



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



**DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE**

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

**SVILUPPO DI UN SISTEMA DI VIRTUALIZZAZIONE DELLA RICARICA DI VEICOLI
ELETTRICI PER SISTEMI CLUSTER**

Relatore: Prof. Nicola Trivellin

Laureando: Giovanni Tomasi

Supervisore aziendale: Ing. Michele Brunelli

ANNO ACCADEMICO 2022 - 2023

Data di Laurea 13 Aprile 2023

Sommario

Questo lavoro di tesi, svolto presso l'azienda Silla Industries, tratta lo sviluppo di un sistema di virtualizzazione della ricarica per cluster di veicoli elettrici andando ad occuparsi, nello specifico, della gestione della simulazione delle colonnine di ricarica coinvolte e della modellizzazione matematica della risposta alla ricarica di vari modelli di veicoli elettrici. Innanzitutto, viene presentata un'introduzione sulle colonnine di ricarica e sulla ricarica dei veicoli elettrici. In seguito, viene esposta la struttura del simulatore, andando a soffermarsi sulle varie parti che lo compongono. Successivamente, viene riportato lo studio dei dati raccolti relativi alle curve di ricarica per vari modelli di veicoli elettrici. Sulla base di questo studio sono state create due tipologie di schemi Simulink per la simulazione della ricarica: una prima tipologia di schema che permette la modellizzazione delle dinamiche di risposta alla ricarica che sono comuni fra i vari modelli di EVs e una seconda serie di schemi che permette di simulare la dinamica di risposta alla ricarica specifica per ogni modello di EV preso in considerazione. Vengono quindi riportate le procedure di modellizzazione matematica dei vari modelli di EVs citate precedentemente per poi proseguire con una fase di confronto tra i dati di ricarica reali e quelli ottenuti tramite la simulazione, al fine di provare l'efficacia delle varie modellizzazioni. Verrà poi discussa l'integrazione dei modelli matematici nel sistema di virtualizzazione di colonnine di ricarica per terminare con la fase di test del sistema completo.

Indice

Sommario	3
Introduzione	7
1 Ricarica dei veicoli elettrici	9
1.1 On-board-charger e processo di ricarica	9
1.2 Infrastrutture di ricarica	12
1.2.1 Normative legate alle infrastrutture di ricarica	13
2 Sistema di virtualizzazione delle colonnine di ricarica	21
2.1 Struttura del simulatore	22
2.2 Applicazione Qt Creator	23
2.2.1 Gestione delle comunicazioni CAN e ModBus RTU	25
2.3 Broker dati Redis	25
3 Curve di ricarica degli EV	29
3.1 Caratteristiche dinamiche Opel Corsa-e	32
3.2 Caratteristiche dinamiche Hyundai Ioniq 5 RWD Progress	33
3.3 Caratteristiche dinamiche Tesla Model S (2014)	34
3.3.1 Tipi di dinamica	36
3.4 Raccolta dati con curve di ricarica alternative	37
3.4.1 Hyundai Ioniq 5 RWD Progress: curve di ricarica alternative	39
3.4.2 Tesla Model S (2014): curve di ricarica alternative	39
4 Modellizzazione matematica degli EV	43
4.1 Modello matematico per veicolo generico	43
4.1.1 Sottosistema <i>Current command actions</i>	44
4.1.2 Sottosistema <i>Power calculation</i>	46

4.1.3 Sottosistema <i>Energy calculation</i>	47
4.2 Modello matematico Hyundai Ioniq 5 RWD Progress	48
4.3 Modello matematico Tesla Model S (2014)	51
5 Integrazione degli schemi Simulink nel sistema di virtualizzazione di colonnine di	
ricarica	55
5.1 Gestione delle simulazioni di ricarica	55
5.2 Generazione del codice degli schemi Simulink e dei file eseguibili	56
5.2.1 Generazione del codice con Embedded Coder	57
5.2.2 Importazione del codice C in Visual Studio e generazione dei file ese-	
guibili	58
6 Procedure di test	61
6.1 Test della comunicazione su bus CAN	61
6.2 Test della comunicazione basata su protocollo ModBus RTU	62
6.3 Test dei file eseguibili dal simulatore	63
6.4 Test del sistema completo con il dispositivo Unità Gestore	65
Appendice A	73
Listati delle funzioni MatLab contenute negli schemi Simulink	73
Bibliografia	90

Introduzione

La mobilità elettrica si è confermata come una delle direzioni da seguire al fine di garantire una rivoluzione dell'attuale sistema dei trasporti verso una soluzione più sostenibile. Per adattarsi a questa trasformazione, i produttori di automobili, incentivati dai vincoli introdotti per agevolarla, hanno investito in questo settore progettando nuovi modelli di veicoli elettrici sempre più all'avanguardia. Vista la forte espansione di questo mercato, è nata la necessità di creare delle normative atte ad unificare le linee guida che i produttori di infrastrutture di ricarica devono seguire per consentire l'interoperabilità dei sistemi coinvolti. Per quanto riguarda le colonnine di ricarica AC, sono d'interesse normative come la IEC 61851 e la IEC 62196, che permettono di definire le linee guida per la progettazione dei suddetti punti di ricarica. Tuttavia, questi standard, non regolano l'andamento della corrente assorbita dall'on-board-charger del veicolo elettrico. Dall'esperienza maturata da Silla Industries, azienda produttrice di dispositivi per la ricarica di veicoli elettrici in corrente alternata, è sorto che durante il processo di ricarica, il valore efficace della corrente assorbita dal veicolo elettrico, assume dei profili che si discostano dal valore di corrente imposto dalla colonnina e che sono differenti a seconda del modello di veicolo che si sta tenendo in considerazione. Il sistema ottenuto tramite questo lavoro di tesi e tirocinio, ha permesso di analizzare i profili di corrente di ricarica relativi a una serie di modelli di veicoli elettrici al fine di definire un modello matematico che possa riprodurli. I modelli ottenuti sono stati poi integrati nel sistema di virtualizzazione della ricarica che ha permesso di simulare un cluster di veicoli elettrici collegati alla rete tramite la simulazione delle colonnine elettriche ad essi collegate. Il sistema ottenuto, offre la possibilità di simulare la risposta di un singolo veicolo alla ricarica, come anche la possibilità di simulare un numero elevato di veicoli elettrici connessi alla rete, al fine di riprodurre contesti di ricarica che possono essere riscontrati nella quotidianità di parcheggi di aziende, centri commerciali, condomini e hotel. Questo sistema di simulazione potrà essere utilizzato per studiare soluzioni atte a rendere più efficiente la ricarica dei veicoli elettrici e, inoltre, permetterà di simulare la richiesta di energia di un cluster di veicoli elettrici connessi alla rete al fine di facilitare lo studio di sistemi volti a gestire queste

situazioni di richiesta di energia elettrica in grande quantità.

Capitolo 1

Ricarica dei veicoli elettrici

In questo Capitolo verrà presentata una panoramica su ciò che riguarda la ricarica dei veicoli elettrici. Dato che questo progetto di tesi e tirocinio tratta la virtualizzazione di un cluster di colonnine di ricarica in corrente AC, inizialmente verrà introdotto l'on-board-charger del veicolo elettrico per poi proseguire con le infrastrutture di ricarica e con le normative che le regolano.

1.1 On-board-charger e processo di ricarica

Uno dei componenti fondamentali per la ricarica dei veicoli elettrici in corrente alternata, è l'on-board-charger. Questo sistema, integrato a bordo del veicolo, permette la conversione di corrente e tensione da AC a DC. Nella ricarica del veicolo elettrico, l'on-board-charger si colloca fra il caricatore e la batteria assicurando efficienza e affidabilità per garantire un tempo di ricarica che sia il più breve possibile, rispettando, inoltre, i vincoli di peso e dimensioni necessari [10]. La struttura base dell'on-board-charger, riportata in Figura 1.1 [1] [19], è composta da:

- un filtro AC d'ingresso per l'attenuazione delle armoniche di corrente di ordine superiore introdotte dal caricatore;
- un convertitore AC/DC seguito da un filtro capacitivo;
- un convertitore DC/DC per la regolazione di corrente e tensione fornite alla batteria durante il processo di ricarica;
- un filtro DC di output, solitamente composto da un induttore, per attenuare i disturbi in alta frequenza introdotti dal convertitore DC/DC.

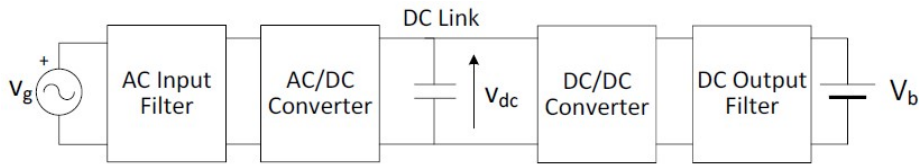


Figura 1.1: Topologia base dell'on-board-charger [1]

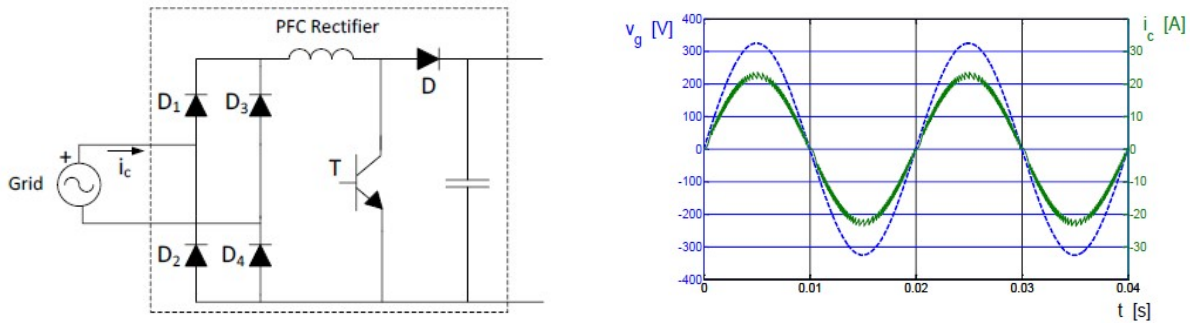


Figura 1.2: PFC rectifier [1]

Una possibile configurazione del convertitore AC/DC per sistemi monofase, può essere quella del PFC rectifier, riportata in Figura 1.2. Questo tipo di convertitore AC/DC garantisce un assorbimento della corrente approssimabile ad un segnale sinusoidale (caratterizzato da una bassa distorsione causata da un ripple sovrainposto dovuto allo switching del transistor T) e in fase con la tensione di rete, assicurando un power factor molto prossimo al valore unitario.

Al fine di garantire l'isolamento galvanico, l'on-board-charger può essere fornito di un convertitore isolato (vedi Figura 1.3).

Gli on-board-charger più moderni vengono resi compatibili con la tecnologia V2G (Vehicle-to-Grid). Questa tecnologia permette di far fluire l'energia immagazzinata nella batteria del veicolo elettrico verso la rete. Tramite questo servizio è possibile sia supportare la rete con un'operazione di load leveling, fornendole energia nei casi in cui questa non riesca a soddisfare la richiesta di energia elettrica, che effettuare una on-line regulation al fine di garantire tensioni e frequenze di rete stabili. Nel caso sia necessario rendere l'on-board-charger compatibile con questa tecnologia, è necessario garantire il flusso di potenza bidirezionale. A tal proposito, è necessario utilizzare convertitori AC/DC e DC/DC bidirezionali (vedi Figura 1.4).

Nella ricarica AC, l'on-board-charger regola tensioni e correnti al fine di rispettare un corretto processo di ricarica che preservi la vita della batteria. Questo processo consiste in due fasi (vedi Figura 1.5):

Carica a corrente costante: in questa prima fase della ricarica la caratteristica tensione-tempo

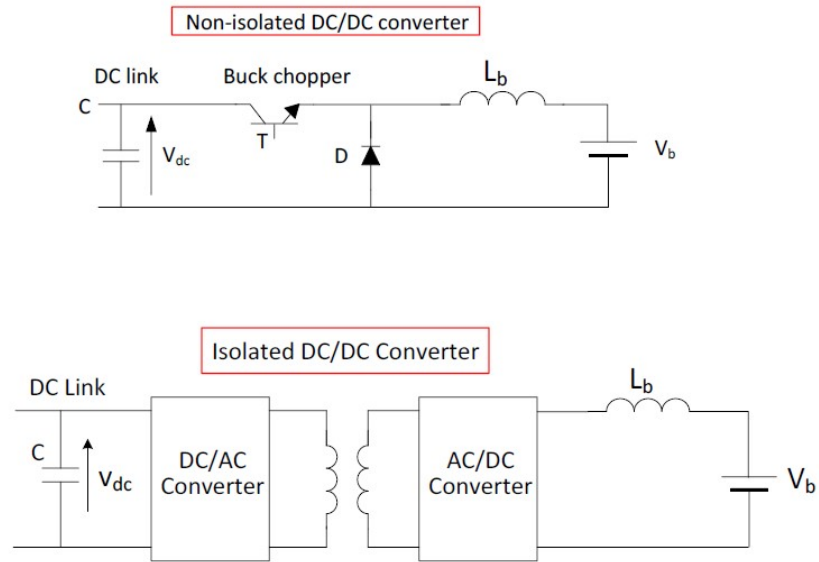


Figura 1.3: Convertitore DC/DC e DC/DC isolato [1]

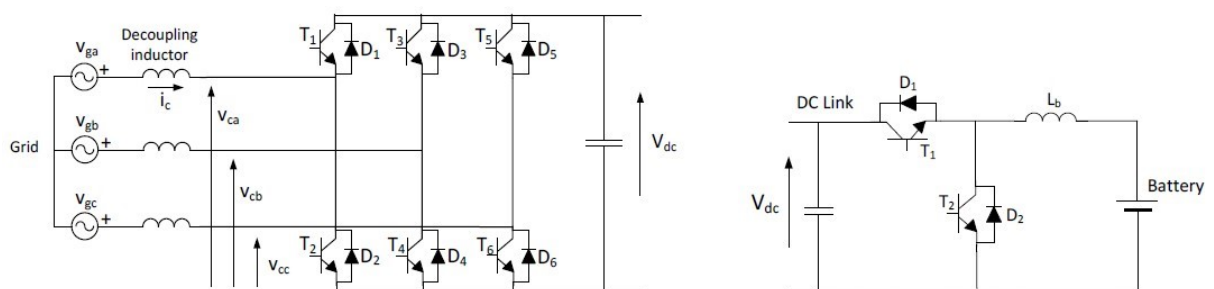


Figura 1.4: Convertitori AC/DC e DC/DC bidirezionali [1]

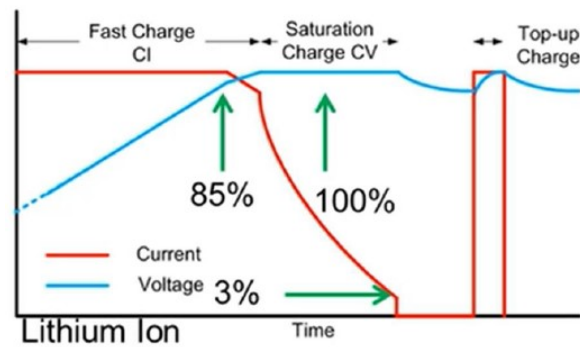


Figura 1.5: Processo di ricarica delle batterie agli ioni di litio [19]

assume un andamento quasi lineare;

Carica a tensione costante: al raggiungimento del valore massimo di tensione, quest'ultima viene mantenuta costante lasciando decrescere la corrente. Una volta che la corrente raggiunge il 3 % del valore massimo, la ricarica viene interrotta. Occasionalmente, può verificarsi una fase di topping charge al fine di mantenere la batteria carica [19].

1.2 Infrastrutture di ricarica

Nell'ambito della mobilità elettrica, le stazioni di ricarica ricoprono un ruolo essenziale per garantire l'integrazione con la rete elettrica e l'uso quotidiano dei veicoli elettrici. Nel caso di ricarica conduttiva, l'infrastruttura di ricarica fa uso di un cavo, una presa e un connettore. A tal proposito, vengono utilizzate due soluzioni: outlet e permanente. Nel caso della soluzione outlet, il cavo di ricarica è integrato a bordo del veicolo, mentre nella soluzione permanente, il cavo di ricarica è integrato nella colonnina di ricarica.

Per quanto riguarda la terminologia che viene utilizzata per descrivere i dispositivi impiegati nella ricarica dei veicoli elettrici troviamo la seguente distinzione:

- l'EVSE (Electric Vehicle Supply Equipment), che comprende i conduttori, la presa di ricarica e tutti gli altri accessori, dispositivi e apparecchiature necessari per garantire lo scambio di energia e la comunicazione fra la colonnina di ricarica e il veicolo elettrico;
- la stazione di ricarica, che permette di erogare l'energia necessaria per la ricarica AC o DC, garantendo la presenza di speciali funzioni di controllo della ricarica (questa è compresa nella EVSE) [1].

1.2.1 Normative legate alle infrastrutture di ricarica

Gli standard che regolano i sistemi e i dispositivi legati alla ricarica dei veicoli elettrici sono l'IEC (International Electrotechnical Commission) 61851 e l'IEC 62196 [15]. Questi standard si occupano prevalentemente di regolamentare i Modi di ricarica, i Casi di connessione e i Tipi di connettori. In particolare, i modi di ricarica sono quattro (denotati con 1, 2, 3 e 4), i Casi di connessione sono tre (denotati con A, B e C) e i Tipi di connettori sono quattro (denotati con 1, 2, 3 e 4) [1].

Modi di ricarica

I quattro Modi di ricarica vengono definiti come segue:

Modo 1: si riferisce alla ricarica lenta da presa di corrente standard utilizzando connettori e cavi che non prevedono funzioni di controllo della ricarica o contatti ausiliari. I valori massimi di corrente e tensione che possono essere raggiunti sono 16 A e 250 V AC in monofase e 16 A e 480 V AC in trifase. L'EVSE relativa al Modo 1 deve prevedere, inoltre, una connessione di protezione verso terra;

Modo 2: si riferisce alla ricarica da presa di corrente standard utilizzando connettori e cavi che prevedono la funzione di control pilot e un sistema di protezione personale contro l'elettro-shock posizionato fra la presa standard e il veicolo elettrico. I valori massimi di corrente e tensione che possono essere raggiunti sono 32 A e 250 V AC in monofase e 32 A e 480 V AC in trifase. Anche in questo caso è prevista una connessione di protezione verso terra;

Modo 3: si riferisce alla ricarica da EVSE connessa in modo permanente alla rete AC che preveda le funzioni di control pilot e di protezione. I valori massimi di corrente e tensione che possono essere raggiunti sono i medesimi del Modo 2. Anche in questo caso è prevista una connessione di protezione verso terra;

Modo 4: si riferisce alla connessione di un EV ad una rete AC o DC utilizzando una EVSE in corrente DC che preveda le funzioni di control pilot e di protezione. Questo Modo di ricarica garantisce ricariche veloci con correnti di ricarica fino a 400 A [15].

Nello specifico, per quanto riguarda questo lavoro di tesi e tirocinio, le infrastrutture di ricarica che sono state prese in considerazione si riferiscono al Modo 3. In Figura 1.6, viene riportato l'interfacciamento fra una stazione di ricarica di Modo 3 e un veicolo elettrico. In

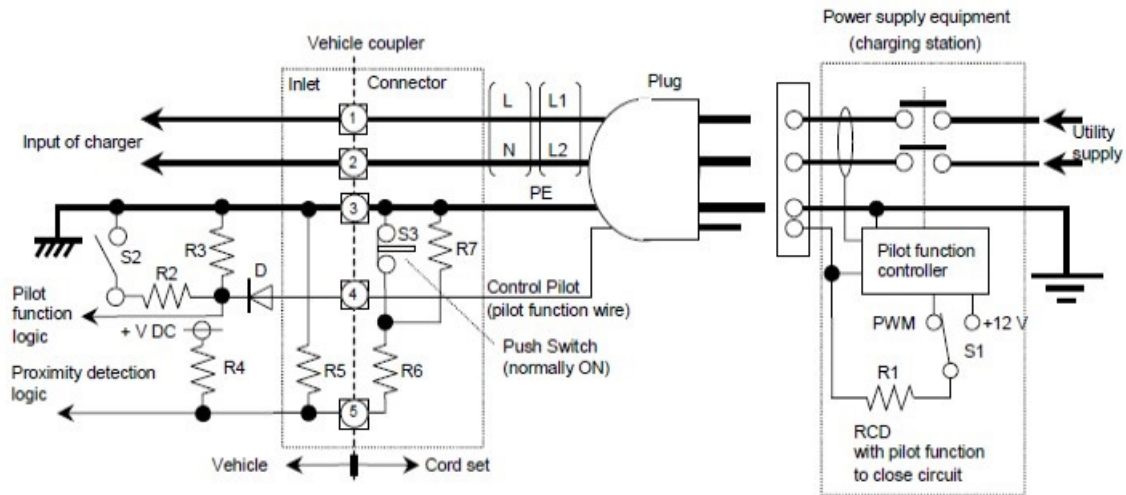
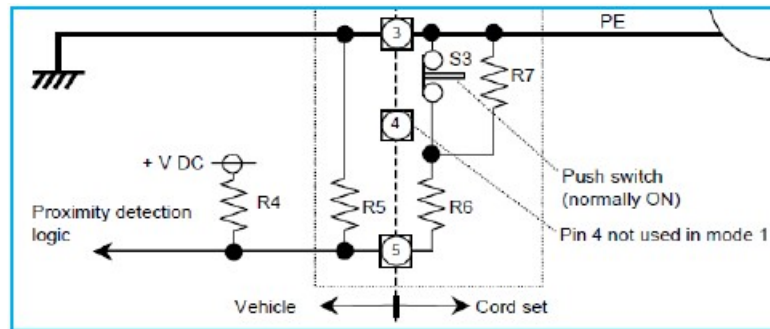


Figura 1.6: Interfacciamento fra EV ed EVSE per il Modo di ricarica 3 [15]

Figura 1.7: Circuito relativo alla *proximity function* [1]

questa Figura troviamo le due funzioni per il controllo della ricarica relative alla *proximity function* e al *control pilot*.

La parte di circuito relativa alla *proximity function*, viene riportata in Figura [1.7]. Tramite questo circuito viene definito lo stato di inserimento del connettore di ricarica nella presa del veicolo elettrico. Come è possibile notare, il circuito è composto da un partitore di tensione e lo stato della connessione viene identificato sulla base della tensione che viene letta al pin *Proximity detection logic*. Questa tensione assumerà valori differenti a seconda che lo switch S3 sia chiuso (cortocircuitando la resistenza R7) o aperto (costituendo una serie fra le resistenze R6 e R7). L'apertura e chiusura dello switch S3 viene gestita meccanicamente dalla connessione o meno del connettore di ricarica nella presa del veicolo. Nel caso in cui il cavo di ricarica non sia integrato nella stazione di ricarica, la *proximity function* viene disposta di una logica per l'identificazione della corrente massima di ricarica che il cavo può sostenere. Come viene riportato in Figura [1.8], questa logica dipende dal valore assunto dalla resistenza R_c .

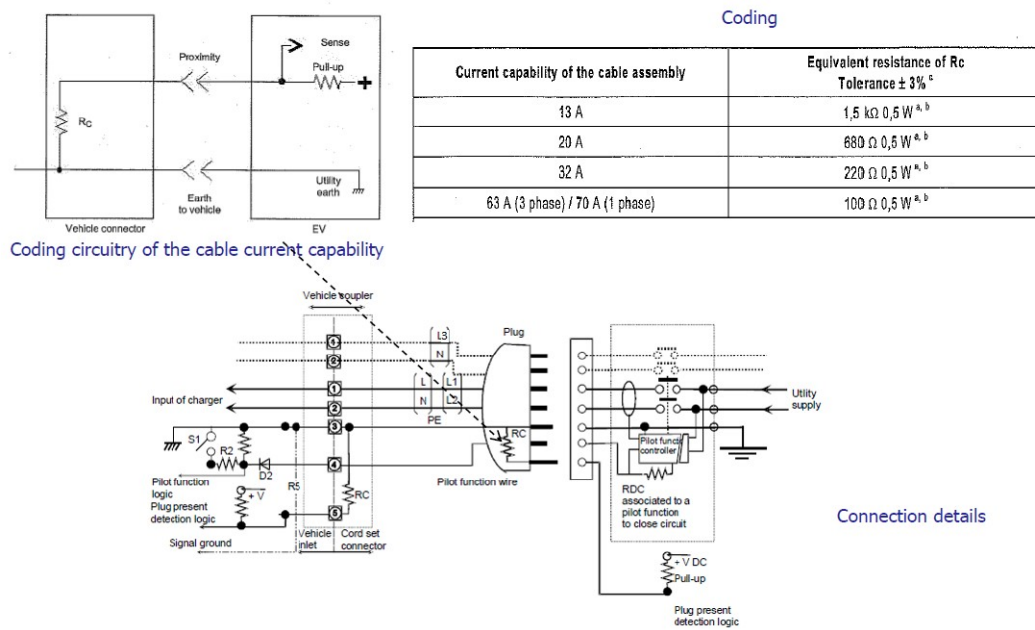


Figura 1.8: Circuito per la logica di decodifica della corrente massima sostenibile dal cavo di ricarica [1]

Per quanto riguarda la funzione di gestione della ricarica legata al *control pilot*, in Figura 1.9 viene riportata la comunicazione fra EV ed EVSE per un ciclo di ricarica. Come è possibile notare, il protocollo che permette di decodificare l'informazione legata alla ricarica si basa su una modulazione PWM, caratterizzata da una tensione di $\pm 12\text{ V}$ alla frequenza di 1 kHz . Il control pilot viene utilizzato per i Modi 2, 3 e 4 e, nel caso di ricarica di Modo 3, permette di comandare correnti che vanno da 6 A a 32 A variando il duty-cycle della modulazione PWM. Sulla base di questo protocollo, gli stati della ricarica vengono identificati come in Figura 1.10. Confrontando le Figure 1.9 e 1.11, è possibile comprendere la logica che sta dietro al protocollo di controllo della ricarica. Facendo riferimento alla Figura 1.9, una tensione di 12 V corrisponde allo stato di veicolo disconnesso (stati A1, A2 in Figura 1.10). Una volta che il connettore di ricarica viene collegato al veicolo, al terminale V_a di Figura 1.11 viene letta una tensione di 9 V (definita dal partitore di tensione fra le resistenze R_1 ed R_3) che determina lo stato di veicolo connesso (stato B1 in Figura 1.10). A questo punto, la stazione di ricarica può comunicare di essere disponibile all'avvio della ricarica imponendo un segnale PWM, il quale duty-cycle, rappresenta la massima corrente che è possibile erogare in quell'istante (stato B2 in Figura 1.10). Nel caso in cui il veicolo sia disponibile a ricevere energia dalla stazione di ricarica, lo switch S_2 verrà chiuso imponendo una tensione di 3 V/6 V al terminale V_a (la tensione dipende dal valore di resistenza R_2 che viene scelta sulla base che sia necessario comunicare la ventilazione dell'area di ricarica o meno) e l'effettivo avvio della ricarica corrispondente allo

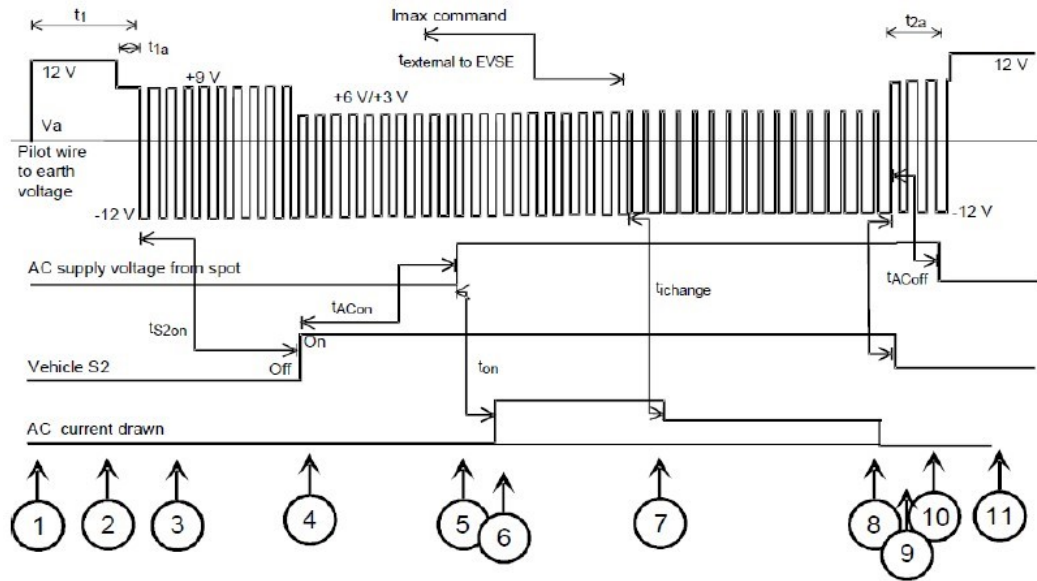


Figura 1.9: Comunicazione fra EV ed EVSE per il controllo della ricarica [15]

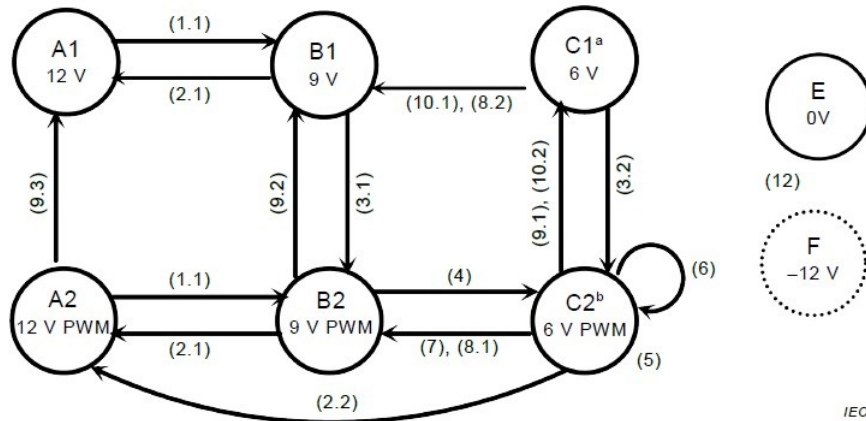


Figura 1.10: Macchina a stati della ricarica [15]

stato C2 in Figura 1.10. Ripercorrendo gli stati rimanenti in Figura 1.10, è possibile identificare le condizioni relative allo stop della ricarica causato da una indisponibilità della stazione di ricarica o del veicolo elettrico. Gli stati E ed F sono accessibili in qualsiasi istante della ricarica e permettono di segnalare un malfunzionamento dell'EVSE. Lo stato E segnala una condizione di errore che può essere legata, per esempio, ad una mancanza di disponibilità di potenza da parte dell'EVSE o ad un corto circuito fra il control pilot e il conduttore di protezione, mentre lo stato F viene intenzionalmente impostato dall'EVSE per segnalare uno stato di guasto per indicare, per esempio, che l'EVSE necessita di manutenzione [15].

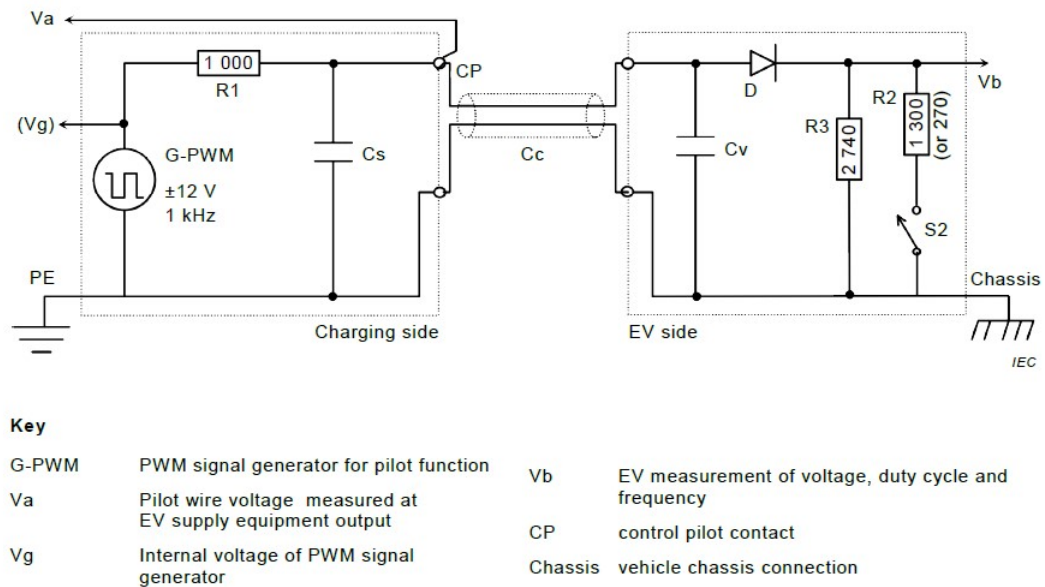


Figura 1.11: Circuito di control pilot tipico [15]

Casi di connessione

I tre Casi di connessione vengono definiti come segue:

Caso A: connessione dell'EV alla rete o ad una stazione di ricarica, utilizzando un cavo permanentemente connesso al veicolo (connessione tipica dei Modi 1 e 2);

Caso B: connessione dell'EV alla rete o ad una stazione di ricarica, utilizzando un cavo che può essere disconnesso ad entrambi i lati (connessione tipica del Modo 3);

Caso C: connessione dell'EV alla rete o ad una stazione di ricarica, utilizzando un cavo permanentemente connesso all'EVSE (connessione tipica dei Modi 3 e 4);

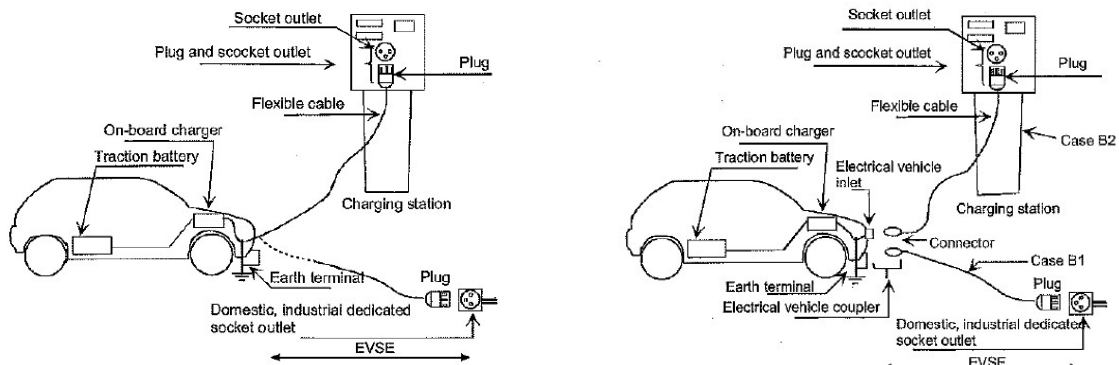
In Figura 1.12 vengono riportati i tre Casi di connessione.

Tipi di connettori

I quattro Tipi di connettori sono i seguenti:

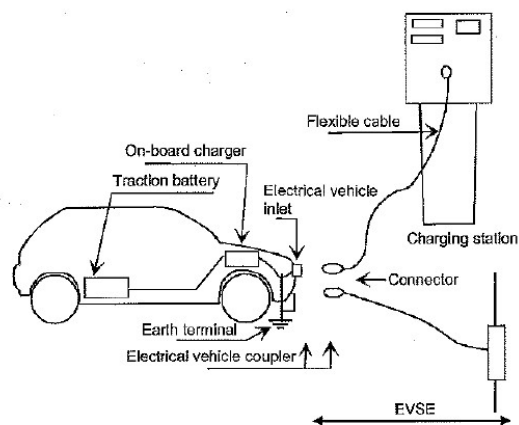
Tipo 1: utilizzato per la ricarica di sistemi monofase fino a 32 A, 250 V. E' provvisto di control pilot, proximity pilot e di collegamento di protezione verso terra;

Tipo 2: utilizzato per la ricarica di sistemi monofase fino a 70 A, 250 V o trifase fino a 63 A, 480 V. E' provvisto di control pilot, proximity pilot e di collegamento di protezione verso terra;



(a) Caso A

(b) Caso B

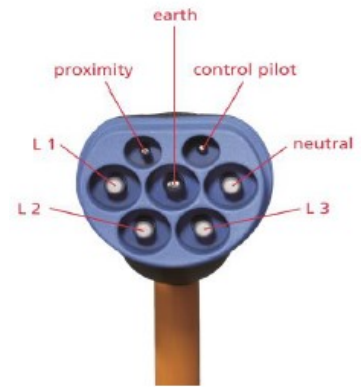


(c) Caso C

Figura 1.12: Casi di connessione [1]



(a) Tipo 1



(b) Tipo 2



Type 3A plug



Type 3C plug

(c) Tipo 3



(d) Tipo 4

Figura 1.13: Tipi di connettori [1]

Tipo 3: questa tipologia si divide in 3A e 3C. Il Tipo 3A viene utilizzato per la ricarica monofase fino a 16 A, 250 V ed è provvisto solo di control pilot e collegamento di protezione verso terra. Il Tipo 3C viene utilizzato per ricarica monofase e trifase fino a 63 A, 480 V ed è provvisto di control pilot, proximity pilot e di collegamento di protezione verso terra;

Tipo 4: questa tipologia viene utilizzata per la ricarica DC [1].

In Figura 1.13 vengono riportati i quattro Tipi di connettori.

Capitolo 2

Sistema di virtualizzazione delle colonnine di ricarica

In questo Capitolo, viene presentato il sistema che è stato sviluppato durante l'esperienza di tirocinio presso Silla Industries per generare la virtualizzazione di un cluster di colonnine di ricarica. Il risultato finale consiste in un simulatore che permette di emulare l'operato di un cluster di colonnine virtualmente interconnesse tramite un bus CAN e che garantisce all'utente la possibilità di interagire con ciascuna di esse grazie ad un'interfaccia grafica. Il tipo di colonnina di ricarica che viene simulato tramite questo sistema consiste in una colonnina AC di Modo 3 per la ricarica monofase e trifase. Le funzioni che vengono rese disponibili dal simulatore sono:

- La configurazione del numero di punti di ricarica disponibili e l'assegnazione degli indirizzi CAN e numeri seriali associati a ciascuno di essi;
- La simulazione degli stati di ricarica delle colonnine come stabiliti nella normativa IEC 61851 [15] e, inoltre, garantisce all'utente la possibilità di interagire con essi grazie ai relativi comandi;
- Consente di impostare la potenza disponibile fornita dalla rete, simulando la presenza di un energy meter SEM Three di Pick Data [5] connesso a monte del cluster;
- Fornisce una sezione dedicata alla gestione delle variabili di input/output delle simulazioni dei modelli matematici relative ai veicoli virtualmente connessi;
- Il processo di simulazione della ricarica può essere interamente controllato dall'operatore o può essere connesso, tramite un bus CAN, ad un dispositivo esterno che ricoprirà il ruolo di primary nella gestione del cluster di colonnine. Nello specifico, questo dispositivo sarà



Figura 2.1: Dispositivo Unità Gestore di Silla Industries [7]

in grado di gestire l'avvio e l'interruzione della ricarica delle colonnine e di modificare la corrente che ciascuna di esse potrà erogare.

Quest'ultima funzionalità, può essere abbinata a logiche di bilanciamento dei carichi al fine di evitare un assorbimento di potenza troppo elevato da parte del cluster. Per garantire questa funzionalità, i dati dell'energy meter simulato relativi alla potenza assorbita dalla rete possono essere trasferiti al primary grazie ad un bus RS-485 che implementa il protocollo ModBus RTU. L'utilizzo del simulatore combinato alla presenza di un primary, rappresenta una configurazione che garantisce una simulazione della ricarica semi-automatica dato che il controllo delle azioni relative al veicolo restano di competenza dell'utente. Nella procedura finale di test, il ruolo di primary per la gestione del cluster di colonnine, è stato ricoperto dal dispositivo Unità Gestore di Silla Industries [7]. Questo dispositivo consente di gestire fino a 50 colonnine di ricarica sia monofase che trifase. Fornisce inoltre:

- la gestione dei carichi, ripartendo tra i punti di ricarica collegati l'energia disponibile senza superare quella effettivamente disponibile;
- è dotato di una porta CAN-BUS per comunicare con le colonnine di ricarica presenti nel cluster;
- è dotato di una porta RS-485 per comunicare con un energy-meter esterno [9].

2.1 Struttura del simulatore

Al fine di rendere il sistema portatile e per agevolare la procedura di test, per questa applicazione sono stati utilizzati software multiplatforma e un design modulare. La modularità ha inoltre

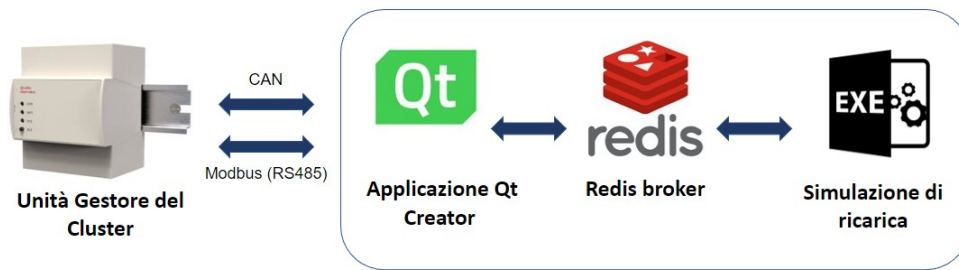


Figura 2.2: Struttura del simulatore e suo interfacciamento con l'Unità Gestore

permesso di utilizzare software dedicati per ogni ambito d'interesse dell'applicazione, in modo tale che questi potessero essere impiegati al meglio. La struttura del simulatore viene riportata in Figura 2.2 (in figura è anche possibile vedere il suo interfacciamento con l'Unità Gestore). Questa è composta da tre parti principali:

- Applicazione Qt Creator dedicata alla gestione delle seguenti funzionalità:
 - generazione dell'interfaccia del simulatore;
 - gestione delle logiche legate ai punti di ricarica del cluster;
 - controllo delle comunicazioni relative ai bus CAN e RS-485, che vengono gestiti tramite due porte COM seriali (vedi sezione 2.2.1);
 - gestione dell'energy meter simulato;
 - impostazione dei parametri di input delle simulazioni di ricarica dei veicoli elettrici virtualizzati e visualizzazione dei risultati di output.
- File eseguibili, ottenuti dagli schemi Simulink, per la simulazione di ricarica dei veicoli elettrici (dei quali si discuterà nei capitoli successivi);
- Broker dati Redis per il trasferimento dei parametri di input dei file eseguibili e per la memorizzazione dei risultati di simulazione.

2.2 Applicazione Qt Creator

Qt Creator è un IDE cross-platform dedicato allo sviluppo software. Questo prodotto permette di sviluppare applicazioni per sistemi desktop, mobile e embedded e garantisce la scrittura del codice nei linguaggi C++, QML, JavaScript, Python ed altri [4]. Per l'implementazione dell'interfaccia del simulatore, è stato scelto di creare un'applicazione desktop in codice C++.

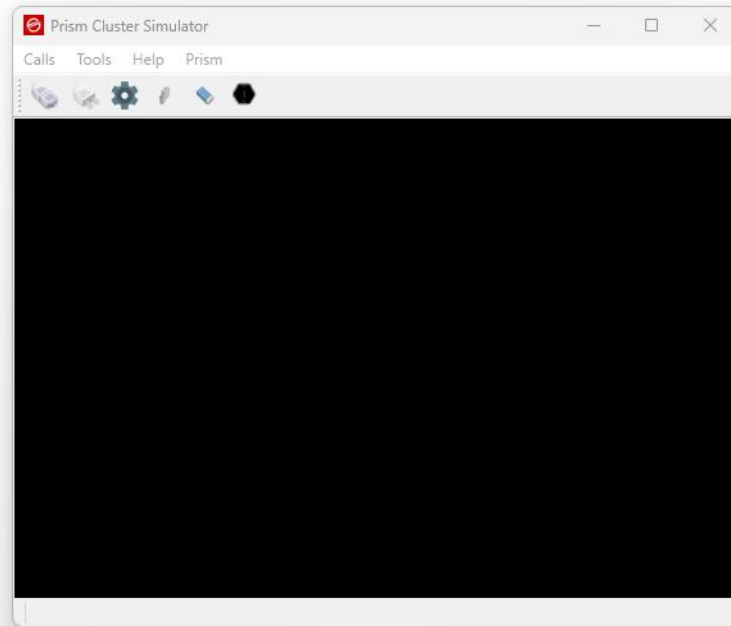


Figura 2.3: Interfaccia del simulatore

In Figura 2.3, viene riportata l'interfaccia del simulatore, dalla quale sarà possibile visualizzare i messaggi inviati e ricevuti relativi alle comunicazioni CAN e Modbus. In questa finestra è presente una barra degli strumenti dalla quale è possibile:

- configurare le comunicazioni seriali dei bus connessi (Figura 2.5);
- connettere e disconnettere le porte seriali tramite le icone riportate in Figura 2.6;
- accedere alla seconda finestra dedicata alle funzionalità principali del cluster (vedi Figura 2.4), come:
 - la selezione dell'indirizzo CAN della colonnina di ricarica con la quale si vuole interagire;
 - la visualizzazione delle informazioni relative ad eventuali comandi provenienti dal primary;
 - l'interazione con la macchina a stati della colonnina selezionata relativa allo standard IEC 61851 [15];
 - la visualizzazione delle informazioni relative all'energy meter simulato;

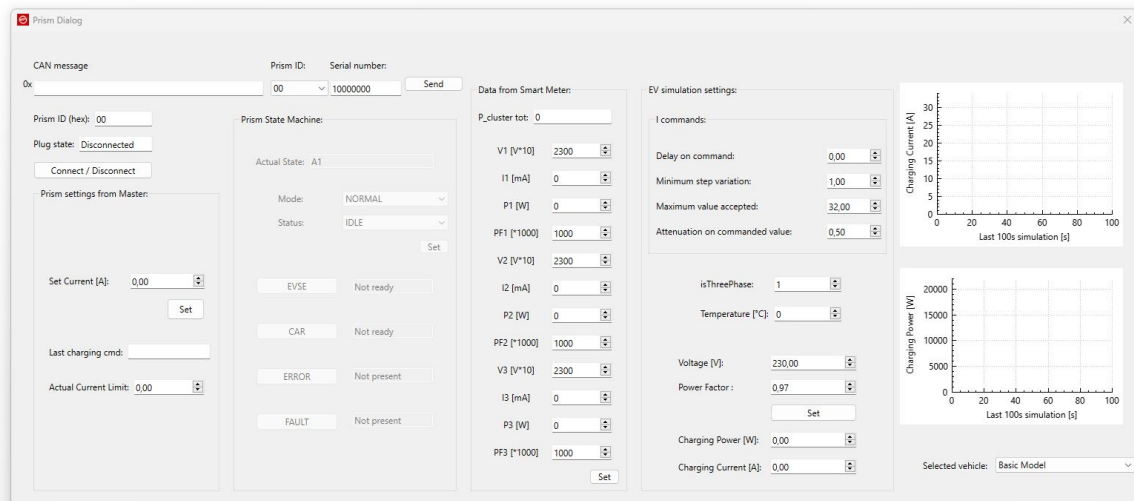


Figura 2.4: Interfaccia per la gestione del cluster

- l’impostazione dei parametri di input delle simulazioni di ricarica e la visualizzazione dei risultati di output relativi al modello di veicolo selezionato, sia a livello numerico che a livello grafico.

2.2.1 Gestione delle comunicazioni CAN e ModBus RTU

I due bus di comunicazione sono stati connessi al simulatore grazie all’utilizzo di due porte COM seriali. La gestione della comunicazione CAN è stata effettuata utilizzando la scheda ClusterBase di Silla che permette di effettuare la conversione CAN-Serial CAN mentre la conversione seriale-RS-485 è stata effettuata utilizzando il convertitore SH-U11G di DSD TECH [13]. La comunicazione CAN, è stata stabilita utilizzando il protocollo CAN di Silla basato sul formato extended frame (CAN 2.0B). La gestione del protocollo ModBus RTU su bus RS-485 è stata invece stabilita basandosi sulla struttura e sui comandi ModBus dell’energy meter SEM Three di Pick Data [5].

2.3 Broker dati Redis

Al fine di garantire la modularità del sistema, lo scambio di dati tra l’interfaccia del simulatore e i file eseguibili relativi agli schemi Simulink, è stato implementato sfruttando le funzionalità offerte da Redis, un sistema di base di dati in memoria centrale che può essere utilizzato come database, cache, broker di messaggi e come streaming engine [12]. Redis permette di interfacc-

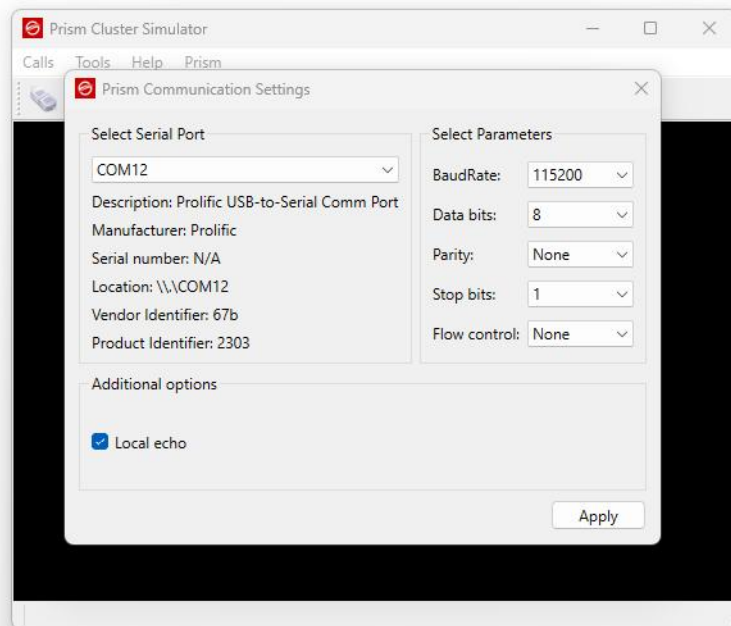


Figura 2.5: Una delle due finestre per la configurazione delle comunicazioni seriali

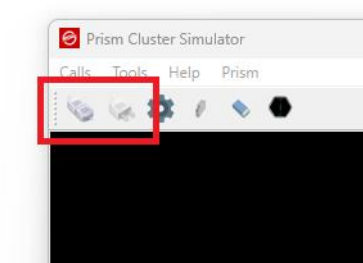


Figura 2.6: Icone di connessione e disconnessione delle porte seriali

ciarsi con il broker di dati tramite riga di comando, utilizzando l'interfaccia redis-cli, o tramite l'impiego di un vasto numero di linguaggi di programmazione. Nello specifico, per questa applicazione, l'interfacciamento è stato realizzato sfruttando la libreria hiredis [8], che offre le funzioni per la generazione di un client tramite un programma scritto in codice C.

Capitolo 3

Curve di ricarica degli EV

Come descritto precedentemente, dall'esperienza maturata da Silla Industries nella produzione di dispositivi per la ricarica in corrente alternata di auto elettriche, è sorta l'evidenza che il valore efficace della corrente assorbita dal veicolo durante il processo di ricarica non segue un andamento univoco basato sulla corrente imposta dalla colonnina ma subisce delle variazioni che sono differenti da un modello di EV all'altro. Queste differenze sono legate alle caratteristiche, al design e agli algoritmi di controllo dell'on-board-charger del quale è equipaggiato il veicolo preso in considerazione. Grazie ai test di ricarica AC effettuati su vari modelli di veicoli elettrici, è stato possibile apprezzare le differenze nell'andamento del valore efficace della corrente assorbita da quest'ultimi, differenze che sono state riscontrate principalmente:

- nel valore massimo di corrente assorbita, limite che viene imposto dalla potenza massima per la quale l'on-board-charger del veicolo è stato progettato;
- nella differenza tra la corrente efficace imposta dalla colonnina di ricarica e quella assorbita dal veicolo;
- nella minima variazione di corrente imposta per la quale si ha una variazione di corrente assorbita;
- e nella dinamica di risposta assunta dalla corrente efficace assorbita dal veicolo al variare della corrente imposta, in termini di tipologia di curva (lineare/esponenziale) e tempo di risposta.

Lo studio è iniziato considerando delle curve di corrente di ricarica, fornite da Silla, relative ai veicoli riportati in Tabella [3.1](#)

I test eseguiti su questi veicoli sono stati effettuati utilizzando un applicativo fornito da Silla per il controllo della colonnina. Questo applicativo permette di imporre alla colonnina una serie

Modello di veicolo	Tipo di ricarica
Opel Corsa-e	monofase
Tesla Model 3	monofase e trifase
Volvo XC40 Recharge	monofase
Volkswagen e-UP	monofase
Hyundai Ioniq 5 RWD Progress	trifase
Tesla Model S (2014)	monofase e trifase

Tabella 3.1: Modelli di veicoli presi in considerazione nella prima fase di raccolta dati

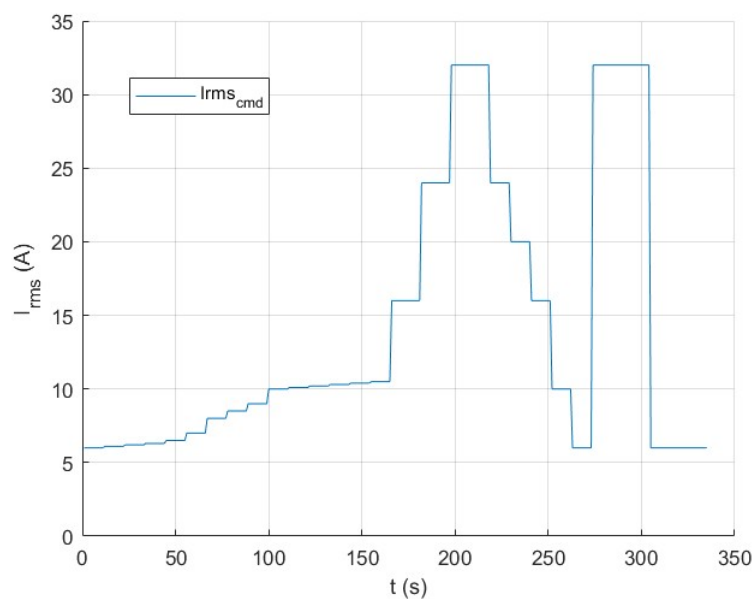


Figura 3.1: Caratteristica di corrente efficace imposta per i test di ricarica dei veicoli elettrici

di valori di corrente efficace di ricarica e, allo stesso tempo, permette di ottenere le informazioni relative alla comunicazione tra veicolo e colonnina stessa. Queste informazioni sono relative a corrente e tensione efficaci, potenza e power factor del veicolo. In Figura 3.2 vengono riportati i grafici relativi alle ricariche che sono state effettuate imponendo la stessa curva di corrente efficace di input, riportata in Figura 3.1, per tutti i modelli di veicoli sopra citati, impostando un periodo di campionamento di un secondo.

Come è possibile notare, ogni veicolo presenta una propria dinamica di risposta alla corrente imposta. Sulla base di questi dati raccolti, è stato eseguito un primo studio per la definizione delle caratteristiche principali di risposta per ogni modello di veicolo. Nelle sezioni che seguono, vengono presentate le analisi delle curve di corrente relative ai modelli Opel Corsa E, Ioniq 5 RWD Progress e Tesla Model S (2014). Si tenga presente che il simulatore che è stato

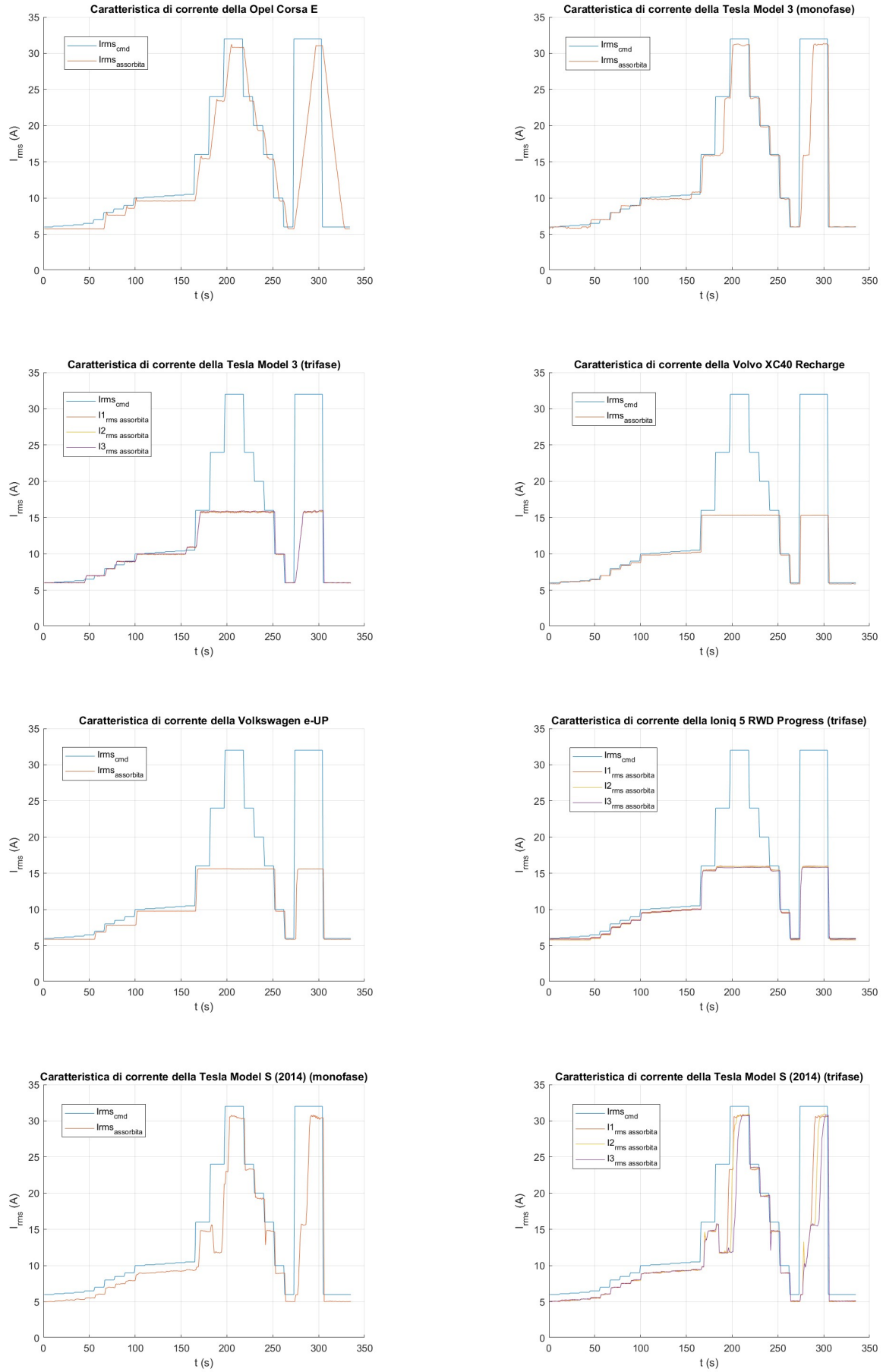


Figura 3.2: Curve di ricarica relative ai veicoli in Tabella 3.1

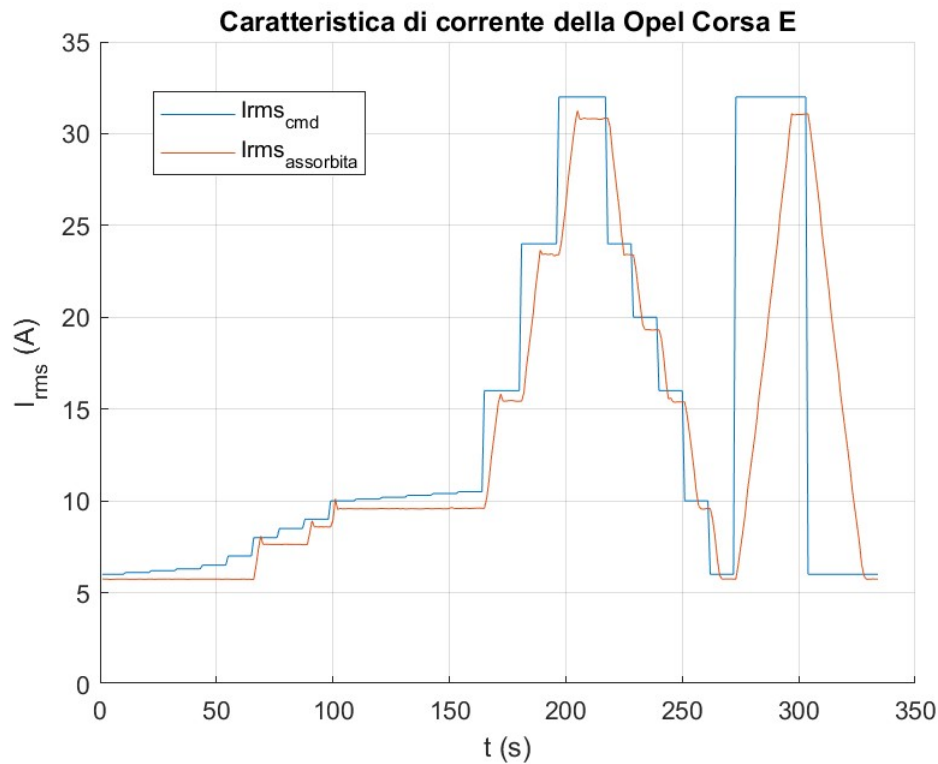


Figura 3.3: Caratteristica di corrente Opel Corsa-e

utilizzato per la procedura di test di ricarica, introduce un ritardo di un secondo nella procedura di acquisizione delle misure a seguito dell'assegnazione di un nuovo valore di corrente comandata. Per questo motivo, le curve riportate in Figura 3.2, presentano un ritardo di risposta di un secondo dovuto a questo errore sistematico.

3.1 Caratteristiche dinamiche Opel Corsa-e

Analizzando la curva di ricarica relativa alla Opel Corsa-e riportata in Figura 3.3, sono state riscontrate le seguenti caratteristiche:

- la risposta del veicolo alla ricarica è lineare con una pendenza della curva che va da un minimo di 0.65 A/s ad un massimo di 0.98 A/s e presenta dei picchi con ampiezza di circa 0.5 A al raggiungimento del valore di corrente efficace a regime;
- la corrente efficace assorbita dal veicolo risulta essere inferiore rispetto a quella comandata dalla colonnina di ricarica;
- il veicolo risponde a variazioni di corrente imposta superiori a 1 A, per $I_{rms_cmd} \leq 7 A$, e uguali ad 1 A per $I_{rms_cmd} > 7 A$.

Modello di veicolo	Opel Corsa-e
Dinamica di risposta	lineare con pendenza $m = 0.65 : 0.98 \text{ A/s}$
I_{delta}	$0.258 \text{ A (@ } I_{\text{rms}_{\text{cmd}}} = 6 \text{ A)}$ intorno a $0.64 \text{ A (@ } 6 \text{ A} < I_{\text{rms}_{\text{cmd}}} < 32 \text{ A)}$ $1.184 \text{ A (@ } I_{\text{rms}_{\text{cmd}}} = 32 \text{ A)}$
I_{step}	$>1 \text{ A (@ } I_{\text{rms}_{\text{cmd}}} \leq 7 \text{ A)}$ $1 \text{ A (@ } I_{\text{rms}_{\text{cmd}}} > 7 \text{ A)}$

Tabella 3.2: Caratteristiche della dinamica di risposta relative all'Opel Corsa-e

Nella Tabella 3.2 vengono riassunte le caratteristiche riscontrate per l'Opel Corsa-e, dove $I_{\text{delta}} = I_{\text{rms}_{\text{cmd}}} - I_{\text{rms}_{\text{assorbita}}}$ e $I_{\text{step}} = I_{\text{rms}_{\text{cmd}}}(t) - I_{\text{rms}_{\text{cmd}}}(t - 1)$ considerando t l'istante al quale sopraggiunge una variazione di $I_{\text{rms}_{\text{cmd}}}$.

3.2 Caratteristiche dinamiche Hyundai Ioniq 5 RWD Progress

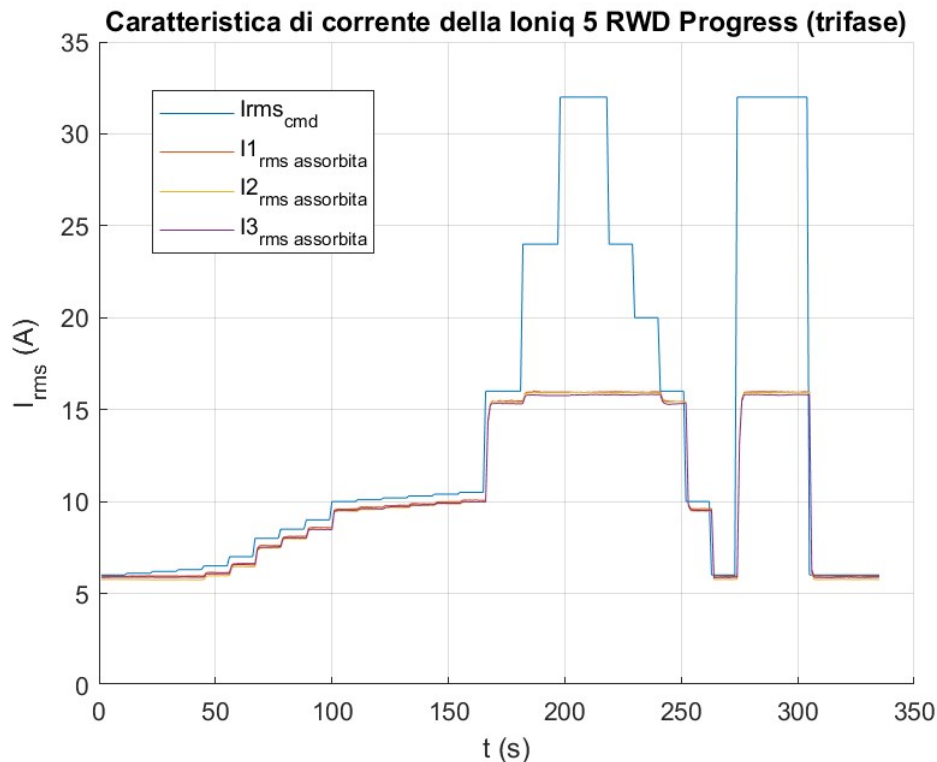


Figura 3.4: Caratteristica di corrente Hyundai Ioniq 5 RWD Progress

Modello di veicolo	Hyundai Ioniq 5 RWD Progress
Dinamica di risposta	esponenziale
I_{delta}	0.1 A (@ $I_{rms_{cmd}} \leq 6,3 A$) intorno a 0.51 A (@ $6,3 A < I_{rms_{cmd}} \leq 16 A$) 0.2 A (@ $I_{rms_{cmd}} > 16 A$)
I_{step}	il veicolo risponde alle variazioni di $I_{rms_{cmd}}$ maggiori di 6,3 A
$I_{rms_{assorbita, max}}$	$16 A - I_{delta}$

Tabella 3.3: Caratteristiche della dinamica di risposta relative alla Hyundai Ioniq 5 RWD Progress

Analizzando la curva di ricarica relativa al veicolo Hyundai Ioniq 5 RWD Progress riportata in Figura 3.4, sono state riscontrate le seguenti caratteristiche:

- la risposta del veicolo alla ricarica è di tipo esponenziale;
- la corrente efficace assorbita dal veicolo risulta essere inferiore rispetto a quella comandata dalla colonnina di ricarica;
- il veicolo risponde a tutte le variazioni di corrente imposta superiori ai 6,3 A;
- caricando il veicolo in trifase, la corrente massima assorbita è $I_{rms_{assorbita}} = 16 A - I_{delta}$ a causa del limite di potenza di 11 kW imposto dall'on-board-charger.

Nella Tabella 3.3 vengono riassunte le caratteristiche riscontrate per la Hyundai Ioniq 5 RWD Progress, dove $I_{delta} = I_{rms_{cmd}} - I_{rms_{assorbita}}$ e $I_{step} = I_{rms_{cmd}}(t) - I_{rms_{cmd}}(t-1)$ considerando t l'istante al quale sopraggiunge una variazione di $I_{rms_{cmd}}$.

3.3 Caratteristiche dinamiche Tesla Model S (2014)

Analizzando la curva di ricarica relativa alla Tesla Model S (2014) riportata in Figura 3.5, sono state riscontrate le seguenti caratteristiche:

- al fine di caratterizzare la risposta del veicolo alla ricarica, la dinamica è stata suddivisa in 5 tipologie di risposta:

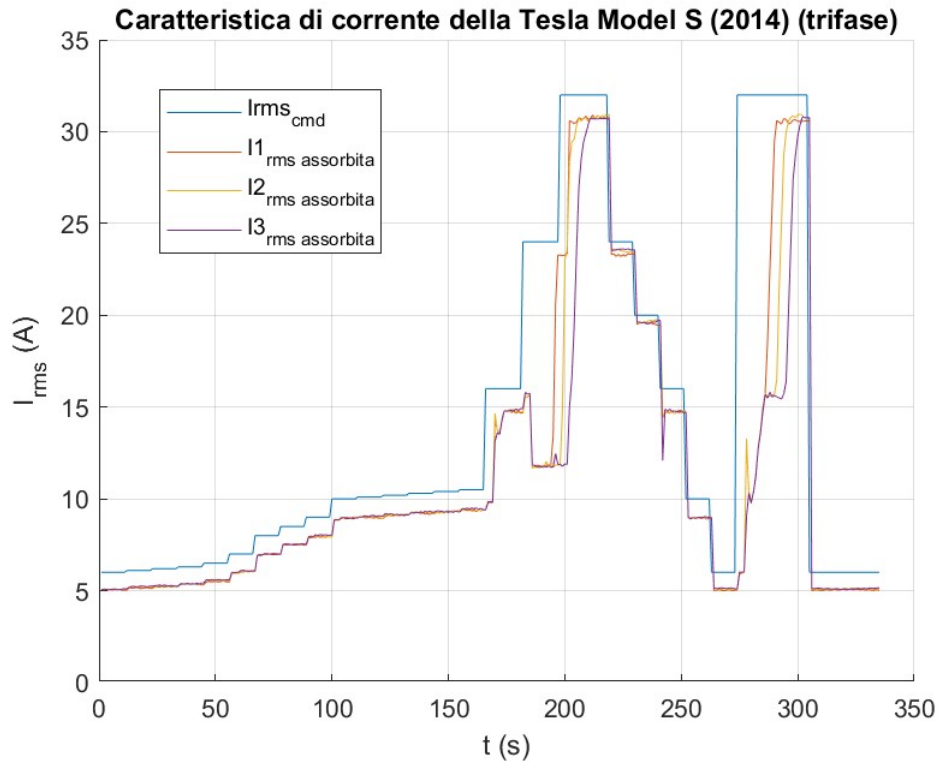


Figura 3.5: Caratteristica di corrente Tesla Model S (2014)

- Tipo 0: per variazioni di $I_{rms_cmd} \leq 1 A$;
 - Tipo 1: per variazioni di $I_{rms_cmd} > 1 A$ con $I_{rms_cmd} \leq 16 A$;
 - Tipo 2: per variazioni di $I_{rms_cmd} > 1 A$ con I_{rms_cmd} che inizia la sua variazione da 16 A;
 - Tipo 3: per condizioni di Tipo 2 nel caso sopraggiunga un nuovo comando di corrente e almeno una delle tre fasi non sia a regime del comando di corrente imposto precedentemente;
 - Tipo 4: per variazioni di $I_{rms_cmd} > 1 A$ con I_{rms_cmd} che esegue una variazione da un valore di $I_{rms_cmd} < 16 A$ ad un valore $I_{rms_cmd} > 16 A$;
 - Tipo 5: per variazioni negative di I_{rms_cmd} (fronte di discesa).
- la corrente efficace assorbita dal veicolo risulta essere inferiore rispetto a quella comandata dalla colonnina di ricarica;
 - il veicolo risponde a tutte le variazioni di corrente imposta.

Come è possibile notare dalla Figura [3.2](#), per la ricarica in monofase la caratteristica assunta dal valore efficace della corrente assorbita dal veicolo segue l'andamento della seconda fase.

Modello di veicolo	Tesla Model S (2014)
Dinamica di risposta	lineare/esponenziale
I_{delta}	0.9 A (@ fronte di salita) 0.4 A (@ $I_{rms_{cmd}} > 16$ A fronte di discesa) 1.13 A (@ $I_{rms_{cmd}} \leq 16$ A fronte di discesa)
I_{step}	il veicolo risponde a tutte le variazioni di $I_{rms_{cmd}}$

Tabella 3.4: Caratteristiche della dinamica di risposta relative alla Tesla Model S (2014)

Nella Tabella 3.4 vengono riassunte le caratteristiche riscontrate per la Tesla Model S (2014), dove $I_{delta} = I_{rms_{cmd}} - I_{rms_{assorbita}}$ e $I_{step} = I_{rms_{cmd}}(t) - I_{rms_{cmd}}(t - 1)$, considerando t l'istante al quale sopraggiunge una variazione di $I_{rms_{cmd}}$.

3.3.1 Tipi di dinamica

Come è possibile notare dalla Figura 3.5, la dinamica di risposta alla ricarica della Tesla Model S (2014) risulta essere molto complessa e poco prevedibile. Di seguito vengono riportate le caratteristiche relative alle tipologie di risposta che sono state possibili osservare sulla base dei dati inizialmente disponibili. Per semplicità, gli slew rates (in termini di variazione di corrente per unità di tempo [A/s]) delle risposte di tipo lineare, sono stati approssimati a valori costanti calcolati sulla media delle pendenze corrispondenti alla tipologia e zona di curva presa in considerazione.

Tipo 0 la risposta è di tipo lineare con slew rate m_1 che dipende dall'ampiezza della variazione di corrente comandata;

Tipo 1 la risposta presenta il comportamento seguente:

- aumenta con andamento lineare e slew rate m_A ;
- si assesta per 2 secondi;
- riprende ad aumentare riportando un picco di corrente una volta che viene superata la metà della transizione;
- a seguito del picco, aumenta con andamento lineare e slew rate m_B fino a raggiungere il valore a regime.

Tipo 2 la risposta presenta il comportamento seguente:

- aumenta con andamento lineare e slew rate m_A ;
- si assesta per 2 secondi al valore 15.7 A;
- diminuisce con andamento lineare e slew rate m_C ;
- si assesta per 8 secondi al valore 11.8 A;
- aumenta fino al raggiungimento del valore a regime con andamento lineare e slew rate m_2 . E' inoltre presente un ritardo di 4 secondi tra la prima fase e la seconda e di 7 secondi tra la prima e la terza.

Tipo 3 per circostanze di Tipo 2, nel caso sopraggiunga un nuovo comando prima che le fasi raggiungano il valore a regime, viene introdotto un ritardo di risposta di 4 secondi;

Tipo 4 la risposta presenta il comportamento seguente:

- aumenta con andamento lineare e slew rate m_A ;
- si assesta per 2 secondi;
- riprende ad aumentare riportando un picco di corrente a circa metà della transizione;
- a seguito del picco, assume un andamento lineare con slew rate m_B fino a raggiungere i 15.7 A;
- una volta raggiunto questo valore, la prima fase prosegue il suo andamento fino a raggiungere il valore a regime con andamento lineare e slew rate m_3 , mentre le rimanenti raggiungono il valore a regime con andamento esponenziale e presentando un ritardo di 4 secondi tra la prima e la seconda e di 7 secondi tra la prima e la terza fase.

Tipo 5 la risposta è di tipo lineare con slew rate m_4 che dipende dall'ampiezza della variazione di corrente comandata;

Nella Tabella [3.5](#) vengono riportate le approssimazioni degli slew rates relativi alle tipologie sopra riportate.

3.4 Raccolta dati con curve di ricarica alternative

Come risulta evidente dall'analisi relativa alla Tesla Model S (2014), in alcuni casi non è possibile caratterizzare la risposta del veicolo basandosi solamente sulle informazioni che vengono

Tesla Model S (2014)	Slew rates
Tipo 0	$m_1 = I_{rms_{cmd}}(t) - I_{rms_{cmd}}(t - 1) $ (considerando t l'istante al quale sopraggiunge una variazione di $I_{rms_{cmd}}$)
Tipo 1	$m_A = 0.68 \text{ A/s}$ $m_B = 1.8 \text{ A/s}$
Tipo 2	$m_A = 0.68 \text{ A/s}$ $m_C = -3.91 \text{ A/s}$ $m_2 = 5.32 \text{ A/s}$
Tipo 4	$m_A = 0.68 \text{ A/s}$ $m_B = 1.8 \text{ A/s}$ $m_3 = 1.75 \text{ A/s}$
Tipo 5	$m_4 = - I_{rms_{cmd}}(t) - I_{rms_{cmd}}(t - 1) $ (considerando t l'istante al quale sopraggiunge una variazione di $I_{rms_{cmd}}$)

Tabella 3.5: Slew rates Tesla Model S (2014)

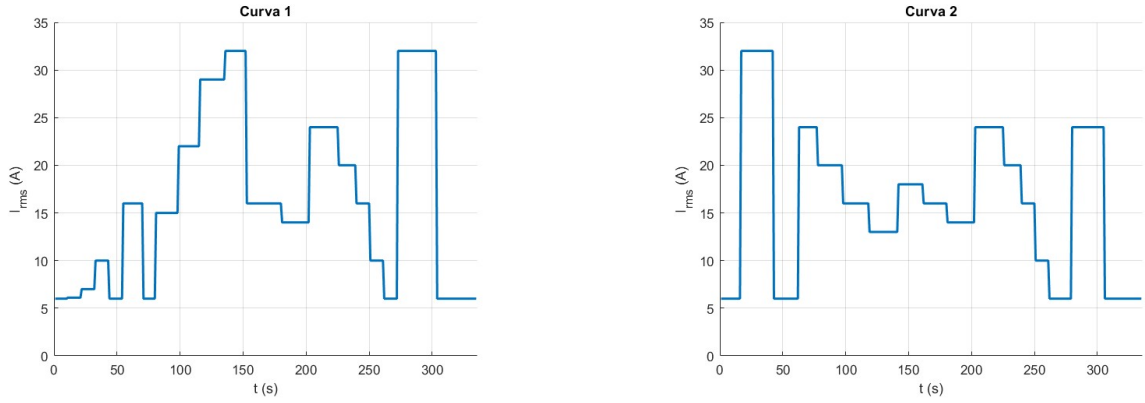


Figura 3.6: Set di curve di ricarica alternative

fornite dalla caratteristica in Figura 3.5. Per questo motivo, al fine di determinare una caratterizzazione il più accurata possibile della dinamica di risposta alla ricarica dei veicoli elettrici presi in considerazione, questi sono stati sottoposti a test di ricarica basati su un secondo set di curve di corrente. Questi profili di corrente, che vengono riportati in Figura 3.6, sono pensati per analizzare determinate circostanze che non sono state rese evidenti dallo studio precedente. In particolare, rispetto al profilo originale riportato in Figura 3.1, questi permettono di analizzare in modo più accurato la ricarica attorno al valore $I_{rms_cmd} = 16\text{ A}$ (che risulta essere critico specialmente per la Tesla Model S) e di testare più circostanze di variazione della corrente I_{rms_cmd} nell'intervallo superiore ai 16 A.

I veicoli che sono risultati disponibili in azienda per il test di questo secondo set di curve di ricarica sono stati la Hyundai Ioniq 5 RWD Progress e la Tesla Model S (2014). Sulla base di questi test, l'analisi iniziale è stata rivista al fine di identificare possibili cambiamenti nella determinazione della dinamica di risposta assunta dai due veicoli.

3.4.1 Hyundai Ioniq 5 RWD Progress: curve di ricarica alternative

La Hyundai Ioniq 5 RWD Progress non presenta diversità nella caratterizzazione della dinamica di risposta rispetto a quanto già determinato nella sezione 3.2. In Figura 3.7 vengono riportate le caratteristiche di corrente ottenute tramite questo secondo set di curve di ricarica.

3.4.2 Tesla Model S (2014): curve di ricarica alternative

Per quanto riguarda i test di ricarica con il veicolo Tesla Model S (2014), sono state notate delle dinamiche differenti rispetto a quanto già determinato nella sezione 3.3. In Figura 3.8 vengono

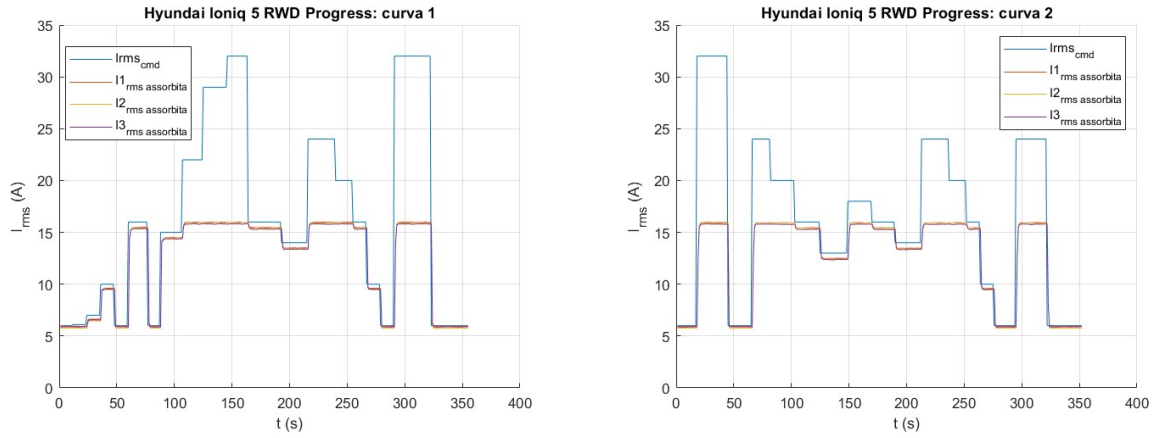


Figura 3.7: Hyundai Ioniq 5 RWD Progress: curve di ricarica alternative

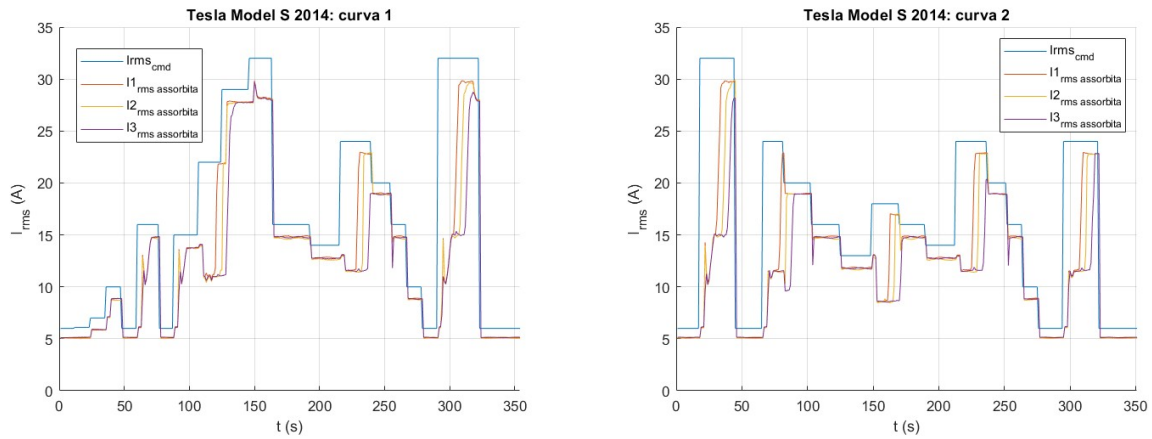


Figura 3.8: Tesla Model S (2014): curve di ricarica alternative

riportate le caratteristiche di corrente ottenute tramite questo secondo set di curve di ricarica.

Le differenze che sono state riscontrate sono le seguenti:

- la dinamica di Tipo 2 si verifica per profili di corrente che iniziano la loro dinamica nell'intervallo di corrente efficace $13 A \leq I_{rms_cmd} \leq 16 A$. Inoltre, il limite inferiore di corrente efficace raggiunto con questa tipologia di dinamica non è facilmente prevedibile;
- nella dinamica di Tipo 4 il valore di corrente efficace al quale si verifica un assestamento di 7 secondi, non è facilmente prevedibile. Sulla base dei test effettuati, una buona approssimazione corrisponde ai 47/100 del valore di corrente efficace I_{rms_cmd} comandato.

Sulla base di quanto è stato riscontrato riguardo alla risposta alla ricarica di questo modello di veicolo, per ricavare una caratterizzazione più accurata della dinamica relativa alla corrente

efficace che viene assunta, sarebbe necessario eseguire dei test più approfonditi.

Capitolo 4

Modellizzazione matematica degli EV

Sulla base dell'analisi sulla raccolta dati relativa alle curve di ricarica dei veicoli elettrici riportata nel Capitolo [3](#), è possibile notare che alcune caratteristiche relative alle differenze tra corrente efficace imposta dalla colonnina e corrente efficace assorbita dal veicolo si ripresentano su più modelli di veicoli. A tal proposito, il processo di modellizzazione matematica si basa inizialmente sulla rappresentazione della risposta alla ricarica di un veicolo generico che possa rappresentare le caratteristiche comuni che sono state riscontrate. Lo scopo di questo primo modello è quello di dare la possibilità all'utente del simulatore di ricarica di modificare i parametri di input del modello al fine di ricreare le situazioni di ricarica che desidera sulla base delle funzionalità rese disponibili. Oltre a questa prima modellizzazione matematica, ne viene presentata una seconda tipologia che vede la riproduzione delle dinamiche di risposta alla ricarica basate su un modello specifico di veicolo. Per quanto riguarda questa seconda serie di modelli, come descritto nella [sezione 3.4](#), la seconda raccolta dati, basata su un set di curve alternative, è stata effettuata solo sui modelli Ioniq 5 RWD Progress e Tesla Model S (2014). Data la maggiore presenza di dati di ricarica, per questa seconda serie di modelli matematici, lo studio si focalizza sulla modellizzazione dei due modelli di veicoli elettrici sopra citati. Le simulazioni di ricarica sono state realizzate tramite l'uso di schemi Simulink [\[14\]](#). Di seguito vengono riportate le tipologie di modelli discusse precedentemente.

4.1 Modello matematico per veicolo generico

Come descritto precedentemente, il primo modello matematico che è stato sviluppato prevede la simulazione delle caratteristiche comuni che sono state riscontrate confrontando le risposte dinamiche dei veicoli elettrici presi in considerazione. Tra queste caratteristiche troviamo:

- il valore massimo di corrente assorbita dal veicolo;
- la differenza tra la corrente efficace imposta dalla colonnina di ricarica e quella assorbita dal veicolo;
- la minima variazione di corrente imposta per la quale si ottiene una variazione di corrente assorbita.

Come è possibile notare, questo tipo di modello non tiene in considerazione la dinamica di corrente assunta dal veicolo al variare della corrente imposta dalla colonnina di ricarica (una variazione di corrente in ingresso corrisponde ad una variazione a gradino dell'uscita). In Figura 4.1 viene riportato lo schema Simulink relativo al modello generico di veicolo elettrico. In Tabella 4.1 troviamo le informazioni relative ai parametri di input/output del sistema. Lo schema è composto da tre blocchi subsystem:

Current command actions permette di ottenere la corrente efficace assorbita dal veicolo andando a modificare quella di ingresso in funzione dei parametri di input del blocco e del modello matematico contenuto al suo interno;

Power calculation sulla base della corrente efficace assorbita dal veicolo calcolata al blocco precedente, determina la potenza attiva che quest'ultimo sta assorbendo;

Energy calculation esegue l'integrale della potenza attiva per ottenere l'energia totale assorbita.

4.1.1 Sottosistema *Current command actions*

In Figura 4.2 viene riportato il sottosistema relativo al subsystem *Current command actions*. Tramite questo blocco è possibile impostare: il limite inferiore di corrente efficace di ricarica (impostato a 6 A secondo lo standard IEC 61851), la quantizzazione della corrente comandata (relativa al parametro I_{step}), il limite superiore della corrente efficace che il veicolo può assorbire (nel caso non sia impostata tramite il parametro I_{maxCar} , questa viene impostata a 32 A secondo lo standard IEC 61851 [15]), la differenza nel valore di corrente efficace assorbita e comandata (relativa al parametro I_{delta}) e un eventuale ritardo di risposta del veicolo. All'Appendice A vengono riportati i listati relativi ai blocchi delle funzioni MatLab presenti nel subsystem in questione.

Variabili di input
$I_{set} = I_{rms_{cmd}}$ $I_{set_{old}} = I_{rms_{cmd}}(t - 1)$ $I_{step} = I_{rms_{cmd}}(t) - I_{rms_{cmd}}(t - 1)$ $t_{delay} = \text{delay di risposta}$ $I_{maxCar} = \text{corrente rms massima che il veicolo può assorbire}$ $I_{delta} = I_{rms_{cmd}} - I_{rms_{assorbita}}$ $V_{grid} = \text{tensione efficace di rete}$ $isThreephase = \text{variabile booleana che indica se la ricarica sta}$ $\text{avvenendo in trifase (1) o in monofase (0)}$
Variabili di output
$I_{charging} = \text{corrente rms assorbita dal veicolo}$ $P = \text{potenza attiva assorbita dal veicolo}$ $energySession = \text{energia assorbita durante la ricarica}$

Tabella 4.1: Parametri di input/output dello schema Simulink

In Figura 4.3 viene riportata la simulazione del blocco *Current command actions*, per la quale i parametri di input sono stati impostati come in Tabella 4.2. Per questa simulazione, il profilo di corrente efficace comandata corrisponde a quello riportato in Figura 3.1.

Parametri di input
$I_{set_{old}} = 0 \text{ A}$
$I_{step} = 1 \text{ A}$
$t_{delay} = 1 \text{ s}$
$I_{maxCar} = 16 \text{ A}$
$I_{delta} = 0.53 \text{ A}$

Tabella 4.2: Parametri di input per la simulazione del blocco *Current command actions*

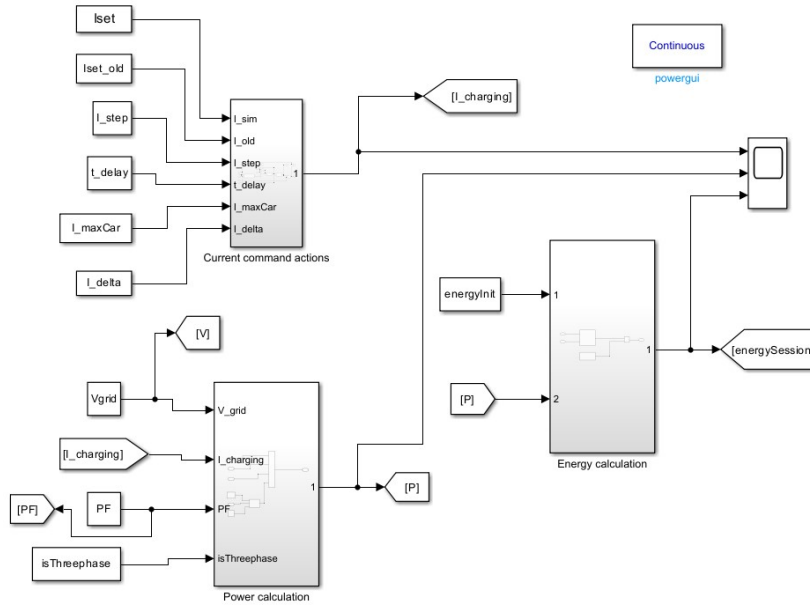
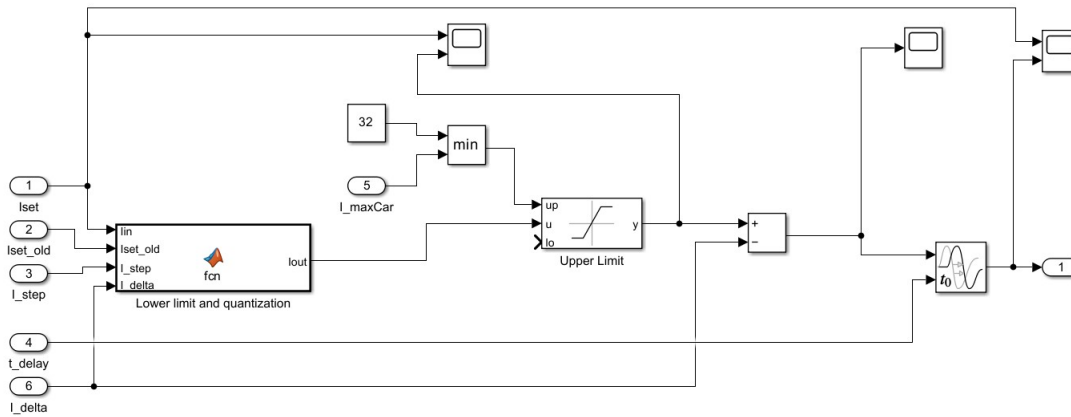


Figura 4.1: Schema Simulink relativo al modello di veicolo generico

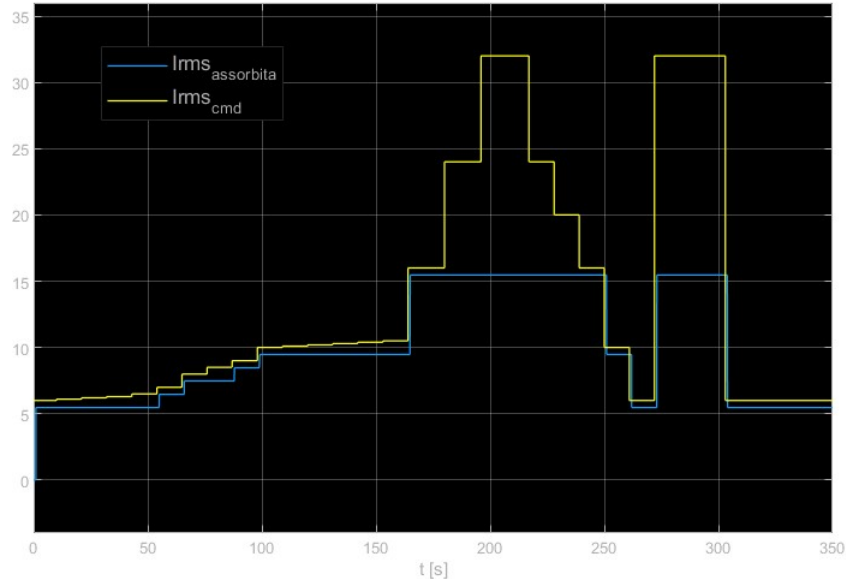

 Figura 4.2: Blocco *Current command actions*

4.1.2 Sottosistema *Power calculation*

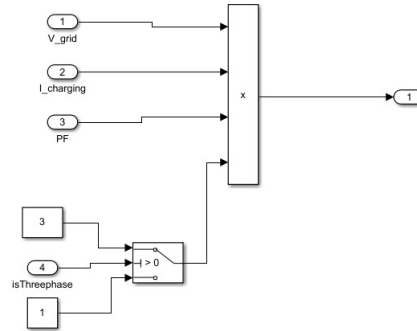
In Figura 4.4 viene riportato il sottosistema relativo al subsystem *Power calculation*. Tramite questo blocco è possibile calcolare la potenza attiva assorbita dal veicolo secondo la relazione:

$$P = V_{rms} * I_{rms} * PF \quad (4.1)$$

Nel calcolo della potenza assorbita dal veicolo, è possibile simulare la presenza di potenza reattiva tramite l'input PF (power factor). In questo modello di veicolo si assume che i valori di corrente efficace assorbita e di tensione efficace di rete siano uguali per le tre fasi. Per questo motivo, nel caso di ricarica trifase, la potenza attiva verrà calcolata secondo la relazione:

Figura 4.3: Simulazione del blocco *Current command actions*

$$P = 3 * V_{rms} * I_{rms} * PF \quad (4.2)$$

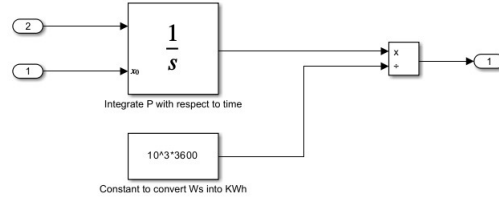
Figura 4.4: Blocco *Power calculation*

4.1.3 Sottosistema *Energy calculation*

In Figura 4.5 viene riportato il sottosistema relativo al subsystem *Energy calculation*. Tramite questo blocco è possibile calcolare l'energia (in kWh) assorbita dal veicolo secondo la relazione:

$$E(t) = \int P(t) dt \quad (4.3)$$

Per eseguire questo calcolo, è stato utilizzato un blocco integratore. L'output di quest'ultimo è stato diviso per il coefficiente $k = 10^3 * 3600$ al fine di ottenere la misura dell'energia in kWh.

Figura 4.5: Blocco *Energy calculation*

4.2 Modello matematico Hyundai Ioniq 5 RWD Progress

Facendo riferimento all'analisi eseguita nella sezione 3.2, viene riportata la modellizzazione matematica della risposta alla ricarica del veicolo Hyundai Ioniq 5 RWD Progress. Sulla base dello schema Simulink riportato in Figura 4.1, le modifiche che sono state apportate per rappresentare la risposta alla ricarica di questo veicolo, sono contenute nel sottosistema *Current command actions* (vedi Figura 4.6). Dato che per questo modello di veicolo, il parametro I_{delta} varia in funzione della corrente I_{rms_cmd} , è stata introdotta una funzione MatLab (*Difference I_delta* in Figura 4.6) per la gestione delle condizioni riportate nella Tabella 3.3. Al fine di riprodurre la risposta esponenziale del veicolo ad una variazione della corrente di ricarica imposta, è stato eseguito un fit della risposta al gradino utilizzando l'applicazione Curve Fitter di MatLab [18]. Il fit è stato effettuato basandosi sull'equazione della risposta al gradino di un sistema del primo ordine (vedi Figura 4.7):

$$y(t) = 1 - e^{-\frac{t}{\tau}} \quad (4.4)$$

Una volta ottenuto il parametro $\tau = \frac{1}{1.78} s$ dal fit, è stato inserito nello schema Simulink il blocco di funzione di trasferimento:

$$H(s) = \frac{1}{1 + s \cdot \tau} \quad (4.5)$$

Dato il confidence bound del 95 % ($\tau = \frac{1}{1.398} : \frac{1}{2.159} s$), il parametro del fit è stato approssimato a $\tau = \frac{1}{2.159} s$ che si è dimostrato meglio rappresentare la risposta analizzata nella Figura 3.4. Al fine di riprodurre caratteristiche di corrente che partono da un valore diverso da zero, è stato scelto un blocco di funzione di trasferimento che garantisse l'impostazione dell'output iniziale. All'Appendice A vengono riportati i listati relativi ai blocchi delle funzioni MatLab presenti nel subsystem in Figura 4.6.

Grazie alla prevedibilità della dinamica di ricarica di questo modello di veicolo, è stato possibile creare un modello matematico che la rappresenti in modo accurato. In Figura 4.8

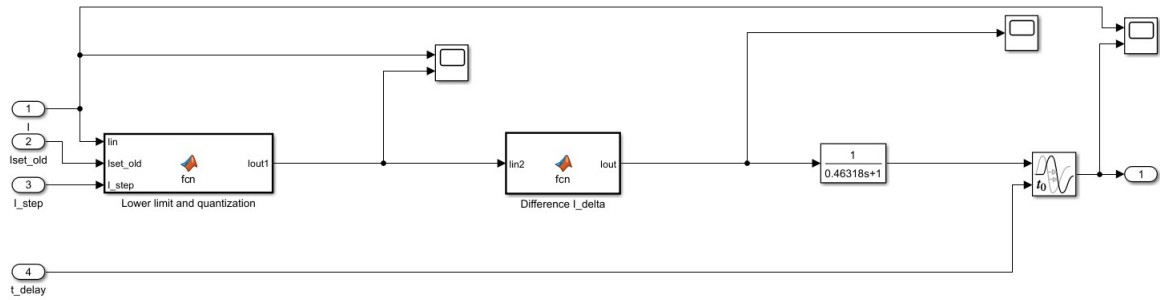


Figura 4.6: Blocco *Current command actions* per il veicolo Hyundai Ioniq 5 RWD Progress

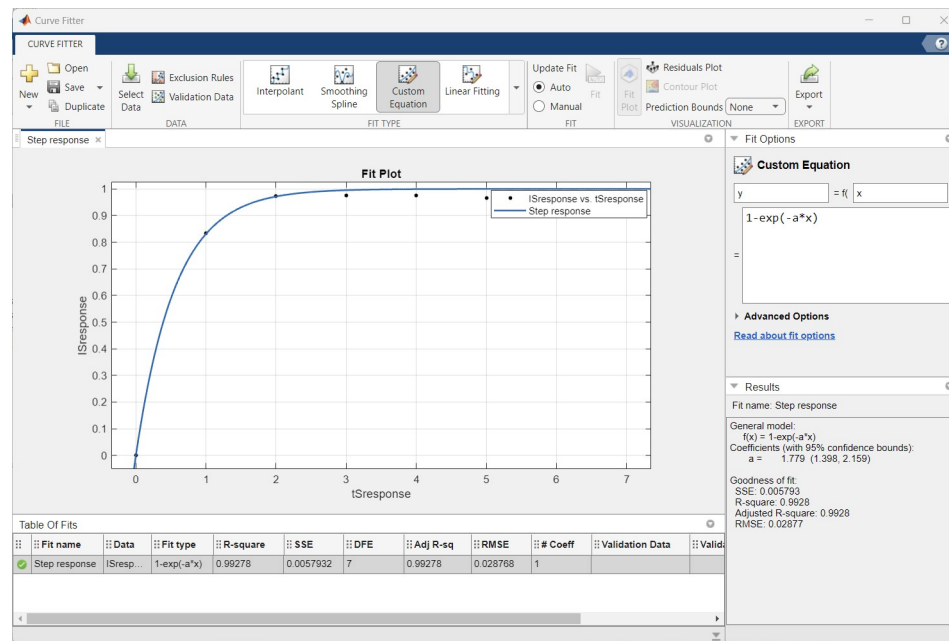


Figura 4.7: Fit della risposta al gradino per il veicolo Hyundai Ioniq 5 RWD Progress

vengono riportate le simulazioni dello schema Simulink relative al veicolo Hyundai Ioniq 5 RWD Progress, nelle quali sono state imposte come input le caratteristiche di corrente efficace riportate nelle Figure 3.1 e 3.6. Come è possibile notare confrontando i risultati di simulazione ottenuti con i dati di simulazione di ricarica reali riportati nelle Figure 3.4 e 3.7, il modello realizzato per questo veicolo riesce a riprodurre in modo accurato la dinamica assunta dal valore efficace della corrente assorbita durante la ricarica. L'unico limite di questo modello è stato riscontrato sui fronti di discesa delle caratteristiche, che risultano essere più repentini nel caso reale. Come nel caso del modello matematico per veicolo generico della sezione 4.1, la potenza è stata calcolata approssimando che il valore efficace della corrente e della tensione di rete siano uguali per le tre fasi. Gli schemi Simulink utilizzati per il calcolo della potenza e dell'energia assorbite dal veicolo sono i medesimi utilizzati per il modello di veicolo generico riportati nelle Figure 4.4 e 4.5.

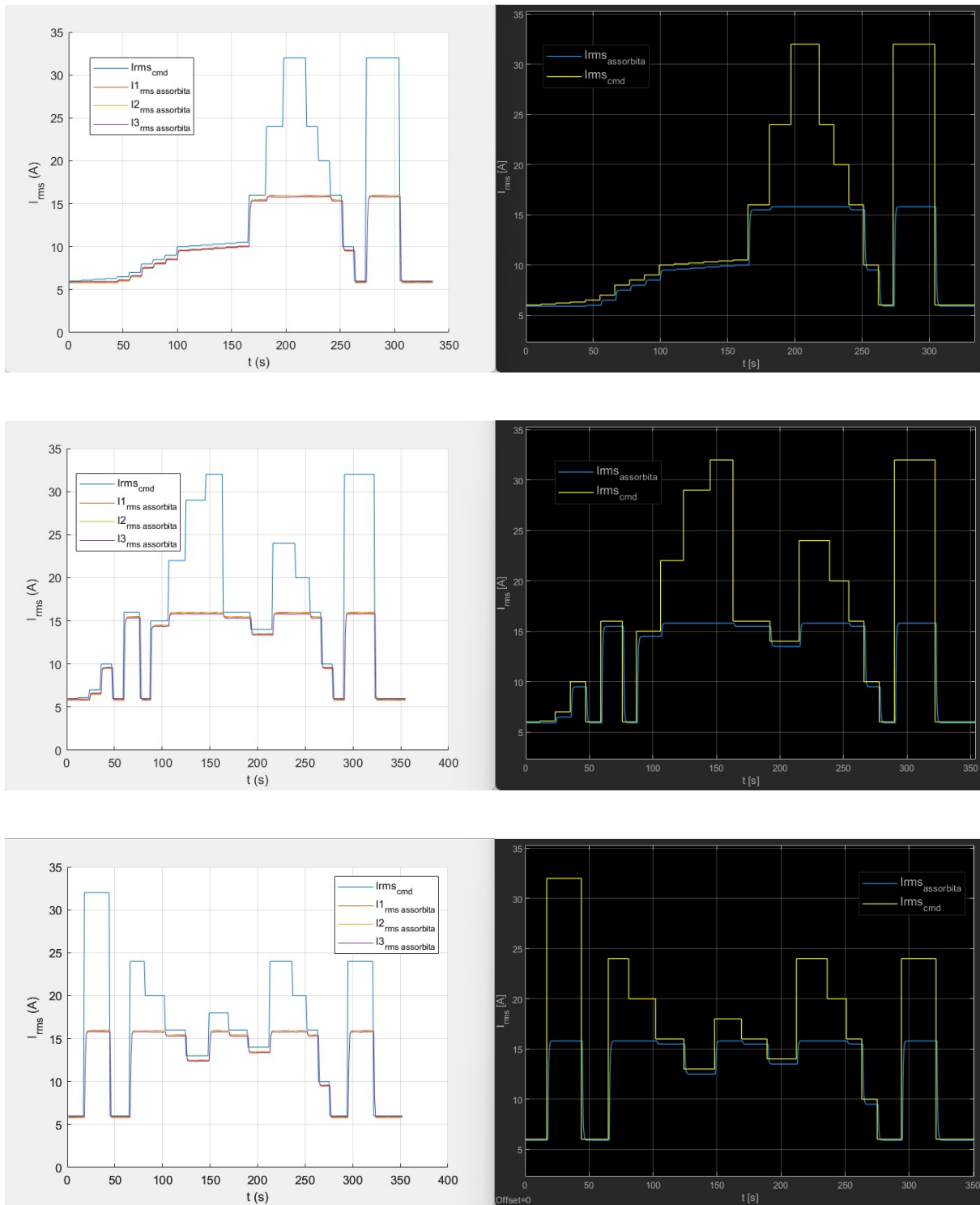


Figura 4.8: Confronto fra le curve di ricarica reali e le simulazioni dello schema Simulink relative al veicolo Hyundai Ioniq 5 RWD Progress

4.3 Modello matematico Tesla Model S (2014)

Facendo riferimento all'analisi eseguita al Capitolo 3 e sulla base dei dati raccolti, viene riportata la modellizzazione matematica della risposta alla ricarica del veicolo Tesla Model S (2014). Data la poca prevedibilità relativa alla dinamica assunta dalla corrente efficace durante la ricarica e data la disponibilità limitata di dati, il modello matematico che è stato creato non garantisce una rappresentazione esatta della risposta alla ricarica ma cerca di darne una riproduzione il più accurata possibile. Sono state inoltre introdotte alcune approssimazioni per quanto riguarda:

- il picco delle tipologie 1 e 4 (vedi sezione 3.3) che è stato trascurato;
- l'andamento esponenziale della dinamica di Tipo 4 è stato approssimato con un andamento lineare.

Rispetto allo schema Simulink riportato in Figura 4.1, in questo caso il valore di corrente efficace assunto durante la ricarica in trifase, non è uguale per le tre fasi, ogni fase assume una dinamica differente. Per questo motivo, lo schema di partenza è stato modificato in modo da poterle gestire separatamente. Le modifiche che sono state fatte, e che vengono riportate nelle Figure 4.9, 4.10 e 4.11, sono relative al livello più superiore dello schema e ai subsystems *Current command actions* e *Power calculation*.

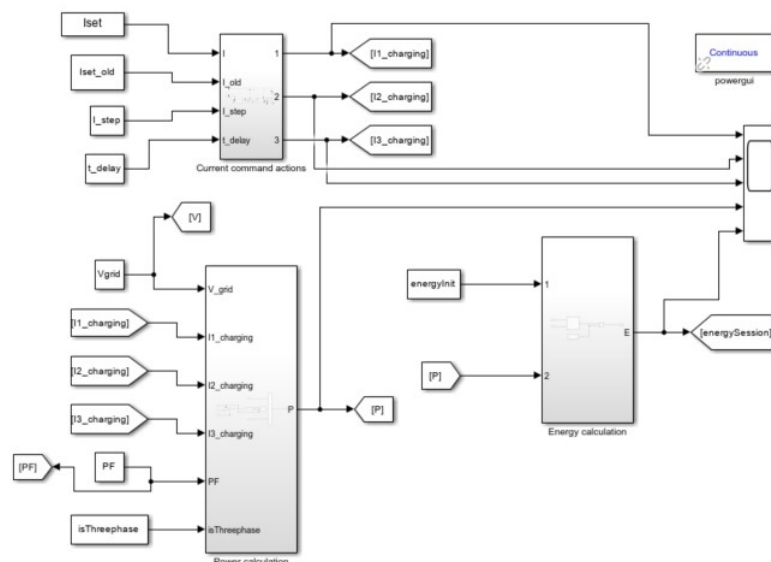


Figura 4.9: Schema Simulink relativo al veicolo Tesla Model S (2014)

Per quanto riguarda la rappresentazione della dinamica di risposta alla ricarica di questo veicolo, le modifiche che sono state apportate rispetto allo schema riportato in Figura 4.2, sono

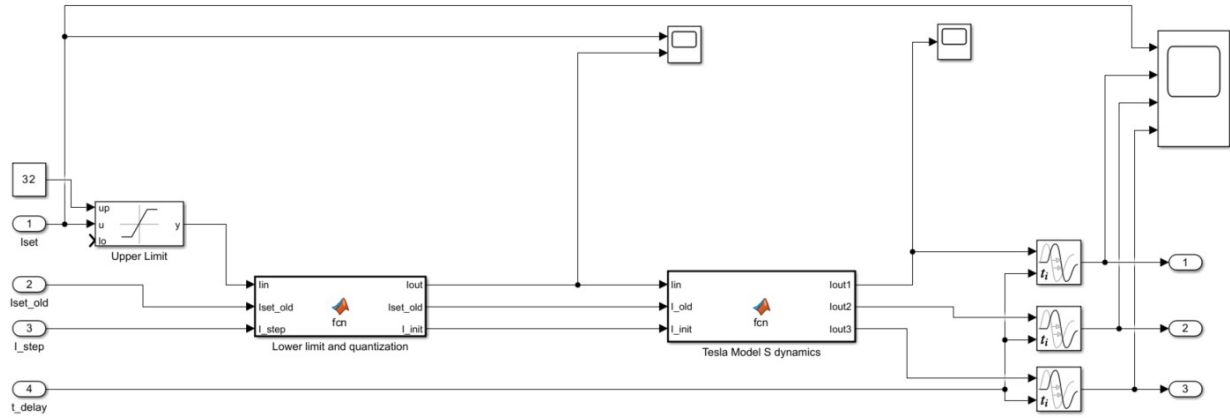


Figura 4.10: Blocco *Current command actions* per il veicolo Tesla Model S (2014)

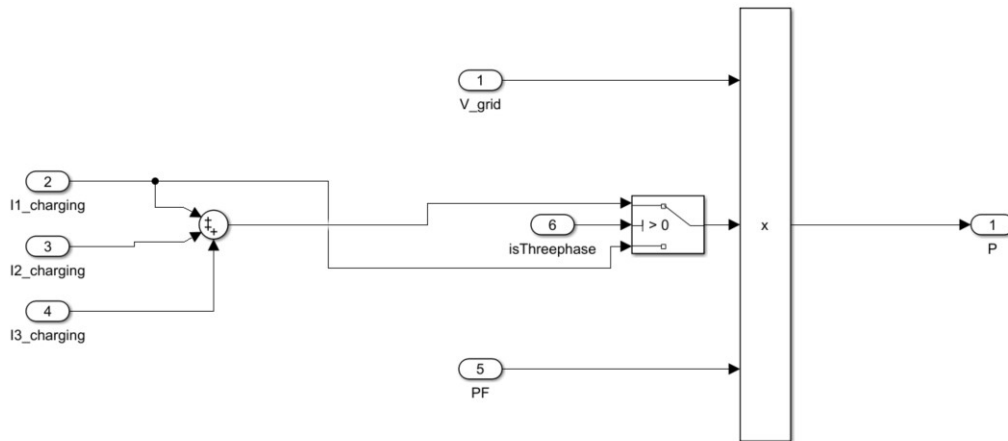


Figura 4.11: Blocco *Power calculation* per il veicolo Tesla Model S (2014)

contenute nel sottosistema *Current command actions* (vedi Figura 4.10) e, più nello specifico, sono rappresentate dalla funzione MatLab *Tesla Model S dynamics*. Il codice MatLab relativo a questa funzione viene riportato nell'Appendice A. Come è possibile notare da questo codice, la logica si basa su una prima identificazione del tipo di dinamica che deve essere rappresentata relativa all'analisi condotta alla sezione 3.3, tramite un costrutto *if/else*, per poi proseguire con una struttura *switch case* dedicata alla rappresentazione della dinamica in questione. Nell'ultimo costrutto *if/else*, invece, viene effettuato un controllo sulle correnti di output della funzione al fine di determinare se il valore a regime è stato raggiunto o meno.

In Figura 4.12 vengono riportate le simulazioni dello schema Simulink relativo al veicolo Tesla Model S (2014), nelle quali sono state imposte come input le caratteristiche di corrente efficace riportate nelle Figure 3.1 e 3.6. Confrontando i risultati di simulazione ottenuti con i dati di simulazione di ricarica reali riportati nelle Figure 3.5 e 3.8, è possibile notare che, come

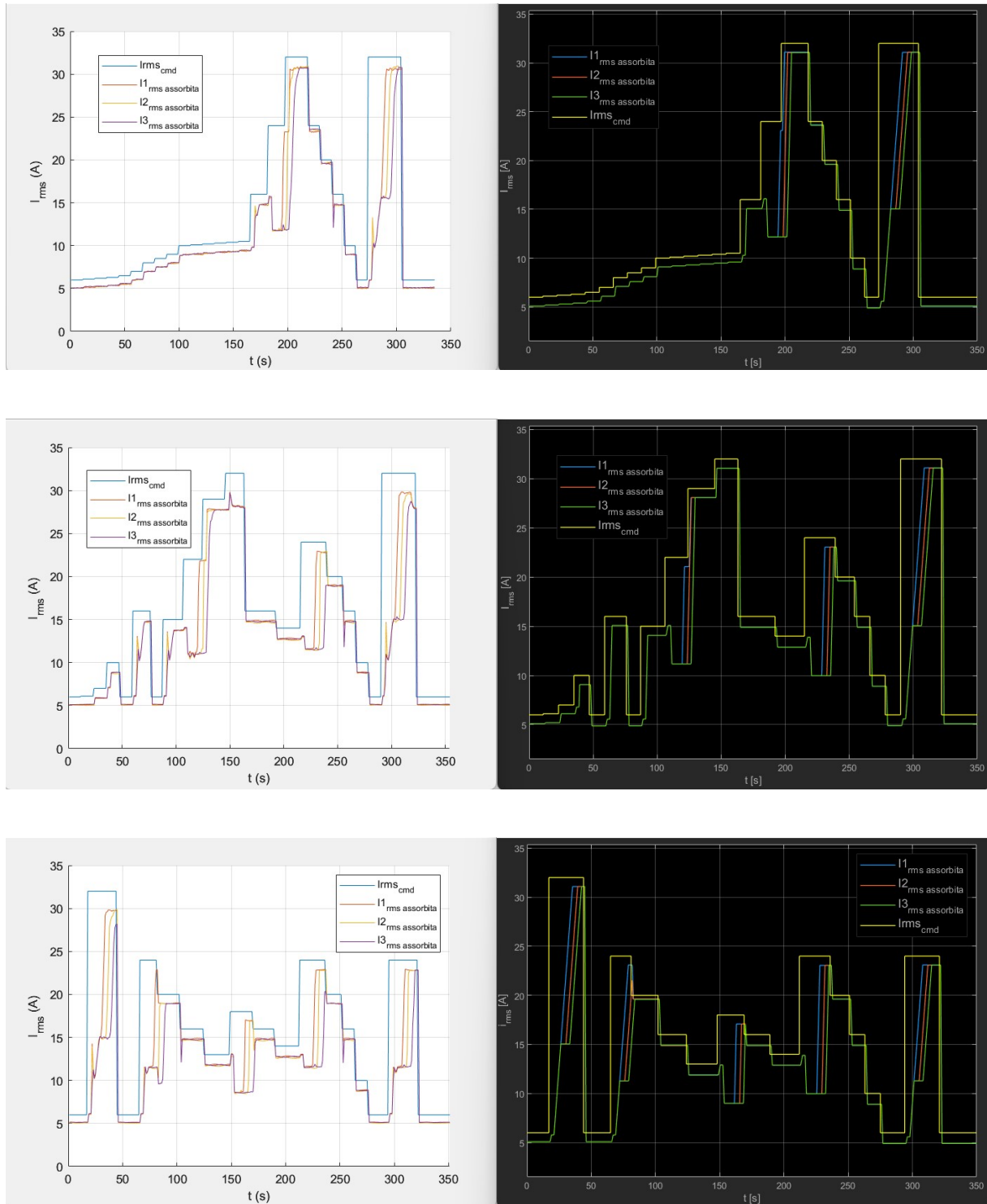


Figura 4.12: Confronto fra le curve di ricarica reali e le simulazioni dello schema Simulink relative al veicolo Tesla Model S (2014)

descritto precedentemente nella sezione [3.4.2](#), sono presenti alcune limitazioni nella accuratezza della riproduzione della dinamica che assume la corrente efficace di questo veicolo durante la ricarica. Nonostante queste limitazioni, però, il modello realizzato garantisce una buona riproduzione di quelle che sono le criticità legate alla ricarica di questo modello di veicolo, soprattutto per quanto riguarda i profili di corrente legati alle dinamiche di Tipo 2 e 4 discussi nella sezione [3.3](#) che sono uniche di questo veicolo. Come nel caso del modello matematico per veicolo generico della sezione [4.1](#), la potenza assorbita dal veicolo è stata calcolata approssimando che il valore efficace della corrente e della tensione di rete siano uguali per le tre fasi. Lo schema Simulink utilizzato per il calcolo dell'energia assorbita dal veicolo è il medesimo utilizzato per i modelli di veicoli presentati precedentemente.

Capitolo 5

Integrazione degli schemi Simulink nel sistema di virtualizzazione di colonnine di ricarica

In questa sezione verrà discusso il metodo che è stato utilizzato per integrare le simulazioni di ricarica dei veicoli elettrici, nel simulatore presentato al Capitolo [2](#). Tramite questa integrazione, è stato possibile definire dal simulatore, quale modello di veicolo elettrico collegare a ciascuna delle colonnine presenti nel cluster del sistema simulato al fine di rappresentarne la dinamica di ricarica.

5.1 Gestione delle simulazioni di ricarica

A livello concettuale, la gestione delle simulazioni di ricarica dei veicoli elettrici presenti nel cluster, viene effettuata eseguendo una simulazione dello schema Simulink relativo al modello di veicolo selezionato, ogni volta che nel simulatore viene rilevato un nuovo comando di corrente imposto ad una delle colonnine di ricarica simulate. Per eseguire le simulazioni degli schemi Simulink dal simulatore, sono stati generati dei file eseguibili per ognuno dei modelli di veicoli che sono stati modellizzati (vedi sezione [5.2](#)). Ad ogni esecuzione di un file eseguibile, corrisponde una simulazione di ricarica di 100 secondi. Il tempo di ogni simulazione è stato imposto uguale per ogni modello di veicolo ed è stato scelto basandosi sulla durata della dinamica del veicolo a risposta più lenta. Il veicolo in questione, è la Tesla Model S (2014) che, come è possibile notare dalla Figura [3.5](#), riporta una dinamica di risposta di circa 30 secondi nel caso della dinamica di Tipo 2 riportata in Tabella [3.5](#). Ogni volta che viene ricevuto un nuovo

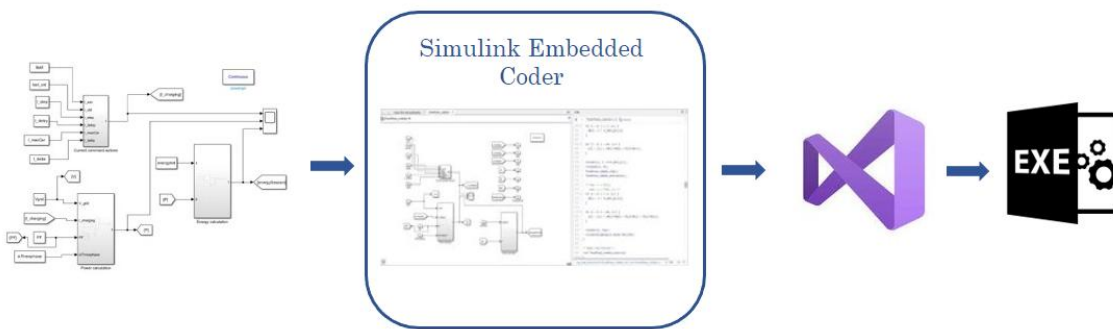


Figura 5.1: Processo di generazione del codice dagli schemi Simulink e dei file eseguibili

comando di corrente, viene assegnata alla colonnina corrispondente una variabile che effettua un salvataggio dell'istante al quale è stato lanciato l'eseguibile. I dati di simulazione ottenuti, salvati in delle strutture array, verranno fatti scorrere al passare del tempo dalla logica interna del simulatore, in modo tale da comunicare all'utente, ad ogni istante, i dati relativi allo stato di ricarica attuale. Dato che ad ogni nuovo comando di corrente, lo stato della ricarica risulta essere differente dal comando precedente, il simulatore esegue un salvataggio del contesto di ricarica alla fine di ogni simulazione per ogni colonnina, in modo tale che questo possa essere assegnato come input all'esecuzione successiva. Nel caso in cui entro 100 secondi non venga ricevuto un nuovo comando di corrente, ovvero nel caso in cui gli array siano stati fatti scorrere completamente, verrà tenuto come valido l'ultimo dato di simulazione disponibile. E' stato possibile adottare questa logica dato che, per ogni modello di veicolo, è sicuro che dopo 100 secondi la dinamica di risposta del veicolo ad un comando di corrente abbia raggiunto il valore a regime.

5.2 Generazione del codice degli schemi Simulink e dei file eseguibili

Come descritto precedentemente, per poter integrare gli schemi Simulink dei modelli matematici degli EV nel simulatore, è stato necessario ottenere da quest'ultimi dei file eseguibili. Questo per garantire un tempo di simulazione degli schemi che fosse il più breve possibile in modo tale da non introdurre dei ritardi nel processo di ricarica dei veicoli simulati.

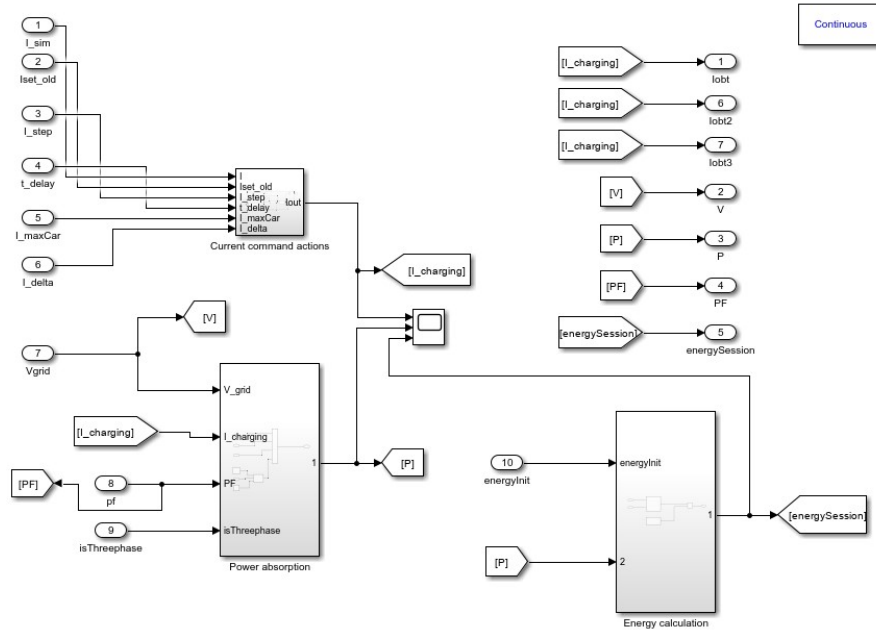


Figura 5.2: Modello di veicolo generico adattato alla generazione del codice con Embedded Coder [3]

5.2.1 Generazione del codice con Embedded Coder

Per ottenere i file eseguibili, prima di tutto è stato generato il codice C dei modelli degli EV utilizzando l'applicazione Embedded Coder di Simulink [3]. Per poter utilizzare questo tool, è stato necessario adattare gli schemi Simulink alla generazione del codice. In Figura 5.2 viene riportato come esempio, lo schema convertito relativo al modello di veicolo generico.

Il processo di adattamento prevede, innanzitutto, che i parametri di ingresso/uscita dello schema vengano sostituiti con dei blocchi di input/output. Inoltre, tutti i parametri interni allo schema come l'impostazione dell'output iniziale per i blocchi di ritardo del sottosistema *Current command actions*, sono stati convertiti in variabili interne di tipo Simulink.Parameter. La simulazione è stata impostata in modo tale da avere un Fixed-step solver con step di simulazione di 0.01 secondi, tempo che si è rilevato far rispettare in modo più preciso le esecuzioni dei modelli matematici degli EV. Nella Tabella 5.1 vengono riassunte le impostazioni di simulazione per l'adattamento degli schemi.

L'ultimo step che è risultato necessario per l'adattamento degli schemi Simulink, si riferisce alla procedura di salvataggio del contesto. Come è possibile notare dai codici MatLab riportati nell'Appendice A, i modelli matematici degli EV fanno uso di variabili persistent per garantire il salvataggio di alcuni parametri tra uno step di simulazione e il successivo. Tuttavia, per assicurare che questo procedimento sia consistente anche tra una esecuzione del codice C e la successiva, le variabili persistent devono essere re-inizializzate sulla base del contesto pre-

Adattamento degli schemi Simulink
Fixed-step solver
Fixed-step size = 0.01 s
Input/Output storage class = Default

Tabella 5.1: Impostazioni degli schemi per l'adattamento alla generazione del codice C

cedente. Questo procedimento viene reso possibile da dei parametri di input/output dedicati a questo scopo. Ad esempio, nello schema riportato alla Figura 5.2, una delle variabili destinate al salvataggio del contesto è $I_{set_{old}}$ che permette di memorizzare il comando di corrente che è stato imposto all'istante precedente.

5.2.2 Importazione del codice C in Visual Studio e generazione dei file eseguibili

La generazione dei file eseguibili relativi agli schemi Simulink, è stata svolta importando il codice C ottenuto tramite Embedded Coder in Visual Studio [16]. A seguito di un'analisi iniziale sul codice C ottenuto, è stato possibile identificare le entry-point function del codice generato, che sono rispettivamente:

initialize function questa funzione permette di inizializzare le variabili di input dello schema e deve essere eseguita solo una volta;

step function ogni sua esecuzione corrisponde all'esecuzione di uno step di simulazione dello schema Simulink. Questa funzione deve essere richiamata periodicamente con periodo corrispondente al tempo di uno step di simulazione;

terminate function utilizzata per terminare l'esecuzione del codice relativo al modello.

Dato che il tempo di simulazione degli schemi è stato gestito direttamente dal simulatore (come descritto nella sezione 5.1), l'esecuzione della *step function* è avvenuta ciclicamente senza introdurre ritardi fra uno step di simulazione e il successivo al fine di ottenere i dati di output del modello nel più breve tempo possibile. Per ogni modello di veicolo elettrico che è stato modellizzato, sono stati creati un progetto su Visual Studio contenente i file ottenuti dalla generazione del codice C e un file main.c che è stato sviluppato per abilitare:

- la connessione con Redis e il trasferimento dei dati di input della simulazione del modello matematico dell'EV provenienti dal broker;
- l'esecuzione della *initialize function*;
- l'esecuzione della *step function* e il salvataggio dei dati di output dello schema in delle strutture array per ogni sua chiamata. Al fine di ottenere una simulazione di 100 secondi, questa procedura è stata svolta in un ciclo *while* di 10000 iterazioni avendo impostato lo step di simulazione a 0.01 secondi;
- l'esecuzione della *terminate function*;
- l'upload nel broker Redis degli array ottenuti relativi ai dati di output della simulazione.

Infine, per ogni progetto di Visual Studio è stato possibile ottenere i file eseguibili per ciascuno degli EV modellizzati in modo tale che questi potessero essere integrati nel simulatore del Capitolo [2](#)

Capitolo 6

Procedure di test

Grazie alla struttura modulare del progetto, è stato possibile suddividere le procedure di test in più step in modo tale da poter isolare e meglio gestire le problematiche di ognuna delle parti. Inizialmente, vengono presentati i test relativi alle comunicazioni nei protocolli CAN e ModBus. Successivamente, verranno discussi la procedura di test dei file eseguibili e, infine, il test dell'interazione fra il sistema completo e il dispositivo Unità Gestore.

6.1 Test della comunicazione su bus CAN

La prima fase di test della comunicazione CAN si è basata sulla verifica della corretta ricezione e del corretto invio dei messaggi CAN relativi al simulatore descritto al Capitolo [2](#). Per questo test, la comunicazione tramite bus CAN è stata stabilita fra il simulatore (tramite connessione COM seriale alla scheda ClusterBase di Silla) e l'USB-to-CAN analyzer IXXAT [\[17\]](#). Per la lettura e l'invio dei messaggi dal convertitore IXXAT è stato utilizzato il software canAnalyzer3 Mini di IXXAT [\[2\]](#). Il setup utilizzato per questo test viene riportato in Figura [6.1](#).

Questo primo test prevedeva l'invio e la ricezione di messaggi CAN inviati manualmente per il test della corretta gestione della porta seriale da simulatore. Successivamente, sono stati testati la gestione automatica dei messaggi ricevuti e l'invio automatico dei messaggi. Queste funzionalità, integrate in Qt Creator, sono state stabilite seguendo il protocollo CAN di Silla. A tal proposito, sono stati testati:

- i comandi di indirizzamento e la corretta risposta a questi comandi delle colonnine configurate nel cluster simulato;

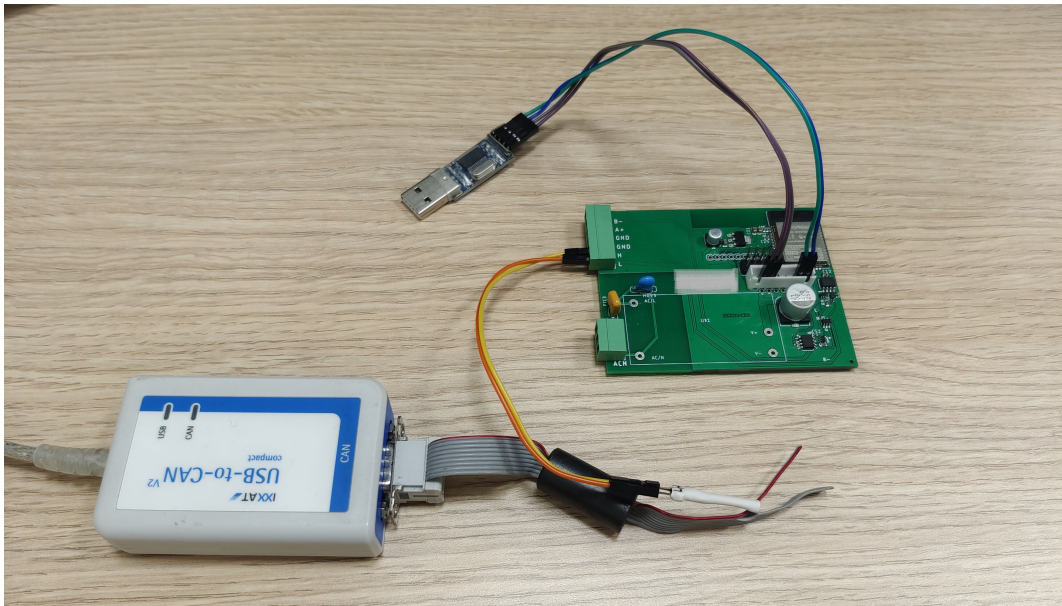


Figura 6.1: Setup di test della comunicazione tramite protocollo CAN fra simulatore e canAnalyzer3 Mini

- le direttive esterne (che possono essere inviate da un eventuale Primary collegato al bus) per la gestione del cluster simulato per quanto riguarda i comandi di avvio e interruzione della ricarica e per l'impostazione di un nuovo comando di corrente imposta;
- la corretta risposta del simulatore per quanto riguarda i messaggi relativi alle informazioni provenienti dalle colonnine simulate.

Anche per questo secondo test, il setup utilizzato corrisponde a quello riportato in Figura 6.1. Come ultimo test, è stata eseguita una simulazione di ricarica di un cluster di 50 colonnine in modo tale da monitorare l'occupazione massima del bus CAN che si avrebbe nel caso di connessione con l'Unità Gestore. E' stato registrato un picco di occupazione del 55 % che garantisce, quindi, una corretta gestione del sistema in questione.

6.2 Test della comunicazione basata su protocollo ModBus RTU

Come descritto nella sezione 2.2.1, la gestione della comunicazione basata su protocollo ModBus RTU è stata stabilita basandosi sulla struttura e sui comandi dell'energy meter SEM Three di Pick Data. Questo dispositivo stabilisce il tipo di comando ModBus e il numero di registro corrispondenti a ciascuna delle misure che offre [6]. Per questo test, la comunicazione basata su protocollo ModBus è stata stabilita fra il simulatore e il software Realterm [11] utilizzando due

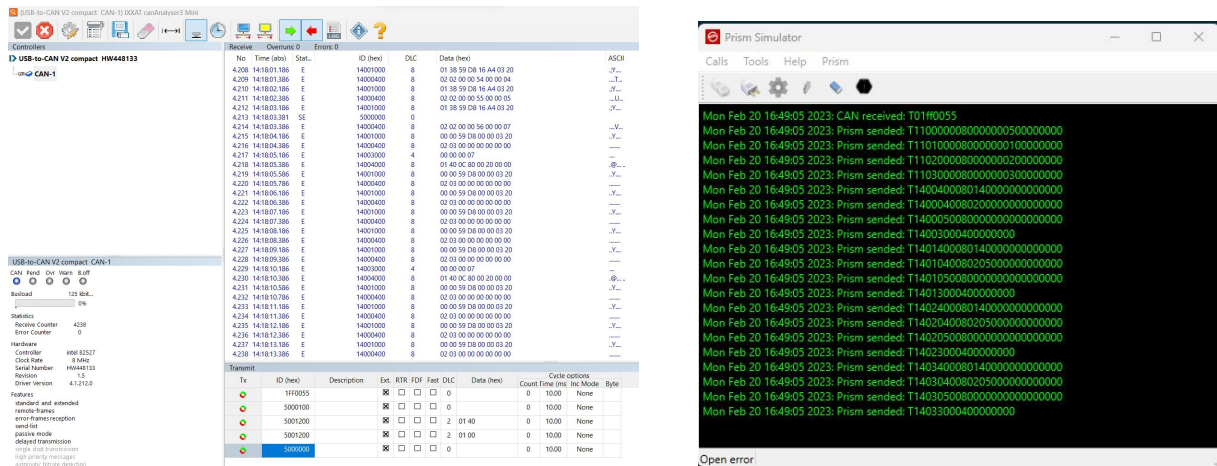


Figura 6.2: Esempio di messaggi inviati e ricevuti dal simulatore e dal software canAnalyzer3 Mini

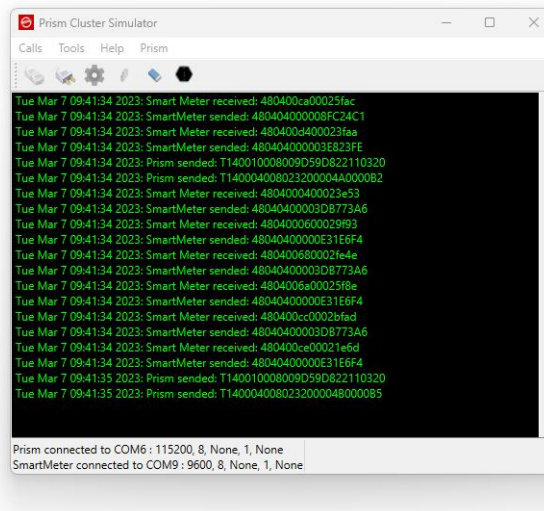


Figura 6.3: Interrogazione dell'energy meter simulato

convertitori SH-U11G di DSD TECH. Tramite questo software è stato possibile "interrogare" l'energy meter, virtualizzato tramite simulatore, per richiedere i dati relativi ai suoi registri. In Figura 6.3 viene riportata una lettura (contraddistinta dalla sigla SmartMeter) dei registri relativi a tensione, corrente, potenza e power factor dell'energy meter simulato per un sistema trifase.

6.3 Test dei file eseguibili dal simulatore

Questa procedura di test è risultata essenziale per la rifinizione della logica relativa ai modelli matematici dei veicoli elettrici. Tramite il simulatore è stato possibile ricreare diverse circostanze di ricarica che non erano state verificate tramite le caratteristiche di corrente efficace imposta

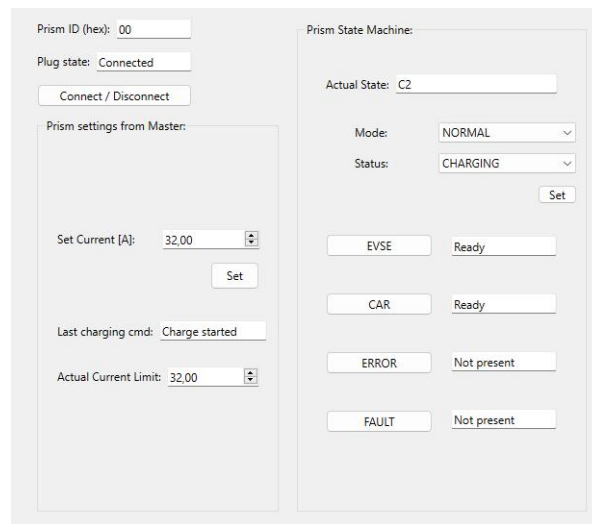


Figura 6.4: Abilitazione alla ricarica della colonnina simulata dall'interfaccia del simulatore

del Capitolo [3](#). Il test è risultato utile anche per la verifica della corretta trasmissione dei dati di simulazione fra file eseguibili, Redis e simulatore. Il setup di test prevedeva l'utilizzo del simulatore nella sua ultima versione, ovvero con l'integrazione dei file eseguibili, e del broker dati Redis. La procedura consisteva nell'abilitazione alla ricarica di una delle colonnine di ricarica del cluster simulato e nella sua connessione virtuale ad un modello di EV fra quelli presentati al Capitolo [4](#). Il test è stato poi ripetuto per ognuno dei veicoli rimanenti. Tramite l'interfaccia del simulatore, il veicolo è stato posto in stato di ricarica (vedi Figura [6.4](#)) ed è stato sottoposto a vari comandi di corrente efficace al fine di verificare che la risposta alla ricarica rispettasse le caratteristiche che sono state riscontrate nell'analisi svolta al Capitolo [3](#).

L'interfacciamento di Redis tra simulatore e file eseguibili ha reso accessibile il controllo dei dati di ricarica durante la simulazione. Tramite il broker è stato possibile verificare che, i dati di input/output e il contesto di ricarica, fossero coerenti con il comando di corrente assegnato. Le simulazioni sono risultate utili anche per verificare che la logica del simulatore fosse corretta per quanto riguarda la gestione delle colonnine di ricarica del cluster in tutte le loro funzionalità. In Figura [6.5](#) viene riportata una simulazione di ricarica in trifase per il veicolo Hyundai Ioniq 5 RWD Progress:

- nella sezione *Prism State Machine* è stata gestita la logica relativa alla macchina a stati della colonnina di ricarica simulata al fine di impostare quest'ultima nello stato *CHARGING*;
- alla sezione *Prism settings from Master* sono stati comandati una serie di valori di corrente efficace (rispettivamente $7\text{ A} \rightarrow 8\text{ A} \rightarrow 12\text{ A} \rightarrow 16\text{ A} \rightarrow 32\text{ A} \rightarrow 16\text{ A} \rightarrow 6\text{ A} \rightarrow 32\text{ A}$);

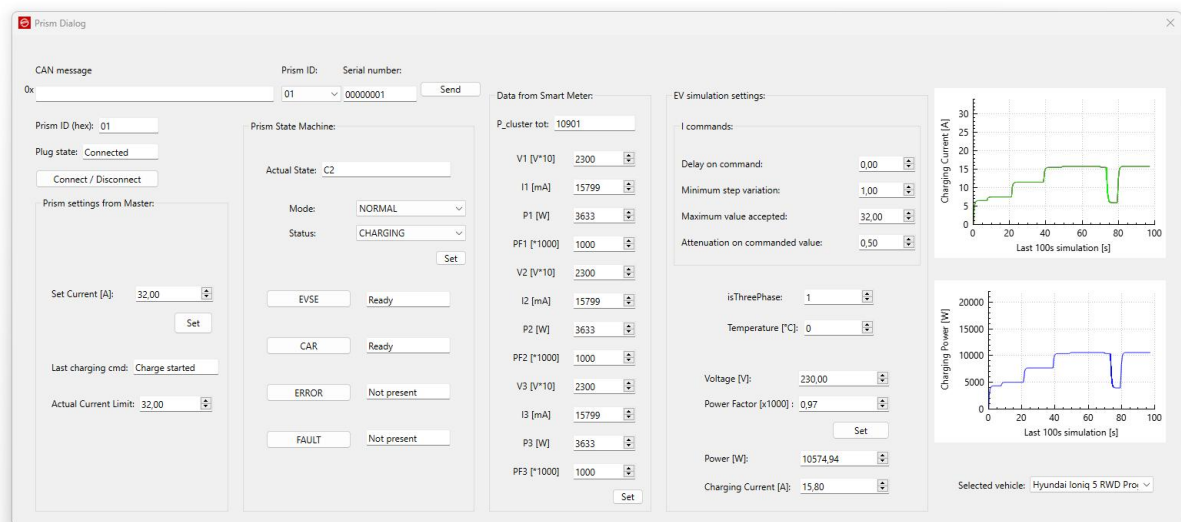


Figura 6.5: Esempio di simulazione di ricarica per il veicolo Hyundai Ioniq RWD Progress

- nella sezione *Data from Smart Meter* vengono riportate le informazioni relative all'energy meter simulato per quanto riguarda la potenza assorbita dalla rete dal cluster (nel caso in esame, il cluster era composto dalla sola colonnina della quale vediamo la simulazione di ricarica in Figura [6.5](#));
- nella sezione più a destra viene riportata una rappresentazione grafica dei dati di simulazione della ricarica.

Ripetendo la procedura di test in esame per vari comandi di corrente, è stato possibile rilevare e correggere alcune problematiche relative ai modelli matematici dei veicoli elettrici del Capitolo [4](#).

6.4 Test del sistema completo con il dispositivo Unità Gestore

In questa sezione vengono riportati i risultati dei test che sono stati eseguiti interfacciando il simulatore di ricarica con il dispositivo Unità Gestore. Nello specifico, questi test sono:

- la verifica che la procedura di indirizzamento venga eseguita correttamente e che l'Unità Gestore identifichi correttamente tutte le colonnine configurate nel cluster simulato;
- il test di ricarica con direttive provenienti dall'Unità Gestore e con cluster configurato in modo tale da essere composto da più di una colonnina di ricarica.

```
1 {
2   "Prism" : [
3     {
4       "PrismSerial" : "1000000",
5       "ID" : "00"
6     },
7     {
8       "PrismSerial" : "00000001",
9       "ID" : "01"
10    },
11    {
12      "PrismSerial" : "00000002",
13      "ID" : "02"
14    },
15    {
16      "PrismSerial" : "00000003",
17      "ID" : "03"
18    },
19    {
20      "PrismSerial" : "00000004",
21      "ID" : "04"
22    },
23    {
24      "PrismSerial" : "00000005",
```

Figura 6.6: File json di configurazione del cluster simulato

La prima fase di test è stata inizialmente svolta con un cluster composto da 5 punti di ricarica e successivamente con un cluster composto da 50 punti di ricarica. In entrambi i casi, l'Unità Gestore ha rilevato correttamente tutte le colonnine configurate nel cluster ed è stato capace di interagire con esse al fine di completare la procedura di indirizzamento dei secondary presenti nel cluster.

Per quanto riguarda la seconda fase di test, la simulazione di ricarica si è inizialmente svolta su un cluster configurato con 4 colonnine di ricarica, ciascuna collegata ad un veicolo elettrico fra i modelli: modello di veicolo generico, Tesla Model S (2014), Hyundai Ioniq 5 RWD Progress e Opel Corsa-e. Ogni punto di ricarica è stato posto nello stato di ricarica (vedi Figura 6.4) e per ciascuno di essi sono stati testati i comandi di avvio/stop della ricarica e set di corrente con diversi valori. Nelle Figure 6.7 e 6.8, vengono riportati i risultati delle simulazioni dei 4 punti di ricarica. L'interfacciamento fra Unità Gestore e simulatore, con questa prima configurazione, non ha riportato alcuna problematica. Il simulatore ha risposto in modo corretto ai comandi inviati dall'Unità Gestore. Inoltre, le simulazioni di ricarica dei veicoli elettrici presi in considerazione, hanno riportato caratteristiche congrue con l'analisi effettuata sui dati di ricarica reali.

Per questa seconda fase di test, si è quindi deciso di testare l'interfacciamento fra i due sistemi con cluster composto da un numero maggiore di punti di ricarica simulati. Il setup di test ha

quindi previsto la configurazione di un cluster di 50 colonnine simulate che sono state poste una ad una in stato di ricarica (vedi Figura 6.4) al fine di verificare il numero massimo di simulazioni di ricarica che possono essere eseguite in contemporanea. Il test ha riportato buoni risultati fino ad un numero di 20 colonnine simulate in stato di ricarica. Al di sopra di questo valore, il dispositivo Unità Gestore ha riportato dei problemi nella gestione del carico di messaggi CAN e ModBus inviati dal simulatore. Infatti, secondo il protocollo CAN di Silla Industries, il numero di messaggi CAN inviati quando un punto di ricarica viene posto nello stato C2, è molto maggiore rispetto ad un punto di ricarica posto nello stato A1. Dato che il dispositivo Unità Gestore è ancora in fase di sviluppo, questo test ha permesso di individuare la problematica relativa ai due protocolli di comunicazione al fine di ottimizzarne la gestione. Fino al numero di 20 punti di ricarica simulati, l'interfacciamento fra i due sistemi ha condotto con successo la gestione dell'avvio/stop della ricarica e della variazione della corrente comandata dal dispositivo Unità Gestore per ognuno di questi punti di ricarica. Inoltre, anche in questo caso, le simulazioni di ricarica dei veicoli presi in considerazione hanno riportato caratteristiche che rispettano i risultati ottenuti dall'analisi svolta sui dati reali.

Al fine di isolare il test al solo simulatore per verificarne il corretto funzionamento, questo è stato interfacciato al software RealTerm, per la gestione della comunicazione ModBus, e al software canAnalyzer3 Mini, per la gestione della comunicazione CAN. Dai due software sono stati gestiti i messaggi CAN e ModBus al fine di simulare il funzionamento dell'Unità Gestore; per quanto riguarda il protocollo Modbus, sono stati inviati periodicamente i messaggi relativi alla richiesta di lettura dei registri dello SmartMeter simulato, mentre per quanto riguarda la comunicazione CAN, sono stati inviati i comandi relativi alla procedura di inizializzazione della comunicazione e i comandi relativi all'avvio/stop della ricarica e al set di corrente imposta. Dai test effettuati su un cluster di 50 colonnine poste in stato di ricarica, è sorto che il simulatore è riuscito a rispondere correttamente ai messaggi inviati tramite i due protocolli di comunicazione. Il simulatore è inoltre riuscito a gestire le simulazioni di ricarica relative ai 50 veicoli elettrici simulati, rispondendo in modo corretto ai comandi di avvio/interruzione della ricarica e ai comandi di variazione di corrente imposta.

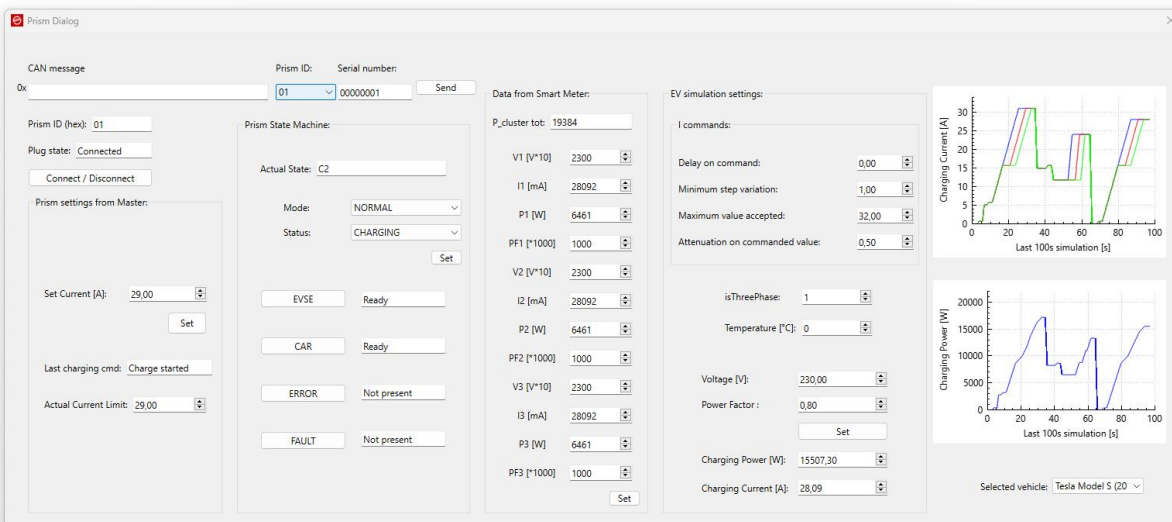
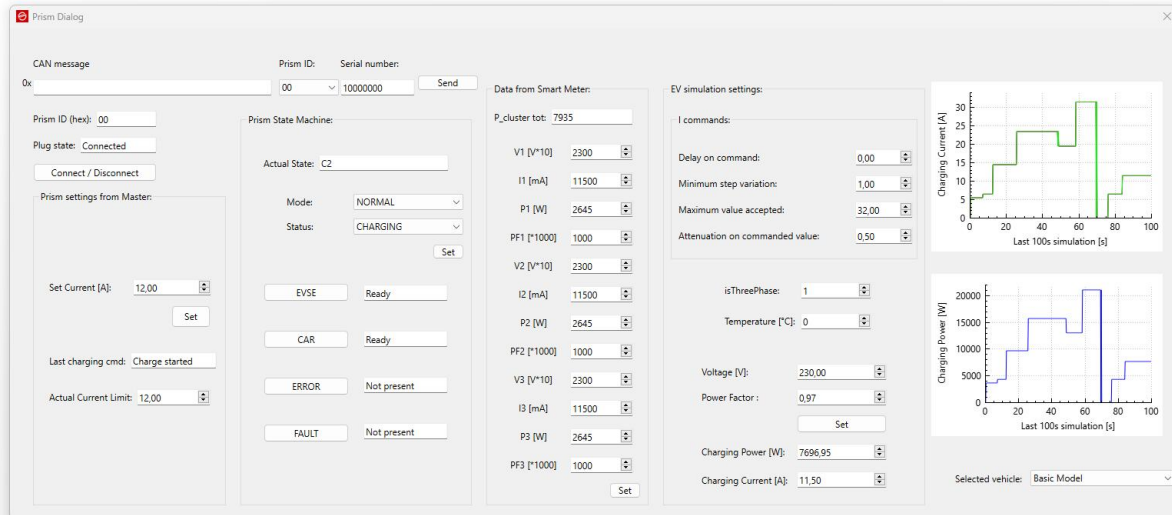


Figura 6.7: Simulazioni di ricarica con il dispositivo Unità Gestore: modello di veicolo generico e Tesla Model S (2014)

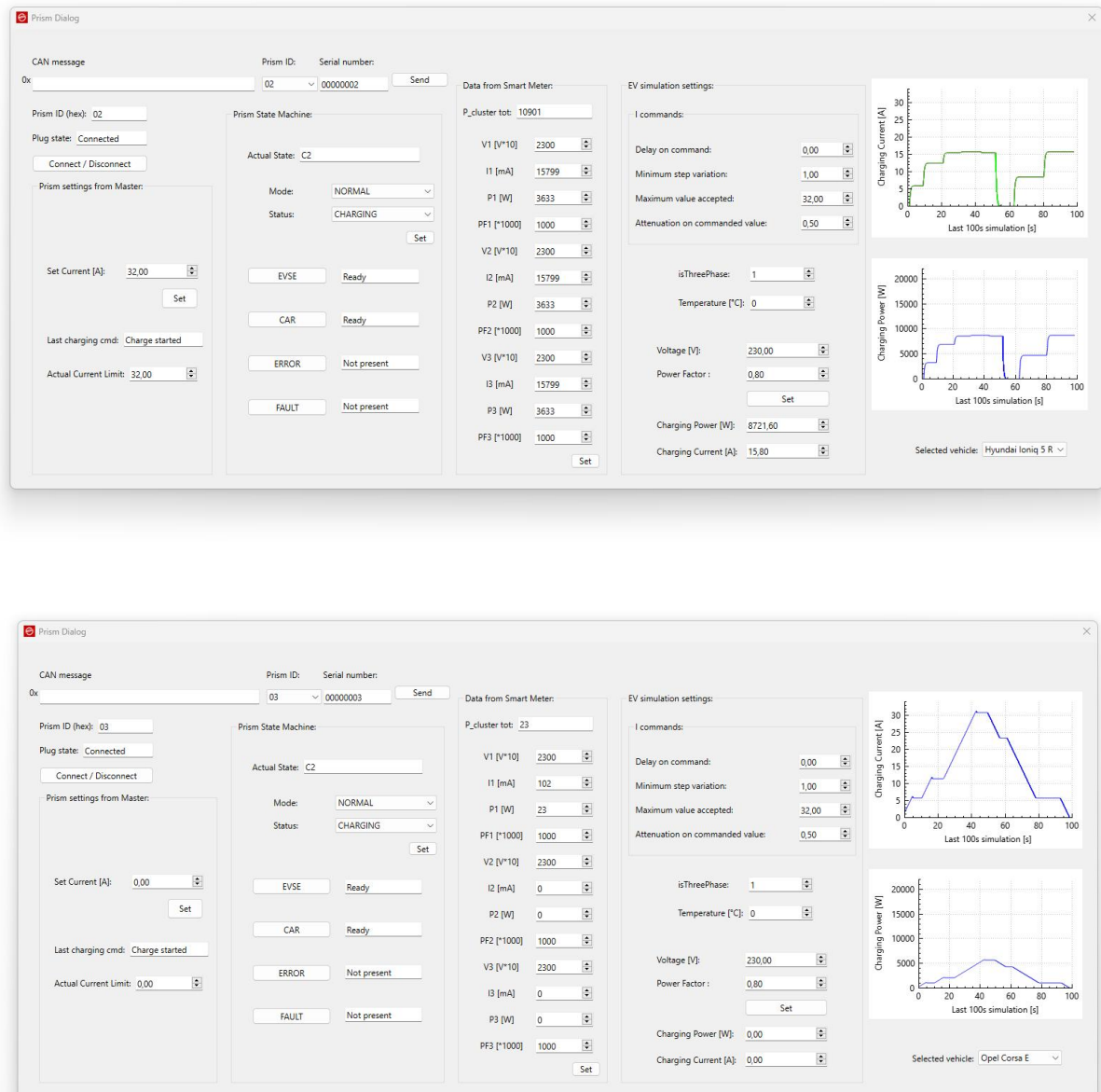


Figura 6.8: Simulazioni di ricarica con il dispositivo Unità Gestore: Hyundai Ioniq 5 RWD Progress e Opel Corsa-e

Conclusioni

All'inizio di questo progetto di tesi, l'obiettivo era quello di realizzare un sistema di simulazione di ricarica AC di un cluster di veicoli elettrici che permettesse di riprodurre le dinamiche che sono state riscontrate durante la ricarica di alcuni modelli di EV.

Il progetto è cominciato con una fase di studio delle colonnine di ricarica AC, focalizzandosi soprattutto sulle informazioni riguardanti la comunicazione fra caricatore e veicolo e sulle fasi della ricarica. E' stata poi svolta una ricerca degli strumenti, fra software e hardware, necessari per la realizzazione del sistema completo.

Successivamente, si è svolta la fase di tirocinio che prevedeva la realizzazione del simulatore, tramite Qt Creator, per la generazione dell'interfaccia del simulatore, la virtualizzazione delle colonnine di ricarica, la gestione delle comunicazioni su bus CAN e RS-485 e per la gestione, lato simulatore, del broker dati Redis .

Lo step successivo ha determinato l'inizio del lavoro di tesi, che si è inizialmente basato sull'analisi dei profili di ricarica di vari modelli di veicoli elettrici e, successivamente, su test di ricarica con caratteristiche di corrente differenti, con l'obiettivo di identificare ulteriori criticità legate alla ricarica. Lo studio si è poi concentrato sulla realizzazione di modelli matematici, su MatLab/Simulink, che potessero rappresentare al meglio la dinamica di ricarica dei veicoli elettrici presi in considerazione.

Una volta confrontati i risultati ottenuti con i dati di ricarica originali, il progetto è proseguito con la generazione del codice C dagli schemi Simulink, mediante il tool Embedded Coder, e con l'importazione del codice ottenuto in una serie di progetti su Visual Studio, al fine di generare dei file eseguibili che prevedessero l'importazione dei dati di input degli schemi Simulink dal broker dati Redis, la simulazione di ricarica del veicolo preso in considerazione e l'upload

dei risultati ottenuti di nuovo su Redis. Quest'ultimi sono poi stati integrati nel simulatore realizzato tramite Qt Creator.

Il progetto si è concluso con la fase di test al fine di verificare che le singole parti del sistema ottenuto funzionassero rispettando le direttive di progetto e, infine, con il test del sistema completo interfacciato al dispositivo Unità Gestore.

I test hanno dimostrato che gli obiettivi iniziali sono stati portati a termine con successo, e che il progetto ha raggiunto buoni risultati, rappresentando un punto di partenza per possibili sviluppi futuri.

Sviluppi futuri

In primis, il sistema finale può essere arricchito con l'integrazione delle dinamiche di ricarica di ulteriori modelli di veicoli elettrici. Come sviluppo ulteriore, il simulatore può essere adattato al fine di rappresentare veicoli elettrici abilitati alla tecnologia V2G. Per garantire questa ultima implementazione, sarebbe necessario studiare le dinamiche assunte dai veicoli elettrici durante il processo di scarica della batteria, al fine di realizzare dei modelli matematici che le possano rappresentare. Inoltre, sarebbe necessario modificare le attuali colonnine di ricarica virtualizzate, al fine di renderle compatibili con la tecnologia V2G. Questa implementazione potrebbe risultare utile per simulare l'energia proveniente da un parcheggio di veicoli elettrici nell'ambito di progetti legati alle smart grids.

Appendice A

Listati delle funzioni MatLab contenute negli schemi Simulink

Listati per il modello di veicolo generico

Listato per la funzione MatLab *Lower limit and quantization*

```
1 function Iout = fcn(Iin, Iset_old, I_step, I_delta)
2
3 persistent Iin_prev;      %used to save Iin(t-1)
4 persistent Iout1_prev;    %used to save Iout1(t-1)
5
6 if isempty(Iout1_prev)
7     Iout1_prev = Iset_old;
8 end
9
10 if isempty(Iin_prev)
11     Iin_prev = Iset_old;
12 end
13
14 %set lower current limit
15 if(Iin<6)
16     Iout = I_delta;
17
18 %set current quantization
19 elseif(abs(Iin-Iin_prev) < I_step)
20     Iout = Iout1_prev;
21 else
22     Iout = Iin;
23     Iin_prev = Iin;
24 end
25
26 Iout1_prev = Iout;
```

```
27  
28 end
```

Listati per lo schema Simulink del veicolo Hyundai Ioniq 5 RWD Progress

Listato per la funzione MatLab *Lower limit and quantization*

```
1 function Iout1 = fcn(Iin, Iset_old, I_step)  
2  
3 persistent Iin_prev;      %used to save Iin(t-1)  
4 persistent Iout1_prev;    %used to save Iout1(t-1)  
5  
6 if isempty(Iout1_prev)  
7     Iout1_prev = Iset_old;  
8 end  
9  
10 if isempty(Iin_prev)  
11     Iin_prev = Iset_old;  
12 end  
13  
14 %set lower current limit  
15 if (Iin < 6)  
16     Iout1 = 0.512;  
17  
18 %set current quantization  
19 elseif (abs(Iin - Iin_prev) < I_step)  
20     Iout1 = Iout1_prev;  
21 else  
22     Iout1 = Iin;  
23     Iin_prev = Iin;  
24 end  
25  
26 Iout1_prev = Iout1;  
27  
28 end
```

Listato per la funzione MatLab *Difference I delta*

```
1 function Iout = fcn(Iin2)  
2  
3 if (Iin2 > 16)  
4     Iout = 16 - 0.2;
```

```

5 elseif(Iin2 >= 6 && Iin2 <= 6.3)
6     Iout = 6-0.1;
7 else
8     Iout = (Iin2-0.512);
9 end

```

Listati per lo schema Simulink del veicolo Tesla Model S (2014)

Listato per la funzione MatLab *Lower limit and quantization*

```

1 function [Iout, Iset_old, I_init]= fcn(Iin, Iset_old, I_step)
2
3 persistent Iin_prev;           %used to save Iin(t-1)
4 persistent Iout_prev;          %used to save Iout(t-1)
5
6 if isempty(Iout_prev)
7     Iout_prev = Iset_old;
8 end
9
10 if isempty(Iin_prev)
11     Iin_prev = Iset_old;
12 end
13
14 %set lower current limit
15 if(Iin<6)
16     Iout = 1.128;              %Idelta
17 %set current quantization
18 elseif(abs(Iin-Iin_prev) < I_step)
19     Iout = Iout_prev;
20 else
21     Iout = Iin;
22     Iin_prev = Iin;
23 end
24
25 I_init = Iset_old;             %used only in initialization
26
27 Iout_prev = Iout;
28
29 end

```

Listato per la funzione MatLab *Tesla Model S dynamics*

```

1 function [Iout1, Iout2, Iout3] = fcn(Iin, I_old, I_init)
2
3 persistent Iout1_temp; %new calculated value for Iout1
4 persistent Iout2_temp; %new calculated value for Iout2
5 persistent Iout3_temp; %new calculated value for Iout3
6 persistent I2;
7 persistent Iin_old; %Iin(t-1)
8 persistent type; %used to select the type of dynamic to plot
9 persistent I_delta; %used to choose the I_delta
10 persistent m; %used to choose the m
11 persistent mp2; %used to choose m referred to phase 2
12 persistent mp3; %used to choose m referred to phase 3
13 persistent stop1; %used to stop the dynamics of phase 1 stop1=0 -> go
    on, stop1=1 -> stop
14 persistent stop2; %used to stop the dynamics of phase 2 stop2=0 -> go
    on, stop2=1 -> stop
15 persistent stop3; %used to stop the dynamics of phase 3 stop3=0 -> go
    on, stop3=1 -> stop
16 persistent count; %used for counting the steps of the dynamic(for pre
    -assessment and peak creation)
17
18 %I_delta for steady-state value Icmd - Iobt
19 %rising edge
20 I_delta1 = 0.9;
21 %falling edge
22 I_delta2 = 0.4; %I > 16 A
23 I_delta3 = 1.128; %I < 16 A
24
25 t = 0.01; %Ts = function sample time
26
27 %persistent variables initialization
28 if isempty(Iin_old)
29     Iin_old = I_old;
30 end
31 if isempty(Iout1_temp)
32     Iout1_temp = I_init;
33 end
34 if isempty(Iout2_temp)
35     Iout2_temp = I_init;
36 end

```

```

37 if isempty(Iout3_temp)
38     Iout3_temp = I_init;
39 end
40 if isempty(I2)
41     I2 = I_init;
42 end
43 if isempty(type)
44     type = 0;
45 end
46 if isempty(I_delta)
47     I_delta = I_delta1;
48 end
49 if isempty(m)
50     m = 0;
51 end
52 if isempty(mp2)
53     mp2 = 0;
54 end
55 if isempty(mp3)
56     mp3 = 0;
57 end
58 if isempty(stop1)
59     stop1 = 0;           %set to avoid the first step in order to set Iout to
                           I_init
60 end
61 if isempty(stop2)
62     stop2 = 0;           %set to avoid the first step in order to set Iout to
                           I_init
63 end
64 if isempty(stop3)
65     stop3 = 0;           %set to avoid the first step in order to set Iout to
                           I_init
66 end
67 if isempty(count)
68     count = 0;
69 end
70
71 %slopes angular coefficient (put below because of persistent variables)
72 %for I_step <= 1
73 %m1 = Iin-Iin_old; %defined below cause otherwise it would change at each

```

```
    cicle
74 %for I_step > 1
75 m2 = 5.317;
76 m3 = 1.755;
77 mA = 0.68;           %slope for pre-assessment
78 mB = 1.8;           %slope for post-settling for 2s
79 mC = -3.909;        %slope for post assessment in type 2
80 %falling edge
81 %m3 = Iin-Iin_old;    %defined below cause otherwise it would change at each
    cicle
82
83 Iout1 = Iout1_temp;    %Iout update      (used cause needed by matlab and to
    set Iout to the precedent value in case)
84 Iout2 = Iout2_temp;
85 Iout3 = Iout3_temp;
86
87 %Select the type of dynamic we are "plotting"
88 if(Iin ~= Iin_old)      %if a change in current command has been
    detected..
89
90     if(Iin-I_delta > Iout1)      %if we are in a rising edge
91
92         I_delta = I_deltal;
93         if(abs(Iin-Iin_old) <= 1)      %if the rising step is lower
    than 1
94             m = abs(Iin-Iin_old);
95             type = 0;
96         elseif(Iin <= 16)      %if the rising step is higher
    than 1 and if Iout doesn't pass through 16 A (Iout is always smaller
    than Iin)
97             m = m2;
98             type = 1;
99         elseif(Iin_old >= 13 && Iin_old <= 16) %if the rising step starts
    from 16 A
100             m = m2;
101             I2 = Iout1;
102             mp2 = m;
103             mp3 = m;
104             type = 2;
105         elseif(Iin_old > 16)      %if the rising step starts from a
```

```

value above 16 A
106         if(count ~= 0)                                %condition to let the previous
state to finish the dynamic of 18 seconds lenght
107             type = 2;
108         else
109             m = m2;
110             mp2 = m;
111             mp3 = m;
112             type = 3;
113         end
114         elseif(Iin_old<16 && Iin>16)                    %if we are starting from below 16 A
and ending above 16 A
115             m = m3;
116             mp2 = m;
117             mp3 = m;
118             type = 4;
119         end
120
121     else                                                %if we are in a falling edge
122
123         m = -abs(Iin-Iin_old);
124         mp2 = m;
125         mp3 = m;
126         type = 5;
127         if(Iin > 16)                                    %selection of the I_delta as a
function of the graph region
128             I_delta = I_delta2;
129         else
130             I_delta = I_delta3;
131         end
132
133     end
134
135     Iin_old = Iin;                                     %update Iin_old for the next cicle
136
137     %used to start the dynamic
138     stop1 = 0;
139     stop2 = 0;
140     stop3 = 0;
141 end

```

```

142
143 %Starts and stops the dynamic for phase 1
144 if(stop1 == 0 || stop2 == 0 || stop3 == 0)
145
146     if(m < 0)           %check for m variation for curve 2 and 3
147         if(Iout2 < (Iin-I_delta))
148             if(type == 2 || type == 3)
149                 mp2 = m2;
150             else
151                 mp2 = m3;
152             end
153         end
154
155         if(Iout3 < (Iin-I_delta))
156             if(type == 2 || type == 3)
157                 mp3 = m2;
158             else
159                 mp3 = m3;
160             end
161         end
162     else
163         if(Iout2 > (Iin-I_delta))
164             mp2 = -m;
165         end
166
167         if(Iout3 > (Iin-I_delta))
168             mp3 = -m;
169         end
170     end
171
172     %dynamic management phase1
173
174     switch type
175
176     case 0
177
178         Iout1_temp = Iout1_temp + m*t;
179
180         %update the other phases (in this condition they have the same
181         %value of the first one)

```



```

182         Iout2_temp = Iout2_temp + m*t;
183         Iout3_temp = Iout3_temp + m*t;
184
185     case 1
186
187         %pre-assessment slope
188         if(count <= 1/t)                                %first second with mA
189             ramp
190                 Iout1_temp = Iout1_temp + mA*t;
191                 Iout2_temp = Iout2_temp + mA*t;
192                 Iout3_temp = Iout3_temp + mA*t;
193                 count = count+1;
194             elseif(count > 1/t && count <= 3*1/t)        %second and third
195                 seconds with assessment
196                     Iout1_temp = Iout1;
197                     Iout2_temp = Iout2;
198                     Iout3_temp = Iout3;
199                     count = count+1;
200             else
201                 Iout1_temp = Iout1_temp + m*t;
202                 Iout2_temp = Iout2_temp + m*t;
203                 Iout3_temp = Iout3_temp + m*t;
204             end
205
206     case 2
207
208         if(Iout1 < I2+1 && count == 0)                    %condition to reach
209             15.7 A and not executing it again
210
211             Iout1_temp = Iout1_temp + mA*t;
212             %update the other phases (in this condition they have the
213             same
214             %value of the first one)
215             Iout2_temp = Iout1_temp + mA*t;
216             Iout3_temp = Iout1_temp + mA*t;
217
218         elseif(count >= 0 && count <= 2*1/t)            %two seconds of
219             assessment

```

```

217         Iout1_temp = Iout1;
218         %update the other phases (in this condition they have the
same
219         %value of the first one)
220         Iout2_temp = Iout2;
221         Iout3_temp = Iout3;
222         count = count+1;
223
224         elseif(count > 2*1/t && count <= 3*1/t)    %negative slope to
reach 11.8 A
225
226         Iout1_temp = Iout1_temp + mC*t;
227         %update the other phases (in this condition they have the
same
228         %value of the first one)
229         Iout2_temp = Iout2_temp + mC*t;
230         Iout3_temp = Iout3_temp + mC*t;
231         count = count+1;
232
233         elseif(count > 3*1/t && count <= 11*1/t)    %8s of assessment
234
235         Iout1_temp = Iout1;
236         %update the other phases (in this condition they have the
same
237         %value of the first one)
238         Iout2_temp = Iout2;
239         Iout3_temp = Iout3;
240         count = count+1;
241
242         elseif(count > 11*1/t && count <= 15*1/t)    %first phase
dynamic
243
244         Iout1_temp = Iout1_temp + m*t;
245         Iout2_temp = Iout2;
246         Iout3_temp = Iout3;
247         count = count+1;
248
249         elseif(count > 15*1/t && count <= 18*1/t)    %if 4s are passed,
also the second phase starts
250

```

```

251         Iout1_temp = Iout1_temp + m*t;
252         Iout2_temp = Iout2_temp + mp2*t;
253         Iout3_temp = Iout3;
254         count = count+1;
255
256         elseif(count > 18*1/t)    %if 3s are passed, also the third
phase starts
257             Iout1_temp = Iout1_temp + m*t;
258             Iout2_temp = Iout2_temp + mp2*t;
259             Iout3_temp = Iout3_temp + mp3*t;
260             count = count+1;
261         end
262
263     case 3
264
265         Iout1_temp = Iout1_temp + m*t;
266         Iout2_temp = Iout2_temp + mp2*t;
267         Iout3_temp = Iout3_temp + mp3*t;
268
269     case 4
270
271         %pre-assessment slope
272         if(count <= 1/t)                                %first second with mA
ramp
273             Iout1_temp = Iout1_temp + mA*t;
274             %update the other phases (in this condition they have the
same
275             %value of the first one)
276             Iout2_temp = Iout2_temp + mA*t;
277             Iout3_temp = Iout3_temp + mA*t;
278             count = count+1;
279
280         elseif(count > 1/t && count <= 3*1/t)    %second and third
seconds with assessment
281
282             Iout1_temp = Iout1;
283             %update the other phases (in this condition they have the
same
284             %value of the first one)
285             Iout2_temp = Iout2;

```

```
286         Iout3_temp = Iout3;
287         count = count+1;
288
289         elseif(Iout1 < Iin*47/100)           %condition to reach 15.7
290         A and not executing it again
291
292         Iout1_temp = Iout1_temp + m*t;
293         %update the other phases (in this condition they have the
294         same
295         %value of the first one)
296         Iout2_temp = Iout2_temp + m*t;
297         Iout3_temp = Iout3_temp + m*t;
298
299         elseif(count > 3*1/t && count <= 7*1/t)    %first phase dynamic
300
301         Iout1_temp = Iout1_temp + m*t;
302         Iout2_temp = Iout2;
303         Iout3_temp = Iout3;
304         count = count+1;
305
306         elseif(count > 7*1/t && count <= 10*1/t)    %if 4s are passed,
307         also the second phase starts
308
309         Iout1_temp = Iout1_temp + m*t;
310         Iout2_temp = Iout2_temp + mp2*t;
311         Iout3_temp = Iout3;
312         count = count+1;
313
314         elseif(count > 10*1/t)    %if 3s are passed, also the third
315         phase starts
316
317         Iout1_temp = Iout1_temp + m*t;
318         Iout2_temp = Iout2_temp + mp2*t;
319         Iout3_temp = Iout3_temp + mp3*t;
320         count = count+1;
321     end
322
323     case 5
324
325         Iout1_temp = Iout1_temp + m*t;
326         Iout2_temp = Iout2_temp + mp2*t;
```

```

322         Iout3_temp = Iout3_temp + mp3*t;
323
324     end
325
326     %checks if threse's the need to make another step for each phase
327     if(abs((Iin-I_delta)-Iout1) <= abs(Iout1_temp-Iout1))    %if the
328     difference between the desired value and the actual one is smaller than
329     the step that the algorithm would make, the slope is stopped
330         stop1 = 1;                                           %stops the
331     dynamic
332         Iout1_temp = Iout1;
333     else
334         Iout1 = Iout1_temp;
335     end
336
337     if(abs((Iin-I_delta)-Iout2) <= abs(Iout2_temp-Iout2))    %if the
338     difference between the desired value and the actual one is smaller than
339     the step that the algorithm would make, the slope is stopped
340         stop2 = 1;                                           %stops the
341     dynamic
342         Iout2_temp = Iout2;
343     else
344         Iout2 = Iout2_temp;
345     end
346
347     if(abs((Iin-I_delta)-Iout3) <= abs(Iout3_temp-Iout3))    %if the
348     difference between the desired value and the actual one is smaller than
349     the step that the algorithm would make, the slope is stopped
350         stop3 = 1;                                           %stops the
351     dynamic
352         Iout3_temp = Iout3;
353         count = 0;                                           %since phase3
354         is always the last to reach the maximum value, count is resetted here
355     else
356         Iout3 = Iout3_temp;
357     end
358 end
359 end

```


Ringraziamenti

Happiness is only real

when shared

- Christopher McCandless

Questo lavoro di tesi segna la conclusione di un percorso di studi che si è dimostrato impegnativo e ricco di sfide ma che, d'altro canto, mi ha regalato emozioni che porterò con me per il resto della mia vita. Di certo alcune di queste sono legate alle soddisfazioni che sono riuscito ad ottenere nella mia carriera universitaria ma qui, in particolare, voglio soffermarmi sulle emozioni che ho avuto la fortuna di condividere con le persone che mi sono state accanto in questo percorso e che hanno contribuito a renderlo così speciale.

Il primo pensiero va ai miei nonni, nonna Ida e nonno Giuseppe da parte di mamma, nonna Giovanna e nonno Silvio da parte di papà. Mi avete aiutato a realizzare i miei sogni, dandomi tutto senza mai chiedere niente in cambio. Conserverò con cura il vostro affetto e la vostra saggezza.

Un ringraziamento particolare va alla mia fidanzata Francesca per essermi sempre stata accanto in questi mesi intensi. Semplicemente, grazie per essere la bella persona che sei.

Vorrei ringraziare anche i miei amici, partendo dalle compagnie di Trissino e di Valdagno. Grazie per avermi regalato ricordi, svago e momenti di spensieratezza. In particolare, ringrazio i miei carissimi amici Dario, Francesco e Melly, sui quali ho sempre saputo di poter contare. Ringrazio anche quella che reputo essere la mia seconda famiglia, composta dai miei coinquilini Silvia, Gabriele e Adele, con cui ho convissuto per gli scorsi 5 anni, e i miei vicini di casa Mike, Davide, Oriana, Chiara, Alberto, Emma e Cristiana con cui ho avuto il piacere di condividere bellissime esperienze.

E' il turno dei compagni di corso! Ringrazio i futuri ingegneri di successo, Antonio, Meggi, Andrea, Marco, Tremo, Pilati e Moz con cui ho avuto il piacere di condividere lo studio e, in particolare, ringrazio il "dream team" composto da Mattia e Bego con cui ho svolto i progetti universitari e con cui ho avuto il piacere di passare gran parte delle mie giornate.

Un doveroso ringraziamento va al mio relatore Nicola Trivellin per il supporto dato in questi mesi e per l'impegno e dedizione dimostrati nel tenere i corsi universitari che ho avuto il piacere di seguire.

Inoltre mando un grande abbraccio a tutto il team di Silla Industries che mi ha accolto a braccia aperte fin dai primi 10 minuti di lavoro. In primis vorrei ringraziare il mio tutor aziendale Michele Brunelli per il sostegno e la motivazione dati negli scorsi mesi e il mio responsabile software Andrea Giusti per essere uno dei migliori che potevo incontrare sia in termini di conoscenze che di gestione del team. Un ringraziamento particolare va a Mattia Donami per l'enorme aiuto e la disponibilità che mi ha dedicato durante tutta la mia permanenza in Silla. Inoltre, ringrazio tutto il team del reparto Ricerca e Sviluppo e, infine, il team del calcetto Silla che è stato fondamentale per lo sfogo nei giorni in cui niente andava secondo i piani.

Infine, come non ringraziare la mia bella famiglia allargata, papà Luca, i miei fratelli e sorelle Alessandro, Whitney, Andrea e Gabrielle, i compagni Maurizio e Irene e in particolare mamma Laura, alla quale voglio dedicare questo lavoro di tesi, per esserci stata sempre e per avermi reso la persona che sono.

Un brindisi a tutti voi,

Giovanni

Bibliografia

- [1] Giuseppe Buja. *Course of Electric Road Vehicles*. 2016-2017.
- [2] IXXAT canAnalyser3 Mini. <http://www.eskorea.net/html/data/support/IxMmV13.pdf>.
- [3] Embedded Coder. <https://it.mathworks.com/products/embedded-coder.html>.
- [4] Qt Creator. <https://www.qt.io/>.
- [5] Energy Meter SEM Three di Pick Data. <https://www.pickdata.net/sem-three-three-phase-modbus-energy-meter>.
- [6] Manual SEM Three di Pick Data. https://www.pickdata.net/sites/default/files/Manual_SEM_Three_V11-300421-EN.pdf.
- [7] SmartHub di Silla Industries. <https://silla.industries/prism-smart-hub/>.
- [8] Libreria hiredis. <https://github.com/redis/hiredis>.
- [9] Silla Industries. <https://silla.industries/>.
- [10] ST On-Board-Chargers. <https://www.st.com/en/applications/electro-mobility/on-board-charger-obc.html>.
- [11] Realterm. <https://realterm.sourceforge.io/>.
- [12] Redis. <https://redis.io/>.
- [13] Convertitore USB-RS-485 SH-U11G. <http://www.dsdtech-global.com/2017/04/SH-U11.html>.
- [14] Simulink. <https://it.mathworks.com/products/simulink.html>.

- [15] IEC Standards. <https://webstore.iec.ch/home>.
- [16] Visual Studio. <https://visualstudio.microsoft.com/it/>.
- [17] IXXAT USB to CAN. <https://www.ixxat.com/products/pc-interfaces-overview/details/usb-to-can-v2-professional>.
- [18] Curve Fitting Toolbox. <https://it.mathworks.com/products/curvefitting.html>.
- [19] Nicola Trivellin, Claudio Narduzzi, Giada Giorgi, and Matteo Bertocco. *Course of Automotive & Domotic*. 2021-2022.