



UNIONE EUROPEA  
Fondo Sociale Europeo  
Fondo Europeo di Sviluppo Regionale



## **Avviso 1735 del 13.07.2017 MIUR**

Progetti di Ricerca Industriale e Sviluppo Sperimentale nelle 12 Aree di Specializzazione individuate dal PNR 2015-2020

---

# **Sviluppo di una strategia di gestione ottimale del consumo aggregato di colonnine di ricarica al fine di limitare i picchi di domanda giornalieri e minimizzare i costi di gestione**

---

***Rapporto Tecnico di Ricerca Industriale D3.6c***



<b>Avviso</b>	Avviso 1735 del 13.07.2017 MIUR
<b>Codice progetto</b>	ARS01_01259
<b>Nome del progetto</b>	Community Energy Storage Gestione Aggregata di Sistemi di Accumulo dell'Energia in Power Cloud
<b>Acronimo</b>	ComESTo
<b>Documento</b>	D3.6c
<b>Tipologia</b>	Rapporto Tecnico
<b>Data di Rilascio</b>	15/09/2021
<b>Obiettivo Realizzativo</b>	OR3
<b>Attività Realizzativa</b>	A3.6
<b>Soggetti Beneficiari Proponenti</b>	UNIVPM, ENEA
<b>Elaborato (Nome, Cognome – Soggetto Beneficiario)</b>	Gabriele Comodi – UNIVPM Lucio Ciabattoni – UNIVPM Stefano Cardarelli – UNIVPM Viviana Cigolotti – ENEA Marialaura Di Somma – ENEA Giorgio Graditi – ENEA
<b>Verificato (Nome, Cognome – Soggetto Beneficiario)</b>	Viviana Cigolotti – ENEA
<b>Approvato (Nome, Cognome – Soggetto Beneficiario)</b>	Membri del PEB

## INDICE

Indice delle figure .....	4
Executive Summary.....	5
1 INTRODUZIONE .....	6
2 Aggregator online Assistant - blocco Simulink.....	7
2.1 Configurazione.....	9
2.2 Funzionamento dell'Aggregator Online Assistant .....	10
2.2.1 AOA Inputs.....	11
2.2.2 AOA Outputs.....	11
3 esempio di strategia di gestione ottimale di un aggregato di colonnine .....	12
3.1 Sistema di Inferenza Fuzzy.....	13
4 Risultati .....	19
5 CONCLUSIONI.....	23
6 BIBLIOGRAFIA.....	24

## INDICE DELLE FIGURE

<i>Figura 1. Rappresentazione grafica dell'architettura attualmente implementata del simulatore. ....</i>	<i>9</i>
<i>Figura 2. Schema concettuale dell'architettura Matlab/Simulink.....</i>	<i>10</i>
<i>Figura 3 - Insiemi fuzzy trapezoidali e triangolari della variabile ChargeRequisite.....</i>	<i>15</i>
<i>Figura 4 - Insiemi fuzzy trapezoidali della variabile RechAvailability. ....</i>	<i>16</i>
<i>Figura 5 - Insiemi fuzzy trapezoidali della variabile SOC. ....</i>	<i>16</i>
<i>Figura 6. Una settimana di profili di ricarica globali. ....</i>	<i>19</i>
<i>Figura 7. Estratto di una simulazione che illustra il comportamento di EV1, EV2 ed EV3 quando vengono fermati fino a quando i requisiti minimi più urgenti non vengono soddisfatti da altri veicoli, come EV7. ....</i>	<i>22</i>

## EXECUTIVE SUMMARY

All'interno del progetto ComESTo, l'attività A3.6 dal titolo "Analisi e definizione di soluzioni di accumulo distribuito come soluzione di distributore ibrido per la ricarica di veicoli elettrici di tipo V2G" ha lo scopo di studiare l'utilizzo delle batterie dei veicoli elettrici come storage distribuiti a servizio degli utenti finali e/o della rete elettrica, sia in forma singola che in forma aggregata. L'attività 3.6 è stata suddivisa in tre sotto-attività corrispondenti ad altrettanti deliverable. Il presente documento è il terzo deliverable dell'attività 3.6.

Nel primo deliverable, dal titolo "Sviluppo e validazione di modelli data-driven in grado di simulare la domanda di energia di colonnine di ricarica per auto elettriche appartenenti a diverse classi di utenti finali", è stato descritto in dettaglio il simulatore di profili di ricarica di singole colonnine di auto elettriche, pubbliche o private (ePopSimulator).

Nel secondo deliverable, dal titolo "Sviluppo di una metodologia per poter modellare un aggregato di colonnine di ricarica per auto elettriche come un sistema di storage equivalente", è stata descritta l'attività svolta per estendere le funzionalità di una singola colonnina di ricarica ad un aggregato di colonnine.

Questo deliverable descrive possibili strategie per la gestione ottimale di un aggregato di colonnine. Il risultato principale di questo deliverable è l'"Aggregator online Assitant" (AOA), un blocco Simulink che permette di importare i risultati dei profili di colonnine aggregati sviluppati dal software presentato nei deliverable A3.6.a (ePopSimulator, singola colonnina) e A3.6b (ePopSimulator, colonnine aggregate) e di poterli gestire/interfacciare con algoritmi di ottimizzazione esterni. Infatti, l'AOA permette di inviare le informazioni (stato di carica, potenza impegnata, durata prevista di ricarica, potenza colonnina...) di tutte le colonnine simulate al potenziale Aggregatore; queste informazioni potranno essere elaborate da un qualsivoglia algoritmo di ottimizzazione che implementi una o più funzioni obiettivo; i risultati dell'ottimizzazione, in termini di scheduling di carica e scarica, verranno di nuovo inviati all'AOA che provvederà a ridefinire il nuovo profilo ottimizzato e a ricominciare il loop.

Grazie all'AOA, sarà quindi possibile utilizzare il software di simulazione ePopSimulator per testare, in assenza di dati reali, una molteplicità di strategie di gestione corrispondenti a diverse tipologie di algoritmi di ottimizzazione e funzioni obiettivo sviluppati da soggetti terzi (soggetti Aggregatori, DSO, Centri di Ricerca, Università) per contesti nazionali o internazionali.

Partendo da dati simulati dal software (ePopSimulator), in questo deliverable vengono presentate possibili strategie per la gestione del processo di carica e scarica da parte di un Aggregatore di colonnine di ricarica. È importante sottolineare che lo scopo del deliverable non è tanto quello di sviluppare/proporre una strategia ottimale in assoluto (infatti gli obiettivi possono variare a seconda dell'utilizzatore o del contesto), quanto piuttosto quella di gestire le priorità di attivazione che il soggetto Aggregatore dovrà assegnare ai veicoli elettrici collegati alle colonnine in modo da ottenere determinati obiettivi quali, ad esempio, il peak shaving e/o la minimizzazione dei costi, garantendo la qualità del servizio all'utente finale.

## 1 INTRODUZIONE

All'interno del progetto ComESTo, l'attività A3.6 dal titolo "Analisi e definizione di soluzioni di accumulo distribuito come soluzione di distributore ibrido per la ricarica di veicoli elettrici di tipo V2G" ha lo scopo di studiare l'utilizzo delle batterie dei veicoli elettrici come storage distribuiti a servizio degli utenti finali e/o della rete elettrica, sia in forma singola che in forma aggregata. L'attività 3.6 è stata suddivisa in tre sotto-attività corrispondenti ad altrettanti deliverable.

In questo terzo deliverable, dal titolo "Sviluppo di una strategia per poter gestire in modo ottimale il consumo aggregato di un gruppo di colonnine di ricarica elettrica con l'obiettivo di limitare i picchi di domanda giornalieri e minimizzare i costi di gestione", viene descritta l'attività che ha portato allo sviluppo della funzione "Aggregator online Assistant" (AOA), una funzione Simulink che permette di importare i risultati dei profili di colonnine aggregati sviluppati dal software (ePopSimulator) presentato nei deliverable A3.6a e A3.6b e di poterli gestire/interfaciare con algoritmi di ottimizzazione sviluppati da soggetti terzi (Aggregatori, Centri di Ricerca, Università) che così potranno essere testati su una platea di dati simulati, in assenza di dati reali.

Lo sviluppo dell'AOA è il risultato principale di questo deliverable. Infatti, le strategie di gestione di un parco elettrico sono molteplici e possono variare a seconda del gestore o del contesto di mercato dell'energia elettrica. Grazie all'AOA sarà possibile interfacciare il software di simulazione sviluppato nell'Attività A3.6 con algoritmi di ottimizzazione e poter sviluppare, anche in assenza di dati reali, strategie di demand side management/flexibility implementabili da parte di un qualsivoglia "gestore".

In particolare, in questo deliverable la metodologia proposta per identificare le priorità di attivazione/gestione da assegnare ai veicoli elettrici collegati alle varie colonnine di ricarica verrà applicata all'obiettivo di ottenere un abbassamento della potenza massima assorbita dalla rete elettrica (peak shaving) cercando di minimizzare i costi di gestione (di ricarica).

Il documento è strutturato come segue: dopo l'introduzione, il capitolo 2 presenta il dettaglio dell'"Aggregator Online Assistant"; il capitolo 3 riporta un esempio di strategia di gestione ottimale di un aggregato di colonnine con sistema di Inferenza Fuzzy; il capitolo 4 riporta i risultati della strategia ottimale e le priorità di gestione dei veicoli elettrici; infine, il capitolo 5 riporta le conclusioni dell'attività svolta.

## 2 AGGREGATOR ONLINE ASSISTANT - BLOCCO SIMULINK

Nei deliverables A3.6a e A3.6b sono stati introdotti, rispettivamente, il simulatore (ePopSimulator) nella versione per una singola stazione di ricarica per veicoli elettrici e la versione per poter simulare un aggregato di stazioni di ricarica di EV considerando le abitudini di diversi gruppi di utenti.

Come descritto nei precedenti deliverables, il core di tali simulatori è la classe consumer che permette di modellare il comportamento del proprietario dell'EV in termini di abitudini di ricarica, percorrenza tipica, tipo di veicolo.

L'output del simulatore online, accessibile dal link <http://193.206.121.28:32000/> (link provvisorio) e descritto nel dettaglio in precedenza, è composto sia da elementi grafici, sia dalla generazione (con possibilità di download) di files in formato .json e .csv.

Proprio a partire da questi files viene generato quello che è l'output del presente deliverable, ovvero l'“Aggregator online Assistant” (AOA), un simulatore in ambiente Matlab/Simulink dell'aggregato di colonnine, interfacciabile con algoritmi di controllo, ottimizzatori, aggregatori, supervisor esterni e basato sulla popolazione di veicoli e abitudini generati dal tool online.

In particolare, come mostrato in Figura 1, la generazione del simulatore Matlab/Simulink avviene a partire dal setup di una simulazione nel tool online. Una volta terminata la simulazione e generati i files di output, ci si concentra su quelli in formato .json, nella fattispecie i files chiamati “RechStationsEvents\_X.json”, che simulano i dati ottenibili mediante la comunicazione con la stazione di ricarica e con il consumer, e contengono tutte le informazioni del veicolo elettrico X nell'istante in cui si connette alla colonnina di ricarica. Tali files hanno il seguente formato:

```
"event_1": {"uuid": "1635394326f446ba82e7daf4c9862d3d", "incremental_id": 1, "time_resolution_min": 30, "simulation_days": 7, "min_rech_en": 17.86, "min_rech_hours": 2.06, "delta_en": 0, "hours_surplus": 10.94, "energy_surplus": 49.21, "plug_in_time": "19:00", "plug_out_time": "08:00", "plugin_time_index": 37, "plugout_time_index": 63, "SOC_perc": 31.72, "SOC_en": 8.57, "rech_power": 4.5, "minsoc_perc": 66.13, "minsoc_en": 17.86, "where": "home_ev", "battery_cap": 27.0, "future_consumption_perc": 59.69, "future_consumption_en": 16.08}
```

Dove:

“uuid” è un identificativo univoco dell'evento di connessione del particolare veicolo alla colonnina;

“incremental\_id” è l'identificativo del veicolo (nella simulazione di un aggregato di veicoli, questo parametro è la chiave per identificare il singolo veicolo) che servirà per tenere traccia del veicolo X-esimo;

"time\_resolution\_min" e "simulation\_days" sono i parametri scelti per la simulazione nel pannello di configurazione iniziale (e saranno quindi fissi per tutti gli eventi nel file .json) e serviranno a settare i parametri globali della simulazione in ambiente Matlab/Simulink;

"min\_rech\_en" rappresenta la minima quantità di energia che il veicolo dovrà avere a fine ricarica;

"min\_rech\_hours" il numero minimo di ore di ricarica alla massima potenza erogabile dalla colonnina che serviranno a raggiungere il valore di "min\_rech\_en". Tale valore non viene utilizzato all'interno del simulatore Matlab/Simulink;

"delta\_en" rappresenta il delta di energia disponibile in ottica V2G all'istante di plug in del veicolo indicato con l'"incremental\_id" precedente. Tale valore non viene utilizzato all'interno del simulatore Matlab/Simulink;

"hours\_surplus" e "energy\_surplus" rappresentano le ore e l'energia in eccesso che potrebbe essere caricata oltre le "min\_rech\_hours". Tali valori non vengono utilizzati all'interno del simulatore Matlab/Simulink;

"plug\_in\_time" e "plug\_out\_time" sono gli orari di connessione serale e disconnessione mattutina dalla stazione di ricarica in questione. Tali valori non vengono utilizzati all'interno del simulatore Matlab/Simulink;

"plugin\_time\_index" e "plugout\_time\_index" rappresentano gli index temporali dall'inizio della simulazione in cui avvengono gli eventi di plug in e plug out (corrispondenti ai "plug\_in\_time" e "plug\_out\_time" precedentemente introdotti). Tali valori serviranno alla modellazione del singolo veicolo X-esimo all'interno dell'ambiente Matlab/Simulink;

"SOC\_perc" e "SOC\_en" rappresentano la SOC in termini di percentuale ed energia nel momento in cui il veicolo si connette alla stazione di ricarica. Tali valori serviranno alla modellazione del singolo veicolo X-esimo all'interno dell'ambiente Matlab/Simulink;

"rech\_power" rappresenta il valore di potenza massimo erogabile dalla stazione di ricarica (ed è un valore fisso in tutti gli eventi per la singola stazione di ricarica) e servirà per caratterizzare la stazione di ricarica che serve il veicolo X-esimo, in particolare sarà il vincolo massimo di potenza erogabile istantaneamente che un algoritmo esterno dovrà considerare;

"minsoc\_perc" e "minsoc\_en" rappresentano la SOC minima (in termini percentuali e di energia) che il veicolo elettrico dovrà avere al momento del plug out dalla stazione di ricarica per permettere al consumer di percorrere il chilometraggio atteso. Tali valori serviranno alla modellazione del singolo veicolo X-esimo all'interno dell'ambiente Matlab/Simulink, in particolare, saranno dei vincoli sempre disponibili per un algoritmo esterno che determinano il valore minimo di carica che il veicolo dovrà avere al termine della carica;

"where" rappresenta la tipologia di stazione di ricarica a cui si è connesso il veicolo identificato da "incremental\_id" e può assumere i valori "home" e "public";

"battery\_cap" rappresenta il valore in kWh di capacità massima della batteria (ed è un valore costante per il singolo veicolo) e servirà per caratterizzare il veicolo X-esimo;

"future\_consumption\_perc" e "future\_consumption\_en" rappresentano la percentuale di SOC e i kWh necessari alla batteria dell'EV per effettuare il chilometraggio atteso del consumer. Tali valori non saranno utilizzati per la modellazione del singolo veicolo X-esimo all'interno dell'ambiente Matlab/Simulink, in quanto verranno sostituiti dalla "minsoc\_en".

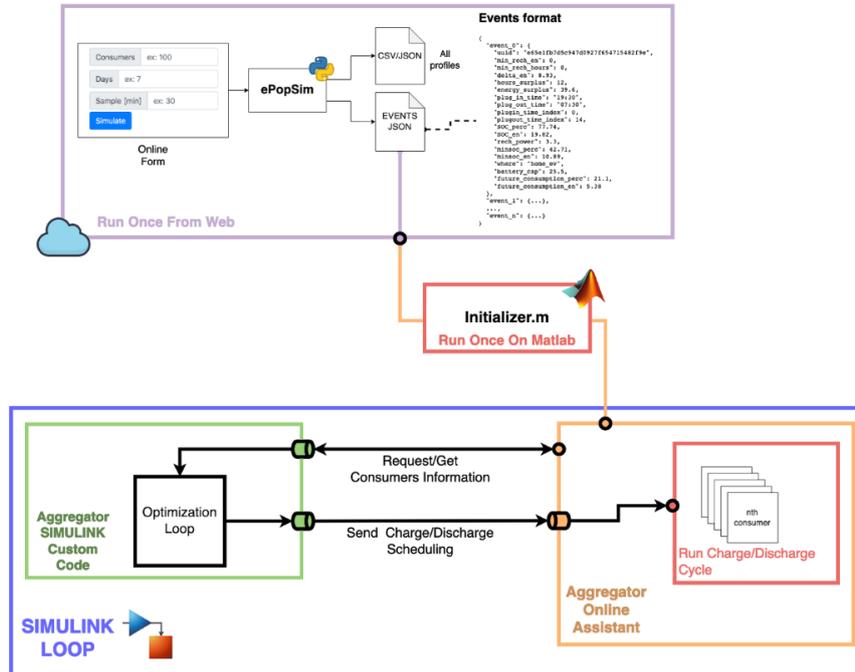


Figura 1. Rappresentazione grafica dell'architettura attualmente implementata del simulatore. Il frame superiore rappresenta il software accessibile dal cloud (ePopSimulator) per eseguire una simulazione senza aggregatore. I requisiti e le abitudini dei clienti simulati vengono quindi scaricati e inviati al blocco simulink offline tramite lo script Initializer.m. Il frame nella parte inferiore rappresenta il loop Simulink. La casella arancione, o Aggregator Online Assistant (AOA), spiegata in dettaglio nel seguito, è il simulatore online di EV e si occupa di applicare le eventuali regole di ricarica imposte dal box in verde (che può rappresentare, ad esempio, un ottimizzatore o un aggregatore con logica personalizzata).

Questa sezione si concentra nella descrizione della configurazione del blocco Simulink, dei suoi ingressi, uscite e del modo con cui interagire. Tenendo conto dello schema di Figura 1, di seguito verranno fornite la descrizione del file Initializer.m e dell'Aggregator Online Assistant.

## 2.1 Configurazione

Il blocco Initializer.m copre un ruolo "ponte" per gestire la dipendenza del blocco Simulink da “ePop Simulator” basato su cloud descritto nelle sezioni precedenti.

In particolare, una volta scaricato l'intero .zip contenente i files .json dal tool di simulazione online, il file initializer.m avrà a disposizione tutte le informazioni necessarie da tutti i diversi veicoli elettrici per la creazione del modello, come dettagliato nella precedente sezione.

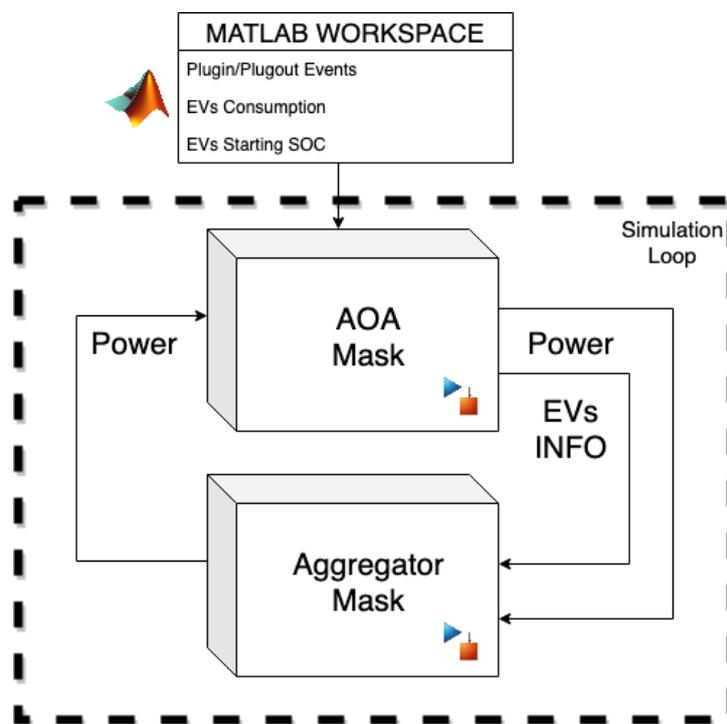
L'inizializzazione dello strumento Simulink richiede quindi i seguenti passaggi:

- 1) Eseguire una simulazione batch sullo strumento basato su cloud con i parametri scelti, il tempo di campionamento, il numero di veicoli e il numero di giorni e scaricare i file .json risultanti.
- 2) Eseguire Initializer.m su MATLAB per popolare lo spazio di lavoro con le dipendenze e le variabili Simulink.

## 2.2 Funzionamento dell'Aggregator Online Assistant

Una volta lanciato il file initializer.m e popolato il workspace del Matlab, sarà possibile lanciare il file EV\_model\_smlk.slx in cui sarà presente il blocco di simulazione relativo a tutta la popolazione di veicoli elettrici precedentemente creata. Tale blocco prende il nome di "Aggregator Online Assistant".

L'Aggregator Online Assistant (AOA) ha un input e due output (di cui uno composto da una serie di informazioni). Se l'ingresso è scollegato, può funzionare come un blocco autonomo e la sua logica di base è la stessa di quella del simulatore basato su cloud (ovvero ogni veicolo, quando collegato, caricherà la batteria con la massima potenza erogabile dalla colonnina fino a quando la SOC raggiunge il 100% o fino al momento del plugout). Le informazioni sul modello EV, gli istanti plug-in / plugout, il consumo di energia, le distanze effettive commutate vengono conservate dalla simulazione basata su cloud, mentre il setpoint di potenza per ciascuna fase di ricarica EV può essere impostato esternamente. Ciò implica che un blocco esterno (es. Un aggregatore delle stazioni di ricarica, un gestore di rete, un ottimizzatore, uno scheduler ecc.) possa simulare e controllare in tempo reale la potenza di carica e scarica per ogni EV monitorandone allo stesso tempo lo stato. Gli input e outputs del blocco AOA sono mostrati in *Figura 2* e dettagliati nel seguito.



*Figura 2. Schema concettuale dell'architettura Matlab/Simulink. AOA Mask, sulla base dei parametri caricati nel workspace, ha un input e due output e può essere controllato esternamente dal blocco Aggregator Mask all'interno del quale possono essere implementate logiche esterne di controllo e/o ottimizzazione opzionali.*

### 2.2.1 AOA Inputs

**Power:** un array di valori di potenza [kW] utilizzati come setpoint di carica / scarica delle stazioni di ricarica. La lunghezza dell'array è uguale al numero di EV considerati e ogni elemento dell'array farà riferimento a un particolare EV. Valori positivi saranno considerati come un'azione di carica per il dato istante di tempo. Se il valore della potenza supera la capacità della stazione del consumatore dato, viene saturato al valore massimo. I valori negativi vengono trattati come un'azione di scarica per il dato istante di tempo. Gli zeri sono trattati come un comando di "attesa". L'EV dato non si scaricherà o ricaricherà per il dato istante di tempo. Il valore "Nan" viene considerato come un'istruzione di "potenza massima". L'EV dato si ricaricherà alla massima potenza consentita (come fa lo strumento basato su cloud).

### 2.2.2 AOA Outputs

**Power:** Per quanto riguarda l'ingresso, si tratta di un array ordinato contenente la potenza effettiva caricata / scaricata per ogni EV in un dato istante di tempo.

**SOC:** Matrice ordinata di ogni stato di carica effettivo di un EV espresso in valore percentuale rispetto alla Battery Capacity massima.

**Time Until Plugout:** Matrice ordinata in cui ogni valore rappresenta gli istanti di tempo rimanenti fino a quando il dato EV verrà scollegato. Nel caso in cui un EV non sia collegato, il valore corrispondente è zero.

**Minimal SOC:** Matrice ordinata che rappresenta la quantità minima di SOC richiesta da ogni EV in un dato istante di tempo per soddisfare i requisiti dell'utente al momento del plugout (Time Until Plugout). Nel caso in cui l'EV non sia collegato al momento in cui viene letto questo array, il rispettivo valore restituito è zero.

**Battery Capacity:** Matrice ordinata contenente la capacità massima della batteria per ogni EV in termini di kWh. Questo array contiene valori costanti per l'intero tempo di simulazione. Aggiornamenti futuri consentiranno la variabilità di questo parametro per ogni EV al fine di modellare (o prevedere esternamente) lo stato di degrado della batteria.

**Home Recharging Capability:** Matrice ordinata contenente la potenza massima della stazione di ricarica domestica [kW]. Questo array contiene valori costanti per l'intero tempo di simulazione.

### 3 ESEMPIO DI STRATEGIA DI GESTIONE OTTIMALE DI UN AGGREGATO DI COLONNINE

In questa sezione viene presentata un'applicazione di demand side flexibility per quanto riguarda la limitazione dei picchi di potenza giornalieri (peak shaving), espandibile poi ad una gestione dei costi di ricarica aggregati. La tecnica che viene considerata nel seguito è quella di creare una classifica di priorità on-line (per ogni istante di tempo) dei veicoli elettrici in quel momento connessi ad una colonnina elettrica.

Le ipotesi considerate sono le più realistiche e allo stesso tempo “restrittive” per tale applicazione, ovvero:

- non è possibile regolare la potenza di carica del veicolo stesso (ma la colonnina può erogare il massimo di potenza o zero);
- non si dispongono informazioni sul “futuro” (non si conoscono cioè gli orari di arrivo dei veicoli elettrici non connessi in un dato istante di tempo)
- si dispongono informazioni solo per i veicoli attualmente connessi (è nota quindi la SOC del veicolo connesso, una stima dell’istante di plugout e della MINSOC necessaria per percorrere il chilometraggio del successivo viaggio).

A livello di vincoli viene introdotto un cap per la potenza di ricarica globale assorbita per ogni istante dalla rete mentre restano validi quelli relativi alla MINSOC.

Nella situazione in cui si volesse espandere tale use case ad un concetto economico, conoscendo (o stimando) il prezzo dell’energia della giornata, fissando un cap di prezzo (fisso o variabile) orario, si può facilmente risalire al cap di potenza ivi considerato, ricadendo quindi nello stesso scenario.

Tramite ePopSimulator online viene creata una popolazione di 100 veicoli elettrici e tutte le relative variabili vengono generate per il periodo di un mese.

All'interno dell'ambiente Simulink viene quindi realizzato un regolatore di potenza personalizzato tramite un sistema di inferenza fuzzy, di seguito dettagliato, in grado di fornire un grado di priorità di ricarica basato sui seguenti input:

$$\mathbf{ChargeRequisite} = \mathbf{MINSOC} - \mathbf{SOC}$$

Una misura di quanto manca per raggiungere la MINSOC

$$\mathbf{RechAvailability} = \frac{(t_{plugout} \times home_{pw})}{batt_{cap}}$$

Una misura di quanta percentuale di batteria del veicolo posso caricare nel restante tempo disponibile (ottenuta moltiplicando il tempo residuo al plugout per la potenza massima erogabile dalla colonnina, normalizzata per la capacità della batteria del veicolo)

**SOC** Lo stato di carica attuale della batteria

Il grado di priorità viene quindi utilizzato per scegliere i veicoli da caricare in base a un ordine decrescente deterministico fino al raggiungimento del limite di potenza.

Nel caso in cui tra le ipotesi si introducesse la possibilità di regolare la potenza erogata dalla colonnina di ricarica, si può introdurre un algoritmo di ottimizzazione a supporto del ranking di priorità o, più semplicemente, andare a “scalare” la potenza massima della colonnina per il valore di priorità del veicolo.

### 3.1 Sistema di Inferenza Fuzzy

Una FIS Zero Order Takagi Sugeno (TS) è stata progettata per ottenere le priorità di carica dei veicoli a partire dalle variabili di input introdotte in precedenza. Le funzioni di appartenenza (membership functions) considerate per gli input sono funzioni trapezoidali. L'insieme fuzzy trapezoidale è parametrizzato da quattro parametri reali scalari: (a, b, c, d) dove  $a < b < c < d$ . La funzione di appartenenza trapezoidale è descritta matematicamente come segue:

$$\mu_S(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b < x < c \\ \frac{d-x}{d-c}, & c < x < d \\ 0, & x > d \end{cases}$$

Quando  $b = c$ , la funzione triangolare può essere considerata come un caso particolare di quella trapezoidale.

Il sistema fuzzy è composto da tre input e un output crisp (il livello di priorità) e la media ponderata viene utilizzata come metodo di defuzzyficazione. I valori degli insiemi fuzzy sono riportati nella Gli insiemi fuzzy delle tre variabili di input utilizzate per la realizzazione dell'inferenza (rispettivamente la variabile ChargeRequisite, la variabile RechAvailability e la variabile SOC) sono rappresentate graficamente in *Figura 3*, *Figura 4* e *Figura 5*.

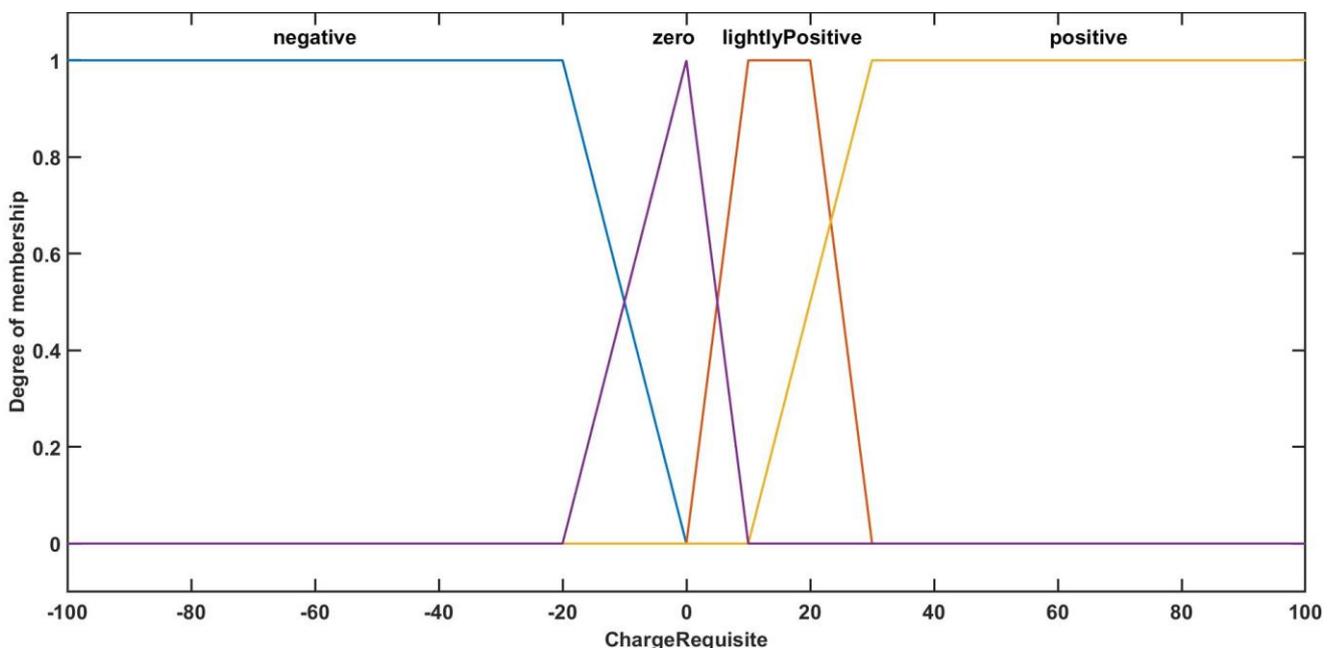


Figura 3 - Insiemi fuzzy trapezoidali e triangolari della variabile ChargeRequisite. In particolare, sono generati 4 insiemi, denominati con variabili linguistiche “negative”, “zero”, “lightlyPositive” e “positive” ad indicare i diversi gradi di differenza tra la SOC attuale e la MINSOC che dovrà esserci alla fine della ricarica.

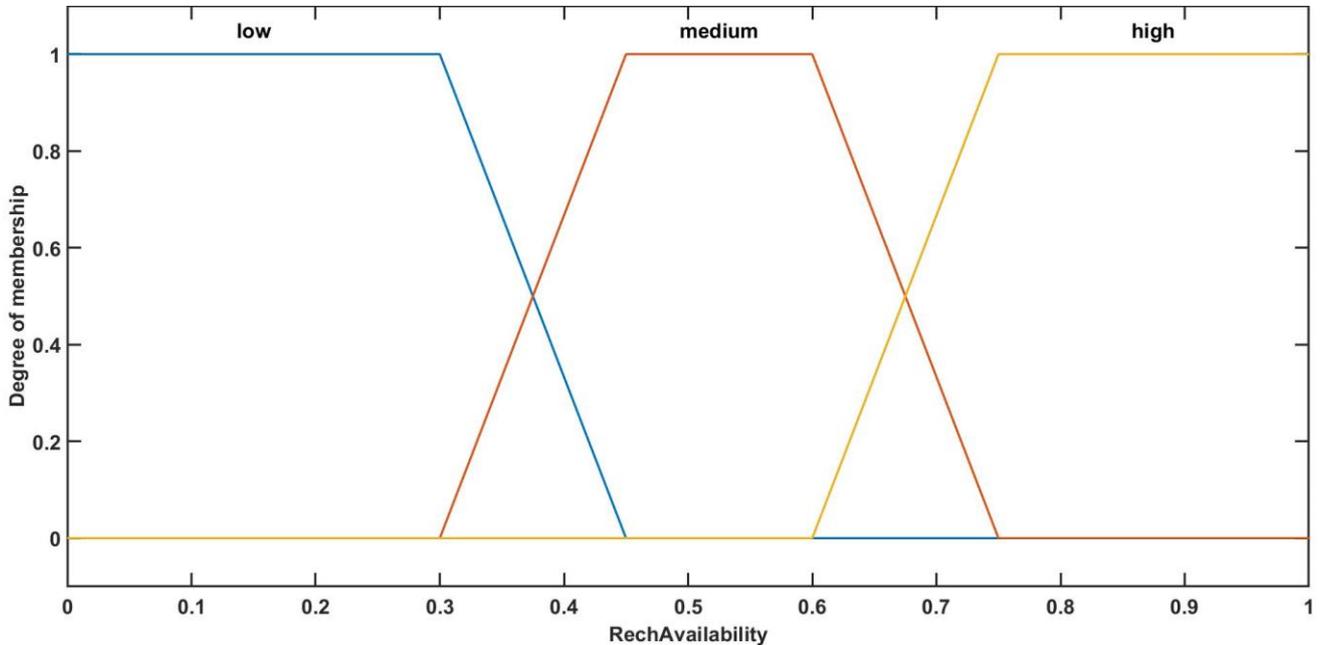


Figura 4 - Insiemi fuzzy trapezoidali della variabile RechAvailability. In particolare sono generati 3 insiemi, denominati con variabili linguistiche “low”, “medium” e “high” ad indicare i concetti di bassa, media o alta capacità residua di ricarica prima del plugin.

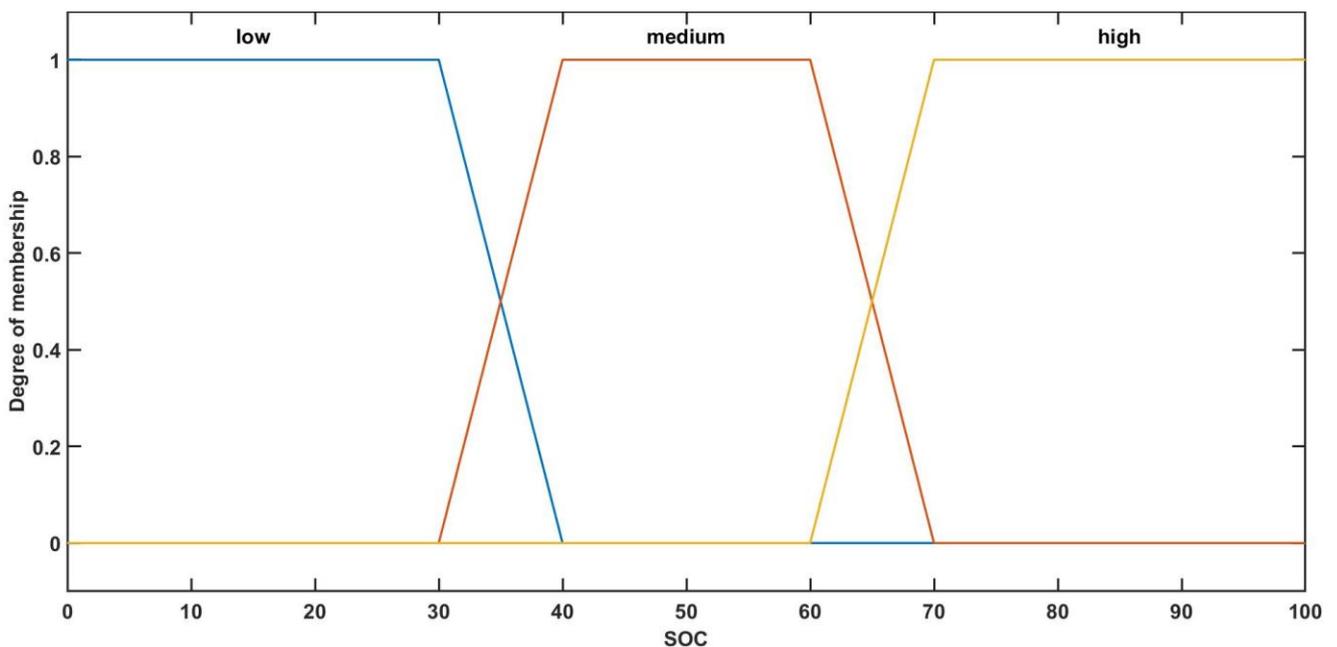


Figura 5 - Insiemi fuzzy trapezoidali della variabile SOC. In particolare, sono generati 3 insiemi, denominati con variabili linguistiche “low”, “medium” e “high” ad indicare i concetti di basso, medio o alto stato di carica della batteria dell’EV.

Tabella 1.

Gli insiemi fuzzy delle tre variabili di input utilizzate per la realizzazione dell’inferenza (rispettivamente la variabile ChargeRequisite, la variabile RechAvailability e la variabile SOC) sono rappresentate graficamente in Figura 3, Figura 4 e Figura 5.

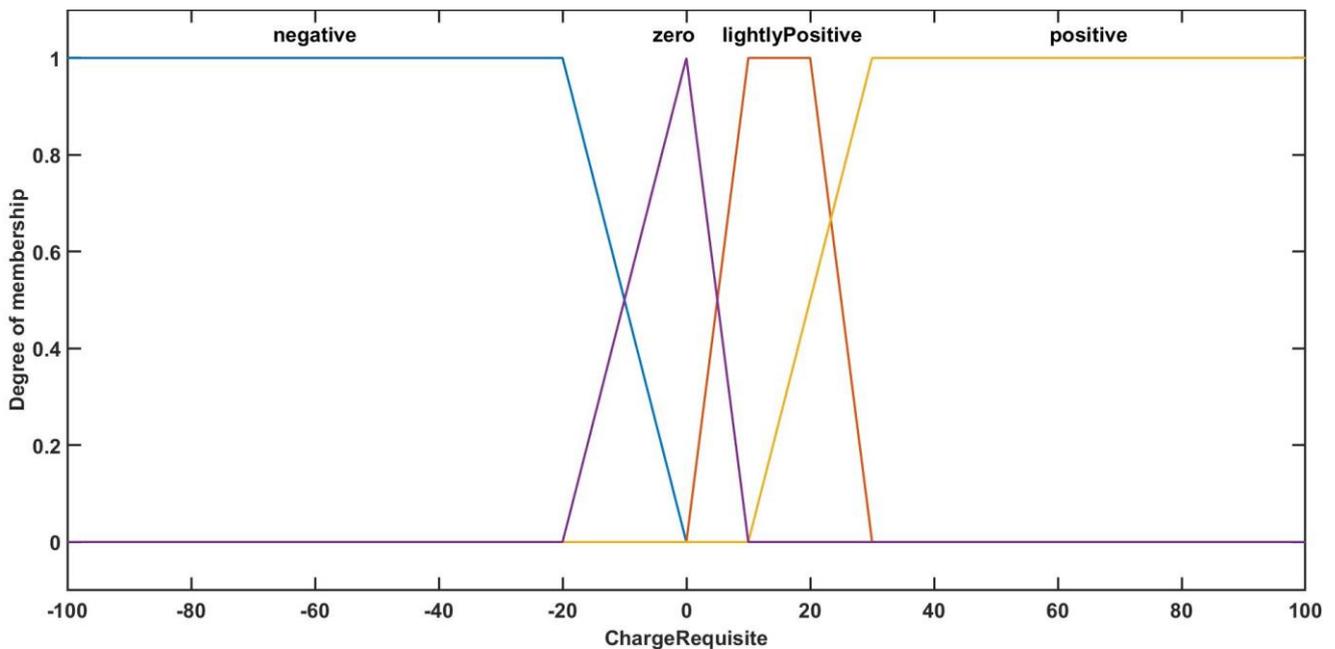


Figura 3 - Insiemi fuzzy trapezoidali e triangolari della variabile ChargeRequisite. In particolare, sono generati 4 insiemi, denominati con variabili linguistiche “negative”, “zero”, “lightlyPositive” e “positive” ad indicare i diversi gradi di differenza tra la SOC attuale e la MINSOC che dovrà esserci alla fine della ricarica.

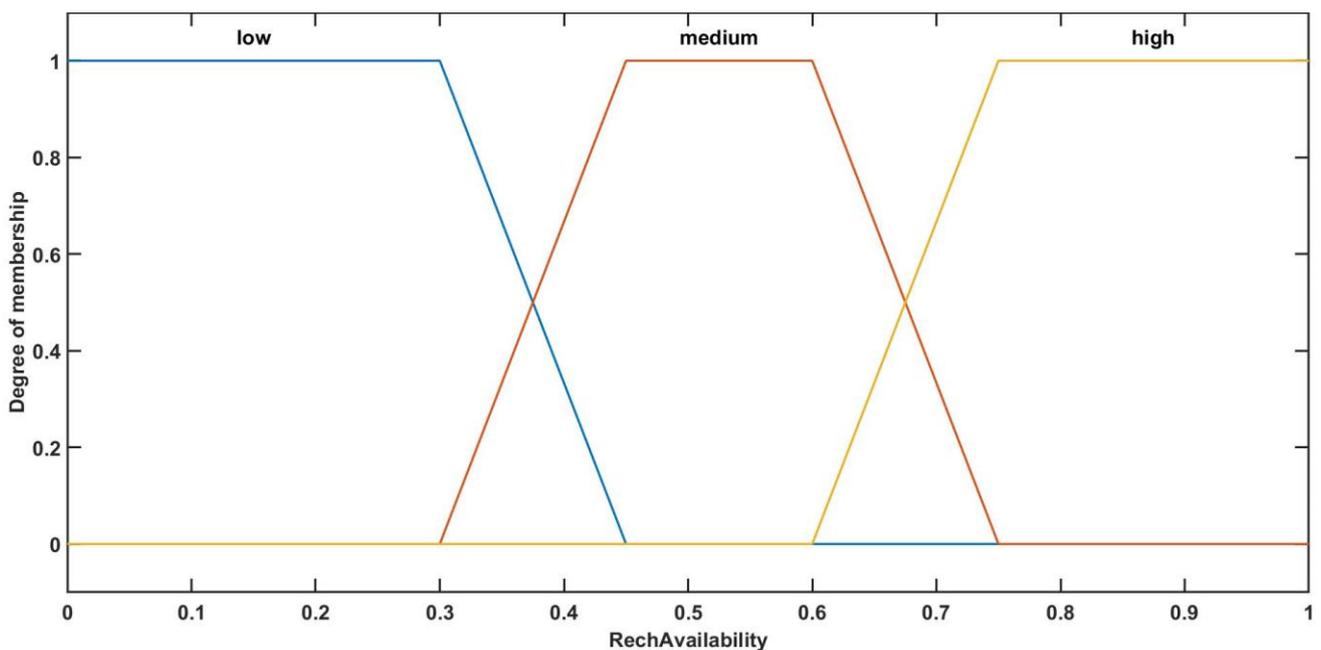


Figura 4 - Insiemi fuzzy trapezoidali della variabile RechAvailability. In particolare sono generati 3 insiemi, denominati con variabili linguistiche “low”, “medium” e “high” ad indicare i concetti di bassa, media o alta capacità residua di ricarica prima del plugout.

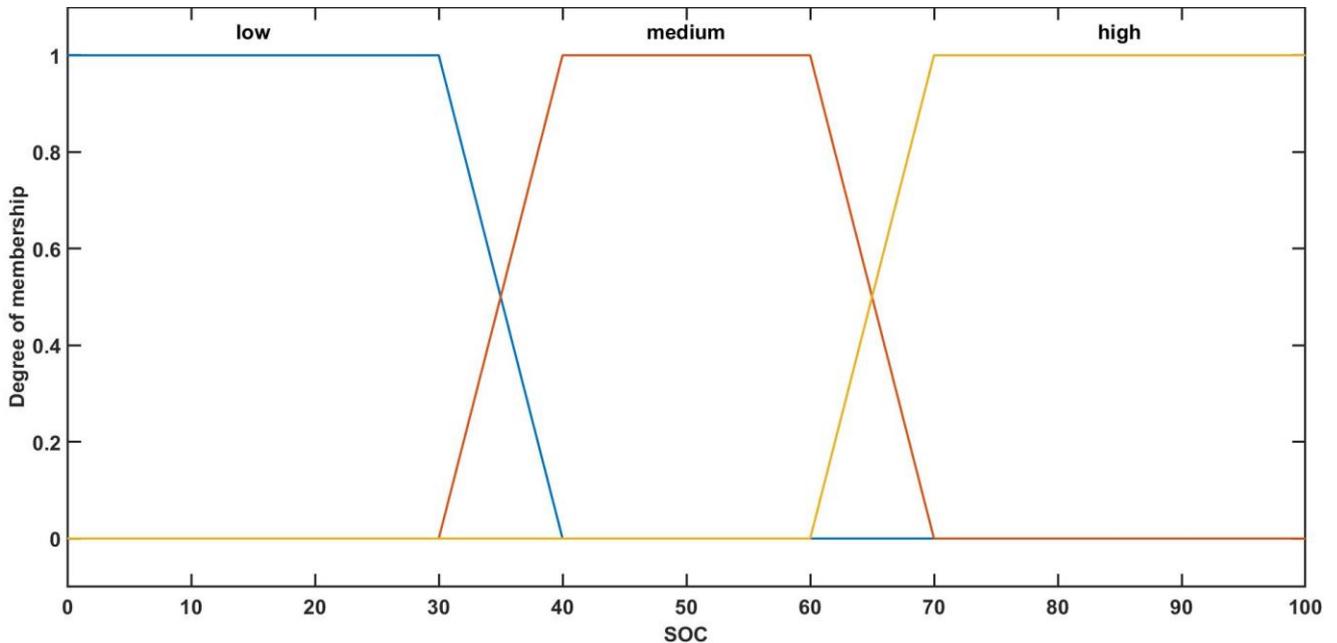


Figura 5 - Insiemi fuzzy trapezoidali della variabile SOC. In particolare, sono generati 3 insiemi, denominati con variabili linguistiche “low”, “medium” e “high” ad indicare i concetti di basso, medio o alto stato di carica della batteria dell’EV.

Tabella 1. Insiemi fuzzy delle variabili di input e output: termini linguistici (linguistic terms) e parametri trapezoidali (a,b,c,d) corrispondenti degli insiemi fuzzy

Variabile di Input	Linguistic terms	Valori di a/b/c/d
ChargeRequisite [%]	Negative	-100/ -100/ -20/ 0
	Zero	-20/ 0/ 0/ 10
	Lightly Positive	0/ 10/ 20/ 30
	Positive	10/ 30/ 100/ 100
RechAvailability [%]	Low	0/ 0/ 30/ 45
	Medium	30/ 45/ 60/ 75
	High	60/ 75/ 100/ 100
SOC [%]	Low	0/ 0/ 30/ 40
	Medium	30/ 40/ 60/ 70
	High	60/ 70/ 100/ 100
Variable di Output	Linguistic terms	Valore crisp
Priority	Null	0
	Low	0,4
	Medium	0,6
	High	0,8
	Absolute	1

Dalle 3 variabili di input sono state definite 36 regole di aggregazione in base all’appartenenza ai diversi insiemi fuzzy. In particolare, la rule base generata, secondo la logica dell’inferenza, in cui il connettore

“and” (&) è rappresentato dall’operatore “minimo” e il connettore “then” (=>) è realizzato con il metodo del troncamento, è la seguente:

If:

- 1 ChargeRequisite==negative & RechAvailability==low & SOC==low => ChargePriority=medium
- 2 ChargeRequisite==negative & RechAvailability==low & SOC==medium => ChargePriority=medium
- 3 ChargeRequisite==negative & RechAvailability==low & SOC==high => ChargePriority=low
- 4 ChargeRequisite==negative & RechAvailability==medium & SOC==low => ChargePriority=medium
- 5 ChargeRequisite==negative & RechAvailability==medium & SOC==medium => ChargePriority=low
- 6 ChargeRequisite==negative & RechAvailability==medium & SOC==high => ChargePriority=null
- 7 ChargeRequisite==negative & RechAvailability==high & SOC==low => ChargePriority=low
- 8 ChargeRequisite==negative & RechAvailability==high & SOC==medium => ChargePriority=low
- 9 ChargeRequisite==negative & RechAvailability==high & SOC==high => ChargePriority=null
- 10 ChargeRequisite==zero & RechAvailability==low & SOC==low => ChargePriority=high
- 11 ChargeRequisite==zero & RechAvailability==low & SOC==medium => ChargePriority=medium
- 12 ChargeRequisite==zero & RechAvailability==low & SOC==high => ChargePriority=low
- 13 ChargeRequisite==zero & RechAvailability==medium & SOC==low => ChargePriority=high
- 14 ChargeRequisite==zero & RechAvailability==medium & SOC==medium => ChargePriority=medium
- 15 ChargeRequisite==zero & RechAvailability==medium & SOC==high => ChargePriority=low
- 16 ChargeRequisite==zero & RechAvailability==high & SOC==low => ChargePriority=medium
- 17 ChargeRequisite==zero & RechAvailability==high & SOC==medium => ChargePriority=low
- 18 ChargeRequisite==zero & RechAvailability==high & SOC==high => ChargePriority=low
- 19 ChargeRequisite==lightlyPositive & RechAvailability==low & SOC==low => ChargePriority=absolute
- 20 ChargeRequisite==lightlyPositive & RechAvailability==low & SOC==medium => ChargePriority=high
- 21 ChargeRequisite==lightlyPositive & RechAvailability==low & SOC==high => ChargePriority=medium
- 22 ChargeRequisite==lightlyPositive & RechAvailability==medium & SOC==low => ChargePriority=high
- 23 ChargeRequisite==lightlyPositive & RechAvailability==medium & SOC==medium => ChargePriority=medium
- 24 ChargeRequisite==lightlyPositive & RechAvailability==medium & SOC==high => ChargePriority=low
- 25 ChargeRequisite==lightlyPositive & RechAvailability==high & SOC==low => ChargePriority=medium

- 
- 26 ChargeRequisite==lightlyPositive & RechAvailability==high & SOC==medium => ChargePriority=medium
  - 27 ChargeRequisite==lightlyPositive & RechAvailability==high & SOC==high => ChargePriority=medium
  - 28 ChargeRequisite==positive & RechAvailability==low & SOC==low => ChargePriority=absolute
  - 29 ChargeRequisite==positive & RechAvailability==low & SOC==medium => ChargePriority=absolute
  - 30 ChargeRequisite==positive & RechAvailability==low & SOC==high => ChargePriority=absolute
  - 31 ChargeRequisite==positive & RechAvailability==medium & SOC==low => ChargePriority=absolute
  - 32 ChargeRequisite==positive & RechAvailability==medium & SOC==medium => ChargePriority=high
  - 33 ChargeRequisite==positive & RechAvailability==medium & SOC==high => ChargePriority=high
  - 34 ChargeRequisite==positive & RechAvailability==high & SOC==low => ChargePriority=absolute
  - 35 ChargeRequisite==positive & RechAvailability==high & SOC==medium => ChargePriority=high
  - 36 ChargeRequisite==positive & RechAvailability==high & SOC==high => ChargePriority=high

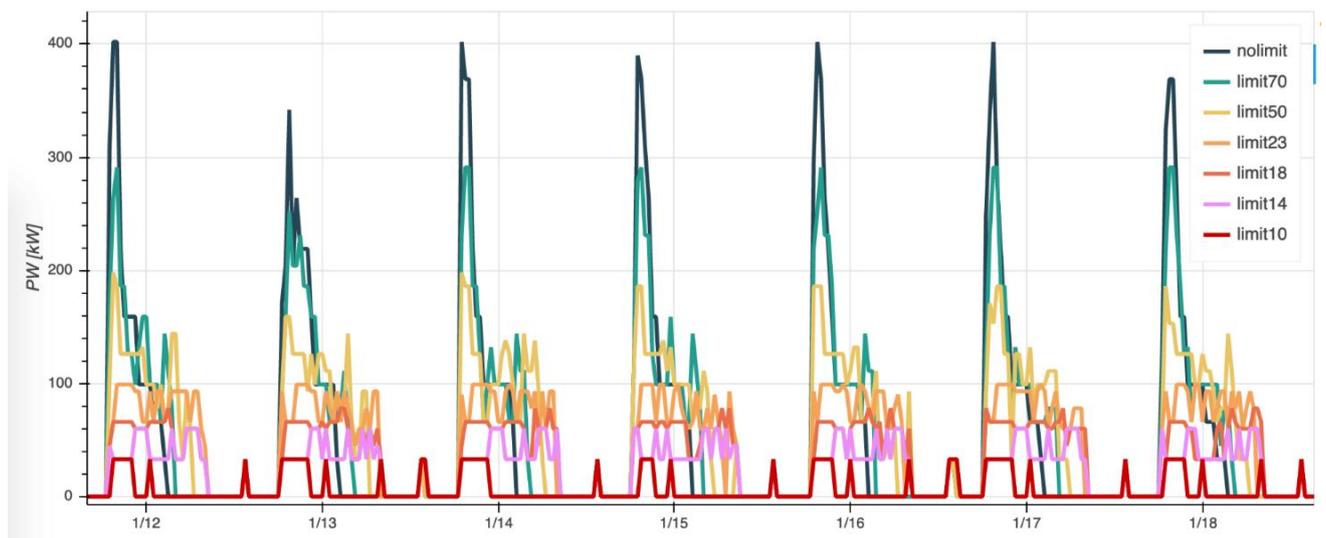
## 4 RISULTATI

Sono considerati diversi scenari con un limite di potenza variabile. In particolare, il cap è fissato rispettivamente al 70%, 50%, 23%, 18%, 14% e 10% della potenza massima richiesta quando tutti i veicoli stanno caricando contemporaneamente (pari a 450 kW nel presente scenario simulato).

Per ogni test viene calcolata una serie di indicatori di performance per valutare l'efficacia della soluzione in termini di "buffer di carica", ovvero la differenza tra il SOC al momento del plugout e il SOC minimo richiesto per eseguire la successiva distanza di pendolarismo:

$$buff = SOC - MINSOC$$

I risultati sono riportati nella *Tabella 2*. Un grafico di una settimana che descrive la potenza istantanea globale fornita dalla rete nei diversi scenari di limite di potenza è mostrato nella *Figura 6*.



*Figura 6. Una settimana di profili di ricarica globali. I limiti di potenza rappresentati sono: 100 % (cioè nessun limite, linea blu), 70 % (linea ciano), 50 % (linea verde), 23 % (linea gialla), 18 % (linea gialla arancione), 14 % (linea arancione) e 10 % (linea rossa).*

In particolare, la *Figura 6* mostra come all'aumentare del vincolo di peak-shaving (limite di potenza con % sempre inferiore) il profilo di ricarica tende a distribuirsi nelle ore successive secondo una logica di load shifting. Infatti, finché il limite di potenza lo consente, il "servizio" di ricarica viene garantito anche se in un intervallo di tempo più lungo, caricando il parco di auto elettriche alla capacità richiesta con una potenza più bassa. Tuttavia, se il limite di potenza scende troppo c'è il rischio che l'Aggregatore non riesca a garantire il "servizio" di ricarica minima richiesto da ciascun utente creando un potenziale disservizio.

Infatti, come riportato nella *Tabella 2*, l'allocazione delle priorità di ricarica basata su fuzzy garantisce un buffer medio positivo fino a un limite di potenza del 14%. Per valori limite uguali o inferiori al 10% il buffer medio risulta negativo, implicando requisiti non soddisfatti per un certo numero di EV al momento del plugout (partenza mattutina o post-pranzo).

La colonna buff-perc mostra la percentuale di eventi pluginout (tra tutti gli EV e durante l'intero periodo di prova) in cui i requisiti degli EV sono completamente soddisfatti. Intorno al limite del 23% mostra un 13,3% di eventi di pluginout in cui il valore è negativo nonostante il valore medio del buffer positivo. Ciò può implicare alcuni valori assoluti bassi di **buff** distribuiti nella vasta gamma di veicoli, determinando così una diminuzione del valore di buff-perc pur mantenendo valori buff-mean positivi.

Tabella 2. I valori del buffer di carica (**buff**), calcolati come riportato in precedenza, sono riportati in termini di media, mediana, deviazione standard, valore minimo e valore massimo sull'intero numero di cicli di carica ed EV. Il termine buff-perc indica la percentuale di EV che sono al di sopra del valore MINSOC alla fine di ogni ciclo di carica. Le ultime due colonne della tabella riportano la media e la deviazione standard dell'istante potenza fornita dal regolatore di potenza.

Test	buff-mean [%]	buff-median [%]	buff-min [%]	buff-max [%]	buff-perc [%]	PW-max [kW]
nolimit	66,8	66,2	9,8	97,9	100	435
limit70	66,8	66,2	5,2	97,9	100	297
limit50	61,4	54,3	4,9	97,9	100	198
limit23	30,2	17,8	-13,3	97,9	80,3	99
limit18	11,4	-4,9	-37,8	97,9	32,2	78
limit14	2,9	-9,3	-60,8	97,9	24,7	60
limit10	-9,2	-26,5	-85,2	97,9	23,4	33

Infine, nella *Figura 7* è possibile apprezzare la logica implementata in azione durante un'effettiva sessione di ricarica notturna per un piccolo lotto di veicoli (10 EV).

La *Figura 7* mostra la validità dell'algorithmo nell'individuare, una volta noti i vincoli, le priorità di ricarica che il potenziale Aggregatore deve assegnare ai veicoli elettrici collegati alle colonnine. Il processo mostrato in *Figura* è svolto in maniera dinamica; ad ogni step di campionamento l'algorithmo di ottimizzazione viene rilanciato e vengono ricalcolate/riaggiornate le priorità di carica.

La *Figura 7* mostra l'andamento del profilo di carica di un campione di 10 veicoli elettrici, prima di commentare l'andamento di ciascuno, alcune considerazioni generali possono essere fatte:

- L'altezza del gradino di incremento della SOC varia a seconda del tipo di colonnina. Sono evidenti quattro differenti altezze dei gradini, corrispondenti ad altrettante potenze: 3.3 kW, 4.5 kW, 6 kW e 11 kW
- L'inizio della serie (curva) corrisponde al momento di connessione dell'EV alla colonnina di ricarica (plugin time); la fine della serie corrisponde al momento della disconnessione.
- Il sampling time è di 30 minuti;
- Al momento della connessione (plugin in) il veicolo passa tutte le informazioni all'algorithmo; durante il primo intervallo di tempo non è prevista nessuna azione; l'azione avviene, se prevista al primo time stamp utile previsto; infatti, le colonnine iniziano la ricarica non immediatamente ma dopo un istante di campionamento, che è il tempo necessario per elaborare le nuove informazioni arrivate dal nuovo veicolo e ricalcolare le nuove priorità di carica;

- Ogni volta che la carica si interrompe dopo una rampa ed inizia un plateau è perché il veicolo elettrico ha raggiunto un livello di priorità più basso di un altro veicolo;

Per quanto riguarda, il profilo dei singoli veicoli, si possono evidenziare le seguenti osservazioni:

- EV1: al momento della connessione, la batteria è già abbastanza carica e quella carica è probabilmente già sufficiente per il giorno dopo; questo spiega perché il veicolo inizia la ricarica molto tardi, a valle di tutti gli altri;
- EV2: parte da una SOC bassissima ma non si attacca subito perché ha una colonnina più potente (6 kW); inoltre, altri veicoli hanno priorità maggiore; non dovendo arrivare ad un SOC elevatissima ed essendo il tempo stimato di plugout sufficiente a raggiungere l'obiettivo, inizia la ricarica più tardi; altri veicoli connessi più tardi hanno una priorità maggiore;
- EV3: ha un comportamento simile ad EV2 ma con una colonnina di ricarica ancora più potente (11 kW); possono essere fatte le stesse considerazioni dell'EV2;
- EV4: ha ricaricato l'auto prima di entrare in casa e la collega pur essendo carica al 100%; il motivo per cui questo EV si connette alla colonnina anche se la sua carica è quello di voler/poter fornire flessibilità alla microrete domestica o alla rete elettrica nazionale nel caso di V2B o V2G;
- EV5: Si comporta come EV2 con minor priorità perché ha minor MINSOC (SOC di carica minima richiesta dall'utente al momento della disconnessione della colonnina) e ha una SOC maggiore al momento della connessione alla colonnina;
- EV6: al momento della connessione alla colonnina di ricarica la SOC della batteria è abbastanza alta; la sua MINSOC piuttosto bassa; ha, quindi, una priorità inferiore rispetto agli altri EV, anche connessi in istanti successivi; la potenza della sua colonnina è bassa (3.3 kW);
- EV7: al momento della connessione alla colonnina di ricarica il SOC della batteria è molto basso; ha un MINSOC alto; la colonnina di ricarica ha una potenza di 4.5 kW; al primo istante disponibile (dopo che l'algoritmo ha ricalcolato le priorità) inizia la carica con una priorità molto elevata rispetto agli altri EV; la priorità si riduce man mano che il SOC si avvicina al MINSOC;
- EV8: al momento della connessione alla colonnina di ricarica il SOC arriva con una certa capacità residua della batteria (SOC > 20 %); altri EV hanno comunque maggiore priorità perché la sua MINSOC è più bassa di altre e la sua colonnina è più potente di altre (6 kW);
- EV9: ha lo stesso comportamento dell'EV7;
- EV 10 arriva per primo con un SOC piuttosto basso; il primo momento utile comincia la ricarica e mantiene una priorità alta per molto tempo; quando il suo SOC si avvicina al MINSOC, perde la priorità rispetto ad altri veicoli.

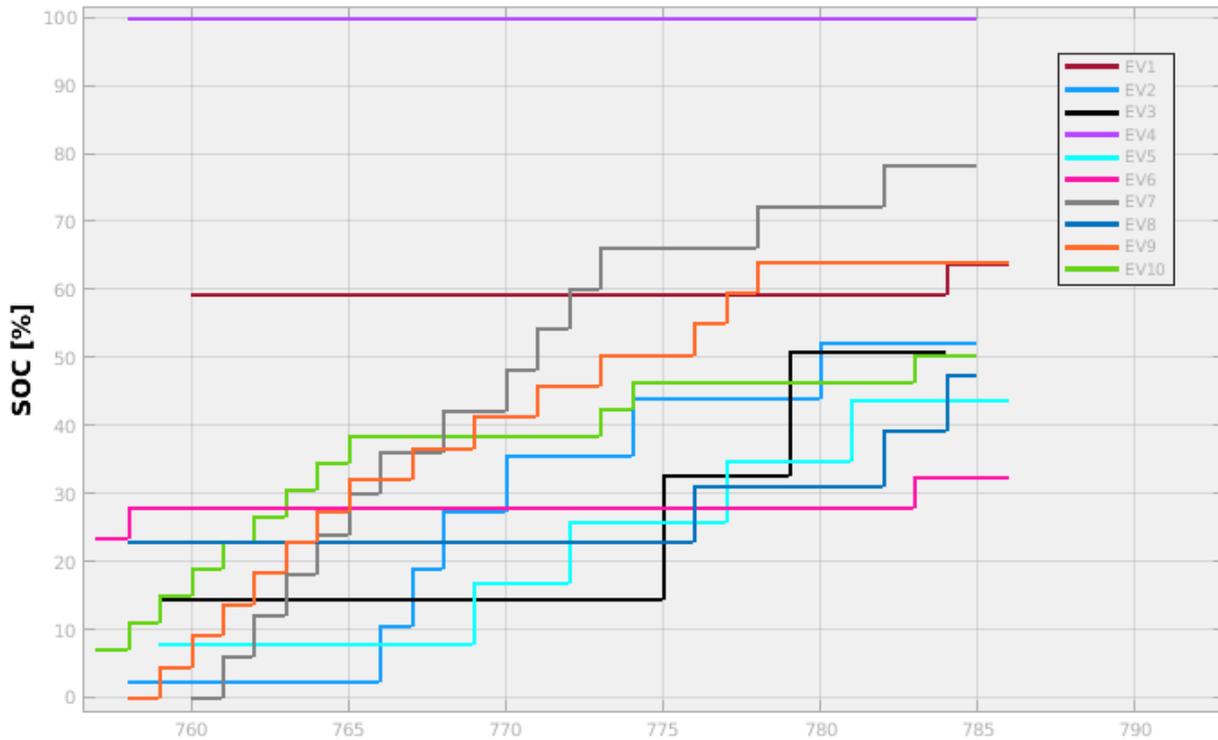


Figura 7. Estratto di una simulazione che illustra il comportamento di EV1, EV2 ed EV3 quando vengono fermati fino a quando i requisiti minimi più urgenti non vengono soddisfatti da altri veicoli, come EV7.

## 5 CONCLUSIONI

Questo deliverable descrive possibili strategie per la gestione ottimale di un aggregato di colonnine. Il risultato principale di questo deliverable è l'“Aggregator online Assitant” (AOA), un blocco Simulink che permette di importare i risultati dei profili di colonnine aggregati sviluppati dal software presentato nei deliverable A3.6.a (ePopSimulator, singola colonnina) e A3.6b (ePopSimulator, colonnine aggregate) e di poterli gestire/interfacciare con algoritmi di ottimizzazione esterni. Infatti, l'AOA permette di inviare le informazioni (stato di carica, potenza impegnata, durata prevista di ricarica, potenza colonnina...) di tutte le colonnine simulate all'Aggregatore; queste informazioni potranno essere elaborate da un qualsivoglia algoritmo di ottimizzazione che implementi una o più funzioni obiettivo; i risultati dell'ottimizzazione, in termini di scheduling di carica e scarica, verranno di nuovo inviati all'AOA che provvederà a ridefinire il nuovo profilo ottimizzato e a ricominciare il loop. Grazie all'AOA, sarà quindi possibile utilizzare il software di simulazione ePopSimulator per testare, in assenza di dati reali, una molteplicità di strategie di gestione corrispondenti a diverse tipologie di algoritmi di ottimizzazione e funzioni obiettivo sviluppati da soggetti terzi (soggetti Aggregatori, DSO, Centri di Ricerca, Università) per contesti nazionali o internazionali.

Lo scopo del deliverable è quello di proporre una metodologia per gestire le priorità di attivazione che il soggetto Aggregatore dovrà assegnare ai veicoli elettrici collegati alle colonnine in modo da ottenere determinati obiettivi. In particolare, in questo deliverable la metodologia proposta per l'identificazione delle priorità di attivazione da assegnare ai veicoli elettrici verrà applicata all'obiettivo di ottenere un abbassamento della potenza massima assorbita dalla rete elettrica (peak shaving) cercando di minimizzare i costi di gestione (di ricarica).

Il processo è gestito in maniera dinamica; ad ogni step di campionamento l'algoritmo di ottimizzazione viene rilanciato e vengono ricalcolate/riaggornate le priorità di carica, in base alle nuove informazioni inviate dalle colonnine (stato aggiornato batterie, nuove auto connesse, e auto disconnesse nel frattempo...). I risultati mostrano la validità dell'algoritmo nell'individuare, una volta noti i vincoli, le priorità di ricarica che il potenziale Aggregatore deve assegnare ai veicoli elettrici collegati alle colonnine. In particolare, all'aumentare del vincolo di peak-shaving (limite di potenza con % sempre inferiore) il profilo di ricarica tende a distribuirsi nelle ore successive secondo una logica di load shifting. Infatti, finché il limite di potenza lo consente, il “servizio” di ricarica viene garantito anche se in un intervallo di tempo più lungo, caricando il parco di auto elettriche alla capacità richiesta con una potenza più bassa. Tuttavia, se il limite di potenza scende troppo c'è il rischio che l'Aggregatore non riesca a garantire il “servizio” di ricarica minima richiesto da ciascun utente creando un potenziale disservizio. Infatti, l'allocazione delle priorità di ricarica basata su fuzzy garantisce un buffer medio positivo fino a un limite di potenza del 14%. Per valori limite uguali o inferiori al 10% il buffer medio risulta negativo, implicando requisiti non soddisfatti per un certo numero di EV al momento del plugout (partenza mattutina o post-pranzo).

In conclusione, questo deliverable completa tutta l'attività A3.6 del progetto ComESTo, relativa allo studio, modellazione/simulazione e definizione di strategie per la gestione di veicoli elettrici, fornendo uno strumento flessibile per la simulazione dei profili di consumo di colonnine di ricarica singole o in forma aggregata, in un contesto in cui è evidente la carenza di dati disponibili per sviluppare strategie di ottimizzazione e business models su colonnine di ricarica. L'AOA consente di interfacciare i dati simulati con possibili algoritmi di ottimizzazione specifici di ciascun Aggregatore o utente finale.

## 6 BIBLIOGRAFIA

Ciabattoni Lucio; Grisostomi, Massimo; Ippoliti, Gianluca; Longhi, Sauro, *Fuzzy logic home energy consumption modeling for residential photovoltaic plant sizing in the new Italian scenario*, Energy,74,359-367,2014, Elsevier

Ciabattoni, Lucio; Ferracuti, Francesco; Grisostomi, Massimo; Ippoliti, Gianluca; Longhi, Sauro, *Fuzzy logic based economical analysis of photovoltaic energy management*, Neurocomputing,170,296-305,2015, Elsevier

Ciabattoni, Lucio; Grisostomi, Massimo; Ippoliti, Gianluca; Pagnotta, D Proietti; Foresi, Gabriele; Longhi, Sauro, *Residential energy monitoring and management based on fuzzy logic*,2015 IEEE International Conference on Consumer Electronics (ICCE),536-539,2015, IEEE