

POLITECNICO DI TORINO

CORSO DI LAUREA MAGISTRALE
IN INGEGNERIA ELETTRICA

TESI DI LAUREA MAGISTRALE

**Integrazione dei sistemi Power-to-Gas
nella Rete di Distribuzione Elettrica**



Candidato:
Gabriele NATOLI

Relatori:
Prof. Ettore BOMPARD
Ing. Andrea MAZZA

A.A. 2019/2020

*Alla mia famiglia.
Ai miei nonni.
Sempre.
Per sempre.*

Indice

Abstract	I
Abstract	II
Elenco delle figure	III
Elenco delle tabelle	VI
Lista dei Simboli	VI
Acronimi	IX
1 Decarbonizzazione del sistema energetico	1
1.1 Riscaldamento globale	1
1.2 Necessità di decarbonizzazione del sistema energetico	3
1.3 Sinergia tra elettricità e gas	6
1.4 Progetto PLANET - H2020	8
1.4.1 Caso studio: uso del P2G per alleviare il flusso di potenza inverso e migliorare gli indici di rete	11
1.5 Flow-chart riassuntivo dell'intero lavoro di tesi	13
2 Il sistema Power-to-Gas	14
2.1 Componenti del sistema	14
2.1.1 Elettrolizzatore	15
2.1.2 Produzione di CO ₂	18
2.1.3 Compressione e stoccaggio dell'idrogeno	20
2.2 Osservazioni finali sul P2G	20
3 Analisi Dati	22
3.1 Studio profili di carico residenziali	22
3.1.1 Identificazione profili di carico	22
3.1.2 Profili a diverse fasce di consumo e classificazione valori	25
3.1.3 Scelta dei profili di carico utilizzabili	27
3.1.4 Ricostruzione per interpolazione lineare vs. ricostruzione basata su estrazioni casuali dei valori: analisi e confronto	29
3.1.5 Giustificazione scelta profili	41
3.1.6 Creazione aggregato di carico residenziale	44
3.2 Studio profili di carico utenti non domestici	48
3.2.1 Estrazione dati da Excel	48

3.2.2	Riporto in per unità dei profili non residenziali	49
4	Studio della rete elettrica Test	59
4.1	Cenni sulla rete utilizzata	59
4.2	Studio della dorsale Fiat della rete Test	62
4.2.1	Ricerca coordinate e creazione aggregati nodali	62
4.2.2	Risoluzione della rete tramite Backward Forward Sweep (BFS)	67
4.3	Studio delle altre dorsali della rete test	73
4.3.1	Creazione aggregati per le altre dorsali della rete	73
4.3.2	Risoluzione della rete completa mediante BFS	81
5	Profili di generazione	84
5.1	Problematiche legate alla presenza di risorse distribuite nella rete elettrica	84
5.2	Costruzione dei profili di generazione fotovoltaica	86
5.2.1	Cenni sul codice di creazione dei profili di irradianza	86
5.2.2	Generazione di profili di irradianza da coordinate geografiche	87
5.2.3	Determinazione della potenza di generazione installata	90
6	Inserimento del Power-to-Gas nella rete	97
6.1	Modellizzazione del P2G tramite codice Matlab®	97
6.2	Inserimento del P2G nella rete Test	100
6.2.1	Presenza di un singolo impianto P2G	100
6.2.2	Presenza di più impianti P2G	105
7	Conclusioni e applicazioni future	112
7.1	Risultati ottenuti	112
7.2	Sviluppi futuri	113
A	Codici Matlab®	114
A.1	Codice: Main_caricamento_profili	114
A.2	Codice: Main_generazione_profili_test	119
A.3	Codice: function_ricostruzione_profili_interpolazione	122
A.4	Codice: function_ricostruzione_profili_random	133
A.5	Codice: Function_energia	138
A.6	Codice: Main_creazione_aggregato_residenziale	141
A.7	Codice: Function_estrazione_profili_non_residenziali	145
A.8	Codice: Main_valutazione_aggregati_nodi	149
A.8.1	Codice: Main_valutazione_aggregati_nodi	150
A.8.2	Codice: Main_random_percentuali_profili	153
A.8.3	Codice: Main_Fase_finale_programma	157
A.8.4	Codice: Main_indici_rete_completo	162
A.9	Codice: Main_ProfiliPV	183
A.9.1	Codice: Main_inserimento_PV	185
A.9.2	Codice: function_backward_forward_sweep	202
A.10	Codice: Main_inserimento_P2G	214
	Bibliografia	222

Abstract

Il lavoro di tesi qui presentato rientra nell'ambito del progetto europeo "PLANET" e si focalizza sulla valutazione dell'impatto di fonti di energia rinnovabile all'interno di una rete elettrica esistente, accoppiata all'uso ottimizzato del Power-to-Gas (P2G).

Il funzionamento della rete è stato simulato inizialmente analizzando dei profili di carico di tipo diverso. Tali profili sono stati ricostruiti in presenza di dati mancanti ed associati ai nodi della rete dopo aver effettuato una ricerca tramite le planimetrie fornite. Questo al fine di ricreare un comportamento complessivo per la rete che fosse il più veritiero possibile.

In parallelo a questo sono stati inseriti i profili relativi alla generazione da fotovoltaico, ricreati tramite un particolare codice Matlab[®] e inseriti nella rete, al fine di valutarne il comportamento in presenza di generazione da fonte rinnovabile.

Infine, è stato inserito il sistema P2G e la rete è stata risolta utilizzando algoritmi di tipo iterativo, in particolare il Backward Forward Sweep (BFS). È stato analizzato il comportamento della rete complessiva sfruttando degli opportuni indici di rete e valutando gli andamenti delle grandezze di maggior interesse, quali tensioni, livello di carico e perdite allocate.

Questo ha permesso di valutare i benefici dell'integrazione del P2G nel complesso della rete di distribuzione.

Abstract

This thesis has been developed within the framework of the European project "PLANET". It focuses on the evaluation of the Renewable Energy Sources (RES) impact on a urban distribution system, and consequently management of the issues they can provoke through Power-to-Gas (P2G) technology.

The loads in the grid were initially simulated by analyzing residential and non-residential load profiles. These profiles were reconstructed when some data were missing and connected to network nodes.

Moreover, the photovoltaics profiles have been inserted in the grid. They have been created through a Matlab[®] script, in order to evaluate the issues they could create.

Finally, the P2G system was added to the grid and the network was resolved using iterative algorithms, in particular the Backward Forward Sweep (BFS). The behavior of the overall network was analyzed using appropriate network indices and the trends of voltages, load levels and allocated losses.

This made possible to evaluate the benefits of integrating Power-to-Gas into the entire distribution network.

Elenco delle figure

1.1	Variazione della concentrazione di CO ₂ , Fonte: [1]	2
1.2	Temperatura terrestre e irradianza solare dal 1880 ad oggi, Fonte: [2]	2
1.3	Obiettivi principali PNIEC per le Fonti Energia Rinnovabile (FER), Fonte: [3]	4
1.4	Set di azioni definite da Terna, Fonte: [4]	5
1.5	Scenario CEN, Fonte: [5]	7
1.6	Scenario DEC, Fonte: [5]	8
1.7	Logo del progetto PLANET, Fonte: [6]	9
1.8	Partner del progetto PLANET, Fonte: [6]	9
1.9	Attività del progetto PLANET, Fonte: [6]	10
1.10	Flow-chart globale del lavoro di tesi	13
2.1	Schema di un impianto Power-to-Gas, Fonte: [7]	14
2.2	Metodi di cattura CO ₂ , Fonte: [8]	19
3.1	Profilo di carico n° 10	23
3.2	Dettaglio profili di carico su <i>Matlab</i> [®]	24
3.3	Punto fuori scala all'interno de profilo di carico n° 28	27
3.4	Profilo di carico n° 115, profilo test	30
3.5	Profilo test con intervalli di dati mancanti	31
3.6	Punti ricostruiti per interpolazione lineare	33
3.7	Ricostruzione di un buco di dati con $A_{int} = 17$ e $\psi_p = 2$	33
3.8	Ricostruzione di un buco di dati con $A_{int} = 17$ e $\psi_p = 3$	34
3.9	Ricostruzione di un buco di dati con $A_{int} = 17$ e $\psi_p = 4$	34
3.10	Flow-chart ricostruzione interpolazione lineare	35
3.11	Intervalli di dati ricostruiti con metodo dei punti casuali	37
3.12	Flow-chart ricostruzione per punti casuali	38
3.13	Profilo di carico (buchi di dati ricostruiti con diverse procedure)	39
3.14	Cumulative profilo test e punti ricostruiti	40
3.15	Differenza cumulative punti ricostruiti - punti profilo test	40
3.16	Profilo test <i>caso 1</i> a) e profilo test <i>caso 2</i> b)	43
3.17	Aggregato profili residenziali basso consumo	46
3.18	Aggregato profili residenziali alto consumo	46
3.19	Aggregato profili residenziali misto	47
3.20	Aggregato profili residenziali misto	47
3.21	Dettaglio foglio excel profili non residenziali (Utenti commerciali)	48
3.22	Profilo commerciale, utente 1, potenza attiva pu a), potenza reattiva pu (tan φ) b)	51

3.23	Profilo commerciale, utente 2, potenza attiva pu a), potenza reattiva pu (tan φ) b)	52
3.24	Profilo commerciale, utente 3, potenza attiva pu a), potenza reattiva pu (tan φ) b)	52
3.25	Profilo commerciale, utente 4, potenza attiva pu a), potenza reattiva pu (tan φ) b)	53
3.26	Profilo commerciale, utente 5, potenza attiva pu a), potenza reattiva pu (tan φ) b)	53
3.27	Profilo industriale, utente 1, potenza attiva pu a), potenza reattiva pu (tan φ) b)	54
3.28	Profilo industriale, utente 2, potenza attiva pu a), potenza reattiva pu (tan φ) b)	54
3.29	Profilo industriale, utente 3, potenza attiva pu a), potenza reattiva pu (tan φ) b)	55
3.30	Profilo Industriale, utente 4, potenza attiva pu a), potenza reattiva pu (tan φ) b)	55
3.31	Profilo industriale, utente 5, potenza attiva pu a), potenza reattiva pu (tan φ) b)	56
3.32	Profilo generico uffici, utente 1, potenza attiva pu a), potenza reattiva pu (tan φ) b)	56
3.33	Profilo generico uffici, utente 2, potenza attiva pu a), potenza reattiva pu (tan φ) b)	57
3.34	Profilo generico uffici, utente 3, potenza attiva pu a), potenza reattiva pu (tan φ) b)	57
3.35	Profilo generico uffici, utente 4, potenza attiva pu a), potenza reattiva pu (tan φ) b)	58
3.36	Profilo generico uffici, utente 5, potenza attiva pu a), potenza reattiva pu (tan φ) b)	58
4.1	Rete Test, Fonte: [9]	60
4.2	Percentuali residenziale (RES), commerciale (COMM), industriale (IND), generico uffici (UFF)	65
4.3	Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Fiat	67
4.4	Modello della linea	71
4.5	Andamento tensioni linea Fiat	72
4.6	Livello di carico linea Fiat	73
4.7	Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Magra	74
4.8	Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Brenta	76
4.9	Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Grosso	78
4.10	Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Chieri	81
5.1	Copertura nuvole a) e irradianza b), singola giornata, Torino	87
5.2	Disposizione spaziale impianti fotovoltaici, Torino	88
5.3	Dettaglio profili di irradianza Torino (giornate dall'11 al 17 gennaio)	88
5.4	Profili di potenza in p.u.	90
5.5	Flow-chart procedimento inserimento PV rete test	92
5.6	Livello di carico nodo 204874	93
5.7	Perdite allocate nodo 204874	93

6.1	Schema impianto Power-to-Gas [10]	97
6.2	Potenza attiva a) e kmol di H ₂ b)	100
6.3	Versi reali corrente (in rosso) e versi ottenuti dal calcolo (in verde)	101
6.4	Tensioni nei nodi della rete test	103
6.5	Correnti nei rami della rete test	103
6.6	Perdite allocate nei nodi della rete Test	104
6.7	Livello di carico rete Test	104
6.8	Tensioni della rete test (presenza di quattro impianti P2G)	107
6.9	Correnti nei rami della rete test (presenza di quattro impianti P2G)	107
6.10	Perdite allocate(presenza di quattro impianti P2G)	108
6.11	livello di carico (presenza di quattro impianti P2G)	108
6.12	Andamento tensioni (presenza di tre impianti P2G)	110

Elenco delle tabelle

2.1	Caratteristiche elettrolizzatori alcalini, Fonte: [7]	16
2.2	Caratteristiche elettrolizzatori PEM, Fonte: [7]	17
3.1	Consumo medio famiglia per numeri di componenti (su base dati ARERA) [11]	26
3.2	Profili ad alto consumo	28
3.3	Profili a basso consumo	29
3.4	Intervallo punti mancanti Profilo di carico n° 51	41
3.5	Norma e rapporto nei due casi considerati	43
3.6	Caratteristiche profili utenti commerciali	49
3.7	Caratteristiche profili utenti industriali	50
3.8	Caratteristiche profili utenti generico uffici	50
4.1	Parametri linea Fiat	61
4.2	Coordinate e stato aree servite dalla dorsale Fiat	62
4.3	Potenze installate nodi rete Fiat	65
4.4	Parametri linea Fiat	69
4.5	Coordinate e stato aree dorsale Magra	74
4.6	Coordinate e stato aree dorsale Brenta	75
4.7	Coordinate e stato aree dorsale Grosso	77
4.8	Coordinate e stato aree dorsale Chieri	79
5.1	Coordinate città di Torino	87
5.2	Indici di rete prima e dopo l'inserimento dei profili di fotovoltaico	95
6.1	Indici di rete (valutazione su singola giornata)	102
6.2	Indici di rete (valutazione su un anno)	105
6.3	Indici di rete, presenza di più impianti P2G (singola giornata)	106
6.4	Indici di rete, presenza di più impianti P2G (valutazione su un anno)	109
6.5	Valore massimo di tensione al variare del numero di impianti P2G	111

-

Lista dei Simboli

$\mathbf{i}_b$	Livello di carico	$E_{\text{persa},\%}$	Energia persa percentuale
$\mathbf{e}_s$	Tensione associata allo slack	E_{persa}	Energia persa
$\mathbf{i}_s$	Corrente associata allo slack	I_b	Corrente del ramo b
$\mathbf{i}_b$	Vettore delle correnti di linea	$i_{\text{max},b}$	Valore limite di corrente della linea b
$\mathbf{i}_s^{(k)}$	Vettore delle correnti nodali	K	Numero nodi della rete
$\mathbf{p}_{\text{Nodo}}^{\text{TOT}}$	profilo di potenza nodale totale	k	Indice dei nodi della rete
$\mathbf{p}_{\text{loss}}$	Perdite	$L_{\text{agg,max}}$	Lunghezza aggregato
$\mathbf{S}_{\text{car,pu}}$	Matrice potenza apparente dei carichi	$L_{M,U}$	Limite punti mancanti nei profili di carico
$\mathbf{Y}_{\text{sh}}$	Matrice delle ammettenze	L_k	Perdite allocate
$\mathbf{Z}_{\text{branch}}$	Matrice delle impedenze di linea	L_p	Lunghezza profilo di carico
ϵ	Tolleranza BFS	$\lim_p$	Limite imperturbato profili di carico
ψ_p	Numero di punti per la ricostruzione di un intervallo mancante unitario	M_E	Massimo della media dei profili ad altro consumo
τ	Numero di intervalli di tempo espresso in ore	$Max_{\text{porz,agg}}$	Massima porzione distinta degli aggregati
τ	Numero di intervalli di tempo	n_{agg}°	Numero di aggregati voluto
φ	Fattore di potenza	$n_{p,\text{ti}}^{\circ}$	Numero punti ricostruzione intervallo mancante di ampiezza generica
$\xi_{E,\text{max}}^{\text{pv}}$	Rapporto massimo energie	n_b°	Numero totale intervalli mancanti
A_{int}	Ampiezza intervallo dati mancante	N_p	Numero di profili che compongono l'aggregato
B	Numero rami della rete	PAC	Profilo di potenza lato rete
b	Indice dei rami della rete	P_{nom}	Potenza nominale
CF	Capacity Factor	$P^{(\text{tipo})}$	Potenza attiva per tipologia
E_{carichi}	Energia carichi passivi rete	$P_{\text{acq}}^{(\text{tipo})}$	Potenza attiva acquistata per tipologia
E_{FER}	Energia Fonti di Energia Rinnovabile		

$P_{pu}^{(tipo)}$	Valore in per unità della potenza attiva dei profili di carico per tipologia	$Q_{pu}^{(tipo)}$	Valore in per unità della potenza reattiva dei profili di carico per tipologia
$p_{M,l}^{af}$	Massimo locale dopo l'intervallo mancante	R_b	Resistenza del ramo b considerato
$p_{m,l}^{af}$	Minimo locale dopo l'intervallo mancante	R_t	Resistenza trasformatore
$p_{M,l}^{bf}$	Massimo locale prima dell'intervallo mancante	RPF	Reverse Power Flow
$p_{m,l}^{bf}$	Minimo locale prima dell'intervallo mancante	RPF_{min}	Tempo di Reverse Power Flow
$P_{inst,max}$	Potenza fotovoltaica massima installabile	Sc	Indice di auto-consumo della rete(Grid self-consumption)
$p_{M,g}$	Massimo globale all'interno degli intervalli utili alla ricostruzione	Ss	Indice di auto-sufficienza della rete(Grid self-sufficiency)
$p_{m,g}$	Minimo globale all'interno degli intervalli utili alla ricostruzione	S_b	Potenza apparente base
p_M	Limite superiore punti dei profili di carico	$t_{m,f}$	Istante di tempo ultimo punto mancante di un intervallo
p_m	Limite inferiore punti dei profili di carico	$t_{m,i}$	Istante di tempo primo punto mancante di un intervallo
p_{start}^{af}	Primo punto dell'intervallo di dati validi dopo la zona di dati mancati	V_b	Tensione base
p_{start}^{bf}	Punto del profilo di carico da cui iniziare la ricostruzione prima dell'intervallo di dati mancanti	X_t	Rattanza trasformatore
p_{stop}^{af}	Ultimo punto dell'intervallo utile per la ricostruzione dopo la zona mancante	Y_t	Ammetenza trasversale
p_{stop}^{bf}	Punto finale dell'intervallo di dati validi prima dell'intervallo da ricostruire	Z_t	Impedenza longitudinale trasformatore
$p_{r,t'}$	Generici punti casuali intervallo ricostruito	$\curvearrowright_{max}$	Numero massimo di iterazioni
$Q^{(tipo)}$	Potenza reattiva per tipologia	$\curvearrowright_I$	Iterazione attuale
$Q_{acq}^{(tipo)}$	Potenza reattiva acquistata per tipologia	α_{th}	Coefficiente di perdita legato alla temperatura
		η_{DC-AC}	Efficienza sistema di conversione
		η_{th}	Efficienza termica
		G	Irradianza
		$NOCT$	Temperatura operativa nominale delle celle fotovoltaiche
		T_a	Temperatura dell'aria
		T_c	Temperatura stimata del pannello fotovoltaico
		L	Matrice delle incidenze

Γ	Inversa della matrice delle incidenze	H_2	Idrogeno elementare
CH_4	Metano	H_2O	Acqua
CO	Monossido di carbonio	O_2	Ossigeno elementare
CO_2	Anidride carbonica	ω	Pulsazione

Acronimi

ARERA Autorità di Regolazione per Energia Reti e Ambiente.	NASA National Aeronautics and Space Administration.
BAU Business as usual.	P2G Power-to-Gas.
BFS Backward Forward Sweep.	P2H Power-to-Heat.
CEN Centralized.	PCMTS Stoccaggio con materiale a cambiamento di fase.
CF Capacity Factor.	PNIEC Piano Nazionale Integrato per l'Energia e il Clima.
CHP Calore ed energia combinato.	PVGIS Photovoltaic Geographical Information System.
DDS Documento di Descrizione degli Scenari.	RPF Reverse Power Flow.
DEC Decentralized.	SDGs Obiettivi di sviluppo sostenibile.
DSO Distribution System Operator.	TTS Stoccaggio di tipo termico tradizionale.
ENTSO-E European Network of Transmission System Operators.	TYNDP Ten Year Network Development Plan.
FER Fonti Energia Rinnovabile.	UNFCCC (COP 21) XXI Conferenza delle Nazioni Unite sui cambiamenti climatici.
GD Generazione Distribuita.	VES Virtual Energy Storage.
GNS Gas Naturale Sintetico.	
IPCC Gruppo intergovernativo sul cambiamento climatico.	

Capitolo 1

Decarbonizzazione del sistema energetico

In questo capitolo vengono riassunti gli eventi preliminari che hanno portato alla creazione degli obiettivi di decarbonizzazione del sistema energetico entro il 2030. Dopo la descrizione della causa principale che ha portato a definire tali obiettivi (ovvero il riscaldamento globale), si presenteranno i piani del panorama italiano e gli scenari che, se implementati, permetteranno il loro raggiungimento mediante una sinergia tra elettricità e gas.

1.1 Riscaldamento globale

Con il termine riscaldamento globale (*Global warming*) si intende l'incremento delle temperature medie della superficie terrestre non correlato a cause naturali.

Tale fenomeno ha radici molto lontane. In particolare, nel 1824 Joseph Fourier [12] conia il termine "*effetto serra*", che indica un particolare fenomeno di regolazione della temperatura del pianeta che consiste nell'accumulo, all'interno dell'atmosfera, di una parte dell'energia termica proveniente dal Sole, per effetto della presenza di alcuni gas.

Successivamente, nel 1859 John Tyndall scopre che cambiamenti nella concentrazione dei gas atmosferici, in particolare dell'anidride carbonica (CO_2), possono causare cambiamenti climatici. Ma è solo nel 1990 che l'IPCC, ovvero il Gruppo intergovernativo sul cambiamento climatico (*Intergovernmental Panel on Climate Change*) presenta il suo primo rapporto, rivelando come l'anidride carbonica contribuisca ad aumentare l'effetto serra e come

le attività umane abbiano contribuito ad aumentare la concentrazione di gas serra all'interno dell'atmosfera [13].

La Figura 1.1, pubblicata dalla National Aeronautics and Space Administration (NASA) [1], mostra la variazione di anidride carbonica in parti per milione, confrontando i campioni atmosferici contenuti nei nuclei di ghiaccio e quelli ottenuti con misurazioni più recenti e fornisce la prova che la CO₂ atmosferica è aumentata dopo la Rivoluzione industriale.

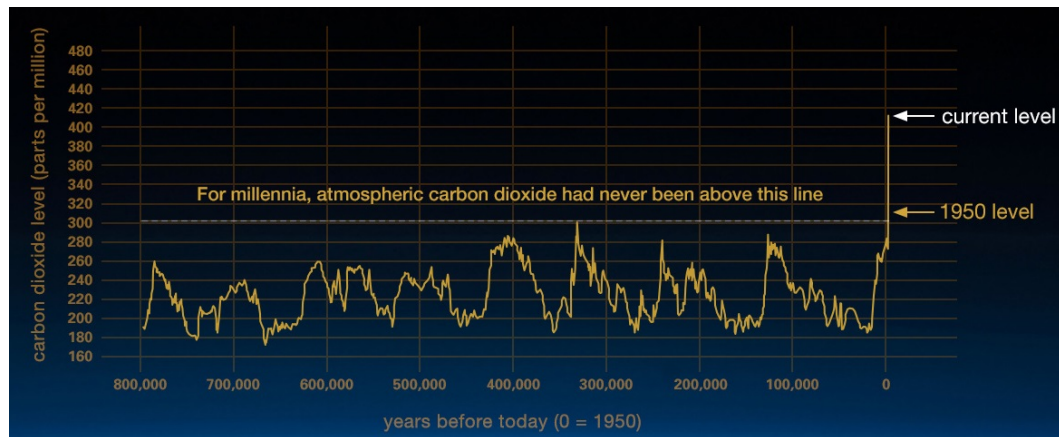


Figura 1.1: Variazione della concentrazione di CO₂, Fonte: [1]

Un'ulteriore prova del riscaldamento globale può essere riscontrata analizzando i dati riportati dalla NASA riguardanti la temperatura superficiale e l'irradianza solare, come mostrato in Figura 1.2.

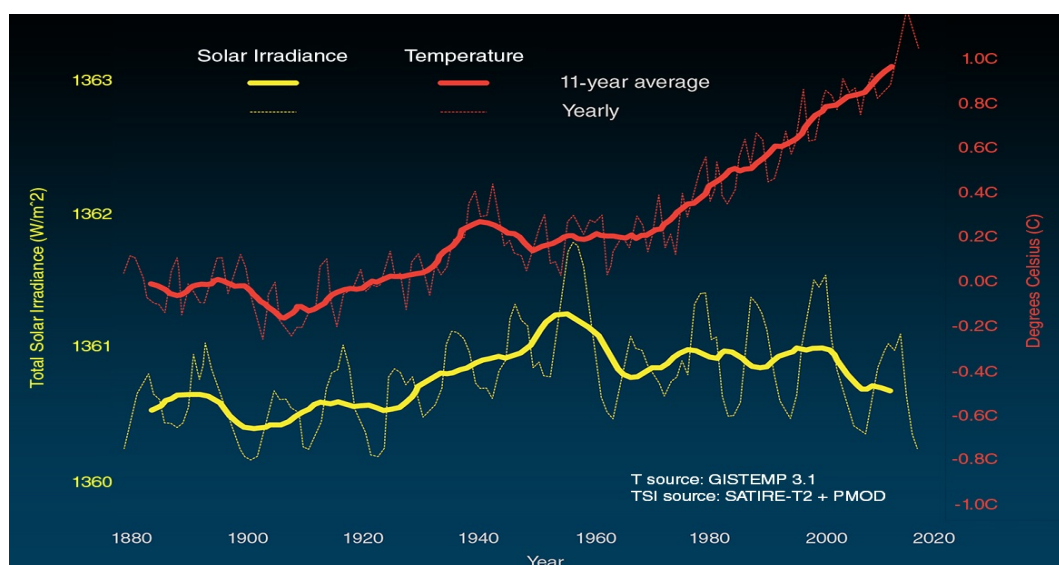


Figura 1.2: Temperatura terrestre e irradianza solare dal 1880 ad oggi, Fonte: [2]

Nella Figura 1.2 vengono confrontati i cambiamenti globali della temperatura superficiale e l'energia del Sole ricevuta dalla Terra in $[W/m^2]$ dal 1880. Le linee più sottili mostrano i livelli annuali, mentre le altre indicano i valori medi con una base di 11 anni, al fine di ridurre il rumore negli insiemi di dati.

Si può notare che la quantità di energia solare ricevuta dalla Terra segue il suo ciclo naturale fatto di incrementi e decrementi, senza avere un aumento netto dagli anni '50 del secolo scorso. Tuttavia, nello stesso periodo la temperatura globale è aumentata notevolmente. Si può dedurre quindi che è improbabile che il riscaldamento terrestre e le temperature osservate nell'ultimo mezzo secolo siano un effetto naturale causato dall'irradianza solare.

1.2 Necessità di decarbonizzazione del sistema energetico

Per tutti i motivi visti nel paragrafo 1.1 risulta necessario decarbonizzare il sistema energetico. Decarbonizzazione vuol dire passare da fonti di energia tradizionale, quindi a combustibili fossili, a fonti di energia rinnovabile, come il fotovoltaico o l'eolico.

Un paese viene definito decarbonizzato quando la quota di energia prodotta da fonti rinnovabili e da gas naturale supera la quantità di energia prodotta tramite fonti fossili, andando a ridurre considerevolmente la quota di emissioni di CO_2 [14].

La transizione verso un sistema decarbonizzato, però, necessita innanzitutto della piena integrazione degli impianti di energia rinnovabile all'interno del sistema energetico in generale ed elettrico in particolare. Questo porterebbe ad agire sull'intera filiera al fine di ottenere un sistema più sostenibile che permetta di raggiungere gli obiettivi proposti.

L'orientamento verso fonti di energia rinnovabile venne definito nel corso della XXI Conferenza delle Nazioni Unite sui cambiamenti climatici (UNFCCC (*COP 21*)), dove la sigla sta per "*United Nations Framework Convention on Climate Change*", tenutasi a Parigi nel dicembre 2015 [15].

In occasione di questo meeting vennero discussi quelli che prendono il nome di "*Accordi di Parigi*", spingendo le Nazioni Unite a riconoscere che l'utilizzo di fonti di energia rinnovabile fossero in grado di contenere il riscaldamento climatico. Per la prima volta furono

messi in agenda ben 17 obiettivi da parte delle nazioni partecipanti, con scadenza nel 2030, chiamati "*Sustainable Development Goals*", Obiettivi di sviluppo sostenibile (SDGs) [15].

L'Unione europea ha preso l'impegno di ridurre le emissioni energetiche almeno del 40% entro il 2030, in questo stesso termine la quota di energia rinnovabile sul totale dei consumi energetici dovrebbe essere di almeno il 27%, tutti questi termini sono contenuti nel pacchetto "Energia pulita", (*Clean Energy Package*), approvato a fine 2016 dalla Commissione europea, al fine di rafforzare l'efficienza energetica degli stati membri e garantire l'approvvigionamento elettrico [16].

In Italia all'inizio del 2019 viene presentato il "Piano Nazionale Integrato per l'Energia e il Clima", PNIEC [3] e il Regolamento 2018/1999, entrambi improntati sull'evoluzione del settore energetico e sulle politiche di sviluppo del sistema energetico nazionale. Questo piano aggiorna gli obiettivi della Strategia Energetica Nazionale (*SEN*), fissando per il 2030 la percentuale dei consumi finali lordi coperti da fonti rinnovabili ad almeno il 30% e prevedendo una riduzione dei consumi di energia primaria del 43%. Mentre sul fronte delle emissioni si punta ad una riduzione dei gas serra, in tutti i settori, di almeno il 33%.

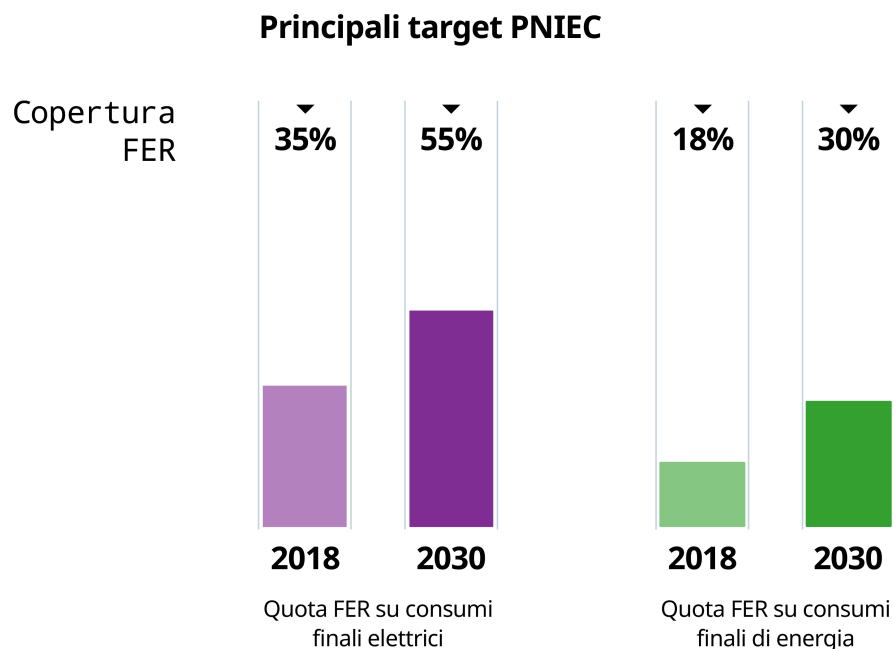


Figura 1.3: Obiettivi principali PNIEC per le Fonti Energia Rinnovabile (FER), Fonte: [3]

Il profilo geografico italiano e quindi le limitate interconnessioni con il resto del continente, nonché i vincoli legati al transito di energia tra il nord e il sud del Paese (isole comprese),

rendono la gestione del sistema elettrico nelle nuove condizioni più difficoltosa e gli obiettivi preposti ancora più ambiziosi. Per Terna, il gestore della rete per la trasmissione dell'energia elettrica italiana, l'integrazione delle fonti rinnovabili nel sistema elettrico è perseguibile solo tramite la realizzazione di alcune azioni, coordinate e coerenti tra loro. Queste azioni vengono individuate in un set ben preciso e sono riconducibili a quattro categorie [4]:

- investimenti nella rete di trasmissione e nelle interconnessioni con l'estero, necessari proprio a perseguire gli obiettivi di decarbonizzazione;
- introduzione di segnali di prezzo di lungo periodo, per stimolare gli investimenti in nuovi settori, fonti rinnovabili o sistemi di accumulo per esempio;
- evoluzione e integrazione dei mercati, per incentivare nuovi servizi necessari nel nuovo contesto e incentivare l'integrazione a livello europeo;
- investimenti in digitalizzazione e innovazione, permettendo la gestione di un sistema elettrico sempre più complesso e annidato.

I punti qui appena presentati vengono descritti in forma più estesa nella Figura 1.4.

1 INVESTIMENTI DI RETE	• Potenziamento dorsali Nord-Sud e rinforzi di rete Sud e isole • Investimenti per regolazione tensione e aumento inerzia del sistema • Interconnessioni con l'estero • Interventi per resilienza
2 SEGNALI DI PREZZO DI LUNGO TERMINE	• Capacity market per promuovere investimenti in impianti termoelettrici di nuova generazione • Aste e contratti di acquisto di energia a lungo termine (PPA) per impianti rinnovabili • Contrattualizzazione a termine tramite procedure competitive per nuova capacità di accumulo, anche idroelettrico
3 EVOLUZIONE E INTEGRAZIONE DEI MERCATI	• Evoluzione della struttura e dei prodotti negoziati sul mercato dei servizi per far fronte alle nuove esigenze (regolazione di tensione, inerzia...) • Partecipazione di "nuove" risorse di flessibilità al mercato dei servizi di dispacciamento: domanda, generazione distribuita, accumuli • Integrazione progressiva con i mercati dei servizi europei
4 INNOVAZIONE E DIGITALIZZAZIONE	• Digitalizzazione della rete di trasmissione (asset e processi) e della gestione del sistema elettrico

Figura 1.4: Set di azioni definite da Terna, Fonte: [4]

Questo ha fatto sì che la diffusione delle risorse ad energia rinnovabile all'interno del panorama elettrico aumentassero, portando quindi alla necessità di far lavorare in sinergia fonti diverse, al fine di ottenere un panorama energetico più "green", è il caso, per esempio, dell'elettricità e del gas, descritto nel Paragrafo [1.3](#).

1.3 Sinergia tra elettricità e gas

Quanto fin qui esposto ha portato a cercare sinergie fra fonti diverse, al fine di ottenere degli scenari attuabili che porteranno ad una decarbonizzazione spinta del sistema energetico.

Attraverso gli scenari che sono stati proposti sarà possibile ottenere, entro il 2030, i vincoli voluti, per poi spingersi al 2040 con vincoli ancora più restrittivi.

Gli scenari futuri sono stati proposti da Terna [\[5\]](#) e si sviluppano partendo da uno scenario base, cioè lo scenario in cui non vengono attuate misure particolari per correggere le azioni normali di mercato. Tale scenario prende il nome di Business as usual (BAU) ed esso porterebbe inevitabilmente a non raggiungere gli obiettivi proposti.

Altri due scenari, chiamati Centralized (CEN) e Decentralized (DEC) permettono invece di raggiungere gli obiettivi del 2030 di decarbonizzazione, quota di fonti di energia rinnovabili ed efficienza energetica e anche le indicazioni, non vincolanti, riguardanti la riduzione delle emissioni di CO₂ nel lungo periodo. Questi scenari sono stati prodotti utilizzando una logica di minimizzazione dei costi di decarbonizzazione e di sviluppi tecnologici alternativi.

Entrambi gli scenari presentano una crescita annua del Pil dell'1.2%, a fronte di un incremento della popolazione di 2.4 milioni di abitanti entro il 2040.

Nello scenario CEN gli obiettivi vengono raggiunti con il contenimento dei consumi, guadagnando in efficienza e non contenendo il fabbisogno. Tutto questo mediante un forte sviluppo nella parte elettrica legata alle energie rinnovabili, sia quelle di tipo non programmabile, quali eolico e fotovoltaico, ma anche sfruttando la disponibilità di quelle programmabili, come il gas naturale, sfruttando anche in certi casi le infrastrutture metanifere esistenti.

Lo scenario DEC invece, ipotizza uno sviluppo maggiore dei sistemi di generazione decen-

tralizzati e quindi lo sviluppo delle rinnovabili non programmabili all'interno del sistema elettrico risulta maggiormente marcato. In entrambi gli scenari è prevista una crescita dei consumi di elettricità, ma il DEC presenta una maggior efficienza complessiva del sistema e una diminuzione moderata dei consumi di gas, stabili invece nello scenario CEN. Le Figure 1.5 e 1.6 riassumono i punti presenti all'interno degli scenari Centralized e Decentralized:



Figura 1.5: Scenario CEN, Fonte: [5]



Figura 1.6: Scenario DEC, Fonte: [5]

Quanto detto è stato elaborato all'interno del Documento di Descrizione degli Scenari (DDS) [17], e le attività sono state eseguite in coerenza con quanto disposto dall'Autorità di Regolazione per Energia Reti e Ambiente (ARERA) con le delibere 654/2017/R/eel [18] e 689/2017/R/gas [19], in analogia al processo europeo che prevede che le associazioni del trasporto elettrico e del gas elaborino gli scenari energetici in maniera congiunta per i rispettivi Piani Sviluppo delle infrastrutture in Europa, inserito all'interno del Ten Year Network Development Plan (TYNDP) dell'European Network of Transmission System Operators (ENTSO-E), la rete europea degli operatori del sistema di trasmissione.

1.4 Progetto PLANET - H2020

Il progetto PLANET [6] è un progetto europeo che si prefigge di facilitare la piena integrazione delle energie di tipo rinnovabile all'interno della rete elettrica, esso racchiude un team composto da 11 partner da 7 Paesi europei, tra di questi, uno dei rappresentanti italiani è proprio il Politecnico di Torino, che svolge il ruolo di coordinatore scientifico.



Figura 1.7: Logo del progetto PLANET, Fonte: [6]



Figura 1.8: Partner del progetto PLANET, Fonte: [6]

Il Politecnico di Torino guida lo sviluppo della tecnologia Power-to-Gas (P2G) all'interno del team di ricerca, svolgendo le attività di validazione del progetto e l'integrazione di modelli di simulazione e di validazione dei sistemi.

Il fondamento su cui si basa l'obiettivo di PLANET è quindi il coordinamento delle reti energetiche, elettricità e gas per esempio, e delle diverse tecnologie per lo stoccaggio e la conversione dell'energia, al fine di consentire un pieno sfruttamento dell'energia prodotta dalle fonti rinnovabili all'interno della rete elettrica [20].

Lo scopo del progetto PLANET sarà poi quello di sviluppare un sistema decisionale utile ai responsabili politici e agli operatori di rete, per supportarli nella scelta di pianificazioni ottimali per la gestione della rete negli scenari energetici futuri, che puntano, come già detto, alla decarbonizzazione completa dei sistemi. Le attività su cui si basa tutto il progetto sono cinque e includono:

1. la modellizzazione delle tecnologie di conversione e di archiviazione che tengano conto dell'impatto reale che avranno sul campo, fra queste tecnologie possiamo annoverare:

- Power-to-Gas (P2G)
- Power-to-Heat (P2H)
- Calore ed energia combinato (CHP)
- Stoccaggio di tipo termico tradizionale (TTS) o Stoccaggio con materiale a cambiamento di fase (PCMTS)
- Virtual Energy Storage (VES)

Tale tesi si concentra sull'uso della tecnologia Power-to-Gas;

2. la simulazione di modelli di reti di elettricità, gas e calore dotate di tecnologie di conversione e stoccaggio, valutandone la stabilità e l'affidabilità.
3. lo sviluppo di un sistema di supporto decisionale (DSS) che consenta la pianificazione della gestione operativa tenendo conto della sinergia fra le varie fonti disponibili
4. la valutazione dell'impatto delle politiche e dei modelli di mercato.
5. un'attività volta alla ricerca di nuovi ruoli e nuovi modelli di business.

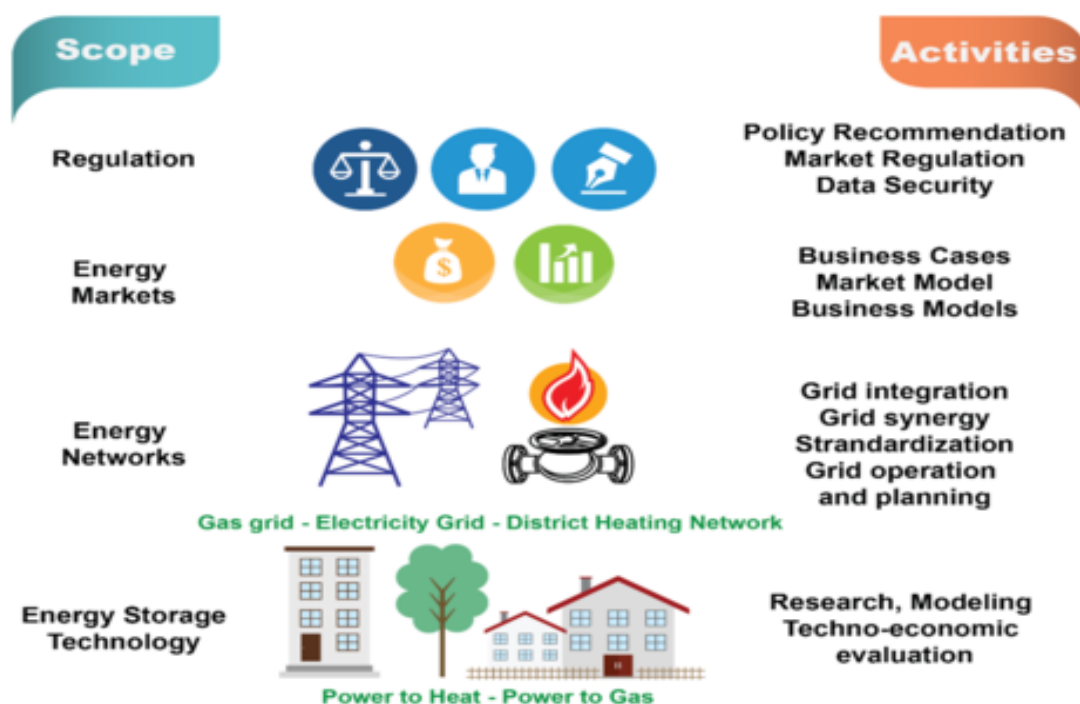


Figura 1.9: Attività del progetto PLANET, Fonte: [6]

1.4.1 Caso studio: uso del P2G per alleviare il flusso di potenza inverso e migliorare gli indici di rete

Il caso studio affrontato in questa tesi si pone sui seguenti obiettivi[21]:

- integrazione P2G per il miglioramento degli indici di rete;
- integrazione del P2G per alleviare il flusso inverso in rete, ovvero il Reverse Power Flow (RPF)

Gli indici di rete rappresentano un metodo efficace per valutare il comportamento della rete elettrica. Mentre la visualizzazione degli andamenti del livello di carico e delle tensioni consente di valutare il comportamento di tali grandezze e la presenza o meno di violazioni dei vincoli, gli indici di rete consentono di comprendere il miglioramento all'interno della rete in seguito all'inserzione di un nuovo carico con determinate caratteristiche.

Gli indici principali sono l'indice di auto-sufficienza e auto-consumo della rete elettrica, definiti nelle equazioni (1.1) ed (1.2).

- Indice auto-sufficienza della rete (Grid self-sufficiency)

$$Ss = \frac{E_{\text{tot,FER}} - RPF}{E_{\text{carichi}}} \cdot 100 \quad (1.1)$$

dove FER sta per Fonti Energia Rinnovabile. Questo indice è indicativo della percentuale di produzione legata a fonti rinnovabili presenti in rete.

- Indice auto-consumo della rete (Grid self-consumption)

$$Sc = \frac{E_{\text{tot,FER}} - RPF}{E_{\text{tot,FER}}} \cdot 100 \quad (1.2)$$

questo indice è il rapporto tra la produzione da fonti di energia rinnovabile usate per alimentare i carichi e la produzione totale. Infine sarà importante anche l'energia persa percentuale definita in (1.3).

- Energia persa percentuale

$$E_{\text{persa},\%} = \frac{\sum \mathbf{P}_{\text{loss}}}{\sum (\text{Re}(S_{\text{carico,pu}}))} \cdot 100 \quad (1.3)$$

dove al numeratore troviamo le perdite totali e al denominatore la potenza attiva dei carichi presenti in rete.

Il Reverse Power Flow è un flusso di potenza che sussiste invece in caso di un esercizio anormale della rete. In questo caso è presente una quantità di energia che fluisce dal sistema di distribuzione al sistema di trasmissione e questa influisce sull'efficacia dei sistemi di protezione della rete di distribuzione.

Questa condizione, anche se non è da considerarsi normale per il sistema, può non implicare alcuna violazione dei vincoli di esercizio, il che significa che tensioni e correnti possono ancora rientrare nell'intervallo ammissibile di valori. Al contrario, invece i casi in cui vengono violati i limiti di corrente, di tensione o entrambi rappresentano condizioni che vanno evitate durante il funzionamento normale. In questi casi il Distribution System Operator (DSO) agisce in modo da evitarle.

La violazione dei vincoli porta la rete a lavorare in condizioni di emergenza, cosa inammissibile.

In questo lavoro di tesi si mira invece a mostrare come l'installazione del sistema Power-to-Gas sia in grado di ridurre la presenza del Reverse Power Flow e migliorare gli indici di rete, aumentando l'*hosting capacity* della rete, cioè la quantità di Generazione Distribuita (GD) che si può connettere al sistema senza la necessità di interventi infrastrutturali.

1.5 Flow-chart riassuntivo dell'intero lavoro di tesi

In Figura 1.10 è stato riportato il flow-chart globale del lavoro di tesi svolto.

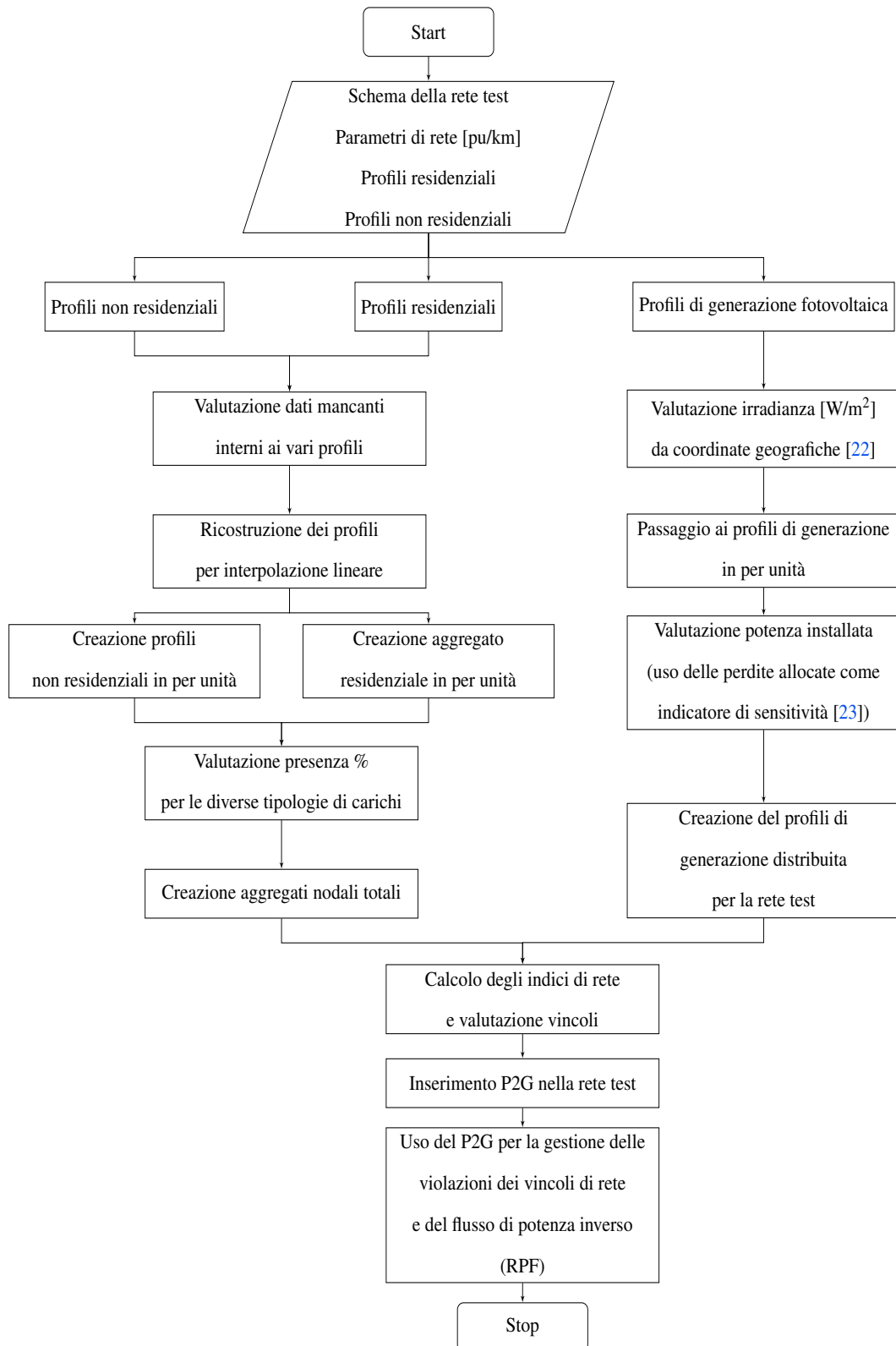


Figura 1.10: Flow-chart globale del lavoro di tesi

Capitolo 2

Il sistema Power-to-Gas

In questo capitolo viene esposto il sistema Power-to-Gas, evidenziando le parti che hanno maggior impatto sul sistema elettrico. Particolare importanza verrà data alla possibilità di catturare la CO_2 , uno dei motivi che renderanno questa tecnologia importante per la futura decarbonizzazione del sistema.

2.1 Componenti del sistema

Il termine Power-to-Gas può indicare dei processi che portano a prodotti finali differenti, per esempio l'Idrogeno, il Gas Naturale Sintetico (GNS) o il Metanolo. La produzione di GNS che è stata considerata in questo lavoro di tesi, prevede un processo detto metanazione. Lo schema semplificato di un tipico impianto P2G è riportato in Figura 2.1:

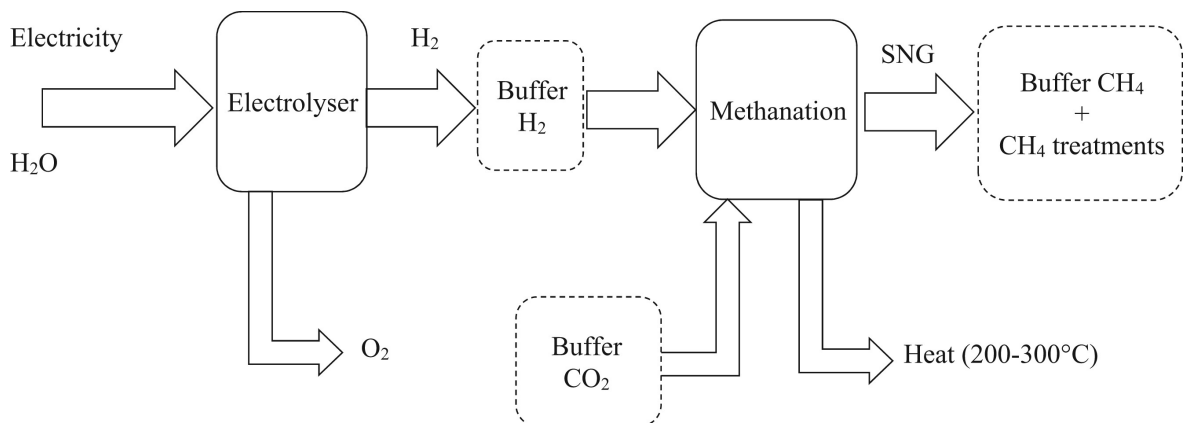
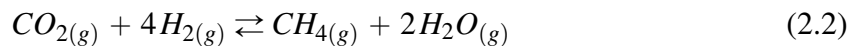
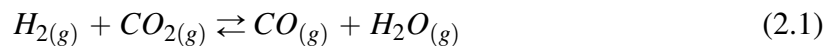


Figura 2.1: Schema di un impianto Power-to-Gas, Fonte: [7]

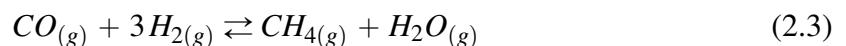
Possiamo quindi distinguere i vari blocchi della catena di produzione del prodotto finale:

- Elettrolizzatore (Electrolyser);
- Stoccaggio (Buffer) H_2 ;
- Stoccaggio (Buffer) CO_2 ;
- Metanazione (Methanation);
- Stoccaggio (Buffer) CH_4 e purificazione (treatments) CH_4 .

In condizioni normali, il sistema P2G necessita di una determinata quantità di potenza elettrica, prelevabile dal sistema elettrico a cui è connesso. Come si vede dalla Figura 2.1 essa rappresenta, ovviamente, uno degli input del sistema, insieme all'acqua e alla CO_2 , mentre i prodotti in uscita saranno GNS e calore. Le reazioni che portano alla produzione di GNS sono le seguenti:



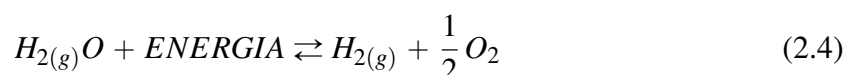
Se è presente una fonte di monossido di carbonio (CO) è possibile ottenere GNS direttamente dalla reazione seguente:



Di seguito saranno brevemente analizzate le parti dell'impianto mostrato in Figura 2.1 che hanno più impatto sul sistema elettrico.

2.1.1 Elettrolizzatore

L'elettrolisi è la produzione di idrogeno dalla dissociazione di H_2O , tale reazione è di tipo endotermico ed è data da:



Gli elettrolizzatori si dividono in base alla tecnologia che utilizzano [24]:

- elettrolizzatori alcalini;
- elettrolizzatori dotati di Membrana Elettrolitica Polimerica (PEM);
- elettrolizzatori all'ossido solido (SOEC).

La tecnologia SOEC è ancora in fase sperimentale e non viene ancora commercializzata.

Quindi le tecnologie che verranno analizzate nel prosieguo saranno le prime due.

1. Elettrolizzatori alcalini

Gli elettrolizzatori di questo tipo sono stati i primi ad essere utilizzati per usi industriali per molti anni. La tecnologia associata ad essi è molto conosciuta. Le caratteristiche principali vengono riportate nella Tabella 2.1.

Tabella 2.1: Caratteristiche elettrolizzatori alcalini, Fonte: [7]

Potenza nominale [MW]	Produzione nominale [Nm ³ /h]	Cosumo energetico [kWh/Nm ³]	Efficienza [%]	Pressione operativa [bar]	Temperatura operativa [°C]	Purezza idrogeno [%]
< 0.055	0.4 ÷ 10	7.5 ÷ 5.4	47.2 ÷ 65.5	1 ÷ 30	65 ÷ 100	99.5 ÷ 99.9
0.055 ÷ 0.25	10 ÷ 43	5.4 ÷ 5	65.5 ÷ 70.8			
0.25 ÷ 0.50	43 ÷ 100	5 ÷ 4.87	70.8 ÷ 72.7			
0.50 ÷ 1.5	100 ÷ 330	4.87 ÷ 4.3	72.7 ÷ 82.3			
1.5 ÷ 3.5	330 ÷ 760	≈ 4.3	≈ 82.3			

Tuttavia, questi elettrolizzatori presentano alcuni svantaggi di utilizzo, ad esempio:

- Gli elettrolizzatori alcalini non sono in grado di funzionare in tutto l'intervallo 0 ÷ 100%, ma presentano un minimo tecnico che va dal 10% al 40% della potenza nominale, quindi se l'ingresso non si trova entro questi limiti è necessario spegnere il sistema.

- Il sistema durante il funzionamento in transitorio non presenta un'elevata reattività (il tempo tipico va dai secondi ai minuti).
- Il tempo di avviamento è elevato: si va da un minimo di 10 minuti fino all'ordine dell'ora, questo dipende dalla purezza del gas.
- A causa della necessità di operazioni di spurgo con azoto, l'elettrolizzatore impiega dai 30 ai 60 minuti per riavviarsi dopo l'arresto.

Il costo di questi impianti è circa di 1000 €/kW e presenta una vita media compresa tra 15 ÷ 30 anni.

2. Elettrolizzatori PEM

La tecnologia PEM per l'elettrolisi dell'acqua è stata sviluppata nel 1966 [7]. Essa si basa sull'uso di una membrana polimerica come elettrolita, mentre come catalizzatori sono usati materiali nobili, quali i metalli del gruppo platino, composto da sei metalli che occupano posizioni contigue nella tavola periodica degli elementi.

I metalli del gruppo del platino sono tutti costosi e questo quindi rappresenta uno svantaggio per la tecnologia, che presenta un costo circa raddoppiato rispetto al caso di elettrolizzatori alcalini, cioè intorno ai 2000 €/kW.

Le caratteristiche principali di questi tipi di elettrolizzatori vengono esposte nella Tabella 2.2:

Tabella 2.2: Caratteristiche elettrolizzatori PEM, Fonte: [7]

Potenza nominale	Produzione nominale	Cosumo energetico	Efficienza	Pressione operativa	Temperatura operativa	Purezza idrogeno
[MW]	[Nm³/h]	[kWh/Nm³]	[%]	[bar]	[°C]	[%]
< 0.012	0.53 ÷ 1.05	6.7	52	10 ÷ 30	20 ÷ 100	>99.9
0.022 ÷ 0.070	2 ÷ 6	7.3 ÷ 6.8	48.5 ÷ 52.1			
0.1 ÷ 0.275	10 ÷ 30	6.2 ÷ 5.8	57.1 ÷ 61.1			

Gli elettrolizzatori di tipo PEM consentono di produrre un idrogeno purissimo e non necessitano di ulteriori apparecchiature di depurazione. Questo, unito ad una produzione di idrogeno sempre più vicina a quella degli elettrolizzatori alcalini, sta spingendo la tecnologia PEM nelle applicazioni.

Inoltre un avvio da freddo in queste apparecchiature può essere completato in pochi minuti. Tuttavia, la durata di vita di questi elettrolizzatori è più bassa rispetto a quella degli alcalini e rientra nell'intervallo di $5 \div 20$ anni.

2.1.2 Produzione di CO₂

Uno degli aspetti positivi nell'utilizzo della tecnologia P2G è la possibilità di utilizzare la CO₂. Si può quindi predisporre l'impianto di un sistema di cattura della CO₂. I principali metodi di cattura sono tre [8]:

- *Pre-combustione*: consiste nella cattura della CO₂ prima della combustione. Il combustibile, carbone o gas, viene trattato e trasformato in gas di sintesi (syngas), i due flussi gassosi vengono poi separati ottenendo un gas ad alta concentrazione di idrogeno per la combustione e un flusso composto da anidride carbonica.
- *Post-combustione*: In questo caso la cattura dell'anidride carbonica avviene alla fine del ciclo, quindi da gas ormai combusti.
- *Ossicombustione*: In questo caso la combustione avviene in presenza di ossigeno puro anziché di aria. Questo fa sì che la concentrazione di CO₂ nella miscela aumenti e renda più facile la cattura dell'anidride carbonica.

I processi appena descritti vengono riportati nella Figura 2.2:

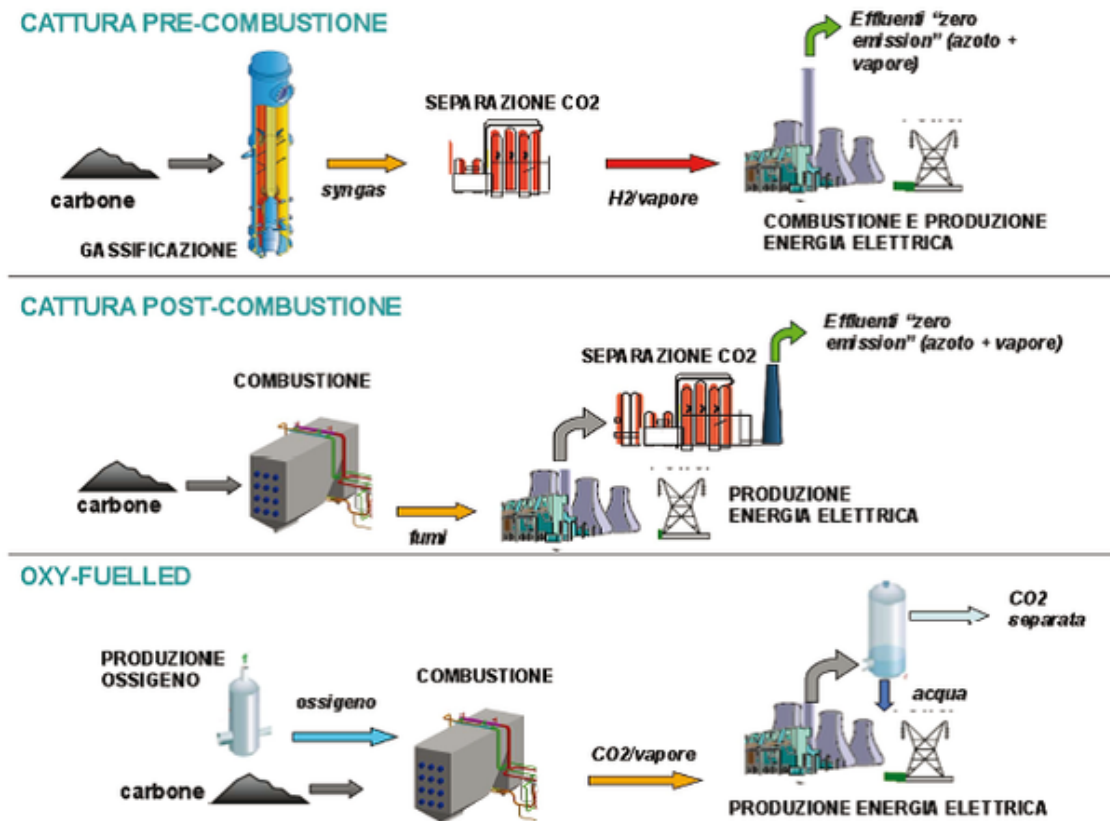


Figura 2.2: Metodi di cattura CO₂, Fonte: [8]

Possiamo poi citare altri due metodi che consentono la cattura dell'anidride carbonica presente nell'ambiente, questi metodi vengono chiamati Carbon Dioxide Removal (CDR) e sono [25]:

- **Rimboschimento:** Questo metodo consente di creare del *Carbon sink*, cioè un deposito di carbone naturale o artificiale che assorbe l'anidride carbonica contribuendo a diminuire la sua concentrazione. Questi depositi possono poi essere utilizzati per generare dei biocombustibili.
- **Cattura atmosferica:** Questa avviene tramite dei processi di filtrazione dell'aria, spesso utilizzando appositi solventi.

Nel caso di un impianto industriale già esistente il metodo migliore per la cattura della CO₂ è dato dalla post-combustione.

Bisogna comunque tenere conto del fatto che qualunque metodo di cattura dell'anidride car-

bonica implica un incremento nella quantità di carburante consumato. Infine l'efficienza di questi sistemi si trova in un range compreso tra 85% ÷ 90%.

2.1.3 Compressione e stoccaggio dell'idrogeno

La compressione e l'immagazzinamento dell'idrogeno [26] rappresenta una fase fondamentale del processo P2G. Nelle applicazioni si punta ad una diminuzione del volume di stoccaggio, per applicazioni mobili. Al giorno d'oggi la principale difficoltà che si trova è relativa alle trasformazioni presenti che sono costose e tecnologicamente complesse.

L'idrogeno presenta una elevata densità di energia per unità di massa, ma scarsa densità energetica volumetrica rispetto agli idrocarburi. Quindi richiede serbatoi di maggiori dimensioni e più pesanti rispetto a quelli degli idrocarburi, a parità di contenuto energetico.

Un aumento nella pressione del gas migliora la densità di energia per unità di volume: questo significa contenitori meno ingombranti, ma egualmente pesanti. Inoltre si avrà la necessità di un compressore, quindi di energia per alimentarlo.

Nel caso di un serbatoio alla pressione di 200 bar il dispendio energetico per passare dalla pressione atmosferica alla pressione del serbatoio è di circa 3.6 kWh/kg_{H₂} [7].

2.2 Osservazioni finali sul P2G

Nel 2016 gli impianti P2G presenti nel mondo erano 14 [27], situati in Germania, Spagna, Inghilterra, Stati Uniti e Canada, ma il numero è destinato a crescere negli anni. Alcuni dei principali motivi che porteranno il numero di sistemi P2G ad aumentare sono stati esposti nei paragrafi precedenti, ma vengono comunque riassunti di seguito e altri vengono aggiunti:

- Possibilità di sfruttare l'energia delle FER, minimizzando i disturbi sulla rete;
- Sfruttamento della rete gas esistente;
- Cattura della CO₂, diminuendone la concentrazione atmosferica;
- Possibilità di generare energia a zero emissioni;

- Spinta nella ricerca nell'ambito delle celle ad elettrolisi, in particolare per quanto riguarda le SOEC, che potranno un giorno soppiantare le PEM;
- Capacità di stoccaggio potenzialmente illimitata grazie alla possibilità di iniezione nella rete del gas.

Capitolo 3

Analisi Dati

In questo capitolo viene illustrata l'analisi dati svolta durante il lavoro di tesi. Tale analisi riguarderà sia profili di carico residenziali che profili riferiti ad utenti non domestici. Tuttavia, la parte preponderante di tale analisi riguarderà i primi, in quanto il metodo di ricostruzione dei profili e lo studio della classe di appartenenza degli stessi è stato sviluppato per i profili residenziali e successivamente applicato anche agli utenti non domestici

3.1 Studio profili di carico residenziali

3.1.1 Identificazione profili di carico

La parte preliminare di questo lavoro di tesi si basa sullo studio dei profili di carico forniti e della loro ricostruzione nel caso di dati mancanti. Sono stati considerati 150 profili di carico, riguardanti una zona compresa nell'area urbana di Torino. Il valore di energia nei vari profili viene fornito in funzione di un'unità di tempo minima pari a 15 minuti. Essendo profili annuali la lunghezza dei dati relativa ad ogni profilo sarà di 35040 intervalli, cioè il numero di quarti d'ora presenti in un anno. In Figura 3.1 è mostrato l'andamento dell'energia di un generico profilo di carico tra quelli disponibili, il profilo scelto è il numero 10.

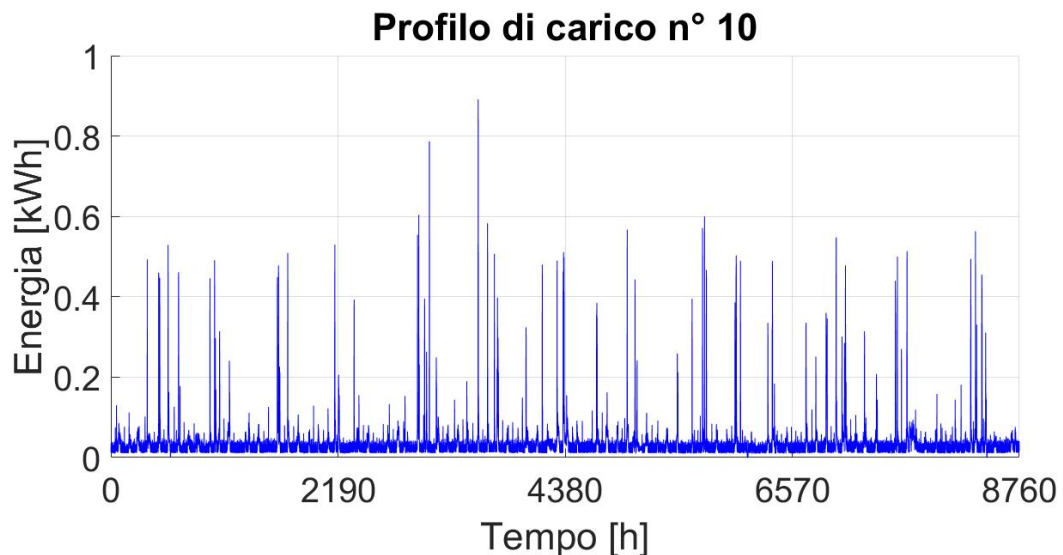


Figura 3.1: Profilo di carico n° 10

Mediante un codice sviluppato in *Matlab*[®], una volta caricati tutti i profili e creata una tabella interna al programma per raccogliere tutti i dati, essi sono stati analizzati. Insieme ai dati numerici relativi ad ogni profilo, erano presenti delle colonne puramente letterali che consentivano di identificare lo stato del dato numerico associato alla medesima posizione. I codici per la codifica presenti si distinguevano in:

- *O*: Per indicare la presenza in quella posizione di un dato valido (Ok);
- *R*: Per indicare un dato che era stato rimpiazzato da un valore valido (Replaced);
- *NA*: Per indicare i dati mancanti (Not-a-Number (NaN) nella codifica *Matlab*[®]);
- *<undefined>*: Queste posizioni rappresentavano dei “buchi” presenti nelle colonne letterali dei dati, verranno discussi successivamente.

In Figura 3.2 è possibile vedere una porzione della matrice associata ai profili di carico e le colonne numeriche e letterali.

	7 VarName7	8 O3	9 VarName9	10 O4	11 VarName11	12 O5	13 VarName13	14 O6	15 VarName15
20445	0.0470 O		0.0440 O		0.0310 O		0.1340 O		0.0080
20446	0.0500 O		0.0430 O		0.0260 O		0.0880 O		0.0070
20447	0.0900 O		0.0410 O		0.0030 O		0.0590 O		0.0080
20448	0.0750 O		0.0300 O		0.0030 O		0.0640 O		0.0080
20449	0.0880 O		0.0140 O		NaN <undefined>		0.0470 O		0.0420
20450	0.0670 O		0.0090 O		NaN <undefined>		0.0450 O		0.0400
20451	0.0650 O		0.0200 O		NaN <undefined>		0.0430 O		0.0270
20452	0.0250 O		0.0460 O		NaN <undefined>		0.0420 O		0.0080
20453	0.0430 O		0.0430 O		NaN <undefined>		0.0420 O		0.0080
20454	0.0420 O		0.0420 O		0.0300 O		0.0420 O		0.0080
20455	0.0700 O		0.0400 O		0.0090 O		0.0410 O		0.0170
20456	0.0660 O		0.0090 O		0.0050 O		0.0410 O		0.0440

Figura 3.2: Dettaglio profili di carico su *Matlab*®

Appurata quindi la struttura della matrice legata ai profili di carico, si è quindi proceduto con l'identificazione di tutte le posizioni, escluse le <undefined>, perché la presenza di queste posizioni è emersa solo successivamente. Sono state create delle nuove matrici contenenti tutte le posizioni associate alla rispettiva codifica. Per far questo è stato utilizzato un semplice codice costituito da cicli *for*, come mostrato nel codice completo [A.1.1](#), righe 74 ÷ 81.

Nel prosieguo dell'analisi ci si è quindi soffermati sulle posizioni legate ai NaN presenti. Per la visualizzazione di questi punti, ad essi è stato associato un valore numerico a piacere, pari a -16: è stato scelto negativo perché, essendo profili di carico, non erano presenti valori minori di zero in altre posizioni. Questa "forzatura" ha consentito nei grafici ottenuti a fine analisi la visualizzazione immediata degli intervalli mancanti e il loro confronto con il profilo "ricostruito".

Ad una prima analisi però non si era a conoscenza della presenza all'interno dei profili di carico dei punti <undefined>: questi ultimi sono stati identificati utilizzando una matrice logica di controllo, ottenuta tramite il confronto tra la matrice delle posizioni associate ai valori mancanti nella matrice numerica dei profili e quella relativa alle posizioni con codifica NA nella matrice logica. Per posizioni coincidenti la matrice logica garantiva in uscita un valore unitario, per posizioni non coincidenti un valore nullo. In questo modo si è riusciti a identificare alcune posizioni, 169 in totale, che rappresentano circa lo 0,0032% dei dati

completi, che presentavano questa caratteristica. A tutte queste posizioni era però associato un valore NaN nella matrice numerica, quindi si è proceduto ad una sostituzione in tali posizioni della sigla NA, come mostrato nel Codice [A.1.1](#), righe 92 ÷ 99.

A questo punto, ottenuta una matrice logica priva di posizioni unitarie, quindi acquisita una conoscenza completa su tutti i profili presenti, si è proceduto ad analizzare i valori numerici presenti alla ricerca di eventuali incongruenze. Procedendo ad una ricerca dei massimi presenti in tutti i profili, al fine di valutare che non ci fossero dati errati si è visto che alcuni profili presentavano all'interno dei set di dati valori troppo alti, legati probabilmente ad un errata misura in quell'istante di tempo. Costruito il vettore contenente i massimi associati ad ogni profilo si è visto che il profilo 98 per esempio presentava un massimo in un intervallo pari a 192 kWh. Tali valori, non filtrati e considerati all'interno del nostro aggregato finale avrebbero comportato un errore abbastanza grossolano. Si è deciso quindi di filtrare tutti i valori associati ai 150 profili ed eseguire una classificazione degli stessi, come verrà spiegato nel Paragrafo [3.1.2](#).

3.1.2 Profili a diverse fasce di consumo e classificazione valori

Partendo dalla conoscenza dei profili ottenuta come descritto nel Paragrafo [3.1.1](#) si è deciso di dividere i profili in due gruppi: in un gruppo sono stati raggruppati i profili a basso consumo, mentre nell'altro quelli ad alto consumo.

Considerando i dati forniti dall'autorità ARERA, una famiglia media italiana ha le seguenti caratteristiche [\[11\]](#):

- composizione di 4 membri;
- 3 kW di potenza utilizzata;
- un consumo di 2700 kWh annui.

È possibile reperire diverse tabelle online che contengono stime del consumo annuo delle famiglie, al variare del numero di persone presenti e/o degli elettrodomestici presenti. Nella Tabella [3.1](#) sono stati riportati i dati relativi a diversi casi:

Tabella 3.1: Consumo medio famiglia per numeri di componenti (su base dati ARERA) [11]

Componenti famiglia	Elettrodomestici	Consumo annuo [kWh/anno]
2 persone	TV, frigo, condizionatore, lavatrice	1400
2 persone	TV, frigo, pc, lavastoviglie lavatrice, condizionatore	2000
2 persone	TV, frigo, pc, lavastoviglie, lavatrice 2 condizionatori, scaldabagno elettrico	2700
4 persone	TV, frigo, pc, lavastoviglie lavatrice, 2 condizionatori	3300
4 persone	2 TV, frigo, 2 pc, lavastoviglie, lavatrice 2 condizionatori, scaldabagno elettrico	3600
5 persone	2 TV, frigo, 2 pc, lavastoviglie lavatrice, 3 condizionatori	5200

Si è scelto quindi di considerare come suddivisione fra i due gruppi il secondo limite relativo ad un nucleo familiare di 4 persone, cioè 3600 kWh/anno. Grazie a questo vincolo è stata quindi creata la funzione “*function_energia*” su *Matlab*[®] e i due gruppi associati ai diversi valori di consumo.

Dei 150 profili di partenza, due sono stati esclusi dall’analisi in quanto formati completamente da zeri, 32 sono stati riconosciuti come profili ad alto consumo e 116 come profili a basso consumo. A partire da questo punto è stato quindi possibile analizzare i profili non più come un blocco unico ma come due blocchi, in modo da ottenere due aggregati, quindi uno riferito ai profili a basso consumo e l’altro a quelli ad alto consumo.

All’interno dei profili erano ancora presenti dei punti fuori scala: questi ultimi sono stati identificati valutando la distanza tra ogni punto appartenente a ciascun profilo e il valore medio (escludendo i NaN) del profilo stesso. Qualora questa distanza fosse stata maggiore di 10

volte il massimo della media dei profili ad alto consumo, ottenuto dalla “*function_energia*”, il punto sarebbe stato ritenuto fuori scala. Essendo tale massimo pari a 0.4356 kWh, il limite per i valori è stato impostato a 4.356 kWh.

La Figura 3.3 mostra uno dei punti fuori scala presenti all’interno del profilo di carico 28.

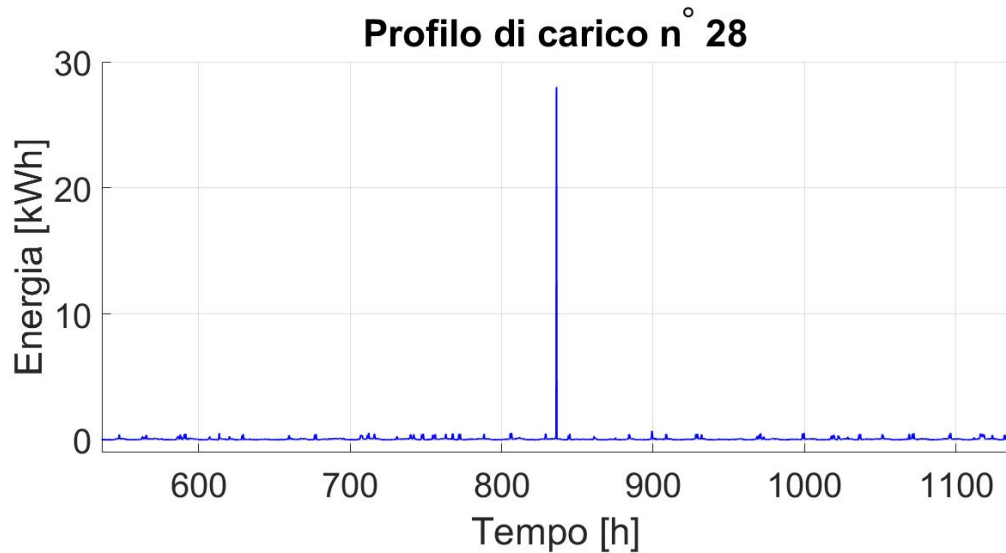


Figura 3.3: Punto fuori scala all’interno de profilo di carico n° 28

Una volta identificati questi punti, essi sono stati trattati come dei NaN. Questo ha comportato la presenza di intervalli di dati mancanti aggiuntivi all’interno dei profili di carico, ma il cui andamento è stato ricostruito, al fine di ottenere un andamento più veritiero del profilo.

Se ciò non fosse stato fatto, la presenza di tali dati avrebbe comportato un possibile errore nel comportamento dell’aggregato di carico finale

3.1.3 Scelta dei profili di carico utilizzabili

La ricostruzione dei profili di carico permette di ottenere un set di dati completamente definito la cui dimensione dipenderà dal parametro scelto.

Tale parametro non può che essere legato all’ampiezza dell’intervallo di dati mancanti ed all’interno dei codici è stato definito come: “*LimitMissingUsers*”. Esso rappresenta il limite massimo sul numero di punti mancanti consecutivi, da adesso in poi definito come $L_{M,U}$ nella trattazione.

I profili vengono ciclizzati considerando diversi valori di $L_{M,U}$; nello specifico si partirà da un valore pari a 10 fino ad un massimo di 17 in quanto, oltre tale valore, l'intervallo è considerato troppo grande per essere ricostruito in modo adatto. Al variare del parametro precedentemente esposto si ottiene quindi un diverso numero di profili disponibili per costruire l'aggregato di carico finale. I risultati ottenuti vengono presentati nelle Tabelle 3.2 e 3.3:

Tabella 3.2: Profili ad alto consumo

N° profili di partenza: 32		
$L_{M,U}$	Profili scartati	Profili mantenuti
10	15	17
11	12	20
12	10	22
13	9	23
14	9	23
15	6	26
16	4	28
17	4	28

Tabella 3.3: Profili a basso consumo

N° profili di partenza: 116		
$L_{M,U}$	Profili scartati	Profili mantenuti
10	96	20
11	82	34
12	68	48
13	60	56
14	58	58
15	49	67
16	40	76
17	33	83

È stato scelto un valore limite sulle posizioni mancanti consecutive pari a 17 in modo da ottenere 28 profili ad alto consumo disponibili sui 32 di partenza e 83 a basso consumo sui 116 d'inizio. Gli andamenti di tali profili potranno essere ricostruiti e i profili completi verranno poi utilizzati per generare l'aggregato di carico finale. Sono stati redatti due metodi di ricostruzione: il primo basato sull'uso dell'interpolazione lineare mentre il secondo sull'utilizzo di punti random generati all'interno di un intervallo definito.

Di questi due metodi si è deciso di utilizzare quello che prevede l'uso dell'interpolazione lineare, in quanto i risultati ottenuti sono migliori rispetto a quelli ottenuti col secondo metodo.

3.1.4 Ricostruzione per interpolazione lineare vs. ricostruzione basata su estrazioni casuali dei valori: analisi e confronto

In questo paragrafo verranno presentati i due metodi che sono stati utilizzati per la ricostruzione dei profili di carico. Come detto, il primo metodo, quello che verrà poi utilizzato, fa uso dell'interpolazione lineare per ricostruire i punti mancanti all'interno dei profili mentre

il secondo ricostruisce i vari punti estraendo dei numeri casuali all'interno di un intervallo di valori definito.

Entrambi i metodi sono stati valutati utilizzando un profilo di carico completo, quindi privo di punti mancanti, al fine di valutare lo scostamento tra l'andamento ricostruito e quello reale del profilo. Il profilo scelto come profilo test è il profilo numero 115, il cui andamento è riportato in Figura 3.4.

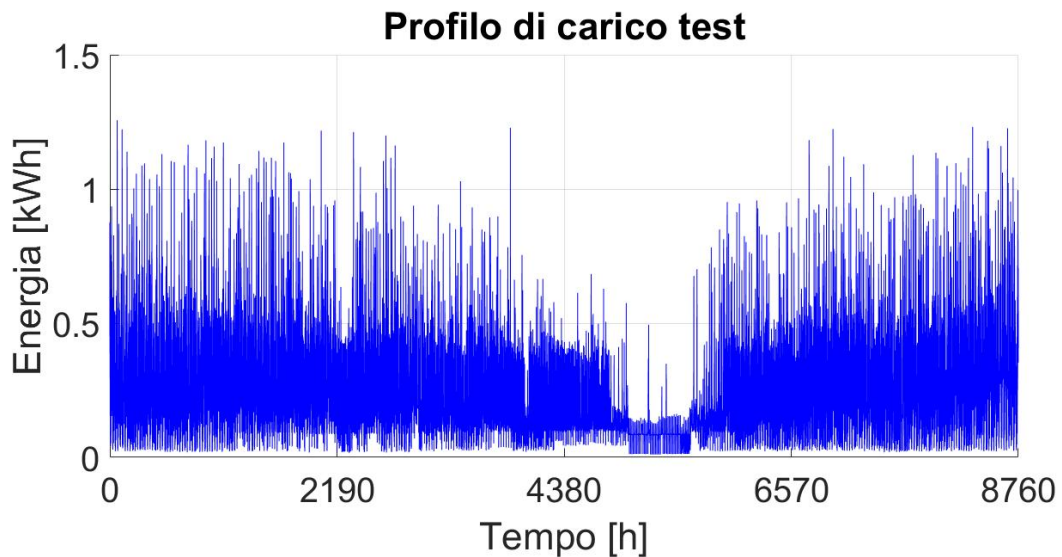


Figura 3.4: Profilo di carico n°115, profilo test

All'interno di questo profilo sono stati inseriti 118 intervalli di dati mancanti di ampiezza $1 \div 3$, generati in posizioni random. Entrambi i metodi di ricostruzione utilizzano un intervallo di punti prima e dopo la zona di dati mancanti per effettuare la ricostruzione. Quindi, un numero più elevato di punti mancanti renderebbe impossibile la ricostruzione del profilo, in quanto potrebbero essere considerati all'interno dell'intervallo di ricostruzione dei punti non validi.

Ai punti mancanti è stato associato il valore -16. Il profilo test con questi dati mancanti è riportato in Figura 3.5.

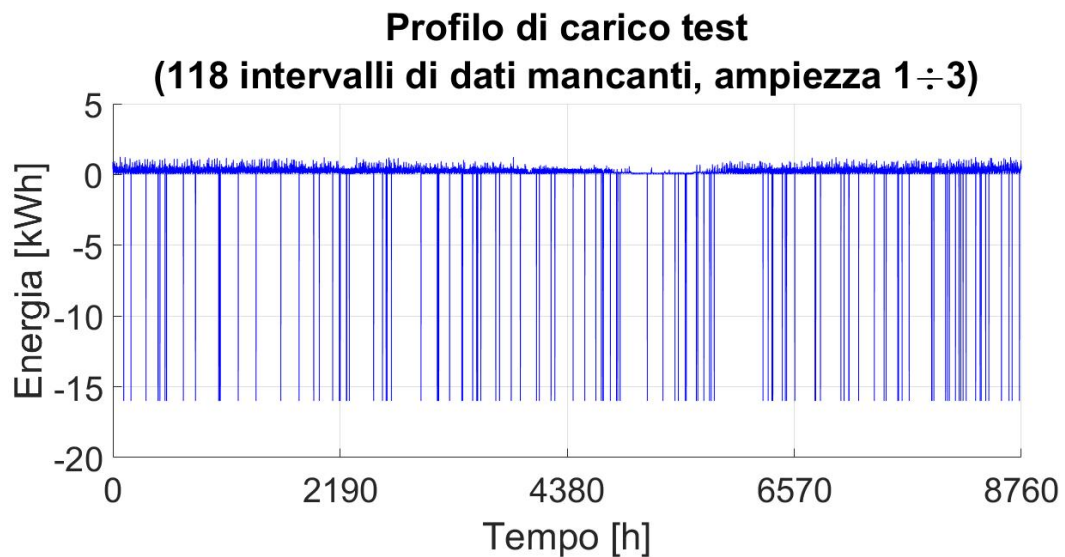


Figura 3.5: Profilo test con intervalli di dati mancanti

1. Ricostruzione per interpolazione lineare

La ricostruzione per interpolazione lineare utilizza un insieme di punti prima e dopo l'intervallo di dati mancanti per generare i nuovi punti e ricostruire così gli intervalli non validi. Come già detto, ai punti mancanti è stato associato un valore negativo (-16), per renderli visualizzabili nei grafici e meglio gestibili durante la ricostruzione.

I limiti imposti sono quindi:

- massimo della media dei profili ad altro consumo (dal Paragrafo 3.1.2), $M_E = 0.4356 \text{ kWh}$;
- limite superiore $p_M = 10 \cdot M_E$;
- limite inferiore $p_m = 0$.

Il limite inferiore è stato settato a zero perché nei profili di carico non sono presenti punti negativi: gli unici punti < 0 saranno quelli relativi ai dati mancanti.

Inizialmente questo metodo di ricostruzione utilizzava un numero di punti fisso per ricreare gli andamenti, indipendentemente dall'ampiezza dell'intervallo da ricostruire. Il numero di punti utilizzato era di 12, ma in questo modo si ricostruiva bene un intervallo di valori non ampio, mentre intervalli più grandi risultavano ricostruiti in maniera meno precisa. Si è quindi deciso di rendere dinamico questo vincolo, introducendo quello che nel codice "*function_dinamic_limit*" prende il nome di "*Ratio*", in questa formalizzazione matematica

definito come ψ_p .

Esso rappresenta il numero minimo di punti da considerare prima e dopo un intervallo unitario da ricostruire, posto pari a 2.

Considerando un intervallo di ampiezza A_{int} , maggiore di 1, si avrà quindi la relazione riportata nell'Equazione (3.1):

$$n_{p,ti}^{\circ} = \psi_p \cdot A_{\text{int}} \quad (3.1)$$

dove, $n_{p,ti}^{\circ}$ rappresenta il numero di punti necessari alla ricostruzione.

Ipotizzando quindi che $t_{m,i}$ rappresenti l'istante di tempo all'interno del profilo di carico in cui è presente il primo punto non valido, i punti a margine degli intervalli di ricostruzione prima e dopo l'intervallo di dati non valido si possono ricavare utilizzando le grandezze introdotte nell'equazione (3.1), ottenendo il set di equazioni riportate in (3.2):

$$\left\{ \begin{array}{l} p_{\text{start}}^{\text{bf}} = P[(t_{m,i} - 1) - (n_{p,ti}^{\circ} - 1)] = P(t_{m,i} - n_{p,ti}^{\circ}) \\ p_{\text{stop}}^{\text{bf}} = P(t_{m,i} - 1) \\ p_{\text{start}}^{\text{af}} = P(t_{m,i} + A_{\text{int}}) \\ p_{\text{stop}}^{\text{af}} = P[(t_{m,i} + A_{\text{int}}) + (n_{p,ti}^{\circ} - 1)] = P(t_{m,i} + 3 \cdot A_{\text{int}} - 1) \end{array} \right. \quad (3.2)$$

avendo indicato con $p_{\text{start}}^{\text{bf}}$ il punto del profilo di carico da cui iniziare la ricostruzione prima dell'intervallo di dati mancanti (bf = before) e $p_{\text{stop}}^{\text{bf}}$ il punto finale dell'intervallo di dati validi sempre prima dell'intervallo da ricostruire. Con $p_{\text{start}}^{\text{af}}$ è stato invece indicato il primo punto dell'intervallo di dati validi dopo la zona di NaN (af = after) e $p_{\text{stop}}^{\text{af}}$ si riferisce all'ultimo punto considerato in tale intervallo.

Una volta ottenute le coordinate per ogni punto l'andamento verrà ricostruito sfruttando la funzione di interpolazione lineare presente su *Matlab*[®]. Un esempio di punti ricostruiti con questo metodo è visualizzabile nella Figura 3.6.

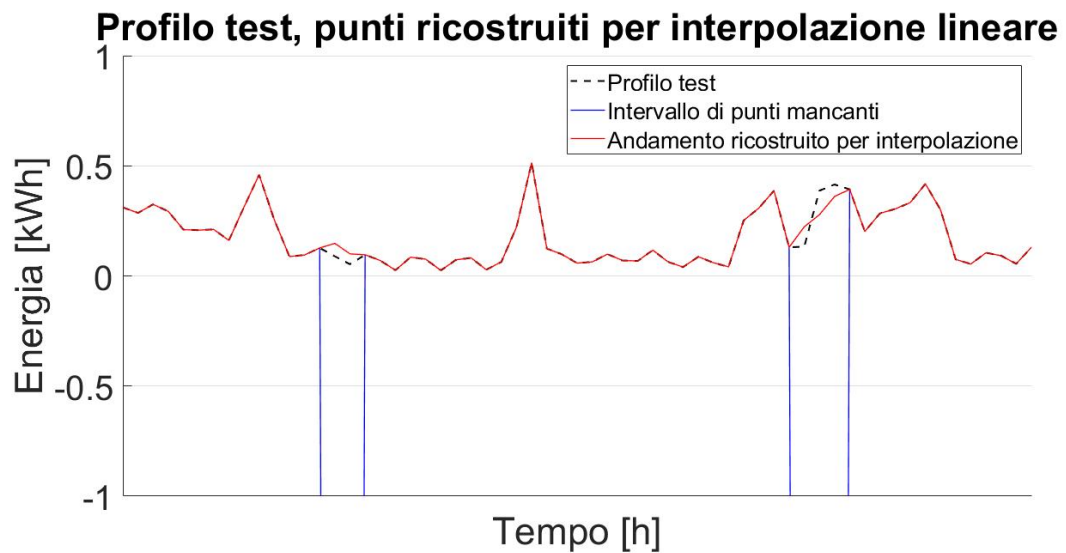


Figura 3.6: Punti ricostruiti per interpolazione lineare

La funzione interpolante utilizzata su *Matlab*[®] è definita come "*interp1*" che consta di diversi metodi, quello scelto è il "*linear*".

Un'ultima osservazione può essere fatta sulla grandezza ψ_p e sul suo legame con il numero di punti utili alla ricostruzione che era stato definito nell'equazione (3.1). Nelle Figure 3.7, 3.8 e 3.9 possiamo vedere la ricostruzione di un intervallo di 17 punti, per diversi valori di ψ_p .

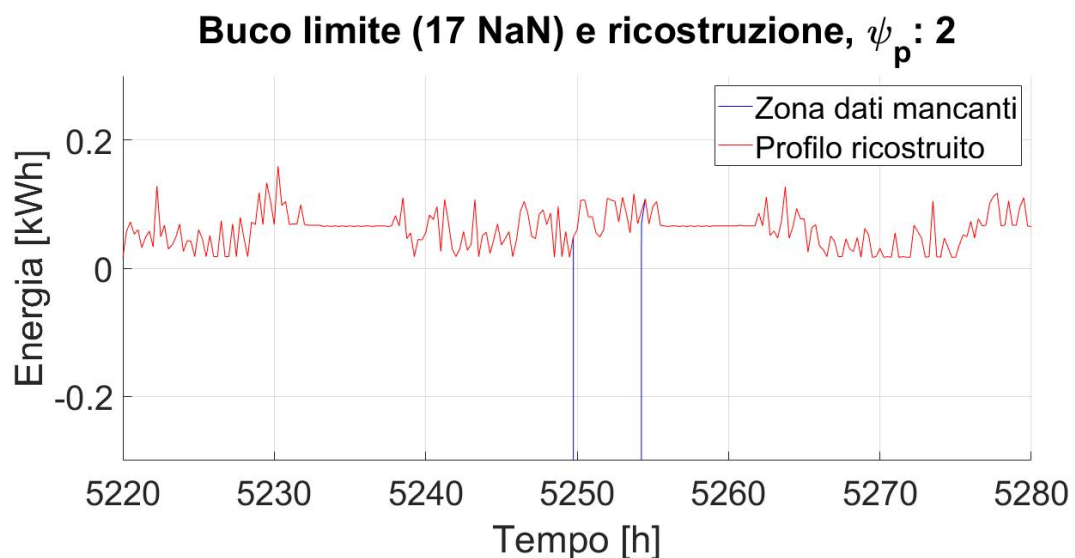


Figura 3.7: Ricostruzione di un buco di dati con $A_{\text{int}} = 17$ e $\psi_p = 2$

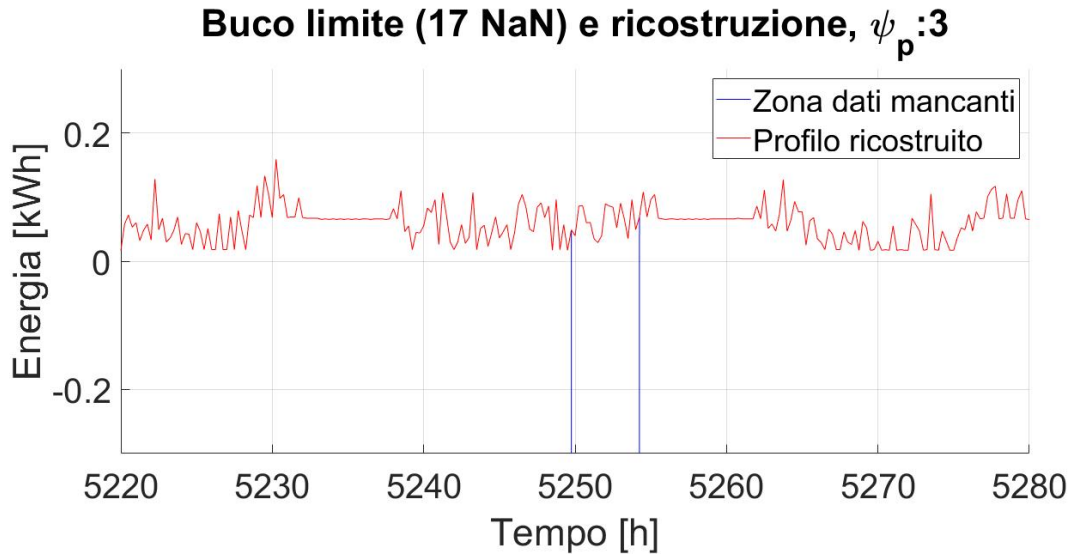


Figura 3.8: Ricostruzione di un buco di dati con $A_{\text{int}} = 17$ e $\psi_p = 3$

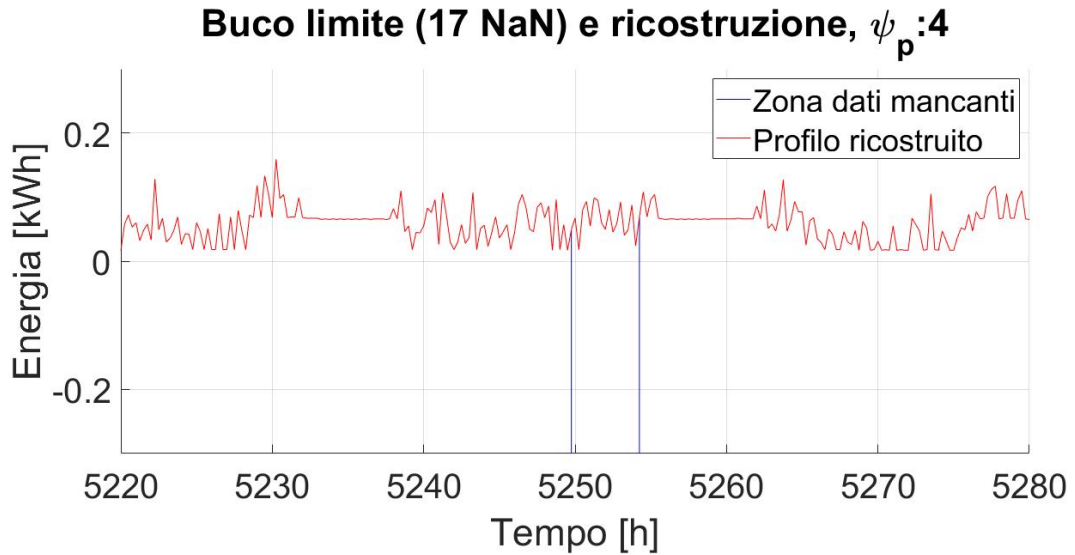


Figura 3.9: Ricostruzione di un buco di dati con $A_{\text{int}} = 17$ e $\psi_p = 4$

Si può notare che le ricostruzioni nelle Figure 3.7 e 3.8 risultano simili mentre quella nella Figura 3.9 risulta differente, ma $\psi_p = 4$ significherebbe considerare 68 punti validi precedenti all'intervallo mancante e 68 dopo tale intervallo, in accordo con l'equazione (3.1). Questo porterebbe ad escludere dei profili, in quanto in uno dei due intervalli allargati potrebbe andare ad inserirsi un NaN, facendo così scartare il profilo considerato. Si è quindi scelto di mantenere un ψ_p pari a 2 nelle ricostruzioni, in modo da ricostruire il maggior numero di profili possibili. In Figura 3.10 è riportato il flow-chart riassuntivo della ricostruzione per interpolazione lineare appena descritta.

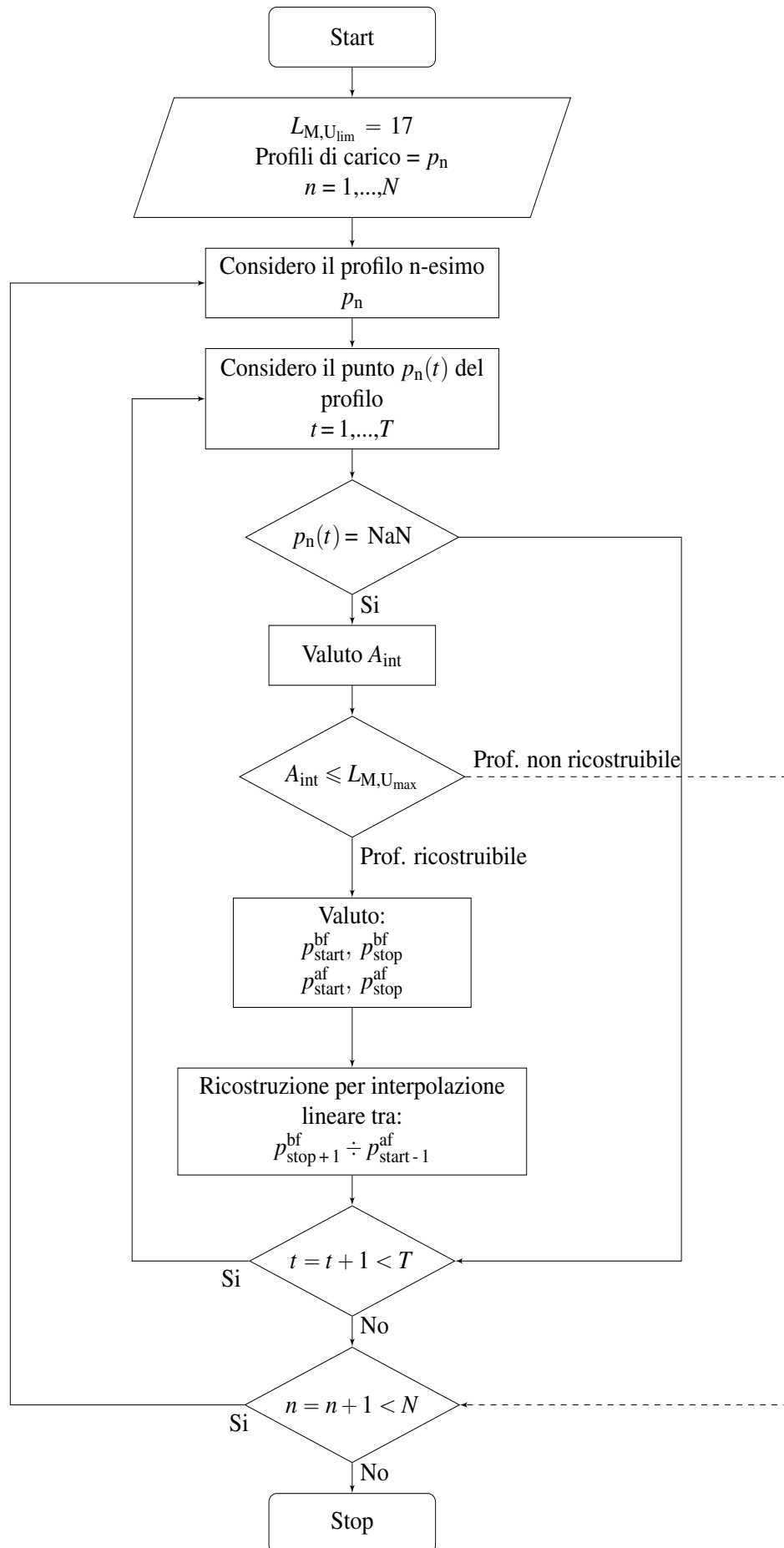


Figura 3.10: Flow-chart ricostruzione interpolazione lineare

2. Ricostruzione mediante punti randomici

Il secondo metodo di ricostruzione prevede l'utilizzo di punti casuali per la ricostruzione degli andamenti dei profili di carico. Questo metodo risulta essere meno preciso rispetto a quello che fa uso dell'interpolazione lineare, ma può risultare utile in applicazioni in cui è necessario un metodo più leggero e veloce di ricostruzione.

I punti casuali che verranno inseriti all'interno dei profili saranno contenuti nell'intervallo fra il minimo globale e il massimo globale dei punti validi del profilo considerati prima e dopo l'intervallo di dati mancante. Questi intervalli saranno valutati sempre utilizzando le equazioni (3.2).

Questo fornisce un vincolo non indifferente sui punti casuali ammessi per la ricostruzione, rendendo il metodo più preciso.

Il profilo che verrà utilizzato è sempre il profilo test riportato in Figura 3.5 in cui sono stati inseriti 118 intervalli di dati non validi. Per ognuno di questi intervalli si determineranno prima dei massimi e minimi locali per i punti che precedono e seguono la zona non valida. E in seguito si sfrutteranno questi punti per determinare il massimo e il minimo globale, quindi gli estremi dell'intervallo di valori che potranno essere estratti per la ricostruzione.

Considerando un intervallo di dati non valido e andando a definire i punti di interesse prima e dopo tale intervallo mediante le equazioni (3.2), si possono determinare i massimi e minimi locali, come riportato nelle equazioni (3.3) e (3.4):

$$\begin{cases} p_{M,l}^{bf} = \max(p_{start}^{bf} \div p_{stop}^{bf}) \\ p_{m,l}^{bf} = \min(p_{start}^{bf} \div p_{stop}^{bf}) \end{cases} \quad (3.3)$$

$$\begin{cases} p_{M,l}^{af} = \max(p_{start}^{af} \div p_{stop}^{af}) \\ p_{m,l}^{af} = \min(p_{start}^{af} \div p_{stop}^{af}) \end{cases} \quad (3.4)$$

dove $p_{M,l}^{bf}$ e $p_{m,l}^{bf}$ rappresentano rispettivamente il massimo locale ed il minimo locale prima della zona di punti mancanti (è stata usata ancora la dicitura bf, *before*, per indicarli), $p_{M,l}^{af}$ e $p_{m,l}^{af}$ rappresentano invece il massimo locale il minimo locale dopo l'intervallo di dati man-

canti (indicati con af, *after*).

Dopo aver definito in ogni intervallo d'interesse le grandezze locali si possono definire le grandezze di massimo e minimo globali fra i due intervalli, come riportato nelle equazioni (3.5) e (3.6):

$$p_{M,g} = \max(p_{M,l}^{bf}, p_{M,l}^{af}) \quad (3.5)$$

$$p_{m,g} = \min(p_{m,l}^{bf}, p_{m,l}^{af}) \quad (3.6)$$

dove, $p_{M,g}$ rappresenta il massimo globale degli intervalli di ricostruzione, mentre $p_{m,g}$ il minimo globale dei suddetti intervalli.

Quindi, considerando $t_{m,i}$ il primo punto non valido di un generico intervallo e $t_{m,f} = t_{m,i} + n$ il punto finale, avremo che tutti i punti da ricostruire, estremi compresi rispetteranno la condizione presentata nell'equazione (3.7):

$$p_{r,t'} = \text{random}(p_{M,g} \div p_{m,g}) \quad t' = t_{m,i}, \dots, t_{m,f} \quad (3.7)$$

avendo indicato con $p_{r,t'}$ i generici punti casuali che comporranno l'intervallo $t_{m,i} \div t_{m,f}$.

Un esempio di ricostruzione utilizzando questo metodo viene mostrato in Figura 3.11

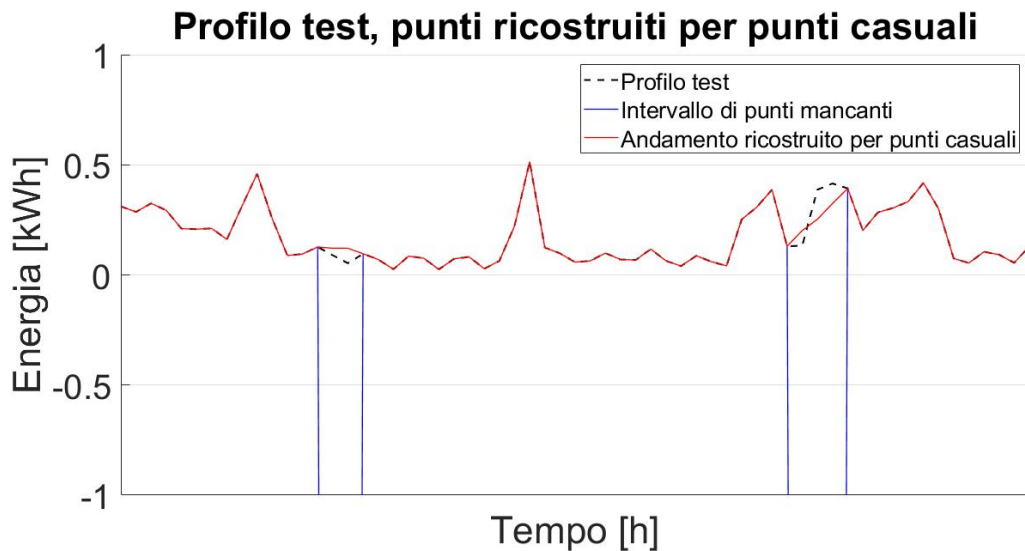


Figura 3.11: Intervalli di dati ricostruiti con metodo dei punti casuali

In Figura 3.12 è riportato il flow-chart riassuntivo della ricostruzione per punti casuali appena descritta.

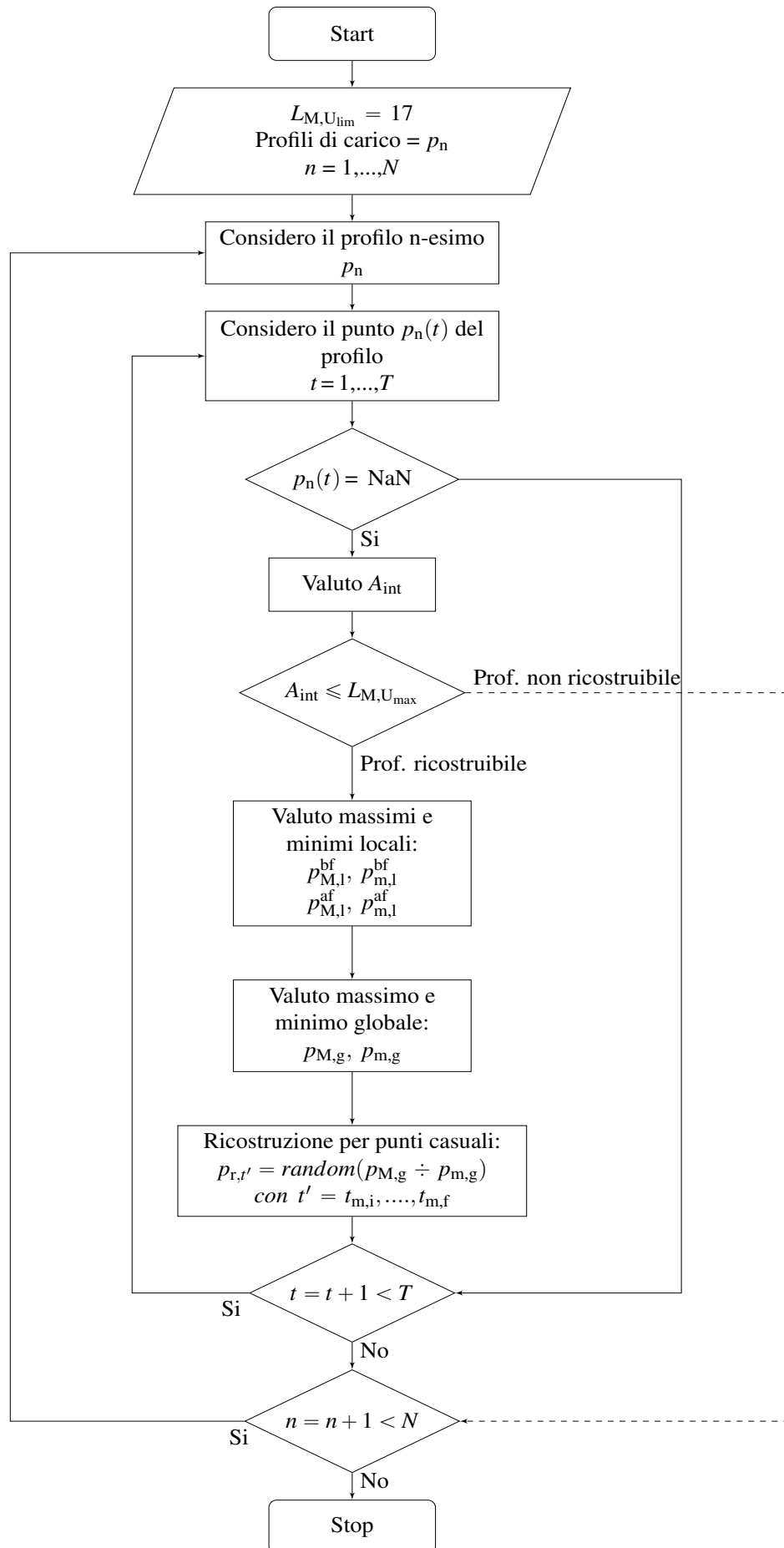


Figura 3.12: Flow-chart ricostruzione per punti casuali

3. Confronto fra i diversi tipi di ricostruzione

Si può fare adesso un confronto fra i due tipi di ricostruzione proposti. Il primo metodo è quello che usa l'interpolazione lineare dei punti nell'intervallo di interesse prima e dopo il buco di dati. Il secondo, invece, è quello che usa punti casuali considerando i massimi e minimi globali sempre negli intervalli prima e dopo il buco di dati.

Ciò che ne deriva, come si potrà osservare, è che il primo metodo funziona meglio del secondo. Questo si vede già con un'ispezione visiva di un tratto di profilo di carico, valutando le differenze di ricostruzione presenti (Figura 3.13), sia visualizzando gli andamenti mediante cumulativa.

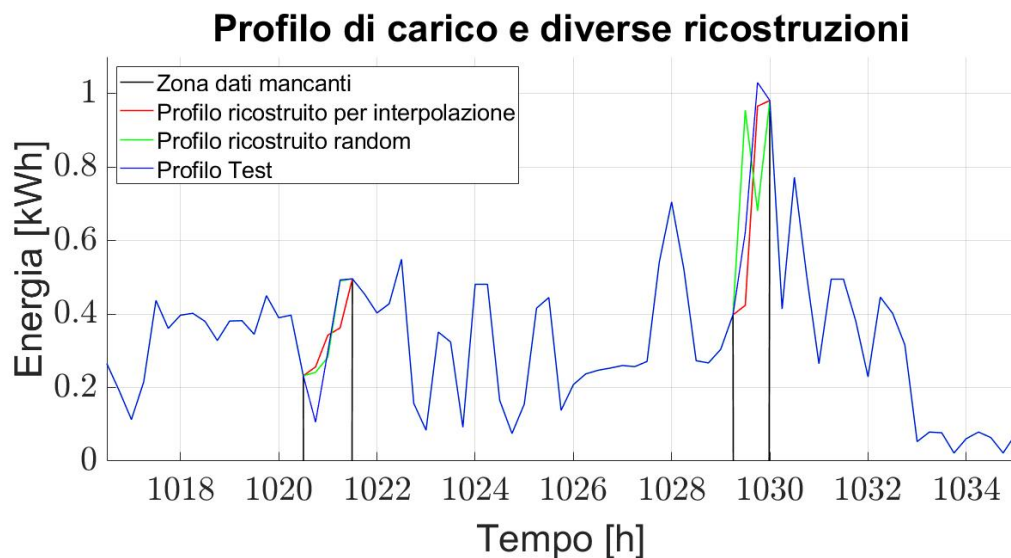


Figura 3.13: Profilo di carico (buchi di dati ricostruiti con diverse procedure)

Inizialmente la cumulativa veniva estratta considerando tutti i 35040 punti di potenza associati al profilo, quindi le ricostruzioni sembravano dare delle cumulative perfettamente sovrapposte e quindi risultati simili. Questo era dovuto al fatto che, pur avendo inserito dei buchi di dati, i punti mancanti erano sempre minori rispetto a quelli appartenenti al profilo originario. Si è quindi deciso di isolare solo gli indici relativi ai punti NaN inseriti nel profilo test e valutare le cumulative in quest'ultimo caso, il grafico è riportato nella Figura 3.14:

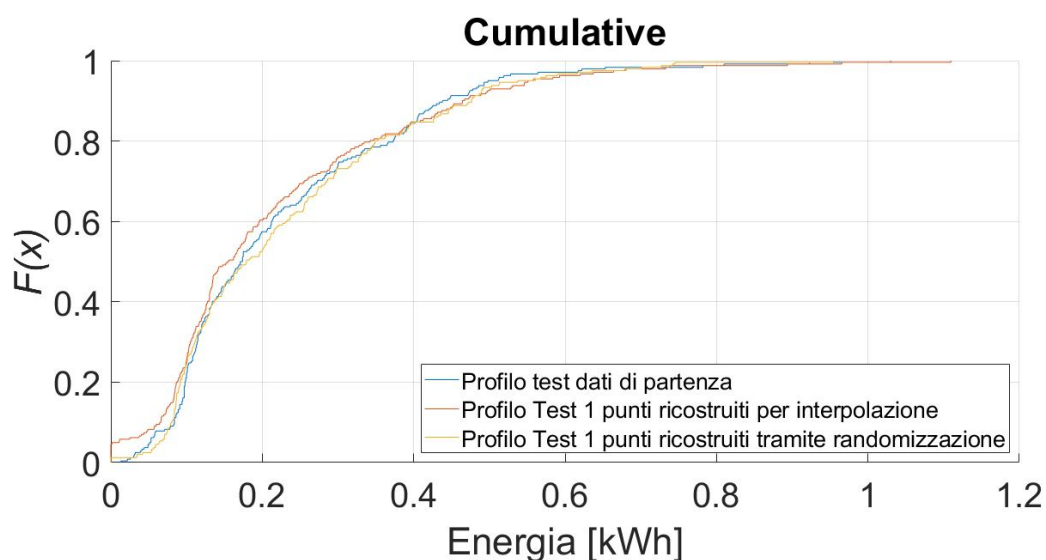


Figura 3.14: Cumulative profilo test e punti ricostruiti

A questo punto sono state calcolate le differenze fra i punti ottenuti mediante le ricostruzioni e i punti appartenenti al profilo originario. Rappresentando le funzioni cumulative di questi ultimi casi (Figura 3.15), si nota che la funzione riferita alla differenza tra il profilo ricostruito per interpolazione lineare e quello originario risulta più vicina all'asse delle ordinate, rispetto a quella contenente la differenza fra il profilo ricostruito in modo casuale e il profilo originario. Si può quindi concludere che il metodo di ricostruzione mediante interpolazione lineare, seppur necessiti di un maggior tempo di calcolo, risulta più preciso rispetto al secondo.

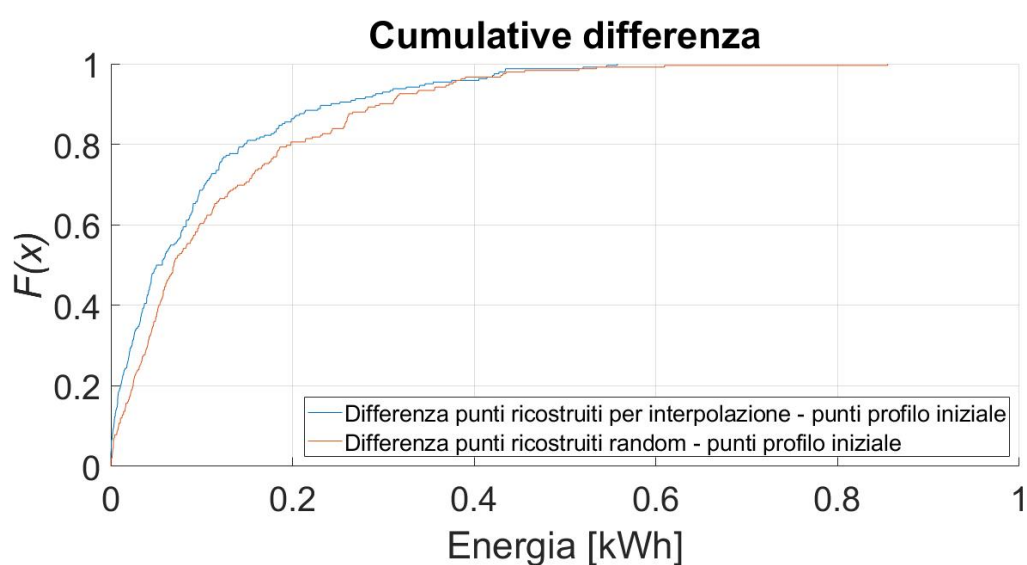


Figura 3.15: Differenza cumulative punti ricostruiti - punti profilo test

3.1.5 Giustificazione scelta profili

Tra i profili analizzati, quello che presenta un maggior numero di intervalli con dati non validi è il profilo numero 51: esso presenta 17 intervalli privi di dati, i cui valori di posizione del primo punto mancante e ampiezza dell'intervallo sono riportati nella Tabella 3.4:

Tabella 3.4: Intervallo punti mancanti Profilo di carico n° 51

Profilo di carico n° 51		
Intervallo n°	Posizione primo punto mancante	Ampiezza intervallo (A_{int}) (incluso primo punto mancante)
1	481	1
2	575	2
3	2111	1
4	2977	1
5	3071	2
6	9025	1
7	9119	2
8	11809	1
9	11903	2
10	17665	1
11	17758	3
12	23713	1
13	23741	1
14	23806	3
15	29281	1
16	29365	12
17	31376	1

Si considera quindi lo stesso profilo test definito in 3.1.4 privo di dati mancanti e ci si

pone in condizioni di worst-case.

All'interno del profilo test verranno inseriti degli intervalli di dati non validi; il numero sarà fissato, ma sarà comunque maggiore rispetto a quello del profilo 51 e di ampiezza minima e massima prefissata. Si avranno quindi due casi: il primo sarà uguale a quello presentato nella Figura 3.5 in cui si avranno 118 intervalli di ampiezza $1 \leq A_{\text{int}} \leq 3$, definito *caso 1*, mentre il secondo, in cui si avranno 56 intervalli di ampiezza $4 \leq A_{\text{int}} \leq 22$ sarà definito come *caso 2*. Verrà utilizzata la norma quadratica per definire la distanza fra i punti nel profilo in cui sono stati inseriti gli intervalli non validi e il corrispondente valore valido presente nel profilo test, come descritto nelle equazioni (3.8).

$$\begin{cases} \|p_{\text{caso 1}} - p_{\text{test}}\| = \sqrt{\sum_{i=1}^n (p_{\text{caso 1},i} - p_{\text{test},i})^2} \\ \|p_{\text{caso 2}} - p_{\text{test}}\| = \sqrt{\sum_{i=1}^n (p_{\text{caso 2},i} - p_{\text{test},i})^2} \end{cases} \quad (3.8)$$

Si riesce così a definire l'errore nei due casi: ciò che ci si aspetta è che, il profilo che presenta un numero di buchi maggiore ma con ampiezza minore (*caso 1*), abbia un valore minore della norma rispetto a quella calcolata nel *caso 2*.

Gli intervalli non validi sono stati inseriti mediante dei generatori di numeri casuali. Gli unici vicoli che vengono considerati riguardano, come già detto, l'ampiezza di tali buchi di dati e, poiché nessun profilo presenta dei valori mancanti nei primi valori, si considera un limite "imperturbato" del profilo. Questo viene calcolato come il rapporto tra la dimensione del profilo test e il numero massimo di intervalli mancanti che vengono inseriti (118 nel nostro caso, cioè quelli relativi al *caso 1*), approssimato per eccesso al fine di ottenere sempre un numero intero (equazione (3.9)).

$$\lim_p = \frac{L_p}{n_b^\circ} = \frac{35040}{118} = 296.9 = 297 \quad (3.9)$$

dove $\lim_p$ rappresenta il limite imperturbato del profilo in cui sicuramente non verranno inseriti punti mancanti, L_p la lunghezza del profilo di carico e n_b° il massimo numero di intervalli mancanti presenti.

Infine, un ultimo vincolo viene posto sulla distanza minima tra gli intervalli, che sarà posta

di almeno 24 posizioni, più un ulteriore margine cautelativo di 2 posizioni. Tale passaggio è visualizzabile nel Codice A.2, righe 25÷51, relativo al *caso 1*.

I profili che si ottengono in seguito all'inserimento del numero di intervalli non validi voluto sono riportati in Figura 3.16:

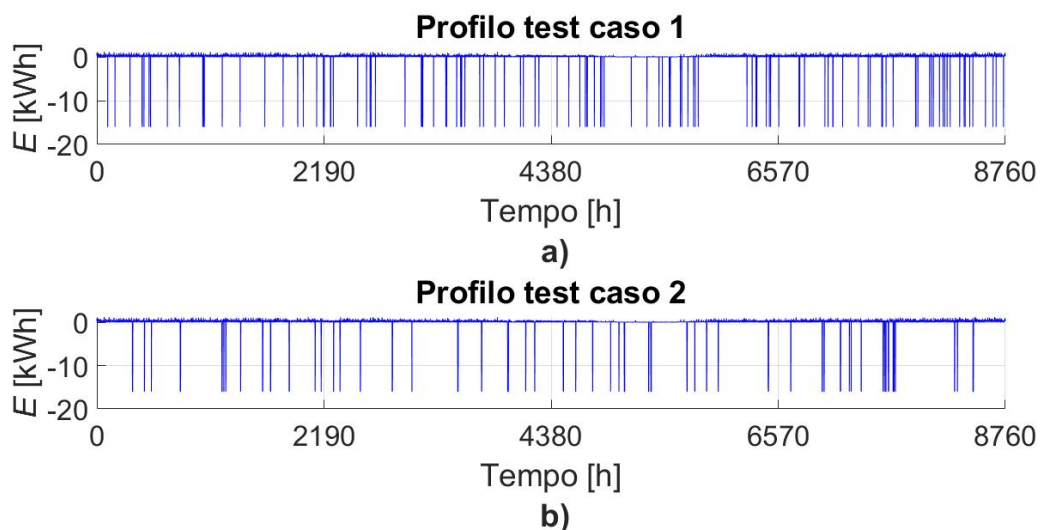


Figura 3.16: Profilo test *caso 1* a) e profilo test *caso 2* b)

Ai punti non validi, come ormai è prassi, è stato associato il valore negativo -16. Si riportano nella Tabella 3.5 i risultati ottenuti sui profili precedentemente descritti.

Tabella 3.5: Norma e rapporto nei due casi considerati

Norma caso 1	Norma caso 1
2.2704	5.7389
Rapporto caso 1	Rapporto caso 2
0.0435	0.1099

Come si vede il calcolo della norma quadratica nel secondo caso non solo è maggiore, ma il rapporto tra gli errori nei due casi è circa di 1:2.5. Ciò comporta che la scelta di profili che presentano un numero elevato di buchi di dati ma di ampiezza piccola, una volta ricostruiti, possono essere considerati nell'aggregato finale, mentre profili che presentano

dei buchi di dati grandi, seppur risultando alla vista ben ricostruiti, introdurrebbero un errore elevato nell'aggregato finale.

3.1.6 Creazione aggregato di carico residenziale

Dopo aver analizzato i profili residenziali si è proceduto alla creazione dell'aggregato totale. Si è deciso di creare un codice che, oltre a generare un aggregato di tipo residenziale, sfruttando tutti i profili utilizzabili consentisse di generare degli aggregati che rispettassero anche diverse condizioni. In particolare si è deciso di generare degli aggregati in maniera puramente casuale, cercando di renderli il più differenti possibile. Vengono considerati essenzialmente tre casi:

1. Aggregati composti da soli profili ad alto consumo;
2. Aggregati composti da soli profili a basso consumo;
3. Aggregati misti, considerando i profili provenienti dai gruppi ad alto e basso consumo secondo determinate percentuali.

Avendo considerato il limite massimo di dati mancanti pari a 17, il calcolo degli aggregati avverrà considerando 28 profili di carico ad alto consumo e 83 a basso. I valori, istante per istante, verranno divisi per 0.25, essendo tali valori energia sul quarto d'ora e volendo ottenere in tal modo dei valori di potenza.

Gli indici dei profili da utilizzare per gli aggregati sono stati generati mediante la funzione randomica di permutazione “*randperm*”, in modo da non utilizzare più volte lo stesso profilo nel singolo aggregato. In generale la relazione che intercorre fra numero profili, numero di aggregati e dimensione degli stessi è riportata nell'equazione (3.10):

$$L_{\text{agg,max}} = \frac{N_p}{n_{\text{agg}}^{\circ}} \quad (3.10)$$

dove si è indicato con $L_{\text{agg,max}}$ la lunghezza dell'aggregato, con N_p il numero di profili che compongono l'aggregato e con n_{agg}° il numero di aggregati voluto.

Quindi se si vorranno ottenere degli aggregati totalmente diversi, cioè con gli indici dei profili *mai* ripetuti in più aggregati occorre che sia rispettata l'equazione (3.11):

$$L_{agg,max} \cdot n_{agg}^{\circ} \leq N_p \quad (3.11)$$

Se l'equazione (3.11) non viene rispettata, gli aggregati saranno allora parzialmente distinti. Per renderli il meno simili possibile, la condizione che si rispetta è quella del “**numero massimo di sequenze di elementi n in classe k** ”, matematicamente definita (3.12):

$$\frac{n!}{k! * (n - k)!} \quad (3.12)$$

Quello che viene fatto all'interno della funzione “*function_indici_aggregati.m*” è quindi un primo sorteggio casuale per un numero di elementi che formeranno la “*massima porzione distinta degli aggregati*”, $Max_{porz,agg}$ data dal rapporto definito nell'equazione (3.13):

$$Max_{porz,agg} = \frac{N_p}{L_{agg,max}} \quad (3.13)$$

Gli elementi rimanenti verranno generati mediante un ulteriore sorteggio di tipo casuale ripetuto al fine di verificare che la porzione di aggregato sia diversa da quella già presente nel profilo e, se la condizione è verificata, gli indici generati verranno inseriti nelle posizioni rimanenti. Il codice utilizzato è riportato nell'Appendice A.6.2.

Nelle Figure 3.17, 3.18 e 3.19 vengono riportati degli aggregati ottenuti dal codice che è stato appena descritto:

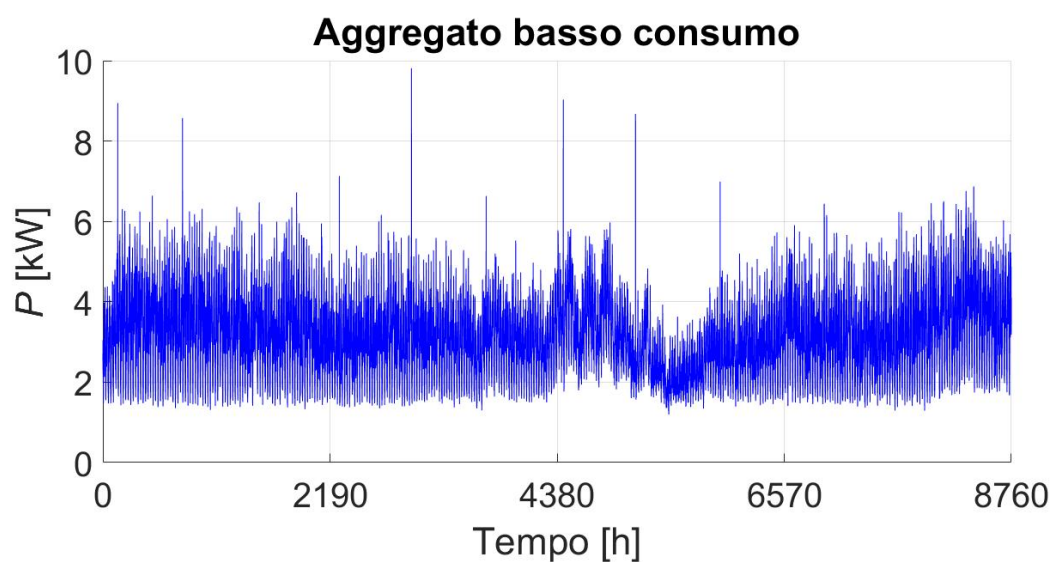


Figura 3.17: Aggregato profili residenziali basso consumo

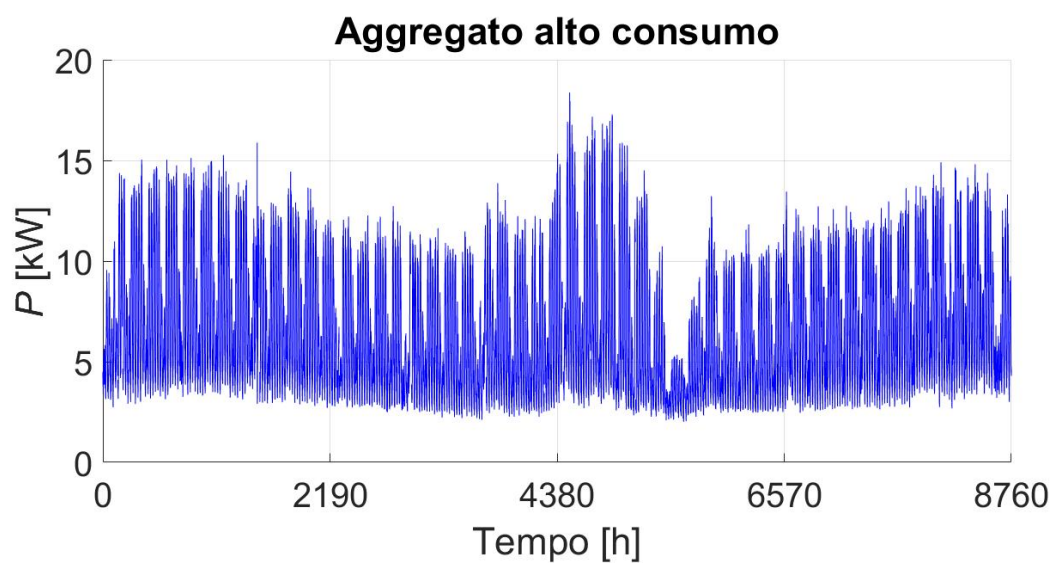


Figura 3.18: Aggregato profili residenziali alto consumo

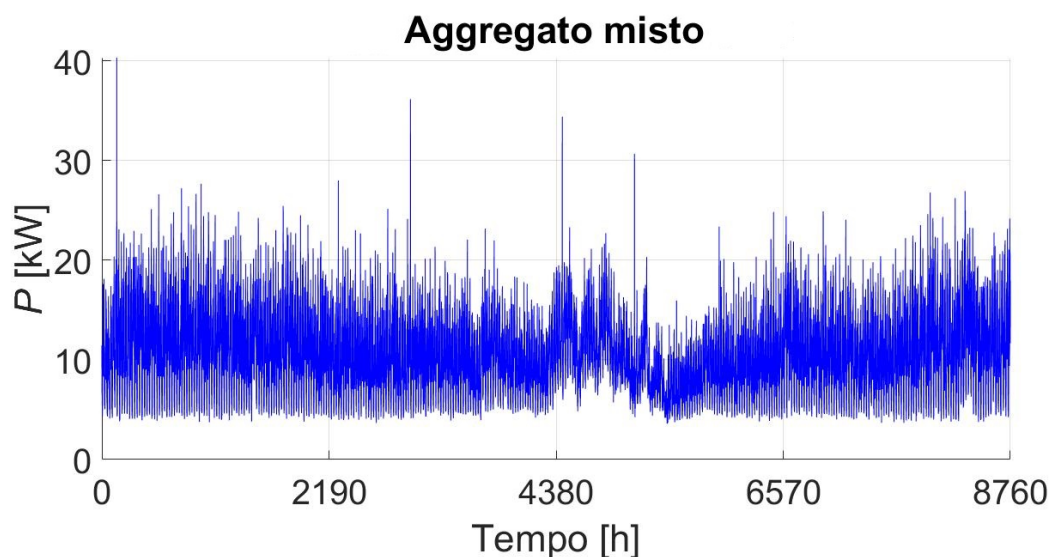


Figura 3.19: Aggregato profili residenziali misto

Nel prosieguo verrà utilizzato solo l'aggregato ottenuto dai profili a basso consumo, con la denominazione di “*Aggregato_TOT*”. Tale aggregato è stato riportato in per unità rispetto alla potenza nominale, tenendo conto della presenza di utenze residenziali con una potenza di 3 kW e altri con una potenza di 4.5 kW, l'aggregato in per unità è mostrato nella [Figura 3.20](#).

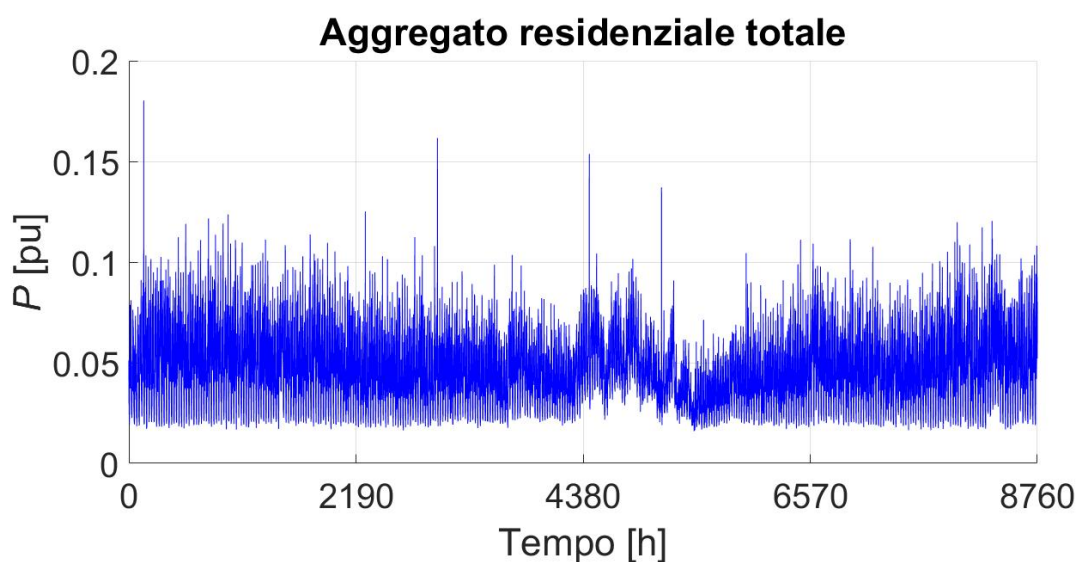


Figura 3.20: Aggregato profili residenziali misto

3.2 Studio profili di carico utenti non domestici

3.2.1 Estrazione dati da Excel

In questo breve paragrafo verrà esposta la procedura di estrazione dei dati dai fogli Excel relativi agli utenti non domestici selezionati. Tali profili appartenevano a diverse tipologie di utenti, divisi in tre categorie:

- utenti commerciali;
- utenti industriali;
- generico uffici.

Per ciascuno di queste tipologie sono stati forniti cinque profili diversi, per un totale quindi di quindici profili. I file Excel forniti erano organizzati come mostrato in Figura 3.21: ogni foglio conteneva i valori di energia attiva e reattiva per un dato profilo ed ogni riga rappresentava una giornata, formata da 96 letture diverse, una ogni quarto d'ora.

Nella prima colonna dei file excel una dicitura diversa indicava se quella riga corrisponde ad un'energia attiva o reattiva ("EA" sarà quindi Energia Attiva, mentre "ERQ1" Energia Reattiva).

	A	B	C	D	E	F	G
1	A1022_TIPO_SEGNANTE	A1022_TIPO_CANALE	A1022_DATA_LETTURA	A1022_LETTURA_V1	A1022_LETTURA_V2	A1022_LETTURA_V3	A1022_LETTURA_V4
2	EA	PREL	01-gen-18	6,732	6,6	6,292	6,908
3	ERQ1	PREL	01-gen-18	4,092	3,872	3,564	3,872
4	EA	PREL	02-gen-18	6,6	6,6	5,94	6,468
5	ERQ1	PREL	02-gen-18	4,004	3,784	3,432	3,696
6	ERQ1	PREL	03-gen-18	4,664	4,576	4,664	4,4
7	EA	PREL	03-gen-18	7,48	7,392	7,304	7,348
8	ERQ1	PREL	04-gen-18	5,456	5,412	5,324	4,884
9	EA	PREL	04-gen-18	8,316	8,096	8,096	8,008
10	EA	PREL	05-gen-18	14,872	8,272	8,228	8,492
11	ERQ1	PREL	05-gen-18	4,972	5,236	5,236	5,104
12	EA	PREL	06-gen-18	10,868	8,492	8,228	8,272
13	ERQ1	PREL	06-gen-18	5,192	5,192	5,148	4,796
14	EA	PREL	07-gen-18	8,492	8,316	7,964	8,184
15	ERQ1	PREL	07-gen-18	5,412	5,192	4,752	4,928
16	ERQ1	PREL	08-gen-18	4,752	4,488	4,4	4,18
17	EA	PREL	08-gen-18	7,436	7,128	6,996	7,04
18	ERQ1	PREL	09-gen-18	4,224	4,488	4,268	4,312
19	EA	PREL	09-gen-18	15,136	15,488	15,444	15,488
20	ERQ1	PREL	10-gen-18	5,324	5,412	4,84	5,06
21	EA	PREL	10-gen-18	8,404	8,228	7,656	8,052
22	EA	PREL	11-gen-18	16,764	14,74	13,288	9,064
23	ERQ1	PREL	11-gen-18	4,872	4,444	4,664	5,192

Figura 3.21: Dettaglio foglio excel profili non residenziali (Utenti commerciali)

Il Codice scritto su *Matlab*[®] e denominato "*function_estrazione_profili_non_residenziali*", riportato in Appendice [A.7.1](#), consente di estrarre i valori di energia attiva e reattiva dai vari fogli Excel.

Tali valori sono stati poi ordinati e disposti in strutture contenenti le sole potenze attive e quelle reattive.

I profili così ottenuti sono stati poi convertiti in potenze e riportati in per unità.

3.2.2 Riporto in per unità dei profili non residenziali

I profili non residenziali sono stati riportati in per unità come è stato fatto per l'aggregato. Questo perché l'andamento di questi profili verrà poi utilizzato per creare un profilo a cui verrà associato un valore di potenza installato, come si vedrà nel Capitolo [4](#) (Paragrafo [4.2.1](#)). Per riportare quindi in p.u. questi profili è stato considerato il valore di potenza acquistata che era stato fornito mediante documentazione. I valori di tale potenza acquistata dei vari profili sono riportati nelle Tabelle [3.6](#), [3.7](#) e [3.8](#):

Tabella 3.6: Caratteristiche profili utenti commerciali

Profili commerciali		
Profilo	Potenza acquistata	Tensione
n°	(kW)	(V)
1	250	22000
2	150	400
3	1183	22000
4	750	22000
5	80.5	400

Tabella 3.7: Caratteristiche profili utenti industriali

Profili industriali		
Profilo	Potenza acquistata	Tensione
n°	(kW)	(V)
1	400	22000
2	850	22000
3	7500	27000
4	113	400
5	131.5	6300

Tabella 3.8: Caratteristiche profili utenti generico uffici

Profili generico uffici		
Profilo	Potenza acquistata	Tensione
n°	(kW)	(V)
1	700	6300
2	105	400
3	150	22000
4	60	400
5	506	22000

Come si può notare, alcuni profili presentano un valore di tensione relativo ad una vecchia porzione della rete di Torino (rete a 6.3 kV). Questi profili non verranno utilizzati nella ripartizione per la creazione degli aggregati nodali. Si utilizzeranno quindi gli andamenti dei profili relativi a carichi alimentati a 400 V e 22 kV.

I profili in per unità vengono quindi ottenuti dividendo i profili di carico in potenza per il valore della potenza acquistata riferito al profilo di carico. Per la potenza reattiva questo

significa che l'andamento riportato sarà legato al fattore di potenza:

$$P_{pu}^{(tipo)} = \frac{P^{(tipo)}}{P_{acq}^{(tipo)}} \quad (3.14)$$

$$Q_{pu}^{(tipo)} = \frac{Q^{(tipo)}}{Q_{acq}^{(tipo)}} = \frac{\sin \varphi}{\cos \varphi} = \tan \varphi \quad (3.15)$$

dove, $P_{pu}^{(tipo)}$ e $Q_{pu}^{(tipo)}$ rappresentano i valori in per unità della potenza attiva e reattiva dei profili per tipologia, con (tipo) = commerciale, industriale, generico uffici. $P^{(tipo)}$ e $Q^{(tipo)}$ i valori di potenza attiva in kW e reattiva in kvar per tipologia e infine $P_{acq}^{(tipo)}$ e $Q_{acq}^{(tipo)}$ i valori di potenza acquistata e potenza reattiva forniti tramite documentazione. Nelle Figure da 3.22 a 3.36 si riportano gli andamenti dei profili in per unità per gli utenti commerciali, industriali e generici uffici ottenuti in questa analisi:

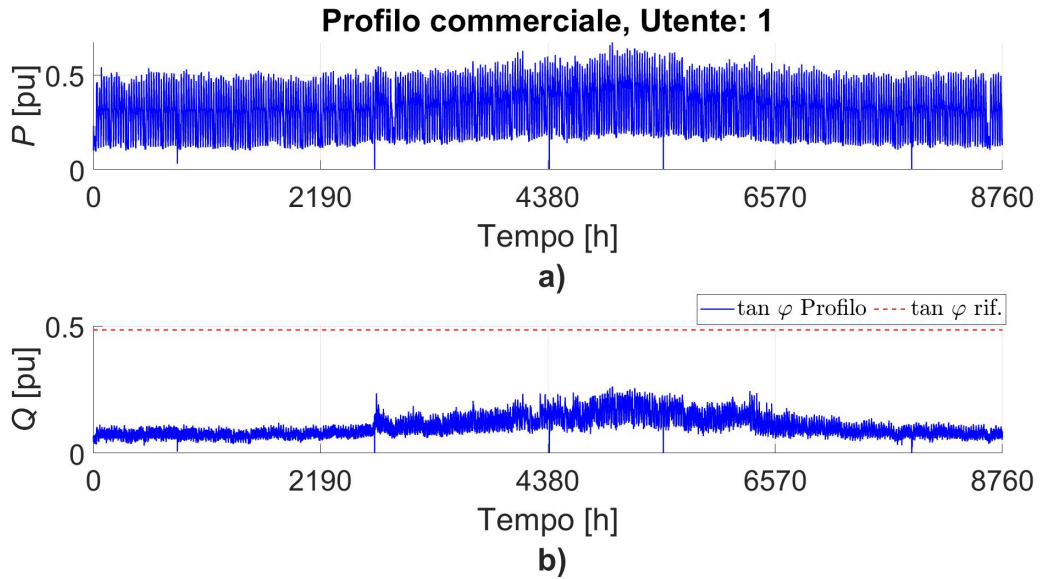


Figura 3.22: Profilo commerciale, utente 1, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

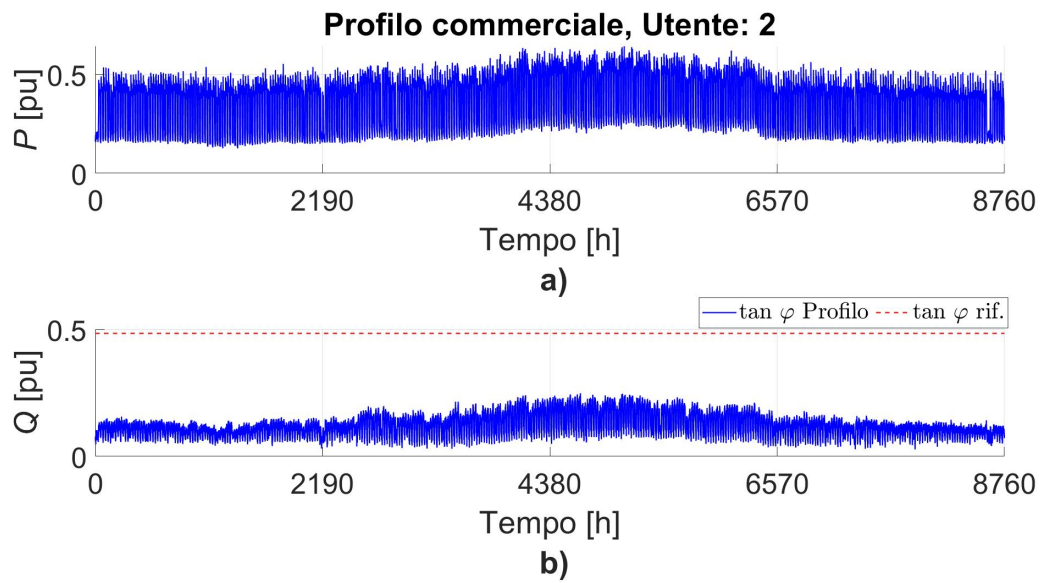


Figura 3.23: Profilo commerciale, utente 2, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

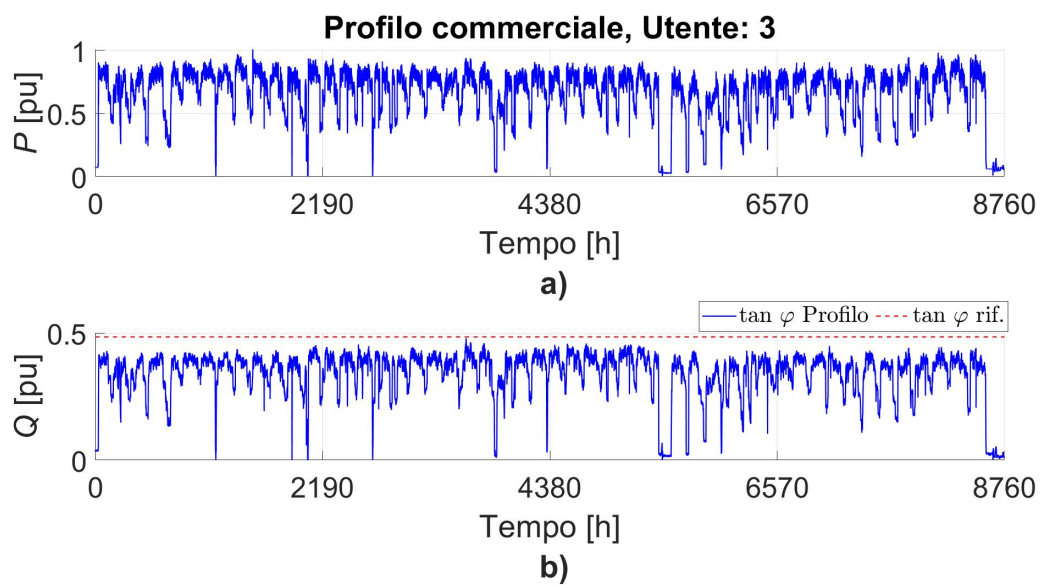


Figura 3.24: Profilo commerciale, utente 3, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

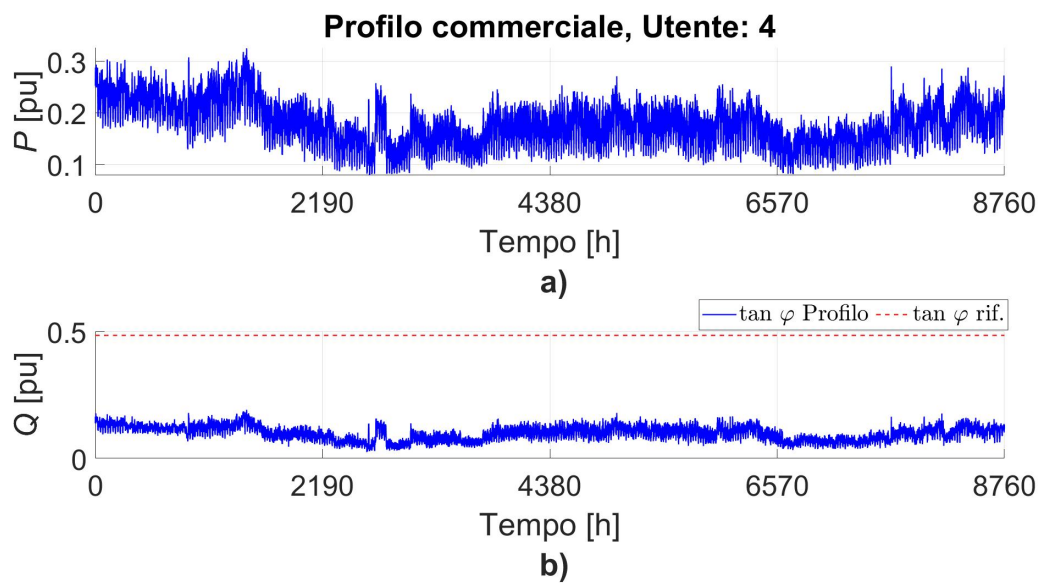


Figura 3.25: Profilo commerciale, utente 4, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$)

b)

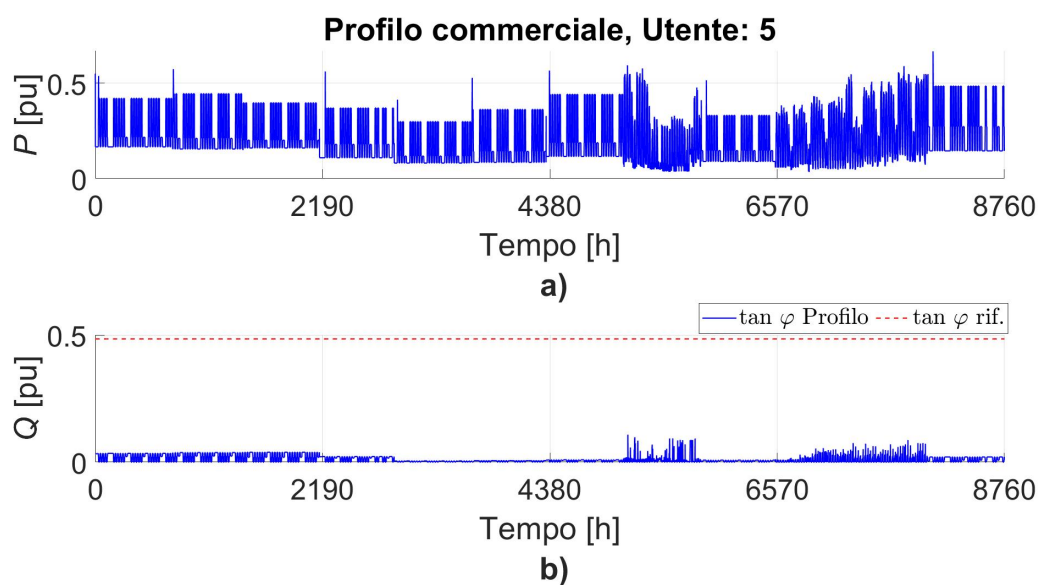


Figura 3.26: Profilo commerciale, utente 5, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$)

b)

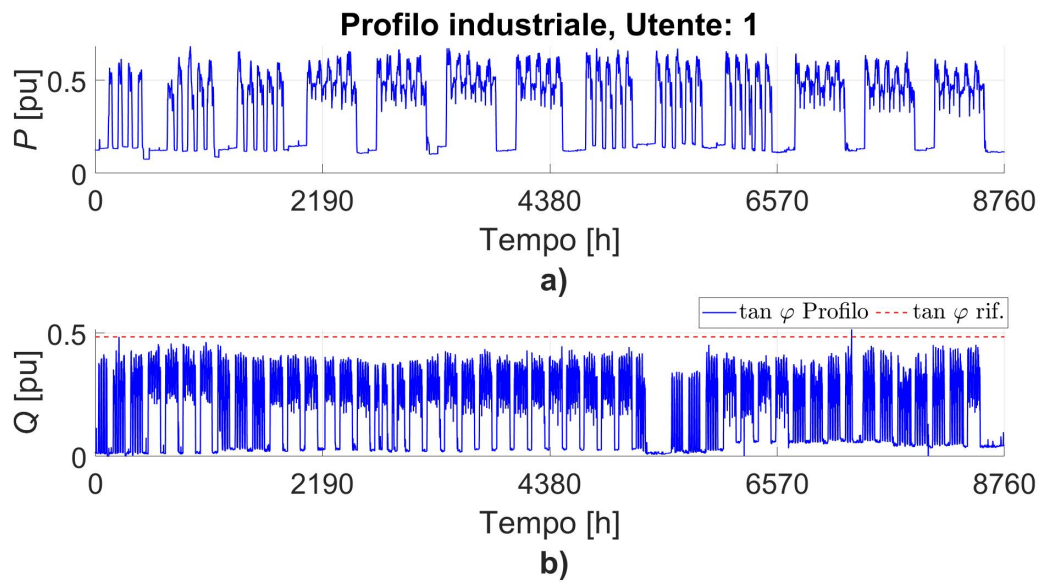


Figura 3.27: Profilo industriale, utente 1, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

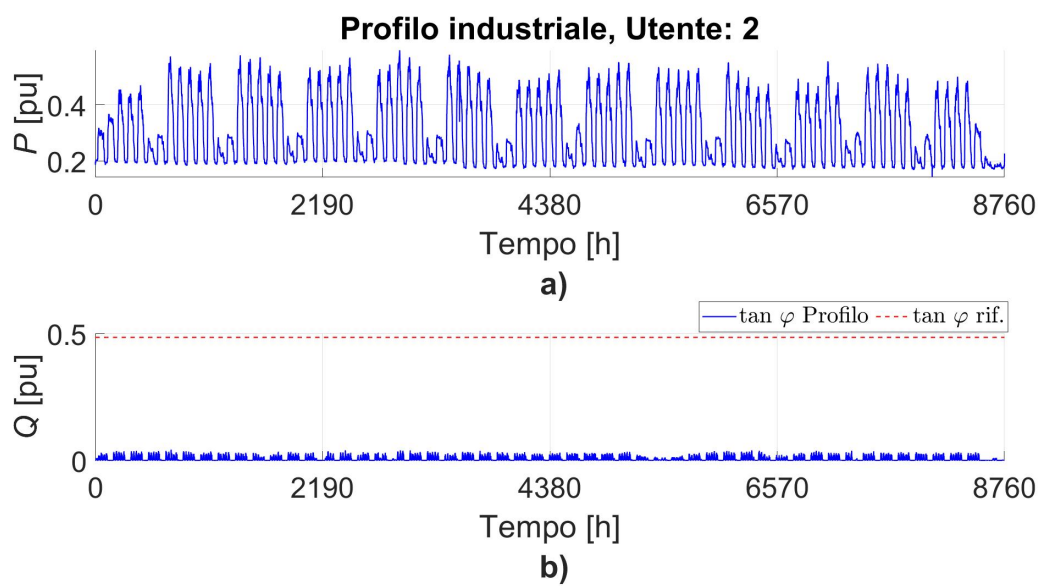


Figura 3.28: Profilo industriale, utente 2, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

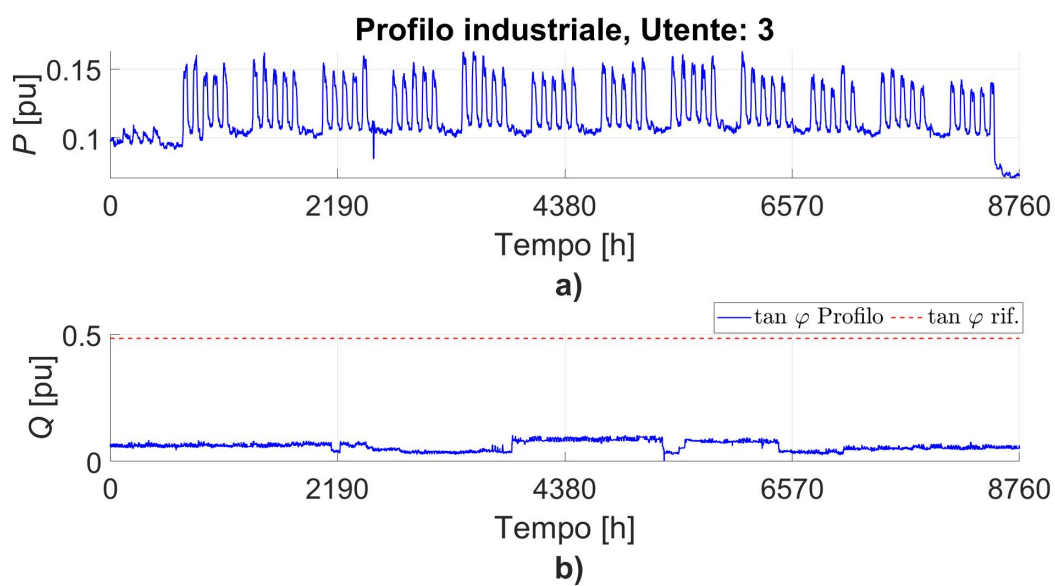


Figura 3.29: Profilo industriale, utente 3, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

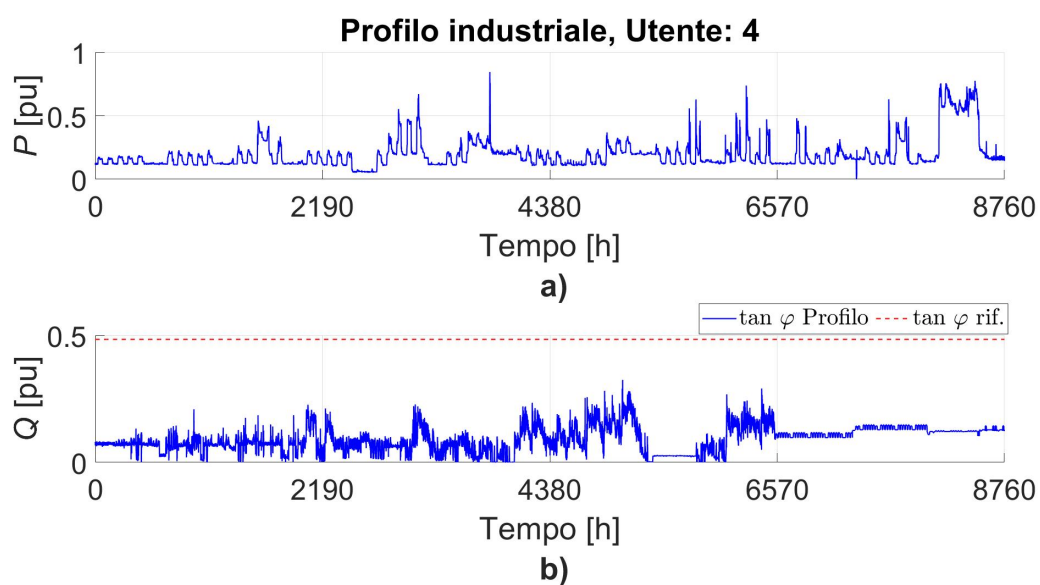


Figura 3.30: Profilo Industriale, utente 4, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

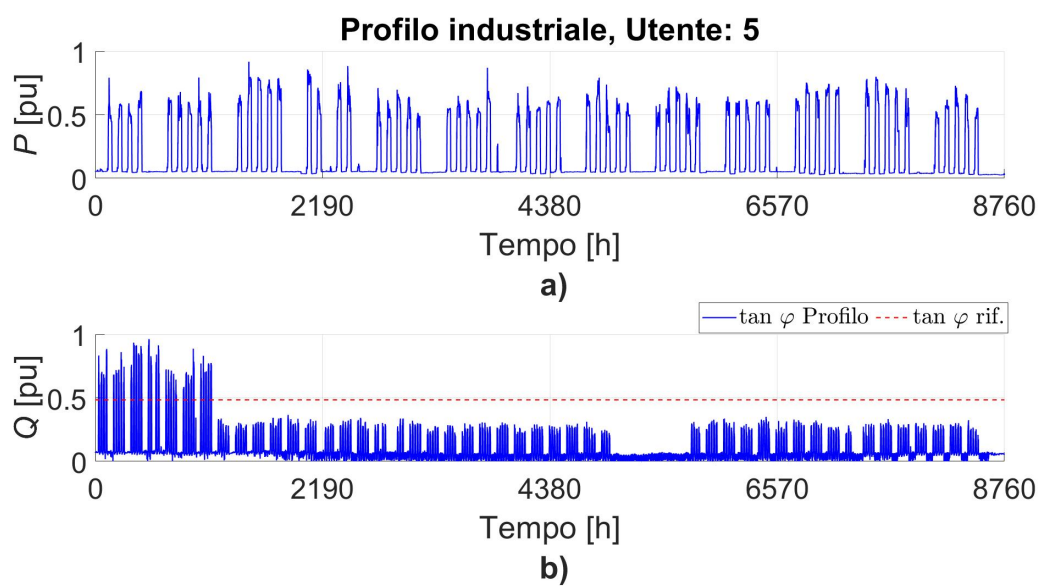


Figura 3.31: Profilo industriale, utente 5, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

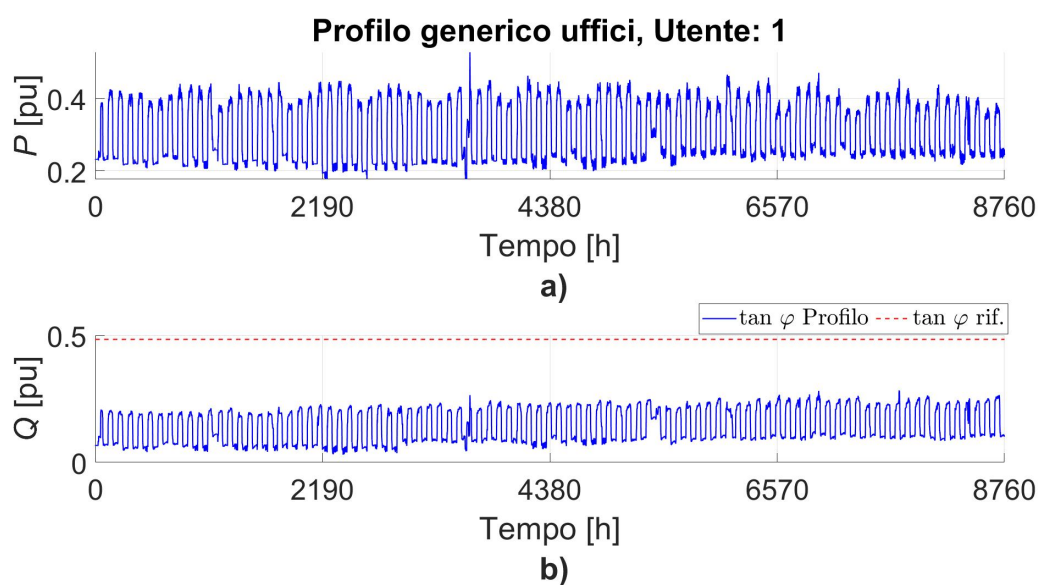


Figura 3.32: Profilo generico uffici, utente 1, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

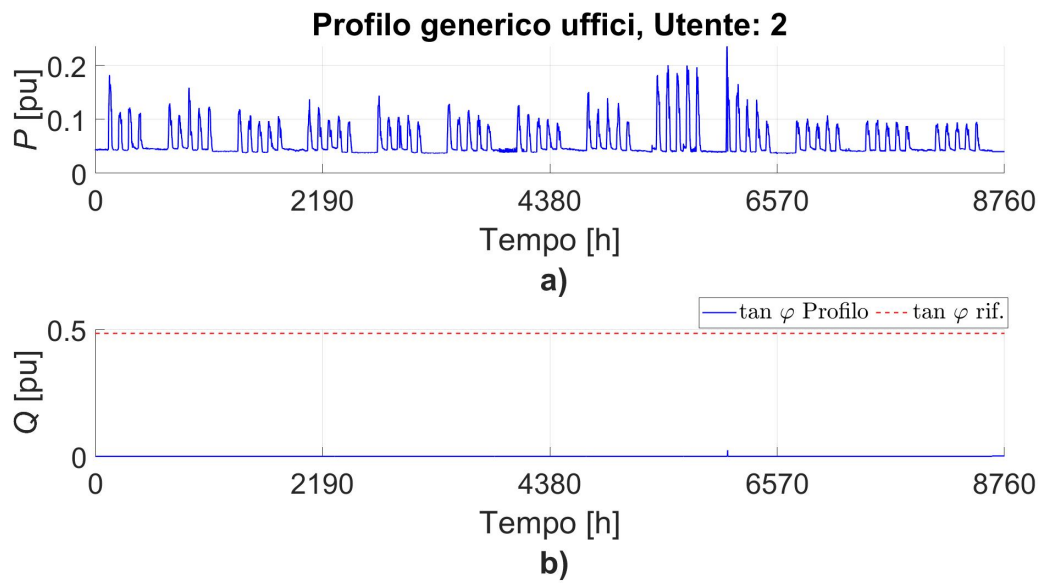


Figura 3.33: Profilo generico uffici, utente 2, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

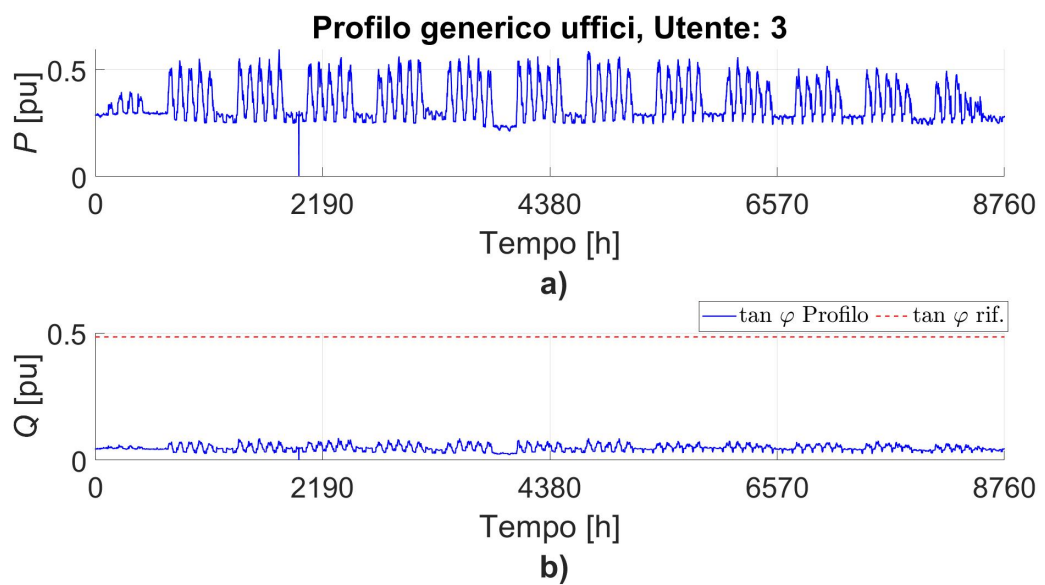


Figura 3.34: Profilo generico uffici, utente 3, potenza attiva pu a), potenza reattiva pu ($\tan \varphi$) b)

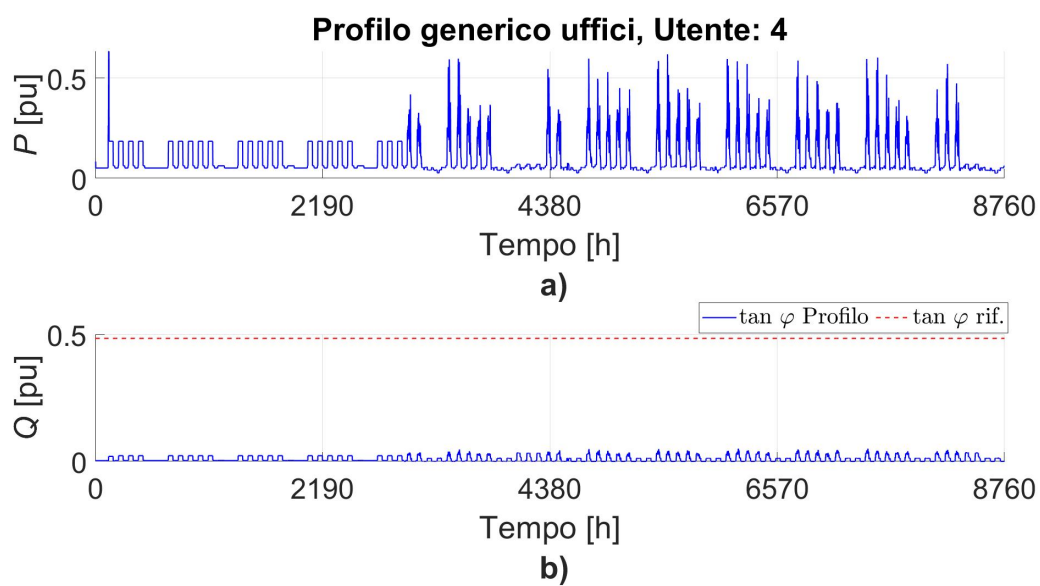


Figura 3.35: Profilo generico uffici, utente 4, potenza attiva pu a), potenza reattiva pu (tan φ) b)

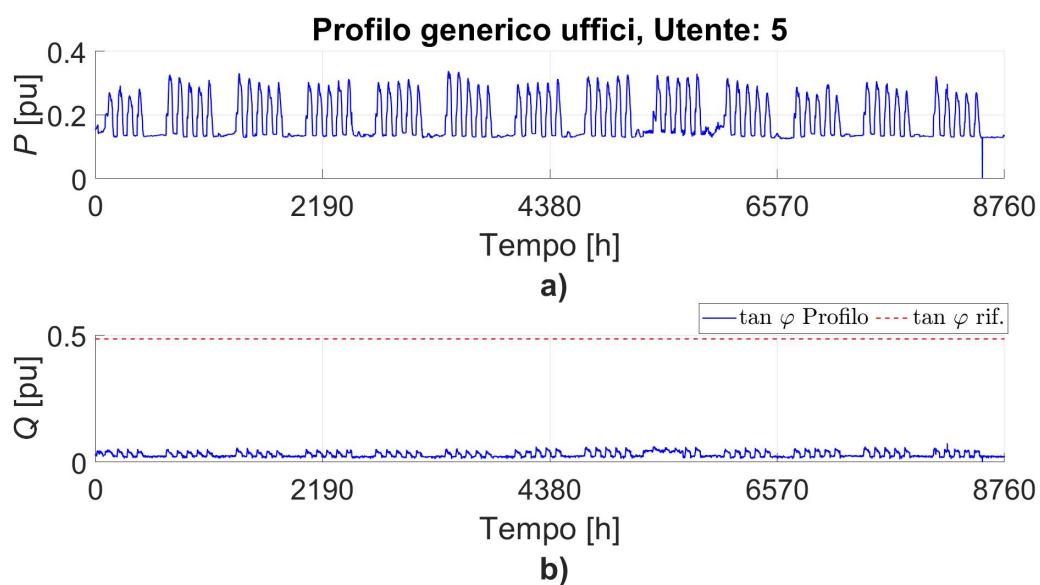


Figura 3.36: Profilo generico uffici, utente 5, potenza attiva pu a), potenza reattiva pu (tan φ) b)

Capitolo 4

Studio della rete elettrica Test

In questo capitolo viene descritta la rete elettrica utilizzata come test per il lavoro di tesi. Tale rete verrà risolta tramite il metodo del Backward-Forward Sweep (BFS). In particolare, tutto ciò che è stato dedotto dall'analisi dati, svolta nel capitolo precedente, troverà ampia utilizzazione in questo capitolo, in quanto bisognerà inserire i vari aggregati nodali all'interno della rete.

4.1 Cenni sulla rete utilizzata

La rete elettrica utilizzata come test è una rete a 55 nodi tratta dalla rete della Città di Torino. Tale rete presenta cinque dorsali, a cui sono stati associati i nomi seguenti (da sinistra verso destra considerando la Figura 4.1):

1. *Magra*;
2. *Brenta*;
3. *Grosso*;
4. *Fiat*;
5. *Chieri*.

Si analizzerà inizialmente la dorsale 4 (Fiat) evidenziata nella Figura 4.1, ma in seguito i medesimi ragionamenti verranno applicati su tutta la rete test.

Nella Figura 4.1 è riportato lo schema della rete test e a fondo di ogni linea è stato riportato il numero, come nell'elenco precedente, che permette di capire la disposizione delle diramazioni presenti:

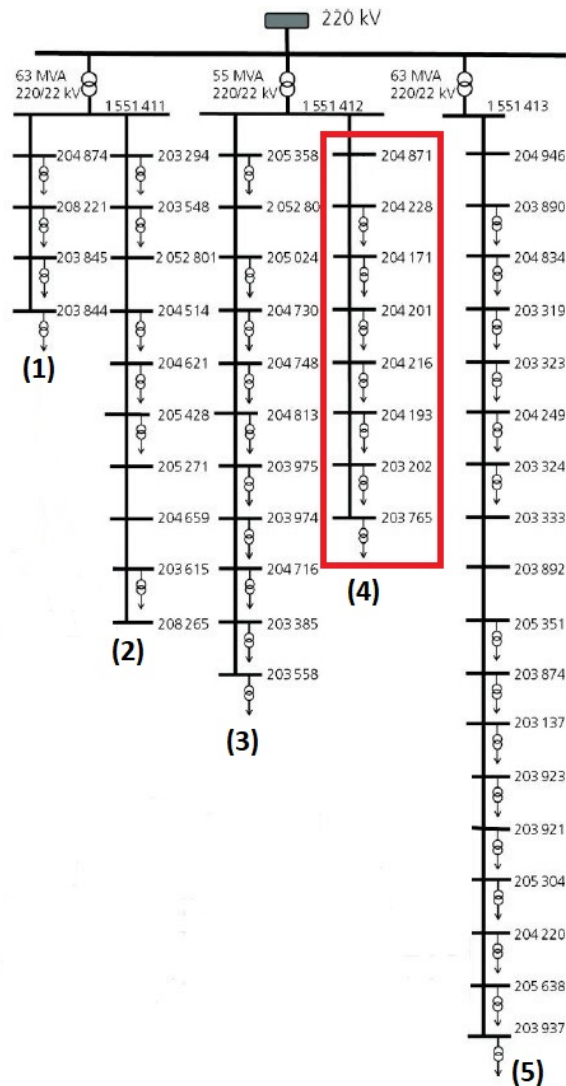


Figura 4.1: Rete Test, Fonte: [9]

Si può notare innanzitutto che la rete presenta tre trasformatori AT/MT¹: due di essi con la stessa potenza nominale di 63 MVA, mentre il terzo, a cui è collegata la dorsale Fiat che andremo ad analizzare, ha potenza nominale pari a 55 MVA.

I parametri relativi alla rete sono stati forniti in per unità, scegliendo come grandezze base $V_b = 22\text{kV}$ e $S_b = 1\text{MVA}$. Tali parametri sono stati quindi riportati nelle relative grandezze chilometriche e moltiplicati per la lunghezza del ramo considerato al fine di ottenere i pa-

¹AT: Alta Tensione; MT: Bassa Tensione

rametri nelle loro unità di misura proprie. Per quanto riguarda la rete Fiat i parametri sono riportati in Tabelle 4.1:

Tabella 4.1: Parametri linea Fiat

Nodo di Partenza	Nodo di Arrivo	R [pu/km]	X [pu/km]	B [pu/km]	L [km]
155141S	204871	2.97521E-04	2.12810E-04	6.34061E-02	2.80100
204871	204228	2.97521E-04	2.12810E-04	6.34061E-02	0.23300
204228	204171	2.97521E-04	2.12810E-04	6.34061E-02	0.35700
204171	204201	4.16072E-04	2.23954E-04	5.58581E-02	0.40900
204201	204216	4.16322E-04	2.23980E-04	5.58462E-02	0.41600
204216	204193	3.88430E-04	2.21074E-04	5.71720E-02	0.28700
204193	203202	3.88430E-04	2.21074E-04	5.71720E-02	0.28000
203202	203765	2.97521E-04	2.15455E-04	5.66905E-02	0.41400

A questo punto, mediante planimetrie, è stata eseguita una ricerca delle varie zone per capire se le zone nelle quali si snodavano le varie dorsali fossero maggiormente residenziali o più commerciali/industriali.

4.2 Studio della dorsale Fiat della rete Test

4.2.1 Ricerca coordinate e creazione aggregati nodali

Considerando la dorsale Fiat, evidenziata nella Figura 4.1, partendo dalle planimetrie della rete è stato possibile associare ad ogni nodo della rete un'indicazione della zona servita, in modo da ricreare un comportamento il più aderente possibile alla realtà. Le conclusioni a cui si è giunti sono riportate nella Tabella 4.2:

Tabella 4.2: Coordinate e stato aree servite dalla dorsale Fiat

N° nodo	Nodo	Coordinate	Stato area
1	204871	N45°4'50.9" E7°40'34.4"	Zona che presenta svariate attività commerciali e supermercato di medie dimensioni, presenza di agglomerato residenziale
2	204228	N45°5'1.9" E7°40'26.0"	Zona perlopiù commerciale e residenziale e supermercato di medie
3	204171	N45°5'8.0" E7°40'31.4"	Zona commerciale presenza inoltre di attività di ristorazione
4	204201	N45°4'59.9" E7°40'45.1"	Zona prevalentemente commerciale presenza di piccoli uffici
5	204216	N45°5'2.2" E7°40'49.4"	Zona prevalentemente

			residenziale
			presenza di parco nelle vicinanze
6	204193	N45°5'3.4" E7°41'0.9"	Zona commerciale presenza di piccoli uffici
7	203202	N45°5'12.0" E7°40'53.3"	Zona scarsamente residenziale, presenza di piccole attività commerciali
8	203765	N45°5'14.4" E7°41'19.5"	Zona residenziale, presenza di piccole attività commerciali

Partendo da queste informazioni mediante il Codice “*Main_valutazione aggregati_nodi*”, riportato in [A.8](#), la rete viene suddivisa in zone con diversi comportamenti:

$$Altamente = \begin{cases} Residenziale \\ Commerciale \\ Industriale \\ Finanziaria (Uffici) \end{cases}$$

$$Mediamente = \begin{cases} Residenziale \\ Commerciale \\ Industriale \\ Finanziaria (Uffici) \end{cases}$$

$$Scarsamente = \begin{cases} Residenziale \\ Commerciale \\ Industriale \\ Finanziaria (Uffici) \end{cases}$$

Ciascun nodo è stato quindi inserito all'interno di una matrice in base al comportamento relativo a quella specifica diramazione. Sono state inserite delle limitazioni, cioè per ciascun nodo è possibile avere solo un comportamento prevalente e uno medio, per gestire al meglio le percentuali relative alle varie presenze che verranno poi associate ai nodi in modo iterativo. Le percentuali sono state quindi sorteggiate in modo casuale definendo i limiti minimi e massimi entro le quali esse dovevano trovarsi. I vincoli inseriti sono i seguenti:

Altamente $\Rightarrow 50\% \div 60\%$;

Mediamente $\Rightarrow 25\% \div 35\%$;

Scarsamente $\Rightarrow 5\% \div 15\%$.

Con un ciclo iterativo è stato dapprima sorteggiato il valore percentuale riferito alla componente "Altamente" presente nel nodo considerato, poi il valore riferito a quella "Mediamente" presente e infine quello legato alla componente o alle componenti "Scarsamente" presenti nel nodo, considerando i vincoli imposti sulle percentuali e l'ulteriore vincolo sulla somma che, ovviamente, deve essere uguale al 100%. Il comando "*ismember*" su *Matlab*[®] fornisce un valore logico in uscita pari a 1 se l'elemento appartiene alla matrice inserita all'interno del comando, o 0 nel caso opposto. Le percentuali vengono quindi generate mediante un ciclo *for*: la prima parte sarà quella relativa alle componenti preponderanti (questo passaggio è riportato nel Codice [A.8.4](#), righe 52 ÷ 65).

Per quanto riguarda invece le componenti che daranno un contributo medio o scarso al profilo nodale, esse verranno gestite dall'operatore "*while*" e vengono imposte con la porzione di Codice [A.8.4](#), alle righe 68 ÷ 101.

I valori percentuali ottenuti sono riportati nella Figura [4.2](#) sotto i campi *RES* (Residenziale), *COMM* (Commerciale), *IND* (Industriale) e *UFF* (Uffici), il campo "*name*" contiene invece il numero del nodo della rete Fiat.

 name	 inBranch	 RES	 COMM	 UFF	 IND	 TOT
204871	'Fiat'	0	0	0	0	0
204228	'Fiat'	32	54	0	14	100
204171	'Fiat'	13	59	0	28	100
204201	'Fiat'	11	51	12	26	100
204216	'Fiat'	51	14	35	0	100
204193	'Fiat'	15	50	35	0	100
203202	'Fiat'	12	52	6	30	100
203765	'Fiat'	58	31	5	6	100

Figura 4.2: Percentuali residenziale (RES), commerciale (COMM), industriale (IND), generico uffici (UFF)

Il nodo 204871 non presenta nessun contributo in quanto tale nodo è un nodo di collegamento. A questo punto, sulla base di queste percentuali, si è passati al calcolo della potenza installata nei vari nodi. Considerando la taglia del trasformatore di cabina MT/BT² di 400 kW e il valore dei carichi installati fornito mediante documentazione, quelli relativi alla rete Fiat sono riportati nella Tabella 4.3:

Tabella 4.3: Potenze installate nodi rete Fiat

Nodo	Potenza utenti residenziali	Potenza utenti non residenziali
	($P_{UT,RES}$) [kW]	($P_{UT,NON RES}$) [kW]
204871	0	0
204228	398	1416.12
204171	335.5	1308.24
204201	95	501.5
204216	203	768
204193	154.5	825
203202	276	1153
203765	280	1334

²BT: Bassa Tensione

È bastato a questo punto moltiplicare le potenze per le varie percentuali precedentemente sorteggiate, per esempio per il primo nodo della rete, cioè il nodo 204228:

$$\begin{aligned} P_{204228}^{\text{RES}} &= \%RES \cdot (P_{\text{UT,RES},204228} + P_{\text{UT,NON RES},204228}) = \\ &= 0.32 \cdot (398 + 1416.12) = 580.5184kW \end{aligned} \quad (4.1)$$

$$\begin{aligned} P_{204228}^{\text{COMM}} &= \%COMM \cdot (P_{\text{UT,RES},204228} + P_{\text{UT,NON RES},204228}) = \\ &= 0.54 \cdot (398 + 1416.12) = 979.6248kW \end{aligned} \quad (4.2)$$

$$\begin{aligned} P_{204228}^{\text{IND}} &= \%IND \cdot (P_{\text{UT,RES},204228} + P_{\text{UT,NON RES},204228}) = \\ &= 0.14 \cdot (398 + 1416.12) = 253.9768kW \end{aligned} \quad (4.3)$$

$$\begin{aligned} P_{204228}^{\text{UFF}} &= \%UFF \cdot (P_{\text{UT,RES},204228} + P_{\text{UT,NON RES},204228}) = \\ &= 0 \cdot (398 + 1416.12) = 0kW \end{aligned} \quad (4.4)$$

Tale procedimento viene poi ripetuto per tutti i nodi della rete considerata e, una volta ottenuti i valori di potenza installata nei nodi per i diversi campi, gli aggregati nodali sono stati calcolati semplicemente moltiplicando il valore di potenza installata ottenuto per l'aggregato residenziale (nel caso della componente domestica) e le altre componenti per il profilo del rispettivo campo. Tutti i contributi sono stati poi sommati, come nell'equazione 4.5:

$$\begin{aligned} \mathbf{P}_{\text{Nodo}}^{\text{TOT}} &= P_{\text{Nodo}}^{\text{RES}} \cdot \mathbf{p}_{\text{pu,agg}}^{\text{RES}} + P_{\text{Nodo}}^{\text{COMM}} \cdot \mathbf{p}_{\text{pu}}^{\text{COMM}} + \\ &\quad + P_{\text{Nodo}}^{\text{IND}} \cdot \mathbf{p}_{\text{pu}}^{\text{IND}} + P_{\text{Nodo}}^{\text{UFF}} \cdot \mathbf{p}_{\text{pu}}^{\text{UFF}} \end{aligned} \quad (4.5)$$

Nodi	inBranch	P_RES	P_IND	P_COMM	P_UFF	Potenza_installata
204871 'Fiat'		0	0	0	0	0
204228 'Fiat'		580.5184	253.9768	979.6248	0	1.8141e+03
204171 'Fiat'		213.6550	460.1800	969.6650	0	1.6435e+03
204201 'Fiat'		65.6150	155.0900	304.2150	71.5800	596.5000
204216 'Fiat'		495.2100	0	135.9400	339.8500	971
204193 'Fiat'		146.9250	0	489.7500	342.8250	979.5000
203202 'Fiat'		171.4800	428.7000	743.0800	85.7400	1429
203765 'Fiat'		936.1200	96.8400	500.3400	80.7000	1.6140e+03

Figura 4.3: Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Fiat

4.2.2 Risoluzione della rete tramite Backward Forward Sweep (BFS)

Gli aggregati nodali precedentemente ottenuti sono stati inseriti all'interno della rete radiale in esame, che è poi stata risolta tramite il metodo di Backward Forward Sweep (BFS). I profili che sono stati utilizzati da qui in avanti sono quelli ottenuti nel paragrafo 4.2.1. Inizialmente il BFS è stato creato per la sola dorsale Fiat, ma il codice *Matlab*[®] è stato poi esteso alla rete completa, comprendente tutte le altre dorsali. La porzione di rete considerata è la stessa evidenziata nella Figura 4.1.

Nel seguito vengono esposte a grandi linee le fasi del metodo di risoluzione della rete. Il metodo BFS [28] è un metodo di tipo iterativo che consta di due fasi:

1. **Backward:** vengono fissate le tensioni dei carichi e si calcolano le correnti di linea (si parte quindi dai punti terminali della rete e si va verso la radice);
2. **Forward:** una volta note le correnti si ricalcolano le tensioni sui carichi, quindi si procede dalla radice della rete verso i punti terminali.

Queste due operazioni vengono eseguite in maniera iterativa finché la differenza fra la tensione di carico valutata nell'iterazione attuale e quella calcolata durante l'iterazione precedente non risulta inferiore ad una imposta. Quando tale condizione risulta verificata, si ha la convergenza del sistema e si è quindi giunti alla soluzione finale.

Il Codice creato prevede quindi una prima fase relativa alla conoscenza della rete: in questa fase verrà creata la matrice delle incidenze $\mathbf{L}$ riguardante la sola linea in esame e la sua inversa, la matrice $\mathbf{\Gamma}$. La matrice $\mathbf{L}$ e la sua inversa, per reti poco estese, possono essere ottenute

anche per ispezione visiva, usando la classica codifica che prevede, nel caso della matrice delle incidenze, che le colonne individuino i nodi e le righe i rami, quindi, scelta una riga, si darà all'elemento un valore -1, 0 o 1 nel modo seguente:

- -1 se il nodo è un nodo d'arrivo per il ramo considerato;
- 1 se il nodo è un nodo di partenza per il ramo considerato;
- 0 se il ramo considerato non contiene il nodo.

Nel caso in esame, considerando la struttura della rete Fiat, si ottiene per essa una matrice 9x9 che è riportata in (4.6):

$$\mathbf{L} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad (4.6)$$

Tale matrice è stata generata all'interno del codice sfruttando la struttura dei parametri della rete, presenti nel Codice “*init_distribution_network_v1*” che era stato fornito e che comprende i dati relativi a tutte le dorsali della rete test completa e ai parametri dei trasformatori presenti. Per la rete Fiat per esempio i parametri sono quelli riportati nella Tabella 4.4:

Tabella 4.4: Parametri linea Fiat

Nodo di Partenza	Nodo di Arrivo	R [Ω]	X [H]	B [F]	L [km]
1551412	204871	0.144000164	0.000327859	4.16999E-07	2.801
204871	204228	0.144000164	0.000327859	4.16999E-07	0.233
204228	204171	0.144000164	0.000327859	4.16999E-07	0.357
204171	204201	0.201378848	0.000345027	3.67358E-07	0.409
204201	204216	0.201499848	0.000345067	3.6728E-07	0.416
204216	204193	0.18800012	0.00034059	3.75999E-07	0.287
204193	203202	0.18800012	0.00034059	3.75999E-07	0.28
203202	203765	0.144000164	0.000331933	3.72833E-07	0.414

Su *Matlab*[®] tutti i valori dei parametri sono stati salvati all'interno di una struttura “*grid*”: nel campo definito “*name*”, gli stessi valori sono stati salvati anche all'interno delle strutture “*inNode*”, se appartenenti alla colonna “*Nodo di partenza*” o in “*outNode*” se appartenenti alla colonna “*Nodo di arrivo*”, quindi è stata generata la “*grid.matrL*”, cioè la matrice delle incidenze voluta.

Scorrendo il campo relativo ai nomi dei nodi della rete, precedentemente esposto, tramite la funzione *find* vengono cercate le similitudini nei campi “*inNode*” o “*outNode*” e ad essi viene associato valore 1 o -1. Tale passaggio è visibile nel Codice [A.9.4](#), righe 114 ÷ 120.

A questo punto si è proceduto alla definizione delle altre matrici necessarie ai calcoli, ovvero la matrice delle impedenze $\mathbf{Z}_{\text{branch}}$ e delle ammettenze trasversali $\mathbf{Y}_{\text{sh}}$. Sono quindi stati calcolati i parametri della rete, facendo attenzione per quanto riguarda il trasformatore, a riportare i parametri espressi in p.u. nelle proprie unità di misura (equazioni (4.7) ÷ (4.10)).

$$R_t = R_2 \cdot \frac{V_b^2}{S_b} \quad (4.7)$$

$$X_t = X_2 \cdot \frac{V_b^2}{S_b} \quad (4.8)$$

$$Z_t = R_t + jX_t \quad (4.9)$$

$$Y_t = (G_{m,2} - jB_{m,2}) \cdot \frac{S_b}{V_b^2} \quad (4.10)$$

dove con R_t e X_t sono state definite la resistenza e la reattanza del trasformatore, Z_t rappresenta l'impedenza longitudinale e Y_t l'ammettenza trasversale.

R_2 , X_2 , $G_{m,2}$ e $B_{m,2}$ sono i parametri del trasformatore forniti e V_b ed S_b le relative grandezze di base.

Si segnala che “*Parametri_linea_ohm(i).Ls*” all'interno del Codice [A.9.4](#), alla riga 191, è stato moltiplicato solo per 1j in quanto nel file dei parametri era già stato moltiplicato per la pulsazione ω .

I parametri della linea invece, essendo definiti in grandezze chilometriche, vengono precedentemente moltiplicate per la lunghezza della linea e salvate all'interno della stessa struttura. Il passaggio è mostrato sempre nel Codice [A.9.4](#), nelle righe 196 ÷ 200.

A questo punto vengono create le matrici in per unità delle impedenze dei rami e delle ammettenze.

$$\mathbf{Z}_{\text{branch,pu}} = \frac{\mathbf{Z}_{\text{branch}}}{Z_b} \quad (4.11)$$

$$\mathbf{Y}_{\text{sh,pu}} = \frac{\mathbf{Y}_{\text{sh}}}{Z_b^{-1}} \quad (4.12)$$

I parametri trasversali vengono quindi aggregati sfruttando la matrice delle incidenze, come mostrato nel Codice [A.9.4](#), riga 221.

$$\mathbf{Y}_{\text{tot,pu}} = |\mathbf{L}^T| \cdot \frac{\mathbf{Y}_{\text{sh,pu}}}{2} \quad (4.13)$$

in accordo con il modello della rete mostrato in Figura [4.4](#):

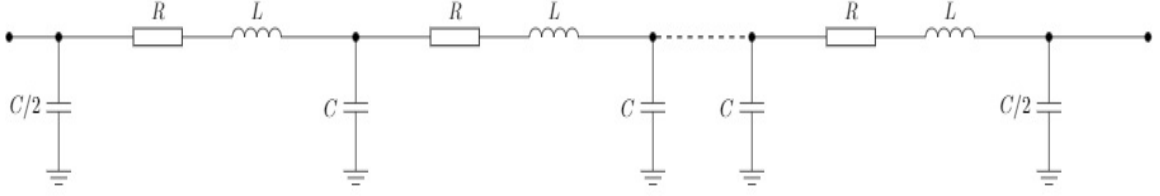


Figura 4.4: Modello della linea

Viene quindi creato il ciclo di *BFS*, dove la “*Scar_{pu}*” rappresenta l’energia apparente che si ottiene istante per istante dai profili di potenza attiva e reattiva, cioè:

$$\mathbf{S}_{\text{car,pu}} = \mathbf{P}_{\text{car,pu}} + j \cdot \mathbf{Q}_{\text{car,pu}} \quad (4.14)$$

Il ciclo BFS sarà quindi composto da diverse iterazioni, durante la generica iterazione k si avrà [28]:

- **fase backward**: si calcola il vettore delle correnti nodali $\mathbf{i}_S^{(k)}$, le componenti saranno:

$$I_{S,i}^{(k)} = \frac{S_{\text{car,pu},i}^*}{V_i^{*(k-1)}} \quad (4.15)$$

essendo carichi a potenza assegnata. Si calcola quindi il vettore $\mathbf{i}_b$ delle correnti di linea:

$$\mathbf{i}_b^{(k)} = \mathbf{\Gamma}^T \mathbf{i}_S^{(k)} \quad (4.16)$$

- **fase forward**: si calcola la tensione in ogni nodo partendo dallo slack:

$$\mathbf{v}^{(k)} = V_0 \mathbf{1} - \mathbf{\Gamma} \mathbf{Z}_{\text{branch}} \mathbf{i}_b^{(k)} \quad (4.17)$$

dove $\mathbf{1}$ è un vettore unitario e $\mathbf{\Gamma} \mathbf{Z}_{\text{branch}} \mathbf{i}_b^{(k)}$ rappresenta per ogni nodo le cadute di tensione che si hanno fra nodo di alimentazione e il nodo considerato. Il processo verrà ripetuto finché non risulterà verificato il criterio di convergenza calcolato per ogni nodo:

$$\max_i \frac{|V_i^{(k+1)} - V_i^{(k)}|}{V_i^{(k)}} < \epsilon \quad i = 1, \dots, n \quad (4.18)$$

con ϵ indicante la tolleranza scelta (tipicamente $10^{-5} \div 10^{-6}$).

Avendo imposto l'uscita del generatore equivalente di Thevenin (rappresentante la rete a monte esercita a 220 kV) come nodo slack, fissando per esso la tensione ad 1 per unità, in Figura 4.5 si riporta l'andamento delle tensioni nei nodi della rete, valutato per un giorno, quindi 96 quarti d'ora.

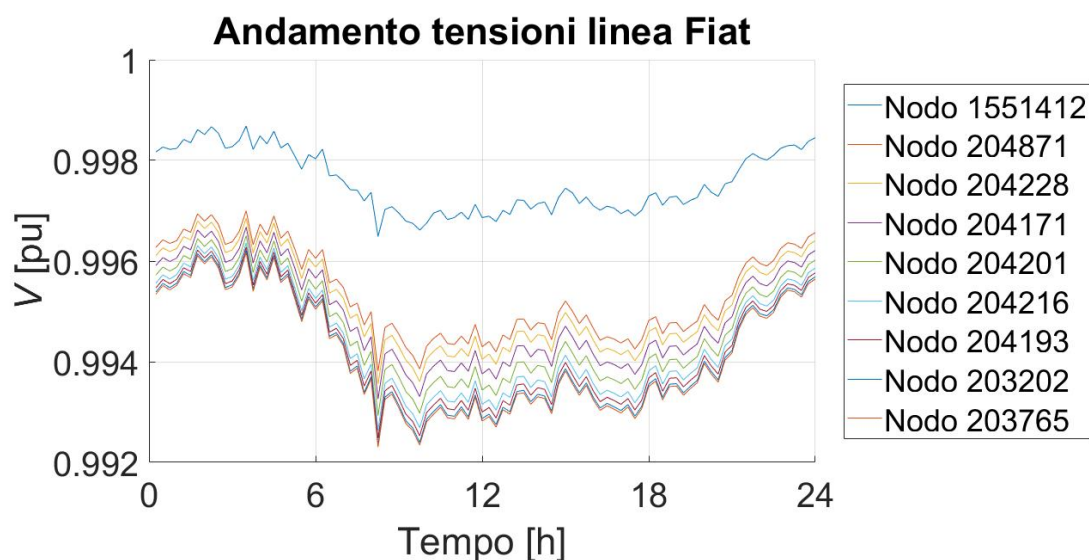


Figura 4.5: Andamento tensioni linea Fiat

Uno dei vincoli più importanti da rispettare è quello sul livello di carico, definito come

$$\dot{\mathbf{i}}_b = \frac{|\mathbf{i}_b|}{\mathbf{i}_{\max,b}} \quad (4.19)$$

dove $\mathbf{i}_b$ è il vettore delle correnti di linea e $\mathbf{i}_{\max,b}$ il vettore contenente i valori limite di corrente per ogni linea.

Il livello di carico dovrà essere minore di 1 p.u. (anche se è ammesso un valore pari ad 1 o massimo 1.1 per un periodo limitato). Nella pratica si cerca di mantenersi al di sotto del 50% per evitare di sollecitare termicamente il cavo. L'andamento del livello di carico per la rete Fiat è riportato in Figura 4.6:

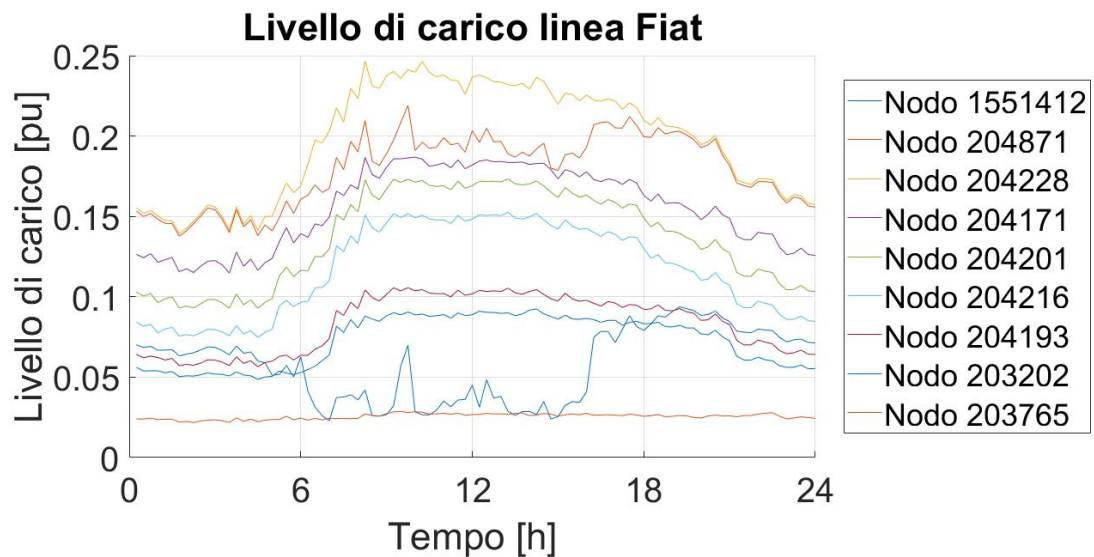


Figura 4.6: Livello di carico linea Fiat

A questo punto, il procedimento fin qui esposto è stato ripetuto per le altre dorsali della rete test.

4.3 Studio delle altre dorsali della rete test

4.3.1 Creazione aggregati per le altre dorsali della rete

In questo paragrafo vengono esposte le tabelle relative alle coordinate geografiche e alle caratteristiche dei nodi delle altre linee della rete, il procedimento per il calcolo degli aggregati nodali è lo stesso esposto nel paragrafo [4.2.1](#) per la rete Fiat.

- Dorsale Magra

Tabella 4.5: Coordinate e stato aree dorsale Magra

N° nodo	Nodo	Coordinate	Stato area
1	204874	N45°6'53.7" E7°43'59.4"	Zona industriale, presenza di attività commerciali medio/grandi
2	204221	N45°6'45.4" E7°43'49.6"	Zona prettamente industriale con attività commerciali sparse
3	203845	N45°5'58.0" E7°42'46.8"	Zona che presenta piccole attività commerciali e agglomerati residenziali sparsi
4	203844	N45°4'59.9" E7°40'45.1"	Zona prevalentemente residenziale, presenza di piccoli uffici

 Nodi	 inBranch	 P_RES	 P_IND	 P_COMM	 P_UFF	 Potenza_installata
204874	"Magra"	0	0	0	0	0
208221	"Magra"	0	0	0	0	0
203845	"Magra"	84.8400	193.9200	327.2400	0	606
203844	"Magra"	242.4900	53.4300	0	115.0800	411

Figura 4.7: Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Magra

- Dorsale Brenta

Tabella 4.6: Coordinate e stato aree dorsale Brenta

N° nodo	Nodo	Coordinate	Stato area
1	203294	N45°6'15.5" E7°41'3.4"	Zona di tipo commerciale presenza di piccole attività industriali
2	203548	N45°6'21.5" E7°41'6.1"	Zona di tipo commerciale, sono presenti piccoli agglomerati residenziali
3	2052801	N45°6'15.7" E7°40'59.3"	Zona perlopiù residenziale, presenza di uffici e piccole attività commerciali
4	204514	N45°6'18.8" E7°40'7.7"	Zona prevalentemente residenziale, presenza di uffici
5	204621	N45°6'28.8" E7°39'10.0"	Zona commerciale e di piccoli laboratori industriali
6	205248	N45°6'37.7" E7°38'59.3"	Zona che presenta svariate attività industriali
7	205271	N45°6'33.7" E7°39'31.5"	Zona perlopiù residenziale presenza di uffici e piccole

- 8 **204659** N45°6'39.2" E7°39'38.0" Zona residenziale,
presenza di piccole
attività commerciali
- 9 **203615** N45°6'46.5" E7°39'20.9" Zona residenziale,
presenza di uffici e piccole
attività commerciali
- 10 **208265** N45°6'54.5" E7°39'16.1" Zona che presenta
attività industriali di medie dimensioni
-

 Nodi	 inBranch	 P_RES	 P_IND	 P_COMM	 P_UFF	 Potenza_installata
203294	"Brenta"	258.7200	646.8000	942.4800	0	1848
203548	"Brenta"	345.4500	0	493.5000	0	838.9500
2052801	"Brenta"	0	0	0	0	0
204514	"Brenta"	280.2800	0	183.2600	75.4600	539.0000
204621	"Brenta"	0	308.6050	547.5250	139.3700	995.5000
205428	"Brenta"	0	258.6600	148.4900	71.8500	479
205271	"Brenta"	0	0	0	0	0
204659	"Brenta"	0	0	0	0	0
203615	"Brenta"	362.7320	0	36.8880	215.1800	614.8000
208265	"Brenta"	0	0	0	0	0

Figura 4.8: Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Brenta

- Dorsale Grosso

Tabella 4.7: Coordinate e stato aree dorsale Grosso

N° nodo	Nodo	Coordinate	Stato area
1	205358	N45°6'10.9" E7°41'51.0"	Zona residenziale, piccole attività commerciali sparse
2	2052802	N45°6'17.7" E7°40'58.9"	Zona che presenta agglomerati residenziali sparsi in tutta l'area e piccoli uffici
3	205024	N45°6'12.6" E7°40'58.3"	Zona prevalentemente residenziale e scarsamente commerciale
4	204730	N45°6'13.1" E7°40'52.6"	Zona residenziale, presenza di attività commerciali
5	204748	N45°6'13.6" E7°40'44.3"	Zona perlopiù residenziale, attività commerciali di piccole dimensioni
6	204813	N45°6'8.4" E7°40'43.4"	Zona con alto numero di agglomerati residenziali
7	203975	N45°6'5.3" E7°40'47.1"	Zona perlopiù residenziale, attività commerciali di piccole dimensioni
8	203974	N45°6'4.0" E7°40'42.7"	Zona residenziale,

presenza di attività commerciali

- 9 **204716** N45°6'1.1" E7°40'50.8" Zona a concentrazione
residenziale, attività commerciali
- 10 **203385** N45°5'59.1" E7°40'46.2" Zona industriale, presenza
di attività commerciali
- 11 **203558** N45°5'59.6" E7°40'42.0" Zona commerciale, presenza
di piccoli uffici

 Nodi	 inBranch	 P_RES	 P_IND	 P_COMM	 P_UFF	 Potenza_installata
205358	"Grosso"	86.1000	12.9150	37.3100	7.1750	143.5000
2052802	"Grosso"	0	0	0	0	0
205024	"Grosso"	933.9324	256.3736	640.9340	0	1.8312e+03
204730	"Grosso"	274.3400	0	141.9000	56.7600	473
204748	"Grosso"	762.9900	0	381.4950	171.0150	1.3155e+03
204813	"Grosso"	953.6824	0	131.5424	460.3984	1.5456e+03
203975	"Grosso"	443.7900	0	207.1020	828.4080	1.4793e+03
203974	"Grosso"	87.3090	0	203.7210	291.0300	582.0600
204716	"Grosso"	441.0236	99.8544	291.2420	0	832.1200
203385	"Grosso"	179.6544	838.3872	479.0784	0	1.4971e+03
203558	"Grosso"	0	631.2880	1.0101e+03	162.3312	1.8037e+03

Figura 4.9: Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Grosso

- Dorsale Chieri

Tabella 4.8: Coordinate e stato aree dorsale Chieri

N° nodo	Nodo	Coordinate	Stato area
1	204946	N45°5'40.5" E7°43'1.9"	Zona residenziale, piccole attività commerciali sparse
2	203890	N45°5'29.5" E7°43'40.6"	Zona industriale di medie dimensioni
3	204834	N45°5'16.2" E7°43'44.2"	Zona commerciale, presenza di piccoli agglomerati residenziali
4	203319	N45°5'9.8" E7°44'0.9"	Zona mediamente residenziale, presenza di piccoli uffici
5	203323	N45°4'51.2" E7°43'54.4"	Zona prevalentemente residenziale, sono presenti piccole attività commerciali
6	204249	N45°4'33.8" E7°43'52.2"	Zona residenziale, presenza di uffici sparsi nella zona
7	203324	N45°4'28.4" E7°43'36.2"	Zona residenziale, presenza di uffici sparsi nella zona
8	203333	N45°4'35.7" E7°43'55.3"	Zona altamente residenziale

9	203892	N45°4'28.2" E7°44'12.2"	Zona con agglomerati residenziali in strada privata
10	205351	N45°4'16.3" E7°43'49.6"	Zona con agglomerati residenziali in strada privata
11	203874	N45°3'28.4" E7°44'27.8"	Zona con agglomerati residenziali in strada privata
12	203137	N45°3'16.2" E7°44'14.4"	Zona con agglomerati residenziali in strada privata
13	203923	N45°3'13.4" E7°44'51.3"	Zona con agglomerati residenziali in strada privata
14	203921	N45°2'57.1" E7°45'12.0"	Zona con agglomerati residenziali in strada privata
15	205304	N45°3'2.2" E7°43'22.5"	Zona con agglomerati residenziali in strada privata
16	204220	N45°2'25.5" E7°43'56.4"	Zona lontana dal centro abitato, presenza di piccole villette
17	25638	N45°2'18.7" E7°44'5.7"	Zona lontana dal centro abitato, presenza di piccole villette

Nodi	inBranch	P_RES	P_IND	P_COMM	P_UFF	Potenza_installata
204946 "Chieri"		0	0	0	0	0
203890 "Chieri"		75.1250	180.3000	0	45.0750	300.5000
204834 "Chieri"		84.8100	0	141.3500	30.8400	257
203319 "Chieri"		692.4360	0	97.1840	425.1800	1.2148e+03
203323 "Chieri"		421.9600	37.6750	256.1900	0	715.8250
204249 "Chieri"		221.9265	66.5780	0	155.3486	443.8530
203324 "Chieri"		216.2700	44.0550	0	140.1750	400.5000
203333 "Chieri"		0	0	0	0	0
203892 "Chieri"		90.1250	21.6300	205.4850	43.2600	360.5000
205351 "Chieri"		125.6750	70.8350	31.9900	0	228.5000
203874 "Chieri"		262.3500	173.2500	59.4000	0	495
203137 "Chieri"		300.1350	88.2750	200.0900	0	588.5000
203923 "Chieri"		501.6000	0	96.8000	281.6000	880
203921 "Chieri"		370.8326	0	75.5400	240.3544	686.7270
205304 "Chieri"		475.6265	294.8884	85.6128	95.1253	951.2530
204220 "Chieri"		217.8800	62.8500	138.2700	0	419
205638 "Chieri"		329.7000	38.4650	27.4750	153.8600	549.5000
203937 "Chieri"		712.4700	447.0400	167.6400	69.8500	1397

Figura 4.10: Potenza installata, per i diversi campi di applicazione, nei vari nodi della rete Chieri

4.3.2 Risoluzione della rete completa mediante BFS

La rete è stata interamente risolta tramite BFS e per ogni dorsale sono state fatte diverse valutazioni per verificare il funzionamento della rete. In particolare è stato valutato il livello di carico (come era già stato fatto per la dorsale Fiat) e il carico totale collegato alle linee. È importante infatti che il carico collegato alle dorsali che fanno fronte ad un trasformatore non abbiano una potenza superiore a quella del trasformatore che alimenta la linea. Ad esempio, considerando le dorsali collegate al trasformatore di potenza apparente nominale $S_{n,trasf} = 55\text{MVA}$, cioè le dorsali Grosso e Fiat in Figura 4.1, è importante che la potenza massima delle due linee non sia superiore alla potenza nominale del trasformatore collegato,

questo al fine di non mandare quest'ultimo in saturazione. Avendo ottenuto la corrente a valle del trasformatore, tale vincolo può essere valutato facilmente. Nel caso considerato si ha:

$$\mathbf{S}_{\text{Grosso,Fiat}} = \mathbf{e}_s \cdot \mathbf{i}_b^* \quad (4.20)$$

$$\max(|\mathbf{S}_{\text{Grosso,Fiat}}| \cdot S_B) < S_{n,\text{trasf}} = 55\text{MVA} \quad (4.21)$$

dove $\mathbf{e}_s$ è il vettore della tensione associata allo slack, $\mathbf{i}_b^*$ è il vettore della somma delle correnti di linea delle dorsali considerate ed S_B la potenza apparente di base.

In maniera analoga per le altre linee. Sono state valutate inoltre le perdite, sia in valore assoluto (equazione (4.22)) che relativo (equazione (4.23)).

$$E_{\text{persa}} = \tau \cdot \sum_{b=1}^B \sum_{t=1}^T R_b \cdot I_{b,t}^2 \quad (4.22)$$

dove, B è il numero di rami, τ è il numero di intervalli di tempo espresso in ore, R_b è la resistenza del ramo b ed $I_{b,t}$ è la corrente che scorre nel ramo b nel tempo t .

$$E_{\text{persa}} \% = \frac{E_{\text{loss}}}{\tau \cdot \sum_{k=1}^K \sum_{\text{tipo}=(\text{res},\text{comm},\text{ind},\text{uff})} P_{\text{nom},k}^{(\text{tipo})} \cdot P_{\text{pu},t}^{(\text{tipo})}} \quad (4.23)$$

dove, K è il numero di nodi, (tipo) è la variabile che indica le diverse tipologie di carico (residenziale, commerciale, industriale, generico uffici), $P_{\text{nom},k}^{(\text{tipo})}$ rappresenta la potenza nominale del carico (tipo) nel nodo k e $P_{\text{pu},t}^{(\text{tipo})}$ Rappresenta il profilo di carico espresso in per unità del carico (tipo) nel tempo t . Infine, è stato valutato il fattore di potenza delle linee, per verificare che esso si mantenesse in un valore accettabile. Questo è stato valutato facilmente nel modo seguente, per esempio per la linea Fiat:

$$\mathbf{S}_{\text{Fiat}} = \mathbf{e}_s \cdot \mathbf{i}_b^* \quad (4.24)$$

$$\tan\varphi = \frac{\text{Im}(\mathbf{S}_{\text{Fiat}})}{\text{Re}(\mathbf{S}_{\text{Fiat}})} \quad (4.25)$$

$$\cos\varphi = \arctan(\tan\varphi) \quad (4.26)$$

A questo punto si è passati all’inserimento nella rete dei profili di potenza da generazione fotovoltaica, argomento che verrà dettagliatamente esposto nel Capitolo [5](#).

Capitolo 5

Profili di generazione

In questo capitolo verranno discusse le problematiche relative alla presenza della generazione distribuita all'interno della rete elettrica. I profili di rinnovabile utilizzati nel lavoro di tesi sono stati generati da un modello già esistente in letteratura che verrà discusso brevemente all'interno del capitolo. Successivamente si procederà all'inserimento dei profili all'interno della rete elettrica test, precedentemente definita.

5.1 Problematiche legate alla presenza di risorse distribuite nella rete elettrica

Negli anni '80 e '90, il concetto di sistemi di generazione e accumulo distribuiti era già presente, ma la mancanza di tecnologie a costi accessibili agli utenti ne ha limitato le applicazioni. Dalla fine degli anni '90 fino al 2010 è stato registrato un aumento notevole delle installazioni, che hanno avuto un impatto significativo sul funzionamento delle reti elettriche. La nomenclatura con le quali vengono definite queste risorse è piuttosto varia, si parla di “embedded generation” in ambito anglosassone, o di “dispersed generation” in Nord America [29].

Lo standard IEEE 1547 [30] ha proposto la seguente definizione:

“Per generazione distribuita si intende una sorgente elettrica connessa direttamente alla rete di distribuzione, o al lato cliente del contatore di energia.”

Le risorse di energia distribuita (DER), possono essere espresse come somma di tre componenti:

$$DER = DG + DR + DS \quad (5.1)$$

Dove:

- *DG*: Generazione Distribuita, *Distributed Generation*, sono risorse locali di generazione poste sulla rete di distribuzione o sulla rete del cliente;
- *DR*: Gestione della domanda, *Demand Response*, sono risorse per il controllo della domanda, riguardano principalmente i sistemi di accumulo o di gestione del carico elettrico;
- *DS*: Accumulo distribuito, *Distributed Storage*, riferito appunto ai sistemi di accumulo (*storage*) distribuito.

A questi concetti si aggiungono le fonti di energia rinnovabile (*FER*).

La rete test simulata fino a questo momento non include elementi di generazione distribuita: è da considerarsi quindi come una rete passiva. Tuttavia, le reti reali presentano una forte componente attiva legata, per esempio, ad impianti di energia rinnovabile. Questo rappresenta un problema per certi aspetti della rete, nel luglio del 2011 l'ENTSO-E, ovvero la rete europea dei gestori di sistemi di trasmissione di energia elettrica, ha evidenziato come l'aumento della generazione fotovoltaica possa rendere critica la sicurezza dei sistemi elettrici nazionali, basti pensare ad una disconnessione contemporanea di un grande numero di impianti che causerebbe una discontinuità nella produzione. È stato quindi necessario modificare le normative nazionali imponendo che i sistemi di protezione di interfaccia e gli inverter rimangano connessi alla rete per variazioni di frequenza tra 47.5 e 51.5 Hz [28].

5.2 Costruzione dei profili di generazione fotovoltaica

5.2.1 Cenni sul codice di creazione dei profili di irradianza

Per generare i profili di irradianza è stato utilizzato uno codice *Matlab*[®] discusso in [22]. Tale codice fornisce in uscita profili di irradianza di lunghezza selezionabile e per coordinate geografiche definibili da parte dell'utente. L'irradianza solare presenta variazioni che sono legate alla dinamica delle nuvole, alla presenza di pioggia e al trasporto di agenti inquinanti presenti nell'aria.

Queste variazioni vanno ad influenzare direttamente la quantità di potenza generata dagli impianti fotovoltaici, andando ad impattare, quindi, sui benefici degli stessi.

Si rivela quindi necessario creare dei modelli utili per predire l'irradianza e che possano poi essere estesi a qualunque località scelta. Vengono utilizzati dei metodi di tipo probabilistico, che selezionano un indice legato alla presenza di nuvole basato su osservazioni reali o determinato da distribuzioni probabilistiche. Il codice utilizzato dapprima genera la copertura di nuvole con la risoluzione di 1 minuto e successivamente calcola l'irradianza, con la medesima risoluzione, basandosi sul fatto che il minuto in questione sia nuvoloso o limpido.

Come verrà esposto nel Paragrafo 5.2.2, sono state inserite le coordinate di latitudine e longitudine relative a Torino e sono stati quindi generati profili di irradianza annuali. In Figura 5.1 si riporta un breve estratto relativo alla copertura di nuvole e di irradianza giornaliera, per un giorno selezionato casualmente:

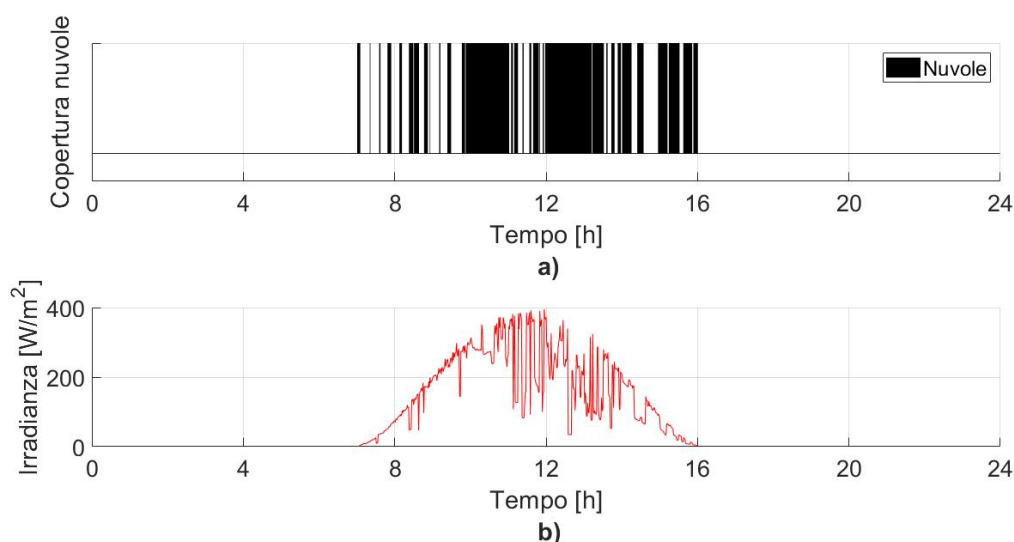


Figura 5.1: Copertura nuvole a) e irradianza b), singola giornata, Torino

Nel primo grafico le zone nere rappresentano la copertura di nuvole lungo l'asse dei tempi. Il secondo grafico rappresenta l'irradianza globale riferita ad un piano orizzontale.

5.2.2 Generazione di profili di irradianza da coordinate geografiche

Poiché il caso studio è incentrato su una porzione della rete di Torino, i profili di irradianza sono stati generati inserendo le coordinate geografiche della città, riportati in Tabella 5.1.

Tabella 5.1: Coordinate città di Torino

Latitudine	N45°4'0''	45.07°
Longitudine	E7°42'0''	7.7°
Altitudine	239 m s.l.m	

Si sono poi selezionati i giorni di inizio e di fine riferiti al periodo in cui generare i profili voluti. Nel nostro caso eravamo interessati a profili annuali ed è stata selezionata come prima giornata il primo gennaio del 2017 mentre come giorno finale il 31 dicembre 2017. La scelta dell'anno è dipesa dai dati a disposizione internamente al programma.

Il codice, una volta inseriti i dati di ingresso, genera 10 profili relativi a 10 impianti fotovoltaici disposti in zone casuali all'interno di un piano che racchiude le coordinate specificate. La disposizione degli impianti è riportata in Figura 5.2:

Disposizione delle case nel dominio spaziale

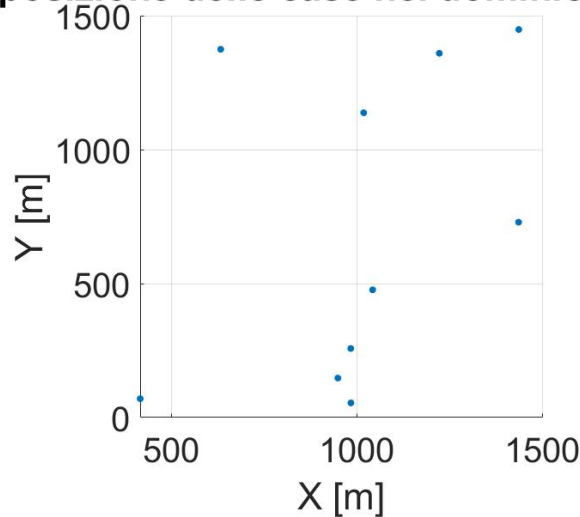


Figura 5.2: Disposizione spaziale impianti fotovoltaici, Torino

Un dettaglio dei profili di irradianza è visualizzabile in Figura 5.3:

Profili di irradianza, Torino

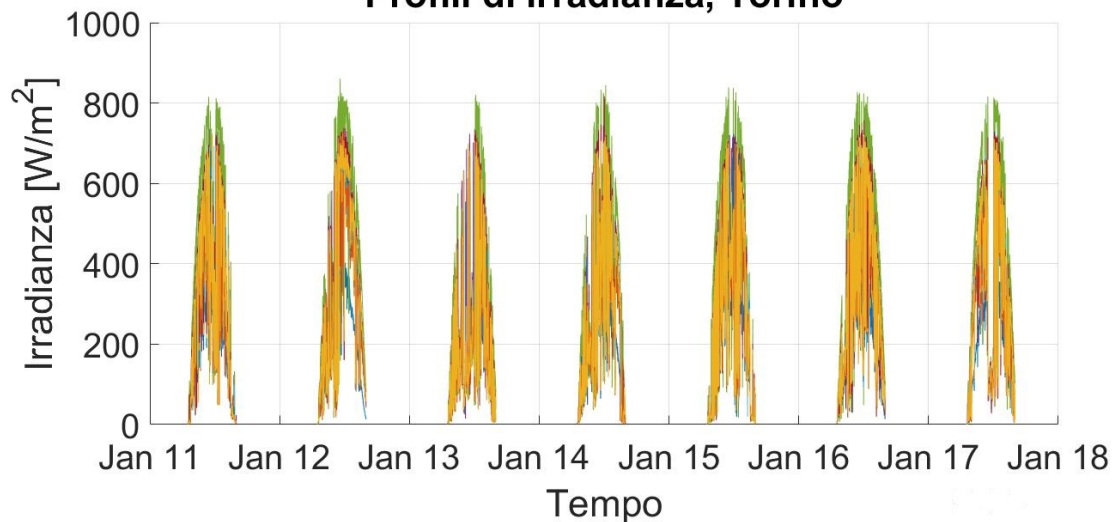


Figura 5.3: Dettaglio profili di irradianza Torino (giornate dall'11 al 17 gennaio)

Da questi profili di irradianza è possibile ottenere i profili di potenza lato rete espressi in per unità rispetto alla potenza nominale dell'impianto. A partire dall'irradianza espressa in $[W/m^2]$ è possibile ricavare il profilo in per unità espresso come rapporto P_{AC} / P_{nom}

tramite le formule (5.2), (5.3) e (5.4) [31] :

$$T_c = T_a + \frac{NOCT - 20}{800} \cdot G \quad (5.2)$$

$$\eta_{th} = 1 - \alpha_{th} \cdot (T_c - 25) \quad (5.3)$$

$$\frac{P_{AC}}{P_{nom}} = \eta_{DC-AC} \cdot \frac{G}{1000} \cdot \eta_{th} \quad (5.4)$$

dove T_c è la temperatura stimata del pannello, T_a è la temperatura dell'aria, espresse in $^{\circ}\text{C}$, $NOCT$ è la temperatura operativa nominale delle celle fotovoltaiche, imposta uguale a 45°C , η_{th} è l'efficienza termica ed indica la riduzione della produzione dovuta alla temperatura del pannello fotovoltaico, G è l'irradianza, espressa in $[\text{W}/\text{m}^2]$, α_{th} è il coefficiente di perdita legato alla temperatura, posto uguale a 0.45% (da dati sperimentali) e infine η_{DC-AC} rappresenta l'efficienza legata ai cavi, alle connessioni e all'inverter presente, posto uguale a 0.828. I valori relativi alla temperatura dell'aria sono stati ricavati grazie al tool online PVGIS [32]: Trattandosi di un profilo orario di temperatura annuale, da questo è stato ottenuto un profilo con time-step di 15 minuti. Mediante un semplice ciclo for sono stati poi ottenuti i profili in p.u. voluti. La porzione di Codice è riportata in 5.2.1 e fa parte del codice completo mostrato in A.9.1

```

35 % Calcolo profili PV in p.u.
36 for i = 1 : length(Tamb_min)
37     for k = 1 : numProfIrr
38         Tc(i,k) = Tamb_min(i,1) + (((NOCT - 20)/800) .*
            house_panel_irradiance_1min(i, k));
39         nt_h(i,k) = 1 - alpha_th .* (Tc(i,k) - 25);
40         P_PVpu(i,k) = n_DC_AC .* (house_panel_irradiance_1min(i, k)
            /1000) .* nt_h(i,k); %Profili PV in p.u.
41     end
42 end

```

Codice 5.2.1: Main_ProfilPV(calcolo profili in per unità)

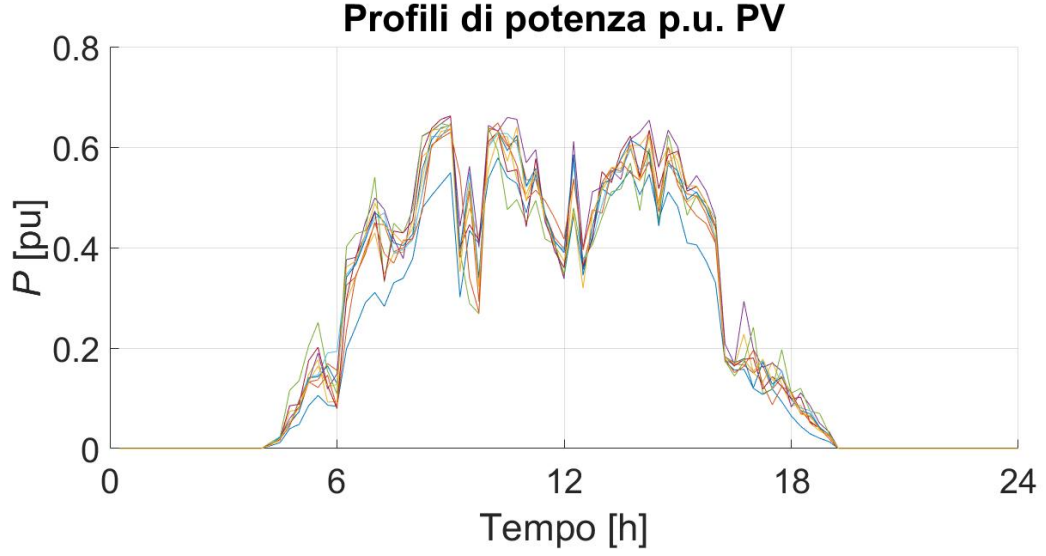


Figura 5.4: Profili di potenza in p.u.

Di questi profili ne è stato scelto uno solo, questo perchè la rete elettrica non presenta un'estensione spaziale tale da giustificare la scelta di profili diversi.

Nel Paragrafo 5.2.3 verrà illustrato il procedimento per la determinazione della potenza installata negli impianti fotovoltaici nella rete test.

5.2.3 Determinazione della potenza di generazione installata

Per l'inserimento del fotovoltaico all'interno della rete test viene utilizzato un procedimento iterativo. Tale processo sfrutta le perdite allocate come indice di sensitività dei nodi [23] per decidere in quale nodo andare ad iniettare della potenza attiva legata alla generazione fotovoltaica. La definizione di perdite allocate è data in eq. (5.5):

$$L_{k,t} = \text{Re} \left\{ \bar{I}_{k,t}^* \sum_{b \in \mathbb{B}_k} \left(R_b \bar{I}_{b,t} \right) \right\} \quad (5.5)$$

dove R_b è la resistenza del ramo b , $\bar{I}_{k,t}^*$ è il complesso coniugato della corrente assorbita dagli elementi connessi al nodo k nel tempo t e $\bar{I}_{b,t}$ è la corrente complessa che scorre nel ramo $b=1, \dots, B$.

Quando un aumento di carico in un nodo porta ad aumentare le perdite totali del sistema si riscontrano delle perdite allocate positive nel nodo. Al contrario, un valore di perdite allocate

negative nel nodo k corrisponde alla riduzione delle perdite totali del sistema se la potenza di carico netta in quel nodo aumenta.

Questo quindi, nella nostra applicazione porterà a distinguere due casi:

1. Perdite allocate $L_{k,t} < 0$
2. Perdite allocate $L_{k,t} > 0$

Nel primo caso verranno valutate le perdite medie ($\bar{L}_k$) per ogni nodo k della rete e il nodo nel quale verrà inserito il contributo in potenza attiva sarà il nodo k' tale che $\bar{L}_{k'} = \min(\bar{L}_k)$. Nel secondo caso invece verrà valutato il prodotto dato da: $\text{sgn}(\bar{P}_k) \cdot \text{sgn}(\bar{L}_k)$; in particolare, si cercheranno quei nodi in cui questo prodotto sarà < 0 e nel nodo k'' , che presenterà modulo maggiore, quindi $|\bar{L}_{k''}| = \max(|\bar{L}_k|)$, si inserirà il contributo in potenza attiva.

Per quanto riguarda i vincoli legati a tale metodo, si avranno tre vincoli: uno legato alla massima potenza installabile in ciascun nodo della rete, un vincolo legato al rapporto fra le energie dei profili di generazione fotovoltaica e quella dei carichi passivi, $\frac{E_{pv}}{E_{carico}}$ (inteso come un coefficiente di penetrazione di generazione fotovoltaica all'interno della rete) e infine, poichè il procedimento viene attuato mediante un codice *Matlab*® e in particolare usando dei cicli *while*, occorre inserire un vincolo legato al massimo numero di iterazioni, onde evitare che il ciclo non incontri mai il criterio di stop.

I valori scelti per i vincoli appena esposti sono riportati di seguito:

- Potenza massima installabile nel nodo: $P_{\text{inst,max}} = 1000 \text{ kW}$
- Rapporto massimo energie: $\xi_{E,\text{max}}^{\text{pv}} = \frac{E_{\text{tot,pv,max}}}{E_{\text{tot,carichi}}} = 0.8$
- Numero massimo di iterazioni: $\overset{\curvearrowright}{I}_{\text{max}} = \text{Iterazioni}_{\text{max}} = 10000$

Il procedimento svolto può essere visualizzato più chiaramente tramite il flow-chat riportato in Figura 5.5:

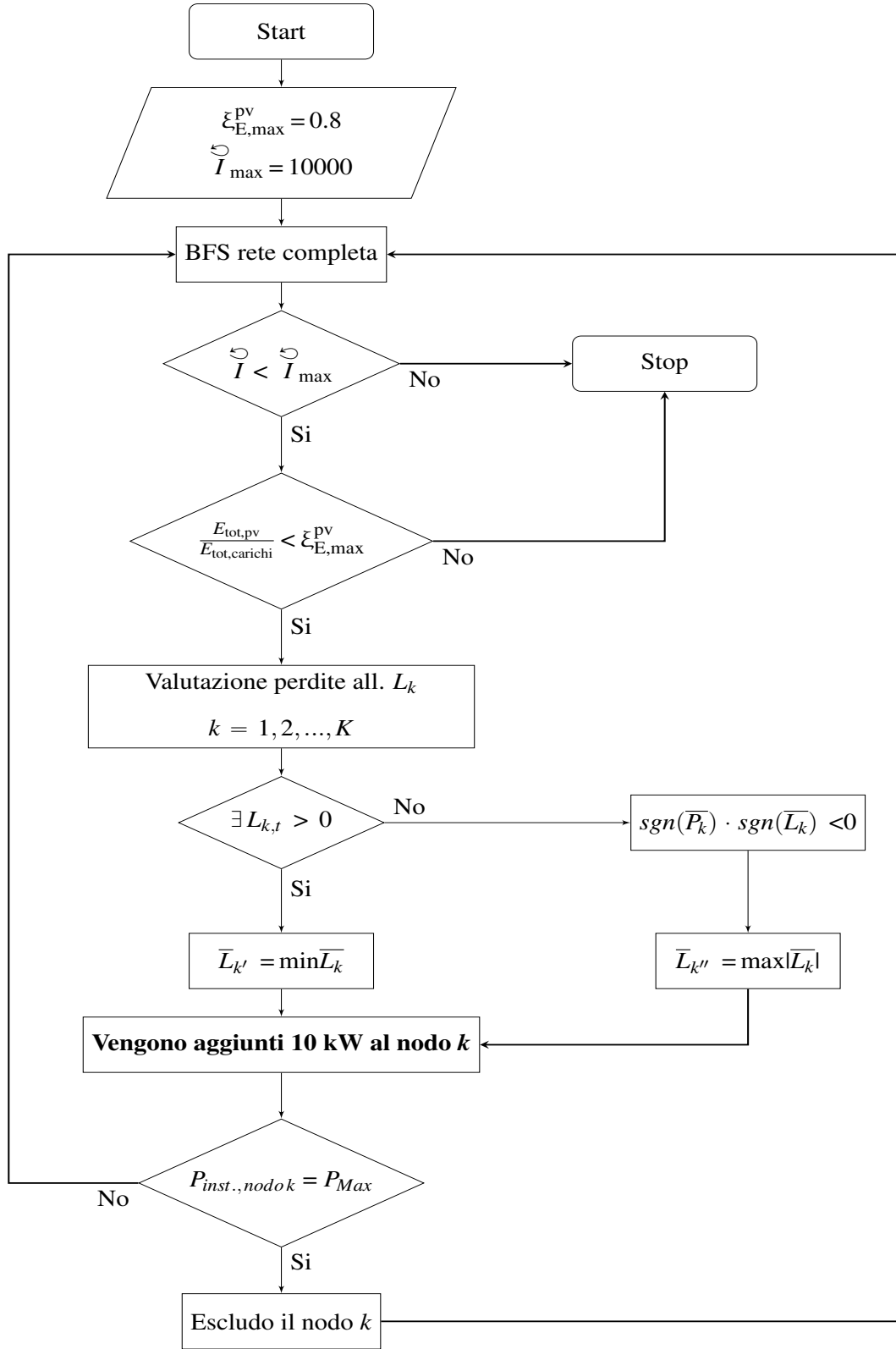


Figura 5.5: Flow-chart procedimento inserimento PV rete test

Vengono riportati nelle Figure 5.6 e 5.7 rispettivamente il livello di carico e l'andamento delle perdite allocate per il nodo 204874 della dorsale *Magra* delle rete riportata in Figura

4.1. È stato scelto questo nodo perché alla fine del processo ha una potenza installata pari alla massima possibile, quindi 1000 kW:

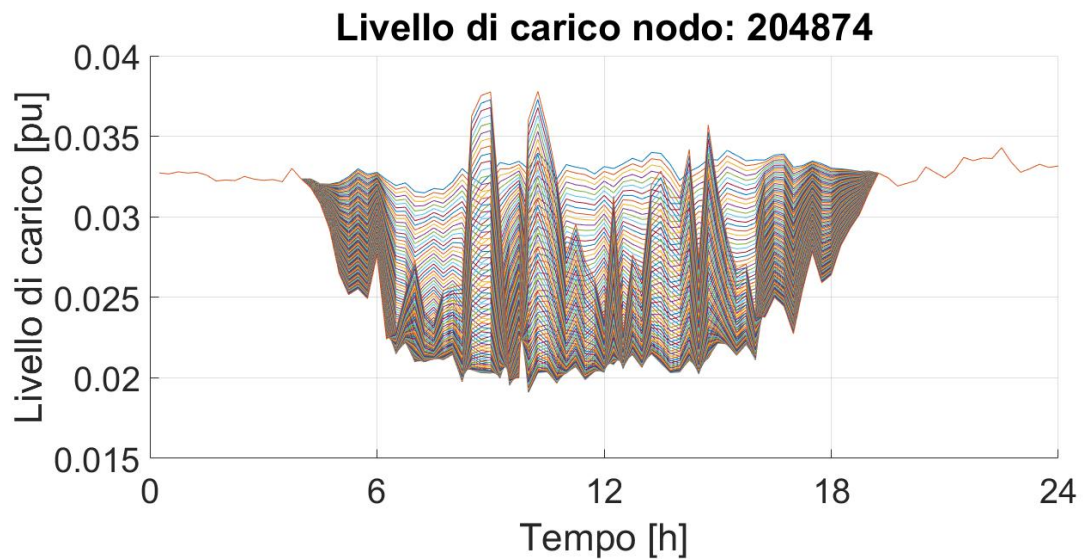


Figura 5.6: Livello di carico nodo 204874

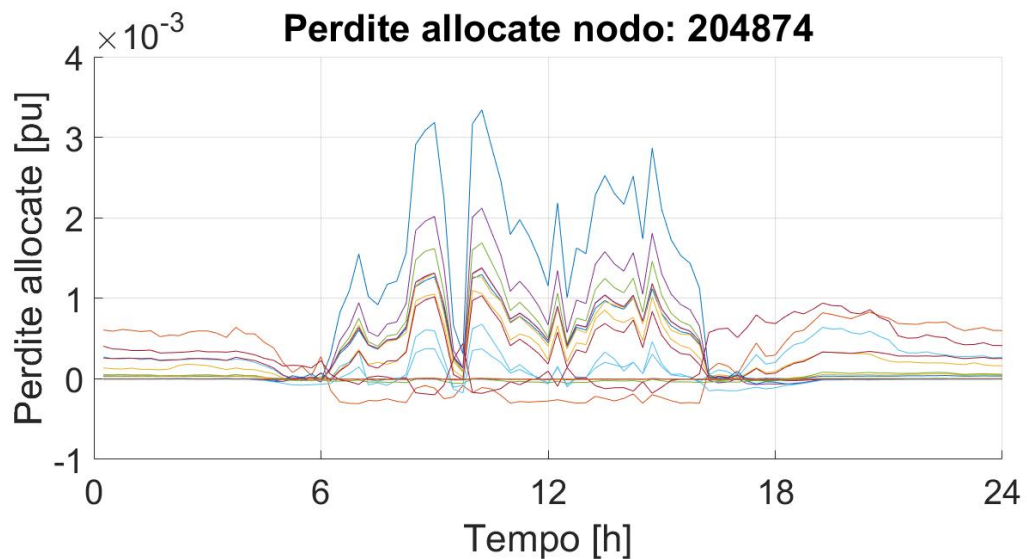


Figura 5.7: Perdite allocate nodo 204874

Come si può notare la variazione del livello di carico del nodo avviene fra le ore 6 e le 18, intervallo in cui gli impianti di generazione produrranno potenza attiva. Stessa cosa è visibile per le perdite allocate nello stesso nodo: la variazione maggiore di tali perdite sarà all'interno dello stesso intervallo di tempo.

Una volta ottenuta una distribuzione di generazione all'interno della rete test, sono stati

valutati gli indici relativi al funzionamento del sistema e sono stati confrontati con quelli relativi alla rete totalmente passiva di partenza. Di seguito vengono riportati brevemente i principali indici ed i valori ottenuti:

1. Energia persa percentuale (%)

$$E_{\text{persa},\%} = \frac{\mathbf{p}_{\text{loss},b}}{\sum (\text{Re}(S_{\text{carico},\text{pu}}))} \cdot 100 \quad (5.6)$$

dove al numeratore troviamo le perdite calcolate tramite BFS, cioè:

$$\mathbf{p}_{\text{loss}} = \sum_{b=1}^B \sum_{t=1}^T R_b \cdot I_{b,t}^2 \quad (5.7)$$

mentre al denominatore la somma delle potenze attive di tutti i carichi passivi presenti in rete.

2. Reverse power flow (MWh)

$$\mathbf{e}_{\text{AT/MT}} = \text{Re}(\tau \cdot \mathbf{e}_s * \mathbf{i}_s) = \{E_{\text{AT/MT},t}\}, t = 1, \dots, T \quad (5.8)$$

$$RPF = \sum_{t: \mathbf{e}_{\text{AT/MT}}(t) < 0} \{E_{\text{AT/MT},t}\} \quad (5.9)$$

dove τ è il numero di intervalli di tempo espresso in ore, $\mathbf{e}_s$ è la tensione associata allo slack nell'istante di tempo j , $\mathbf{i}_s$ è il vettore della somma delle correnti che scorrono nei nodi connessi allo slack, il simbolo $\cdot$ rappresenta il prodotto scalare, mentre $*$ il prodotto di Hadamard, cioè elemento per elemento.

L' RPF rappresenta l'iniezione di energia al sistema di trasmissione attraverso il trasformatore AT/MT.

3. Reverse power flow (min)

$$RPF_{\min} = \sum_t (\tau_j) \quad \forall t : \mathbf{e}_s(t) \cdot \mathbf{i}_s(t) < 0 \quad (5.10)$$

Quindi il valore sarà dato dalla somma degli intervalli τ_j in cui è verificata la condizione $\mathbf{e}_s(t) * \mathbf{i}_s(t) < 0$.

4. Indice auto-sufficienza della rete (Grid self-sufficiency) (%)

$$S_s = \frac{E_{\text{tot,FER}} - RPF}{E_{\text{carichi}}} \cdot 100 \quad (5.11)$$

dove FER sta per Fonti Energia Rinnovabile, questo indice rappresenta la percentuale di produzione legata a fonti di energia rinnovabile presenti nella rete.

5. Indice auto-consumo della rete (Grid self-consumption) (%)

$$S_c = \frac{E_{\text{tot,FER}} - RPF}{E_{\text{tot,FER}}} \cdot 100 \quad (5.12)$$

Questo indice rappresenta il rapporto tra la produzione da fonti di energia rinnovabile usate per alimentare i carichi e la produzione totale delle stesse. Per quanto riguarda la nostra rete test possiamo distinguere gli indici prima e dopo l'inserimento della produzione fotovoltaica. Non essendo prima presenti fonti rinnovabili interne alla rete, $E_{\text{tot,FER}}$ è nulla all'inizio. I risultati ottenuti prima e dopo l'inserimento dei profili di fotovoltaico sono riportati in Tabella 5.2:

Tabella 5.2: Indici di rete prima e dopo l'inserimento dei profili di fotovoltaico

Indici di rete	Rete senza PV	Rete con PV
$P_{\text{PV,inst}} (kW)$	0	30320
$E_{\text{persa, \%}} (\%)$	0.33	1.3
$RPF (MWh)$	0	60.72
$RPF (min)$	0	1380
$S_s (\%)$	0	50.02
$S_c (\%)$	Ind.	62.51

Come già detto, la rete si presentava come una rete totalmente passiva prima dell'inserimento dei profili di fotovoltaico. Questo spiega perché alcuni degli indici siano nulli, o nel caso di Sc indeterminati. È possibile notare che l'energia persa dopo l'inserimento del PV aumenta, questo a causa dell'eccesso di generazione presente. Gli indici di auto-sufficienza e di auto-consumo aumentano dopo l'inserimento della generazione locale, essendo adesso le grandezze presenti al numeratore e, per quanto riguarda Sc , anche al denominatore, diverse da zero.

Nel Capitolo 6 si andrà ad inserire all'interno della rete l'impianto *Power-to-Gas*, modellizzato sempre tramite *Matlab*[®], questo porterà ad un miglioramento degli indici di rete che verrà successivamente discusso.

Capitolo 6

Inserimento del Power-to-Gas nella rete

In questo capitolo verrà discusso il sistema P2G che è stato inserito all'interno della rete test e saranno valutati gli effetti su tutto il sistema.

Sarà prima esposta la modellizzazione del sistema e successivamente verranno valutati gli indici di rete e gli andamenti delle grandezze più importanti.

6.1 Modellizzazione del P2G tramite codice Matlab®

In questo paragrafo verrà brevemente discussa la funzione utilizzata per modellizzare il sistema P2G. Il sistema considerato è riportato in Figura 6.1:

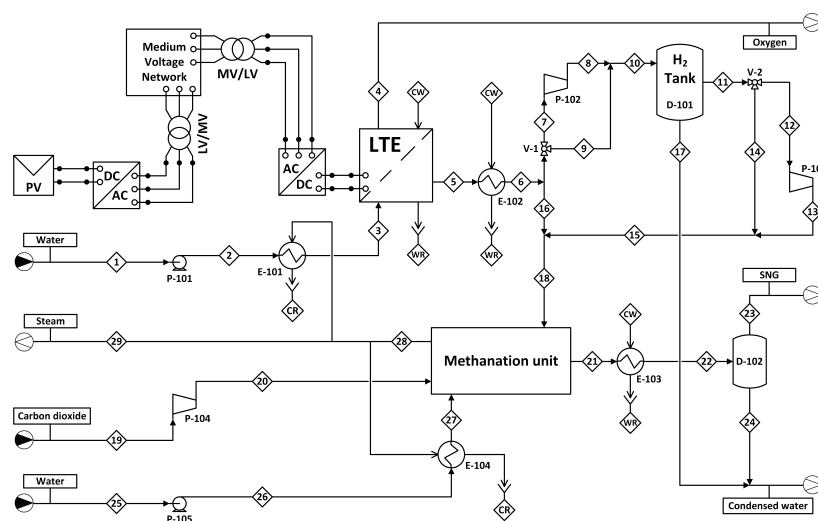


Figura 6.1: Schema impianto Power-to-Gas [10]

La taglia del sistema è stata impostata considerando un intervallo compreso tra 1 MW ÷ 5 MW.

Nel caso in esame la taglia è stata scelta pari a $P_{\text{nom}} = 3 \text{ MW}$. All'interno della funzione che descrive il sistema P2G, denominata "*funzione_ptgnode_new*" vengono eseguiti diversi passi di calcolo. Innanzitutto dai valori di RPF viene ricavato l'andamento del profilo del sistema P2G nel nodo di installazione.

Il processo utilizzato è di tipo iterativo, è infatti presente un ciclo *while* che racchiude la condizione sulla tolleranza, definita da "*Ptg.sim.TOLL*" all'interno del codice e quella sul numero di iterazioni, necessaria per evitare di rimanere incastrati all'interno del ciclo, quest'ultima definita da "*Ptg.sim.ITER_MAX*", i valori di queste due grandezze sono i seguenti:

- *Toll* (Ptg.sim.TOLL): $1 \cdot 10^{-6}$;
- I_{max} (Ptg.sim.ITER_MAX) = 50.

La riga ha la forma riportata in 6.1.1:

```
14 while err_rel > Ptg.sim.TOLL && iter <= Ptg.sim.ITER_MAX
```

Codice 6.1.1: function_P2G(Vincoli tolleranza e numero iterazioni)

Il numero di iterazione è un semplice indice che viene incrementato ad ogni ciclo mentre l'errore relativo viene calcolato come nell'equazione (6.1):

$$e_{\text{rel},t} = \frac{P_{\text{aux},t} - P_{\text{aux},t-1}}{P_{\text{aux},t}} \quad (6.1)$$

dove $e_{\text{rel},t}$ rappresenta l'errore relativo nell'istante di tempo t , con $P_{\text{aux},t}$ si indica la potenza degli ausiliari del sistema, mentre $P_{\text{aux},t-1}$ rappresenta l'errore nell'istante $t - 1$.

Il termine $P_{\text{aux},t}$ sarà costituito quindi da vari contributi, il compressore relativo alla CO₂, P-104 nella Figura 6.1. Un quantitativo di potenza sarà legato alle pompe dell'acqua, P-101 e P-105, gli scaldatori, sempre relativi all'acqua, E-101 ed E-104 e infine i compressori collegati all'ingresso e all'uscita del serbatoio d'idrogeno, cioè P-102 e P-103. Tutti gli elementi sono visualizzabili sempre nella Figura 6.1. A questi si aggiungeranno i contributi

di potenza dissipata nell'unità di metanazione, composta da tre metanatori e infine un ultimo termine legato all'elettrolizzatore alcalino, quindi:

$$P_{aux,t} = P_{CO_2} + P_{H_2O} + P_{H_2O,risc.} + P_{H_2,compr.1} + P_{H_2,compr.2} + |P_{Met,t}| + |P_{Aec,t}| \quad (6.2)$$

La dinamica dell'elettrolizzatore viene definita dall'equazione differenziale (6.3):

$$\frac{dy}{dt} = \frac{(x(t - \delta) - p_{Aec}(t))}{T_{pol}} \quad (6.3)$$

Dove, δ rappresenta il ritardo in secondi della dinamica del primo ordine, pari a 25 s, p_{Aec} la pressione dell'elettrolizzatore alcalino, questa pari ad 1 MPa ed infine T_{pol} il tempo in secondi della dinamica del primo ordine, pari a 10 s.

La parte preponderante della funzione *function_ptgnode_new* analizza l'andamento del serbatoio di idrogeno, vengono calcolate diverse grandezze per ogni istante:

- n_{H_2} : Portata [kmol/s] di H_2 prodotta dall'elettrolizzatore alcalino;
- $H_{2,tank}$: Quantità [kmol] di H_2 contenuta nel serbatoio.
- $n_{H_{2,tank}}$: Portata [kmol/s] di H_2 , che viene inviata al serbatoio tramite il compressore che è posizionato a monte.
- $n_{H_{2,tank,bp}}$: Portata [kmol/s] di H_2 , immessa nel serbatoio senza l'uso di un compressore, questo si verifica quando $p_{serb.} < p_{AEC}$;
- $n_{H_{2,met}}$: Portata [kmol/s] di H_2 che viene successivamente miscelata con la CO_2 e quindi utilizzata per produrre metano.

Un esempio dell'andamento che si ottiene utilizzando questa funzione viene riportato nella Figura 6.2, è stato considerato un intervallo di tempo di due settimane:

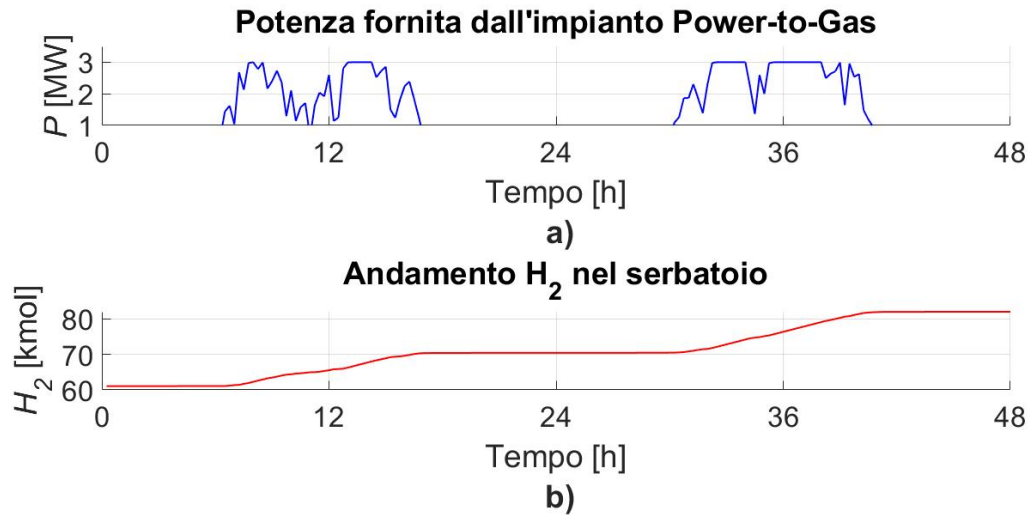


Figura 6.2: Potenza attiva a) e kmol di H_2 b)

Nel Paragrafo 6.2 è stato valutato l'effetto del P2G sulla rete elettrica, confrontando il valore degli indici di rete ottenuti dopo il suo inserimento con quelli ottenuti in 5.2.3 riportati nella tabella 5.2.

6.2 Inserimento del P2G nella rete Test

6.2.1 Presenza di un singolo impianto P2G

Il codice Matlab[®] utilizzato è stato chiamato "*Main_inserimento_Ptg*", all'interno di questo codice vengono richiamate diverse funzioni, cioè:

- *function_parametri_Ptg*: questa funzione carica tutti i parametri dei vari componenti del sistema Power-to-Gas;
- *function_backward_forward_sweep*: funzione che calcola il BFS della rete nel modo già discusso precedentemente in 4.2.2;
- *ptgnode_v2_new*: Questa funzione modella il sistema P2G analizzandolo ad ogni passo di calcolo e fornisce in uscita l'andamento del livello del serbatoio di idrogeno e il valore di potenza attiva del sistema;

- *function_backward_forward_sweep_NO_SCAR*: Questa funzione è uguale a quella per il calcolo del BFS, con l'unica differenza che la matrice dei carichi, cioè " S_{car} " non viene calcolata all'interno della funzione ma viene fornita dall'esterno dopo l'aggiunta del contributo di potenza del sistema P2G.

Per quanto riguarda il profilo del sistema P2G si è partiti dall'RPF in ogni nodo, calcolato come:

$$RPF(t) = \text{Re}(v_k(t) \cdot i_b(t)) \quad (6.4)$$

Dove, t rappresenta l'istante di tempo considerato, v_n la tensione nel nodo n nell'istante di tempo t , ed infine i_b la corrente che scorre nel ramo b , cioè il ramo che collega il nodo k della rete radiale, nell'istante t .

Si otterrà quindi una matrice contenente istante per istante in ogni nodo il valore del RPF per ogni nodo. A questo punto, essendo interessati ai valori di potenza che dai nodi fluiscono verso lo slack, è stato sufficiente annullare quelli con il segno opposto. Tuttavia il BFS inverte i segni delle correnti, per esempio, considerando una piccola porzione della rete test originaria di Figura 4.1:

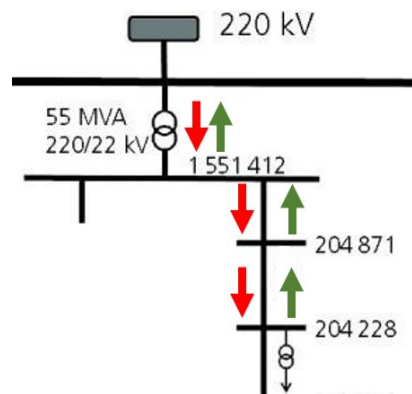


Figura 6.3: Versi reali corrente (in rosso) e versi ottenuti dal calcolo (in verde)

I versi delle frecce in rosso rappresentano i versi reali delle correnti, mentre in verde sono stati rappresentati i versi ottenuti con il calcolo. Questi ultimi verranno posti a zero per ottenere il profilo voluto.

Per quanto riguarda il numero di impianti, inizialmente si è deciso di inserire un solo impianto P2G, ponendolo nel nodo che presenta il maggior valore di flusso di potenza inverso,

in modo da massimizzare l'effetto benefico che l'inserzione del P2G può avere sulla rete. Il nodo scelto è il numero 204946 della rete test di Figura 4.1.

A questo punto è stata eseguita una simulazione su una giornata per verificare i valori degli indici di rete, già discussi nel Paragrafo 5.2.3. I risultati sono riportati nella Tabella 6.1:

Tabella 6.1: Indici di rete (valutazione su singola giornata)

Indici di rete	Rete senza P2G	Rete con P2G
(valori ottenuti in 5.2)		
$E_{\text{persa},\%} (\%)$	1.3	0.8
$RPF (MWh)$	60.72	39.87
$RPF (min)$	1380	1305
$S_s (\%)$	50.02	65.02
$S_c (\%)$	62.51	75.16

Possiamo notare che l'inserimento di un solo impianto P2G aiuta a migliorare gli indici di rete. In particolare, le perdite percentuali risultano diminuite del 38% e gli indici di Self-sufficiency (5.11) e di Self-consumption (5.12) aumentano entrambi, andando quindi ad attestare un miglioramento nell'autosufficienza della rete e nell'auto consumo dell'energia rinnovabile prodotta nella rete.

Si può notare, infine, che è ancora presente un discreto quantitativo di RPF (pari circa a 39.87 MWh): questo, come già detto rappresenta il flusso di potenza che scorre verso il nodo di slack. Tale valore risulta comunque ridotto del 34% rispetto alla condizione senza P2G.

Nelle Figure 6.4 e 6.5 vengono mostrati gli andamenti delle tensioni nodali e delle correnti nei rami dopo l'inserimento dell'impianto P2G.

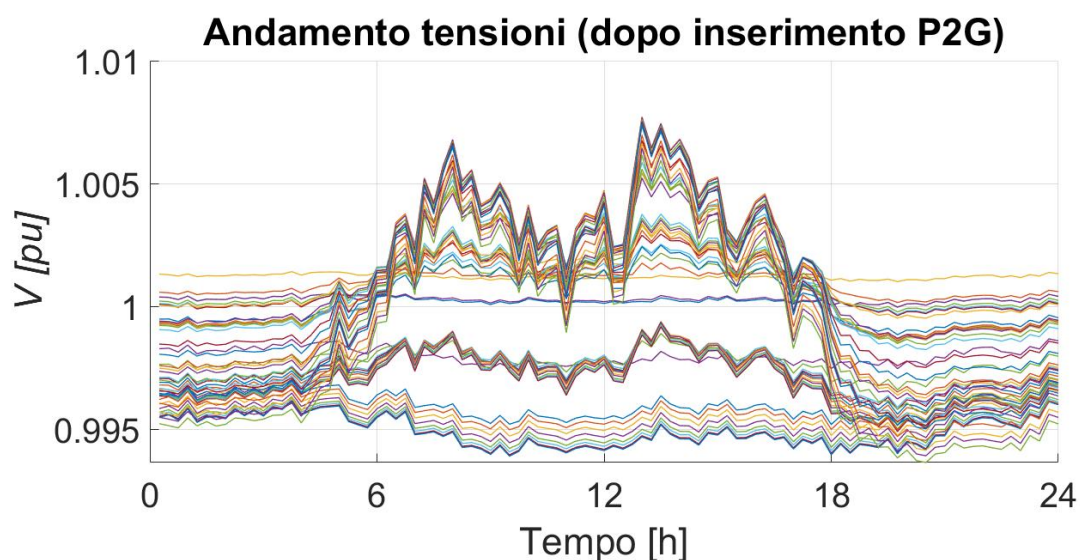


Figura 6.4: Tensioni nei nodi della rete test

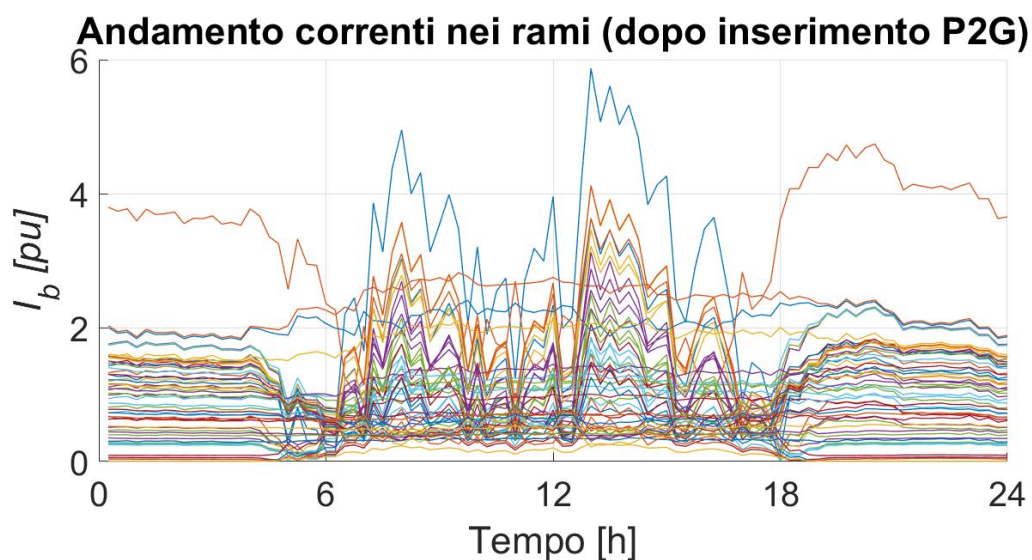


Figura 6.5: Correnti nei rami della rete test

Rivestono particolare importanza le tensioni, in quanto si vedrà che un andamento più "appiattito" delle stesse può essere ottenuto considerando la presenza di più impianti P2G all'interno della rete Test.

Si riportano per completezza anche gli andamenti delle perdite e del livello di carico dei vari nodi. Il nodo che presenta un'escursione maggiore delle perdite e del livello di carico è il nodo 203548, come era plausibile, questo per via della presenza del nuovo carico legato all'impianto P2G. Gli andamenti risultano comunque accettabili e sono mostrati nelle Figure [6.6](#) e [6.7](#):

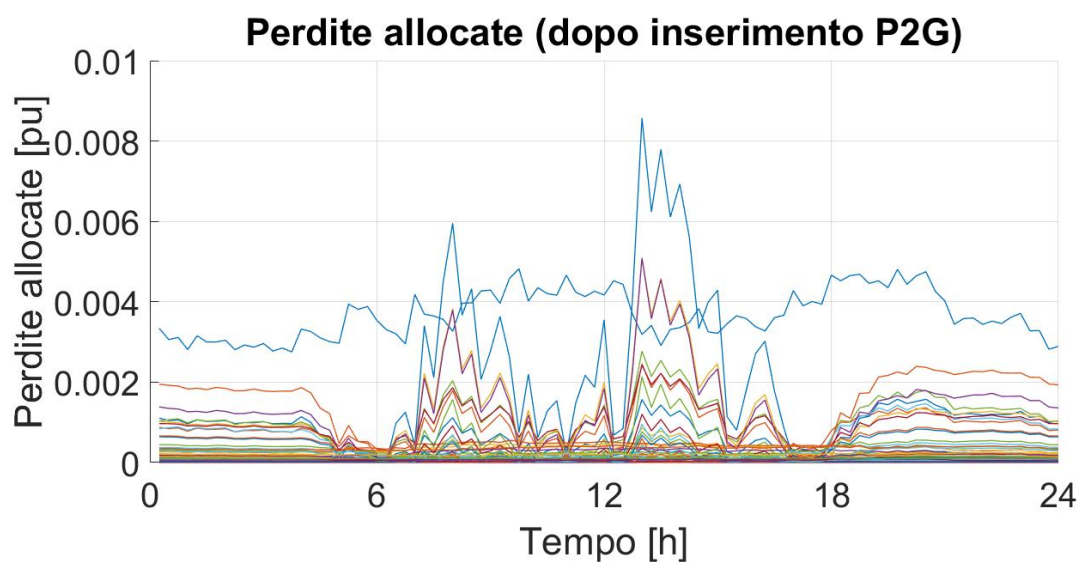


Figura 6.6: Perdite allocate nei nodi della rete Test

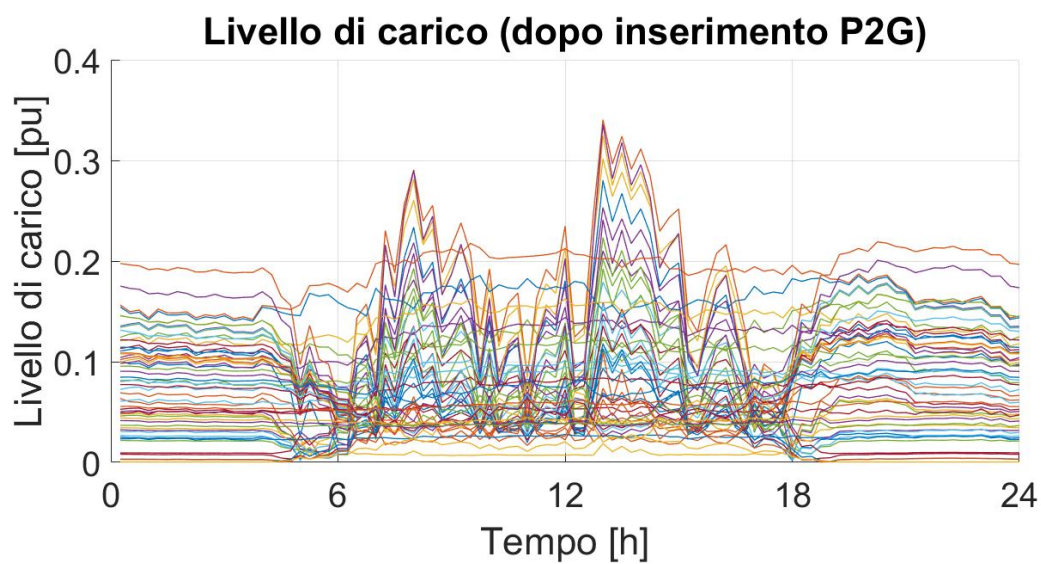


Figura 6.7: Livello di carico rete Test

La simulazione completa, fatta su un arco di tempo annuale fornisce i valori degli indici riportati nella Tabella [6.2](#).

Tabella 6.2: Indici di rete (valutazione su un anno)

Indici di rete	Rete senza P2G	Rete con P2G
$E_{\text{persa}, \%}$ (%)	1.7	0.8
RPF (MWh)	21414.28	14704
RPF (min)	495360	383550
S_s (%)	44.2	57
S_c (%)	52.5	69

Come si può notare, su base annuale il miglioramento degli indici di rete segue la stessa linea di quello visto confrontando i valori della Tabella 6.1. Riveste particolare importanza l'RPF che viene ridotto del 31% in seguito all'inserimento di un impianto P2G nella rete elettrica. Infine è stato valutato il Capacity Factor (CF) percentuale, che rappresenta il rapporto tra l'energia assorbita dall'impianto E_{pr} in un arco di tempo annuale e l'energia assorbita se l'impianto avesse lavorato alla potenza nominale in modo continuativo durante lo stesso lasso di tempo E_{nom} :

$$CF_{\%} = \frac{E_{\text{pr}}}{E_{\text{nom}}} \cdot 100 = 64\% \quad (6.5)$$

Nel Paragrafo 6.2.2 si è quindi deciso di provare un approccio multi-impianto al fine di valutare il comportamento degli indici di rete e l'andamento delle varie grandezze.

6.2.2 Presenza di più impianti P2G

L'inserimento di un impianto P2G ha migliorato gli indici di rete, come visto nella Tabella 6.1. Si è deciso a questo punto di valutare il comportamento della rete considerando un approccio multi impianto.

Sono stati considerati quattro impianti P2G con le stesse caratteristiche. Essi avranno quindi uguale P_{nom} , pari a 3 MW e stesse caratteristiche dell'impianto riportato in Figura 6.1. Gli impianti sono stati inseriti in diversi nodi della rete test di Figura 4.1. I nodi in cui sono stati inseriti sono il numero 204874 della dorsale *Magra*, 203294 della dorsale *Brenta*, 2052802

della dorsale *Grosso* e 204946 della dorsale *Chieri*. I sistemi P2G sono stati inseriti in modo progressivo, valutando di volta in volta gli indici di rete. I risultati sono riportati nella Tabella 6.3, in cui sono stati riportati anche quelli relativi al singolo impianto visti nel Paragrafo 6.2.

Tabella 6.3: Indici di rete, presenza di più impianti P2G (singola giornata)

Indici di rete	N° Impianti P2G			
	1	2	3	4
(da 6.1)				
$E_{\text{persa},\%} (\%)$	0.8	0.4	0.3	0.29
$RPF (MWh)$	39.87	15.21	1.44	0
$RPF (min)$	1305	840	150	0
$S_s (\%)$	65.02	78.31	85.73	86.51
$S_c (\%)$	75.16	90.53	99.1	100

Come si può vedere dai valori riportati nella Tabella 6.3, l'energia persa percentuale diminuisce e risulta dimezzata nel passaggio da uno a due impianti e poi si mantiene pressoché costante. Il flusso inverso di potenza si annulla in presenza di 4 impianti P2G e gli indici di auto-consumo (S_c) e auto-sufficienza (S_s) della rete aumentano.

Nelle Figure 6.8 e 6.9 sono riportate rispettivamente le tensioni e le correnti di linea della rete test in presenza di quattro impianti.

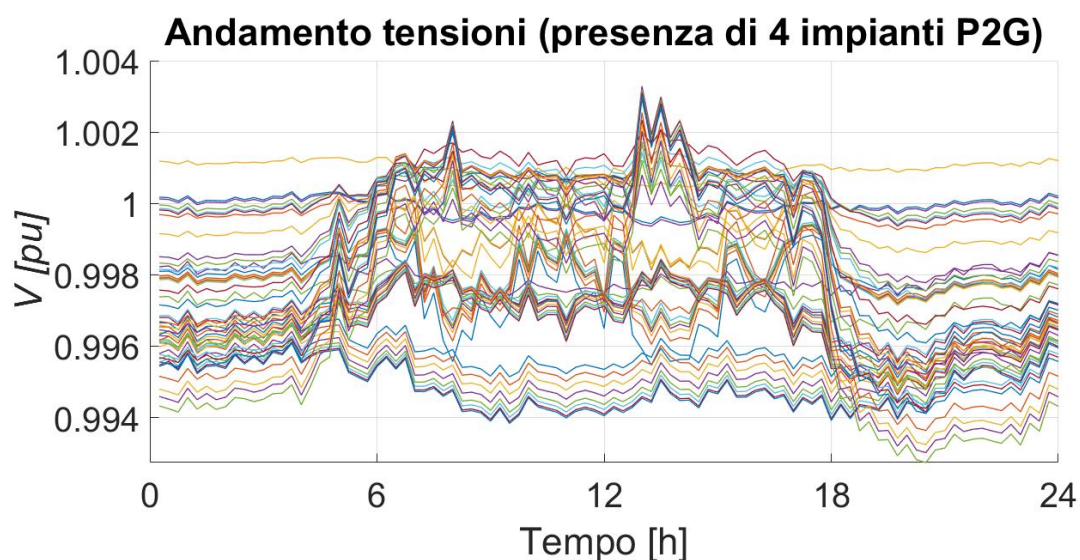


Figura 6.8: Tensioni della rete test (presenza di quattro impianti P2G)

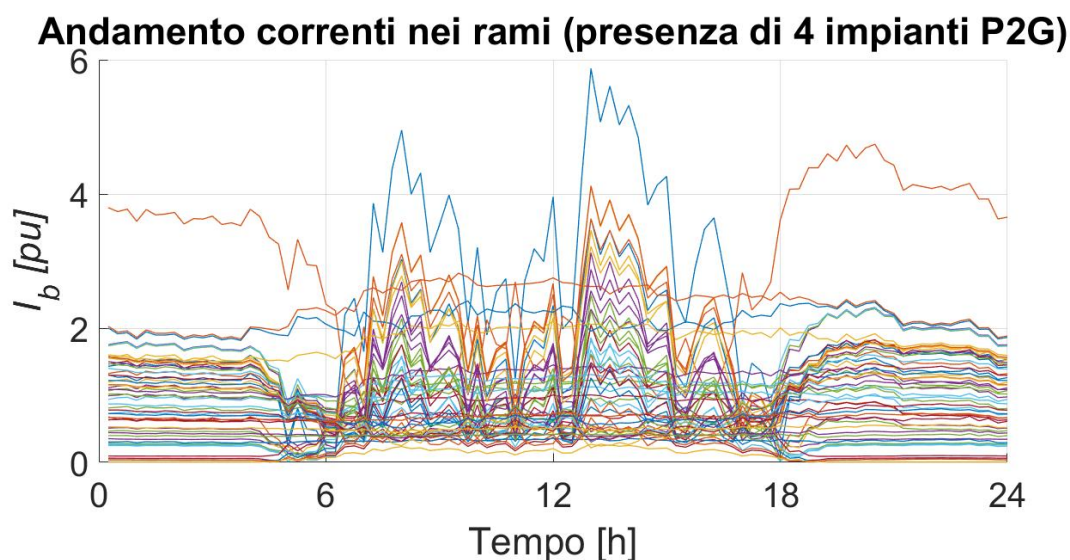


Figura 6.9: Correnti nei rami della rete test (presenza di quattro impianti P2G)

Come si può notare le tensioni risultano avere picchi più bassi rispetto a quelle di [Figura 6.4](#).

Le Figure [6.10](#) e [6.11](#) mostrano invece l'andamento delle perdite allocate e del livello di carico. Si può notare che le perdite risultano ridotte rispetto a quelle riportate in [Figura 6.6](#), mentre il livello di carico rientra nei limiti accettabili.

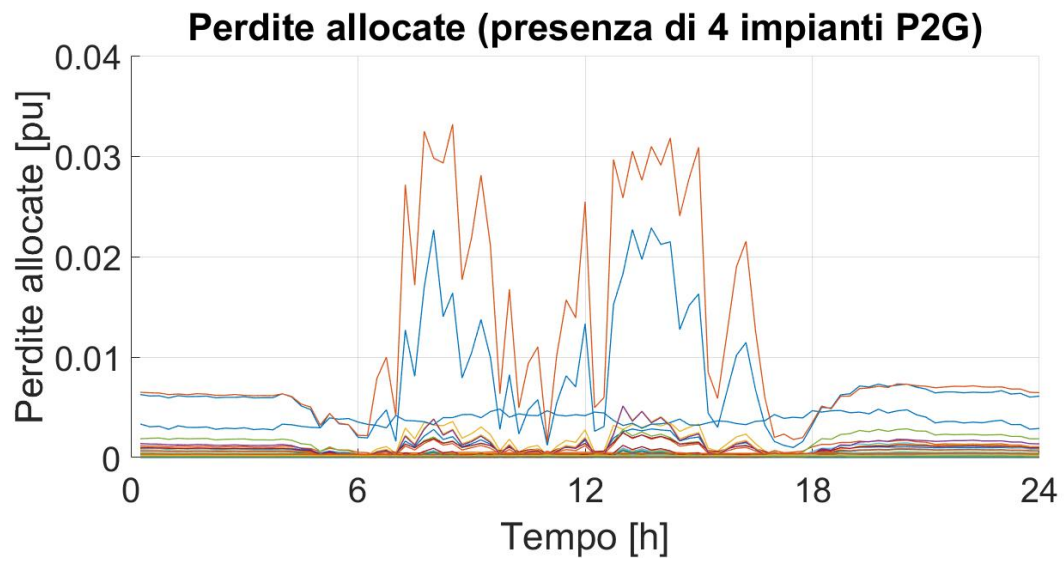


Figura 6.10: Perdite allocate(presenza di quattro impianti P2G)

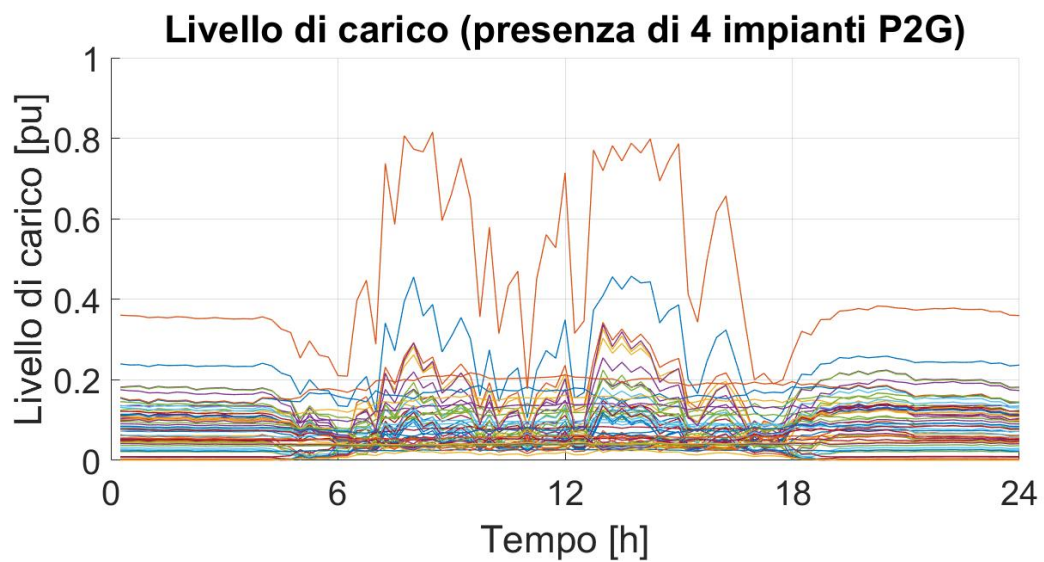


Figura 6.11: livello di carico (presenza di quattro impianti P2G)

Anche in questo caso sono stati valutati gli indici di rete su base annuale, i risultati sono riportati nella Tabelle [6.4](#).

Tabella 6.4: Indici di rete, presenza di più impianti P2G (valutazione su un anno)

Indici di rete	N° Impianti P2G			
	1	2	3	4
$E_{\text{persa},\%} (\%)$	0.8	0.38	0.28	0.27
$RPF (MWh)$	14704	6742	1683.5	21.92
$RPF (min)$	383550	276945	83820	3420
$S_s (\%)$	57	71	80.4	83.45
$S_c (\%)$	69	85	96.3	99.95

Si può notare come in questo caso sia presente ancora un flusso di RPF; tuttavia i valori ottenuti risultano soddisfacenti. La riduzione dell'RPF con quattro impianti è del 99%. Infine, anche in questo caso, è stato calcolato il Capacity Factor, in questo caso definito come $CF_{\text{numnodo},\%}$, dove con "numnodo" si intende il nodo in cui sono stati inseriti gli impianti P2G (equazioni (6.6) ÷ (6.9)).

$$CF_{204874,\%} = \frac{E_{\text{pr}}}{E_{\text{nom}}} \cdot 100 = 57\% \quad (6.6)$$

$$CF_{203294,\%} = \frac{E_{\text{pr}}}{E_{\text{nom}}} \cdot 100 = 56\% \quad (6.7)$$

$$CF_{2052802,\%} = \frac{E_{\text{pr}}}{E_{\text{nom}}} \cdot 100 = 50\% \quad (6.8)$$

$$CF_{204946,\%} = \frac{E_{\text{pr}}}{E_{\text{nom}}} \cdot 100 = 42\% \quad (6.9)$$

Come si può notare dall'equazione (6.9) il CF dell'impianto inserito nel nodo 204946 è inferiore al 50%, questo significa che la produzione di energia rispetto all'energia nominale non è soddisfacente. Questo impianto è stato rimosso per valutare il comportamento dei fattori di capacità degli altri impianti presenti in rete in assenza di esso (equazioni (6.10) ÷ (6.12)).

$$CF_{204874,\%} = \frac{E_{pr}}{E_{nom}} \cdot 100 = 58\% \quad (6.10)$$

$$CF_{203294,\%} = \frac{E_{pr}}{E_{nom}} \cdot 100 = 57.5\% \quad (6.11)$$

$$CF_{2052802,\%} = \frac{E_{pr}}{E_{nom}} \cdot 100 = 55\% \quad (6.12)$$

In quest'ultimo caso, considerando tre impianti P2G nella rete, i valori dei CF si mantengono in linea con quelli riportati nelle equazioni (6.6), (6.7) e (6.8), aumentando leggermente. Le tensioni risultano meno piatte rispetto alla Figura 6.8 ma rientrano comunque nell'intervallo ammissibile. L'andamento delle tensioni è riportato in Figura 6.12.

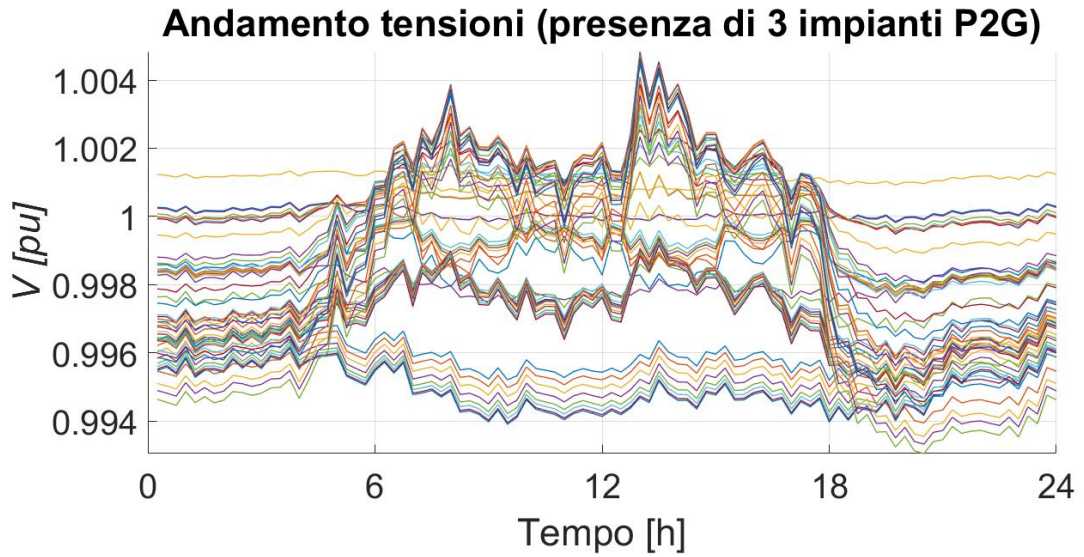


Figura 6.12: Andamento tensioni (presenza di tre impianti P2G)

Nella Tabella 6.5 vengono riportati i massimi valori di tensione per la giornata considerata, si può notare come il valore decresca all'aumentare del numero di impianti.

Tabella 6.5: Valore massimo di tensione al variare del numero di impianti P2G

	N° Impianti P2G			
	1	2	3	4
$V_{\max}$	1.0077	1.0063	1.0048	1.0033

Capitolo 7

Conclusioni e applicazioni future

In questo capitolo vengono commentati i risultati ottenuti dalle varie simulazioni ed esposti i possibili sviluppi futuri del lavoro

7.1 Risultati ottenuti

In passato i sistemi di distribuzione erano di tipo passivo, ma la presenza sempre più marcata delle FER connesse alla rete di distribuzione ne ha cambiato la natura, poichè essa presenta dei flussi bidirezionali che possono interferire col suo corretto funzionamento.

La presenza massiccia di FER all'interno di una rete può essere causa di congestioni. Regioni che presentano un'elevata penetrazione di fotovoltaico all'interno del sistema elettrico, per esempio la Germania [7], hanno evidenziato che l'inserimento di impianti P2G sparsi sul territorio e connessi alla rete elettrica consentirebbe di assorbire circa il 20% dell'energia solare in eccesso. Il sistema P2G rappresenterà sicuramente un importante tassello nel futuro dei sistemi elettrici di potenza.

Nel lavoro di tesi presentato, il sistema P2G ha permesso di migliorare gli indici di rete e gli andamenti di tensione e perdite allocate. La rete in cui è stato inserito è stata totalmente simulata partendo dai profili di carico di tipo residenziale e non e questo ha permesso di ottenere degli aggregati nodali il più possibile realistici. In seguito, all'interno della rete sono stati inseriti dei profili di generazione fotovoltaica, considerando un coefficiente di penetrazione dell'80%.

Il successivo inserimento di un impianto P2G all'interno della rete ha permesso di ridurre l'energia persa percentuale dall'1.7% allo 0.8%, riducendo l'RPF del 31% con un solo impianto e del 92% con tre impianti sparsi all'interno della rete test.

La presenza di più impianti tende quindi a migliorare gli indici di rete fornendo una migliore gestione dei vincoli.

7.2 Sviluppi futuri

Questo lavoro ha mostrato come il sistema P2G possa dare un valido contributo nei sistemi di distribuzione elettrica. Le applicazioni future possono riguardare vari aspetti:

- Creazione e gestione di un mercato di vendita riguardante il GNS: la diffusione di impianti P2G consente di trasformare il surplus di generazione in GNS, il quale può essere commercializzato in un apposito mercato, al momento non ancora costituito.
- Ottimizzazione di sistemi energetici multi vettore: utilizzo del GNS per alimentazione veicoli: il ruolo del GNS nella mobilità è ancora in fase di sperimentazione, ma i veicoli alimentati con una miscela di gas naturale e idrogeno rappresenteranno un modo per abbattere l'inquinamento legato ai combustibili fossili. Uno degli sviluppi futuri può appunto essere lo studio e l'ottimizzazione di questi sistemi multi vettore.
- Integrazione P2G in una rete elettrica situata in una zona altamente industriale e gestione delle emissioni di CO₂: uno dei pregi fondamentali della tecnologia P2G è la possibilità di utilizzare la CO₂ catturata da appositi sistemi, abbattendone il quantitativo presente in aria. Questa tecnologia può essere sfruttata in una rete elettrica posta in una zona altamente industriale, riducendo così la quota delle emissioni di anidride carbonica e gestendone il livello in un'ottica di decarbonizzazione.

Appendice A

Codici Matlab[®]

In questa appendice sono stati inseriti tutti i codici Matlab[®] utilizzati durante questo lavoro di tesi, suddivisi per funzioni e codici principali.

A.1 Codice: Main_caricamento_profili

Questo codice permette di caricare i profili di carico all'interno del *workspace* e di cercare i punti non validi all'interno dei profili.

```
1 % For PLANET
2 % Dati: Profili di carico PrDataTot
3 % PrDataTot: [35040 x 150] 150 profili totali x 35040 Quarti d'
   ora annuali
4 % Caricamento lista profili
5 FileList = {'Profili_1_50', 'Profili_51_100', 'Profili_101_125',
   , 'Profili_126_150'};
6
7 cont = 1;
8 Table = [];
9 C_Table = [];
10 PrDataTot = []; % Matrice dati numerica
11 C_PrDataTot = []; % Matrice dati letterale
12
```

```

13 for iFile = 1:numel(FileList) % Carico i profili e genero le
    matrici dati di tutti i profili in successione
14     display(FileList(iFile));
15     profile = load(string(FileList(iFile)));
16     Table = profile(1,1).Untitled;
17     C_Table = profile(1,1).Untitled;
18     V = Table.Properties.VariableNames;
19
20 for i=[1:width(Table)]
21     v_is_cell(i)=~ isnumeric(Table.(V{i}));
22     g_is_cell(i)=~ iscategorical(C_Table.(V{i}));
23 end
24 Table(:,v_is_cell) =[];
25 Table;
26 C_Table(:,g_is_cell) =[];
27 C_Table;
28 v_is_cell = logical([]);
29 g_is_cell = logical([]);
30 x=string([]);
31
32 for j=1:width(Table)
33     x(j) = "res_num" + num2str(cont);
34     cont = cont+1;
35 end
36 Table.Properties.VariableNames = x;
37 C_Table.Properties.VariableNames = x;
38 C_PrData = table2array(C_Table);
39 PrData = table2array(Table);
40 PrDataTot = [PrDataTot, PrData];
41 C_PrDataTot = [C_PrDataTot, C_PrData];
42 end
43 %%
44 % Trovo valori <undefined> nella matrice letterale C_PrDataTot
45 Undefined= zeros(size(C_PrDataTot,2));
46

```

```

47 for jj=1:size(C_PrDataTot,2)
48     temp = find(isundefined(C_PrDataTot(:,jj)));
49     Undefined(1:length(temp),jj) = temp;
50     clear temp
51 end
52 % Verifico che ai valori <undefined> corrispondano valori Nan
    nella PrDataTot
53
54 for jj = 1:size(C_PrDataTot,2)
55     for kk = 1:size(C_PrDataTot,1)
56         if isnan(PrDataTot(kk,jj)) && isundefined(C_PrDataTot(kk,jj)
57             )
58             C_PrDataTot(kk,jj)= 'NA'; % Sovrascrivo ai valori <
                undefined> la sigla NA
59         end
60     end
61 %%
62 % NA: Matrice posizioni punti Na in C_PrDataTot (1)
63 % Cerco punti non disponibili (NA) in C_PrDataTot (1)
64 not_available = zeros(1,size(C_PrDataTot,2));
65
66 for jj=1:size(C_PrDataTot,2)
67     temp = find(C_PrDataTot(:,jj) == 'NA');
68     not_available(1:length(temp),jj) = temp;
69     clear temp
70 end
71 % R: Matrice posizioni Replaced C_PrDataTot (2)
72 % Cerco punti rimpiazzati (R) in C_PrDataTot (2)
73 R = zeros(1,size(C_PrDataTot,2));
74
75 for jj=1:size(C_PrDataTot,2)
76     temp = find(C_PrDataTot(:,jj) == 'R');
77     R(1:length(temp),jj) = temp;
78     clear temp

```

```

79 end
80 % Nan: NA: Matrice posizioni Not-a-number in PrDataTot (3)
81 % Cerco punti non disponibili (Nan) in PrDataTot (3)
82 Nan = zeros(size(PrDataTot,2));
83
84 for jj=1:size(PrDataTot,2)
85     temp = find(isnan(PrDataTot(:,jj)));
86     Nan(1:length(temp),jj) = temp;
87     clear temp
88 end
89 % Sostituisco ai punti Nan il valore -16 per visualizzarli nel
    plot finale
90
91 for jj = 1:size(PrDataTot,2)
92     for kk = 1:size(PrDataTot,1)
93         if isnan(PrDataTot(kk,jj))
94             PrDataTot(kk,jj)= -16;
95         end
96     end
97 end
98 %%
99 % Matrice logica di verifica
100 Logic = all(Nan==not_available);
101 find(Logic==0) % Restituisce 0 nel caso siano presenti Nan

```

Codice A.1.1: Main_caricamento_profili(codice completo)

I punti non validi vengono ricercati tramite la parte di codice evidenziata in [A.1.2](#).

```

80 % Nan: NA: Matrice posizioni Not-a-number in PrDataTot (3)
81 % Cerco punti non disponibili (Nan) in PrDataTot (3)
82 Nan = zeros(size(PrDataTot,2));
83
84 for jj=1:size(PrDataTot,2)
85     temp = find(isnan(PrDataTot(:,jj)));
86     Nan(1:length(temp),jj) = temp;

```

```

87     clear temp
88 end

```

Codice A.1.2: Main_caricamento_profili(ricerca punti NaN all'interno dei profili)

I punti con sigla "NA" all'interno delle colonne letterali sono stati ricercati tramite la porzione di codice riportata in [A.1.3](#).

```

62 % NA: Matrice posizioni punti Na in C_PrDataTot (1)
63 % Cerco punti non disponibili (NA) in C_PrDataTot (1)
64 not_available = zeros(1,size(C_PrDataTot,2));
65
66 for jj=1:size(C_PrDataTot,2)
67     temp = find(C_PrDataTot(:,jj) == 'NA');
68     not_available(1:length(temp),jj) = temp;
69     clear temp
70 end

```

Codice A.1.3: Main_caricamento_profili(ricerca punti con sigla NA nelle colonne letterali)

Le posizioni ottenute tramite i codici [A.1.2](#) e [A.1.3](#) sono state confrontate per assicurarsi non fossero stati omessi dei punti. Nel codice [A.1.4](#) viene mostrata la porzione di codice contenente la matrice logica utilizzata per il controllo.

```

99 % Matrice logica di verifica
100 Logic = all(Nan==not_available);
101 find(Logic==0) % Restituisce 0 nel caso siano presenti Nan

```

Codice A.1.4: Main_caricamento_profili(matrice logica di controllo)

A.2 Codice: Main_generazione_profili_test

Il codice utilizzato per la creazione dei profili in cui sono stati inseriti degli intervalli di dati mancanti, da utilizzare per valutare gli algoritmi di ricostruzione dei profili è riportato in [A.2.1](#).

```
1 % Inizializzazione generatore di numeri casuali
2 rng(1,'twister');
3 %%
4 % Richiamo file profili e selezione profilo test scelto (
   Profilo 115)
5 FileList = {'Profili_1_50', 'Profili_51_100', 'Profili_101_125',
   , 'Profili_126_150'};
6 LimitMissing = 10;
7 PrDataTot = Funzione_profili_di_carico(FileList,LimitMissing);
8 PrDataTest = PrDataTot(:,115);
9 PrDataTest_1 = PrDataTest;
10 PrDataTest_2 = PrDataTest;
11 %%
12 % Creazione struttura bug (posizioni primo NaN e numero
   elementi per gruppo)
13 bug{2} = [];
14 %%
15 % Primo caso test: (118 bug, ampiezza minima 1, ampiezza
   massima 3)
16 n_bug_1 = 118;%118; %360 %408
17 sizeminbug_1 = 1;
18 sizeMAXbug_1 = 3;
19
20 % Secondo caso test: (56 bug, ampiezza minima 4, ampiezza
   massima 22)
21 n_bug_2 = 56;
22 sizeminbug_2 = 4;
23 sizeMAXbug_2 = 22;
24 %%
```

```

25 % Nessun profilo presenta dei valori NaN entro i primi valori,
    viene considerato un limite "imperturbato" del profilo test,
        calcolato
26 % come rapporto tra il numero dei punti e il numero max dei bug
    che si vogliono inserire.
27 %Tale limite viene considerato invariato anche per il secondo
    profilo che si va a generare.
28 lim_bug = size(PrDataTest,1)/n_bug_1;
29
30 if lim_bug ~= isinteger(lim_bug)
31     lim_bug = ceil(lim_bug);
32 end
33 % Creazione posizioni randomiche primo valore NaN
34 % Le posizioni randomiche vengono generate considerando una
    distanza minima di 24 posizioni, per evitare di creare buchi
        troppo grandi in questo caso
35 % studio e un'ulteriore margine "cautelativo" di 2 posizioni.
36 bond = 30; %24; % Distanza minima tra i buchi
37 margin_c = 2; % Margine "cautelativo"
38
39 % Primo caso
40 random_1_pos = ones(n_bug_1,1);
41
42 for i = 1:length(random_1_pos)-1
43     random_1_pos(i) = sort(randi([lim_bug size(PrDataTest,1)]))
        ;
44     if(random_1_pos(i+1) < random_1_pos(i) + bond)
45         random_1_pos(i+1) = random_1_pos(i) + bond + margin_c;
46     else
47         i=i+1;
48     end
49 end
50 random_1_pos = sort(random_1_pos); % Ordinamento posizioni bug
51 random_1_num = randi([sizeminbug_1 sizeMAXbug_1],n_bug_1,1); %
    Ampiezze bug

```

```

52 % Secondo caso
53 random_2_pos = ones(n_bug_2,1);
54
55 for i = 1:length(random_2_pos)-1
56     random_2_pos(i) = sort(randi([lim_bug size(PrDataTest,1)]))
57     ;
58     if(random_2_pos(i+1) < random_2_pos(i) + bond)
59         random_2_pos(i+1) = random_2_pos(i) + bond + 2 *
60             margin_c;
61     else
62         i=i+1;
63     end
64 end
65 random_2_pos = sort(random_2_pos); % Ordinamento posizioni bug
66 random_2_num = randi([sizeminbug_2 sizeMAXbug_2],n_bug_2,1); %
67     Ampiezze bug
68 % Primo caso test
69 bug{1}.pos = random_1_pos; % Popolamento struttura bug (
70     posizioni)
71 bug{1}.num = random_1_num; % Popolamento struttura bug (numero
72     elementi per gruppo)
73 % Secondo caso test
74 bug{2}.pos = random_2_pos; % Popolamento struttura bug (
75     posizioni)
76 bug{2}.num = random_2_num; % Popolamento struttura bug (numero
77     elementi per gruppo)
78 %%
79 % Inserimento bug in PrDataTest
80
81 % Primo caso test
82 for n = 1:size(random_1_pos,1)
83     PrDataTest_1(bug{1}.pos(n):(bug{1}.pos(n)+bug{1}.num(n)-1))
84         = NaN;
85 end
86 index_1 = find(isnan(PrDataTest_1));

```

```

79 PrDataTest_1(index_1) = -16; % Sostituzione -16 ai NaN inseriti
80 index_1_control = find(isnan(PrDataTest_1)); % Controllo sulla
    presenza di ulteriori NaN
81 display(index_1_control);
82
83 % Secondo caso test
84 for n = 1:size(random_2_pos,1)
85     PrDataTest_2(bug{2}.pos(n):(bug{2}.pos(n)+bug{2}.num(n)-1))
        = NaN;
86 end
87 index_2 = find(isnan(PrDataTest_2));
88 PrDataTest_2(index_2) = -16; % Sostituzione -16 ai NaN inseriti
89 index_2_control = find(isnan(PrDataTest_2)); % Controllo sulla
    presenza di ulteriori NaN
90 display(index_2_control);

```

Codice A.2.1: Main_generazione_profili_test(codice completo)

A.3 Codice: function_ricostruzione_profili_interpolazione

Il codice utilizzato per la ricostruzione tramite interpolazione lineare è riportato in [A.3.1](#).

```

1 function [Profs , NewData, ToComplete] =
    function_ricostruzione_interpolazione(PrData_newFIN,
        LimitMissing, MAX_1, Ratio)
2
3 usrPars.limHigh = 10*MAX_1;          % limite assoluto massimo
4 usrPars.limLow = 0;                  % limite assoluto minimo (con 0 non
    accetta dati negativi)
5 usrPars.profiles = 1:size(PrData_newFIN,2); % profili da
    elaborare
6 usrPars.profSize = length(PrData_newFIN); % numero dati per
    profilo
7 usrPars.Ratio = Ratio-1; % Dati ricostruzione buco unitario

```

```

8 Profs{size((PrData_newFIN),2)} = [];
9
10 for i = 1:size(PrData_newFIN,2)
11     Profs{i}.rawdata = PrData_newFIN(:,i);
12 end
13
14 % Look out for err data
15 for ind1 = 1 : length(usrPars.profiles)
16     curProf = usrPars.profiles(ind1);
17     % Init data structure
18     Profs{curProf}.errData = [];
19     Profs{curProf}.errBlocks = [];
20     % Filter data: NOT NaN, >= limLow, <= limHigh
21     validPattern = and(Profs{curProf}.rawdata >= usrPars.limLow
22         ,Profs{curProf}.rawdata <= usrPars.limHigh);
23     Profs{curProf}.errData = find(validPattern == 0);
24     for ind2 = 1 : length(Profs{curProf}.errData)
25         if isempty(Profs{curProf}.errBlocks)
26             Profs{curProf}.errBlocks(1).pos = Profs{curProf}.
27                 errData(ind2);          % Position
28             Profs{curProf}.errBlocks(1).dim = 1;
29                                     % Size
30             Profs{curProf}.errBlocks(1).okBefore = Profs{
31                 curProf}.errData(ind2)-1; % Valid data BEFORE
32 elseif Profs{curProf}.errData(ind2-1)==Profs{curProf}.
33     errData(ind2)-1
34         Profs{curProf}.errBlocks(end).dim = Profs{curProf}.
35             errBlocks(end).dim+1;
36 else
37     Profs{curProf}.errBlocks(end+1).pos = Profs{curProf}
38         }.errData(ind2);
39     Profs{curProf}.errBlocks(end).dim = 1;
40     Profs{curProf}.errBlocks(end).okBefore = (Profs{
41         curProf}.errData(ind2)-1)-(Profs{curProf}.
42         errBlocks(end-1).pos+Profs{curProf}.errBlocks(

```

```

        end-1).dim);
34     end
35 end
36 Profs{curProf}.dataMean = mean(Profs{curProf}.rawdata(
    validPattern));
37 for ind3 = 1 : length(Profs{curProf}.errBlocks) % Valid
    data AFTER
38     if ind3 ~= length(Profs{curProf}.errBlocks)
39         Profs{curProf}.errBlocks(ind3).okAfter = abs((Profs{
            curProf}.errBlocks(ind3).pos + Profs{curProf}.
            errBlocks(ind3).dim)-(Profs{curProf}.errBlocks(
            ind3+1).pos - 1));
40     else
41         Profs{curProf}.errBlocks(ind3).okAfter = abs((Profs
            {curProf}.errBlocks(ind3).pos + Profs{curProf}.
            errBlocks(ind3).dim) - length(PrData_newFIN));
42     end
43 end
44 % Init data structure
45 Profs{curProf}.overData = [];
46 Profs{curProf}.overBlocks = [];
47 % Look for over data
48 Profs{curProf}.overData = find(validPattern==0);
49 end
50
51 % Cerco punti iniziali e finali per interpolazione
52 for indhor = 1 : size(Profs,2)
53     for indver = 1 : size(Profs{indhor}.errBlocks,2)
54         if (Profs{indhor}.errBlocks(indver).dim <= LimitMissing
            ) % Vincolo sul LimitMissing
55             Profs{indhor}.errBlocks(indver).bfaf = usrPars.
                Ratio * Profs{indhor}.errBlocks(indver).dim;
56             if (Profs{indhor}.errBlocks(indver).okBefore >= Profs{
                indhor}.errBlocks(indver).bfaf && Profs{indhor}.
                errBlocks(indver).okAfter >= Profs{indhor}.errBlocks

```

```

(indver).bfaf)
57 Profs{indhor}.errBlocks(indver).ToCompleteStart = (
    Profs{indhor}.errBlocks(indver).pos - 1) - (Profs
    {indhor}.errBlocks(indver).bfaf-1);
58 Profs{indhor}.errBlocks(indver).ToCompleteStop = (
    Profs{indhor}.errBlocks(indver).pos + Profs{
    indhor}.errBlocks(indver).dim) + (Profs{indhor}.
    errBlocks(indver).bfaf-1);
59 else
60     Profs{indhor}.errBlocks(indver).ToCompleteStart =
        [];
61     Profs{indhor}.errBlocks(indver).ToCompleteStop =
        [];
62 end
63 else
64     Profs{indhor}.errBlocks(indver).ToCompleteStart =
        [];
65     Profs{indhor}.errBlocks(indver).ToCompleteStop =
        [];
66 end
67 end
68 ERRBLOCKS{indhor} = Profs{indhor}.errBlocks;
69 end
70 NewData{size(Profs,2)}=[];
71 ToComplete{size(Profs,2)} = [];
72
73 for indhor = 1 : size(ToComplete,2)
74     NewData{indhor} = Profs{indhor}.rawdata;
75     for indver = 1 : size(Profs{indhor}.errBlocks,2)
76         if (isempty(Profs{indhor}.errBlocks(indver).
            ToCompleteStart) == 0 && isempty(Profs{indhor}.
            errBlocks(indver).ToCompleteStop) == 0)
77             ToComplete{indhor}.index(indver).startBefore =
                Profs{indhor}.errBlocks(indver).ToCompleteStart;
78             ToComplete{indhor}.index(indver).stopBefore = Profs

```

```

79         {indhor}.errBlocks(indver).pos - 1;
ToComplete{indhor}.index(indver).startAfter = Profs
        {indhor}.errBlocks(indver).pos + Profs{indhor}.
        errBlocks(indver).dim;
80 ToComplete{indhor}.index(indver).stopAfter = Profs{
        indhor}.errBlocks(indver).ToCompleteStop;
81 ToComplete{indhor}.index(indver).dim = Profs{indhor}
        }.errBlocks(indver).dim;
82
83 curProf = Profs{indhor}.rawdata; % Funzione Profili
84 curErrBlk = ToComplete{indhor}.index(indver); %
        Funzione ToComplete per interpolazione
85 if (size([Profs{indhor}.errBlocks(1:end).
        ToCompleteStart],2) == size([Profs{indhor}.errBlocks
        (1:end)],2))
86     ToComplete{indhor}.index(indver).stdBefore = std(
        curProf(curErrBlk.startBefore : curErrBlk.
        stopBefore));
87 ToComplete{indhor}.index(indver).stdAfter = std(
        curProf(curErrBlk.startAfter : curErrBlk.
        stopAfter));
88 ToComplete{indhor}.index(indver).std = max([
        ToComplete{indhor}.index(indver).stdBefore
        ToComplete{indhor}.index(indver).stdAfter]);
89 ToComplete{indhor}.index(indver).interp = transpose
        (interp1([curErrBlk.startBefore:curErrBlk.
        stopBefore,curErrBlk.startAfter:curErrBlk.
        stopAfter],curProf([curErrBlk.startBefore:
        curErrBlk.stopBefore,curErrBlk.startAfter:
        curErrBlk.stopAfter]),[curErrBlk.stopBefore+1:
        curErrBlk.startAfter-1],'linear'));
90 ToComplete{indhor}.index(indver).vectvariations =
        ToComplete{indhor}.index(indver).std*randn(
        ToComplete{indhor}.index(indver).dim,1);
91 NewData{indhor}((ToComplete{indhor}.index(indver).

```



```

    stopBefore + 1):(ToComplete{indhor}.index(indver
    ).startAfter - 1)) = max(ToComplete{indhor}.
    index(indver).interp + ToComplete{indhor}.index(
    indver).vectvariations,0);

92
93     elseif (size([Profs{indhor}.errBlocks(1:end) .
    ToCompleteStart],2) < size([Profs{indhor}.errBlocks
    (1:end)],2))
94         ToComplete{indhor} = [];
95         NewData{indhor}(1:(length(PrData_newFIN))) = zeros(
    size(1:length(PrData_newFIN)));
96     end
97     end
98 end
99 end
100 end

```

Codice A.3.1: Function_ricostruzione_profili_interpolazione(codice completo)

Per la ricostruzione mediante interpolazione sono stati prima settati i limiti:

- Massimo dei valori medio dei profili ad alto consumo: "*MAX_I*";
- Limite superiore dei valori: "*usrPars.limHigh = 10·MAX_I*";
- Limite inferiore (fissato a 0 in quanto si tratta di profili di carico): "*usrPars.limLow = 0*";
- Numero di profili da analizzare: "*usrPars.profiles = 1:size(PrData_newFIN,2)*";
- Lunghezza dei profili da analizzare: "*usrPars.profSize = length(PrData_newFIN)*";
- Numero minimo di punti (punti per ricostruzione intervallo unitario): "*Ratio = 2*";

Il Ratio rappresenta il numero minimo di punti da considerare prima e dopo l'intervallo di valori presente, esso è stato esposto nel paragrafo 3.1.3. Verranno quindi ricercati all'interno dei profili i punti che superano il limite superiore e il limite inferiore (ai punti NaN è

stato associato il valore -16) e viene creata quindi la matrice logica denominata “*validPattern*” e vengono ricercate le posizioni a cui è associato il valore logico 0, come riportato nella porzione di codice [A.3.2](#).

```
21     validPattern = and(Profs{curProf}.rawdata >= usrPars.limLow
      ,Profs{curProf}.rawdata <= usrPars.limHigh);
22     Profs{curProf}.errData = find(validPattern == 0);
```

Codice A.3.2: Main_ricostruzione_profili_interpolazione(matrice logica punti non validi)

A partire dalla “*curProf.errData*” verrà creata la struttura “*Profs.errBlocks*”, questa struttura conterrà diversi campi, i più importanti saranno i seguenti:

- **.pos**: questo campo sarà popolato dai valori delle posizioni relativi ai primi valori NaN trovati per ogni intervallo di dati;
- **.dim**: questo campo sarà popolato dal valore della dimensione dell’intervallo di dati, considerando il primo valore incluso;
- **.bfaf**: questo campo conterrà il numero di valori che verranno considerati prima e dopo l’intervallo di dati per la ricostruzione dell’intervallo mancante;
- **.okBefore**: in questo campo verranno inseriti il numero di punti buoni prima dell’intervallo di dati, quindi il numero di valori buoni fra l’intervallo considerato e l’intervallo precedente;
- **.okAfter**: in questo campo verranno inseriti il numero di punti buoni dopo l’intervallo di dati, quindi il numero di valori buoni fra l’intervallo considerato e l’intervallo successivo.

Per quanto riguarda il primo campo relativo alla posizione, esso sarà uguale al primo elemento dei blocchi di posizioni contenute in “*ProfscurProf.errData*” in quanto il primo elemento di ogni blocco rappresenta il punto di inizio della ricostruzione come riportato in [A.3.3](#).

```

25         Profs{curProf}.errBlocks(1).pos = Profs{curProf}.
           errData(ind2);           % Position

```

Codice A.3.3: Main_ricostruzione_profili_interpolazione(primo elemento non valido degli intervalli mancanti)

Per considerare quindi solo il primo elemento di ogni gruppo e non elementi con posizioni consecutive è stato introdotto l'indice “*ind2*” sulla lunghezza dei profili [A.3.4](#).

```

28         elseif Profs{curProf}.errData(ind2-1)==Profs{curProf}.
           errData(ind2)-1
29         Profs{curProf}.errBlocks(end).dim = Profs{curProf}.
           errBlocks(end).dim+1;

```

Codice A.3.4: Main_ricostruzione_profili_interpolazione(determinazione ampiezza intervalli)

Facendo incrementare nel caso di punti consecutivi il campo relativo alla dimensione dell'intervallo di dati. Il campo “*okBefore*” verrà quindi popolato considerando la posizione precedente la prima posizione di ogni intervallo, quindi “*ProfscurProf.errData(ind2)-1*” a cui vanno sottratti le posizioni totali dell'intervallo precedente, come riportato nella porzione di codice [A.3.5](#).

```

33         Profs{curProf}.errBlocks(end).okBefore = (Profs{
           curProf}.errData(ind2)-1)-(Profs{curProf}.
           errBlocks(end-1).pos+Profs{curProf}.errBlocks(
           end-1).dim);

```

Codice A.3.5: Main_ricostruzione_profili_interpolazione(punti validi prima dell'intervallo non valido)

Allo stesso modo i punti validi dopo ogni intervallo vengono calcolati come mostrato nel codice [A.3.6](#).

39

```

    Profs{curProf}.errBlocks(ind3).okAfter = abs((Profs{
        curProf}.errBlocks(ind3).pos + Profs{curProf}.
        errBlocks(ind3).dim) - (Profs{curProf}.errBlocks(
        ind3+1).pos - 1));

```

Codice A.3.6: Main_ricostruzione_profilo_interpolazione(punti validi dopo l'intervallo non valido)

L'indice “*ind3*” ha lo stesso campo dell'indice “*ind2*” ed è stato aggiunto per semplificare la ricerca di eventuali malfunzionamenti nel codice. Infine, il campo “*.bfaf*” è ottenuto moltiplicando il valore del Ratio per la dimensione dell'intervallo mancante, il calcolo è mostrato nel codice [A.3.7](#).

55

```

    Profs{indhor}.errBlocks(indver).bfaf = usrPars.
        Ratio * Profs{indhor}.errBlocks(indver).dim;

```

Codice A.3.7: Main_ricostruzione_profilo_interpolazione(intervallo utile per la ricostruzione)

Come detto questo campo rappresenta il numero di punti per la ricostruzione che verranno considerati prima e dopo l'intervallo mancante. Se il numero di punti necessario alla ricostruzione è maggiore del numero di punti buoni disponibile prima o dopo l'intervallo, il profilo non verrà ricostruito, in caso contrario le posizioni relative alla ricostruzione verranno salvate nella struttura “*ToCompleteFIN*”, anche questa composta da diversi campi:

- **.startBefore**: in questo campo verrà inserita la prima posizione buona prima dell'intervallo di dati mancanti, da cui inizierà la ricostruzione;
- **.stopBefore**: in questo campo verrà inserita l'ultima posizione buona per la ricostruzione, sarà quindi l'indice pari alla posizione del primo valore NaN-1;
- **.startAfter**: in questo campo verrà inserita la prima posizione buona dell'intervallo di ricostruzione dopo l'intervallo mancante, quindi la posizione dell'ultimo valore NaN+1;

- **.stopAfter**: in questo campo verrà inserita l'ultima posizione dopo l'intervallo di dati considerato per la ricostruzione;
- **.stdBefore**: deviazione standard intervallo di valori precedenti l'intervallo di dati mancante ;
- **.stdAfter**:deviazione standard intervallo di valori successivi all'intervallo di dati mancante;
- **.std**:massimo tra i valori in “.stdBefore” ed in “.stdAfter”;
- **.interp**: nuovi valori ottenuti con l'interpolazione lineare;
- **.vectvariations**: prodotto ottenuto dal valore relativo al campo “.std” moltiplicato un intervallo di valori casuale di dimensione pari all'intervallo mancante considerato.

I valori relativi ai primi 4 campi possono essere ottenuti semplicemente, nel modo riportato in [A.3.8](#).

```

77         ToComplete{indhor}.index(indver).startBefore =
           Profs{indhor}.errBlocks(indver).ToCompleteStart;
78         ToComplete{indhor}.index(indver).stopBefore = Profs
           {indhor}.errBlocks(indver).pos - 1;
79         ToComplete{indhor}.index(indver).startAfter = Profs
           {indhor}.errBlocks(indver).pos + Profs{indhor}.
           errBlocks(indver).dim;
80         ToComplete{indhor}.index(indver).stopAfter = Profs{
           indhor}.errBlocks(indver).ToCompleteStop;

```

Codice A.3.8: Main_ricostruzione_profili_interpolazione(punti estremi intervalli usati per la ricostruzione)

Per quanto riguarda gli ultimi 5 campi precedentemente esposti, questi vengono calcolati a partire da queste ultime espressioni descritte, considerando per semplicità le seguenti espressioni, mostrate in [A.3.9](#), per abbreviare le righe di codice.

```

83         curProf = Profs{indhor}.rawdata; % Funzione Profili
84         curErrBlk = ToComplete{indhor}.index(indver); %
            Funzione ToComplete per interpolazione

```

Codice A.3.9: Main_ricostruzione_profili_interpolazione(abbreviazioni)

Si ottengono quindi i risultati voluti tramite il codice [A.3.10](#).

```

86         ToComplete{indhor}.index(indver).stdBefore = std(
            curProf(curErrBlk.startBefore : curErrBlk.
                stopBefore));
87         ToComplete{indhor}.index(indver).stdAfter = std(
            curProf(curErrBlk.startAfter : curErrBlk.
                stopAfter));
88         ToComplete{indhor}.index(indver).std = max([
            ToComplete{indhor}.index(indver).stdBefore
            ToComplete{indhor}.index(indver).stdAfter]);
89         ToComplete{indhor}.index(indver).interp = transpose
            (interp1([curErrBlk.startBefore:curErrBlk.
                stopBefore,curErrBlk.startAfter:curErrBlk.
                stopAfter],curProf([curErrBlk.startBefore:
                curErrBlk.stopBefore,curErrBlk.startAfter:
                curErrBlk.stopAfter]),[curErrBlk.stopBefore+1:
                curErrBlk.startAfter-1], 'linear'));
90         ToComplete{indhor}.index(indver).vectvariations =
            ToComplete{indhor}.index(indver).std*randn(
                ToComplete{indhor}.index(indver).dim,1);

```

Codice A.3.10: Main_ricostruzione_profili_interpolazione(ultime grandezze per la ricostruzione)

Si calcola quindi la deviazione standard prima della zona di dati mancanti, (*stdBefore*) e dopo, (*stdAfter*). Tra i due valori si sceglie il massimo e si scrive il valore nel campo “*std*”. Successivamente vengono calcolati i nuovi valori mediante l’interpolazione lineare, un set di valori della stessa dimensione dell’intervallo di dati viene generato dall’espressione con-

tenuta nel campo “*vectvariations*”, questi ultimi però vengono calcolati in modo casuale moltiplicando dei numeri generati casualmente mediante “*randn*” e il valore della deviazione standard scelto precedentemente. A questo punto i valori che verranno inseriti nei profili ricostruiti in “*NewData*” saranno ottenuti dalla somma dei valori ottenuti mediante l’interpolazione più il corrispettivo valore generato casualmente in “*vectvariations*”, confrontando punto per punto i valori con lo zero scegliendo il massimo tra i due, la porzione di codice relativa alla scrittura dei nuovi dati è riportata in [A.3.11](#):

```

91      NewData{indhor}((ToComplete{indhor}.index(indver).
          stopBefore + 1):(ToComplete{indhor}.index(indver)
          ).startAfter - 1)) = max(ToComplete{indhor}.
          index(indver).interp + ToComplete{indhor}.index(
          indver).vectvariations,0);

```

Codice A.3.11: Main_ricostruzione_profili_interpolazione(dati ottenuti dopo la ricostruzione)

A.4 Codice: function_ricostruzione_profili_random

Il codice utilizzato per la ricostruzione tramite punti casuali è riportato in [A.4.1](#).

```

1  function [PrDataTest_2_with_random]=
      funzione_ricostruzione_random(PrDataTest, PrDataTest_1,
      PrDataTest_2, sizeMAXbug_2, random_1_num, random_1_pos,
      random_2_num,random_2_pos)
2
3  % Sovrascrittura PrDataTot_new
4  PrData_newFIN = [PrDataTest PrDataTest_1 PrDataTest_2];
5  LimitMissingUsers = 2 * sizeMAXbug_2;
6  MAX_1 = 0.4356;
7
8  for k = 1: length(LimitMissingUsers)
9      LimitMissing = LimitMissingUsers(k);

```

```

10     Ratio = 2; % Rapporto di ricostruzione di riferimento per
        buco unitario
11     [Profs , NewData, ToComplete] = Dinamic_Limit(PrData_newFIN
        , LimitMissing, MAX_1, Ratio);
12
13     % Ricostruzione grandezze
14     for i = 1 : size(Profs,2)
15         Profs_FIN{k,i} = Profs{1,i};
16         ToComplete_FIN{k,i} = ToComplete{1,i};
17         NewData_FIN{k,i} = NewData{1,i};
18     end
19 end
20
21 NewDataFIN = cell2mat(NewData_FIN);
22 %%
23 for hor = 2:3
24     for ver = 1 : size(ToComplete_FIN{hor}.index,2)
25         if (hor == 2)
26             Pro2(ver, 1) = max(PrData_newFIN(ToComplete{hor}.index(
                ver).startBefore : ToComplete{hor}.index(ver).
                stopBefore, hor));
27             Pro2(ver, 2) = min(PrData_newFIN(ToComplete{hor}.index(
                ver).startBefore : ToComplete{hor}.index(ver).
                stopBefore, hor));
28             Pro2(ver, 3) = max(PrData_newFIN(ToComplete{hor}.index(
                ver).startAfter : ToComplete{hor}.index(ver).
                stopAfter, hor));
29             Pro2(ver, 4) = min(PrData_newFIN(ToComplete{hor}.index(
                ver).startAfter : ToComplete{hor}.index(ver).
                stopAfter, hor));
30         elseif (hor == 3)
31             Pro3(ver, 1) = max(PrData_newFIN(ToComplete{hor}.index(
                ver).startBefore : ToComplete{hor}.index(ver).
                stopBefore, hor));
32             Pro3(ver, 2) = min(PrData_newFIN(ToComplete{hor}.index(

```



```

        ver).startBefore : ToComplete{hor}.index(ver).
        stopBefore , hor));
33     Pro3(ver, 3) = max(PrData_newFIN(ToComplete{hor}.index(
        ver).startAfter : ToComplete{hor}.index(ver).
        stopAfter , hor));
34     Pro3(ver, 4) = min(PrData_newFIN(ToComplete{hor}.index(
        ver).startAfter : ToComplete{hor}.index(ver).
        stopAfter , hor));
35     end
36 end
37 end
38 %%
39
40 for i = 1 : length(Pro2)
41     MAX1(i,1) = max([Pro2(i,1) Pro2(i,3)]);
42     MIN1(i,1) = min([Pro2(i,2) Pro2(i,4)]);
43 end
44 %
45 PrDataTest_1_with_random = PrDataTest_1;
46
47 for k = 1 : length(MAX1)
48     a = MIN1(k,1);
49     b = MAX1(k,1);
50     r = (b-a).*rand(random_1_num(k),1) + a;
51     r_range = [min(r) max(r)];
52     PrDataTest_1_with_random(random_1_pos(k,1) :
        random_1_pos(k,1) + random_1_num(k,1)-1,1)=r(1:
        random_1_num(k),1);
53 end
54 clear k;
55
56 for i = 1 : size(Pro3,1)
57     MAX2(i,1) = max([Pro3(i,1) Pro3(i,3)]);
58     MIN2(i,1) = min([Pro3(i,2) Pro3(i,4)]);
59 end

```

```

60 %
61 PrDataTest_2_with_random = PrDataTest_2;
62
63 for k = 1 : length(MAX2)
64     a = MIN2(k,1);
65     b = MAX2(k,1);
66     r = (b-a).*rand(random_2_num(k),1) + a;
67     r_range = [min(r) max(r)];
68     PrDataTest_2_with_random(random_2_pos(k,1) :
        random_2_pos(k,1) + random_2_num(k,1) -1,1)=r(1:
        random_2_num(k),1);
69 end
70
71 end

```

Codice A.4.1: function_ricostruzione_profilo_random(codice completo)

Per la ricostruzione mediante punti casuali viene sfruttata la struttura “*ToComplete*” che contiene le gli indici da cui partire per la ricostruzione nei campi “*StartBefore*”, “*StopBefore*”, “*StartAfter*” e “*StopAfter*”. Tramite un semplice ciclo *for* vengono calcolati i massimi e i minimi locali in ogni intervallo precedente e seguente l’intervallo di dati mancanti, vengono poi scelti il massimo e il minimo globale tra i due, la porzione di codice è riportata in [A.4.2](#).

```

23 for hor = 2:3
24     for ver = 1 : size(ToComplete_FIN{hor}.index,2)
25         if (hor == 2)
26             Pro2(ver, 1) = max(PrData_newFIN(ToComplete{hor}.index(
                ver).startBefore : ToComplete{hor}.index(ver).
                stopBefore, hor));
27             Pro2(ver, 2) = min(PrData_newFIN(ToComplete{hor}.index(
                ver).startBefore : ToComplete{hor}.index(ver).
                stopBefore, hor));
28             Pro2(ver, 3) = max(PrData_newFIN(ToComplete{hor}.index(
                ver).startAfter : ToComplete{hor}.index(ver).
                stopAfter, hor));

```

```

29         Pro2(ver, 4) = min(PrData_newFIN(ToComplete{hor}.index(
        ver).startAfter : ToComplete{hor}.index(ver).
        stopAfter, hor));
30     elseif (hor == 3)
31         Pro3(ver, 1) = max(PrData_newFIN(ToComplete{hor}.index(
        ver).startBefore : ToComplete{hor}.index(ver).
        stopBefore, hor));
32         Pro3(ver, 2) = min(PrData_newFIN(ToComplete{hor}.index(
        ver).startBefore : ToComplete{hor}.index(ver).
        stopBefore, hor));
33         Pro3(ver, 3) = max(PrData_newFIN(ToComplete{hor}.index(
        ver).startAfter : ToComplete{hor}.index(ver).
        stopAfter, hor));
34         Pro3(ver, 4) = min(PrData_newFIN(ToComplete{hor}.index(
        ver).startAfter : ToComplete{hor}.index(ver).
        stopAfter, hor));
35     end
36 end
37 end

```

Codice A.4.2: function_ricostruzione_profili_random(massimi e minimi locali e globali)

I numeri casuali contenuti nell'intervallo considerato vengono quindi calcolati mediante un ciclo *for*, passando attraverso il vettore "*r*" che genera dei numeri casuali tra il minimo e il massimo specificati, definiti rispettivamente "*a*" e "*b*" in questo ciclo e mediante il comando *rand* genera il numero esatto di valori grazie all'informazione contenuta in "*random_1_num(k)*", la riga specificata è riportata in A.4.4:

```

50     r = (b-a).*rand(random_1_num(k),1) + a;

```

Codice A.4.3: function_ricostruzione_profili_random(generazione numeri casuali)

I valori ottenuti vengono poi inseriti nel profilo mediante una semplice sovrascrittura.

```

52      PrDataTest_1_with_random(random_1_pos(k,1) :
      random_1_pos(k,1) + random_1_num(k,1)-1,1)=r(1:
      random_1_num(k),1);

```

Codice A.4.4: function_ricostruzione_profili_random(scrittura valori ottenuti)

Il codice completo riguardante questo piccolo ciclo è riportato in [A.4.5](#)

```

47 for k = 1 : length(MAX1)
48     a = MIN1(k,1);
49     b = MAX1(k,1);
50     r = (b-a).*rand(random_1_num(k),1) + a;
51     r_range = [min(r) max(r)];
52     PrDataTest_1_with_random(random_1_pos(k,1) :
      random_1_pos(k,1) + random_1_num(k,1)-1,1)=r(1:
      random_1_num(k),1);
53 end

```

Codice A.4.5: function_ricostruzione_profili_random(ciclo di ricostruzione)

A.5 Codice: Function_energia

Questa funzione consente di dividere i profili di carico in profili in base al consumo medio per una famiglia di 4 persone. Tale valore è riportato nella Tabella [3.1](#) ed è pari a 3600 kWh/anno.

I profili di carico risulteranno divisi quindi in due gruppi, "profili ad alto consumo" e "profili a basso consumo".

```

64 % Questa funzione divide sulla base del valore dell'energia
      media i profili
65 % di carico in profili ad alto e a basso consumo.
66 function [MAX, PrData, set] = Funzione_Energia(PrDataTot)
67 %%
68 % Consumo medio annuo famiglia italiana 4 persone (FONTE ARERA)

```

```

69 Energia_media = 3600; % kWh
70 Vincolo_Energia = Energia_media/size(PrDataTot,1);
71
72 %% Calcolo medie e divisione tra profili ALTO o basso consumo
73 Media = zeros(1,size(PrDataTot,2));
74 Media_low = zeros(1,size(PrDataTot,2));
75 Media_high= zeros(1,size(PrDataTot,2));
76
77 for col = 1 : size(PrDataTot,2)
78
79     iii = find(PrDataTot(:,col) >= 0);
80     Media(1,col)=sum(PrDataTot(iii,col))/size(PrDataTot(iii,col),1);
81
82     if (Media(1,col) < Vincolo_Energia)
83         Media_low(1,col) = Media(1,col);
84     else
85         Media_high(1,col) = Media(1,col);
86     end
87 end
88
89 high_profiles = 0;
90 low_profiles = 1;
91 disp('Numero di profili ALTO consumo'); disp(size(Media_high(
    Media_high>0),2)); % Numero di Profili ALTO CONSUMO
92 disp('Numero di profili basso consumo'); disp(size(Media_low(
    Media_low>0),2)); % Numero di Profili basso CONSUMO
93
94 prompt = 'high=0, low=1: '; % Questa riga permette di
    scegliere se analizzare i Profili ALTO consumo o quelli
    basso consumo
95 set = input(prompt); %0 %1
96
97 MAX_high = max(Media_high);
98 MAX_low = max(Media_low);

```

```

99
100 %%
101 % Setto i massimi in modo automatico dopo aver scelto quali
    profili analizzare
102 flag_profiles = 0;
103 MAX = max(Media_high);
104 Media = Media_high;
105
106 if (set == 1)
107     flag_profiles = 1;
108     MAX = max(Media_low);
109     Media = Media_low;
110 end
111
112 % Creazione nuova PrData per cluster
113 PrData = [];
114
115 for i = 1 : size(Media,2)
116     if ( Media(1,i) > 0 )
117         PrData(:,end+1) = PrDataTot(:,i);
118     end
119 end
120
121 if (set == 0)
122     PrData_high = PrData;
123 elseif (set ==1)
124     PrData_low = PrData;
125 end
126
127 end

```

Codice A.5.1: function_energia(codice completo)

A.6 Codice: Main_creazione_aggregato_residenziale

Il codice presentato in questa sezione consente di creare l'aggregato di tipo residenziale in per unità, partendo dai profili ricostruiti.

```
1 load('workspaceLOW');
2 rng('shuffle');
3
4 LimitMissingUsers = 10:17;
5 for k = 1: length(LimitMissingUsers)
6     LimitMissing = LimitMissingUsers(k);
7     Ratio = 2; % Rapporto di ricostruzione di riferimento per
           buco unitario
8     [Profs , NewData, ToComplete] =
           function_ricostruzione_profili_interpolazione(
           PrData_newFIN, LimitMissing, MAX, Ratio);
9     % Ricostruzione grandezze
10    for i = 1 : size(Profs,2)
11        Profs_FIN{k,i} = Profs{1,i};
12        ToComplete_FIN{k,i} = ToComplete{1,i};
13        NewData_FIN{k,i} = NewData{1,i};
14    end
15 end
16
17 nnn = find(isnan(PrData(:,:)));
18 PrData(nnn)= -16;
19
20 for n = 1:size(NewData_FIN,1)
21     new = cell2mat(NewData_FIN(n,:));
22 end
23
24 %%
25
26 Basso_consumo = new;
27
```

```

28 % Rimuovo colonne completamente nulle
29 Basso_consumo( :, all(~Basso_consumo,1)) = [];
30
31 % Passaggio a profili potenza nei quarti d'ora
32 Basso_consumo = Basso_consumo./0.25;
33
34 %%
35
36 Basso_consumo_3kW = [];
37 Basso_consumo_4_5kW = [];
38
39 for k = 1 : size(Basso_consumo,2)
40     if (max(Basso_consumo(:,k)) <= 3)
41         Basso_consumo_3kW(:,end+1) = Basso_consumo(:,k);
42     else
43         Basso_consumo_4_5kW(:,end+1) = Basso_consumo(:,k);
44     end
45 end
46
47 %%
48 % Basso consumo 3 kW
49 numProfili = size(Basso_consumo_3kW,2);
50 numAggregati = 1;
51 dimAggregati = size(Basso_consumo_3kW,2);
52
53 [Indici_Aggregati] = funzione_indici_aggregati(numProfili,
54     numAggregati, dimAggregati);
55
56
57 Indici_Aggregati_Basso_consumo = Indici_Aggregati;
58
59 for ver = 1 : size(Indici_Aggregati,1)
60     for hor = 1 : size(Indici_Aggregati,2)
61         Potenze_basso(:,ver) = Basso_consumo(:,
62             Indici_Aggregati_Basso_consumo(ver,hor));
63     end
64 end

```



```

61 Potenze_Basso_TOT_3kW(i,hor) = sum(Potenze_basso(i,:));
62 end
63 end
64 end
65 clear numProfili numAggregati dimAggregati Indici_Aggregati;
66
67 % Basso consumo 4.5 kW
68 numProfili = size(Basso_consumo_4_5kW,2);
69 numAggregati = 1;
70 dimAggregati = size(Basso_consumo_4_5kW,2);
71
72 [Indici_Aggregati] = function_indici_aggregati(numProfili,
        numAggregati, dimAggregati);
73
74 Indici_Aggregati_Basso_consumo = Indici_Aggregati;
75
76 for ver = 1 : size(Indici_Aggregati,1)
77     for hor = 1 : size(Indici_Aggregati,2)
78         Potenze_basso(:,ver) = Basso_consumo(:,
                Indici_Aggregati_Basso_consumo(ver,hor));
79         for i = 1 : size(Potenze_basso,1)
80             Potenze_Basso_TOT_4_5kW(i,hor) = sum(Potenze_basso(i,:));
81         end
82     end
83 end
84
85 %%
86
87 Pot_tot = ones(size(Potenze_Basso_TOT_3kW,1),1);
88 for i = 1: 35040
89     Pot_tot(i,1) = Potenze_Basso_TOT_3kW(i,1) +
            Potenze_Basso_TOT_4_5kW(i,1);
90 end
91
92 Aggregato_TOT = Pot_tot/((size(Basso_consumo_3kW,2)*3) + (size(

```

```
Basso_consumo_4_5kW,2)*4.5));
```

Codice A.6.1: Main_creazione_aggregato_residenziale(codice completo)

La funzione "*function_indici_aggregati*" gestisce gli indici dei profili residenziali che compongono l'aggregato.

```
1 function [Indici_Aggregati] = function_indici_aggregati(  
    numProfili, numAggregati, dimAggregati)  
2  
3 Indici_Aggregati = ones(dimAggregati,numAggregati); % Matrice  
    indici aggregati  
4 if (numAggregati*dimAggregati <= numProfili)  
5     disp('Aggregati completamente distinti');  
6     Indici_Aggregati = reshape(randperm(numProfili,  
        numAggregati*dimAggregati),[dimAggregati numAggregati]);  
7 elseif (numProfili > dimAggregati) && (factorial(numProfili)/(  
    factorial(numProfili-dimAggregati)*factorial(dimAggregati)))  
8 % Le condizioni precedenti permettono di ottenere aggregati  
9 % parzialmente distinti( all'interno di ogni aggregato non  
    si ripete lo stesso profilo e non ci saranno due  
    aggregati perfettamente identici  
10 % avremo almeno un profilo diverso).  
11 % La condizione con il fattoriale viene legata alla  
    condizione di numero massimo di sequenze di elementi n  
    in classe k,  
12 %in modo tale che non si ripeta la stessa sequenza, questa  
    definita come il numero di combinazioni di elementi in  
    classe k  
13 % matematicamente:  $n!/(k!*(n-k)!)$   
14 disp('Aggregati parzialmente distinti');  
15 maxAggrDistinti = fix(numProfili/dimAggregati);  
16 Indici_Aggregati = reshape(randperm(numProfili,dimAggregati  
    *maxAggrDistinti),[dimAggregati maxAggrDistinti]); %  
    massima porzione di matrice completamente diversa  
17
```

```

18 for ind=1:(numAggregati-maxAggrDistinti)
19     checkKey = prod(Indici_Aggregati); % Chiavi univoche
        per controllare differenza
20     flagContinue = true;
21     while (flagContinue == true) % Ripete fino a quando
        risulta diverso
22         dataAdd = transpose(randperm(numProfili,dimAggregati)
            ); % Genera un nuovo aggregato
23         if isempty(find(checkKey==prod(dataAdd), 1)) %
            Controlla che sia diverso
24             flagContinue = false;
25             Indici_Aggregati = [Indici_Aggregati,dataAdd]; % "
                Attacca" i nuovi dati ai precedenti
26         end
27     end
28 end
29
30 else
31     disp('Impossibile generare aggregati senza elementi
        completamente ripetuti');
32     Indici_Aggregati = [];
33 end
34
35 disp(Indici_Aggregati);
36
37 end

```

Codice A.6.2: function_indici_aggregati(codice completo)

A.7 Codice: Function_estrazione_profili_non_residenziali

I dati dei profili non residenziali, contenuti in file Excel, sono stati estratti tramite il codice [A.7.1](#).

I valori di Potenza attiva e reattiva sono poi stati ordinati in strutture diverse.

```

1 function [P_profili_comm, Q_profili_comm, P_profili_ind,
   Q_profili_ind, P_profili_uff, Q_profili_uff] =
   funzione_ricostruzione_profili_da_excel(name, type)
2
3 str = strcat(name,type);
4
5 Profili = {};
6 for strInd = 1 : size(str,2)
7
8     [status,sheets,xlFormat] = xlsfinfo(str(strInd));
9     numPagine = size(sheets,2);
10
11 for pagInd = 1 : numPagine
12
13     [Dati, Testo] = xlsread(str(strInd),pagInd);
14
15     if (strInd == 1)
16         Profili(pagInd).Comm_num = Dati;
17         Profili(pagInd).Comm_txt = Testo;
18         Profili(pagInd).Comm_txt(1,:) = [];
19         Profili(pagInd).Comm_txt(:,4:end) = [];
20
21     elseif (strInd == 2)
22         Profili(pagInd).Ind_num = Dati;
23         Profili(pagInd).Ind_txt = Testo;
24         Profili(pagInd).Ind_txt(1,:) = [];
25         Profili(pagInd).Ind_txt(:,4:end) = [];
26
27     elseif (strInd == 3)
28         Profili(pagInd).Uff_num = Dati;
29         Profili(pagInd).Uff_txt = Testo;
30         Profili(pagInd).Uff_txt(1,:) = [];
31         Profili(pagInd).Uff_txt(:,4:end) = [];
32 end

```

```

33 end
34 end
35
36 % Per utenti COMMERCIALI
37 for prInd = 1 : size(Profili,2)
38
39     LogicP(:,prInd) = strcmp(Profili(prInd).Comm_txt(:,1),'EA')
40     ;
41
42     % Potenza Attiva
43     A = find(LogicP(:,prInd) == 1);
44     P_A = Profili(prInd).Comm_num(A,:);
45     L = transpose(P_A);
46     P_profili_comm(:,prInd) = L(:)./0.25;
47
48     % Potenza Reattiva
49     B = find(LogicP(:,prInd) == 0);
50     P_Q = Profili(prInd).Comm_num(B,:);
51     R = transpose(P_Q);
52     Q_profili_comm(:,prInd) = R(:)./0.25;
53
54     Control = [];
55     for i=1:size(A,1)
56         Control = find(A(i,1)==B(:,1));
57         if (isempty(Control)== 0)
58             disp('Problema')
59         end
60     end
61 end
62
63 % Per utenti Industriali
64 for prInd = 1 : size(Profili,2)
65
66     LogicP(:,prInd) = strcmp(Profili(prInd).Ind_txt(:,1),'EA');

```

```

67
68         % Potenza Attiva
69         A = find(LogicP(:,prInd) == 1);
70         P_A = Profili(prInd).Ind_num(A,:);
71         L = transpose(P_A);
72         P_profili_ind(:,prInd) = L(:)./0.25;
73
74         % Potenza Reattiva
75         B = find(LogicP(:,prInd) == 0);
76         P_Q = Profili(prInd).Ind_num(B,:);
77         R = transpose(P_Q);
78         Q_profili_ind(:,prInd) = R(:)./0.25;
79
80         Control = [];
81         for i=1:size(A,1)
82             Control = find(A(i,1)==B(:,1));
83             if (isempty(Control)== 0)
84                 disp('Problema')
85             end
86         end
87     end
88
89     % Per generico Uffici
90     for prInd = 1 : size(Profili,2)
91
92         LogicP(:,prInd) = strcmp(Profili(prInd).Uff_txt(:,1),'EA');
93
94         % Potenza Attiva
95         A = find(LogicP(:,prInd) == 1);
96         P_A = Profili(prInd).Uff_num(A,:);
97         L = transpose(P_A);
98         P_profili_uff(:,prInd) = L(:)./0.25;
99
100        % Potenza Reattiva
101        B = find(LogicP(:,prInd) == 0);

```

```

102     P_Q = Profili(prInd).Uff_num(B,:);
103     R = transpose(P_Q);
104     Q_profili_uff(:,prInd) = R(:)./0.25;
105
106     Control = [];
107     for i=1:size(A,1)
108         Control = find(A(i,1)==B(:,1));
109         if (isempty(Control)== 0)
110             disp('Problema')
111         end
112     end
113 end
114 %%
115 % NaN profili commerciali
116 P_profili_comm(find(isnan(P_profili_comm(:,:)))) = -16;
117 Q_profili_comm(find(isnan(Q_profili_comm(:,:)))) = -16;
118
119 % NaN profili industriali
120 P_profili_ind(find(isnan(P_profili_ind(:,:)))) = -16;
121 Q_profili_ind(find(isnan(Q_profili_ind(:,:)))) = -16;
122
123 % NaN profili generico uffici
124 P_profili_uff(find(isnan(P_profili_uff(:,:)))) = -16;
125 Q_profili_uff(find(isnan(Q_profili_uff(:,:)))) = -16;

```

Codice A.7.1: Function_ricostruzione_profili_da_excel(ciclo completo)

A.8 Codice: Main_valutazione_aggregati_nodi

In questa sezione viene esposto il codice utilizzato per la generazione degli aggregati nodali della rete test 4.1, tale codice è stato diviso in sottosezioni. I codici sono i seguenti:

- "Main_valutazione_aggregati_nodi";
- "Main_random_percentuali_profili";

- "Main_Fase_finale_programma";
- "Main_indici_rete_completo".

A.8.1 Codice: Main_valutazione_aggregati_nodi

In questo codice vengono valutati i rapporti percentuali dei vari tipi di profilo (residenziale, industriale, commerciale, generico uffici) rispetto all'energia totale.

```

1  % Time step (quarto d'ora) dei profili
2  timeStep_min = 15;
3
4  % Taglia trasformatori cabina MT/BT
5  A_trasf = 600; % Potenza nominale trasformatore ( kVA )
6
7  % Creazione rete
8  branchNames = {"PIS_Param_Magra","PIS_Param_Brenta","
    PIS_Param_Grosso","PIS_Param_Fiat","PIS_Param_Chieri"}; %#ok
    <CLARRSTR>
9  noPowNodes = [2052801, 205271, 204659, 208265, 2052802, 204946,
    203333, 203892];
10 [grid] = funzione_creazione_rete(branchNames, noPowNodes);
11
12 for Ind = 1 : size(grid.nodes,2)
13     grid.nodes(Ind).RES = 0;
14     grid.nodes(Ind).COMM = 0;
15     grid.nodes(Ind).UFF = 0;
16     grid.nodes(Ind).IND = 0;
17     grid.nodes(Ind).TOT = 0;
18 end
19 %
20
21 % Carico aggregato residenziale e vari profili di carico
22 % % Profili = dir('F:\CODICI\Profili_pu\*.mat');
23

```



```

24 for i = 1:size(Profili,1)
25     load(Profili(i).name)
26 end
27 %
28 % Elimino i profili a 6300 V
29 % Le tensioni dei profili a 6300 V sono state poste a zero (
    profilo industriale 5 e profilo uffici 1)
30 % La V del terzo profilo commerciale viene stata posta a zero
    in quanto
31 % l'andamento del profilo viene giudicato non utilizzabile
32 V_comm = [22000      400      0      22000      400];
    % Tensioni utenze commerciali
33 V_ind = [22000      22000      22000      400      0]; %
    Tensioni utenze industriali
34 V_uff = [0      400      22000      400
    22000]; % Tensioni utenze uffici
35
36 for i = 1 : 5
37     if (V_comm(1,i) == 0)
38         P_profili_comm_pu(:,i) = 0;
39     end
40     if (V_ind(1,i) == 0)
41         P_profili_ind_pu(:,i) = 0;
42     end
43     if (V_uff(1,i) == 0)
44         P_profili_uff_pu(:,i) = 0;
45     end
46
47 end
48
49
50 % Calcolo area sottesa ai profili di carico in pu (Energia in
    pu)
51 %
52 Energia_res_pu = timeStep_min/60 * sum(Aggregato_TOT);

```

```

53 %
54 Energia_comm_pu = timeStep_min/60 * sum(P_profili_comm_pu);
55 %
56 Energia_ind_pu = timeStep_min/60 * sum(P_profili_ind_pu);
57 %
58 Energia_uff_pu = timeStep_min/60 * sum(P_profili_uff_pu);
59 %
60 Energia_TOT = Energia_res_pu + Energia_comm_pu + Energia_ind_pu
    + Energia_uff_pu;
61
62 % Calcolo rapporti tra le energie
63 %
64 Rapp_RES = (Energia_res_pu ./ Energia_TOT) .* 100;
65 Rapp_COMM = (Energia_comm_pu ./ Energia_TOT) .* 100;
66 Rapp_IND = (Energia_ind_pu ./ Energia_TOT) .* 100;
67 Rapp_UFF = (Energia_uff_pu ./ Energia_TOT) .* 100;
68
69
70
71 %Variabili per visualizzazione rapporti
72
73 for i = 1: size(Rapp_RES,2)
74 Rapp_res_percent{i} = cellstr(num2str(Rapp_RES(:,i), "%g%%"));
75 end
76
77 for i = 1: size(Rapp_COMM,2)
78 Rapp_comm_percent{i} = cellstr(num2str(Rapp_COMM(:,i), "%g%%"));
79     ;
80 end
81
82 for i = 1: size(Rapp_IND,2)
83 Rapp_ind_percent{i} = cellstr(num2str(Rapp_IND(:,i), "%g%%"));
84 end
85
86 for i = 1: size(Rapp_UFF,2)

```

```

86 Rapp_uff_percent{i} = cellstr(num2str(Rapp_UFF(:,i), "%g%%"));
87 end

```

Codice A.8.1: Main_valutazione_aggregati_nodi(ciclo completo)

A.8.2 Codice: Main_random_percentuali_profili

Il codice presentato in questa sezione permette di generare delle percentuali casuali per le diverse tipologie di profilo per ogni nodo.

```

1  rng(1, 'twister')
2
3  Nodi_di_collegamento = [204874 208221 204871 2052801 205271
    204659 208265 2052802 204946 203333]; % qui sono inseriti i
    nodi di collegamento presenti nella rete elettrica,
4
    % questi nodi devono essere
    presenti SOLO in questo
    vettore,
5
    % in quanto non risulta
    necessario generare un
    aggregato per essi
6
7  % Si suddivide la rete in zone ALTAMENTE residenziali,
    ALTAMENTE industriali,
8  % ALTAMENTE commerciali e con ALTA concentrazione di uffici:
9  Alt_RES = [203844 204514 204216 203765 203615 205358 205024
    204730 204748 204813 204716 203319 203323 204249 203324
    205351 203874 203137 203923 203921 205304 204220 205638
    203937 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0];
10 Alt_IND = [0 0 205428 203385 203890 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0];
11 Alt_COMM = [203294 203548 204621 203845 204228 204171 204201
    204193 203202 203558 204834 203892 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

```

    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
12 Alt_UFF = [203975 203974 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0];
13
14 % Si suddivide la rete in zone MEDIAMENTE residenziali,
    MEDIAMENTE industriali,
15 % MEDIAMENTE commerciali e con MEDIA concentrazione di uffici:
16 Med_RES = [203548 204228 203975 203890 204834 203892 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0];
17 Med_IND = [203294 203845 204621 203202 204201 204171 203558
    205351 203874 205304 203937 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
18 Med_COMM = [0 204514 0 205428 203765 205358 205024 204730
    204748 203974 204716 203385 203323 203137 204220 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0];
19 Med_UFF = [203844 204216 204193 203615 204813 203319 204249
    203324 203923 203921 205638 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
20
21 % Si suddivide la rete in zone SCARSAMENTE residenziali,
    SCARSAMENTE industriali,
22 % SCARSAMENTE commerciali e con SCARSA concentrazione di uffici
    :
23 Scars_RES = [203845 204171 204201 204193 203202 203294 203974
    203385 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0];
24 Scars_IND = [203844 204228 203765 205358 205024 204813 204716
    203323 204249 203324 203137 204220 205638 203892 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0];
25 Scars_COMM = [204216 203615 204813 203975 203319 205351 203874
    203923 203921 205304 205638 203937 0 0 0 0 0 0 0 0 0 0 0 0

```

```

    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
26 Scars_UFF = [203548 204514 205428 0 0 203202 203765 204201
    204621 205358 204730 204748 203558 203890 204834 203323
    205304 203937 203892 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0];
27
28 % Controlli sui nodi
29 find(Alt_RES(:, :) == Alt_IND(:, :))
30 find(Alt_RES(:, :) == Alt_COMM(:, :))
31 find(Alt_RES(:, :) == Alt_UFF(:, :))
32 find(Alt_IND(:, :) == Alt_COMM(:, :))
33 find(Alt_IND(:, :) == Alt_UFF(:, :))
34 find (Alt_COMM(:, :) == Alt_UFF(:, :))
35 %
36 find (Med_RES(:, :) == Med_IND(:, :))
37 find(Med_RES(:, :) == Med_COMM(:, :))
38 find(Med_RES(:, :) == Med_UFF(:, :))
39 find(Med_IND(:, :) == Med_COMM(:, :))
40 find(Med_IND(:, :) == Med_UFF(:, :))
41 find(Med_COMM(:, :) == Med_UFF(:, :))
42 %
43 find (Alt_RES(:, :) == Med_RES(:, :))
44 find(Alt_RES(:, :) == Scars_RES(:, :))
45 find(Alt_IND(:, :) == Med_IND(:, :))
46 find(Alt_IND(:, :) == Scars_IND(:, :))
47 find(Alt_COMM(:, :) == Med_COMM(:, :))
48 find(Alt_COMM(:, :) == Scars_COMM(:, :))
49 find(Alt_UFF(:, :) == Med_UFF(:, :))
50 find(Alt_UFF(:, :) == Scars_UFF(:, :))
51
52 % Ciclo sui nodi della rete test
53 for i = 5 : size(grid.nodes,2)
54     if (ismember(grid.nodes(i).name, Nodi_di_collegamento) ==
        false)
55         disp(i) % Numero nodo

```

```

56 % Sorteggio prima la componente preponderante
57 if (ismember(grid.nodes(i).name, Alt_RES) == true)
58     grid.nodes(i).RES = randi([50, 60], 1);
59 elseif (ismember(grid.nodes(i).name, Alt_IND) == true)
60     grid.nodes(i).IND = randi([50, 60], 1);
61 elseif (ismember(grid.nodes(i).name, Alt_COMM) == true)
62     grid.nodes(i).COMM = randi([50, 60], 1);
63 elseif (ismember(grid.nodes(i).name, Alt_UFF) == true)
64     grid.nodes(i).UFF = randi([50, 60], 1);
65 end
66
67 % Le altre componenti vengono sorteggiate fino a quando
68     non risulta verificata la condizione sulla somma
69 while (not(grid.nodes(i).RES + grid.nodes(i).COMM +
70     grid.nodes(i).UFF + grid.nodes(i).IND == 100))
71     if (ismember(grid.nodes(i).name, Med_RES) == true)
72         grid.nodes(i).RES = randi([25, 35], 1);
73     end
74
75     if (ismember(grid.nodes(i).name, Med_IND) == true)
76         grid.nodes(i).IND = randi([25, 35], 1);
77     end
78
79     if (ismember(grid.nodes(i).name, Med_COMM) == true)
80         grid.nodes(i).COMM = randi([25, 35], 1);
81     end
82
83     if (ismember(grid.nodes(i).name, Med_UFF) == true)
84         grid.nodes(i).UFF = randi([25, 35], 1);
85     end
86
87     if (ismember(grid.nodes(i).name, Scars_RES) == true)
88         )
89         grid.nodes(i).RES = randi([5, 15], 1);
90     end

```

```

88
89         if (ismember(grid.nodes(i).name, Scars_IND) == true
90             )
91             grid.nodes(i).IND = randi([5, 15], 1);
92         end
93
94         if (ismember(grid.nodes(i).name, Scars_COMM) ==
95             true)
96             grid.nodes(i).COMM = randi([5, 15], 1);
97         end
98         if (ismember(grid.nodes(i).name, Scars_UFF) == true
99             )
100             grid.nodes(i).UFF = randi([5, 15], 1);
101         end
102     end
103
104     grid.nodes(i).TOT = grid.nodes(i).RES + grid.nodes(i).
105         IND + grid.nodes(i).COMM + grid.nodes(i).UFF;
106 end

```

Codice A.8.2: Main_random_percentuali_profili(codice completo)

Le righe 8 ÷ 27 rappresentano la suddivisione dei nodi per zone altamente, mediamente e scarsamente residenziali, industriali, commerciali e generico uffici.

A.8.3 Codice: Main_Fase_finale_programma

Il codice presentato in questa sezione permette di generare la struttura che conterrà i profili nodali finali. In questo codice sono stati inoltre suddivisi per ogni nodo i vari profili non domestici in per unità.

```

1 % Creazione struttura Profili_scelti_Nodi, in questa struttura
   saranno contenuti i
2 % valori di potenza installata e i profili di potenza attiva e
   reattiva

```

```

3 Profili_scelti_Nodi = {};
4 countNodi = 1;
5 for IndNodi = 1 : size(grid.nodes,2)
6
7     Profili_scelti_Nodi(IndNodi).Nodi = grid.nodes(IndNodi)
        .name;
8     Profili_scelti_Nodi(IndNodi).inBranch = grid.nodes(
        IndNodi).inBranch;
9     Profili_scelti_Nodi(IndNodi).Energia_res_pu = 0;
10    Profili_scelti_Nodi(IndNodi).Energia_ind_pu = 0;
11    Profili_scelti_Nodi(IndNodi).Energia_comm_pu = 0;
12    Profili_scelti_Nodi(IndNodi).Energia_uff_pu = 0;
13
14    Profili_scelti_Nodi(IndNodi).PinstallataNodo = 0;
15    Profili_scelti_Nodi(IndNodi).P_RES = 0;
16    Profili_scelti_Nodi(IndNodi).P_IND = 0;
17    Profili_scelti_Nodi(IndNodi).P_COMM = 0;
18    Profili_scelti_Nodi(IndNodi).P_UFF = 0;
19    Profili_scelti_Nodi(IndNodi).Potenza_installata = 0;
20
21 end
22
23 %%
24 % Nodi MAGRA
25 NumNodiMagra = 4;
26 Tipo_di_utenze = {'Residenziale', 'Industriale', 'Commerciale',
    'Uffici'};
27 % % Nodo %1 %2 %3 %4
28 Utente = [ 3, 1, 4, 4]; % Indici Rapporti percentuali utenti
    Magra
29
30 for IndType = 1 : size(Tipo_di_utenze,2)
31 for NodiMagra = 1 : NumNodiMagra
32 TMP_Magra(IndType,NodiMagra) = Utente(1,NodiMagra);
33 end

```



```

34 end
35
36 for ind = 1 : NumNodiMagra
37     Indici.Magra(ind).RES = TMP_Magra(1,ind);
38     Indici.Magra(ind).IND = TMP_Magra(2,ind);
39     Indici.Magra(ind).COMM = TMP_Magra(3,ind);
40     Indici.Magra(ind).UFF = TMP_Magra(4,ind);
41
42 end
43
44 %%
45 % Nodi BRENTA
46 NumNodiBrenta = 10;
47 Tipo_di_utenze = {'Residenziale', 'Industriale', 'Commerciale',
48                 'Uffici'};
49 % % Nodo %1 %2 %3 %4 %5 %6 %7 %8 %9 %10
50 Utente = [ 4, 1, 2, 4, 2, 4, 1, 1, 4, 1]; % Indici
51 Rapporti percentuali utenti Brenta
52
53 for IndType = 1 : size(Tipo_di_utenze,2)
54     for NodiBrenta = 1 : NumNodiBrenta
55         TMP_Brenta(IndType,NodiBrenta) = Utente(1,NodiBrenta);
56     end
57 end
58
59 for ind = 1 : NumNodiBrenta
60     Indici.Brenta(ind).RES = TMP_Brenta(1,ind);
61     Indici.Brenta(ind).IND = TMP_Brenta(2,ind);
62     Indici.Brenta(ind).COMM = TMP_Brenta(3,ind);
63     Indici.Brenta(ind).UFF = TMP_Brenta(4,ind);
64
65 end
66
67 %%
68 % Nodi GROSSO

```

```

67 NumNodiGrosso = 11;
68 Tipo_di_utenze = {'Residenziale', 'Industriale', 'Commerciale',
    'Uffici'};
69 % % Nodo %1 %2 %3 %4 %5 %6 %7 %8 %9 %10 %11
70 Utente = [ 4, 1, 2, 4, 4, 5, 4, 4, 2, 2, 4]; % Indici
    Rapporti percentuali utenti Grosso
71
72 for IndType = 1 : size(Tipo_di_utenze,2)
73 for NodiGrosso = 1 : NumNodiGrosso
74 TMP_Grosso(IndType,NodiGrosso) = Utente(1,NodiGrosso);
75 end
76 end
77
78 for ind = 1 : NumNodiGrosso
79     Indici.Grosso(ind).RES = TMP_Grosso(1,ind);
80     Indici.Grosso(ind).IND = TMP_Grosso(2,ind);
81     Indici.Grosso(ind).COMM = TMP_Grosso(3,ind);
82     Indici.Grosso(ind).UFF = TMP_Grosso(4,ind);
83
84 end
85
86 %%
87 % Nodi FIAT
88 NumNodiFiat = 8;
89 Tipo_di_utenze = {'Residenziale', 'Industriale', 'Commerciale',
    'Uffici'};
90 % % Nodo %30 %31 %32 %33 %34 %35 %36 %37
91 Utente = [ 4, 1, 4, 4, 5, 4, 2, 4]; % Indici Rapporti
    percentuali utenti Fiat
92
93 for IndType = 1 : size(Tipo_di_utenze,2)
94 for NodiFiat = 1 : NumNodiFiat
95 TMP_Fiat(IndType,NodiFiat) = Utente(1,NodiFiat);
96 end
97 end

```

```

98
99 for ind = 1 : NumNodiFiat
100     Indici.Fiat(ind).RES = TMP_Fiat(1,ind);
101     Indici.Fiat(ind).IND = TMP_Fiat(2,ind);
102     Indici.Fiat(ind).COMM = TMP_Fiat(3,ind);
103     Indici.Fiat(ind).UFF = TMP_Fiat(4,ind);
104
105 end
106 %%
107 % Nodi CHIERI
108 NumNodiChieri = 18;
109 Tipo_di_utenze = {'Residenziale', 'Industriale', 'Commerciale',
    'Uffici'};
110 % % Nodo %1 %2 %3 %4 %5 %6 %7 %8 %9 %10 %11 %12
    %13 %14 %15 %16 %17 %18
111 Utente = [ 1, 2, 4, 4, 1, 4, 4, 1, 2, 4, 4, 2,
    4, 4, 2, 4, 2, 4]; % Indici Rapporti percentuali
    utenti Chieri
112
113 for IndType = 1 : size(Tipo_di_utenze,2)
114     for NodiChieri = 1 : NumNodiChieri
115         TMP_Chieri(IndType,NodiChieri) = Utente(1,NodiChieri);
116     end
117 end
118
119 for ind = 1 : NumNodiChieri
120     Indici.Chieri(ind).RES = TMP_Chieri(1,ind);
121     Indici.Chieri(ind).IND = TMP_Chieri(2,ind);
122     Indici.Chieri(ind).COMM = TMP_Chieri(3,ind);
123     Indici.Chieri(ind).UFF = TMP_Chieri(4,ind);
124
125 end

```

Codice A.8.3: Main_Fase_finale_programma(codice completo)

A.8.4 Codice: Main_indici_rete_completo

Il codice presentato in questa sezione serve a popolare la struttura "*Profili_scelti_Nodi*" contenente i vari profili nodali totali.

```
1  % Magra 5 - 8
2  zoneType = ["RES", "IND", "COMM", "UFF"];
3
4  for ind1 = 1 : NumNodiMagra
5
6      for ind2 = 1 : length(zoneType)
7          tmpInd = getfield(Indici.Magra(ind1),zoneType(ind2));
8          if tmpInd == 0
9              tmpVal = 0;
10         elseif (ind2 == 1)
11             tmpVal = eval(strcat("Energia_",lower(zoneType(
12                 ind2)),"_pu","(1,",num2str(ind2),")"));
13             Profili_scelti_Nodi(ind1 + 4) = setfield(
14                 Profili_scelti_Nodi(ind1 + 4),strcat("
15                 Energia_",lower(zoneType(ind2)),"_pu"),
16                 tmpVal);
17
18         else
19             tmpVal = eval(strcat("Energia_",lower(zoneType(ind2
20                 )),"_pu","(1,",num2str(tmpInd),")"));
21         end
22         Profili_scelti_Nodi(ind1 + 4) = setfield(
23             Profili_scelti_Nodi(ind1 + 4),strcat("Energia_",
24             lower(zoneType(ind2)),"_pu"),tmpVal);
25     end
26 end
27
28 %%
29
30 % Brenta 9 - 18
```

```

24 zoneType = ["RES", "IND", "COMM", "UFF"];
25
26 for ind1 = 1 : NumNodiBrenta
27
28     for ind2 = 1 : length(zoneType)
29         tmpInd = getfield(Indici.Brenta(ind1),zoneType(ind2));
30         if tmpInd == 0
31             tmpVal = 0;
32         elseif (ind2 == 1)
33             tmpVal = eval(strcat("Energia_",lower(zoneType(
34                 ind2)),"_pu","(1,",num2str(ind2),")"));
35             Profili_scelti_Nodi(ind1 + 8) = setfield(
36                 Profili_scelti_Nodi(ind1 + 8),strcat("
37                 Energia_",lower(zoneType(ind2)),"_pu"),
38                 tmpVal);
39
40         else
41             tmpVal = eval(strcat("Energia_",lower(zoneType(ind2
42                 )), "_pu", "(1,", num2str(tmpInd), ")"));
43         end
44         Profili_scelti_Nodi(ind1 + 8) = setfield(
45             Profili_scelti_Nodi(ind1 + 8),strcat("Energia_",
46             lower(zoneType(ind2)),"_pu"),tmpVal);
47     end
48 end
49
50 %%
51
52 % Grosso 19 - 29
53 zoneType = ["RES", "IND", "COMM", "UFF"];
54
55 for ind1 = 1 : NumNodiGrosso
56
57     for ind2 = 1 : length(zoneType)
58         tmpInd = getfield(Indici.Grosso(ind1),zoneType(ind2));

```

```

52     if tmpInd == 0
53         tmpVal = 0;
54     elseif (ind2 == 1)
55         tmpVal = eval(strcat("Energia_",lower(zoneType(
                    ind2))), "_pu", "(1, ", num2str(ind2), ")"));
56         Profili_scelti_Nodi(ind1 + 18) = setfield(
                    Profili_scelti_Nodi(ind1 + 18), strcat("
                    Energia_", lower(zoneType(ind2)), "_pu"),
                    tmpVal);
57
58     else
59         tmpVal = eval(strcat("Energia_",lower(zoneType(ind2
                    )), "_pu", "(1, ", num2str(tmpInd), ")"));
60     end
61     Profili_scelti_Nodi(ind1 + 18) = setfield(
                    Profili_scelti_Nodi(ind1 + 18), strcat("Energia_",
                    lower(zoneType(ind2)), "_pu"), tmpVal);
62 end
63 end
64
65 %%
66 % Fiat 30 - 37
67 zoneType = ["RES", "IND", "COMM", "UFF"];
68
69 for ind1 = 1 : NumNodiFiat
70
71     for ind2 = 1 : length(zoneType)
72         tmpInd = getfield(Indici.Fiat(ind1), zoneType(ind2));
73         if tmpInd == 0
74             tmpVal = 0;
75         elseif (ind2 == 1)
76             tmpVal = eval(strcat("Energia_",lower(zoneType(
                    ind2))), "_pu", "(1, ", num2str(ind2), ")"));
77             Profili_scelti_Nodi(ind1+29) = setfield(
                    Profili_scelti_Nodi(ind1+29), strcat("

```

```

    Energia_", lower(zoneType(ind2)), "_pu"),
    tmpVal);

78
79     else
80         tmpVal = eval(strcat("Energia_", lower(zoneType(ind2
            )), "_pu", "(1, ", num2str(tmpInd), ")"));
81     end
82     Profili_scelti_Nodi(ind1+29) = setfield(
        Profili_scelti_Nodi(ind1+29), strcat("Energia_", lower
            (zoneType(ind2)), "_pu"), tmpVal);
83     end
84 end
85
86 %%
87 % Chieri 38 - 55
88 zoneType = ["RES", "IND", "COMM", "UFF"];
89
90 for ind1 = 1 : NumNodiChieri
91
92     for ind2 = 1 : length(zoneType)
93         tmpInd = getfield(Indici.Chieri(ind1), zoneType(ind2));
94         if tmpInd == 0
95             tmpVal = 0;
96         elseif (ind2 == 1)
97             tmpVal = eval(strcat("Energia_", lower(zoneType(
                ind2)), "_pu", "(1, ", num2str(ind2), ")"));
98             Profili_scelti_Nodi(ind1+37) = setfield(
                Profili_scelti_Nodi(ind1+37), strcat("
                Energia_", lower(zoneType(ind2)), "_pu"),
                tmpVal);
99
100        else
101            tmpVal = eval(strcat("Energia_", lower(zoneType(ind2
                )), "_pu", "(1, ", num2str(tmpInd), ")"));
102        end

```

```

103     Profili_scelti_Nodi(ind1+37) = setfield(
        Profili_scelti_Nodi(ind1+37),strcat("Energia_",lower
        (zoneType(ind2)),"_pu"),tmpVal);
104     end
105 end
106 %%
107 Ind_Magra = 5 : 8;
108
109 Load_Magra_RES_riferimento = [0; 0; 25; 201.5];
110 Load_Magra_3phase_riferimento = [0; 0; 581; 209.5];
111
112 for i = 1 : size(Load_Magra_RES_riferimento,1)
113     P_RIF_Magra(i,1) = (Load_Magra_RES_riferimento(i,1) +
        Load_Magra_3phase_riferimento(i,1));
114 end
115
116 for Ind_Nodi_Magra = 1: size(Ind_Magra,2)
117
118     Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
        PinstallataNodo = P_RIF_Magra((Ind_Nodi_Magra));
119
120     if Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
        Energia_res_pu > 0
121         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).P_RES =
            P_RIF_Magra(Ind_Nodi_Magra,1) * ((grid.nodes(
                Ind_Magra(Ind_Nodi_Magra)).RES / 100));
122     else
123         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
            P_RES = 0;
124     end
125
126     if Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
        Energia_ind_pu > 0
127
128         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).

```



```

129         P_IND = P_RIF_Magra(Ind_Nodi_Magra,1) * ((grid.
130         nodes(Ind_Magra(Ind_Nodi_Magra)).IND / 100));
131
132     else
133
134         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
135         P_IND = 0;
136     end
137
138     %
139     if Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
140         Energia_comm_pu > 0
141
142         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
143         P_COMM = P_RIF_Magra(Ind_Nodi_Magra,1) *
144         (((grid.nodes(Ind_Magra(Ind_Nodi_Magra)).
145         COMM / 100)));
146     else
147
148         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
149         P_COMM = 0;
150     end
151
152     %
153     if Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
154         Energia_uff_pu > 0
155
156         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
157         P_UFF = P_RIF_Magra(Ind_Nodi_Magra,1) * (((
158         grid.nodes(Ind_Magra(Ind_Nodi_Magra)).UFF /
159         100)));
160     else
161
162         Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).

```

```

P_UFF = 0;

152 end
153 %
154 Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
    Potenza_installata = Profili_scelti_Nodi(Ind_Magra(
        Ind_Nodi_Magra)).P_RES + Profili_scelti_Nodi(
        Ind_Magra(Ind_Nodi_Magra)).P_COMM +
        Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).P_IND
        + Profili_scelti_Nodi(Ind_Magra(Ind_Nodi_Magra)).
        P_UFF;

155
156 %
157 Ind_int = Ind_Magra(Ind_Nodi_Magra);
158 if (ismember(grid.nodes(Ind_int).name,
    Nodi_di_collegamento) == false)
159     Profili_scelti_Nodi(Ind_int).TOT =
        Profili_scelti_Nodi(Ind_int).P_RES .*
        Aggregato_TOT;
160     Profili_scelti_Nodi(Ind_int).TOT_Q = tan(acos(0.92)
        ) .* Aggregato_TOT;
161
162     for ind2=2:length(zoneType)
163
164         tmpInd = getfield(Indici.Magra(Ind_Nodi_Magra),
            zoneType(ind2));
165         if not(tmpInd == 0)
166             Profili_scelti_Nodi(Ind_int).TOT =
                Profili_scelti_Nodi(Ind_int).TOT +
                getfield(Profili_scelti_Nodi(Ind_int),
                    strcat("P_",zoneType(ind2))) .* eval(
                        strcat("P_profili_",lower(zoneType(ind2)
                            ),"_pu(:,",num2str(tmpInd),",")"));
167             Profili_scelti_Nodi(Ind_int).TOT_Q =
                Profili_scelti_Nodi(Ind_int).TOT_Q +
                getfield(Profili_scelti_Nodi(Ind_int),

```

```

168         strcat("P_",zoneType(ind2))) .* eval(
169             strcat("Q_profili_",lower(zoneType(ind2)
170                 ),"_pu(:,",num2str(tmpInd),")"));
171     end
172     else
173         Profili_scelti_Nodi(Ind_int).TOT = zeros(size(
174             eval(strcat("P_profili_",lower(zoneType(ind2)
175                 )), "_pu(:,",num2str(tmpInd),")")),1),1);
176         Profili_scelti_Nodi(Ind_int).TOT_Q =
177             Profili_scelti_Nodi(Ind_int).TOT;
178     end
179 end
180 %
181 Ind_Brenta = 9 : 18;
182 Load_Brenta_RES_riferimento = [427; 241; 0; 71; 129.5; 150.5;
183     0; 0; 163.5; 0];
184 Load_Brenta_3phase_riferimento = [1421; 746; 0; 468; 866;
185     328.5; 0; 0; 451.3; 0];
186 for i = 1 : size(Load_Brenta_RES_riferimento,1)
187     P_RIF_Brenta(i,1) = (Load_Brenta_RES_riferimento(i,1) +
188         Load_Brenta_3phase_riferimento(i,1));
189 end
190 for Ind_Nodi_Brenta = 1: size(Ind_Brenta,2)
191     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
192         PinstallataNodo = P_RIF_Brenta((Ind_Nodi_Brenta));
193
194     if Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
195         Energia_res_pu > 0
196         Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).P_RES

```

```

= P_RIF_Brenta(Ind_Nodi_Brenta,1) * ((grid.nodes(
Ind_Brenta(Ind_Nodi_Brenta)).RES / 100));
192 else
193     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
        P_RES = 0;
194 end
195
196 if Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
    Energia_ind_pu > 0
197
198     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
        P_IND = P_RIF_Brenta(Ind_Nodi_Brenta,1) * ((grid
        .nodes(Ind_Brenta(Ind_Nodi_Brenta)).IND / 100));
199
200 else
201
202     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
        P_IND = 0;
203 end
204
205 %
206 if Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
    Energia_comm_pu > 0
207
208     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)
        ).P_COMM = P_RIF_Brenta(Ind_Nodi_Brenta,1)
        * (((grid.nodes(Ind_Brenta(Ind_Nodi_Brenta))
        .COMM / 100)));
209
210 else
211
212     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
        P_COMM = 0;
213
214 end
%

```

```

215     if Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
        Energia_uff_pu > 0
216
217
218         Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)
            ).P_UFF = P_RIF_Brenta(Ind_Nodi_Brenta,1) *
            (((grid.nodes(Ind_Brenta(Ind_Nodi_Brenta)).
                UFF / 100)));
219
220     else
221
222         Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
            P_UFF = 0;
223     end
224     %
225     Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
        Potenza_installata = Profili_scelti_Nodi(Ind_Brenta(
            Ind_Nodi_Brenta)).P_RES + Profili_scelti_Nodi(
            Ind_Brenta(Ind_Nodi_Brenta)).P_COMM +
            Profili_scelti_Nodi(Ind_Brenta(Ind_Nodi_Brenta)).
            P_IND + Profili_scelti_Nodi(Ind_Brenta(
            Ind_Nodi_Brenta)).P_UFF;
226
227     %
228     Ind_int = Ind_Brenta(Ind_Nodi_Brenta);
229     if (ismember(grid.nodes(Ind_int).name,
        Nodi_di_collegamento) == false)
230         Profili_scelti_Nodi(Ind_int).TOT =
            Profili_scelti_Nodi(Ind_int).P_RES .*
            Aggregato_TOT;
231         Profili_scelti_Nodi(Ind_int).TOT_Q = tan(acos(0.92)
            ) .* Aggregato_TOT;
232
233         for ind2=2:length(zoneType)
234

```

```

235         tmpInd = getfield(Indici.Brenta(Ind_Nodi_Brenta
           ),zoneType(ind2));
236     if not(tmpInd == 0)
237         Profili_scelti_Nodi(Ind_int).TOT =
           Profili_scelti_Nodi(Ind_int).TOT +
           getfield(Profili_scelti_Nodi(Ind_int),
           strcat("P_",zoneType(ind2))) .* eval(
           strcat("P_profili_",lower(zoneType(ind2)
           ),"_pu(:,",num2str(tmpInd),")"));
238         Profili_scelti_Nodi(Ind_int).TOT_Q =
           Profili_scelti_Nodi(Ind_int).TOT_Q +
           getfield(Profili_scelti_Nodi(Ind_int),
           strcat("P_",zoneType(ind2))) .* eval(
           strcat("Q_profili_",lower(zoneType(ind2)
           ),"_pu(:,",num2str(tmpInd),")"));
239
240     end
241 end
242 else
243     Profili_scelti_Nodi(Ind_int).TOT = zeros(size(
           eval(strcat("P_profili_",lower(zoneType(ind2)
           )), "_pu(:,",num2str(tmpInd),")"),1),1);
244     Profili_scelti_Nodi(Ind_int).TOT_Q =
           Profili_scelti_Nodi(Ind_int).TOT;
245 end
246 end
247 %
248 Ind_Grosso = 19 : 29;
249
250 Load_Grosso_RES_riferimento = [24; 0; 393; 53.5; 310; 296; 264;
           111; 196; 280.5; 433];
251 Load_Grosso_3phase_riferimento = [119.5; 0; 1438.24; 419.5;
           1005.5; 1348.28; 1215.3; 471.06; 636.12; 1216.62; 1370.68];
252
253 for i = 1 : size(Load_Grosso_RES_riferimento,1)

```

```

254     P_RIF_Grosso(i,1) = (Load_Grosso_RES_riferimento(i,1) +
        Load_Grosso_3phase_riferimento(i,1));
255 end
256
257 for Ind_Nodi_Grosso = 1: size(Ind_Grosso,2)
258
259     Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
        PinstallataNodo = P_RIF_Grosso((Ind_Nodi_Grosso));
260
261     if Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
        Energia_res_pu > 0
262         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).P_RES
            = P_RIF_Grosso(Ind_Nodi_Grosso,1) * ((grid.nodes(
                Ind_Grosso(Ind_Nodi_Grosso)).RES / 100));
263     else
264         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
            P_RES = 0;
265     end
266
267     if Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
        Energia_ind_pu > 0
268
269         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
            P_IND = P_RIF_Grosso(Ind_Nodi_Grosso,1) * ((grid
                .nodes(Ind_Grosso(Ind_Nodi_Grosso)).IND / 100));
270
271     else
272
273         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
            P_IND = 0;
274     end
275
276     %
277     if Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
        Energia_comm_pu > 0

```

```

278
279         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)
                ).P_COMM = P_RIF_Grosso(Ind_Nodi_Grosso,1)
                * (((grid.nodes(Ind_Grosso(Ind_Nodi_Grosso))
                .COMM / 100)));

280     else
281
282         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
                P_COMM = 0;
283     end
284
285     %
286     if Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
                Energia_uff_pu > 0
287
288
289         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)
                ).P_UFF = P_RIF_Grosso(Ind_Nodi_Grosso,1) *
                (((grid.nodes(Ind_Grosso(Ind_Nodi_Grosso)).
                UFF / 100)));
290
291     else
292
293         Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
                P_UFF = 0;
294     end
295     %
296     Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
                Potenza_installata = Profili_scelti_Nodi(Ind_Grosso(
                Ind_Nodi_Grosso)).P_RES + Profili_scelti_Nodi(
                Ind_Grosso(Ind_Nodi_Grosso)).P_COMM +
                Profili_scelti_Nodi(Ind_Grosso(Ind_Nodi_Grosso)).
                P_IND + Profili_scelti_Nodi(Ind_Grosso(
                Ind_Nodi_Grosso)).P_UFF;
297

```



```

298 %
299 Ind_int = Ind_Grosso(Ind_Nodi_Grosso);
300 if (ismember(grid.nodes(Ind_int).name,
301 Nodi_di_collegamento) == false)
302     Profili_scelti_Nodi(Ind_int).TOT =
303         Profili_scelti_Nodi(Ind_int).P_RES .*
304         Aggregato_TOT;
305     Profili_scelti_Nodi(Ind_int).TOT_Q = tan(acos(0.92)
306         ) .* Aggregato_TOT;
307
308     for ind2=2:length(zoneType)
309
310         tmpInd = getfield(Indici.Grosso(Ind_Nodi_Grosso
311             ),zoneType(ind2));
312         if not(tmpInd == 0)
313             Profili_scelti_Nodi(Ind_int).TOT =
314                 Profili_scelti_Nodi(Ind_int).TOT +
315                 getfield(Profili_scelti_Nodi(Ind_int),
316                     strcat("P_",zoneType(ind2))) .* eval(
317                         strcat("P_profili_",lower(zoneType(ind2)
318                             ),"_pu(:,",num2str(tmpInd),")"));
319             Profili_scelti_Nodi(Ind_int).TOT_Q =
320                 Profili_scelti_Nodi(Ind_int).TOT_Q +
321                 getfield(Profili_scelti_Nodi(Ind_int),
322                     strcat("P_",zoneType(ind2))) .* eval(
323                         strcat("Q_profili_",lower(zoneType(ind2)
324                             ),"_pu(:,",num2str(tmpInd),")"));
325
326         end
327     end
328 else
329     Profili_scelti_Nodi(Ind_int).TOT = zeros(size(
330         eval(strcat("P_profili_",lower(zoneType(ind2)
331             )), "_pu(:,",num2str(tmpInd),")")),1,1);
332     Profili_scelti_Nodi(Ind_int).TOT_Q =

```

```

                                Profili_scelti_Nodi(Ind_int).TOT;
316         end
317     end
318     Ind_Fiat = 30 : 37;
319
320     Load_Fiat_RES_riferimento = [0; 398; 335.5; 95; 203; 154.5;
                                276; 280];
321     Load_Fiat_3phase_riferimento = [0; 1416.12; 1308; 501.5; 768;
                                825; 1153; 1334];
322
323     for i = 1 : size(Load_Fiat_RES_riferimento,1)
324         P_RIF_Fiat(i,1) = (Load_Fiat_RES_riferimento(i,1) +
                                Load_Fiat_3phase_riferimento(i,1));
325     end
326
327     for Ind_Nodi_Fiat = 1: size(Ind_Fiat,2)
328
329         Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
                                PinstallataNodo = P_RIF_Fiat((Ind_Nodi_Fiat));
330
331         if Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
                                Energia_res_pu > 0
332             Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).P_RES =
                                P_RIF_Fiat(Ind_Nodi_Fiat,1) * ((grid.nodes(Ind_Fiat(
                                Ind_Nodi_Fiat)).RES / 100));
333         else
334             Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).P_RES
                                = 0;
335         end
336
337         if Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
                                Energia_ind_pu > 0
338
339             Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).P_IND
                                = P_RIF_Fiat(Ind_Nodi_Fiat,1) * ((grid.nodes(

```

```

Ind_Fiat(Ind_Nodi_Fiat)).IND / 100));
340
341 else
342
343     Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).P_IND
        = 0;
344
345 end
346
347 %
348 if Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
    Energia_comm_pu > 0
349
        Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
            P_COMM = P_RIF_Fiat(Ind_Nodi_Fiat,1) * (((
                grid.nodes(Ind_Fiat(Ind_Nodi_Fiat)).COMM /
                100)));
350
351 else
352
353     Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).P_COMM
        = 0;
354
355 end
356
357 %
358 if Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
    Energia_uff_pu > 0
359
        Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
            P_UFF = P_RIF_Fiat(Ind_Nodi_Fiat,1) * (((
                grid.nodes(Ind_Fiat(Ind_Nodi_Fiat)).UFF /
                100)));
360
361 else
362
363     Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).P_UFF

```

```

= 0;

364 end
365 %
366 Profili_scelti_Nodi(Ind_Fiat(Ind_Nodi_Fiat)).
    Potenza_installata = Profili_scelti_Nodi(Ind_Fiat(
        Ind_Nodi_Fiat)).P_RES + Profili_scelti_Nodi(Ind_Fiat
        (Ind_Nodi_Fiat)).P_COMM + Profili_scelti_Nodi(
        Ind_Fiat(Ind_Nodi_Fiat)).P_IND + Profili_scelti_Nodi
        (Ind_Fiat(Ind_Nodi_Fiat)).P_UFF;

367
368 %
369 Ind_int = Ind_Fiat(Ind_Nodi_Fiat);
370 if (ismember(grid.nodes(Ind_int).name,
    Nodi_di_collegamento) == false)
371     Profili_scelti_Nodi(Ind_int).TOT =
        Profili_scelti_Nodi(Ind_int).P_RES .*
        Aggregato_TOT;
372     Profili_scelti_Nodi(Ind_int).TOT_Q = tan(acos(0.92)
        ) .* Aggregato_TOT;

373
374 for ind2=2:length(zoneType)
375
376     tmpInd = getfield(Indici.Fiat(Ind_Nodi_Fiat),
        zoneType(ind2));
377     if not(tmpInd == 0)
378         Profili_scelti_Nodi(Ind_int).TOT =
            Profili_scelti_Nodi(Ind_int).TOT +
            getfield(Profili_scelti_Nodi(Ind_int),
            strcat("P_",zoneType(ind2))) .* eval(
            strcat("P_profili_",lower(zoneType(ind2)
            ),"_pu(:,",num2str(tmpInd),",")"));
379         Profili_scelti_Nodi(Ind_int).TOT_Q =
            Profili_scelti_Nodi(Ind_int).TOT_Q +
            getfield(Profili_scelti_Nodi(Ind_int),
            strcat("P_",zoneType(ind2))) .* eval(

```

```

380         strcat("Q_profili_",lower(zoneType(ind2)
381         ),"_pu(:,",num2str(tmpInd),")"));
382     end
383 end
384 else
385     Profili_scelti_Nodi(Ind_int).TOT = zeros(size(
386         eval(strcat("P_profili_",lower(zoneType(ind2
387         )), "_pu(:,",num2str(tmpInd),")")),1),1);
388     Profili_scelti_Nodi(Ind_int).TOT_Q =
389         Profili_scelti_Nodi(Ind_int).TOT;
390 end
391 end
392 Ind_Chieri = 38 : 55;
393
394 Load_Chieri_RES_riferimento = [ 0; 27; 56; 264; 168; 109; 142;
395     0; 99.5; 68; 52.5; 48; 304; 145.5; 339; 28.5; 103; 436];
396 Load_Chieri_3phase_riferimento = [0; 273.5; 201; 950.8; 585.5;
397     334.853; 258.5; 0; 261; 160.5; 442.5; 540.5; 576; 541.227;
398     612.253; 390.5; 446.5; 961];
399
400 for i = 1 : size(Load_Chieri_RES_riferimento,1)
401     P_RIF_Chieri(i,1) = (Load_Chieri_RES_riferimento(i,1) +
402         Load_Chieri_3phase_riferimento(i,1));
403 end
404
405 for Ind_Nodi_Chieri = 1: size(Ind_Chieri,2)
406
407     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
408         PinstallataNodo = P_RIF_Chieri((Ind_Nodi_Chieri));
409
410     if Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
411         Energia_res_pu > 0
412         Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).P_RES
413             = P_RIF_Chieri(Ind_Nodi_Chieri,1) * ((grid.nodes(

```

```

Ind_Chieri(Ind_Nodi_Chieri)).RES / 100));
403 else
404     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        P_RES = 0;
405 end
406
407 if Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        Energia_ind_pu > 0
408
409     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        P_IND = P_RIF_Chieri(Ind_Nodi_Chieri,1) * ((grid
            .nodes(Ind_Chieri(Ind_Nodi_Chieri)).IND / 100));
410
411 else
412
413     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        P_IND = 0;
414 end
415
416 %
417 if Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        Energia_comm_pu > 0
418
419     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)
        ).P_COMM = P_RIF_Chieri(Ind_Nodi_Chieri,1)
        * (((grid.nodes(Ind_Chieri(Ind_Nodi_Chieri))
            .COMM / 100)));
420 else
421
422     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        P_COMM = 0;
423 end
424
425 %
426 if Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).

```

```

Energia_uff_pu > 0

427
428
429         Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)
            ).P_UFF = P_RIF_Chieri(Ind_Nodi_Chieri,1) *
            (((grid.nodes(Ind_Chieri(Ind_Nodi_Chieri)).
                UFF / 100)));

430
431     else
432
433         Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
            P_UFF = 0;

434     end
435     %
436     Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
        Potenza_installata = Profili_scelti_Nodi(Ind_Chieri(
            Ind_Nodi_Chieri)).P_RES + Profili_scelti_Nodi(
            Ind_Chieri(Ind_Nodi_Chieri)).P_COMM +
            Profili_scelti_Nodi(Ind_Chieri(Ind_Nodi_Chieri)).
            P_IND + Profili_scelti_Nodi(Ind_Chieri(
            Ind_Nodi_Chieri)).P_UFF;

437
438     %
439     Ind_int = Ind_Chieri(Ind_Nodi_Chieri);
440     if (ismember(grid.nodes(Ind_int).name,
        Nodi_di_collegamento) == false)
441         Profili_scelti_Nodi(Ind_int).TOT =
            Profili_scelti_Nodi(Ind_int).P_RES .*
            Aggregato_TOT;
442         Profili_scelti_Nodi(Ind_int).TOT_Q = tan(acos(0.92)
            ) .* Aggregato_TOT;

443
444         for ind2=2:length(zoneType)
445
446             tmpInd = getfield(Indici.Chieri(Ind_Nodi_Chieri

```

```

),zoneType(ind2));
447 if not(tmpInd == 0)
448     Profili_scelti_Nodi(Ind_int).TOT =
        Profili_scelti_Nodi(Ind_int).TOT +
        getfield(Profili_scelti_Nodi(Ind_int),
        strcat("P_",zoneType(ind2))) .* eval(
        strcat("P_profili_",lower(zoneType(ind2)
        ),"_pu(:,",num2str(tmpInd),")"));
449     Profili_scelti_Nodi(Ind_int).TOT_Q =
        Profili_scelti_Nodi(Ind_int).TOT_Q +
        getfield(Profili_scelti_Nodi(Ind_int),
        strcat("P_",zoneType(ind2))) .* eval(
        strcat("Q_profili_",lower(zoneType(ind2)
        ),"_pu(:,",num2str(tmpInd),")"));
450
451     end
452 end
453 else
454     Profili_scelti_Nodi(Ind_int).TOT = zeros(size(
        eval(strcat("P_profili_",lower(zoneType(ind2)
        )), "_pu(:,",num2str(tmpInd),")"),1),1);
455     Profili_scelti_Nodi(Ind_int).TOT_Q =
        Profili_scelti_Nodi(Ind_int).TOT;
456 end
457 end

```

Codice A.8.4: Main_indici_rete_completo(codice completo)

A.9 Codice: Main_ProfilPV

Il Codice riportato in questa sezione permette di calcolare i profili in per unità a partire dai profili di irradianza ottenuti tramite il modello esposto in [22].

Le formule utilizzate sono le (5.2), (5.3) e (5.4). Sono stati inoltre utilizzati i valori di temperatura ottenuti tramite il Photovoltaic Geographical Information System (PVGIS) [32].

Essendo presenti dieci profili di irradianza diversi sono stati ottenuti un egual numero di profili di potenza in per unità. Di questi è stato utilizzato un solo profilo per l'inserimento del contributo legato al fotovoltaico all'interno della rete, riportato nel Codice A.9.2.

```
1 %% Estrazione temperatura da Excel
2 %
3 name = "Dati_";
4 type = "TORINO_2017_2017.xlsm";
5 str = strcat(name,type);
6 [status,sheets,xlFormat] = xlsfinfo(str);
7 sheetsDati = 2;
8 [Testo, Dati] = xlsread(str,sheetsDati);
9 T2m = {}; % Temperatura
10
11 for i = 1 : length(Dati)-1
12     T2m(i,1) = Dati(i+1,4);
13 end
14
15 T2m(strcmp(T2m(:,1), '#N/A')) = {'NaN'};
16 T2m = str2double(T2m(1:8760,1));
17
18 for i = 1 : 8760
19     T2m_1(:,i) = repmat(T2m(i,1),60,1);
20 end
21 T2m_min = reshape(T2m_1, [size(T2m_1,2)*size(T2m_1,1) 1]);
22
23 %% Profili PV per unit
24 load('house_panel_irradiance_ANNUALE.mat') % Profili irradianza
```

```

(W / m^2)
25 house_panel_irradiance_1min = house_panel_irradiance;
26 nnn = find(isnan(house_panel_irradiance_1min(:,:))); %
    Eliminazione NaN
27 house_panel_irradiance_1min(nnn) = 0;
28 numProfIrr = 10;
29 NOCT = 45; % Gradi centigradi
30 Tamb_min = T2m_min; % Valori di temperatura
31 Tc = ones(size(Tamb_min,1),10);
32 alpha_th = 0.45/100;
33 n_DC_AC = 0.828;
34
35 % Calcolo profili PV in p.u.
36 for i = 1 : length(Tamb_min)
37     for k = 1 : numProfIrr
38         Tc(i,k) = Tamb_min(i,1) + (((NOCT - 20)/800) .*
            house_panel_irradiance_1min(i, k));
39         nt_h(i,k) = 1 - alpha_th .* (Tc(i,k) - 25);
40         P_PVpu(i,k) = n_DC_AC .* (house_panel_irradiance_1min(i, k)
            /1000) .* nt_h(i,k); %Profili PV in p.u.
41     end
42 end

```

Codice A.9.1: Main_ProfilPV(codice completo)

A.9.1 Codice: Main_inserimento_PV

Questo codice inserisce, tramite un processo iterativo, il contributo di potenza di generazione fotovoltaica all'interno della rete test. Il procedimento è descritto nel Capitolo 5.

```
1  rng('default')
2
3  %%
4  % Set data
5      days = 365; % numero di giorni che si vogliono esaminare (1
        giorno, 2 giorni, ecc....)
6  for i = 1 : days
7      data_User = datetime(2017,1,i); % Inserire data giorno (
        yyyy/m/d)
8      [indici_giorno_selezionato] = Indici_giorno_selezionato(
        data_User);
9      I(i,:) = indici_giorno_selezionato;
10     indici_giorno_selezionato_FIN = reshape(I,1,[]); % convert
        matrix to row vector
11 end
12 indici_giorno_selezionato = indici_giorno_selezionato_FIN;
13 tau = 15/60;
14 h = 24;
15 T_max = h*days;
16 Tempo = tau : tau : T_max;
17 %%
18 % Profili di carico
19 load('PowerMatrix_P.mat')
20 load('PowerMatrix_Q.mat')
21
22 if not(size(PowerMatrix_P,1) == size(PowerMatrix_Q,1) && size(
        PowerMatrix_P,2) == size(PowerMatrix_Q,2))
23     msgbox('Problema con i profili di carico')
24     return % non ha senso andare avanti
```

```

25 end
26
27 %
28 PowerMatrix_P = PowerMatrix_P(indici_giorno_selezionato(1) :
    indici_giorno_selezionato(end), 2:end);
29 PowerMatrix_Q = PowerMatrix_Q(indici_giorno_selezionato(1) :
    indici_giorno_selezionato(end), 2:end);
30
31
32 %%
33 % Fotovoltaico
34 load('ProfiliPV_pu.mat')
35 Profili_PV_15min = ones(size(PowerMatrix_P));
36 ProfiliPV_pu = ProfiliPV_pu(indici_giorno_selezionato(1) :
    indici_giorno_selezionato(end), :);
37 num_profilo_PV = 9; % Indice profilo PV in p.u. scelto
38 %%
39
40 Pot_installata_PV = zeros(1, size(PowerMatrix_P,2)); % Potenza
    installata Fotovoltaico
41 Profili_PV = Pot_installata_PV .* Profili_PV_15min; % Profili
    di potenza Fotovoltaica p.u.
42 Profili_PV_pu = ProfiliPV_pu;
43 %[PowerMatrix_P, PowerMatrix_Q, Profili_PV, Profili_PV_pu] =
    function_passaggio_profili_orari(PowerMatrix_P,
    PowerMatrix_Q, Profili_PV, Profili_PV_pu);
44 ProfiliPV_pu = Profili_PV_pu;
45 % Valuto perdite allocate
46 [Perdite, V0, Ibase, V_n_tot, Ib_abs, Ib_mat, Is_fin,
    Perdite_allocate, Perdite_allocate_tot, Scar_pu,
    livello_carico_tot] = function_backward_forward_sweep(
    PowerMatrix_P, PowerMatrix_Q, Profili_PV);
47 %[Perdite_allocate] = function_backward_forward_sweep(
    PowerMatrix_P, PowerMatrix_Q, Profili_PV);
48 %Perdite_allocate = Perdite_allocate';

```

```

49
50 flag_Perdite_allocate = 0; % Inizializzazione flag sulle
    Perdite Allocate
51 Stato_Perdite = find(Perdite_allocate(:, :) < 0);
52
53 %% Salvataggio grandezze importanti
54 % caso1
55 V_n_tot_noPV = V_n_tot;
56 Ib_mat_noPV = transpose(Ib_mat);
57 Ib_mat_noPV_abs = abs(Ib_mat_noPV);
58 Ib_mat_noPV_abs_A = Ib_mat_noPV_abs .* Ibase; % A
59 Perdite_noPV = transpose(Perdite);
60 livello_carico_tot_noPV = livello_carico_tot;
61 %%
62 % caso1
63 diary on
64 % Valutazione indici di rete prima dell'inserimento del
    fotovoltaico
65 disp('Rete passiva');
66 %
67 Sbase = 1000; %kVA
68 tau = 15/60;
69 % Totale
70 disp('Totale')
71 I = transpose(Ib_mat);
72 I_1 = sum(I(:, 1:3), 2);
73 Srete = V0*ones(size(PowerMatrix_P, 1), 1) .* conj(I_1);
74 Sslack_real = real(Srete);
75 Sslack_imm = imag(Srete);
76 tan_fi = Sslack_imm ./ Sslack_real;
77 cosfi_medio = cos(atan(mean(tan_fi)));
78
79 cosfi_medio_1 = cosfi_medio;
80 disp('cosfi medio Totale: '); disp(cosfi_medio_1);
81 %

```

```

82 Perdite_noPV_tot_kw = sum(sum(Perdite_noPV)) .* Sbase; % kW
83 Perdite_noPV_tot = (tau * Perdite_noPV_tot_kw) / (10^3);
84 disp('Perdite_totali_NO_PV_[MWh]:'); disp(Perdite_noPV_tot);
85 %
86 Perdite_noPV_tot_pu = sum(sum(Perdite_noPV));
87 disp('Perdite_totali_NO_PV_[pu]:'); disp(Perdite_noPV_tot_pu);
88 %
89 disp('Energia_persa_Totale')
90 Perdite_totali = Perdite(:,:);
91 Scar_pu_totali = Scar_pu(:,:);
92 aa_totali = sum(sum(Perdite_totali));
93 bb_totali = sum(sum(real(Scar_pu_totali)));
94 cC_totali = (aa_totali/bb_totali) * 100;
95
96 cC_totali_1 = cC_totali;
97 disp(chop(cC_totali_1,2));
98 %
99 Ib_mat_trans = transpose(Ib_mat);
100 Ib_mat_slack = Ib_mat_trans(:,1:3);
101 Sslack_totali = V0*ones(size(PowerMatrix_P,1),3) .* conj(
    Ib_mat_slack);
102 bB_totali = -(aa_totali / sum(sum(real(Sslack_totali),1)))*100;
103
104 bB_totali_1 = bB_totali;
105 disp(chop(bB_totali_1,2))
106 %
107 % RPF
108 tau = 15/60;
109 S_not_pu = V0*ones(size(PowerMatrix_P,1),3) .* conj((
    Ib_mat_slack)); %.* Sbase;
110 S_not_pu = S_not_pu .* Sbase;
111 P_not_pu = real(S_not_pu);
112 E_HV_MV = tau .* P_not_pu;
113 %E_HV_MV = tau .* (V0*ones(size(PowerMatrix_P,1),3) .* (
    Ib_mat_slack));

```

```

114 e_HV_MV = real(E_HV_MV);
115 ind_x = find(e_HV_MV(:, :) > 0);
116
117 RPF = sum(e_HV_MV(ind_x)); %kWh
118 RPF_MWh = RPF / (10^3); %MWh
119
120 RPF_MWh_1 = RPF_MWh;
121 disp('RPF_MWh: '); disp(RPF_MWh_1);
122 %
123 %
124 RPF_pu = RPF / (tau * Sbase);
125
126 RPF_pu_1 = RPF_pu;
127 disp('RPF_p.u.: '); disp(RPF_pu_1);
128
129
130 % RPF TIME
131 Ib_slack_tot = sum(Ib_mat_slack, 2);
132 time_step = 15; % 15 min %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
133 RPF_min = size(find(E_HV_MV(:, :) > 0), 1) * time_step;
134 RPF_min_tot_1 = RPF_min;
135 disp('RPF_min: '); disp(RPF_min_tot_1);
136
137 % GRID SELF SUFFICIENCY
138
139 E_tot_RES = tau * sum(sum(Profili_PV));
140 E_loads = tau * sum(sum(PowerMatrix_P));
141
142 Ss = ((E_tot_RES - (RPF)) / E_loads) * 100;
143
144 Ss_1 = Ss;
145 disp('Ss: '); disp(Ss_1);
146
147 % GRID SELF CONSUMPTION
148

```

```

149 Sc = ((E_tot_RES - (RPF)) / (E_tot_RES)) * 100;
150
151 Sc_1 = Sc;
152 disp('Sc:'); disp(Sc_1);
153
154 diary off
155 %%
156 [aa,bb]=ind2sub(size(Stato_Perdite),Stato_Perdite);
157 cc=unique(bb);% Per verifica di quali siano i nodi con perdine
    negative all'inizio
158
159 %% Fine aggiunta
160 if (isempty(Stato_Perdite) == true)
161     flag_Perdite_allocate = 1; % Le perdite allocate sono tutte
        > 0
162 else
163     flag_Perdite_allocate = - 1; % Ci sono punti < 0
164 end
165
166 %% Strutture utilizzate dentro il ciclo while
167
168 Potenza_installata_PV_MAX = 1000; %kW
169
170 posizione_minimo = -1;
171 posizione_non_disp = [1 2 3];
172 pos = [];
173
174 nodo_di_partenza = -1;
175 %posizione_non_disp = [1 2 3];
176 nodo = [];
177 %%
178 cont = [];
179 cont(1,1) = 1;
180 cont(2,1) = 0;
181 %% Cicli per aggiunta PV

```



```

182
183 % SET PER INSERIMENTO PV
184 iterazioni = 1;
185 iterazioni_MAX = 10000; % Max numero di iterazioni accettabile
186 rapporto_PV_carichi = 0.8;
187 numero_nodi = size(PowerMatrix_P,2);
188 step = 1;
189 step_1 = 1;
190 B = 10:10:1000;
191 %
192 % Inserimento PV
193 while (sum(sum(Profili_PV))/sum(sum(PowerMatrix_P)) <
    rapporto_PV_carichi) % Rapporto (Energia_PV) / (
    Energia_carichi)
194     %if (flag_Perdite_allocate == 1)
195     %     keyboard
196     %end
197     while (flag_Perdite_allocate == 1)
198         Perdite_allocate_medie = mean(Perdite_allocate);
199
200         Perdite_allocate_medie(1, 1:3) = 10e6;
201         Perdite_allocate_medie(1, posizione_non_disp) = 10e6;
202         [valore_minimo, posizione_min] = min(
            Perdite_allocate_medie);
203         valore_minimo(end+1) = valore_minimo;
204         pos(end+1) = posizione_min;
205
206         if (not(posizione_minimo == posizione_min))
207             posizione_minimo = posizione_min;
208             Pot_installata_PV(1,posizione_minimo) =
                Pot_installata_PV(1,posizione_minimo) + 10;
209         else
210             Pot_installata_PV(1,posizione_minimo) =
                Pot_installata_PV(1,posizione_minimo) + 10;
211         end

```

```

212
213     if (Pot_installata_PV(1,posizione_minimo) >=
214         Potenza_installata_PV_MAX)
215
216         posizione_non_disp(end+1) = posizione_minimo;
217         Pot_installata_PV(1,posizione_minimo) =
218             Potenza_installata_PV_MAX;
219     end
220
221     if (length(posizione_non_disp) == length(
222         Perdite_allocate_medie))
223         break
224     end
225
226     for i = 1 : 54
227         %num_profilo_PV = qw;%randi([1 10],i,1);
228
229         Profili_PV(:,i) = Pot_installata_PV(1,i) .*
230             ProfiliPV_pu(:,num_profilo_PV(1)); % Profili di
231             potenza Fotovoltaica p.u.
232     end
233
234     [Perdite, V0, Ibase, V_n_tot, Ib_abs, Ib_mat, Is_fin,
235         Perdite_allocate, Perdite_allocate_tot, Scar_pu,
236         livello_carico_tot] =
237         function_backward_forward_sweep(PowerMatrix_P,
238             PowerMatrix_Q, Profili_PV);
239
240     Perdite_Allocate_primo = Perdite_allocate;
241
242     livello_carico_tot_primo = livello_carico_tot; %%%!
243     Vn_tot_primo = V_n_tot; %%%!
244
245     if (not(isempty(a) == true))
246         flag_Perdite_allocate = -1;
247     end
248
249

```

```

238         step_1 = step_1+1;
239     end
240
241
242     while (flag_Perdite_allocate == -1)
243         Perdite_allocate_medie = mean(Perdite_allocate);
244
245         %         Perdite_allocate_medie(1, 1:3) = 10e6;
246         Perdite_allocate_medie(1, posizione_non_disp) = 10e6;
247
248         Potenza_media = mean(PowerMatrix_P);
249         Potenza_media(1,posizione_non_disp) = 10e6;
250
251         for kk = 1 : length(Potenza_media)
252             if (Potenza_media(kk) == 0)
253                 funz_sign(1,kk) = sign(Perdite_allocate_medie(
254                     kk));
255             else
256                 funz_sign(1,kk) = sign(Potenza_media(kk)) *
257                     sign(Perdite_allocate_medie(kk));
258             end
259         end
260
261         if (isempty(find(funz_sign(1,:) < 0, 1)))
262             flag_Perdite_allocate = 1;
263             break
264         end
265
266         funz_sign(2,:) = abs(Perdite_allocate_medie);
267         index = find(funz_sign(1,:) < 0);
268         [valore, posizione] = max(funz_sign(2,index));
269         valore(end+1) = valore;
270         pos(end+1) = posizione;
271         nodo(end+1) = index(posizione);

```

```

271
272     if (not(nodo_di_partenza == nodo(end)))
273         nodo_di_partenza = nodo(end);
274         Pot_installata_PV(:,nodo_di_partenza) =
                Pot_installata_PV(:,nodo_di_partenza) + 10;
275     else
276         Pot_installata_PV(:,nodo_di_partenza) =
                Pot_installata_PV(:,nodo_di_partenza) + 10;
277     end
278
279     if (Pot_installata_PV(:,nodo_di_partenza) >=
        Potenza_installata_PV_MAX)
280
281         posizione_non_disp(end+1) = nodo_di_partenza;
282         Pot_installata_PV(:,nodo_di_partenza) =
                Potenza_installata_PV_MAX;
283     end
284
285     if (length(posizione_non_disp) == length(
        Perdite_allocate_medie))
286         break
287     end
288
289     for i = 1 : numero_nodi
290         Profili_PV(:,i) = Pot_installata_PV(1,i) .*
                ProfiliPV_pu(:,num_profilo_PV(1)); % Profili di
                potenza Fotovoltaica p.u.
291     end
292
293     Perdite_Allocate_old = Perdite_allocate;
294     [Perdite, V0, Ibase, V_n_tot, Ib_abs, Ib_mat, Is_fin,
        Perdite_allocate, Perdite_allocate_tot, Scar_pu,
        livello_carico_tot] =
        function_backward_forward_sweep(PowerMatrix_P,
        PowerMatrix_Q, Profili_PV);

```

```

295
296         if (isempty(find(funz_sign(1,:) < 0, 1)))
297             flag_Perdite_allocate = 1;
298         end
299         step = step + 1;
300     end
301
302
303     iterazioni = iterazioni + 1;
304     cont(1,iterazioni) = iterazioni;
305     cont(2,iterazioni) = sum(sum(Profili_PV))/sum(sum(
        PowerMatrix_P));
306     disp(iterazioni)
307     disp(sum(sum(Profili_PV))/sum(sum(PowerMatrix_P)))
308     if (iterazioni > iterazioni_MAX)
309         disp('Iterazioni massime raggiunte')
310         break
311     end
312 end
313
314
315 %% Salvataggio grandezze importanti
316 % caso2
317 V_n_tot_PV = V_n_tot;
318 Ib_mat_PV = transpose(Ib_mat);
319 Ib_mat_PV_abs = abs(Ib_mat_PV);
320 Ib_mat_PV_abs_A = Ib_mat_PV_abs .* Ibase; % A
321 Perdite_PV = transpose(Perdite);
322 livello_carico_tot_PV = livello_carico_tot;
323 %%
324 % caso2
325 diary on
326 % Valutazione indici di rete dopo l'inserimento del
    fotovoltaico
327 disp('Rete con presenza di fotovoltaico');

```

```

328 %
329 Sbase = 1000; %kVA
330
331 % Totale
332 disp('Totale')
333 I = transpose(Ib_mat);
334 I_1 = sum(I(:,1:3),2);
335 Srete = V0*ones(size(PowerMatrix_P,1),1) .* conj(I_1);
336 Sslack_real = real(Srete);
337 Sslack_imm = imag(Srete);
338 tan_fi = Sslack_imm ./ Sslack_real;
339 cosfi_medio = cos(atan(mean(tan_fi)));
340 %
341 cosfi_medio_2 = cosfi_medio;
342 disp('cosfi_medio_Totale:'); disp(cosfi_medio_2);
343 %
344 Perdite_PV_tot_kw = sum(sum(Perdite_PV)) .* Sbase; % kW
345 Perdite_PV_tot = (tau * Perdite_PV_tot_kw) / (10^3);
346 disp('Perdite_totali_CON_PV[MWh]:'); disp(Perdite_PV_tot);
347 %
348 Perdite_PV_tot_pu = sum(sum(Perdite_PV));
349 disp('Perdite_totali_CON_PV[pu]:'); disp(Perdite_PV_tot_pu);
350 %
351 Rapp_Perdite_tot = Perdite_PV_tot / Perdite_noPV_tot;
352 disp('Rapporto_perdite(rete_con_PV/rete_senza_PV):'); disp(
    Rapp_Perdite_tot);
353 %
354 disp('Energia_persa_Totale')
355 Perdite_totali = Perdite(:,:);
356 Scar_pu_totali = Scar_pu(:,:);
357 aa_totali = sum(sum(Perdite_totali));
358 bb_totali = sum(sum(real(Scar_pu_totali)));
359 cC_totali = (aa_totali/bb_totali) * 100;
360
361 cC_totali_2 = cC_totali;

```

```

362 disp(chop(cC_totali_2,2));
363 %
364 Ib_mat_trans = transpose(Ib_mat);
365 Ib_mat_slack = Ib_mat_trans(:,1:3);
366 Sslack_totali = V0*ones(size(PowerMatrix_P,1),3) .* conj(
    Ib_mat_slack);
367 bB_totali = -(aa_totali / sum(sum(real(SSlack_totali),1)))*100;
368
369 bB_totali_2 = bB_totali;
370 disp(chop(bB_totali_2,2))
371 %
372 % RPF
373 tau = 15/60;
374 S_not_pu = V0*ones(size(PowerMatrix_P,1),3) .* conj((Ib_mat_PV
   (:,1:3))); %.* Sbase;
375 S_not_pu = S_not_pu .* Sbase;
376 P_not_pu = real(S_not_pu);
377 E_HV_MV = tau .* P_not_pu;
378 %E_HV_MV = tau .* (V0*ones(size(PowerMatrix_P,1),3) .* (
    Ib_mat_slack));
379 e_HV_MV = real(E_HV_MV);
380 ind_x = find(e_HV_MV(:, :) > 0);
381
382 RPF = sum(e_HV_MV(ind_x)); %kWh
383 RPF_MWh = RPF/(10^3); %MWh
384
385 RPF_MWh_2 = RPF_MWh;
386 disp('RPF_MWh:'); disp(RPF_MWh_2);
387 %
388 %
389 RPF_pu = RPF / (tau * Sbase);
390
391 RPF_pu_2 = RPF_pu;
392 disp('RPF_pu:'); disp(RPF_pu_2);
393

```

```

394
395 % RPF TIME
396 Ib_slack_tot = sum(Ib_mat_slack,2);
397
398 RPF_min = size(find(E_HV_MV(:, :) > 0), 1) * time_step;
399 RPF_min_tot_2 = RPF_min;
400 disp('RPF_min:'); disp(RPF_min_tot_2);
401
402 % GRID SELF SUFFICIENCY
403
404 E_tot_RES = tau * sum(sum(Profili_PV));
405 E_loads = tau * sum(sum(PowerMatrix_P));
406
407 Ss = ((E_tot_RES - (RPF)) / E_loads) * 100;
408
409 Ss_2 = Ss;
410 disp('Ss:'); disp(Ss_2);
411
412 % GRID SELF CONSUMPTION
413
414 Sc = ((E_tot_RES - (RPF)) / (E_tot_RES)) * 100;
415
416 Sc_2 = Sc;
417 disp('Sc:'); disp(Sc_2);
418
419 % Rapporto RPF (caso2 / caso1)
420 Rapp_RPF_MWh = RPF_MWh_2 / RPF_MWh_1;
421 disp('Rapporto_RPF_MWh:'); disp(Rapp_RPF_MWh);
422 %
423 Rapp_RPF_pu = RPF_pu_2 / RPF_pu_1;
424 disp('Rapporto_RPF_pu:'); disp(Rapp_RPF_pu);
425 %
426 Rapp_RPF_min = RPF_min_tot_2 / RPF_min_tot_1;
427 disp('Rapporto_RPF_min:'); disp(Rapp_RPF_min);
428

```



```

429 % Rapporto tensione (caso2 / caso1)
430 for i = 1:size(V_n_tot_noPV,1)
431     for k = 1:size(V_n_tot_noPV,2)
432         Rapp_V(i,k) = V_n_tot_PV(i,k) ./ V_n_tot_noPV(i,k);
433     end
434 end

```

Codice A.9.2: Main_inserimento_PV(codice completo)

All'interno del Codice riportato in [A.9.2](#) vengono richiamate diverse funzioni.

La funzione "*function_indici_giorno_selezionato*" serve solamente per generare degli indici in termini di quarti d'ora sui profili annuali. Essa risulta utile se si vuol analizzare una giornata specifica anziché l'intero anno. Tale funzione viene esposta nel Codice [A.9.3](#).

```

1  % Questa funzione fornisce degli indici giornalieri su una data
   scelta
2  % dall'utente. Gli indici finali sono il quarto d'ora iniziale
   e finale che
3  % possono essere utilizzati come indici nelle matrici annuali
   dei profili
4
5  function [indici_giorno_selezionato] =
   function_indici_giorno_selezionato(data_User)
6
7  tt = data_User;
8
9  giorni = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
10 ore_mese_min = ones(1, length(giorni));
11 h = 24;
12 min = 60;
13 quarto_d_ora = 15;
14
15 for i = 1 : length(giorni)
16     ore_mese_min(1,i) = (h*min*giorni(i)) / quarto_d_ora;
17 end

```

```

18
19 ore_mese_quarto_d_ora_cumulativo = ones(1,length(ore_mese_min))
    ;
20
21 for i = 1 : length(ore_mese_min)
22     if i == 1
23         ore_mese_quarto_d_ora_cumulativo(i) = ore_mese_min(1);
24     else
25         ore_mese_quarto_d_ora_cumulativo(i) = sum(ore_mese_min(1:i));
26     end
27 end
28
29 calendario = zeros(3,sum(giorni));
30 calendario(1,:) = 1 : 1: sum(giorni);
31 i = 1;
32 calendario(2, 1 : 31) = 1 : 1 : 31;
33 i = i + 1;
34 calendario(2, 31 + 1 : 59) = 1 : 1 : 28;
35 i = i + 1;
36 calendario(2, 59 + 1 : 90) = 1 : 1 : giorni(i);
37 i = i + 1;
38 calendario(2, 90 + 1 : 120) = 1 : 1 : giorni(i);
39 i = i + 1;
40 calendario(2, 120 + 1 : 151) = 1 : 1 : giorni(i);
41 i = i + 1;
42 calendario(2, 151 + 1 : 181) = 1 : 1 : giorni(i);
43 i = i + 1;
44 calendario(2, 181 + 1 : 212) = 1 : 1 : giorni(i);
45 i = i + 1;
46 calendario(2, 212 + 1 : 243) = 1 : 1 : giorni(i);
47 i = i + 1;
48 calendario(2, 243 + 1 : 273) = 1 : 1 : giorni(i);
49 i = i + 1;
50 calendario(2, 273 + 1 : 304) = 1 : 1 : giorni(i);
51 i = i + 1;

```

```

52 calendario(2, 304 + 1 : 334) = 1 : 1 : giorni(i);
53 i = i + 1;
54 calendario(2, 334 + 1 : 365) = 1 : 1 : giorni(i);
55
56 for k = 1 : 365
57     if k == 1
58         calendario(3,k) = 96;
59     else
60         calendario(3,k) = calendario(3,k-1)+96;
61     end
62 end
63
64 t1 = datetime(2017,1,1);
65 t2 = datetime(2017,12,31);
66 t = t1:caldays(1):t2;
67
68 index = (find(t(:)==tt));
69
70 indici_giorno_selezionato = (calendario(3,index) - 95) :
    calendario(3,index);
71
72 end

```

Codice A.9.3: Function_indici_giorno_selezionato(codice completo)

La funzione "*function_backward_forward_sweep*" esegue il BFS su tutta la rete test [4.1](#) e ne calcola gli andamenti di tensione, correnti, livello di carico e perdite allocate, il codice di questa funzione viene riportato nella sottosezione [A.9.2](#).

A.9.2 Codice: function_backward_forward_sweep

```
1 % Questa funzione calcola il BFS della rete completa, in uscita
   vengono
2 % esportate tensioni, correnti, perdite della rete. In ingresso
   sono
3 % presenti le matrici di potenza attiva, reattiva e la matrice
   dei profili
4 % fotovoltaici.
5
6 function [Perdite, V0, Ibase, V_n_tot, Ib_abs, Ib_mat, Is_fin,
   Perdite_allocate, Perdite_allocate_tot, Scar_pu,
   livello_carico_tot] = function_backward_forward_sweep(
   PowerMatrix_P, PowerMatrix_Q, Profili_PV)
7 %%
8 % Tempo per il BFS
9 Ntime_steps = size(PowerMatrix_P,1); % 35040 salvo diverse
   indicazioni
10 time_step = 1 : 1 : Ntime_steps;
11 %
12 % Parametri linea
13 run init_distribution_network_v1.m
14 %
15 % Grandezze base
16 % 1)
17 V1 = 220000; % V primario
18 V2 = 22000; % V secondario
19 t_trasf = V1/V2; % Rapporto di trasformazione trafo rete
   distribuzione
20 %
21 Vbase = V2;
22 Sbase = 1000; %kVA
23 %
24 Ibase = (Sbase*10^3)/(sqrt(3)*Vbase);
25 Zbase = Vbase/ (sqrt(3) * Ibase);
```

```

26 %%
27 grid.BFSpars.tolerance = 1e-5; % Tolleranza BFS
28 branchNames = {"PIS_Param_Magra","PIS_Param_Brenta","
    PIS_Param_Grosso","PIS_Param_Fiat","PIS_Param_Chieri"}; %#ok
    <CLARRSTR>
29 noPowNodes = [2052801, 205271, 204659, 208265, 2052802, 204946,
    203333, 203892, 204871];
30
31 grid.nodes(1).name = 1;
32 grid.nodes(1).isPow = false;
33 grid.nodes(1).inBranch = "Root";
34 grid.nodes(2).name = 1551411;
35 grid.nodes(2).isPow = false;
36 grid.nodes(2).inBranch = "Trasf";
37 grid.nodes(1).inLines = 1;
38 grid.nodes(2).inLines = 1;
39 grid.lines(1).inNode = 1;
40 grid.lines(1).outNode = 1551411;
41 grid.Imax_A_Magra = MVA_1/(sqrt(3) * Vbase); % Corrente
    nominale trafo 63 MVA
42 grid.Imax_A_Brenta = grid.Imax_A_Magra; % Dorsali collegate
    allo stesso trafo
43
44 grid.nodes(3).name = 1551412;
45 grid.nodes(3).isPow = false;
46 grid.nodes(3).inBranch = "Trasf";
47 grid.nodes(1).inLines = [grid.nodes(1).inLines, 2];
48 grid.nodes(3).inLines = 2;
49 grid.lines(2).inNode = 1;
50 grid.lines(2).outNode = 1551412;
51 grid.Imax_A_Grosso = MVA_2/(sqrt(3) * Vbase); % Corrente
    nominale trafo 55 MVA
52 grid.Imax_A_Fiat = grid.Imax_A_Grosso; % Dorsali collegate allo
    stesso trafo
53

```

```

54 grid.nodes(4).name = 1551413;
55 grid.nodes(4).isPow = false;
56 grid.nodes(4).inBranch = "Trasf";
57 grid.nodes(1).inLines = [grid.nodes(1).inLines, 3];
58 grid.nodes(4).inLines = 3;
59 grid.lines(3).inNode = 1;
60 grid.lines(3).outNode = 1551413;
61 grid.Imax_A_Chieri = MVA_1/(sqrt(3) * Vbase); % Corrente
    nominale trafo 63 MVA
62
63 for brInd = 1 : length(branchNames)
64     curBranch = eval(branchNames{brInd});
65     for ind = 1 : size(curBranch,1)
66         if (brInd == 1)
67             grid.Imax_A_Magra(end+1) = curBranch(ind,10);
68         end
69         if (brInd == 2)
70             grid.Imax_A_Brenta(end+1) = curBranch(ind,10);
71         end
72         if (brInd == 3)
73             grid.Imax_A_Grosso(end+1) = curBranch(ind,10);
74         end
75         if (brInd == 4)
76             grid.Imax_A_Fiat(end+1) = curBranch(ind,10);
77         end
78         if (brInd == 5)
79             grid.Imax_A_Chieri(end+1) = curBranch(ind,10);
80         end
81     end
82 end
83
84 for brInd=1:length(branchNames)
85     curBranch = eval(branchNames{brInd});
86     for ind=1:size(curBranch,1)

```

```

87         foundNode = [find([grid.nodes(:).name] == curBranch(ind
88             ,1))];
89         if not isempty(foundNode)
90             grid.nodes(foundNode).inLines = [grid.nodes(
91                 foundNode).inLines, length(grid.lines)+1];
92             grid.lines(end+1).inNode = curBranch(ind,1);
93         else
94             grid.nodes(end+1).name = curBranch(ind,1);
95             if(find(noPowNodes == grid.nodes(end).name))
96                 grid.nodes(end).isPow = false;
97             else
98                 grid.nodes(end).isPow = true;
99             end
100             grid.nodes(end).inBranch = extractAfter(
101                 branchNames{brInd},10);
102             grid.nodes(end).inLines = length(grid.lines)+1;
103             grid.lines(end+1).inNode = curBranch(ind,1);
104         end
105         grid.nodes(end+1).name = curBranch(ind,2);
106         if(find(noPowNodes == grid.nodes(end).name))
107             grid.nodes(end).isPow = false;
108         else
109             grid.nodes(end).isPow = true;
110         end
111         grid.nodes(end).inBranch = extractAfter(branchNames{
112             brInd},10);
113         grid.nodes(end).inLines = length(grid.lines);
114         grid.lines(end).outNode = curBranch(ind,2);
115     end
116 end
117
118 grid.matrL = zeros(length(grid.lines),length(grid.nodes));
119 for ind=1:length(grid.lines)
120     foundNode = find([grid.nodes(:).name] == grid.lines(ind).
121         inNode);

```

```

117     grid.matrL(ind,foundNode) = 1;
118     foundNode = find([grid.nodes(:).name] == grid.lines(ind).
        outNode);
119     grid.matrL(ind,foundNode) = -1;
120 end
121
122 grid.matrL = grid.matrL(:,2:end);
123 radial_check = det(grid.matrL);
124 L = grid.matrL;
125
126 % Verifica se la matrice L risulta invertibile
127 if det(L)== 0
128     msgbox('La matrice L non risulta invertibile!');
129     return % non ha senso andare avanti
130 end
131
132 grid.matrGamma = inv(grid.matrL);
133 Gamma = inv(L);
134 incidenze = L;
135 %%
136
137 Param_line_ohm = [];
138 for brInd=1:length(branchNames)
139     curBranch = eval(branchNames{brInd});
140     for ind=1:size(curBranch,1)
141         Param_line_ohm (end+1,1) = curBranch(ind,1);
142         Param_line_ohm (end,2) = curBranch(ind,2);
143         Param_line_ohm (end,3) = (curBranch(ind,3) * curBranch
            (ind,9)); %(2/3) * (curBranch(ind,3) * curBranch(
                ind,9)); % Rs
144         Param_line_ohm (end,4) = (curBranch(ind,5) * curBranch
            (ind,9)); %(2/3)* (curBranch(ind,5) * curBranch(ind
                ,9)); % Ls
145         Param_line_ohm (end,5) = (curBranch(ind,7) * curBranch
            (ind,9)); % Cs

```



```

146
147     end
148 end
149
150 Parametri_linea_ohm = {};
151 Parametri_linea_ohm.Nome = [];
152 Parametri_linea_ohm.Start = [];
153 Parametri_linea_ohm.Stop = [];
154 Parametri_linea_ohm.Rs = [];
155 Parametri_linea_ohm.Ls = [];
156 Parametri_linea_ohm.Zs = [];
157 Parametri_linea_ohm.Cs = [];
158 Parametri_linea_ohm.Ysh = [];
159
160 for i = 1 : size([grid.lines.inNode],2)
161     if (i <= 3)
162         Parametri_linea_ohm(i).Nome = {'Trasf'};
163     elseif ( i >3 && i<=7)
164         Parametri_linea_ohm(i).Nome = {'Magra'};
165     elseif ( i >7 && i<=17)
166         Parametri_linea_ohm(i).Nome = {'Brenta'};
167     elseif ( i >17 && i<=28)
168         Parametri_linea_ohm(i).Nome = {'Grosso'};
169     elseif ( i >28 && i<=36)
170         Parametri_linea_ohm(i).Nome = {'Fiat'};
171     elseif ( i >36)
172         Parametri_linea_ohm(i).Nome = {'Chieri'};
173     end
174 end % ciclo per nomi linee
175
176 c = 1;
177 for i = 1 : size([grid.lines.inNode],2)
178     Parametri_linea_ohm(i).Start = grid.lines(i).inNode;
179     Parametri_linea_ohm(i).Stop = grid.lines(i).outNode;
180     if (i<4)

```

```

181     if (Parametri_linea_ohm(i).Stop == 1551411 ||
Parametri_linea_ohm(i).Stop == 1551413)
182         Parametri_linea_ohm(i).Rs = R_1 * (22000^2/63e6);
183         Parametri_linea_ohm(i).Ls = X_1 * (22000^2/63e6);
184         Parametri_linea_ohm(i).Zs = Parametri_linea_ohm(i).Rs +
            (1j * Parametri_linea_ohm(i).Ls);
185         Parametri_linea_ohm(i).Cs = 0;
186         Parametri_linea_ohm(i).Ysh = (Gm_1 - 1j * Bm_1) * (1/(
            Vbase^2/MVA_1));
187
188     elseif (Parametri_linea_ohm(i).Stop == 1551412)
189         Parametri_linea_ohm(i).Rs = R_2 * (22000^2/55e6);
190         Parametri_linea_ohm(i).Ls = X_2 * (22000^2/55e6);
191         Parametri_linea_ohm(i).Zs = Parametri_linea_ohm(i).Rs +
            (1j * Parametri_linea_ohm(i).Ls);
192         Parametri_linea_ohm(i).Cs = 0;
193         Parametri_linea_ohm(i).Ysh = (Gm_2 - 1j * Bm_2) * (1/(
            Vbase^2/MVA_2));
194     end
195     else
196         Parametri_linea_ohm(i).Rs = Param_line_ohm(c,3);
197         Parametri_linea_ohm(i).Ls = Param_line_ohm(c,4);
198         Parametri_linea_ohm(i).Zs = Parametri_linea_ohm(i).Rs
            + (1j * w * Parametri_linea_ohm(i).Ls);
199         Parametri_linea_ohm(i).Cs = Param_line_ohm(c,5);
200         Parametri_linea_ohm(i).Ysh = 1 /(-1j / (w *
            Parametri_linea_ohm(i).Cs));
201
202         c = c+1;
203     end
204 end
205
206
207 Zb_pu = zeros(size([grid.lines.inNode],2));
208 for i = 1 : size([grid.lines.inNode],2)

```

```

209     Zbranch_pu(i,i) = (Parametri_linea_ohm(i).Zs / Zbase);
210     Ysh_pu(i,1) = (Parametri_linea_ohm(i).Ysh / (1/Zbase));
211 end
212 check = det(Zb_pu);
213
214 % Dati rami
215 Zbranch = Zbranch_pu;
216
217 % Ammettenza trasversale totale linee
218 Ysh = Ysh_pu;
219
220 % Aggregazione per nodi dei parametri trasversali delle linee
221 Ytotnodi = abs(transpose(incidenze))* Ysh/2;
222
223 % Numero nodi rete
224 NumNodi = 54;
225
226 for IndNodi = 1 : NumNodi % ciclo sui carichi (NODO = GRANDEZZE
    NODALI!)
227     for time = 1 : length(time_step)
228         Ptemp(time, IndNodi) = PowerMatrix_P(time, IndNodi) -
            Profili_PV(time,IndNodi); %potenza attiva al nodo k nel
            tempo
229         Qtemp(time, IndNodi) = PowerMatrix_Q(time, IndNodi);
230         Scar_pu(time, IndNodi) = (Ptemp(time, IndNodi) + 1i * Qtemp(
            time, IndNodi)) / Sbase; %potenza apparente in [p.u.]
231     end
232 end
233
234 Scar_pu = transpose(Scar_pu);
235 %%
236 % Ciclo BFS
237 i = 1; %indice di calcolo
238 while (i < Ntime_steps + 1)
239     % Inizializzazione matrici per output tensioni e correnti

```

```

240 V = zeros(NumNodi, Ntime_steps); % Tensione ai nodi in [p.u
    .]
241 Is = zeros(NumNodi, Ntime_steps); % Corrente ai carichi [p.
    u.]
242 Ib = zeros(NumNodi, Ntime_steps); % Correnti di linea [p.u
    .]
243
244 errore = 1; %errore iniziale (per entrare nel while)
245 max_num_iterazioni = 300; %numero massimo di iterazioni per
    evitare che resti intrappolato nel while
246
247 % TOLLERANZA RICHIESTA
248 tolleranza = 1e-5;
249
250 DV = zeros(Ntime_steps, 1); %funzione obiettivo
251 Vr = 1;
252 cont = 0; %contatore ausiliario
253
254 %Tensione nodo slack
255 V0 = 1; % Tensione di slack in p.u.
256 V0vett = V0 * ones(NumNodi, 1); % Profilo piatto di
    tensione
257 V = V0vett; % Tensioni nodali, stato di start: Profilo
    piatto
258
259 while (errore > tolleranza)
260     Is_carico = conj(Scar_pu(:, i) ./ V);
261     Is_h = V .* Ytotnodi;%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
262     Is = Is_carico + Is_h; %corrente totale
263     Ib = transpose(Gamma) * Is; %corrente di linea
264
265 % Fase di forward : calcolo tensioni nodali calcolando le
    cadute

```

```

266     V_n = V0vett - Gamma * Zbranch * Ib; % Cadute di
        tensione
267     errore = max(abs(V_n - V) ./ abs(V_n)); %calcolo errore
268     V = V_n; %aggiornamento tensione
269     cont = cont +1;
270     if cont >= max_num_iterazioni %superamento numero
        massimo di iterazioni consentite
271         keyboard
272         msgbox('Nessuna convergenza entro il numero di
            iterazioni previste!');
273         return %non ha senso andare avanti
274     end
275 end
276 %
277 Perdite(:, i) = diag(real(Zbranch)).*abs(Ib).^2; %perdite
        joule nei rami
278 V_n_tot(i,:) = abs(V); % tensioni nodali
279 Ib_abs(:, i) = abs(Ib); % valore assoluto dellecorrenti nei
        rami
280 Ib_mat(:,i) = Ib; % correnti nei rami
281 Is_fin(:,i) = Is;
282 Perdite_allocate(:, i)= real(conj(Is) .* (Gamma * (diag(
        real(Zbranch)) .* Ib)));
283 Perdite_allocate_tot(i) = sum(Perdite_allocate(:, i));
284
285 DV(i) = max(abs(Vr-max(V)),abs(Vr-min(V)));
286 i = i+1;
287 end
288 Scar_pu = transpose(Scar_pu);
289 end
290 Perdite_allocate = Perdite_allocate';
291
292 % Indici reti
293 IndMagra = [1, 4 : 7];
294 IndBrenta = [1, 8 : 17];

```

```

295 IndGrosso = [2, 18 : 28];
296 IndFiat = [2, 29 : 36];
297 IndChieri = [3, 37 : 54];
298 %
299 Ib_abs_Magra = [Ib_abs(IndMagra,:)];
300 Ib_mat_Magra = [Ib_mat(IndMagra,:)];
301
302 Ib_abs_Brenta = [Ib_abs(IndBrenta,:)];
303 Ib_mat_Brenta = [Ib_mat(IndBrenta,:)];
304
305 Ib_abs_Grosso = [Ib_abs(IndGrosso,:)];
306 Ib_mat_Grosso = [Ib_mat(IndGrosso,:)];
307
308 Ib_abs_Fiat = [Ib_abs(IndFiat,:)];
309 Ib_mat_Fiat = [Ib_mat(IndFiat,:)];
310
311 Ib_abs_Chieri = [Ib_abs(IndChieri,:)];
312 Ib_mat_Chieri = [Ib_mat(IndChieri,:)];
313 %
314 % Livello di carico:
315 Imax_Magra = transpose(grid.Imax_A_Magra) / Ibase; % corrente
           max in pu
316 Imax_Brenta = transpose(grid.Imax_A_Brenta) / Ibase; % corrente
           max in pu
317 Imax_Grosso = transpose(grid.Imax_A_Grosso) / Ibase; % corrente
           max in pu
318 Imax_Fiat = transpose(grid.Imax_A_Fiat) / Ibase; % corrente max
           in pu
319 Imax_Chieri = transpose(grid.Imax_A_Chieri) / Ibase; % corrente
           max in pu
320 %%
321 % Magra
322 V_n_tot_Magra = [V_n_tot(:,IndMagra)];
323 livello_carico_Magra = Ib_abs_Magra ./ repmat(Imax_Magra,1,size
           (PowerMatrix_P,1));

```

```

324 livello_carico_Magra = livello_carico_Magra';
325 % Brenta
326 V_n_tot_Brenta = [V_n_tot(:,IndBrenta)];
327 livello_carico_Brenta = Ib_abs_Brenta ./ repmat(Imax_Brenta,1,
    size(PowerMatrix_P,1));
328 livello_carico_Brenta = livello_carico_Brenta';
329 % Grosso
330 V_n_tot_Grosso = [V_n_tot(:,IndGrosso)];
331 livello_carico_Grosso = Ib_abs_Grosso ./ repmat(Imax_Grosso,1,
    size(PowerMatrix_P,1));
332 livello_carico_Grosso = livello_carico_Grosso';
333 % Fiat
334 V_n_tot_Fiat = [V_n_tot(:,IndFiat)];
335 livello_carico_Fiat = Ib_abs_Fiat ./ repmat(Imax_Fiat,1,size(
    PowerMatrix_P,1));
336 livello_carico_Fiat = livello_carico_Fiat';
337 % Chieri
338 V_n_tot_Chieri = [V_n_tot(:,IndChieri)];
339 livello_carico_Chieri = Ib_abs_Chieri ./ repmat(Imax_Chieri,1,
    size(PowerMatrix_P,1));
340 livello_carico_Chieri = livello_carico_Chieri';
341
342 livello_carico_tot = [livello_carico_Magra(:,1),
    livello_carico_Grosso(:,1), livello_carico_Chieri(:,1),
    livello_carico_Magra(:,2:end), livello_carico_Brenta(:,2:end
    ), livello_carico_Grosso(:,2:end), livello_carico_Fiat(:,2:
    end), livello_carico_Chieri(:,2:end)];
343
344 end

```

Codice A.9.4: Function_backward_forward_sweep(codice completo)

A.10 Codice: Main_inserimento_P2G

Il codice A.10.1 inserisce il contributo del P2G all'interno della rete test 4.1 e calcola gli indici di rete.

Cambiando il comando alla riga 80, "*Num_Impianti*", in base a quanti impianti sono presenti e i nodi in cui sono installati, tramite il comando alla riga 81, "*pos_I*" è possibile simulare la presenza di un singolo P2G o di diversi impianti presenti nella rete.

```
1 Nodi = [ 1551411 1551412 1551413 204874 208221 203845 203844
          203294 203548 2052801 204514 204621 205428 205271 204659
          203615 208265 205358 2052802 205024 204730 204748 204813
          203975 203974 204716 203385 203558 204871 204228 204171
          204201 204216 204193 203202 203765 204946 203890 204834
          203319 203323 204249 203324 203333 203892 205351 203874
          203137 203923 203921 205304 204220 205638 203937];
2 Legenda_Nodi = Nodi';
3 Legenda_Nodi(:,2) = linspace(1,size(Legenda_Nodi,1),size(
    Legenda_Nodi,1));
4 %%
5 % Set data
6     days = 1; numero di giorni che si vogliono esaminare (1
        giorno, 2 giorni, ecc....)
7 for i = 1 : days
8     data_User = datetime(2017,1,i); % Inserire data giorno (
        yyyy/m/d)
9     [indici_giorno_selezionato] = Indici_giorno_selezionato(
        data_User);
10    I(i,:) = indici_giorno_selezionato;
11    indici_giorno_selezionato_FIN = reshape(I,1,[]); % convert
        matrix to row vector
12 end
13 indici_giorno_selezionato = indici_giorno_selezionato_FIN;
14 tau = 15/60;
15 h = 24;
```



```

16 T_max = h*days;
17 Tempo = tau : tau : T_max;
18 %%
19 % Parametri_Ptg
20 taglie_P2G = [3 3 3 3];
21 [Ptg,tan_phi,keepstep,MET_MIN_FRAC,H2_BUFFER_TIME,STOICH,
    YIELD_EL_H2,CO2_CONVERSION,mw_CH4,mw_H2,LHV_CH4,LHV_H2,
    YIELD_H2_SNG,n_H2_MET_MIN,H2_TANK_MAX,H2_TANK_in,TIME_1] =
    function_parametri_Ptg(taglie_P2G);
22 %%
23 % Profili di carico
24 load('PowerMatrix_P.mat')
25 load('PowerMatrix_Q.mat')
26
27 if not(size(PowerMatrix_P,1) == size(PowerMatrix_Q,1) && size(
    PowerMatrix_P,2) == size(PowerMatrix_Q,2))
28     msgbox('Problema con i profili di carico')
29     return % non ha senso andare avanti
30 end
31 %
32 PowerMatrix_P = PowerMatrix_P(indici_giorno_selezionato(1) :
    indici_giorno_selezionato(end), 2:end);
33 PowerMatrix_Q = PowerMatrix_Q(indici_giorno_selezionato(1) :
    indici_giorno_selezionato(end), 2:end);
34 %%
35 % Fotovoltaico
36 load('ProfiliPV_annuali.mat')
37 Profili_PV = ProfiliPV_annuali;
38 Profili_PV = Profili_PV(indici_giorno_selezionato(1) :
    indici_giorno_selezionato(end), :);
39
40 %%
41 Ntime_steps = size(PowerMatrix_P,1); % 35040 salvo diverse
    indicazioni
42 time_step = 1 : 1 : Ntime_steps;

```

```

43 %%
44 % BFS rete NO PTG
45 [Perdite, V0, Ibase, V_n_tot, Ib_abs, Ib_mat, Is_fin,
    Perdite_allocate, Perdite_allocate_tot, Scar_pu,
    livello_carico_tot] = function_backward_forward_sweep(
    PowerMatrix_P, PowerMatrix_Q, Profili_PV);
46 clc;
47 %
48 V_n_tot_PV = V_n_tot;
49 Ib_mat_PV = transpose(Ib_mat);
50 Ib_mat_PV_abs = abs(Ib_mat_PV);
51 Ib_mat_PV_abs_A = Ib_mat_PV_abs .* Ibase; % A
52 Perdite_PV = transpose(Perdite);
53 livello_carico_tot_PV = livello_carico_tot;
54 %
55 %% CONTROLLO P2G - REVERSE POWER FLOW
56 Out=[];
57
58 Ib_mat = Ib_mat';
59 revPF = real(V_n_tot .* conj(Ib_mat)); % Reverse Power Flow
60
61 % Numero impianti P2G e nodi d'installazione
62 Num_Impianti = 4; % Definire numero impianti
63 pos_1 = [4 8 19 37];
64 nodi_installazione = [];
65 revPF_1 = revPF;
66 cC_totali_2fin = [];
67 RPF_MWh_2 = [];
68 RPF_MWh_2min = [];
69 Ss_2fin = [];
70 Sc_2fin = [];
71 I = Num_Impianti;
72 % Contatori ausiliari
73 II = 1;
74 i = 1;

```

```

75 kks = 1;
76
77 P_TOT_1 = zeros(size(PowerMatrix_P,1),length(nodi_installazione
    ));
78 while I > 0
79 acca = sum(revPF_1(:,4:end),1);
80 [val pos] = max(acca(:,:));
81 revPF_1(:,3 + pos) = -10^6;
82     if Num_Impianti ==1
83         nodi_installazione = pos_1(kks);
84     else
85         nodi_installazione_temp = pos_1(kks);
86         if isempty(nodi_installazione) == true
87             nodi_installazione(1,1) = nodi_installazione_temp
                (1,1);
88         else
89             nodi_installazione(1,end+1) =
                nodi_installazione_temp(1,1);
90             i = i+1;
91         end
92     end
93
94 for time = 1 : length(time_step)
95 for bb=1:length(nodi_installazione)
96     if revPF(time,bb)<=0
97         revPF(time,bb) = 0;
98     end
99 end
100 end
101 Ptg.taglie = taglie_P2G;
102 Ptg.profili = revPF;
103
104 for h = 1 : length(time_step)
105 for aa = 1 : length(nodi_installazione)
106     ERIP=ERIP_1(:,aa);

```

```

107         P2G_CAPACITY=P2G_CAPACITY_1;
108         nodi = nodi_installazione(aa);
109         tind= h;
110
111         Out= function_ptgnode_new(Ptg,tind,Out,nodi);
112         P_temp(h,1) = Out.P_temp(h,1);
113         P_TOT_1(h,aa) = P_temp(h,1);
114         h2_tank_temp(h,1) = Out.h2_tank(h,1);
115         H2_TANK(h,aa) = h2_tank_temp(h,1);
116     end
117 end
118
119 for v = 1 : length(nodi_installazione)
120     for h = 1 : length(time_step)
121         Scar_pu(h,nodi_installazione(v)) = Scar_pu(h,
122             nodi_installazione(v)) + P_TOT_1(h,v);
123     end
124 end
125 [Perdite, V0, Ibase, V_n_tot, Ib_abs, Ib_mat, Is_fin,
126     Perdite_allocate, Perdite_allocate_tot, Scar_pu,
127     livello_carico_tot] =
128     function_backward_forward_sweep_NO_SCAR(Scar_pu,
129     PowerMatrix_P, PowerMatrix_Q, Profili_PV);
130
131 %
132 V_n_tot_PTG = V_n_tot;
133 Ib_mat_PTG = transpose(Ib_mat);
134 Ib_mat_PTG_abs = abs(Ib_mat_PV);
135 Ib_mat_PTG_abs_A = Ib_mat_PTG_abs .* Ibase; % A
136 Perdite_PTG = transpose(Perdite);
137 livello_carico_tot_PTG = livello_carico_tot;
138 %%
139 % Indici rete
140 time_st = 15; % time step
141 h_1 = 60; % min

```

```

137 tau = time_st/h_1; % step minimo in ore (1/4)
138 % Valutazione indici di rete dopo l'inserimento del Ptg
139 disp('Rete con presenza di impianti Ptg');
140 %
141 Sbase = 1000; %kVA
142 disp('Energia persa Totale')
143 Perdite_totali = Perdite(:,:);
144 Scar_pu_totali = Scar_pu(:,:);
145 aa_totali = sum(sum(Perdite_totali));
146 bb_totali = sum(sum(real(Scar_pu_totali)));
147 cC_totali = (aa_totali/bb_totali) * 100;
148 cC_totali_2 = cC_totali;
149     if isempty(cC_totali_2fin)
150         cC_totali_2fin(1,1) = cC_totali_2;
151     else
152         cC_totali_2fin(1,end+1) = cC_totali_2;
153     end
154 disp(chop(cC_totali_2,2));
155 %
156 Ib_mat_trans = transpose(Ib_mat);
157 Ib_mat_slack = Ib_mat_trans(:,1:3);
158 Sslack_totali = V0*ones(size(PowerMatrix_P,1),3) .* conj(
    Ib_mat_slack);
159 bB_totali = -(aa_totali / sum(sum(real(Sslack_totali),1)))*100;
160 bB_totali_2 = bB_totali;
161 disp(chop(bB_totali_2,2))
162 %
163 %   RPF
164 S_not_pu = V0*ones(size(PowerMatrix_P,1),3) .* conj((Ib_mat_PTG
    (:,1:3))); %.* Sbase;
165 S_not_pu = S_not_pu .* Sbase;
166 P_not_pu = real(S_not_pu);
167 E_HV_MV = tau .* P_not_pu;
168 %E_HV_MV = tau .* (V0*ones(size(PowerMatrix_P,1),3) .* (
    Ib_mat_slack));

```

```

169 e_HV_MV = real(E_HV_MV);
170 ind_x = find(e_HV_MV(:, :) > 0);
171
172 RPF = sum(e_HV_MV(ind_x)); %kWh
173 RPF_MWh = RPF/(10^3); %MWh
174     if isempty(RPF_MWh_2)
175         RPF_MWh_2(1,1) = RPF_MWh;
176     else
177         RPF_MWh_2(1,end+1) = RPF_MWh;
178     end
179 disp('RPF_MWh: '); disp(RPF_MWh_2(1,end));
180
181 % RPF TIME
182 Ib_slack_tot = Ib_mat_slack(:,3); %sum(Ib_mat_slack,2);
183 RPF_min = size(find(E_HV_MV(:, :) > 0),1) * time_st;
184     if isempty(RPF_MWh_2min)
185         RPF_MWh_2min(1,1) = RPF_min;
186     else
187         RPF_MWh_2min(1,end+1) = RPF_min;
188     end
189 disp('RPF_min: '); disp(RPF_min);
190 %
191 % GRID SELF SUFFICIENCY
192 E_tot_RES = tau * sum(sum(Profili_PV));
193 E_loads = tau * sum(sum(PowerMatrix_P));
194 Ss = ((E_tot_RES - (RPF)) / E_loads) * 100;
195 Ss_2 = Ss;
196     if isempty(Ss_2fin)
197         Ss_2fin(1,1) = Ss_2;
198     else
199         Ss_2fin(1,end+1) = Ss_2;
200     end
201 disp('Ss: '); disp(Ss_2);
202 %
203 % GRID SELF CONSUMPTION

```

```

204 Sc = ((E_tot_RES - (RPF)) / (E_tot_RES)) * 100;
205 Sc_2 = Sc;
206     if isempty(Sc_2fin)
207         Sc_2fin(1,1) = Sc_2;
208     else
209         Sc_2fin(1,end+1) = Sc_2;
210     end
211 disp('Sc:␣'); disp(Sc_2);
212 % Incremento contatori
213 II = II+1;
214 I = I-1;
215
216 kks = kks+1;
217 if kks ==size(pos_1,2)+1
218     break
219 end
220
221 end

```

Codice A.10.1: Main_inserimento_P2G(codice completo)

Bibliografia

- [1] NASA, “Climate change: How do we know?.” <https://climate.nasa.gov/evidence/>, 2020.
- [2] NASA, “Temperature vs solar activity.” https://climate.nasa.gov/climate_resources/189/graphic-temperature-vs-solar-activity/, 2020.
- [3] Terna, “Terna insight.” <https://www.terna.it/it/sistema-elettrico/ruolo-terna/insight>, 2020.
- [4] Terna, “Elettricità e gas, la grande alleanza per la nuova energia verde.” <https://lightbox.terna.it/it/scenari-elettricit -gas>, 2018.
- [5] Terna, “Piano di sviluppo rete, scenari.” <https://www.terna.it/it/sistema-elettrico/rete/piano-sviluppo-rete/scenari>, 2019.
- [6] Planet, “Progetto planet.” <https://www.h2020-planet.eu/>, 2019.
- [7] A. Mazza, E. Bompard, and G. Chicco, “Applications of power to gas technologies in emerging electrical systems,” May 2018. doi.org/10.1016/j.rser.2018.04.072.
- [8] Ri-energia, “Carbon capture and storage: un contributo alla riduzione della co2?.” <https://rienergia.staffettaonline.com/articolo/32786/Carbon+Capture+and+Storage:+un+contributo+alla+riduzione+della+CO2/Tautonico>, 2017.
- [9] L. Bottaccioli, A. Estebarsari, E. Patti, E. Pons, and A. Acquaviva, “Planning and real-time management of smart grids with high pv penetration in italy,” 2000. [https://doi.org/10.1016/S0378-7796\(01\)00101-8](https://doi.org/10.1016/S0378-7796(01)00101-8).

- [10] A. Mazza, F. Salomone, S. Bensaid, E. Bompard, and G. Chicco, “Impact of power-to-gas on distribution systems with large renewable energy penetration,” In revisione.
- [11] “Consumo medio energia elettrica di una famiglia in italia.” <https://bolletta-energia.it/comparazione/consumo-medio-famiglia>.
- [12] Wikipedia, “Riscaldamento globale.” https://it.wikipedia.org/wiki/Riscaldamento_globale.
- [13] Wikipedia, “Gruppo intergovernativo sul cambiamento climatico.” https://it.wikipedia.org/wiki/Gruppo_intergovernativo_sul_cambiamento_climatico.
- [14] Terna, “D come decarbonizzare.” <https://lightbox.terna.it/it/d-come-decarbonizzare>, 2019.
- [15] Wikipedia, “XXI Conferenza delle Nazioni Unite sui cambiamenti climatici (UNFCCC (COP 21)).” https://it.wikipedia.org/wiki/XXI_Conferenza_delle_Parti_dell%27UNFCCC.
- [16] Commissione europea, “Clean energy for all europeans package.” https://ec.europa.eu/energy/topics/energy-strategy/clean-energy-all-europeans_en, 2016.
- [17] Terna, “Documento di descrizione degli scenari.” https://download.terna.it/terna/DDS%20libro%2009%2030%2017h15_8d745ced8696c60.pdf, 2019.
- [18] ARERA, “Delibera 654/2017/r/eel.” <https://www.arera.it/allegati/docs/17/654-17.pdf>, 2017.
- [19] ARERA, “Delibera 689/2017/r/gas.” <https://www.arera.it/allegati/docs/17/689-17.pdf>, 2017.
- [20] Commissione europea, “Planning and operational tools for optimising energy flows and synergies between energy networks.” <https://ec.europa.eu/inea/en/horizon-2020/projects/h2020-energy/grids-storage/planet>, 2019.

- [21] PLANET, “3.1 power-to-gas deployment for system control (uc1),” pp. 101–102, 2019.
https://static1.squarespace.com/static/5a3297ded7bdce9ea2f39f1a/t/5ddd2b5842e31745e888026b/1574775656517/D4.1_Planet_Final_PU.pdf.
- [22] J. Bright, C. Smith, P. Taylor, and R. Crook, “Stochastic generation of synthetic minutely irradiance time series derived from mean hourly weather observation data,” 2015.
[dx.doi.org/10.1016/j.solener.2015.02.032](https://doi.org/10.1016/j.solener.2015.02.032).
- [23] A. Mazza and G. Chicco, “Losses allocated to the nodes of a radial distribution system with distributed energy resources — a simple and effective indicator,” 2019. doi.org/10.1109/SEST.2019.8849127.
- [24] E. Giglio, F. A. Deorsola, M. Gruber, S. R. Harth, E. A. Morosanu, D. Trimis, S. Bensaid, and P. Raffaele, “Power to gas trough high temperature electrolysis and carbon dioxide methanation: reactor design and process modeling,” 2018. doi.org/10.1021/acs.iecr.8b00477.
- [25] Wikipedia, “Cattura e sequestro del carbonio.” https://it.wikipedia.org/wiki/Cattura_e_sequestro_del_carbonio.
- [26] Wikipedia, “Immagazzinamento dell'idrogeno.” https://it.wikipedia.org/wiki/Immagazzinamento_dell%27idrogeno.
- [27] Labelab, “La produzione di idrogeno e metano sintetico da fonte rinnovabile non programmabile.” <https://www.labelab.it/site/wp-content/uploads/sites/5/atti/WD-Ancona.pdf>, 2016.
- [28] G. Chicco, “Appunti del corso di distribuzione dell'energia elettrica.” Politecnico di Torino, 2018.
- [29] T. Ackermann, G. Andersson, and L. Söder, “Distributed generation: a definition,” 2018. doi.org/10.1680/jensu.17.00066.

- [30] IEEE SA, “1547-2018 - ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces.” <https://standards.ieee.org/standard/1547-2018.html>.
- [31] F. Spertino, F. Corona, and P. Di Leo, “Limits of advisability for master–slave configuration of dc–ac converters in photovoltaic systems,” vol. 2, no. 4, pp. 547–554, 2012. doi.org/10.1109/JPHOTOV.2012.2203793.
- [32] European Commission, Joint Research Centre Energy Efficiency and Renewables Unit, “Photovoltaic geographical information system (pvgis).” <https://ec.europa.eu/jrc/en/pvgis>, 2001.
- [33] ENTSO-E, “Ten year network development plan.” <https://tyndp.entsoe.eu/>.