SHIFT_simulator_io_interface

SHIFT Simulator — Input/Output Interface Sheet

Owner: AI & Control Systems cluster
Status: Draft v0.1
Purpose: Define the contract between the SHIFT microgrid simulator and everything that talks to it (the AI/RL policy, the rule-based baseline, the validation/benchmark layer, and the Economics team's downstream cost & CO_2 accounting).

> This sheet covers **what the simulator consumes** (configuration, exogenous time-series, control actions) and **what it returns** at every timestep. It is a living document — items flagged `TBD` need a decision from the cluster or whoever.

---

0. Conventions

Timestep

- Symbol: `dt`
- Default assumption: **1 hour** (matches typical PV/load profiles and the 1-year benchmarking horizon in the project brief). `TBD` whether a sub-hourly step is needed for battery fast-response behaviour. (I did 1 hour for this cause 15 mins time steps will be quite compute heavy for RL training, so we do need to talk about that).

Sign conventions

Quantity	Positive means	Negative means
`P_battery`	discharging (supplying the bus)	charging (absorbing from the bus)
`P_grid`	importing from grid	exporting to grid
`n_H2_net`	net production into tank	net consumption from tank

All other power values (`P_PV`, `P_electrolyser`, `P_fuelcell`, `P_load`) are **non-negative** by definition.

Power balance (invariant the simulator must satisfy each step)

```
P_PV_used + P_fuelcell + P_grid + P_battery
   = P_load_served_critical + P_load_served_noncritical + P_electrolyser
```

Any residual after the policy's action is absorbed by **unmet load** (critical first is *not* a sensible policy — critical must always be prioritised; see §4d).

Units

SI throughout. Power in **W**, energy in **Wh**, mass in **kg**, mol in **mol**, time in **s** (or **h** if hourly steps). `TBD` to lock this down before code is written. (I don't know the conventions for this, I let AI handle this, not an electrical engineer sorry).

---

1. Static Configuration Inputs

Set once per simulation run. These are the constants Robert's doc lists in **bold**.

1.1 Electrolyser

Symbol	Name	Unit	Notes
`N_c_ele`	Number of cells	—	
`mu_F`	Faraday efficiency	—	Usually difficult to find; fall back to literature average (~0.95–0.99 for PEM)
`P_ele_max`	Rated / max power	W	
`P_ele_min`	Minimum operating power	W	Below this the electrolyser shuts off (efficiency cliff)
`I_ele_min`	Minimum operating current	A	Alternative to `P_ele_min`
`cal_H2`	H_2 production calibration factor	—	If we end up reading H_2 flow from measured data

1.2 Fuel cell

Symbol	Name	Unit	Notes
`N_c_fc`	Number of cells	—	
`utilisation_fc`	H_2 utilisation	%	
`P_fc_max`	Maximum output power	W	Project spec: 100 kW
`V_fc_min`, `V_fc_max`	Operating voltage range	V	

1.3 Hydrogen tank

Symbol	Name	Unit	Notes
`V_H2_max`	Maximum stored volume	L (or kg)	Project spec: up to 200 kg total across two tanks
`V_H2_init`	Initial fill level	L (or kg)	
`T_tank`	Operating temperature	K	`TBD` — isothermal assumption likely fine
`p_tank`	Operating pressure	atm (or bar)	`TBD` — isobaric vs. ideal-gas model

1.4 PV

Symbol	Name	Unit	Notes
`A_PV`	Total panel area	m^2	Project spec: 2,300 m^2
`mu_PV`	Panel efficiency	—	
`P_PV_max`	Inverter / rated cap	W	Optional; if set, clipping occurs above this

1.5 Battery

Symbol	Name	Unit	Notes
`E_rated`	Rated energy capacity	Wh	
`Q_rated`	Rated charge capacity	Ah	

Symbol	Name	Unit	Notes
`P_batt_charge_max`	Max charge power	W	
`P_batt_discharge_max`	Max discharge power	W	
`SoC_init`	Initial state of charge	— (0–1)	
`SoC_min`, `SoC_max`	Operating window	— (0–1)	E.g. 0.1–0.9
`eta_batt_ch`, `eta_batt_dis`	Round-trip efficiencies	—	Often split into charge & discharge

1.6 Load

Symbol	Name	Unit	Notes
`V_load`	Bus voltage	V	Constant, per Robert's note

1.7 Grid

Symbol	Name	Unit	Notes
`P_grid_max`	Max connection capacity (import & export)	W	
`V_grid`	Bus voltage	V	Constant, per Robert's note

---

2. Per-Timestep Inputs

2.1 Exogenous time-series (read from data; not under policy control)

Symbol	Name	Unit	Source
`G(t)`	PV generated power	W	Forecasted
`L_crit(t)`	Critical hospital load demand	W	ICU, life-support, etc. — must always be served
`L_noncrit(t)`	Non-critical hospital load demand	W	Lighting, HVAC, admin

Symbol	Name	Unit	Source
`grid_on(t)`	Grid availability (1/0)	—	Models outages
`price(t)`	Grid electricity price	€/kWh	Import & export prices — `TBD` whether they differ
`CO2_int(t)`	Grid carbon intensity	kg CO_2 / kWh	For emissions accounting

2.2 Control inputs (from the AI/RL policy or rule-based baseline)

At each timestep the controller emits **setpoints**, not direct power flows — the simulator clips them to physical limits and reports actuals (I have no clue what this means, the AI corrected me for this).

Symbol	Name	Unit	Range
`u_ele(t)`	Electrolyser power setpoint	W	`[0, P_ele_max]`; values in `(0, P_ele_min)` round to 0 (off)
`u_fc(t)`	Fuel cell power setpoint	W	`[0, P_fc_max]`
`u_batt(t)`	Battery setpoint (signed)	W	`[-P_batt_charge_max, P_batt_discharge_max]`
`sw_ele(t)`	Electrolyser ON/OFF	1/0	Optional — can be folded into `u_ele = 0`
`sw_fc(t)`	Fuel cell ON/OFF	1/0	Optional — can be folded into `u_fc = 0`

Note on the grid: the grid is typically treated as the **slack bus** — whatever the power-balance residual is after all other components act, the grid absorbs (within `P_grid_max` and `grid_on(t)`). The RL action space probably does *not* include the grid directly (Same for this)

---

3. Per-Timestep Outputs (what the simulator returns)

3.1 State variables (carried forward to next step)

Symbol	Name	Unit
`SoC(t)`	Battery state of charge	— (0–1)

Symbol	Name	Unit
`H2_level(t)`	H_2 stored in tank	mol (or kg)
`T_tank(t)`	Tank temperature	K *(only if non-isothermal model)*
`p_tank(t)`	Tank pressure	bar *(if modelled)*

3.2 Power flows

Symbol	Name	Unit
`P_PV_available(t)`	PV power available given irradiance	W
`P_PV_used(t)`	PV power actually consumed	W
`P_PV_curtailed(t)`	PV potential that was thrown away	W
`P_ele(t)`	Actual electrolyser consumption	W
`P_fc(t)`	Actual fuel cell output	W
`P_batt(t)`	Actual battery flow (signed)	W
`P_grid(t)`	Actual grid flow (signed)	W
`P_load_served_crit(t)`	Critical load served	W
`P_load_served_noncrit(t)`	Non-critical load served	W

3.3 Mass flows (hydrogen)

Symbol	Name	Unit
`n_H2_prod(t)`	H_2 produced by electrolyser	mol/s (or kg/h)
`n_H2_cons(t)`	H_2 consumed by fuel cell	mol/s (or kg/h)
`n_H2_net(t)`	Net flow into tank	mol/s

3.4 KPIs and accounting (the heart of the task)

Symbol	Name	Unit	Notes
`unmet_crit(t)`	Unserved critical load	W	**Must be near-zero** — primary resilience KPI
`unmet_noncrit(t)`	Unserved non-critical load	W	
`curtailment(t)`	= `P_PV_curtailed(t)`	W	Restated as a KPI for clarity

Symbol	Name	Unit	Notes
`cost(t)`	Operating cost of this step	€	`TBD` — see §5
`CO2(t)`	Emissions of this step	kg CO_2	`TBD` — see §5

3.5 Diagnostic / efficiency (recommended; ties directly to the AI-improvements doc)

The "AI improvements" doc flagged ~30–40% electrolyser losses and ~40% fuel-cell losses as the biggest optimisation targets. If the simulator returns the per-step efficiencies, the RL reward and the post-hoc analysis can both attack those losses directly.

Symbol	Name	Unit
`eta_ele(t)`	Electrolyser efficiency this step	—
`eta_fc(t)`	Fuel cell efficiency this step	—
`eta_PV(t)`	Effective PV efficiency this step	—
`eta_batt(t)`	Battery round-trip-equivalent	—

---

4. Priority rules the simulator should enforce

These aren't I/O but they're contract-level — the policy needs to know what the simulator will do with a setpoint that violates physics or safety.

1. **Critical load is sacred.** If the policy's setpoints leave critical load unserved while the battery has charge, fuel cell can run, or grid is available, the simulator should override the policy in that order and report the override (e.g. via a flag in the output) so it shows up in training.
2. **State limits are hard.** `SoC ∈ [SoC_min, SoC_max]`, `H2_level ∈ [0, V_H2_max]`. Setpoints that would breach these are clipped; the difference shows up in `P_batt(t)` vs `u_batt(t)`.
3. **Electrolyser dead-zone.** Setpoints in `(0, P_ele_min)` are rounded to 0.
4. **Grid outage.** If `grid_on(t) = 0`, `P_grid(t) = 0` regardless of slack residual; the residual becomes `unmet_*` or `curtailment`.

---

5. Open questions (need answers before v1.0)

- **Cost & CO_2 scope.** Currently flagged TBD in §3.4. If Economics handles these downstream, the simulator just needs to return `P_grid(t)` + the exogenous `price(t)`, `CO2_int(t)`, and they compute cost/CO_2 themselves. If the simulator computes them, it also needs to know if there's an O&M cost per kWh through the electrolyser/fuel cell, a degradation cost, etc.
- **Tank model.** Isothermal + ideal gas, or do we need temperature and pressure dynamics?
- **Battery degradation.** Modelled or ignored?
- **Sub-hourly dynamics.** Does the battery's fast-response role require `dt < 1 h`? If so, mixed time-scale handling between battery (minutes) and electrolyser (hours) becomes a design question.
- **Action space format.** Continuous setpoints (`u_ele ∈ [0, P_ele_max]`) vs. discrete buckets — affects what the RL team expects as input.
- **Forecast vs. realised exogenous inputs.** The policy may see a *forecast* of `G(t+1)` and `L(t+1)`; the simulator advances with the *realised* values. Need to decide if the simulator delivers both or only realised.
- **Critical vs non-critical split — data availability.** Confirm with the hospital data source whether the load profile can actually be split, or whether we need a proxy fraction (e.g. assume X% of total load is critical).
- Should the action space *structurally* prevent simultaneous electrolyser and fuel cell operation (mode-switch / hybrid action space), or allow them independently and let the reward penalise simultaneous operation? Has efficiency, sample-efficiency, and policy-expressiveness trade-offs.
