Optimization techniques for energy systems: the Haeolus case

Luigi Glielmo, Professor of Automatic Control
with the support of *Muhammad Bakr Abdelghany*, PhD candidate

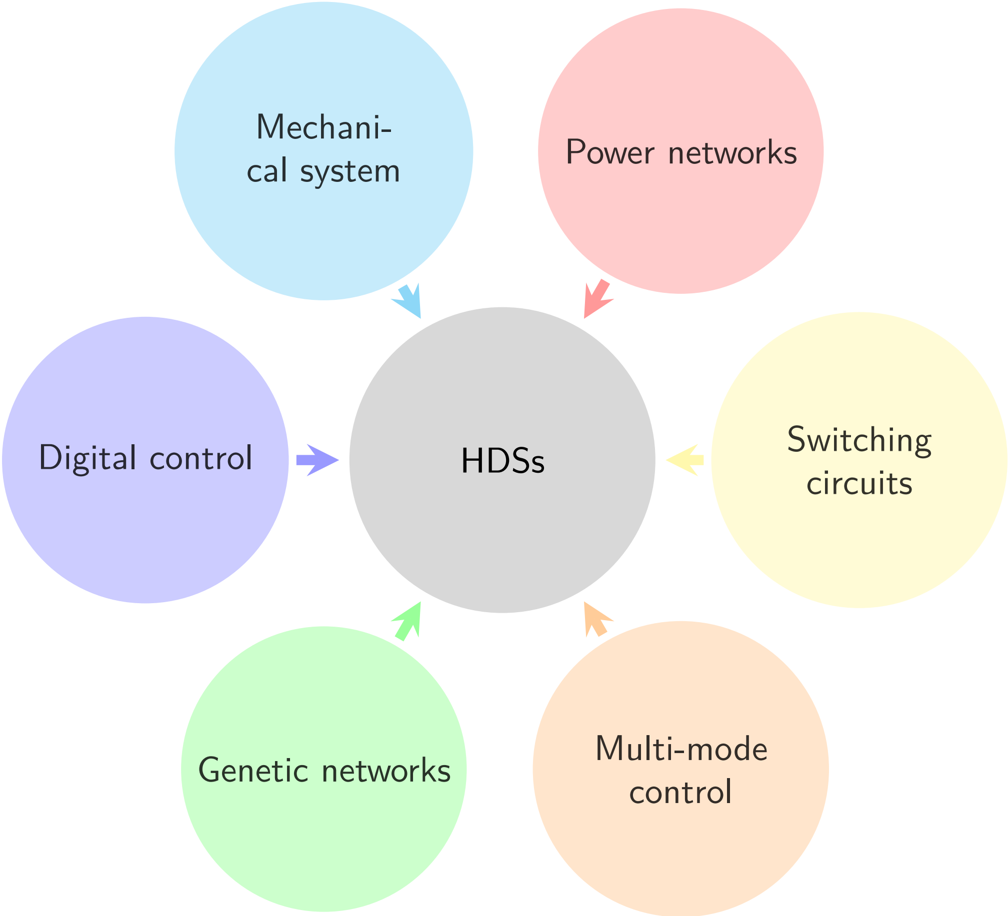
GRACE, the Group for Research on Automatic Control Engineering
Department of Engineering
Università del Sannio

July 1, 2021

Table of Contents

Scope of hybrid dynamical systems (HDSs) research

HDSs combine **continuous states** and **discrete states** dynamics.



Motivation and approach

Common features in applications:

- Variables changing continuously (e.g., physical quantities) and discretely (e.g., logic variables, resetting timers);
- Abrupt changes in dynamics (changes in the environment, control decisions, communication events, or failures).

Driving questions:

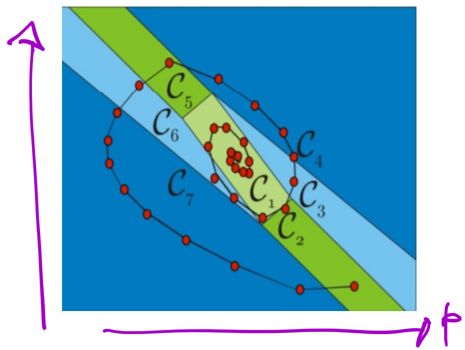
- How can we systematically design such systems with provable robustness to uncertainties arising in real-time applications?

Approaches:

- capture continuous and discrete behavior using dynamical modeling;
- analysis of stability and control design using control theoretical tools;
- numerical (and possible experimental) validation.

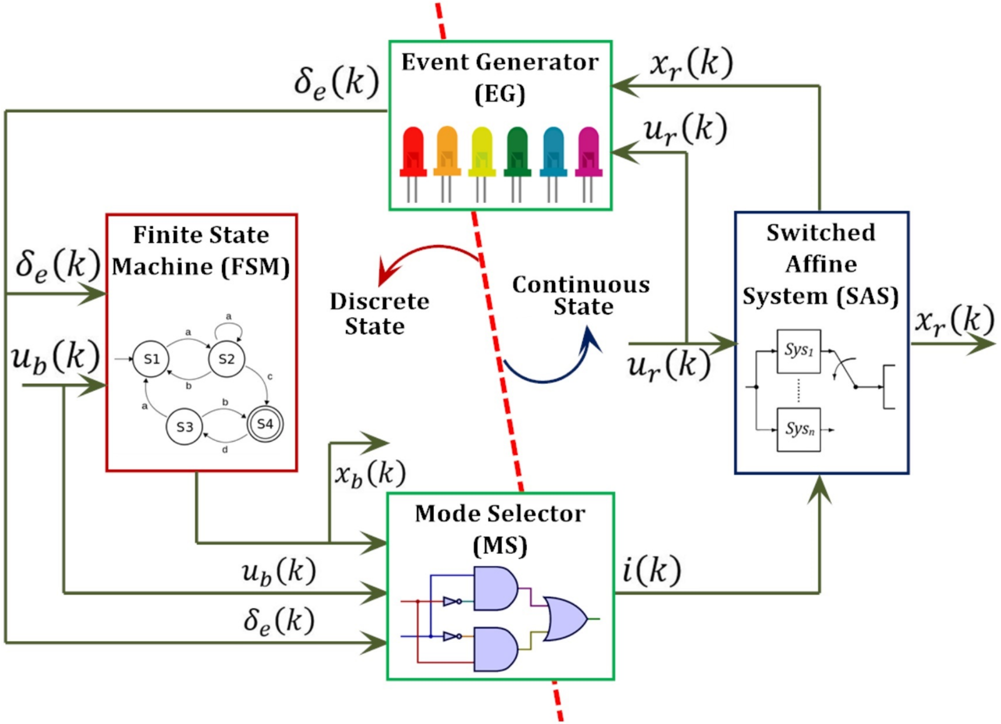
Hybrid dynamical systems

- Descriptive enough to capture the behavior of the system
  - continuous states dynamics (physical systems);
  - logic components (switches, automata);
  - interconnection between logic and dynamics.
- Simple enough for solving analysis and synthesis problem.
- A wide range of system can be modeled within such framework
  - Discrete hybrid automata;
  - Piecewise affine systems;
  - Mixed-logical dynamical (MLD) systems.



Discrete hybrid automata I

- DHA is a hybrid system representation that combines finite state machines (FSM) for discrete events, switched affine systems (SAS) for continuous evolution, event generator (EG), and mode selector (MS).



Discrete hybrid automata II

- An SAS is a set of linear systems switched by an integer variable:

$$x_r(k+1) = A_{i(k)}x_r(k) + B_{i(k)}u_r(k) + f_{i(k)},$$
$$y(k) = C_{i(k)}x_r(k) + D_{i(k)}u_r(k) + g_{i(k)},$$

  where
  - $x_r \in \mathcal{X}_r$ is the continuous state vector;
  - $u_r \in \mathcal{U}_r$ is the exogenous continuous input vector;
  - $y_r \in \mathcal{Y}_r$ is the continuous output vector;
  - $\{A, B, C, D\}$ is a tuple of state space matrices;
  - f and g are affine offset values, $i \in J \subseteq \mathbb{N}_0$ is the active mode.
- An EG is a mathematical relation that maps linear affine constraints conditions into logic values

$$\delta_e(k) = \mathcal{F}_e(x_r, u_r(k), k),$$
$$\mathcal{F}_e = \mathcal{X}_r \times \mathcal{U}_r \times \mathbb{N}_0 \to \mathcal{B} \subseteq \{0,1\}^{n_e},$$

  where

Discrete hybrid automata III

- δ_e is a binary coded variable describes a discrete event;
- $\mathcal{F}_e$ is a mapping relation in a linear hyperplane.

- An automaton is a transition relation among discrete value finite states according to logical event or guard condition. The transition relation of the FSM in the DHA is defined as:

$$x_b(k+1) = \mathcal{F}_a(x_b, u_b(k), \delta_e(k)),$$
$$\mathcal{F}_a : \mathcal{X}_b \times \mathcal{U}_b \times \mathcal{B} \rightarrow \mathcal{B} \subseteq \{0,1\}^{n_e},$$

where

- $x_b \in \mathcal{X}_b$ and $u_r \in \mathcal{U}_r$ are the discrete state and input vectors;
- $\mathcal{F}_a$ is a deterministic transition relation.

- An MS is a mathematical relation that maps linear affine constraints conditions into logic values:

$$i(k) = \mathcal{F}_M(x_b, u_b(k), \delta_e(k)),$$
$$\mathcal{F}_M : \mathcal{X}_b \times \mathcal{U}_b \times \mathcal{B} \rightarrow \mathcal{B} \subseteq J.$$

Mixed logical dynamical (MLD) system I

- Goal: describe hybrid system in form compatible with optimization software:
  - continuous and Boolean variables;
  - linear equalities and inequalities.
- Idea: associate to each Boolean variable s_i a binary integer variable δ_i:

$$[s_i = \text{true}] \iff \{\delta_i = 1\}, \quad [s_i = \text{false}] \iff \{\delta_i = 0\}$$

  and embed them into a set of constraints as linear integer inequalities.
- Two main steps:
  1. translation of logic constraints into linear integer inequalities;
  2. translation of continuous and logical constraints into linear mixed-integer relations.
- Final result: a compact model with linear equalities and inequalities involving real and binary variables.

Mixed logical dynamical (MLD) system II

- By converting logic relations into mixed-integer linear inequalities, a DHA can be rewritten as the Mixed Logical Dynamical (MLD) system

$$
\begin{aligned}
x(k+1) &= Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k), \\
y(k) &= Cx(k) + D_1 u(k) + D_2 \delta(k) + D_3 z(k), \\
E_2 \delta(k) + E_3 z(k) &\leq E_1 u(k) + E_4 x(k) + E_5,
\end{aligned}
$$

  where

  - $x \in \mathbf{R}^{n_r} \times \{0,1\}^{n_b}$ is the continuous and discrete states vector;
  - $u \in \mathbf{R}^{m_r} \times \{0,1\}^{m_b}$ is the continuous and discrete input vector;
  - $y \in \mathbf{R}^{p_r} \times \{0,1\}^{p_b}$ is the continuous and discrete output vector;
  - $z \in \mathbf{R}^{n_z}$ and $\delta \in \mathbf{R}^{n_d}$ are the continuous and binary auxiliary variables mode of the system.

- The translation to MLD can be automatized.
- MLD models allow solving MPC, state estimation, and fault detection problems via mixed-integer programming.

Mixed logical dynamical (MLD) system III

- MLD equivalences are defined as

Relation	Logic	MLD Inequalities
NOT ($\sim$)	$\sim s$	$1 - \delta$
AND ($\wedge$)	$s_3 = s_1 \wedge s_2$	$\delta_3 \leq \delta_1$ $\delta_3 \leq \delta_2$ $\delta_3 \geq \delta_1 + \delta_2 - 1$
OR ($\vee$)	$s_3 = s_1 \vee s_2$	$\delta_3 \geq \delta_1$ $\delta_3 \geq \delta_2$ $\delta_3 \leq \delta_1 + \delta_2$
IMPLY ($\implies$)	$s_1 \implies s_2$	$\delta_1 - \delta_2 \leq 0$
IFF ($\iff$)	$s_1 \iff s_2$	$\delta_1 = \delta_2$
	$[s = \text{true}] \iff [f(\mathbf{r}) \leq 0]$	$f(\mathbf{r}) \leq M(1-\delta)$ $f(\mathbf{r}) \geq \epsilon + (m - \epsilon)\delta$
IF-then-else	$r_2 = \begin{cases} r_1 & \text{if } s = \text{true} \\ 0 & \text{if } s = \text{false} \end{cases}$	$r_2 \leq M\delta$ $r_2 \geq m\delta$ $r_2 \geq r_1 - M(1-\delta)$ $r_2 \leq r_1 + m(1-\delta)$

$\delta_1 \leq \delta_2$

where M is a sufficiently large number (for the problem at hand), m is a sufficiently small number (for the problem at hand), e.g. $m = -M$, and $\epsilon > 0$ is a small tolerance.

Piecewise affine (PWA) system I

- another common class for hybrid systems.
- equivalent to MLD systems.
- have the capability to approximate nonlinear dynamics.
- representation has a set of polyhedral partitions.
- PWA system with bounded states and inputs and s regions

$$
\begin{aligned}
x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)}, \\
y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)}, \\
i(k) &= \text{such that } [x(k)\ u(k)]' \in \mathcal{P}_i,
\end{aligned}
$$

with

- $\mathcal{P}_i = \{[x(k)\ u(k)]' : H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)}\}$;
- $i \in J \subseteq \mathbb{N}_0$ is the current active polyhedron.

Piecewise affine (PWA) system II

- introduce S Boolean variables s_i, $i = 1, \ldots, S$, the corresponding binary variables δ_i and the logic constraints

$$[s_i = \text{true}] \iff [\delta_i = 1]$$
$$\bigoplus_i s_i = \text{true} \iff \sum_i \delta_i = 1$$

- introduce auxiliary real vectors z_i and w_i defined by if-then-else rules

$$z_i = \begin{cases} A_{i(k)} + Bu + f_i, & \text{if } \delta_i = 1 \\ 0, & \text{otherwise} \end{cases} \qquad w_i = \begin{cases} C_i x + D_i u + g_i, & \text{if } \delta_i = 1 \\ 0, & \text{otherwise;} \end{cases}$$

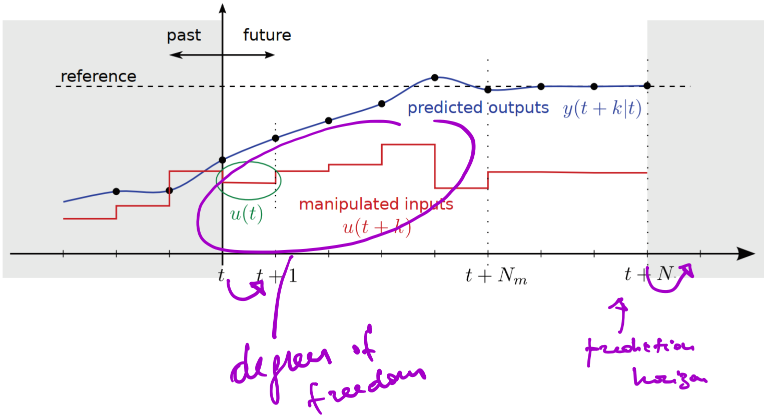
- convert the relations above into mixed-integer inequalities;
- update the state and output equations

$$\begin{cases} x(k+1) = \sum_i z_i, \\ y(k) = \sum_i w_i. \end{cases}$$

Model predictive control (MPC)

MPC is a control method for handling input and state constraints within an optimal control setting.

Principle of predictive control



Why to use MPC?

- It handles multivariable interactions.
- It handles input and state constraints.
- It can push the plants to their limits of performance.
- It is easy to explain to operators and engineers.

$$\underline{x}_1 \leq x_1(t) \leq \bar{x}_1$$

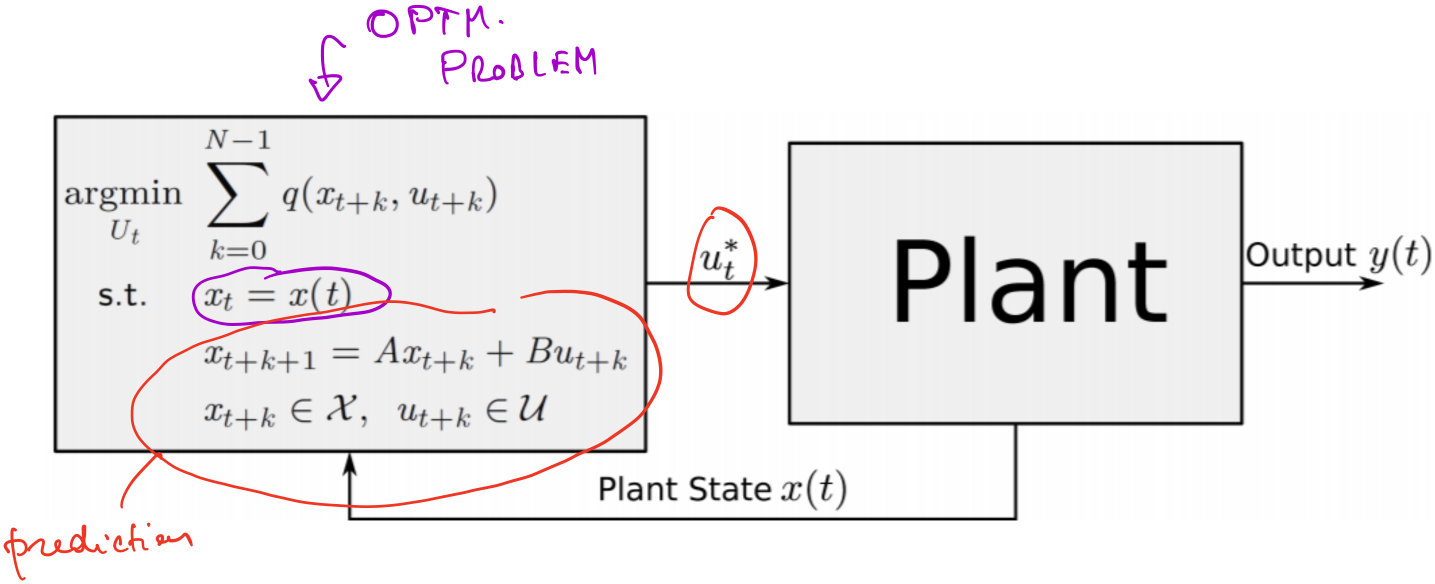
MPC: Mathematical formulation

$$
\begin{aligned}
\boldsymbol{u}_t^{N-1\,*} &= \arg\min_{\boldsymbol{u}_t^{N-1}} \sum_{k=0}^{N-1} q\left(x_{t+k}, u_{t+k}\right) \\
\text{s.t. } x_{t+k+1} &= q\left(x_{t+k}, u_{t+k}\right), \\
x_{t+k} &\in \mathcal{X}_{t+k}, \\
u_{t+k} &\in \mathcal{U}_{t+k}.
\end{aligned}
$$

The problem is defined by

- **objective** that is minimized
  - e.g., distance from origin, sum of squared/absolute errors, costs,...
- internal **system model** to predict system behavior
  - e.g., linear, nonlinear, single-/multi-variable,...
- **constraints** that have to be satisfied
  - e.g., on inputs, outputs, states, and linear, quadratic,...

MPC: Mathematical Formulation



- At each sample time:
  - measure/estimate current state $x(t)$;
  - find the optimal input sequence for the entire planning window N:
    - $\boldsymbol{u}_t^{N-1*} = [u_t^*, \ldots, u_{t+N-1}^*]'$;
  - implement only the first control action u_t^*.

HAEOLUS PROJECT

- HAEOLUS – Hydrogen-Aeolic Energy with Optimised Electrolysers Upstream of Substation
    - Funding of 7 $M€$;
    - Produce a total of 120 tons of hydrogen by 2021.
- Raggovidda wind park in Varanger peninsula (Norway)
    - 45 MW built of 200 MW concession;
    - Bottleneck to main grid 95 MW;
    - Capacity Factor of 50%.

HAEOLUS CONTRIBUTION

Dynamic Plant Model

Designed and developed to implement multi-level controllers

Features

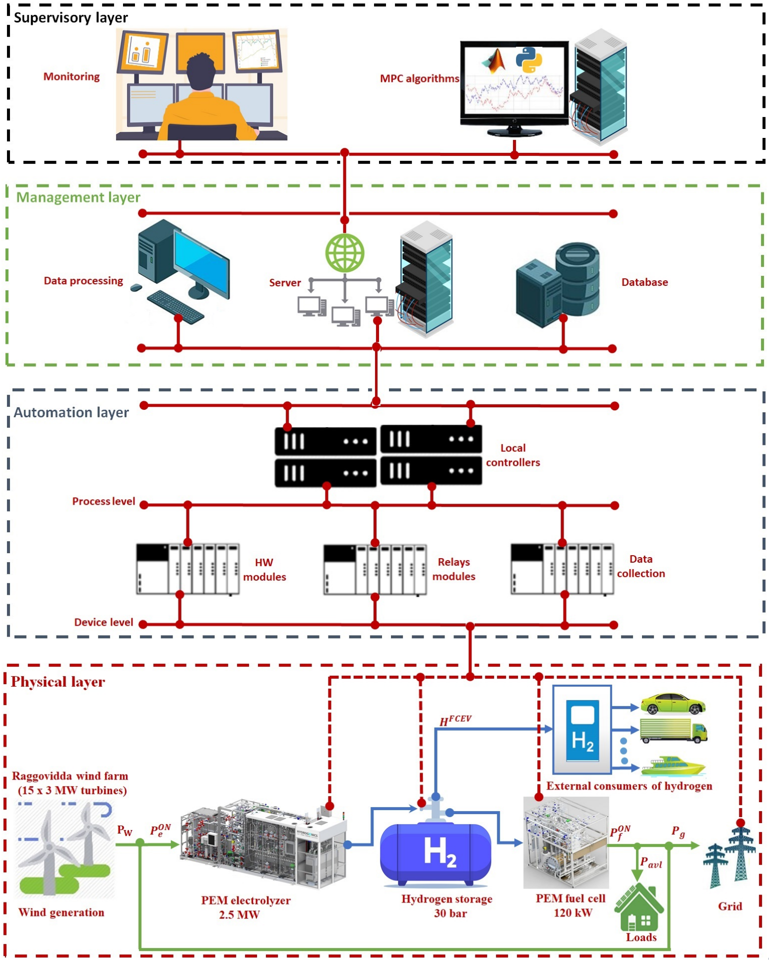
- A hybrid system
- Different time scales of physical and market phenomena

Plant and scenario for HAEOLUS project

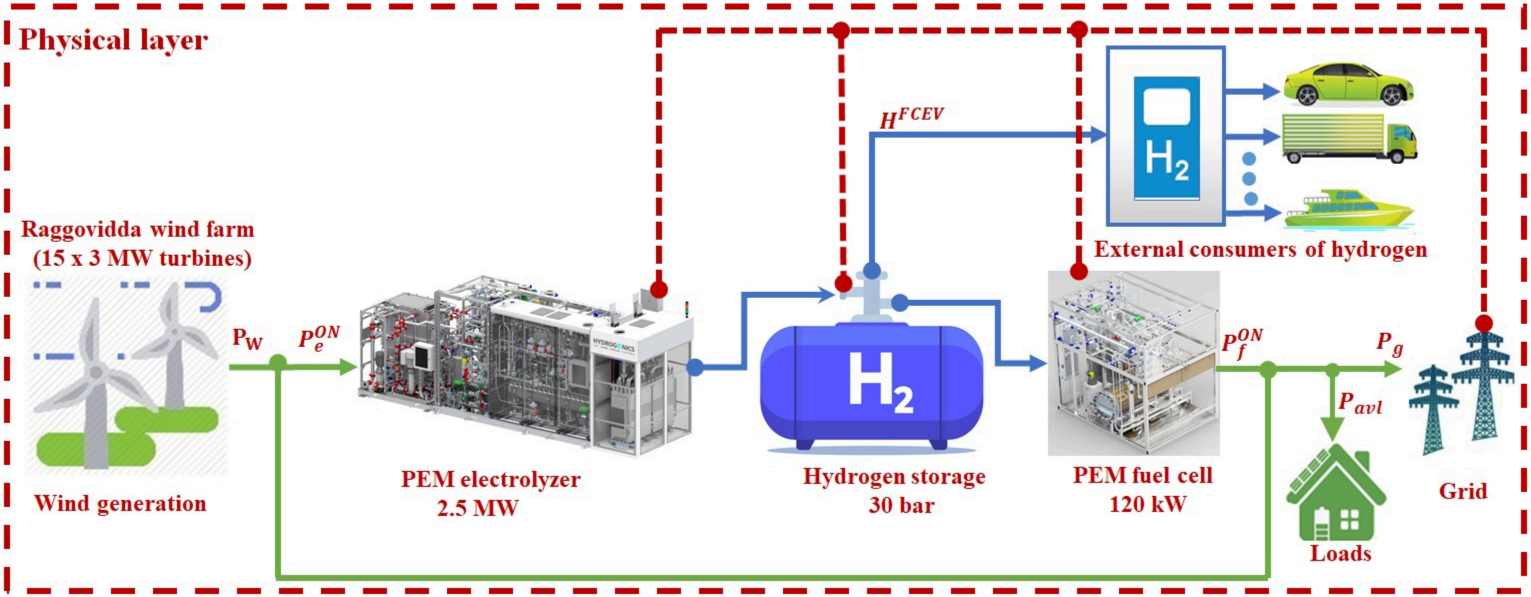
- The standard architecture with 4 layers:
    - supervisory layer;
    - management layer;
    - automation layer;
    - physical layer.

Plant and scenario for HAEOLUS project

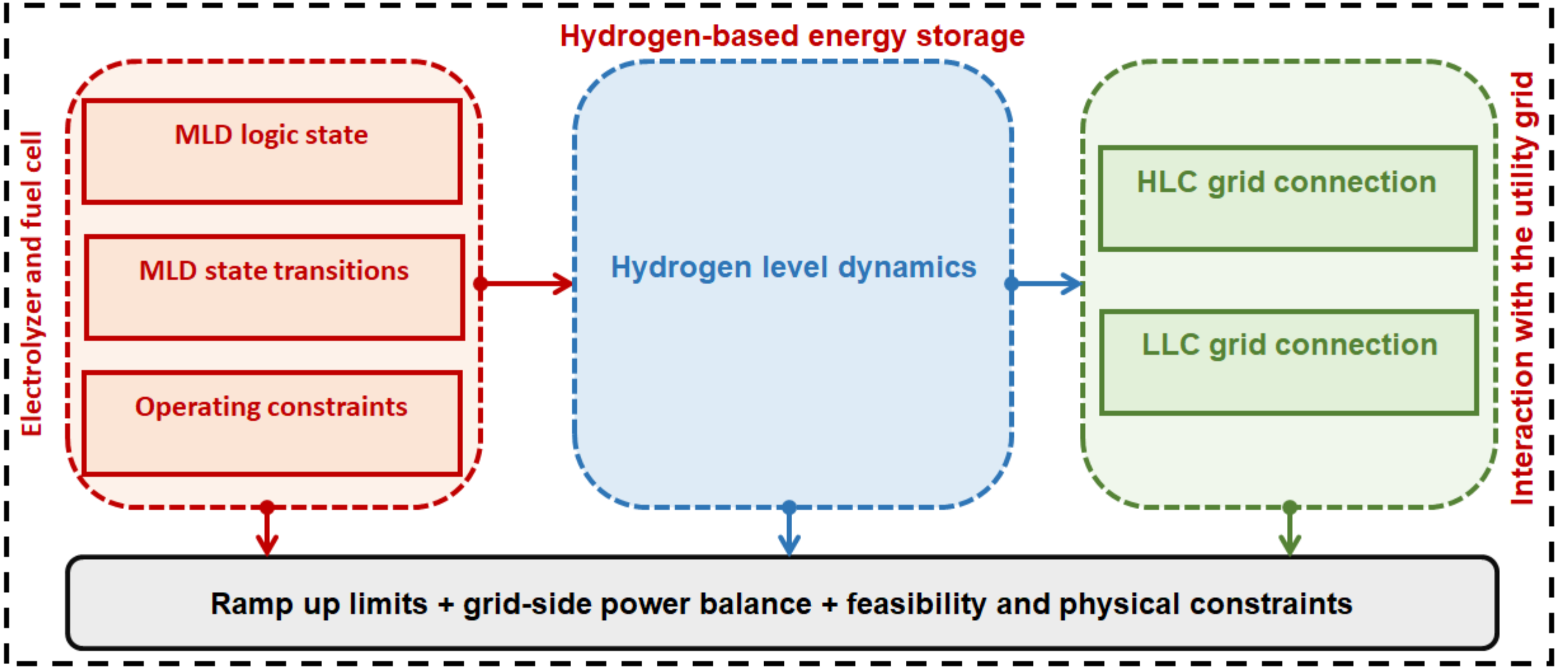
- The standard architecture with 4 layers:
  - supervisory layer;
  - management layer;
  - automation layer;
  - physical layer.



Physical layer of HAEOLUS project I



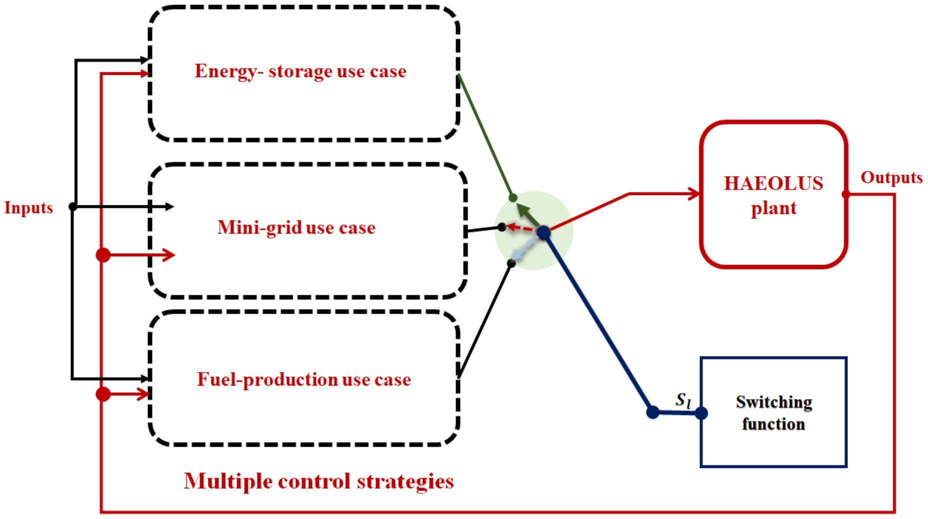
Physical layer of HAEOLUS project II



Alternative scenarios I

The International Energy Agency-Hydrogen Implementing Agreement (IEA-HIA) identified three different use cases regarding the possible operations for wind farm paired to a hydrogen-based storage system:

- Energy-storage operates for power smoothing.
- Mini-grid operates for demand side management:
  - islanded mode;
  - weakly connected mode.
- Fuel-production operates for hydrogen production.

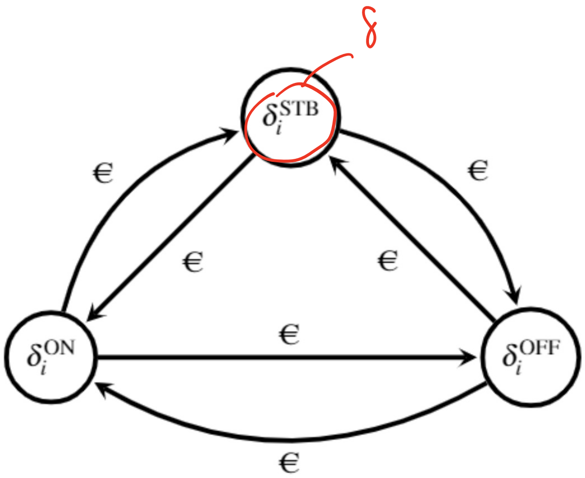


Alternative scenarios II

Control and optimization problems

- operate devices by means of logic commands (ON, OFF,STB);
- convert/electrify suitable amounts of energy/hydrogen;
- minimize costs for long term profitability;
- fulfill (scenario-dependent) constraints and requirements;
- rely on forecasts while managing the uncertainty.

MLD modeling of devices operational modes I



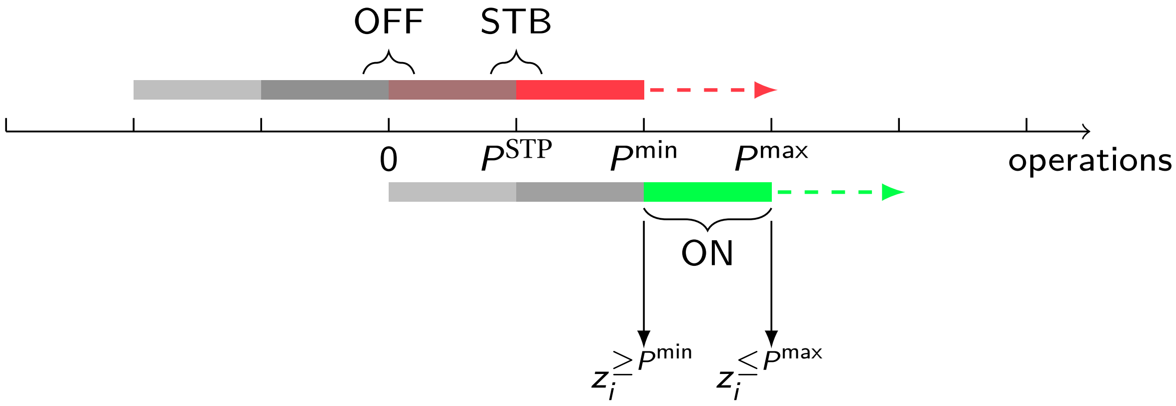
Dynamic Modeling

- ON/OFF/STB: modes of the devices (electrolyzer and fuel cell);
- δ_i^{ON}, δ_i^{OFF} and δ_i^{STB}: the devices discrete states;
- $\sigma_{\alpha_i}^{\beta}$: the state transitions of the devices;
- the arcs explain the state transitions.

MLD modeling of devices operational modes II

- The three logic discrete states of the devices ON, OFF, STB characterize the model.
- The states have been modeled with mutually exclusive logical variables:

$$\delta_i^{\mathrm{ON}}(t) + \delta_i^{\mathrm{OFF}}(t) + \delta_i^{\mathrm{STB}}(t) = 1.$$



MLD modeling of devices operational modes III

everything is LINEAR

MLD constraints of the ON state (δ^{ON})

$$z_i^{\geq P^{\min}}(t) = \begin{cases} 1 & P_i(t) \geq P^{\min} \\ 0 & P_i(t) < P^{\min} \end{cases}$$

$$z_i^{\leq P^{\max}}(t) = \begin{cases} 0 & P_i(t) > P^{\max} \\ 1 & P_i(t) \leq P^{\max} \end{cases}$$

$$-P_i(t) + P^{\min} \leq M(1 - z_i^{\geq P^{\min}}(t))$$

$$-P_i(t) + P^{\min} \geq \epsilon + (m - \epsilon) z_i^{\geq P^{\min}}(t)$$

$$P_i(t) - P^{\max} \leq M(1 - z_i^{\leq P^{\max}}(t))$$

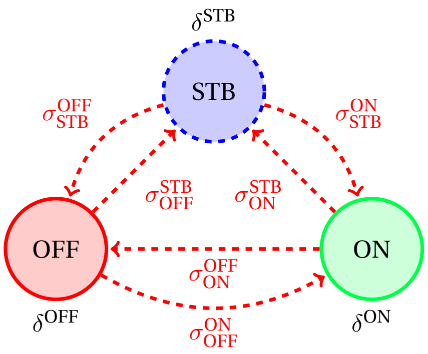
$$P_i(t) - P^{\max} \geq \epsilon + (m - \epsilon) z_i^{\leq P^{\max}}(t)$$

$P_i(t)$ can be linked to δ_i^{ON} with inequalities through $z_i^{\geq P^{\min}}$ and $z_i^{\leq P^{\max}}$

$$\delta_i^{\mathrm{ON}}(t) \implies z_i^{\geq P^{\min}}(t) \iff \delta_i^{\mathrm{ON}}(t) - z_i^{\geq P^{\min}}(t) \leq 0$$

$$\delta_i^{\mathrm{ON}}(t) \implies z_i^{\leq P^{\max}}(t) \iff \delta_i^{\mathrm{ON}}(t) - z_i^{\leq P^{\max}}(t) \leq 0$$

MLD constraints of the devices state transitions

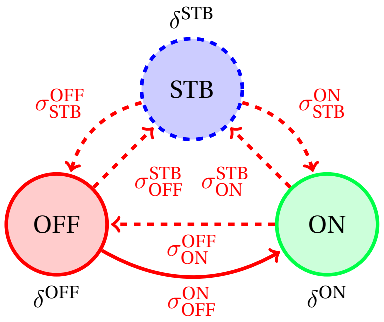


MLD modeling of devices operational modes V

MLD constraints of the devices state transitions

$$\sigma_{\text{OFF}}^{\text{ON}}(t) \iff \delta^{\text{OFF}}(k-1) \wedge \delta^{\text{ON}}(t)$$

$$\sigma_{\text{OFF}}^{\text{ON}}(t) \equiv \begin{cases} \sigma_{\text{OFF}}^{\text{ON}}(t) \leq \delta^{\text{OFF}}(k-1) \\ \sigma_{\text{OFF}}^{\text{ON}}(t) \leq \delta^{\text{ON}}(t) \\ \sigma_{\text{OFF}}^{\text{ON}}(t) \geq \delta^{\text{OFF}}(k-1) + \delta^{\text{ON}}(t) - 1 \end{cases}$$



Hydrogen tank dynamics I

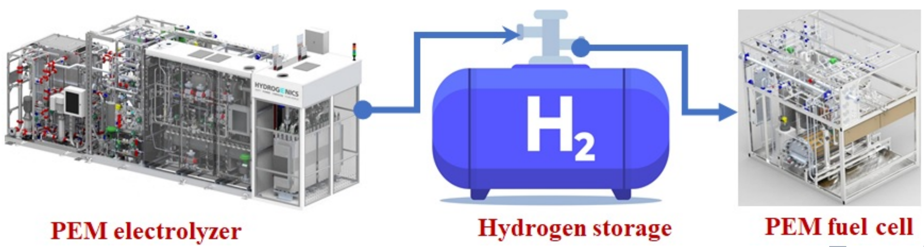
System state space model

$$H(t+1) = H(t) + \eta_e(t) P_e(t) \delta_e^{\mathrm{ON}}(t) T_s - \frac{P_f(t)\delta_f^{\mathrm{ON}}(t) T_s}{\eta_f(t)}$$

The hydrogen level dynamics are defined as a function of

- logic variables;
- the hydrogen production efficiency $\eta_e(t)$;
- the hydrogen consumption efficiency $\eta_f(t)$;

and subject to $\delta_e^{\mathrm{ON}}(t) + \delta_f^{\mathrm{ON}}(t) \leq 1$.



Hydrogen tank dynamics II

MATLAB code of implied operator model

```
LOH   = sdpvar(1,T+1,'full'); % Hydrogen level
z_e   = sdpvar(1,T,'full');   % Auxiliary var
for k= 1:N_T
     de =  binvar(2,1); df =  binvar(2,1);
     F1 = [sum(de)==1,implies(de(1),[z_e(:,k) == p_e(:,
    k),d_ONN_e(:,k) == 1]);
     implies(de(2), [z_e(:,k) ==0, d_ONN_e(:,k) == 0])
    ];
     F2 = [sum(df)==1,implies(df(1),[z_f(:,k)== p_f(:,k
    ), d_ONN_f(:,k) == 1]);
     implies(df(2),[z_f(:,k)==0, d_ONN_f(:,k) == 0])];
     Model = [LOH(:,k+1) == LOH(:,k)+(eta_e.*z_e(:,k)*
    Ts)-(eta_df.*z_f(:,k)*Ts)];
     Con= [Con, Model,LOH(1) == iniLOH, F1, F2];
     Con = [Con, minLOH<=LOH(:,k)<= maxLOH];
end
```

Hydrogen tank dynamics III

MATLAB code of Big-M model

```
LOH  = sdpvar(1,T+1,'full'); % Hydrogen level
z_e  = sdpvar(1,T,'full');   % Auxiliary var
z_f  = sdpvar(1,T,'full');   % Auxiliary var
for k= 1:N_T
    z_i   = [z_e(:,k);z_f(:,k)];
    %Big-M formulated Model
    Const = [Const,-M*d_ONN_i<= z_i<= M*d_ONN_i];
    Const = [Const,-M*(1-d_ONN_i)<= z_i-p_i<= M*(1-
   d_ONN_i)];
    Model = [LOH(:,k+1) == LOH(:,k)+(eta_e.*z_e(:,k)*
   Ts)-(eta_df.*z_f(:,k)*Ts)];
    Const = [Const, Model,LOH_HL(1) == iniLOH_HL];
    Const = [Const,minLOH<=LOH_HL(:,k)<= maxLOH];
end
```

What are the constraints?

- MLD constraints of the devices states;
- MLD constraints of the devices state transitions;
- power balancing equation;
- feasibility and operating constraints:
    - lower and upper power bound on the devices;
    - lower and upper hydrogen storage level;
    - dump load constraint;
- warm and the cold start times of the devices;
- power ramp up/down bounds.

MPC I

$$\boldsymbol{u}_t^{N-1*} = \arg\min_{\boldsymbol{u}_t^{N-1}} \sum_{j=0}^{N-1} \textbf{Costs}(u_{t+j})$$

s.t.

$$u_{t+j} \in \mathcal{U}_{t+j}.$$

- t is the actual time;
- N is the prediction horizon;
- $\boldsymbol{u}_t^{N-1} = [u_t, \ldots, u_{t+N-1}]'$.

At each time step t:

- take measurements from the plant;
- minimize costs s.t. constraints;
- use u_t^* and discard $u_{t+1}^*, \ldots, u_{t+N-1}^*$;
- increment t.

MPC II

$$\boldsymbol{u}_t^{N-1\,*} = \arg\min_{\boldsymbol{u}_t^{N-1}} \sum_{j=0}^{N-1} \textbf{Costs}(u_{t+j})$$

s.t.

$$u_{t+j} \in \mathcal{U}_{t+j}.$$

- t is the actual time;
- N is the prediction horizon;
- $\boldsymbol{u}_t^{N-1} = [u_t, \ldots, u_{t+N-1}]'$.

Costs given by:

- tracking errors;
- output power smoothing;
- euros payed/earned for buying/selling energy;
- wearing-out due to modes switching;
- ...

MPC III

$$\boldsymbol{u}_t^{N-1\,*} = \arg\min_{\boldsymbol{u}_t^{N-1}} \sum_{j=0}^{N-1} \mathbf{Costs}(u_{t+j})$$

s.t.

$$u_{t+j} \in \mathcal{U}_{t+j}.$$

- t is the actual time;
- N is the prediction horizon;
- $\boldsymbol{u}_t^{N-1} = [u_t, \dots, u_{t+N-1}]'$.

$\mathcal{U}$ given by:

- $P_{avl}, P_e, P_f, \delta^{ON}, \delta^{OFF}, \dots$

Cost functions

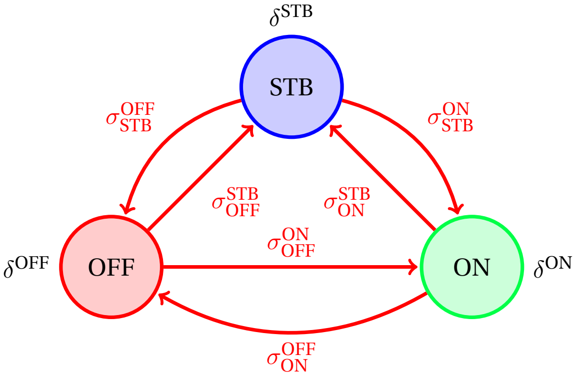
Devices cost functions

$$
\begin{aligned}
J(t) = &\left(\frac{S_{\mathrm{rep}}}{\mathrm{NH}} + C^{\mathrm{OM}}\right)\delta^{\mathrm{ON}}(t) \\
&+ C^{\mathrm{ON}}_{\mathrm{OFF}}\sigma^{\mathrm{ON}}_{\mathrm{OFF}}(t) \\
&+ \mathrm{Cos}^{\mathrm{OFF}}_{\mathrm{ON}}\sigma^{\mathrm{OFF}}_{\mathrm{ON}}(t) \\
&+ C^{\mathrm{STB}}_{\mathrm{ON}}\sigma^{\mathrm{STB}}_{\mathrm{ON}}(t) \\
&+ C^{\mathrm{ON}}_{\mathrm{STB}}\sigma^{\mathrm{ON}}_{\mathrm{STB}}(t) \\
&+ C^{\mathrm{OFF}}_{\mathrm{STB}}\sigma^{\mathrm{OFF}}_{\mathrm{STB}}(t) \\
&+ C^{\mathrm{STB}}_{\mathrm{OFF}}\sigma^{\mathrm{STB}}_{\mathrm{OFF}}(t) \\
&+ c(t)P^{\mathrm{STB}}\delta^{\mathrm{STB}}(t)
\end{aligned}
$$

Fuel cell elect.

J_f

J_e



where

- S_{rep}: the *i*-device stack replacement cost;
- C^{OM}: the *i*-device operation and maintenance cost;
- $c(t)$: the power spot price.

Output power smoothing (OPS) cost function I

The OPS is achieved by accounting for previously available power values and adequately evaluating the scheduling of future power ahead of time. The OPS cost function is minimized through the linear weight cost term

OPS cost function

$$J_s(t) = \sum_{j=0}^{N-1} \sum_{\tau=1}^{\tau_B} \omega^{t+j,\tau} y^{t+j,\tau}$$

where

- $y^{t+j,\tau}$ is the bound on the difference of past and future available power value;
- $\omega^{t+j,\tau}$ is a weighting factor with τ runs within the set $\{1, \ldots, \tau_B\}$.

$y_t^{t+j,\tau}$ is subject to the following constraint

$$y^{t+j,\tau} \geq 0$$

$$y^{t+j,\tau} \geq |P_{\text{avl}}(t+j) - P_{\text{avl}}(t+j-\tau)| - \bar{y}^{\tau}$$

~ derivative

- $\bar{y}^{\tau}$ is a threshold of the grid operator.

Cost functions

One goal of the system is to track the load demand with the available power. The load tracking (LT) cost function is given by the mismatch between P_{ref} and P_{avl} as

LT cost function

$$J_l(t) = \sum_{j=0}^{N-1} \Big(P_{\mathrm{avl}}(t+j) - P_{\mathrm{ref}}(t+j) \Big)^2.$$

where

- P_{avl} is the smoothed available power;
- P_{ref} is the local load demand.

The total cost function at time t is given by

Total cost function

$$J(t) = \sum_{j=0}^{N-1} \rho_l J_l(t+j) + \rho_e J_e(t+j) + \rho_f J_f(t+j).$$

where

- ρ_l, ρ_e and ρ_f are positive weighting scalars.

Multi-objective optimization I

$$\min \quad \left\{ J_s(t),\ J(t) \right\}$$

s.t.

System constraints,

Power smoothing constraints.

The problem is recast sequentially (two stage sequential optimization)

1. give unconditional priority to the OPS problem;
2. pass the optimal value J_s^* of OPS as a further constraint in the 2^{nd} problem.

1^{st} problem

$$\min \ J_s(t)$$

s.t.

System constraints.

2^{nd} problem

$$\min J(t)$$

s.t.

System constraints,

$J_s(t) \leq J_s^*$.

Test runs (Case I) I

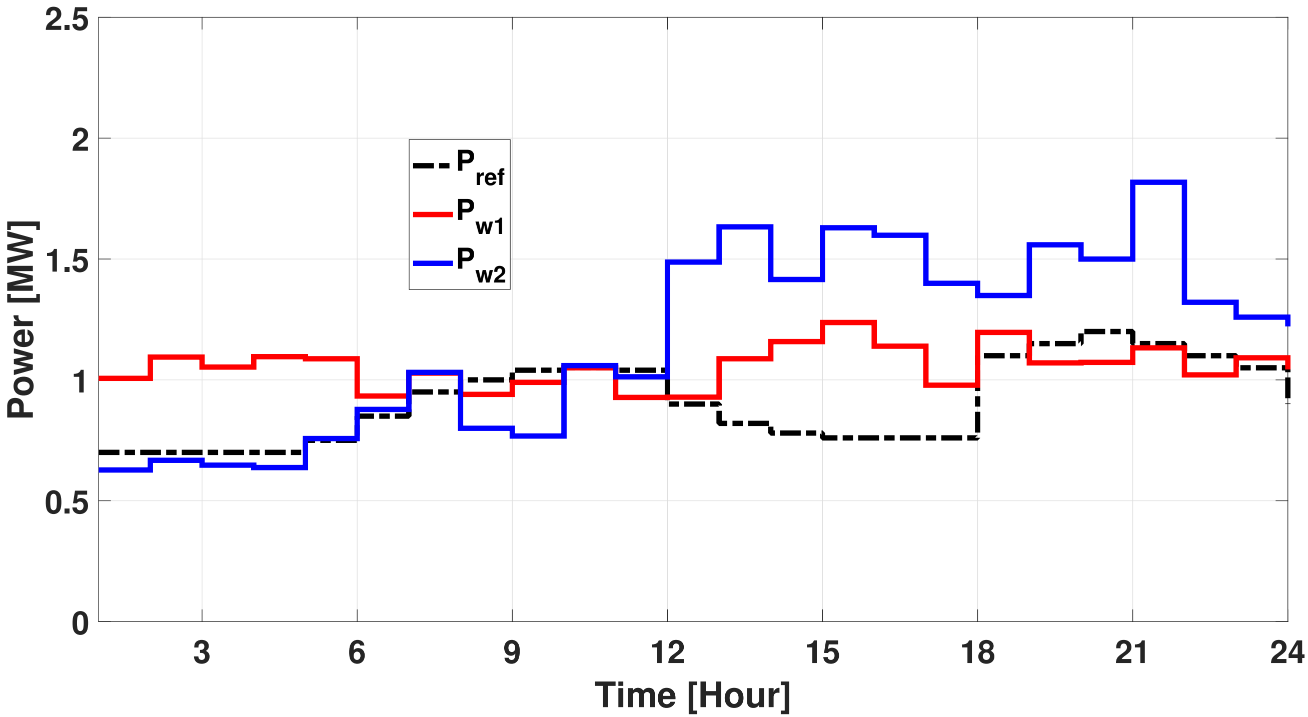


Figure: Wind and operator power profiles.

Test runs (Case I) II

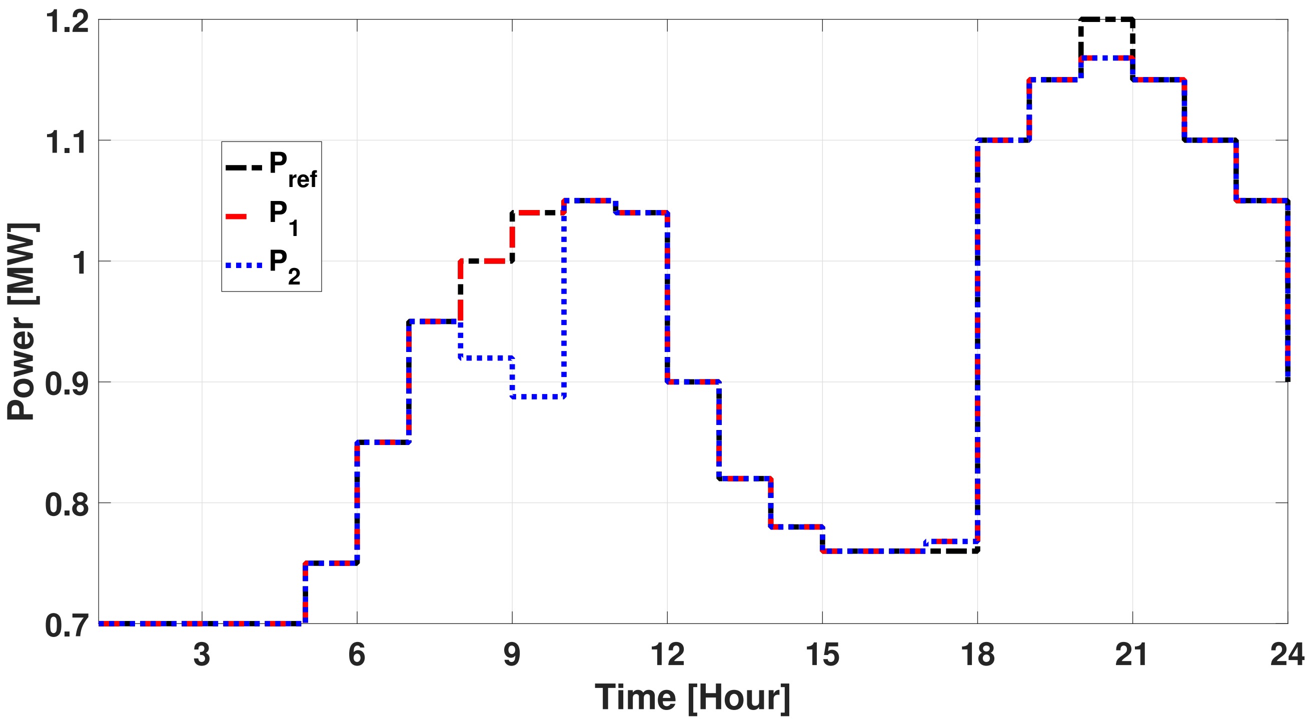


Figure: Smoothed available power profiles.

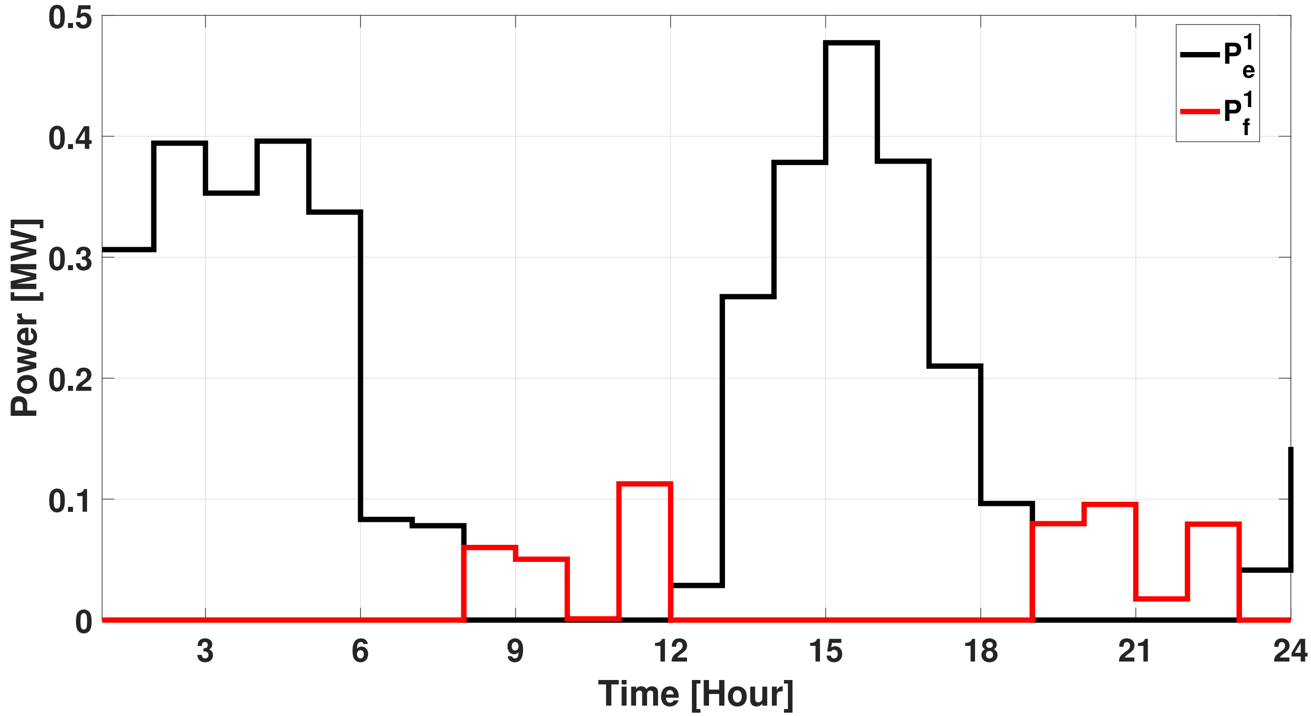


Figure: Control response of the devices.

Test runs (Case I) IV

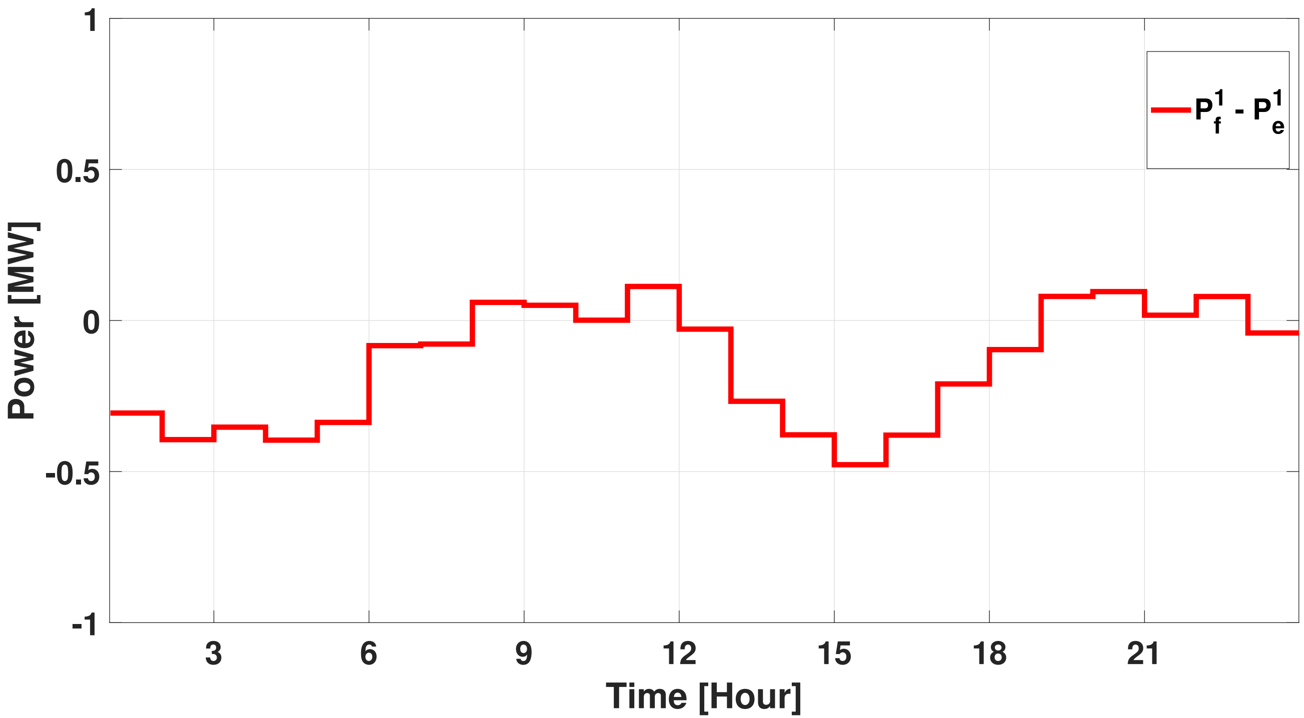


Figure: Net hydrogen storage power $P_{H_2} = P_f - P_e$.

Test runs (Case I) V

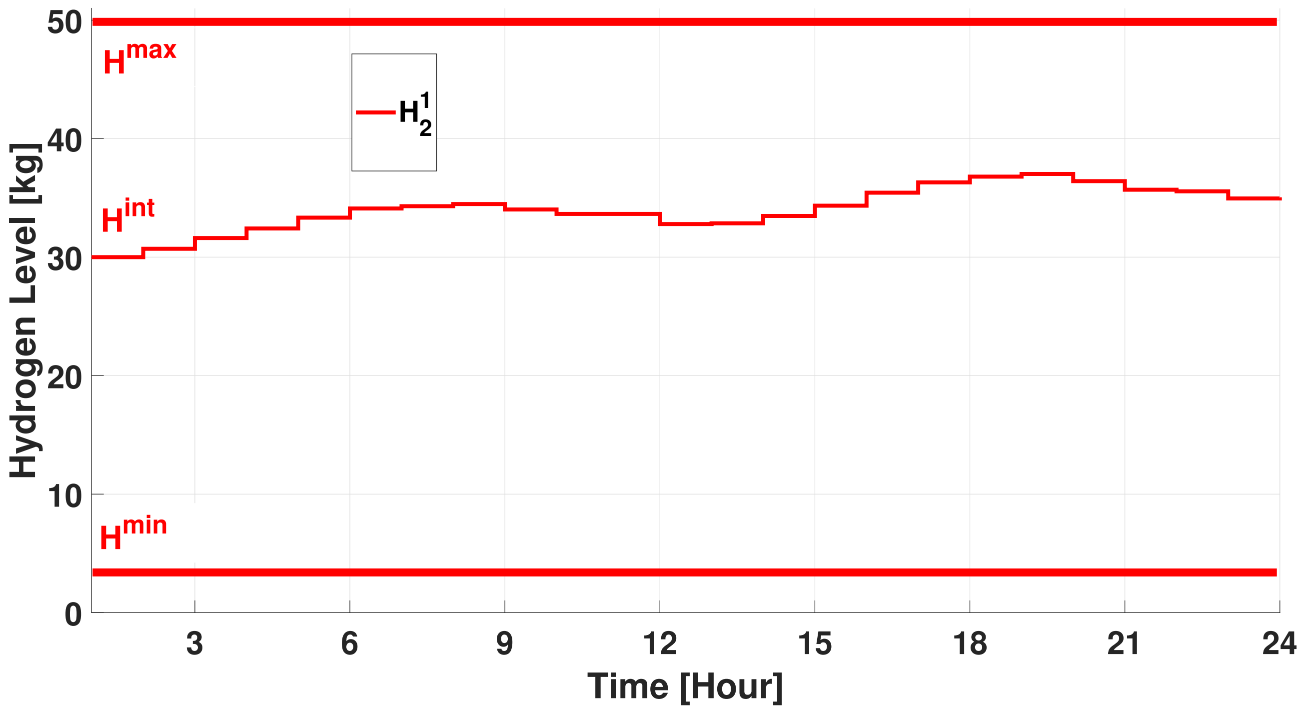


Figure: Control response of hydrogen storage H.

Test runs (Case II) I

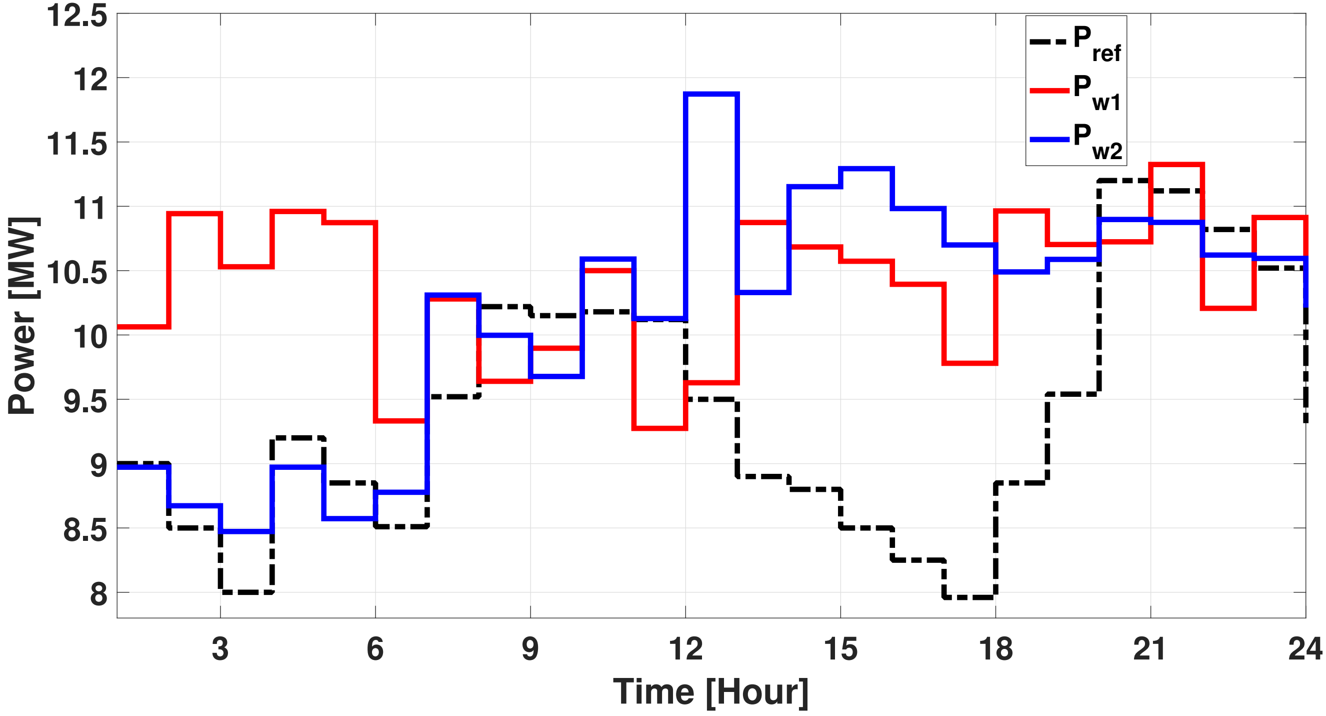


Figure: Wind and operator power profiles.

Test runs (Case II) II

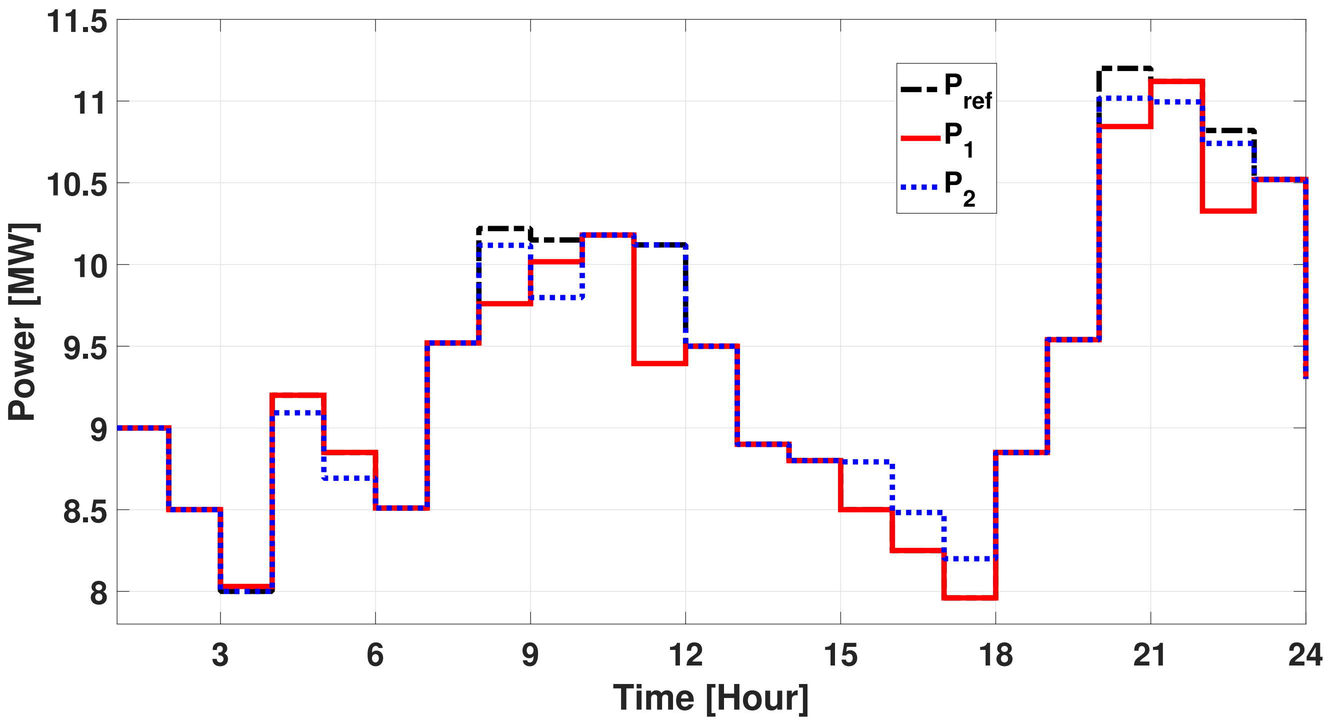


Figure: Smoothed available power profiles.

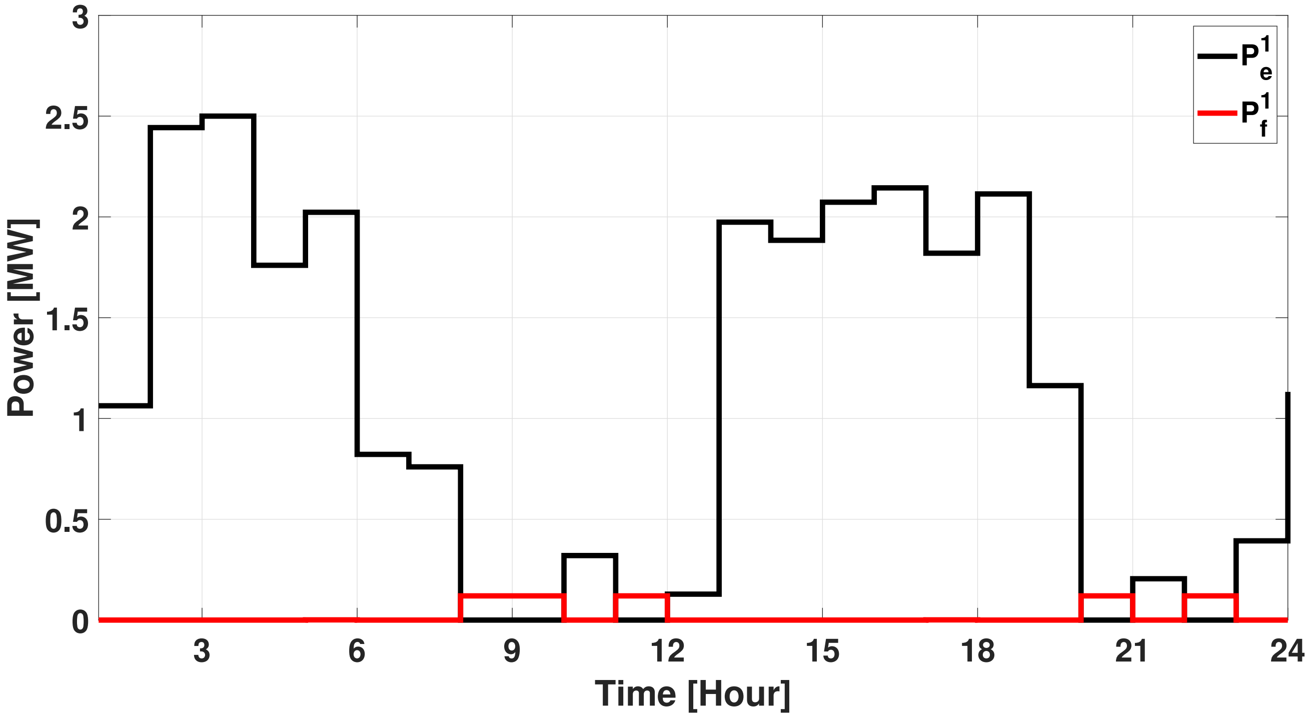


Figure: Control response of the devices.

Test runs (Case II) IV

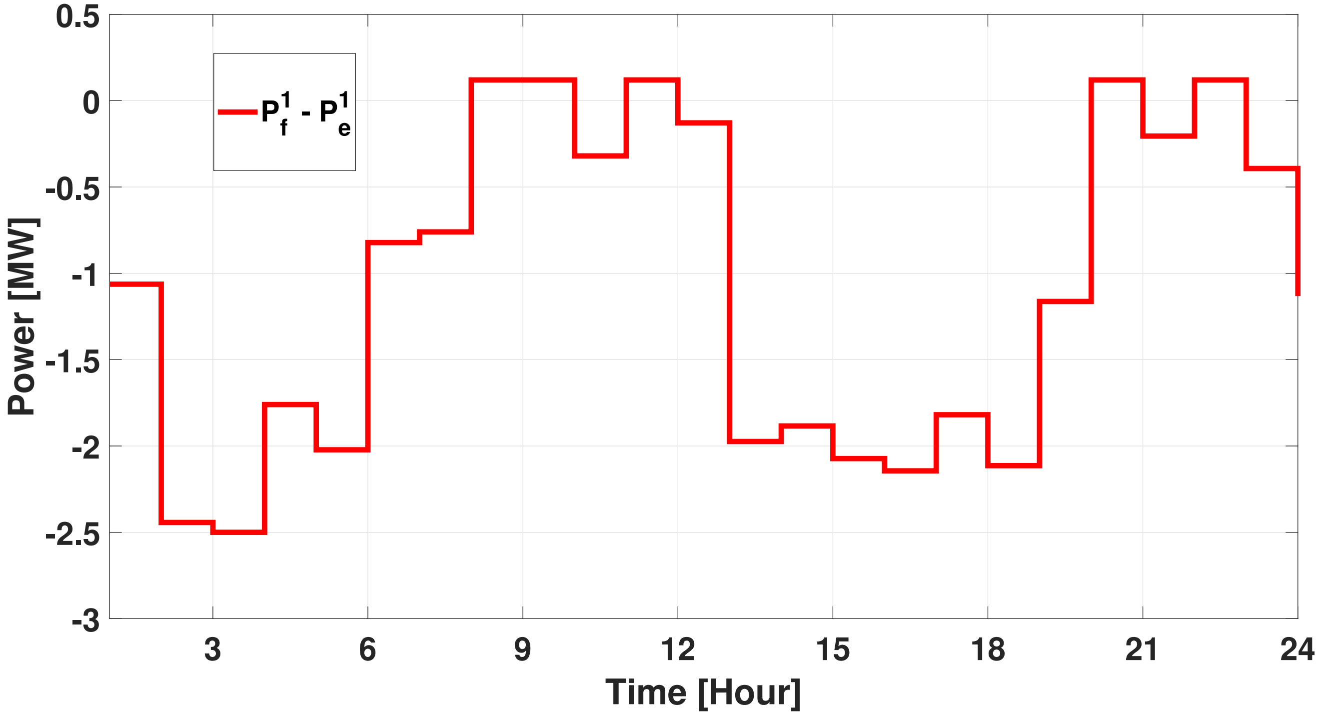


Figure: Net hydrogen storage power $P_{H_2} = P_f - P_e$.

Test runs (Case II) V

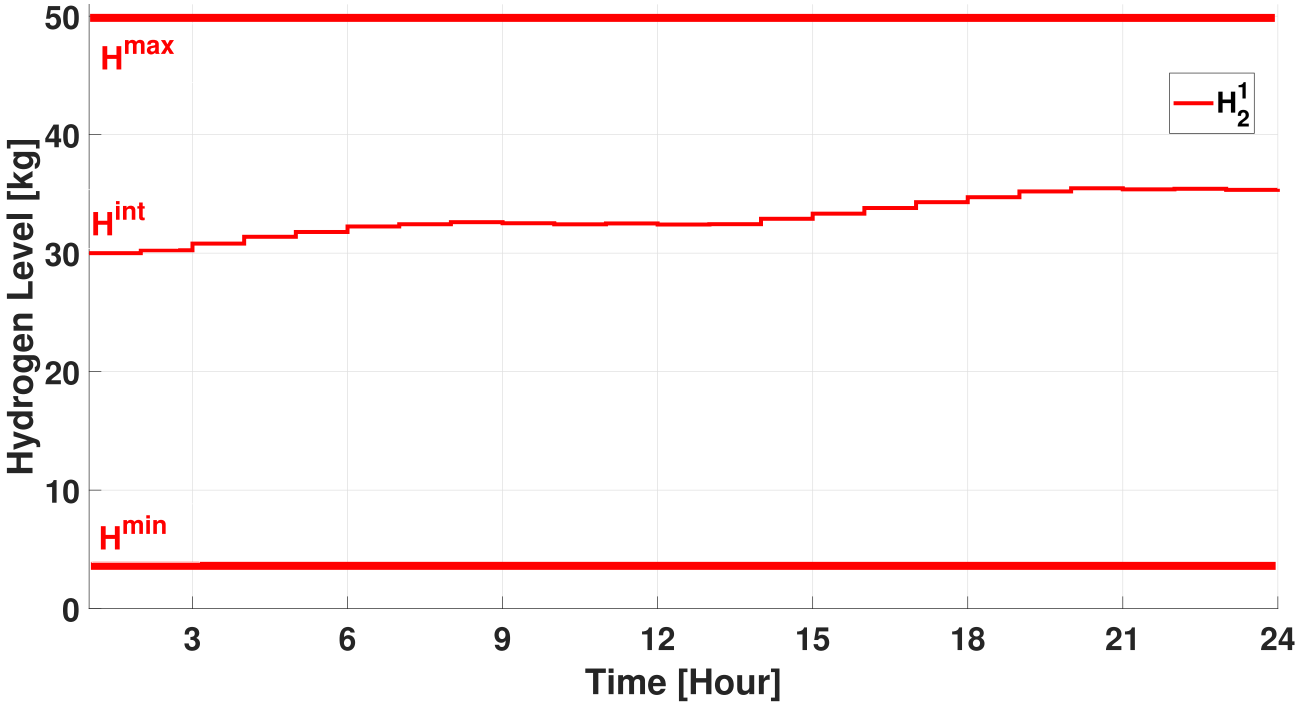


Figure: Control response of hydrogen storage H.

Thanks

- to **Muhammad Bakr Abdelghany**, **Muhammad Faisal**, **Davide Liuzza**, and **Valerio Mariani**
- to all partners
  - **UBFC**
  - **KES**
  - **SINTEF**
  - **Tecnalia**
  - **Hydrogenics**
  - **Varanger Kraft**

H2AEOLUS

Hydrogen-**A**eolic **E**nergy with **O**ptimised e**L**ectrolysers
Upstream of **S**ubstation

This project has received funding from the Fuel Cells and Hydrogen 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement № 779469.

FCH

FUEL CELLS AND HYDROGEN JOINT UNDERTAKING