeze(0))[0]

    reward = 0.0
    cost = 0.0

    # --- Energy Reward (still primary objective) ---
    reward -= 0.01 * (energy / 1e6)  # Energy in MWh

    # --- Safety Violations (binary check) ---

    temperatures = sensor_data[:, 0]
    humidities = sensor_data[:, 3]
    airflow = true_action[4]  # in L/m²/s

    # Check for any violation
    temp_violation = (temperatures > 25.0).any() or (temperatures < 22.0).any()
    humidity_violation = (humidities > 0.55).any() or (humidities < 0.2).any()
    airflow_violation = airflow < 0.0667  # L/s threshold

    violated = temp_violation or humidity_violation or airflow_violation

    # Apply single heavy cost if any constraint is violated
    if violated:
        cost += 10.0  # Heavy penalty for *any* violation
    else:
        reward += 0.1  # Small bonus if no violation this step

    return reward, cost
```

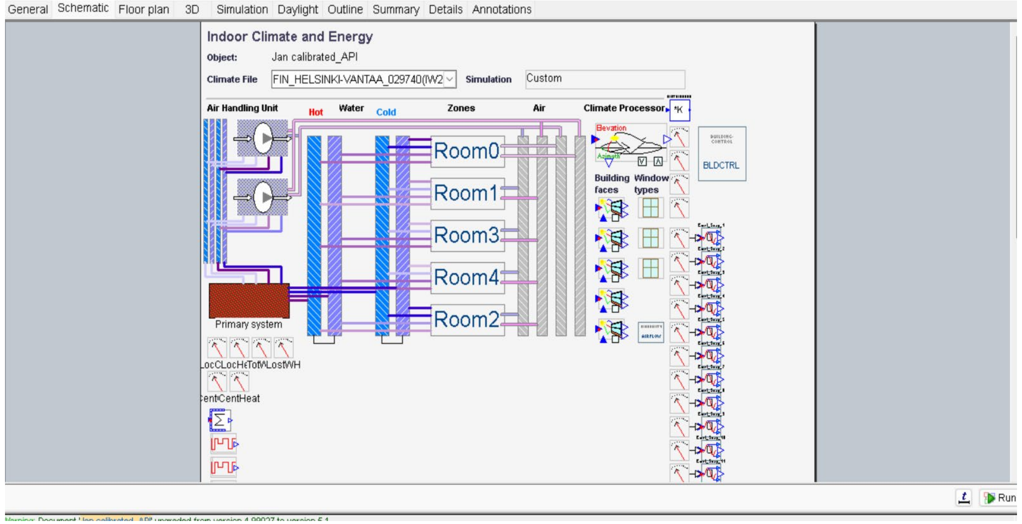
STPA analysis and safety shield

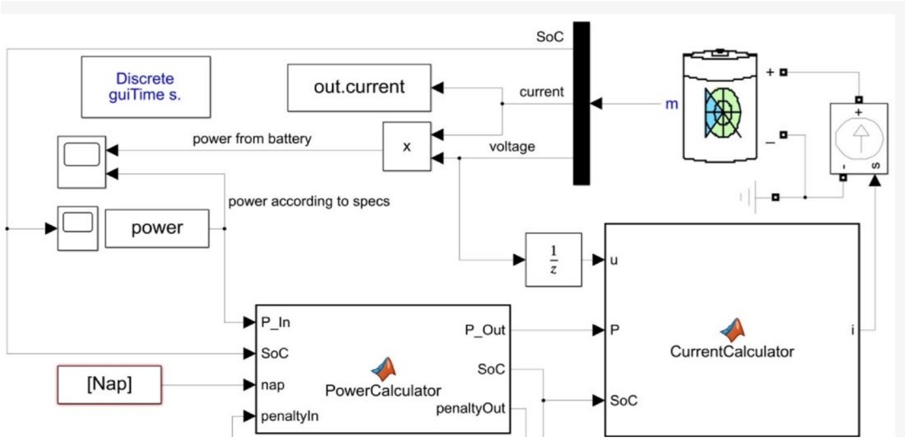


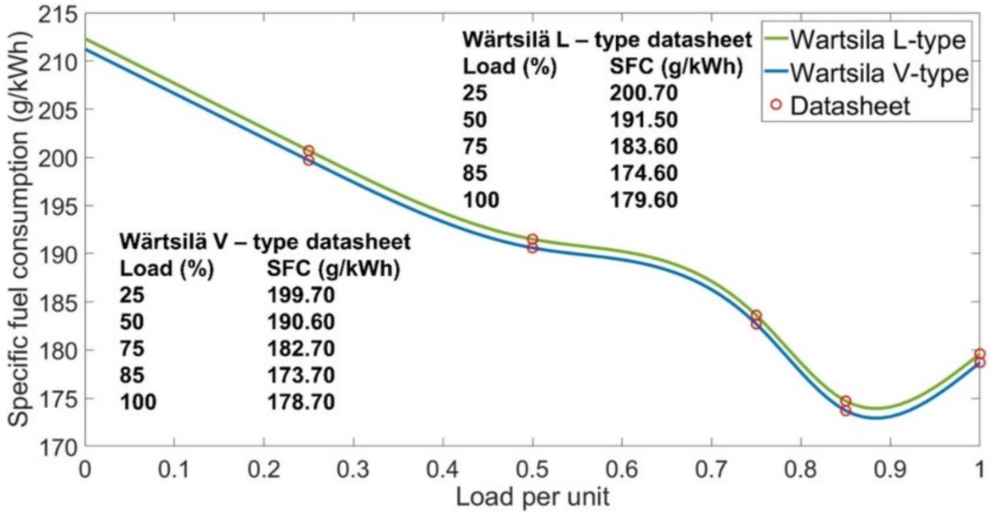
- Shield implementation includes a function that returns which constraints have been violated.
- Blocks actions that lead to a violation and requests a new action.

White box simulation model of the ship energy system

- Cruise ship Hotel with HVAC and Lighting (Ida ICE)
- Battery storage model (Simulink)
- Generation (Load curve)
- High fidelity data but slow







Safe RL Agent - results

- Environment based on black-box model
- Discrete action space (10 bins)
- Implemented:
  - PPO (baseline)
  - Penalized PPO (P3O) (safe RL)
- Goal: Minimize energy & safety violations

TABLE II
INITIAL RESULTS OF REINFORCEMENT LEARNING EXPERIMENTS

Algorithm	Power consumption	Safety violations
Open loop control	7.0 ± 2.5 MWh	388.5 ± 151
PPO	3.6 ± 1.4 MWh	405.6 ± 121.5
P3O	6.7 ± 3.3 MWh	**167.0 ± 139.8**

TABLE III
INITIAL RESULTS OF INTEGRATING THE SAFETY SHIELD AND THE LLM-BASED REWARD FUNCTION

Algorithm	Power consumption	Safety violations
P3O	6.7 ± 3.3 MWh	167.0 ± 139.8
P3O + LLM RF	10.9 ± 4.6 MWh	72.1 ± 83.7
P3O + shield	6.4 ± 3.0 MWh	64.4 ± 55.5
P3O + shield + LLM RF	11.5 ± 4.8 MWh	**48.9 ± 47.7**

Kiitos
aalto.fi