



**SAPIENZA**  
UNIVERSITÀ DI ROMA

FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE, INFORMATICA E STATISTICA

TESI DI LAUREA IN INGEGNERIA INFORMATICA E AUTOMATICA

# **Controllo di un sistema di accumulo dell'energia elettrica per la fornitura di servizi di regolazione di potenza in reti elettriche di distribuzione**

**Relatore:**

**Prof. Alessandro Di Giorgio**

**Laureando:**

**Emanuele De Santis**  
**Matricola: 1664777**

**ANNO ACCADEMICO 2016/2017**



# Indice

<b>Sommario</b>	<b>1</b>
<b>1 Introduzione</b>	<b>2</b>
<b>2 Formalizzazione matematica</b>	<b>4</b>
2.1 Definizione dell'ambiente di lavoro . . . . .	4
2.1.1 Notazione . . . . .	4
2.1.2 Dati di linea . . . . .	5
2.2 Controllo dell'ESS in cabina primaria . . . . .	7
2.3 Controllo dell'ESS in cabina secondaria . . . . .	9
<b>3 Simulazione di rete nel caso con ESS in cabina primaria</b>	<b>12</b>
3.1 Simulazione con disturbo sinusoidale . . . . .	12
3.2 Simulazione con disturbo randomico . . . . .	14
3.3 Simulazione con disturbo a campana . . . . .	16
<b>4 Simulazione di rete nel caso di ESS lungolinea</b>	<b>20</b>
4.1 Simulazioni con previsioni di potenza costanti . . . . .	21
4.1.1 Simulazione con disturbo sinusoidale . . . . .	21
4.1.2 Simulazione con disturbo randomico . . . . .	23
4.1.3 Simulazione con disturbo a campana . . . . .	24
4.2 Simulazioni con previsioni di potenza reali . . . . .	26
4.2.1 Simulazione con disturbo sinusoidale . . . . .	27
4.2.2 Simulazione con disturbo randomico . . . . .	28
4.2.3 Simulazione con disturbo dell'ampiezza rispetto alla previsione . . . . .	28

<b>Conclusioni</b>	<b>30</b>
<b>A Codice Matlab</b>	<b>31</b>
A.1 newton_raphson.m . . . . .	31
A.2 optimization.m . . . . .	35
A.3 Sample_ESS.m . . . . .	36
A.4 optimization_secondary.m . . . . .	39
A.5 Secondary_ESS.m . . . . .	44
A.6 Real_ESS.m . . . . .	49
<b>Bibliografia</b>	<b>54</b>

# Sommario

In questa tesi si propone una strategia di controllo di un sistema di accumulo dell'energia elettrica (Energy Storage System - ESS) per applicazioni nell'ambito delle reti di distribuzione. L'obiettivo del controllo è quello di mitigare eventuali disturbi sulla rete di trasmissione dovuti a variazioni di potenza in uno qualsiasi dei nodi di rete rispetto alla pianificazione *day-ahead*. In particolare si cerca di inseguire il valore di riferimento dato dalla potenza calcolata il giorno prima sul nodo in cabina primaria (cioè il collegamento tra rete di trasmissione e rete di distribuzione).

Nello specifico sono state analizzate due situazioni operative: una in cui l'ESS è installato in cabina primaria, l'altra in cui l'ESS è posto lungolinea (in una delle cabine secondarie).

La strategia di controllo consiste nella risoluzione delle equazioni del Power-Flow per il calcolo del punto di lavoro della rete e nella risoluzione di problemi di ottimizzazione con vincoli lineari per ripartire in modo ottimo il disturbo di potenza attiva tra la rete di trasmissione e lo storage. Nel secondo caso operativo il problema di Power-Flow è stato linearizzato per permettere di mantenere lineari i vincoli del problema di ottimizzazione.

L'approccio proposto è stato validato in simulazione considerando diverse condizioni di lavoro: ESS installato in cabina primaria ed ESS installato lungolinea. Inoltre è stata simulata anche una situazione realistica con previsioni di potenza disponibili sul sito di Terna S.p.A.

Dalle simulazioni si evince che il sistema rimane stabile semplicemente in risposta a disturbi sinusoidali (in generale a media nulla), mentre non riesce a mantenere la stabilità in caso di disturbi randomici. Se il disturbo è impulsivo, invece, il sistema rimane asintoticamente stabile. Per le simulazioni sono stati utilizzati dati di rete reali forniti da un Distribution System Operator locale.

# Capitolo 1

## Introduzione

Con la nascita delle reti per la trasmissione e la distribuzione dell'energia elettrica ci si è trovati di fronte al problema del calcolo dei parametri di rete quali tensioni, angoli di fase e potenze attive e reattive.

Finora, grazie ad ottimi modelli statistici, è stato possibile calcolare il punto di lavoro della rete il giorno precedente (attraverso la *pianificazione day-ahead*) ed avere una imprecisione sulla pianificazione di circa il 3%.

Con una maggiore penetrazione delle fonti rinnovabili l'imprecisione di questi modelli è aumentata sensibilmente e questo porta spesso a dover far fronte, in maniera poco ottimale, a situazioni di ammanco o surplus di potenza (ad esempio dovendo accendere sul momento impianti di generazione tradizionali).

Una soluzione a questo problema può essere quella di posizionare sulla rete di distribuzione dei sistemi di accumulo dell'energia elettrica (**Energy Storage System - ESS**) che permettono di immagazzinare energia quando c'è un surplus di potenza e rilasciarla quando c'è maggiore richiesta.

L'utilizzo tipico dei sistemi di storage è quello di gestione dei picchi di carico sulla rete di distribuzione, anche dovuti a cali di produzione di sorgenti intermittenti (come impianti di generazione da energie rinnovabili) [1]. Inoltre si possono fronteggiare situazioni critiche (ad esempio blackout totali o parziali) in contesti più piccoli di quelli analizzati in questa trattazione (ad esempio nelle *microgrids*). [2]

Nel contesto delle reti di distribuzione, invece, sono stati installati alcuni ESS (nell'ambito del progetto europeo GRID4EU) in Francia ed in Italia.

In Francia sono stati implementati 3 livelli di accumulo: il primo a livello abitativo (connesso direttamente ad impianti PV domestici), il secondo a livello residenziale e l'ultimo a livello di rete di distribuzione. In quest'ultimo

livello sono presenti degli ESS sia in cabina primaria (1MW, 600kWh) che nelle cabine secondarie (250kW, 600kWh) [2]

In Italia, invece, Enel ha installato degli storage in sperimentazione da 1MW, 1MWh, che consentono la modulazione di potenza attiva sulla rete di distribuzione. Questi ESS sono controllati direttamente dal DSO (*Distribution System Operator*). [2]

In assenza di ESS, al verificarsi di un disturbo significativo rispetto alla pianificazione, i generatori sincroni tradizionali non sono sempre in grado di gestire rapidamente il disturbo e di mantenere stabile la rete. Al contrario, un ESS riesce a rispondere molto più rapidamente a variazioni (anche impulsive) di potenza attiva o reattiva (nel caso di studio qui esposto si considereranno solamente variazioni di potenza attiva). Ad esempio è possibile controllare rapidamente variazioni di potenza reattiva tramite sistemi FACTS (*Flexible AC Transmission System*) [3]. Inoltre gli ESS riescono a fornire rapidamente grandi quantità di energia [3, 4], evitando quindi di far variare la velocità di rotazione dei rotori dei generatori tradizionali (il che comporta un minor rischio che vadano in instabilità a causa di angoli di fase troppo ampi)

La soluzione tecnologica offerta dai sistemi di storage è, purtroppo, non ancora attuabile in larga scala a causa degli alti costi delle batterie e dei regolatori e a causa della bassa efficienza degli accumulatori. [2]

Un altro problema è quello di calcolare in ogni istante il punto di lavoro della rete ed attuare un controllo sull'ESS in modo che non raggiunga mai la completa carica o scarica.

Questa tesi ha come obiettivo la progettazione e la simulazione di due controlli di potenza per l'ESS:

1. in cui lo storage si trova direttamente collegato alla cabina primaria (che è la cabina che collega rete di trasmissione e rete di distribuzione);
2. in cui lo storage si trova collegato ad un nodo della rete di distribuzione diverso da quello in cabina primaria (cabina secondaria).

L'approccio di controllo esposto in questa relazione viene trattato similmente (ma con un controllore MPC - Model Predictive Control) da [1, 5, 6], inoltre è stato studiato, in altre varianti, da [7] (che prevede un approccio single-time-slot come quello qui esposto). Sono presenti articoli che trattano il problema di utilizzo di storage in modo ottimale su reti di trasmissione anziché di distribuzione [8, 9].

Sono presenti anche approcci MPC a casi di studio quali sistemi multi-storage nel contesto residenziale [10].

# Capitolo 2

## Formalizzazione matematica

### 2.1 Definizione dell'ambiente di lavoro

#### 2.1.1 Notazione

Nella trattazione si utilizzerà la seguente notazione [1]:

$P_i^l$	Potenza attiva sul nodo i-esimo, positiva se entrante
$Q_i^l$	Potenza reattiva sul nodo i-esimo, positiva se entrante
$P_{st}^{pv}$	Potenza attiva del nodo 2 secondo la previsione short-term
$V_i^l$	Tensione sul nodo i-esimo
$\delta_i^l$	Angolo di fase del nodo i-esimo rispetto al nodo di Slack
$z_i$	Impedenza della linea i-esima
$y_i$	Ammettenza della linea i-esima
$y_{ij}$	Ammettenza in posizione $i,j$ sulla matrice delle ammettenze Y
$G_{ij}$	Parte reale dell'ammettenza $y_{ij}$
$B_{ij}$	Parte immaginaria dell'ammettenza $y_{ij}$
$P^g$	Potenza attiva sulla linea di trasmissione
$Q^g$	Potenza reattiva sulla linea di trasmissione
$P^s$	Potenza attiva sull'ESS
$Q^s$	Potenza reattiva sull'ESS
$x$	Energia immagazzinata nell'ESS
$P_{da}^g$	Potenza attiva di riferimento sulla rete di trasmissione
$x_{ref}$	Energia di riferimento immagazzinata nell'ESS
$T$	Tempo di campionamento



### 2.1.2 Dati di linea

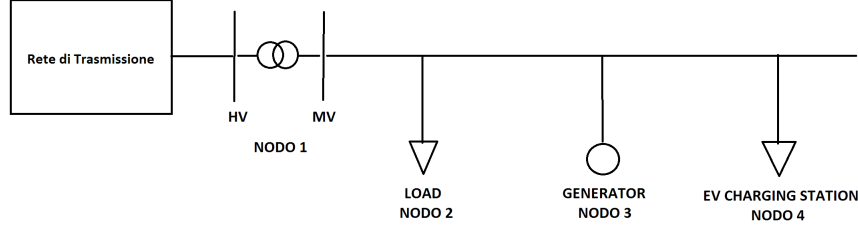


Figura 2.1: Schema della rete di distribuzione presa in esame

Lo studio viene fatto su una rete di distribuzione rappresentata in figura 2.1, composta da quattro nodi, di cui uno in cabina primaria (*Nodo 1*) e tre in cabina secondaria. L'ESS, che non viene rappresentato in figura 2.1, viene inizialmente posto sul Nodo 1, poi su uno qualunque tra gli altri 3 nodi.

Questa rete, per quanto semplice, rappresenta bene una rete di distribuzione reale con collegamenti ad albero.

Le specifiche dei cavi utilizzati sono:

- Cavo in rame di tipo *RG7H1R* con una resistenza  $R = 0.248\Omega$ , una reattanza induttiva  $X_L = 0.2\Omega$  e una capacità  $C = 0.62\mu F$ , che collega i nodi 1-2;
- Cavo di alluminio di tipo *ARG7H1RX* con una resistenza  $R = 0.412\Omega$ , una reattanza induttiva  $X_L = 0.2\Omega$  e una capacità  $C = 0.58\mu F$ , che collega i nodi 2-3 e 3-4

Si ottengono quindi le impedenze delle tre linee

$$z_1 = 0.248 + j0.22$$

$$z_2 = 0.412 + j0.20$$

$$z_3 = 0.412 + j0.20$$

Calcolando quindi la matrice delle ammettenze si ottiene:

$$Y = \begin{pmatrix} 2.2565 - j2 & -2.2565 + j2 & 0 & 0 \\ -2.2565 + j2 & 4.2208 - j2.9535 & -1.9643 + j0.9535 & 0 \\ 0 & -1.9643 + j0.9535 & 3.9286 - j1.907 & -1.9643 + j0.9535 \\ 0 & 0 & -1.9643 + j0.9535 & 1.9643 - j0.9535 \end{pmatrix}$$

Inoltre il trasformatore in cabina primaria è sottoposto a tensioni sugli avvolgimenti di  $150\text{ KV}/20\text{ KV}$  ed ha una potenza nominale di  $40\text{ MVA}$ . I trasformatori in cabina secondaria sono sottoposti a tensioni sugli avvolgimenti di  $20\text{ KV}/0.4\text{ KV}$  ed hanno una potenza nominale di  $630\text{ KVA}$ .

Le specifiche sui trasformatori hanno influenzato il dimensionamento delle potenze attive e reattive dei carichi presenti sulla rete, nonché le tensioni sia in cabina primaria che nelle cabine secondarie.

In particolare sono state fissate le seguenti potenze per i vari carichi:

Nodo	Potenza Attiva	Potenza Reattiva	$\cos\phi$
2	600KW	190KVA <sub>r</sub>	0.953
3	620KW	50KVA <sub>r</sub>	0.997
4	620KW	50KVA <sub>r</sub>	0.997

Il nodo 1, che svolge il ruolo di nodo di Slack, ha una tensione fissata a 20KV ed un angolo di fase di  $0^\circ$

Inoltre le specifiche fissate per l'ESS sono:

- Potenza attiva massima 2MW
- Energia immagazzinata massima 2MWh
- Riferimento di carica 50%

## 2.2 Controllo dell'ESS in cabina primaria

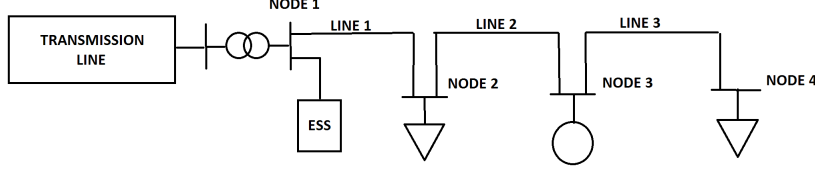


Figura 2.2: Schema della rete di distribuzione con l'ESS sul nodo in cabina primaria

Il problema qui trattato consiste nel determinare il controllo di potenza dell'ESS in modo da minimizzare lo scarto fra flusso di potenza e riferimento sul nodo 1 e di minimizzare a sua volta lo scarto tra lo stato di carica dell'ESS e un valore di riferimento.

Poichè questi due obiettivi sono in contrasto è necessario trovare un controllo che minimizzi il più possibile entrambi gli scarti.

Si può notare che è possibile disaccoppiare la risoluzione del problema di Power Flow e la ricerca della soluzione ottima del problema, infatti, una volta calcolata la potenza sul nodo 1, questa diventa vincolo per il successivo problema di ottimizzazione.

Si assume di possedere una previsione *day-ahead* di potenza attiva e reattiva per i nodi 2, 3 e 4, e di tensione ed angolo di fase per il nodo 1. Questi dati possono essere usati per risolvere il seguente problema di Power Flow (ad esempio tramite Newton-Raphson)

$$\begin{cases} P_2^l = V_2^{l^2} G_{22} + \sum_{j \neq 2} V_2^l V_j^l (B_{2j} \sin(\delta_2^l - \delta_j^l) + G_{2j} \cos(\delta_2^l - \delta_j^l)) \\ Q_2^l = -V_2^{l^2} B_{22} + \sum_{j \neq 2} V_2^l V_j^l (G_{2j} \sin(\delta_2^l - \delta_j^l) - B_{2j} \cos(\delta_2^l - \delta_j^l)) \\ P_3^l = V_3^{l^2} G_{33} + \sum_{j \neq 3} V_3^l V_j^l (B_{3j} \sin(\delta_3^l - \delta_j^l) + G_{3j} \cos(\delta_3^l - \delta_j^l)) \\ Q_3^l = -V_3^{l^2} B_{33} + \sum_{j \neq 3} V_3^l V_j^l (G_{3j} \sin(\delta_3^l - \delta_j^l) - B_{3j} \cos(\delta_3^l - \delta_j^l)) \\ P_4^l = V_4^{l^2} G_{44} + \sum_{j \neq 4} V_4^l V_j^l (B_{4j} \sin(\delta_4^l - \delta_j^l) + G_{4j} \cos(\delta_4^l - \delta_j^l)) \\ Q_4^l = -V_4^{l^2} B_{44} + \sum_{j \neq 4} V_4^l V_j^l (G_{4j} \sin(\delta_4^l - \delta_j^l) - B_{4j} \cos(\delta_4^l - \delta_j^l)) \end{cases}$$

ponendo  $V_2^l, V_3^l, V_4^l, \delta_2^l, \delta_3^l, \delta_4^l$  come guess iniziali del problema.

In questo modo è possibile ottenere tutte le variabili di stato  $V_i^l$  e  $\delta_i^l$  e le potenze  $P_1^l := P_{da}^g$  e  $Q_1^l := Q_{da}^g$  relative al nodo 1.

Durante la pianificazione *day-ahead* è possibile considerare  $P^s = 0$  e  $Q^s = 0$  poiché il carico è già bilanciato e, se non dovesse variare rispetto alla pianificazione, potrebbe essere sostenuto dalla sola rete di trasmissione.

Nel momento in cui avviene, invece, una variazione su uno qualsiasi dei nodi 2, 3 o 4 (nel caso preso in esame varia la potenza sul nodo 2 e diventa  $P_{st}^{pv}$ ), allora  $P_1^l$  e  $Q_1^l$  diventano i riferimenti di potenza nel contesto *real-time* (e verranno chiamati quindi  $P_{da}^g$  e  $Q_{da}^g$ ) e, ricalcolando il problema di Power Flow, si ottiene un nuovo set di potenza  $P^g$  e  $Q^g$  per il nodo in cabina primaria.

In cabina primaria si ha dunque:

$$\begin{cases} P^g = P_{st}^g + P^s \\ Q^g = Q_{st}^g + Q^s \end{cases}$$

Si nota che  $P^g$  e  $Q^g$  sono effettivamente calcolati, mentre, nella trattazione in cui l'ESS è installato su un nodo lungolinea, non possono essere calcolati finché non viene risolto il problema di ottimizzazione.

Ipotizzando che lo storage possa fornire tutta la potenza reattiva di differenza rispetto alla pianificazione *day-ahead*, rimane solo il problema di calcolare i valori ottimi di potenza attiva sulla rete di trasmissione e sull'ESS. Questi possono essere facilmente trovati risolvendo il seguente problema di ottimizzazione quadratica con vincoli lineari:

$$\begin{cases} \min [(P_{st}^g(t_k) - P_{da}^g)^2 + ((x(t_k) + P^s(t_k)T) - x_{ref})^2] \\ P^g = P_{st}^g(t_k) + P^s(t_k) \end{cases}$$

Viene quindi introdotta una dipendenza dal tempo  $t_k$ , poiché ogni tempo di campionamento  $T$  (solitamente ogni 15 minuti) deve essere nuovamente risolto sia il problema di Power Flow che il problema di ottimizzazione (infatti la carica dello storage varia in base alla decisione presa al tempo  $t_{k-1}$  e quindi la soluzione ottima al tempo  $t_k$  può essere differente da quella al tempo  $t_{k-1}$ ).

Perciò, rispetto alla sola risoluzione del problema di Power Flow con pianificazione *day-ahead*, questa soluzione prevede che l'algoritmo venga eseguito *real-time*.

## 2.3 Controllo dell'ESS in cabina secondaria

Come nel caso esposto finora, si vuole determinare il controllo di potenza dell'ESS in modo che minimizzi lo scarto tra flusso di potenza e riferimento sul nodo 1 e lo scarto fra stato di carica dell'ESS e riferimento.

In questo caso, però, il problema si complica notevolmente dal punto di vista matematico, poiché non è più possibile disaccoppiare il problema di Power-Flow da quello di ottimo vincolato; infatti non è possibile trovare il punto di lavoro della rete finché non si decide quanta potenza erogare o assorbire tramite lo storage, e viceversa non è possibile decidere la potenza dell'ESS finché non si conosce il punto di lavoro della rete.

Questo implica la necessità di inserire l'intero problema di Power-Flow (2.1) come vincolo del problema di ottimizzazione, che quindi non dovrà determinare solamente potenza sulla cabina primaria e potenza dell'ESS, ma anche tensioni e angoli di carico di tutta la rete, nonché la potenza reattiva in cabina primaria.

Si ha a disposizione una rete di distribuzione come in figura 2.3, e si dispone di una previsione *day-ahead* di potenza sui nodi 2, 3 e 4, mentre si conosce la tensione e l'angolo di fase al nodo 1 (nodo di Slack). Si risolve quindi un primo problema di Power Flow tramite l'algoritmo di Newton-Raphson per determinare il punto di lavoro *day-ahead* della rete. Si ipotizza poi che durante l'esercizio uno dei nodi (in questo caso sempre il nodo 2) subisca un disturbo di potenza attiva rispetto alla pianificazione. E' necessario, quindi, calcolare la ripartizione ottimale del disturbo tra la rete di trasmissione e lo storage.

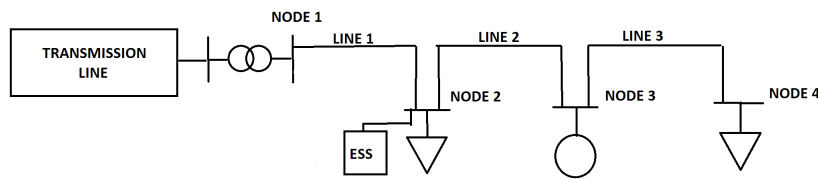


Figura 2.3: Schema della rete di distribuzione con l'ESS in cabina secondaria

L'ESS viene posto in cabina secondaria (più particolarmente sulla stessa del nodo che introduce il disturbo rispetto alla pianificazione *day-ahead*).

$$\begin{cases}
P_2^l = V_2^{l^2} G_{22} + \sum_{j \neq 2} V_2^l V_j^l (B_{2j} \text{sen}(\delta_2^l - \delta_j^l) + G_{2j} \cos(\delta_2^l - \delta_j^l)) \\
P_3^l = V_3^{l^2} G_{33} + \sum_{j \neq 3} V_3^l V_j^l (B_{3j} \text{sen}(\delta_3^l - \delta_j^l) + G_{3j} \cos(\delta_3^l - \delta_j^l)) \\
P_4^l = V_4^{l^2} G_{44} + \sum_{j \neq 4} V_4^l V_j^l (B_{4j} \text{sen}(\delta_4^l - \delta_j^l) + G_{4j} \cos(\delta_4^l - \delta_j^l)) \\
Q_2^l = -V_2^{l^2} B_{22} + \sum_{j \neq 2} V_2^l V_j^l (G_{2j} \text{sen}(\delta_2^l - \delta_j^l) - B_{2j} \cos(\delta_2^l - \delta_j^l)) \\
Q_3^l = -V_3^{l^2} B_{33} + \sum_{j \neq 3} V_3^l V_j^l (G_{3j} \text{sen}(\delta_3^l - \delta_j^l) - B_{3j} \cos(\delta_3^l - \delta_j^l)) \\
Q_4^l = -V_4^{l^2} B_{44} + \sum_{j \neq 4} V_4^l V_j^l (G_{4j} \text{sen}(\delta_4^l - \delta_j^l) - B_{4j} \cos(\delta_4^l - \delta_j^l)) \\
P_1^l = V_1^{l^2} G_{11} + \sum_{j \neq 1} V_1^l V_j^l (B_{1j} \text{sen}(\delta_1^l - \delta_j^l) + G_{1j} \cos(\delta_1^l - \delta_j^l)) \\
Q_1^l = -V_1^{l^2} B_{11} + \sum_{j \neq 1} V_1^l V_j^l (G_{1j} \text{sen}(\delta_1^l - \delta_j^l) - B_{1j} \cos(\delta_1^l - \delta_j^l))
\end{cases} \quad (2.1)$$

Poiché si vogliono mantenere lineari i vincoli del problema di ottimizzazione, è necessario linearizzare le equazioni del problema di Power Flow tramite lo sviluppo di Taylor al primo ordine. Risulta perciò possibile scrivere il problema come

$$\begin{pmatrix} P \\ Q \end{pmatrix} = \begin{pmatrix} P(V_0, \delta_0) \\ Q(V_0, \delta_0) \end{pmatrix} + J|_{V_0, \delta_0} \begin{pmatrix} V - V_0 \\ \delta - \delta_0 \end{pmatrix}$$

Le variabili di interesse  $V$  e  $\delta$  sono solo quelle relative ai nodi 2, 3 e 4 poiché tensione e angolo di fase sul nodo 1 sono fissate. Perciò anche la matrice  $J$  (jacobiana del sistema 2.1) è calcolata derivando parzialmente rispetto alle stesse variabili.

Per il problema di ottimizzazione è necessario introdurre molte più variabili rispetto alla trattazione precedente. In questo caso è stato scelto di mettere in funzione obiettivo le variabili di stato del problema di Power Flow anziché la potenza dell'ESS, in modo da semplificare la matrice che rappresenta i vincoli del problema. Si è quindi ottenuto il sistema:

$$\left\{ \min \left\{ [P^g - P_{da}^g]^2 + \left[ x + \left( P^{pv}(V_0, \delta_0) - P_{st}^{pv} + J_1|_{V_0, \delta_0} \begin{pmatrix} V - V_0 \\ \delta - \delta_0 \end{pmatrix} \right) T - x_{ref} \right]^2 \right\} \right. \\ \left. \left( \begin{pmatrix} 0 & -1 \\ 0 & 0 \\ \cdot & \cdot \\ J|_{(V_0, \delta_0)} & \cdot \\ \cdot & \cdot \\ -1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V - V_0 \\ \delta - \delta_0 \\ P^g \\ P^s \end{pmatrix} = \begin{pmatrix} P_{st}^{pv} - P_2(V_0, \delta_0) \\ P_3 - P_3(V_0, \delta_0) \\ P_4 - P_4(V_0, \delta_0) \\ Q_2 - Q_2(V_0, \delta_0) \\ Q_3 - Q_3(V_0, \delta_0) \\ Q_4 - Q_4(V_0, \delta_0) \\ -P^g(V_0, \delta_0) \\ Q_1 - Q_1(V_0, \delta_0) \end{pmatrix} \right) \right\}$$

Con  $J_1$  si indica la prima riga della matrice jacobiana.

Da questo sistema si evince che le variabili del problema diventano 8 (le tensioni e angoli di fase non noti, la potenza della rete di trasmissione e la potenza dello storage) mentre nella trattazione precedente si riusciva a scrivere il problema con una sola variabile.

Il problema di ottimizzazione ha necessità di un punto di linearizzazione (denotato da  $(V_0, \delta_0)$ ), che può essere scelto, in base alla tipologia di disturbo: pari al punto di lavoro della rete come nella previsione *day-ahead*, oppure pari al punto di lavoro calcolato nel campione precedente.

In questa trattazione si è considerato il caso di linearizzazione con il punto di lavoro *day-ahead*. Questa scelta comporta il fatto che non si potrà controllare correttamente lo storage se il disturbo si allontana dal punto di lavoro *day-ahead*. In entrambi i casi non è possibile discostarsi dal punto di linearizzazione, a meno di commettere un errore rispetto al sistema non linearizzato.

Si nota inoltre che non tutte le variabili del problema compaiono in funzione obiettivo, ma vengono calcolate indirettamente tramite i vincoli presenti.

## Capitolo 3

# Simulazione di rete nel caso con ESS in cabina primaria

Per simulare l'algoritmo di controllo sviluppato è stata usata la piattaforma *Matlab*. Sono state sviluppate due funzioni: una per il calcolo del problema di Power Flow tramite l'algoritmo di Newton-Raphson e l'altra per la risoluzione del problema di ottimizzazione.

Inoltre è stato sviluppato uno script di simulazione che riproduce la variazione della potenza attiva sul nodo 2 rispetto alla pianificazione.

### 3.1 Simulazione con disturbo sinusoidale

E' stata condotta una prima simulazione che prevede che il disturbo (cioè la potenza attiva sul nodo 2) sia di tipo sinusoidale. Questa simulazione serve a mostrare come il controllo reagisce a disturbi a media nulla.

Dalla figura 3.1 si può notare che il sistema complessivo risulta stabile semplicemente rispetto a segnali sinusoidali (che non eccedano le specifiche di potenza dei trasformatori o le specifiche di carica dell'ESS). Dalla figura 3.1 inoltre si nota quanto sia importante l'utilizzo dell'ESS, in quanto la potenza che proviene dalla rete di trasmissione è pressoché costante sebbene la potenza sul nodo 2 cambi sensibilmente.

Inoltre è stata eseguita una simulazione con un peso maggiore del termine di potenza della linea di trasmissione rispetto a quello della potenza dell'ESS, come mostrato in figura 3.2 (in modo quindi da favorire l'uso di potenza dallo



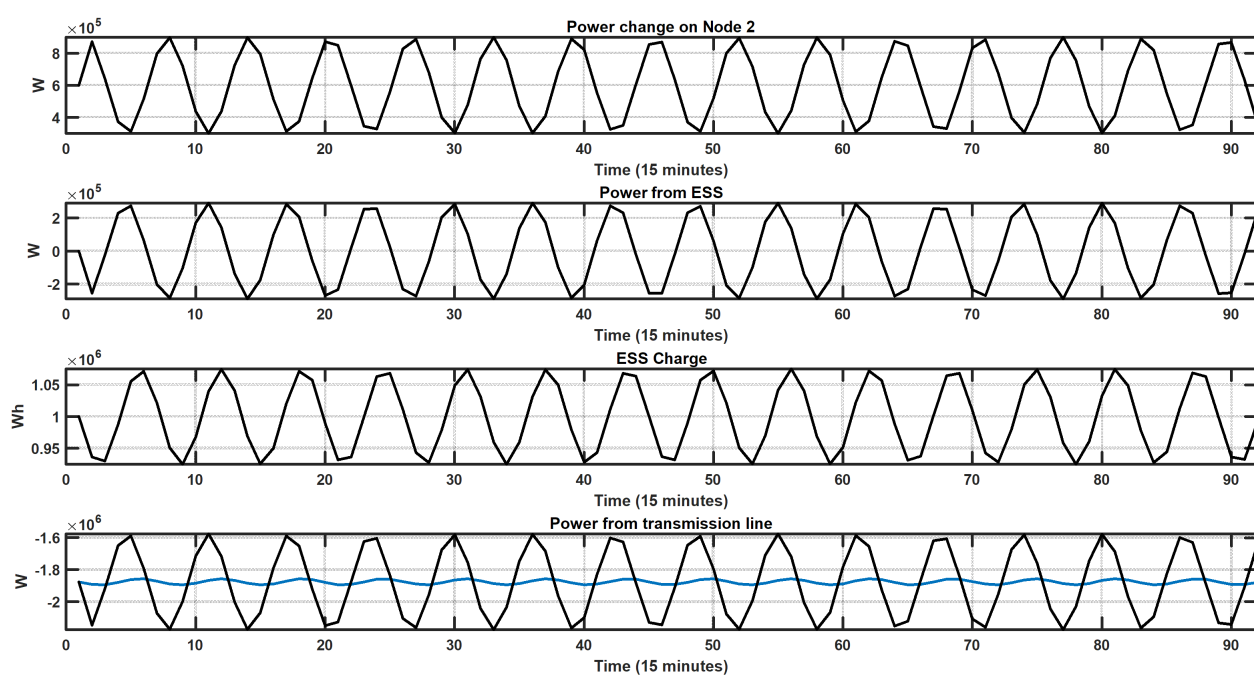


Figura 3.1: Simulazione sinusoidale

storage piuttosto che della rete di trasmissione). Questo ha mostrato come la potenza  $P^g$  sia ancora più vicina al riferimento (a scapito di  $P^s$  che invece subisce quasi tutte le variazioni provenienti dal disturbo).

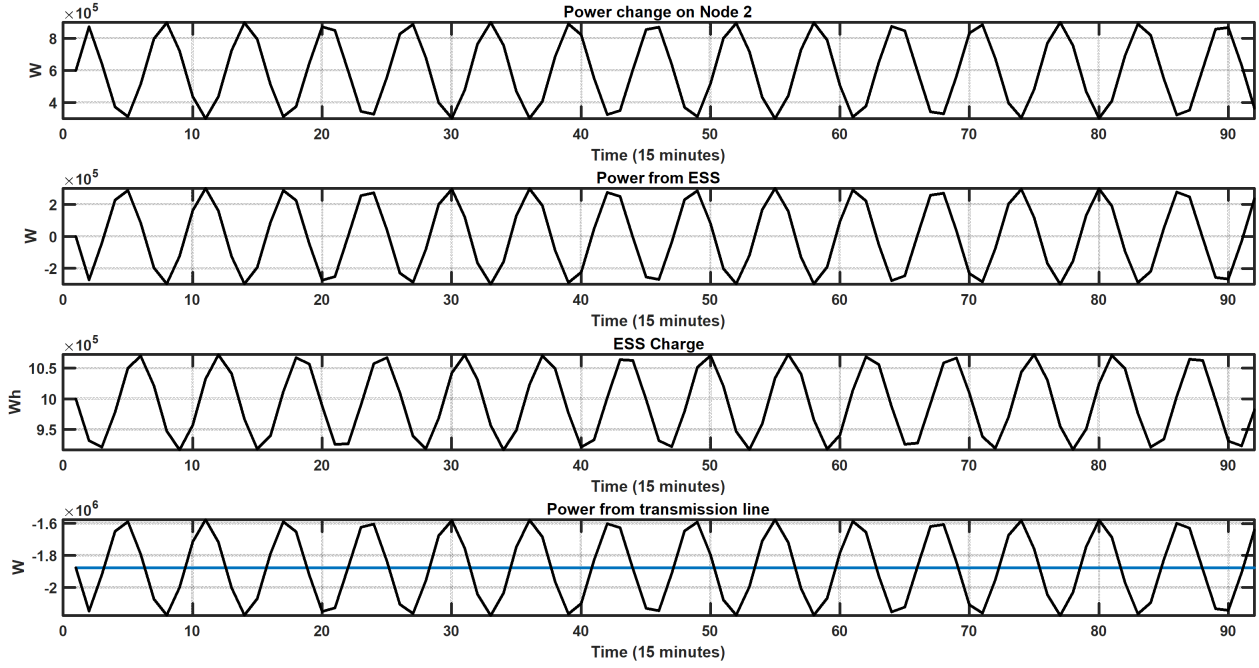


Figura 3.2: Simulazione sinusoidale con peso maggiore alla potenza della linea di trasmissione

### 3.2 Simulazione con disturbo randomico

La seconda simulazione prevede che il disturbo (cioè la potenza attiva sul nodo 2) sia randomico. In questa simulazione si mostra che il controllo non riesce a mantenere la stabilità quando i disturbi non hanno media nulla, mostrando quindi i limiti di utilizzo di questo controllo. Dalla figura 3.3 è possibile vedere bene che la carica  $x$  immagazzinata nello storage diverge (in questo caso positivamente) e, nel tempo, probabilmente raggiungerebbe il limite di massima carica (che è fissato per costruzione a  $2MWh$ ). Questa instabilità è comunque mitigata dalla rete di trasmissione, che si trova costretta a variare rispetto al riferimento.

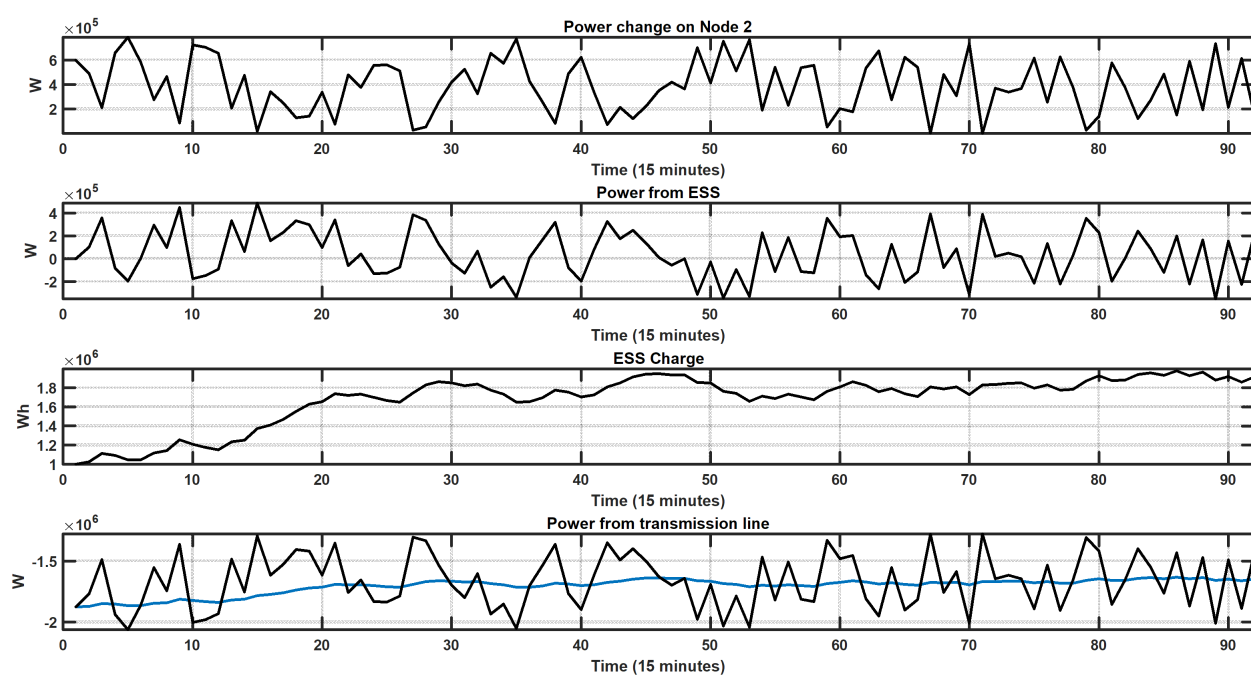


Figura 3.3: Simulazione randomica

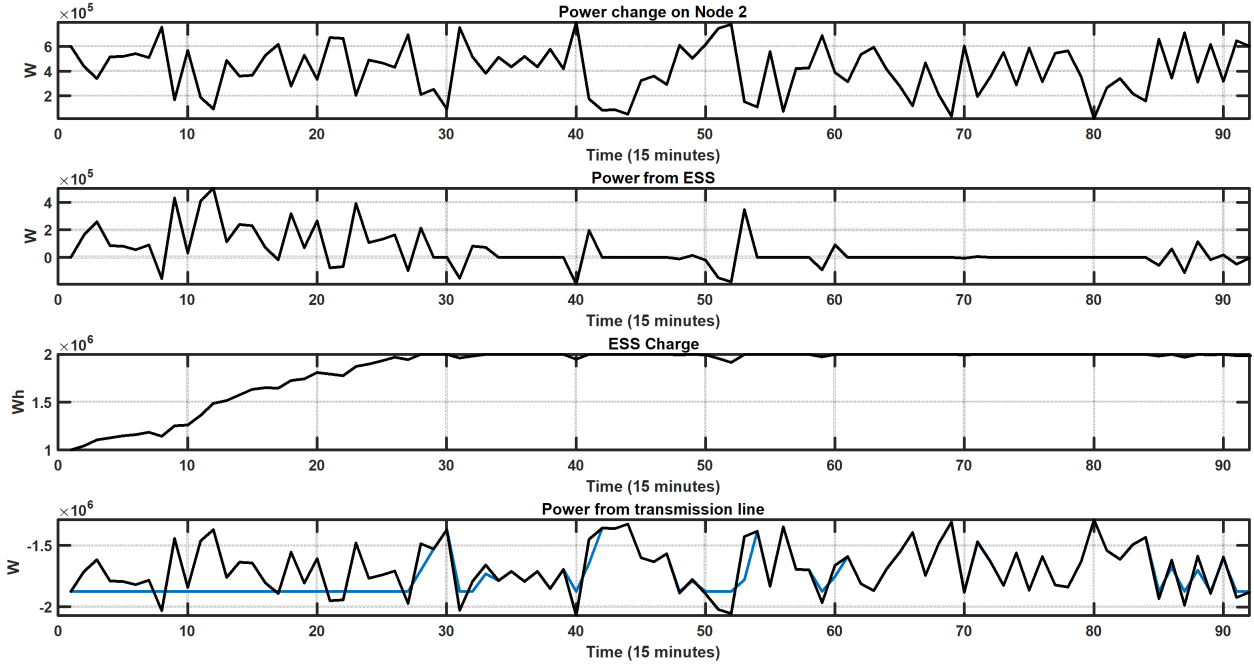


Figura 3.4: Simulazione randomica con peso maggiore alla potenza della linea di trasmissione

Nel caso in cui, invece, si dà un peso maggiore al termine di potenza della linea di trasmissione rispetto a quello di potenza dell'ESS, come mostrato in figura 3.4, allora lo storage va in saturazione (infatti raggiunge il 100% della sua carica). Da quel momento in poi tutte le variazioni della potenza  $P_{st}^{pv}$  vengono assorbite solamente dalla rete di trasmissione, che si trova costretta a seguire le variazioni di  $P_{st}^{pv}$  come se lo storage non fosse presente.

### 3.3 Simulazione con disturbo a campana

La terza simulazione prevede che il disturbo sia a campana. Questa situazione rappresenta bene un carico connesso ad un impianto fotovoltaico che, durante il giorno e in assenza di nuvole, ha una curva di produzione simile a quella di una campana, che si sottrae alla potenza richiesta dal carico. In questo caso è possibile notare, come mostrato in figura 3.5, che sia lo storage che la linea di trasmissione riescono a tornare al loro riferimento una volta

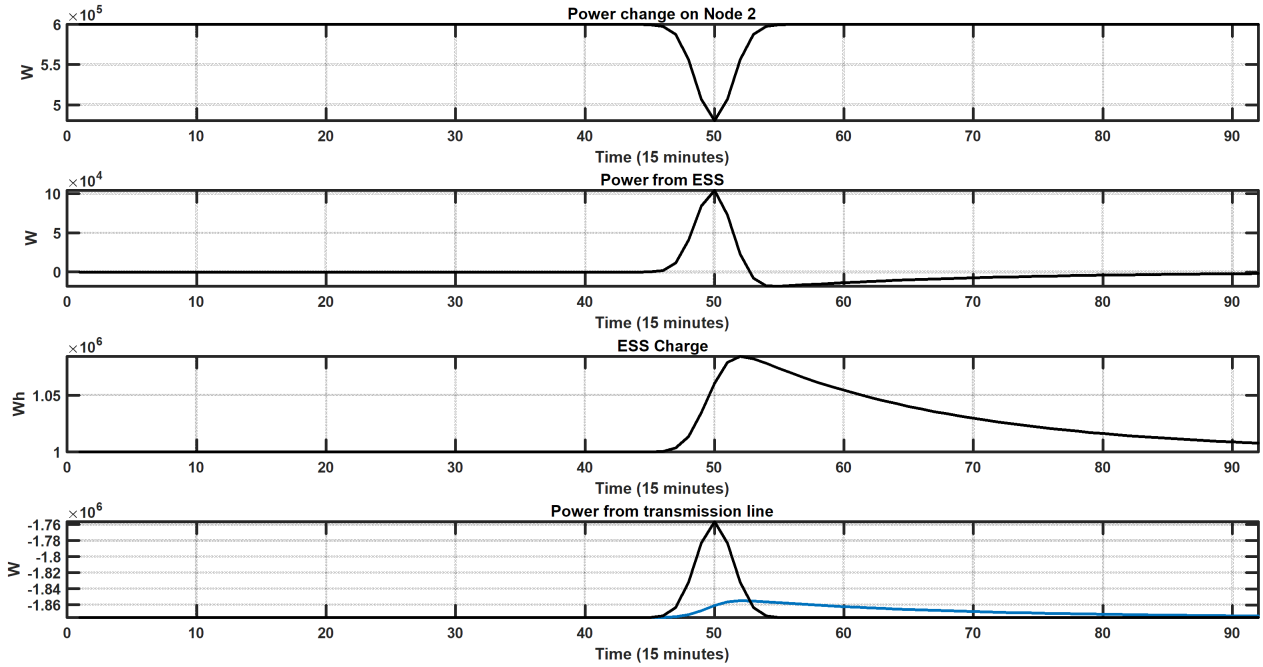


Figura 3.5: Simulazione con disturbo a campana

finito il disturbo. Durante il disturbo, però, si nota un grande aumento della carica nello storage. In questo caso la carica dello storage è rimasta entro i limiti, ma sarebbe potuto andare in saturazione se il disturbo avesse avuto maggiore intensità (come mostrato in figura 3.6), ottenendo quindi una grande variazione della potenza della rete di trasmissione.

Dando un peso maggiore al termine di potenza della linea di trasmissione rispetto a quello di potenza dell'ESS, come mostrato in figura 3.7, si ottiene che la potenza sulla rete di trasmissione è praticamente pari al riferimento per tutte le 24h, e quindi non permette allo storage di scaricarsi e tornare al suo valore di riferimento una volta finito il disturbo.

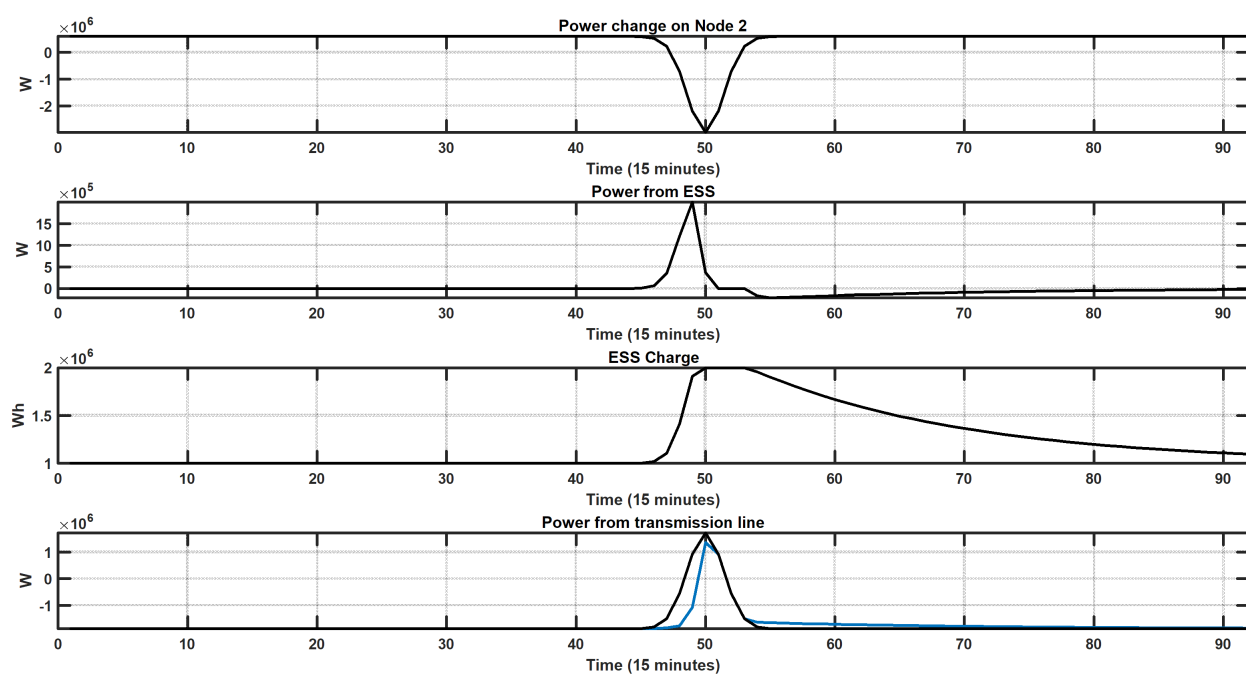


Figura 3.6: Simulazione con disturbo a campana con maggiore intensità del disturbo

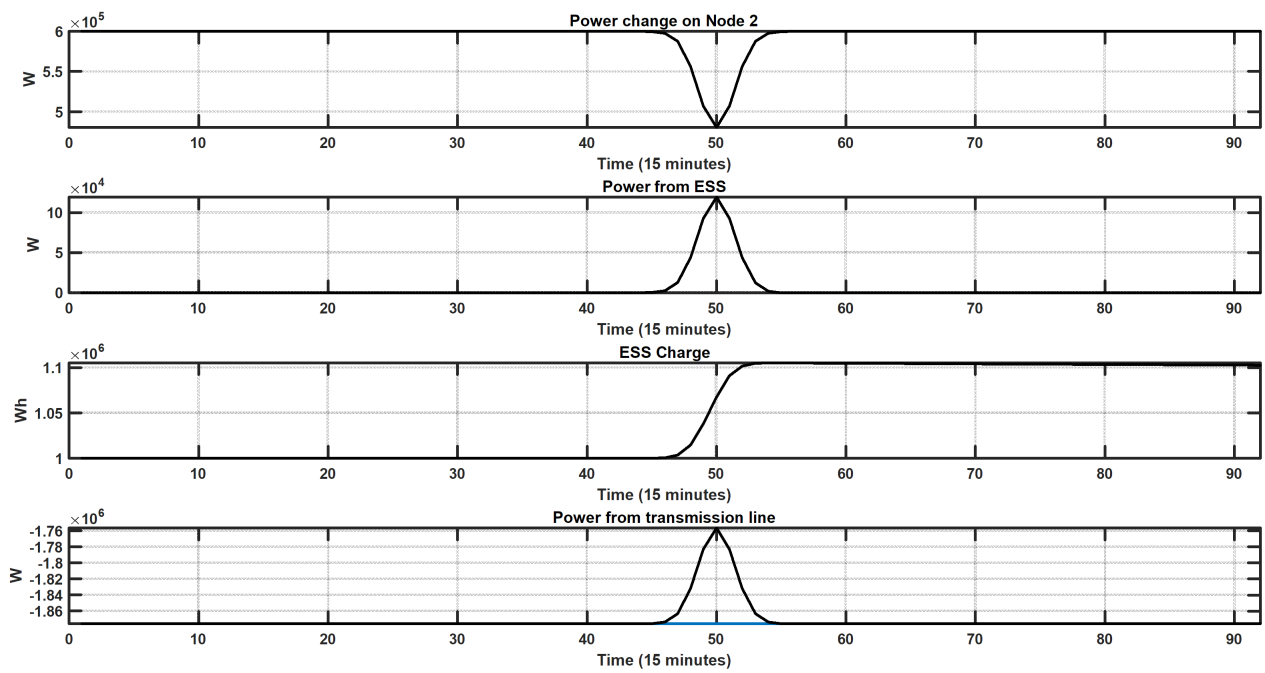


Figura 3.7: Simulazione con disturbo a campana con peso maggiore alla potenza della linea di trasmissione

## Capitolo 4

# Simulazione di rete nel caso di ESS lungolinea

In questa seconda fase di simulazione sono state sviluppate una funzione *Matlab* per la risoluzione del problema di ottimizzazione e uno script che riproduce la variazione di potenza attiva sul nodo 2 e raccoglie i risultati dell'ottimizzazione. Poiché il sistema è stato linearizzato, le simulazioni contengono un errore rispetto al comportamento reale dovuto appunto alla linearizzazione delle equazioni del Power Flow. Questo errore può essere misurato confrontando la soluzione del problema di ottimo (cioè le variabili di stato della rete) con quella derivante dalla risoluzione del problema di Power Flow tramite Newton-Raphson.

In particolare, aumentando progressivamente il disturbo in ampiezza è possibile notare un errore percentuale considerevole (nel caso peggiore) rispetto al problema non linearizzato a partire da un disturbo di ampiezza dell'ordine di  $10^6/10^7$ W. Per disturbi minori di questa soglia l'errore è minore dello 0.5%.



## 4.1 Simulazioni con previsioni di potenza costanti

### 4.1.1 Simulazione con disturbo sinusoidale

Questa simulazione utilizza un disturbo di potenza attiva sul nodo 2 di tipo sinusoidale. E' stato necessario porre questo disturbo molto piccolo rispetto alle potenze in gioco per limitare l'errore dovuto alla linearizzazione. In ogni caso è presente un piccolo errore dovuto alla linearizzazione del sistema.

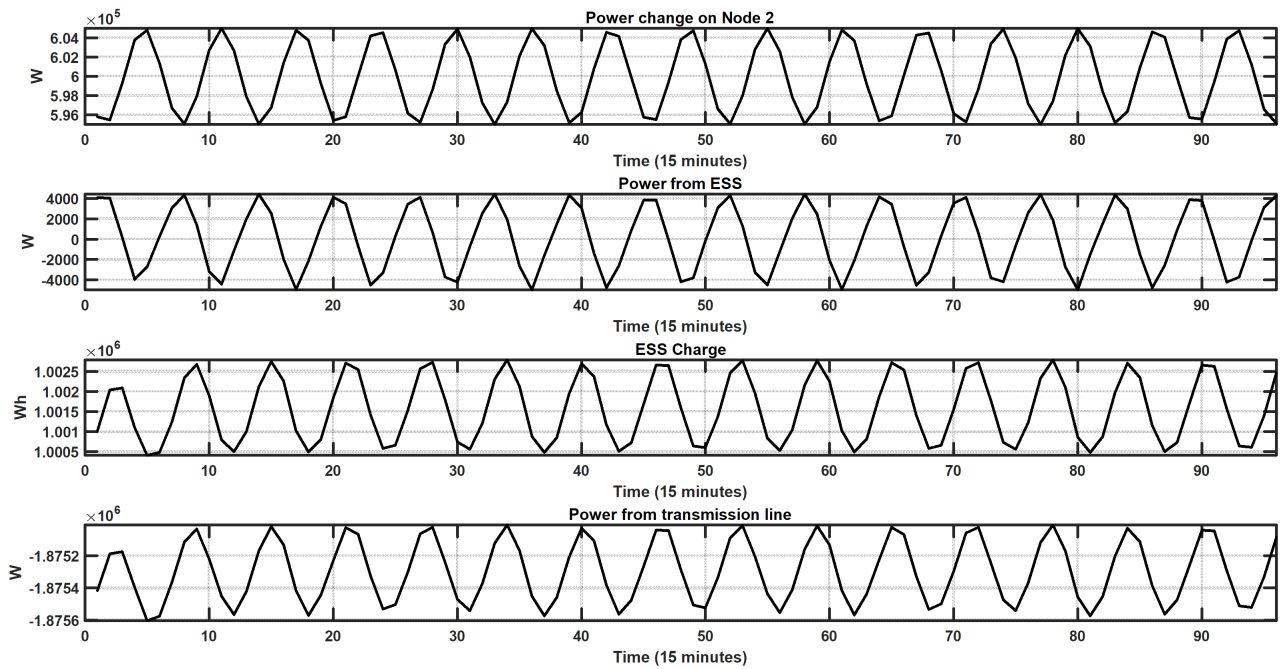


Figura 4.1: Simulazione con disturbo sinusoidale e con ESS in cabina secondaria

Tramite questa tipologia di simulazione è stato possibile notare come il controllo continua a preservare la stabilità dello storage attorno al riferimento desiderato in caso di disturbi sinusoidali (come mostrato in figura 4.1) oppure, più in generale, a media nulla.

E' stata simulata anche la situazione con un peso maggiore in funzione obiettivo alla potenza sulla rete di trasmissione rispetto a quella dello storage, che rappresenta un caso reale in cui il Transmission System Operator (TSO) impone dei vincoli molto rigidi sulla variazione di potenza in cabina primaria (figura 4.2). In questo modo la potenza sulla cabina primaria proveniente dalla rete di trasmissione rimane pressoché costante lungo tutta la simulazione (le variazioni mostrate nella simulazione sono dell'ordine di  $10^{-3}W$ ).

Si nota però che c'è un comportamento leggermente divergente nella carica dello storage, dovuto ad un calcolo impreciso (a causa della linearizzazione) della potenza erogata dallo storage. Quando questo è chiamato a subire tutta la variazione di potenza del nodo 2 questa imprecisione porta al fatto che l'energia immagazzinata diverga verso l'alto. Nel caso invece di pesi bilanciati la rete di trasmissione supplisce a questa imprecisione di calcolo mantenendo il sistema stabile.

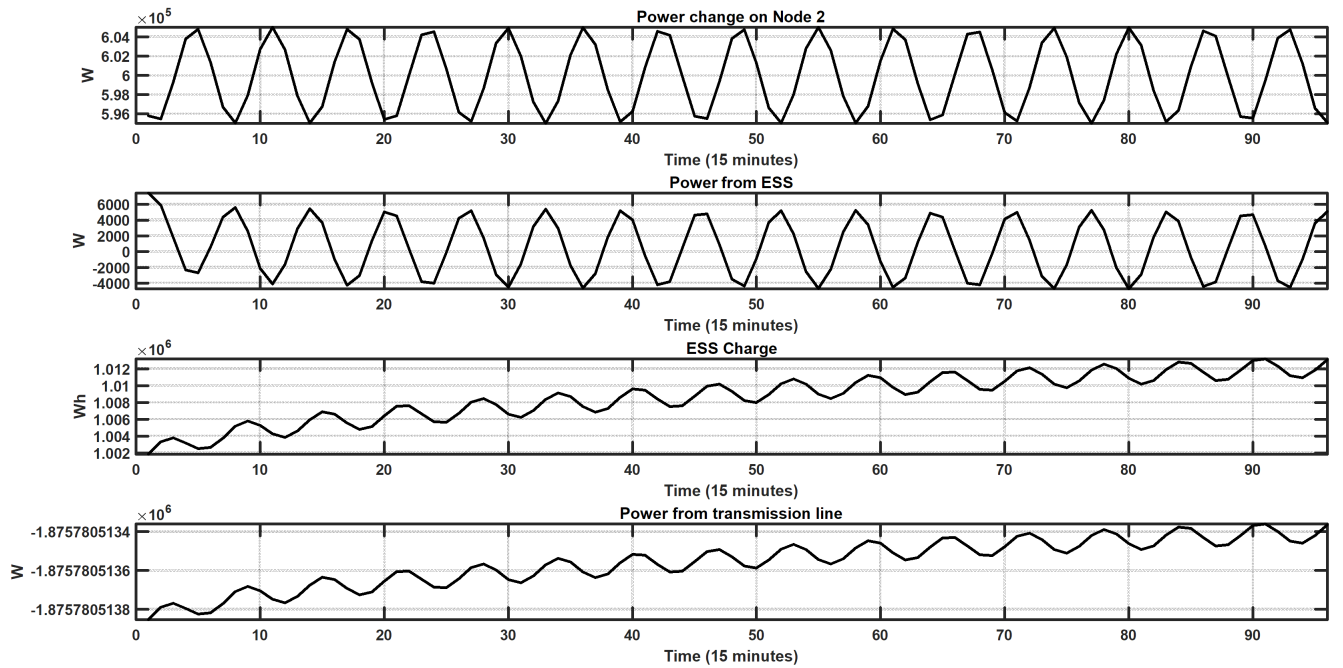


Figura 4.2: Simulazione con disturbo sinusoidale, con ESS in cabina secondaria e con peso maggiore sulla potenza in cabina primaria

### 4.1.2 Simulazione con disturbo randomico

Questa simulazione ha lo scopo di mostrare come il sistema si comporta in caso di disturbo randomico a media non nulla. In questo caso il sistema rimane stabile (come mostrato in figura 4.3), poiché il disturbo è molto piccolo rispetto alla potenza prevista *day-ahead*. In caso di disturbo più ampio il sistema sarebbe andato sicuramente in saturazione (verso l'alto o verso il basso). Non è stato possibile però mostrare questo caso poiché la linearizzazione vincola il punto di lavoro della rete con disturbo vicino al punto di lavoro *day-ahead*.

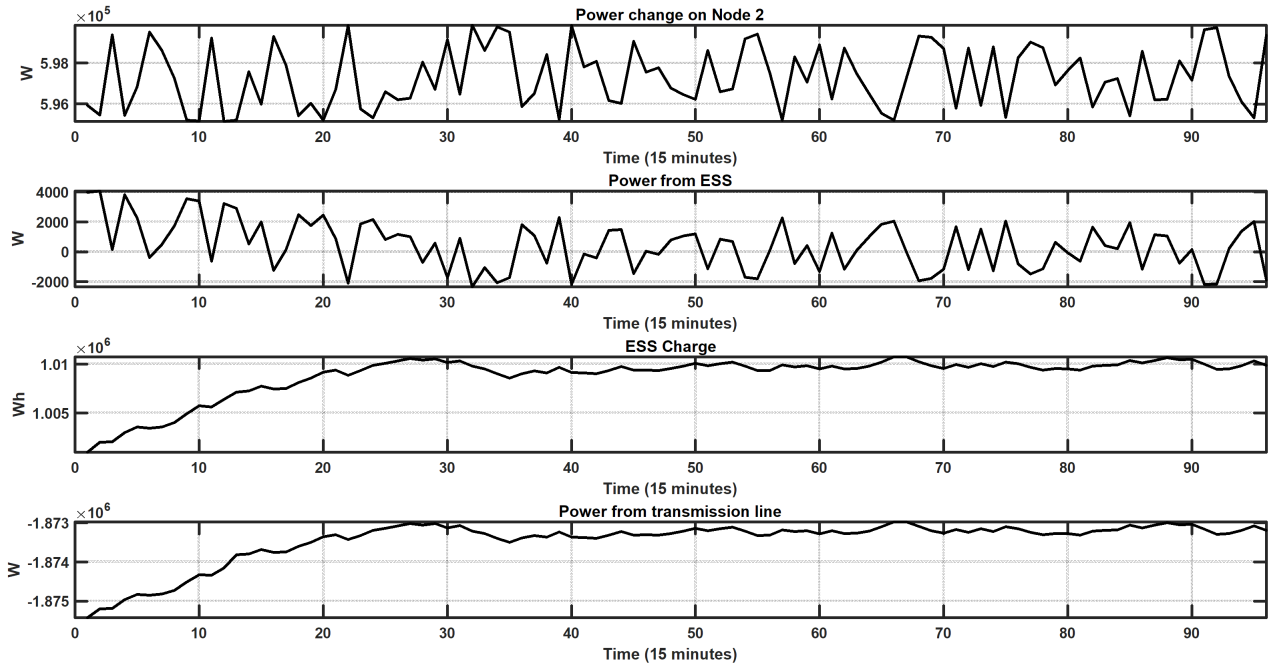


Figura 4.3: Simulazione con disturbo randomico e con ESS in cabina secondaria

A maggior ragione, nel caso di peso maggiore alla potenza della rete di trasmissione, il sistema sarebbe andato in saturazione.

Nella simulazione mostrata in figura 4.4 si ottiene lo stesso problema della figura 4.2, cioè che il sistema diverge verso l'alto a causa di errori di linearizzazione. La potenza dalla rete di trasmissione mostrata in figura varia nell'ordine dei  $10^{-2}W$ , perciò può essere considerata pressoché costante.

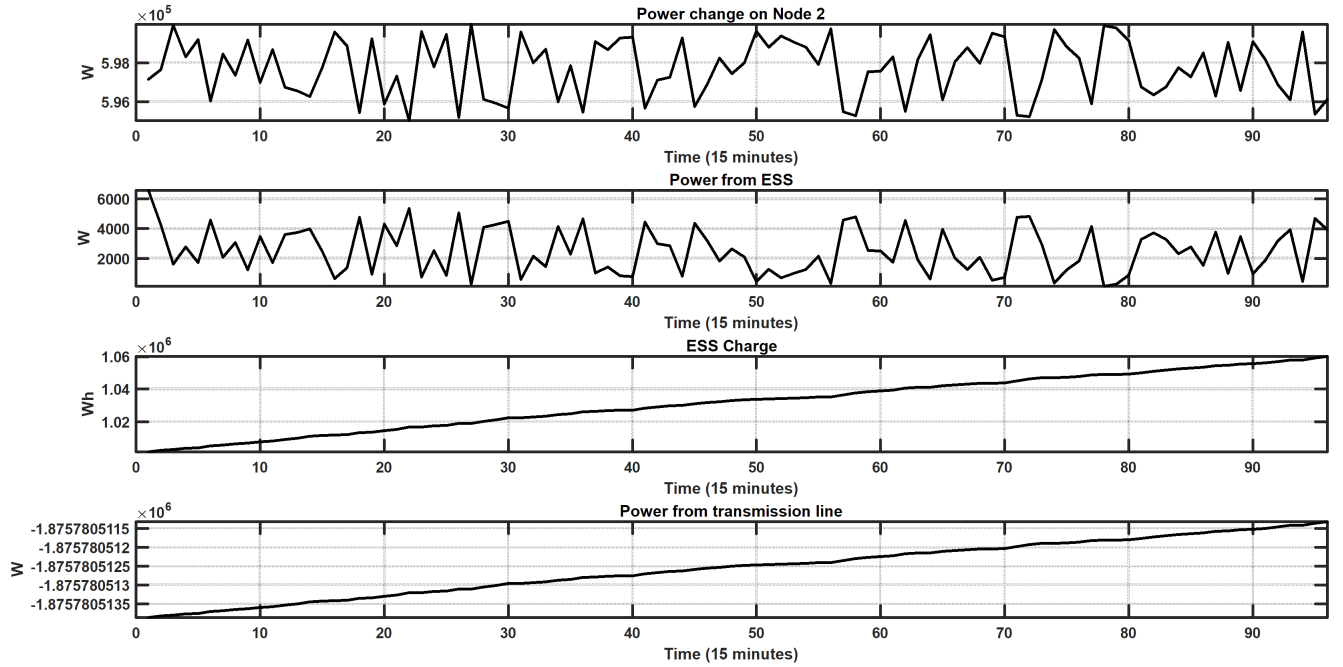


Figura 4.4: Simulazione con disturbo randomico, con ESS in cabina secondaria e con peso maggiore sulla potenza in cabina primaria

### 4.1.3 Simulazione con disturbo a campana

Questa simulazione mostra il caso verosimile di un impianto fotovoltaico connesso al nodo 2 e non previsto durante la pianificazione *day-ahead*. L'ampiezza del disturbo è ridotta rispetto alla corrispondente simulazione con l'ESS in cabina primaria, a causa della linearizzazione nel punto di lavoro determinato dalla pianificazione *day-ahead*.

In ogni caso si nota dalla figura 4.5 il profilo già visto nella simulazione in figura 3.5. Rimane il fatto che il sistema riesce a mantenere la stabilità semplice (infatti la carica dello storage riesce a tornare al riferimento, a meno di un piccolo errore, una volta terminato il disturbo a campana).

Nel caso di peso maggiore sulla potenza della rete di trasmissione, come mostrato in figura 4.6, si ottiene che la potenza in cabina primaria rimane circa uguale (con un errore di  $10^{-2}W$ ) a quella calcolata durante la pianificazione *day-ahead* e quindi l'ESS prende tutto il disturbo proveniente dal carico connesso al nodo 2. Si nota un profilo diverso rispetto a quello in figura 3.7,

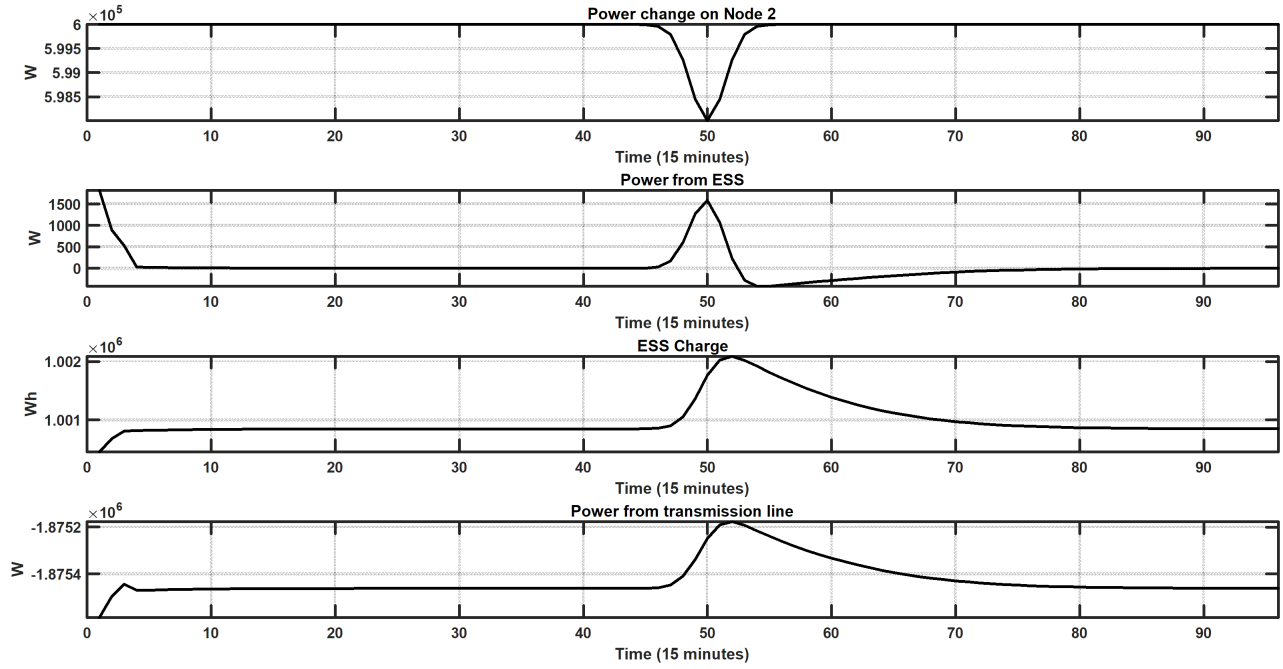


Figura 4.5: Simulazione con disturbo a campana e con ESS in cabina secondaria

a causa dell'errore introdotto durante la linearizzazione, come avviene anche nelle corrispondenti simulazioni sinusoidale e randomica.

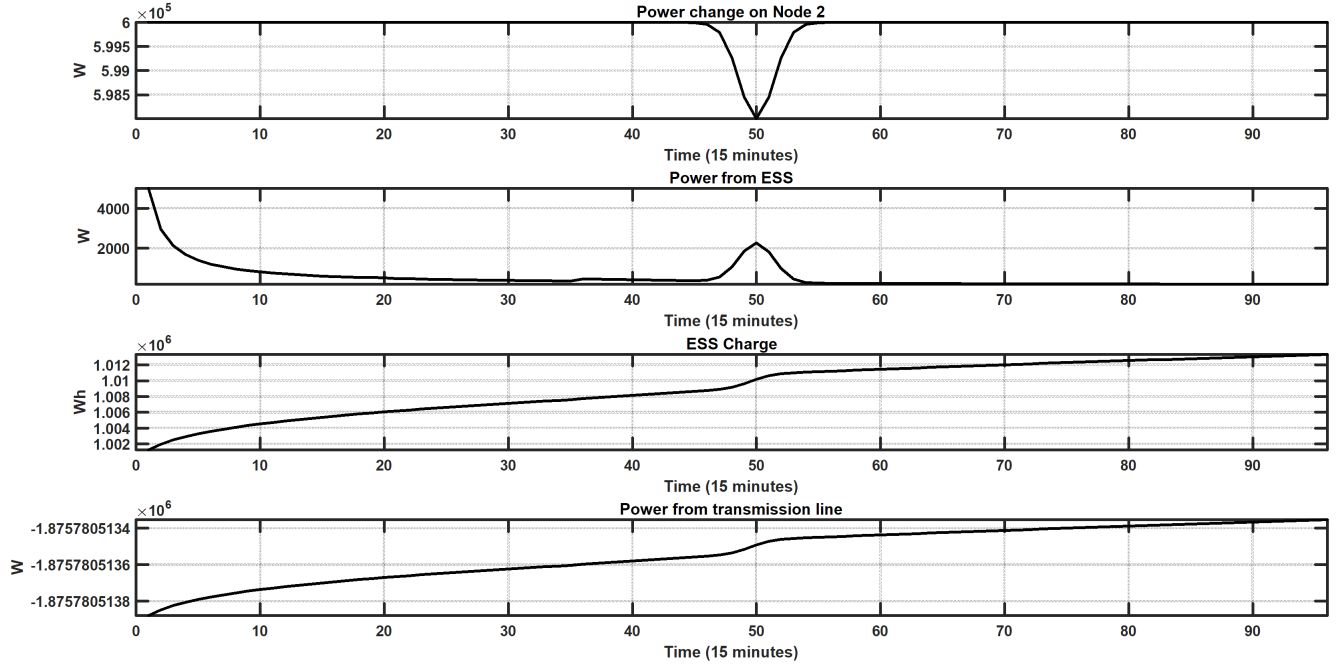


Figura 4.6: Simulazione con disturbo a campana, con ESS in cabina secondaria e con peso maggiore sulla potenza in cabina primaria

## 4.2 Simulazioni con previsioni di potenza reali

In questo capitolo si mostra come il sistema con ESS in cabina secondaria reagisce ad una situazione reale, cioè che non ha un riferimento costante per tutte le 24h ma varia in base all'ora della giornata. Infatti è possibile consultare su [11] i dati aggiornati giorno per giorno sulle potenze stimate ed effettivamente consumate in rete. Questi dati sono relativi alla rete di trasmissione italiana, sebbene siano corrispondenti a quelli tipici di una rete di distribuzione classica (opportunamente riscalati in base alla potenza dei trasformatori in cabina secondaria).

E' stato perciò possibile utilizzare questi dati come riferimento di potenza per tutti i nodi PQ (nodi 2, 3, 4) e calcolare quindi come lo storage si comporta in caso di variazioni rispetto al riferimento stabilito *day-ahead*.

### 4.2.1 Simulazione con disturbo sinusoidale

In questa simulazione è stato aggiunto al riferimento un disturbo sinusoidale di ampiezza circa uguale a quello che si avrebbe con l'imprecisione della pianificazione a causa di generatori di energie rinnovabili. E' possibile notare che lo storage si comporta in modo analogo alla simulazione in figura 4.1, poiché lo storage si occupa solo della variazione di potenza (e quindi della sinusoide in blu in figura 4.7) mentre la rete di trasmissione, come da pianificazione, si occupa di fornire potenza come nel profilo giornaliero (e quindi segue il profilo di potenza in nero in figura 4.7)

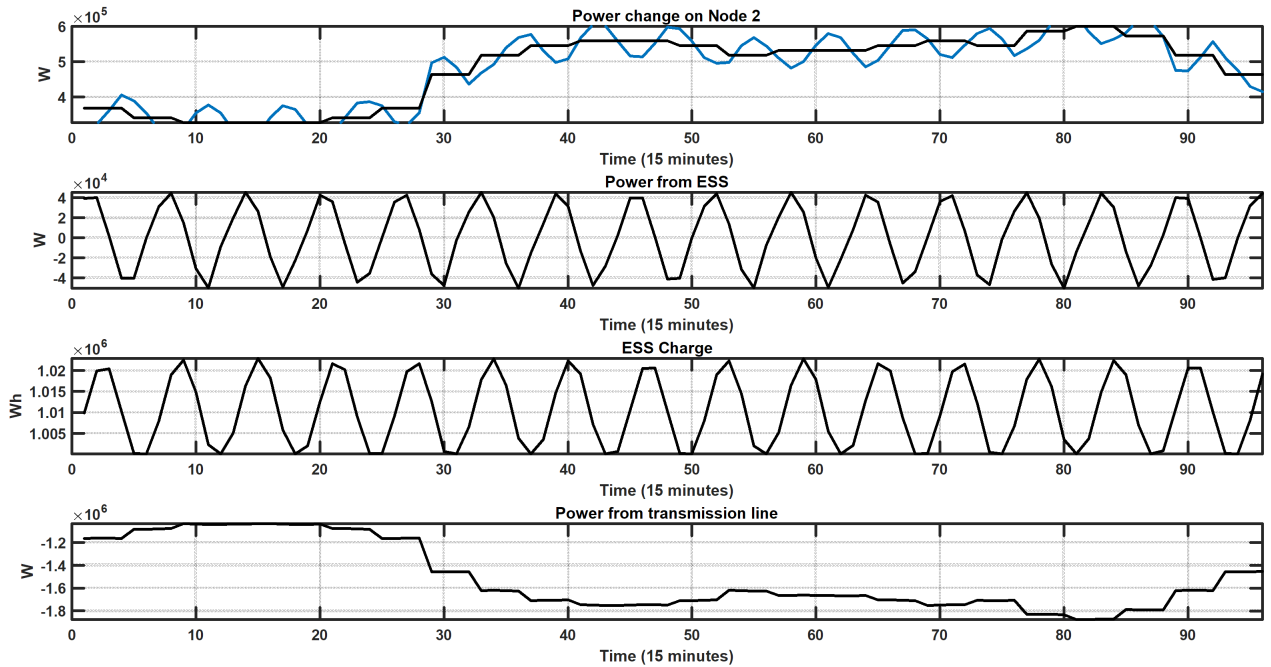


Figura 4.7: Simulazione reale con disturbo sinusoidale

Inoltre il sistema riesce a rimanere stabile e la potenza della rete di trasmissione non subisce variazioni significative rispetto alla previsione *day-ahead*.

### 4.2.2 Simulazione con disturbo randomico

In questa simulazione, che sottrae alla previsione un valore randomico di potenza (con ampiezza massima del 10% rispetto al riferimento *day-ahead*), si nota che lo storage si porta ad un livello di carica superiore rispetto a quello di riferimento. Nelle 24h riesce a rimanere comunque stabile, ma, se la simulazione procedesse oltre, si noterebbe che lo storage andrebbe in saturazione (come è possibile notare anche nella simulazione in figura 4.3).

Anche in questo caso si riesce a inseguire il riferimento *day-ahead* in cabina primaria.

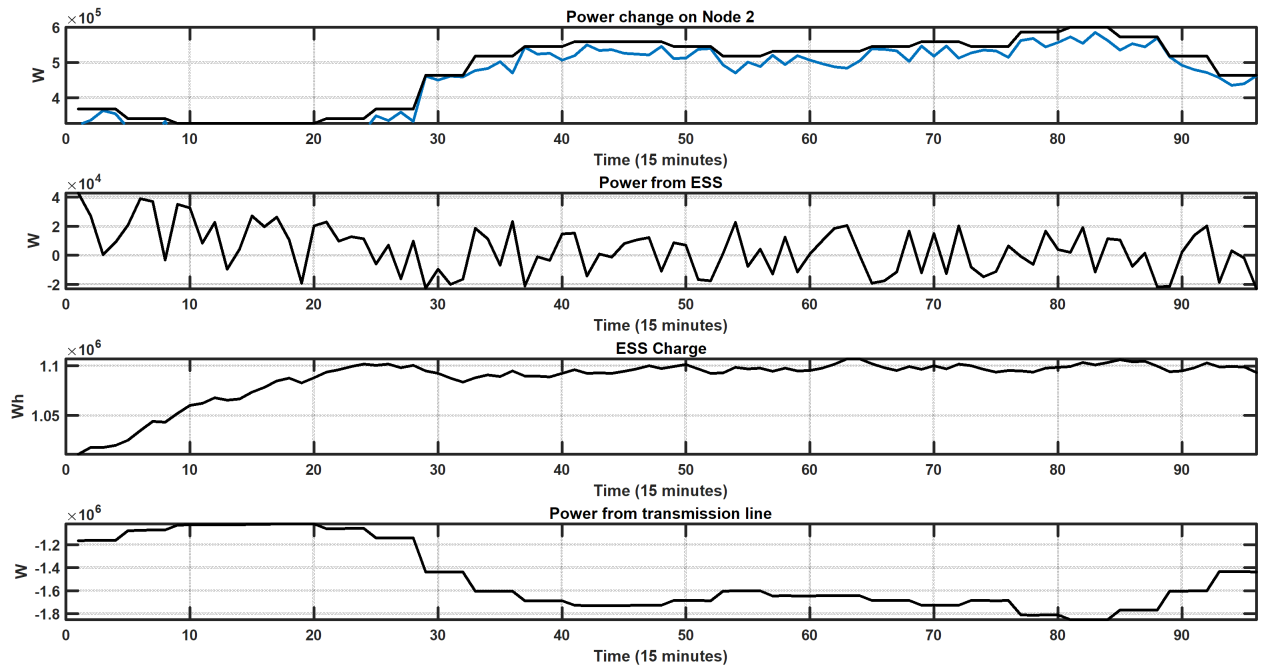


Figura 4.8: Simulazione reale con disturbo sinusoidale

### 4.2.3 Simulazione con disturbo dell'ampiezza rispetto alla previsione

In questa tipologia di simulazione si mostra un caso realistico di disturbo rispetto alla previsione, ovvero un assorbimento minore di potenza lungo



tutte le 24h. Questa tipologia di disturbo è molto comune e si verifica in caso di pianificazione errata di generazione da rinnovabili oppure in caso di pianificazione errata dei consumi degli utenti.

Tramite la simulazione in figura 4.9 si nota che lo storage è costretto a caricarsi, poiché assorbe parte della potenza che non viene effettivamente assorbita dal nodo 2. Nell'arco delle 24h il controllo non porta lo storage in saturazione, ma in caso la simulazione fosse prolungata andrebbe certamente in saturazione.

Poiché però allo scadere delle 24h viene rifatta la pianificazione per il giorno dopo, è possibile considerare lo storage come generatore per farlo tornare al riferimento entro le 24h successive. Lo stesso approccio può essere usato in caso di disturbo randomico come in figura 4.8.

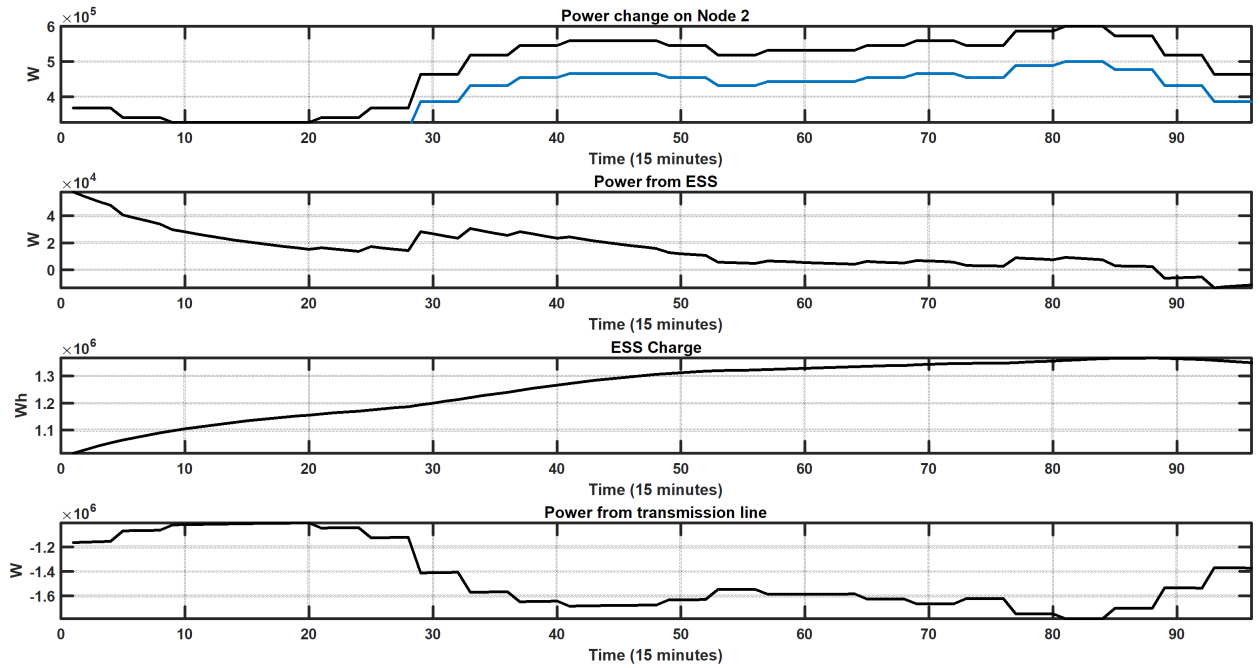


Figura 4.9: Simulazione reale con disturbo dell'ampiezza rispetto alla previsione

# Conclusioni

Da quanto si evince dalle varie simulazioni mostrate, la soluzione di adottare un ESS per stabilizzare la rete di distribuzione a fronte di imprecisioni nella pianificazione *day-ahead* risulta essere valida. Le simulazioni hanno mostrato infatti che, se il disturbo è a media nulla oppure è impulsivo, lo storage riesce a mantenere la stabilità e allo stesso tempo riesce a mitigare considerevolmente la variazione di potenza sul trasformatore in cabina primaria (punto di connessione con la rete di trasmissione). Questo risultato si ottiene sia nel caso di ESS in cabina primaria che nel caso in cui sia posto lungolinea.

Se invece il disturbo è molto ampio (confrontabile con i valori di potenza in cabina secondaria) allora il sistema può andare in instabilità, soprattutto se si ha un disturbo a media non nulla. Questo comportamento viene ancora più accentuato nel caso di peso maggiore in funzione obiettivo sul termine di potenza della rete di trasmissione.

Nel secondo caso di studio si può notare che per disturbi con ampiezza circa il 10% della potenza pianificata *day-ahead* (percentuale che rappresenta l'imprecisione delle pianificazioni *day-ahead* in caso di forte penetrazione di energie rinnovabili), il punto di lavoro (tensioni e angoli di carico) nel caso di sistema controllato si discosta da quello oggetto della linearizzazione, su cui si basa il controllo, non più dello 0.3%. Questo indica che la scelta di semplificare il problema (mantenendo quindi vincoli lineari nel problema di ottimo) è valida e permette comunque una rappresentazione adeguatamente precisa della rete.

Dalle simulazioni reali è possibile notare che la soluzione dell'ESS è ancora valida poiché riesce a mantenere la stabilità nelle 24h e consente alla rete di trasmissione di rimanere molto vicina al riferimento *day-ahead*, sia in caso di disturbi a media nulla che in caso di disturbi a media non nulla. Dall'ultima simulazione (quella che effettivamente avviene di solito sulle reti elettriche) si nota che il sistema mantiene le sue proprietà come nelle precedenti simulazioni.

# Appendice A

## Codice Matlab

### A.1 newton\_raphson.m

```
function [Res, p1, q1]=newton_raphson(Guess, G, B, P, Q, tol)
format long;
syms v1 v2 v3 v4 d1 d2 d3 d4;
temp=[Guess(1) Guess(2) Guess(3) Guess(4) Guess(5) Guess(6)
      Guess(7) Guess(8)];
%Functions definition
P2(v1, v2, v3, v4, d1, d2, d3, d4)= v2^2*G(2,2)+v2*v1*(B(2,1)
    *sin(d2-d1)+G(2,1)*cos(d2-d1))+
    v2*v3*(B(2,3)*sin(d2-d3)+G(2,3)*cos(d2-d3))+ v2*v4*(B
        (2,4)*sin(d2-d4)+G(2,4)*cos(d2-d4));
Q2(v1, v2, v3, v4, d1, d2, d3, d4)=-v2^2*B(2,2)+v2*v1*(G(2,1)
    *sin(d2-d1)-B(2,1)*cos(d2-d1))+
    v2*v3*(G(2,3)*sin(d2-d3)-B(2,3)*cos(d2-d3))+ v2*v4*(G
        (2,4)*sin(d2-d4)-B(2,4)*cos(d2-d4));
P3(v1, v2, v3, v4, d1, d2, d3, d4)= v3^2*G(3,3)+v3*v1*(B(3,1)
    *sin(d3-d1)+G(3,1)*cos(d3-d1))+
    v3*v2*(B(3,2)*sin(d3-d2)+G(3,2)*cos(d3-d2))+ v3*v4*(B
        (3,4)*sin(d3-d4)+G(3,4)*cos(d3-d4));
```

```

Q3(v1, v2, v3, v4, d1, d2, d3, d4)=-v3^2*B(3,3)+v3*v1*(G(3,1)
    *sin(d3-d1)-B(3,1)*cos(d3-d1))+
    v3*v2*(G(3,2)*sin(d3-d2)-B(3,2)*cos(d3-d2))+ v3*v4*(G
        (3,4)*sin(d3-d4)-B(3,4)*cos(d3-d4));
P4(v1, v2, v3, v4, d1, d2, d3, d4)= v4^2*G(4,4)+v4*v1*(B(4,1)
    *sin(d4-d1)+G(4,1)*cos(d4-d1))+
    v4*v2*(B(4,2)*sin(d4-d2)+G(4,2)*cos(d4-d2))+ v4*v3*(B
        (4,3)*sin(d4-d3)+G(4,3)*cos(d4-d3));
Q4(v1, v2, v3, v4, d1, d2, d3, d4)=-v4^2*B(4,4)+v4*v1*(G(4,1)
    *sin(d4-d1)-B(4,1)*cos(d4-d1))+
    v4*v2*(G(4,2)*sin(d4-d2)-B(4,2)*cos(d4-d2))+ v4*v3*(G
        (4,3)*sin(d4-d3)-B(4,3)*cos(d4-d3));

```

```

%Jacobian Matrix

```

```

J=jacobian([P2, P3, P4, Q2, Q3, Q4], [v2 v3 v4 d2 d3 d4]);

```

```

for i=1:1000
    fprintf('Iteration_%d\n', i);
    P2g=double(P2(temp(1), temp(2), temp(3), temp(4), temp(5),
        temp(6), temp(7), temp(8)));
    Q2g=double(Q2(temp(1), temp(2), temp(3), temp(4), temp(5),
        temp(6), temp(7), temp(8)));
    P3g=double(P3(temp(1), temp(2), temp(3), temp(4), temp(5),
        temp(6), temp(7), temp(8)));
    Q3g=double(Q3(temp(1), temp(2), temp(3), temp(4), temp(5),
        temp(6), temp(7), temp(8)));
    P4g=double(P4(temp(1), temp(2), temp(3), temp(4), temp(5),
        temp(6), temp(7), temp(8)));
    Q4g=double(Q4(temp(1), temp(2), temp(3), temp(4), temp(5),
        temp(6), temp(7), temp(8)));
    b=[P(2)-P2g; P(3)-P3g; P(4)-P4g; Q(2)-Q2g; Q(3)-Q3g; Q(4)-
        Q4g];

```

```

Y=double(J(temp(1), temp(2), temp(3), temp(4), temp(5),
    temp(6), temp(7), temp(8)));
%x is the increment found at the current iteration
x=Y\b;

%update temp variable with increment
temp(2)=temp(2)+x(1);
temp(3)=temp(3)+x(2);
temp(4)=temp(4)+x(3);
temp(6)=temp(6)+x(4);
temp(7)=temp(7)+x(5);
temp(8)=temp(8)+x(6);
if abs(x(1))<tol && abs(x(2))<tol && abs(x(3))<tol && abs(
    x(4))<tol && abs(x(5))<tol && abs(x(6))<tol
    break;
else
    bias=[abs(x(1)) abs(x(2)) abs(x(3)) abs(x(4)) abs(x(5))
        ) abs(x(6))];
    fprintf('Maximum bias = %f\n', max(bias));
end
end
Res=temp;
P1(v1, v2, v3, v4, d1, d2, d3, d4)=v1^2*G(1,1)+v1*v2*(B(1,2)*
    sin(d1-d2)+G(1,2)*cos(d1-d2))+
    v1*v3*(B(1,3)*sin(d1-d3)+G(1,3)*cos(d1-d3))+ v1*v4*(B
        (1,4)*sin(d1-d4)+G(1,4)*cos(d1-d4));

Q1(v1, v2, v3, v4, d1, d2, d3, d4)=-v1^2*B(1,1)+v1*v2*(G(1,2)
    *sin(d1-d2)-B(1,2)*cos(d1-d2))+
    v1*v3*(G(1,3)*sin(d1-d3)-B(1,3)*cos(d1-d3))+ v1*v4*(G
        (1,4)*sin(d1-d4)-B(1,4)*cos(d1-d4));

p1=double(P1(temp(1), temp(2), temp(3), temp(4), temp(5),

```

```
temp(6), temp(7), temp(8)));  
q1=double(Q1(temp(1), temp(2), temp(3), temp(4), temp(5),  
temp(6), temp(7), temp(8)));  
  
end
```

Questo codice itera l'algoritmo di Newton-Raphson finchè non viene raggiunta una certa tolleranza `tol` tra la soluzione al passo  $k - 1$  e quella al passo  $k$ , oppure finchè non viene raggiunto il numero massimo di iterazioni (in questo caso fissato a 1000).

L'algoritmo di Newton-Raphson è stato implementato esattamente come risulta dalla formalizzazione matematica, senza alcuna ottimizzazione. I valori sono stati convertiti a `double` poichè la struttura dati `sym` è molto lenta nei vari calcoli e aumentava esponenzialmente il tempo di calcolo della soluzione.

Questa funzione restituisce sia le variabili di rete (nel vettore `Res`) che le potenze attiva e reattiva sul nodo 1.

Nell'implementazione del metodo di Newton-Raphson sono state usate direttamente le equazioni per questa particolare tratta di rete di distribuzione, perciò non è possibile modificare la topologia di rete a meno di dover modificare la matrice delle ammettenze o le funzioni del sistema.

## A.2 optimization.m

```
function [pess, ptl] = optimization(x0, essref, ptlref,
    newpower, ess_state, Tc)
    %Optimize function for ESS Power Flow (Phase I)
    %x0 starting point, essref reference charge of ESS, ptlref
    reference active
    %power from/to trasmission line, newpower new transmission
    active power found with
    %Newton-Raphson, ess_state current energy state of ESS, Tc
    sample time (in
    %hours, usually 1/4 of hour)
    f=@(x)((newpower-x(2)-ptlref)^(2)+
        (ess_state+x(2)*Tc-essref)^2);
    A=[0 1;
        0 Tc];
    b=[2000000; 2000000-ess_state];
    Aeq=[1 1]
    beq=newpower;
    x=fmincon(f, x0, A, b, Aeq, beq);
    pess=x(2);
    ptl=x(1);
end
```

Questo codice calcola il problema di ottimizzazione descritto nel capitolo 2.2 attraverso la funzione Matlab **fmincon**.

Sono stati impostati i vincoli di potenza sul nodo 1 (la somma delle potenze attive provenienti dalla rete di trasmissione e dallo storage deve essere uguale alla potenza calcolata nel precedente problema di Power-Flow).

Nei casi in cui si è dato un peso maggiore alla potenza della linea di trasmissione rispetto che a quella dello storage, è stata modificata la funzione obiettivo nel seguente modo:

```
f=@(x)(100*(newpower-x(2)-ptlref)^(2)+
    (ess_state+x(2)*Tc-essref)^2);
```

### A.3 Sample\_ESS.m

```

clear all
close all
clc
%Variables definition
G=[2.2566      -2.2566      0      0;
   -2.2566      4.2208      -1.9643      0;
    0      -1.9643      3.9286      -1.9643;
    0      0      -1.9643      1.9643];

B=[-2      2      0      0;
    2      -2.9529      0.9535      0;
    0      0.9535      -1.907      0.9535;
    0      0      0.9535      -0.9535];

Guess=[20000      20000      20000      20000      0      0
        0      0];
P=[0 600000 620000 620000];
Q=[0 190000 50000 50000];
%First Newton_Raphson iteration
[state_variables , p1, q1]=newton_raphson(Guess , G, B, P, Q,
    0.00001);
%Now simulation of power demand change on node 2
P_new=P;
ess_state=1000000; % 50% of charge initially
Iter=1:1:97;
ChargeESS=zeros(0,97);
ChargeESS(1)=ess_state;
PTL=zeros(0,97);
PTL(1)=p1;
P_else=zeros(0,97);
P_else(1)=p1;

```



```

PESS=zeros(0,97);
P_2=zeros(0, 97);
P_2(1)=P(2);
for i=2:97
    %P_new(2)=rand()*630000;
    %P_new(2)=P(2)+300000*sin(i);
    P_new(2)=P(2)-300000*((1/sqrt(pi*2))*exp(-(((i-50)/2)^2))
        ); %Gauss distribution
    P_2(i)=P_new(2);
    fprintf('Iteration %d, new P is %f\n',i, P_new(2));
    [state_variables, p1_new, q1_new]=newton_raphson(Guess, G
        , B, P_new, Q, 0.00001);
    P_else(i)=p1_new;
    [pess, ptl]=optimization([0 0], 1000000, p1, p1_new,
        ess_state, 1/4);
    ess_state=ess_state+pess*(1/4);
    ChargeESS(i)=ess_state;
    PTL(i)=ptl;
    PESS(i)=pess;
end

figure

subplot(4,1,1)
plot(Iter, P_2, 'black', 'LineWidth',2)
xlim([0 92])
ylim([min(P_2) max(P_2)])
xlabel('Time (15 minutes)', 'FontSize', 11 , 'fontweight', 'b'
    , )
title('Power change on Node 2', 'FontSize', 11 , 'fontweight'
    , 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')

```

```

set(gcf, 'Color', 'w')
grid on

subplot(4,1,2)
plot(Iter,PESS, 'black', 'LineWidth',2)
xlim([0 92])
ylim([min(PESS) max(PESS)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' )
title('Power_from_ESS', 'FontSize', 11 , 'fontweight', 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

subplot(4,1,3)
plot(Iter,ChargeESS, 'black', 'LineWidth',2)
xlim([0 92])
ylim([min(ChargeESS) max(ChargeESS)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' )
title('ESS_Charge', 'FontSize', 11 , 'fontweight', 'b' )
ylabel('Wh', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

subplot(4,1,4)
plot(Iter,PTL, Iter, P_else,'black', 'LineWidth',2)
xlim([0 92])
ylim([min(P_else) max(P_else)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' )

```

```

title('Power from transmission line', 'FontSize', 11 , '
      fontweight', 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

```

Questo script inizializza la matrice delle ammettenze (divisa in parte reale  $G$  e parte immaginaria  $B$ ), così come le guess per la risoluzione del problema di Power-Flow tramite Newton-Raphson, le potenze attive e reattive note (relative ai nodi PQ) e la condizione iniziale e di riferimento per lo storage. Nel ciclo viene variata la potenza sul nodo 2, viene conseguentemente ricalcolato il Power-Flow e infine si calcola il nuovo profilo di potenza per lo storage e la rete di trasmissione tramite la funzione di ottimizzazione. Questo procedimento viene ripetuto 96 volte poiché il tempo di campionamento è 15 minuti e la simulazione copre le 24h.

L'ultima parte del codice stampa i vari grafici risultanti dalla simulazione.

## A.4 optimization\_secondary.m

```

function [state_variables, pess, p1, p2, p3, p4, q1, q2, q3,
        q4] = optimization_secondary( x0, essref, ptlref, newpower,
        , ess_state, prev_state, Tc, G, B)

syms v1 v2 v3 v4 d1 d2 d3 d4;
P2(v1, v2, v3, v4, d1, d2, d3, d4)= v2^2*G(2,2)+v2*v1*(B
    (2,1)*sin(d2-d1)+G(2,1)*cos(d2-d1))+...
v2*v3*(B(2,3)*sin(d2-d3)+G(2,3)*cos(d2-d3))+ v2*v4*(B
    (2,4)*sin(d2-d4)+G(2,4)*cos(d2-d4));
Q2(v1, v2, v3, v4, d1, d2, d3, d4)=-v2^2*B(2,2)+v2*v1*(G
    (2,1)*sin(d2-d1)-B(2,1)*cos(d2-d1))+...
v2*v3*(G(2,3)*sin(d2-d3)-B(2,3)*cos(d2-d3))+ v2*v4*(G
    (2,4)*sin(d2-d4)-B(2,4)*cos(d2-d4));
P3(v1, v2, v3, v4, d1, d2, d3, d4)= v3^2*G(3,3)+v3*v1*(B
    (3,1)*sin(d3-d1)+G(3,1)*cos(d3-d1))+...

```

```

v3*v2*(B(3,2)*sin(d3-d2)+G(3,2)*cos(d3-d2))+ v3*v4*(B
    (3,4)*sin(d3-d4)+G(3,4)*cos(d3-d4));
Q3(v1, v2, v3, v4, d1, d2, d3, d4)=-v3^2*B(3,3)+v3*v1*(G
    (3,1)*sin(d3-d1)-B(3,1)*cos(d3-d1))+...
v3*v2*(G(3,2)*sin(d3-d2)-B(3,2)*cos(d3-d2))+ v3*v4*(G
    (3,4)*sin(d3-d4)-B(3,4)*cos(d3-d4));
P4(v1, v2, v3, v4, d1, d2, d3, d4)= v4^2*G(4,4)+v4*v1*(B
    (4,1)*sin(d4-d1)+G(4,1)*cos(d4-d1))+...
v4*v2*(B(4,2)*sin(d4-d2)+G(4,2)*cos(d4-d2))+ v4*v3*(B
    (4,3)*sin(d4-d3)+G(4,3)*cos(d4-d3));
Q4(v1, v2, v3, v4, d1, d2, d3, d4)=-v4^2*B(4,4)+v4*v1*(G
    (4,1)*sin(d4-d1)-B(4,1)*cos(d4-d1))+...
v4*v2*(G(4,2)*sin(d4-d2)-B(4,2)*cos(d4-d2))+ v4*v3*(G
    (4,3)*sin(d4-d3)-B(4,3)*cos(d4-d3));

%Jacobian Matrix
Y=jacobian([P2, P3, P4, Q2, Q3, Q4], [v2 v3 v4 d2 d3 d4])
    ;

P1(v1, v2, v3, v4, d1, d2, d3, d4)=v1^2*G(1,1)+v1*v2*(B
    (1,2)*sin(d1-d2)+G(1,2)*cos(d1-d2))+...
v1*v3*(B(1,3)*sin(d1-d3)+G(1,3)*cos(d1-d3))+ v1*v4*(B
    (1,4)*sin(d1-d4)+G(1,4)*cos(d1-d4));
Q1(v1, v2, v3, v4, d1, d2, d3, d4)=-v1^2*B(1,1)+v1*v2*(G
    (1,2)*sin(d1-d2)-B(1,2)*cos(d1-d2))+...
v1*v3*(G(1,3)*sin(d1-d3)-B(1,3)*cos(d1-d3))+ v1*v4*(G
    (1,4)*sin(d1-d4)-B(1,4)*cos(d1-d4));

Y1=jacobian([P1, Q1], [v2 v3 v4 d2 d3 d4]);

p2=double(P2(prev_state(1), prev_state(2), prev_state(3),
    prev_state(4), prev_state(5), prev_state(6),
    prev_state(7), prev_state(8)));

```

```

J=double(Y(prev_state(1), prev_state(2), prev_state(3),
    prev_state(4), prev_state(5), prev_state(6),
    prev_state(7), prev_state(8)));
J1=double(Y1(prev_state(1), prev_state(2), prev_state(3),
    prev_state(4), prev_state(5), prev_state(6),
    prev_state(7), prev_state(8)));

fprintf('Power_on_node2: %f\n', p2);
fprintf('New_Power_on_node2: %f\n', newpower(2));
f=@(x)(10000000*(x(7)-ptlref)^2+((ess_state+(x(8)*Tc))-
    essref)^2);

% x(1)=V2-prev_state(2) x(2)=V3-prev_state(3) x(3)=V4-
    prev_state(4)
% x(4)=d2-prev_state(6) x(5)=d3-prev_state(7) x(6)=d4-
    prev_state(8)
%prev_state=[v1, v2, v3, v4, d1, d2, d3, d4] is
    prev_state model

Aeq=[J(1,1)  J(1,2) J(1,3)  J(1,4) J(1,5)  J(1,6) 0 0
      -1 ;
J(2,1)  J(2,2) J(2,3)  J(2,4) J(2,5)  J(2,6) 0 0
;
J(3,1)  J(3,2) J(3,3)  J(3,4) J(3,5)  J(3,6) 0 0
;
J(4,1)  J(4,2) J(4,3)  J(4,4) J(4,5)  J(4,6) 0 0
;
J(5,1)  J(5,2) J(5,3)  J(5,4) J(5,5)  J(5,6) 0 0
;
J(6,1)  J(6,2) J(6,3)  J(6,4) J(6,5)  J(6,6) 0 0
;
J1(1,1) J1(1,2) J1(1,3) J1(1,4) J1(1,5) J1(1,6) -1 0
;

```

```

J1(2,1) J1(2,2) J1(2,3) J1(2,4) J1(2,5) J1(2,6) 0 0
];

beq=[newpower(2)-p2;
    newpower(3)-double(P3(prev_state(1), prev_state(2),
        prev_state(3), prev_state(4), prev_state(5),
        prev_state(6), prev_state(7), prev_state(8)));
    newpower(4)-double(P4(prev_state(1), prev_state(2),
        prev_state(3), prev_state(4), prev_state(5),
        prev_state(6), prev_state(7), prev_state(8)));
    newpower(6)-double(Q2(prev_state(1), prev_state(2),
        prev_state(3), prev_state(4), prev_state(5),
        prev_state(6), prev_state(7), prev_state(8)));
    newpower(7)-double(Q3(prev_state(1), prev_state(2),
        prev_state(3), prev_state(4), prev_state(5),
        prev_state(6), prev_state(7), prev_state(8)));
    newpower(8)-double(Q4(prev_state(1), prev_state(2),
        prev_state(3), prev_state(4), prev_state(5),
        prev_state(6), prev_state(7), prev_state(8)));
    -double(P1(prev_state(1), prev_state(2), prev_state
        (3), prev_state(4), prev_state(5), prev_state(6),
        prev_state(7), prev_state(8)));
    newpower(5)-double(Q1(prev_state(1), prev_state(2),
        prev_state(3), prev_state(4), prev_state(5),
        prev_state(6), prev_state(7), prev_state(8)))]];

x=fmincon(f, x0, Aeq, beq);
%x is the new state_variables vector (but x(7) is P1 and
    x(8) is Pess)

state_variables=[prev_state(1) x(1)+prev_state(2) x(2)+
    prev_state(3) x(3)+prev_state(4) prev_state(5) x(4)+

```

```

        prev_state(6) x(5)+prev_state(7) x(6)+prev_state(8)];
    pessi=x(8);
    p1=x(7);
    p2=double(P2(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
    p3=double(P3(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
    p4=double(P4(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
    q1=double(Q1(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
    q2=double(Q2(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
    q3=double(Q3(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
    q4=double(Q4(state_variables(1), state_variables(2),
        state_variables(3), state_variables(4),
        state_variables(5), state_variables(6),
        state_variables(7), state_variables(8)));
end

```

Questa seconda funzione di ottimizzazione si occupa di trovare la riparti-

zione ottimale del disturbo di potenza tra storage in cabina secondaria e rete di trasmissione. E' stato necessario inserire tutte le equazioni del Power-Flow come già fatto nella sezione A.1 per poter calcolare il punto di lavoro una volta linearizzato il problema.

Questa funzione utilizza la funzione Matlab `fmincon` per il calcolo del punto di lavoro ottimo. Inoltre ritorna tutte le variabili di stato, insieme alle potenze sul nodo 1 (potenza erogata dalla rete di trasmissione) e sullo storage.

## A.5 Secondary\_ESS.m

```
clear all
close all
clc

G=[2.2566      -2.2566      0      0;
   -2.2566      4.2208     -1.9643      0;
    0     -1.9643      3.9286     -1.9643;
    0      0     -1.9643      1.9643];

B=[-2      2      0      0;
    2     -2.9529      0.9535      0;
    0      0.9535     -1.907      0.9535;
    0      0      0.9535     -0.9535];

Guess=[20000      20000      20000      20000      0      0
        0      0];

P=[0 600000 620000 620000];
Q=[0 190000 50000 50000];

V_base=20000;
S_base=630000;
Y_base=S_base/(V_base^2);
```



```

[state_variables, p1, q1]=newton_raphson(Guess./V_base, G./
    Y_base, B./Y_base, P./S_base, Q./S_base, 0.000001);

fprintf('P1:_%f\nQ1:_%f\nV1:_%f\nV2:_%f\nV3:_%f\nV4:_%f\nD1:_%f\nD2:_%f\nD3:_%f\nD4:_%f\n', p1*S_base, q1*S_base,
    state_variables(1)*V_base, state_variables(2)*V_base,
    state_variables(3)*V_base, state_variables(4)*V_base,
    state_variables(5), state_variables(6), state_variables(7)
    , state_variables(8));

P(1)=p1*S_base;
Q(1)=q1*S_base;
%No changes on Node 2
ess_state=1000000;
essref=1000000;
ptlref=p1*S_base;
x0=[state_variables(2) state_variables(3) state_variables(4)
    state_variables(6) state_variables(7) state_variables(8) 0
    0];
newpower=[P Q];
Tc=1/4;

num=96;

Iter=1:1:num;
ChargeESS=zeros(1,num);
ChargeESS(1)=ess_state;
PTL=zeros(1,num);
PTL(1)=p1;
PESS=zeros(1,num);
P_2=zeros(1, num);

```

```

P_2(1)=P(2);

for i=1:num
fprintf('Iteration_□d\n\n', i);
%newpower(2)=P(2)-5000*sin(i);
newpower(2)=P(2)-5000*((1/sqrt(pi*2))*exp(-(((i-50)/2)^2)));
%newpower(2)=P(2)-5000*rand();
P_2(i)=newpower(2);
[state_variables, pess, p1, p2, p3, p4, q1, q2, q3, q4]=
    optimization_secondary(x0, essref/S_base, ptlref/S_base,
    newpower./S_base, ess_state/S_base, state_variables, Tc, G
    ./Y_base, B./Y_base);
fprintf('\n\nNEW\nV1:□f\nV2:□f\nV3:□f\nV4:□f\nD1:□f\n
    nD2:□f\nD3:□f\nD4:□f\n', state_variables(1)*V_base,
    state_variables(2)*V_base, state_variables(3)*V_base,
    state_variables(4)*V_base, state_variables(5),
    state_variables(6), state_variables(7), state_variables(8)
    );
fprintf('PESS:□f\nP1:□f\nP2:□f\nP3:□f\nP4:□f\nQ1:□f\nQ2
    :□f\nQ3:□f\nQ4:□f\n', pess*S_base, p1*S_base, p2*S_base,
    p3*S_base, p4*S_base, q1*S_base, q2*S_base, q3*S_base, q4
    *S_base);

ess_state=ess_state+pess*S_base*Tc;
ChargeESS(i)=ess_state;
PTL(i)=p1*S_base;
PESS(i)=pess*S_base;
end;

```

```
figure
```

```
subplot(4,1,1)
plot(Iter, P_2, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(P_2) max(P_2)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' ')
title('Power_change_on_Node_2', 'FontSize', 11 , 'fontweight'
      , 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on
```

```
subplot(4,1,2)
plot(Iter,PESS, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(PESS) max(PESS)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' ')
title('Power_from_ESS', 'FontSize', 11 , 'fontweight', 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on
```

```
subplot(4,1,3)
plot(Iter,ChargeESS, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(ChargeESS) max(ChargeESS)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' ')
```

```

    ' )

title('ESS_Charge', 'FontSize', 11 , 'fontweight', 'b' )
ylabel('Wh', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

subplot(4,1,4)
plot(Iter,PTL, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(PTL) max(PTL)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
    ' )
title('Power_from_transmission_line', 'FontSize', 11 , '
    fontweight', 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

```

Questo script esegue le simulazioni sulla rete in caso lo storage si trovi in cabina secondaria.

Inizialmente definisce tutte le variabili che rappresentano la rete (matrici B e G, potenze dei nodi PQ, guess iniziali, stato iniziale e riferimento di carica dell'ESS) e calcola il punto di lavoro *day-ahead* della rete (tramite l'algoritmo di Newton-Raphson descritto in sezione A.1). Si nota che è stato necessario porre tutte le condizioni iniziali e i dati del problema in *per-unit*, altrimenti in simulazione ci sarebbero stati errori di calcolo dovuti alla precisione limitata della virgola mobile.

Nel ciclo viene modificata la potenza richiesta dal carico connesso al nodo 2 e si calcola il punto di lavoro ottimo in seguito al disturbo, utilizzando la funzione descritta nella sezione A.4.

Le ultime istruzioni stampano il grafico risultante dalla simulazione effettuata.

## A.6 Real\_ESS.m

```

clear all
close all
clc

G=[2.2566      -2.2566      0      0;
   -2.2566      4.2208      -1.9643      0;
    0      -1.9643      3.9286      -1.9643;
    0      0      -1.9643      1.9643];

B=[-2      2      0      0;
    2      -2.9529      0.9535      0;
    0      0.9535      -1.907      0.9535;
    0      0      0.9535      -0.9535];

Guess=[20000      20000      20000      20000      0      0
        0      0];

A=[27  27  27  27  25  25  25  25  24  24  24  24  24  24  24
   24  24  24  24  24  25  25  25  25  27  27  27  27  34  34
   34  34  38  38  38  38  40  40  40  40  41  41  41  41
  41  41  41  41  40  40  40  40  38  38  38  38  39  39  39
  39  39  39  39  39  40  40  40  40  41  41  41  41  40  40
  40  40  43  43  43  43  44  44  44  44  42  42  42  42  38
   38  38  38  34  34  34  34];

P2=A.*(600000/44);
P3=A.*(620000/44);
P4=A.*(620000/44);
Q2=A.*(190000/44);

```

```

Q3=A.*(50000/44);
Q4=A.*(50000/44);

num=96;

P1=zeros(1, num);
Q1=zeros(1, num);

V_base=20000;
S_base=630000;
Y_base=S_base/(V_base^2);

ess_state=1000000;
essref=1000000;
Tc=1/4;

Iter=1:1:num;
ChargeESS=zeros(1,num);
ChargeESS(1)=ess_state;
PTL=zeros(1,num);

PESS=zeros(1,num);
P_2=zeros(1, num);

for i=1:num
    P=[P1(i) P2(i) P3(i) P4(i)];
    Q=[Q1(i) Q2(i) Q3(i) Q4(i)];
    [state_variables, p1, q1]=newton_raphson(Guess./
        V_base, G./Y_base, B./Y_base, P./S_base, Q./S_base
        , 0.000001);

```

```

P1(i)=p1*S_base;
Q1(i)=q1*S_base;

ptlref=P1(i);

x0=[state_variables(2) state_variables(3)
     state_variables(4) state_variables(6)
     state_variables(7) state_variables(8) 0 0];
newpower=[P Q];

fprintf('Iteration_%d\n\n', i);
%newpower(2)=P(2)-50000*sin(i);
%newpower(2)=P(2)-5000*((1/sqrt(pi*2))*exp(-(((i-50)
    /2)^2)));
%newpower(2)=P(2)-50000*rand();
%newpower(2)=P(2);
newpower(2)=A(i)*(500000/44);
P_2(i)=newpower(2);
[state_variables, pess, p1, p2, p3, p4, q1, q2, q3,
    q4]=optimization_secondary(x0, essref/S_base,
    ptlref/S_base, newpower./S_base, ess_state/S_base,
    state_variables, Tc, G./Y_base, B./Y_base);
fprintf('\n\n\nNEW\nV1:_%f\nV2:_%f\nV3:_%f\nV4:_%f\n
    nD1:_%f\nD2:_%f\nD3:_%f\nD4:_%f\n',
    state_variables(1)*V_base, state_variables(2)*
    V_base, state_variables(3)*V_base, state_variables
    (4)*V_base, state_variables
    (5), state_variables(6), state_variables(7),
    state_variables(8));

fprintf('PESS:_%f\nP1:_%f\nP2:_%f\nP3:_%f\nP4:_%f\nQ1
    :_%f\nQ2:_%f\nQ3:_%f\nQ4:_%f\n',pess*S_base, p1*

```

```

        S_base, p2*S_base, p3*S_base, p4*S_base, q1*S_base
        , q2*S_base, q3*S_base, q4*S_base);

    ess_state=ess_state+pess*S_base*Tc;
    ChargeESS(i)=ess_state;
    PTL(i)=p1*S_base;
    PESS(i)=pess*S_base;
end;

figure

subplot(4,1,1)
plot(Iter, P_2, Iter, P2, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(P2) max(P2)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
    ' )
title('Power_change_on_Node_2', 'FontSize', 11 , 'fontweight'
    , 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

subplot(4,1,2)
plot(Iter,PESS, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(PESS) max(PESS)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
    ' )
title('Power_from_ESS', 'FontSize', 11 , 'fontweight', 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')

```



```

set(gcf, 'Color', 'w')
grid on

subplot(4,1,3)
plot(Iter,ChargeESS, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(ChargeESS) max(ChargeESS)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' )
title('ESS_Charge', 'FontSize', 11 , 'fontweight', 'b' )
ylabel('Wh', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

subplot(4,1,4)
plot(Iter,PTL, 'black', 'LineWidth',2)
xlim([0 num])
ylim([min(PTL) max(PTL)])
xlabel('Time_(15_minutes)', 'FontSize', 11 , 'fontweight', 'b'
      ' )
title('Power_from_transmission_line', 'FontSize', 11 , '
      fontweight', 'b' )
ylabel('W', 'FontSize', 11 , 'fontweight', 'b' )
set(gca, 'LineWidth', 2, 'FontSize', 11, 'fontweight', 'b')
set(gcf, 'Color', 'w')
grid on

```

# Bibliografia

- [1] A. D. Giorgio, F. Liberati, A. Lanna, A. Pietrabissa, and F. D. Priscoli, “Model predictive control of energy storage systems for power tracking and shaving in distribution grids,” *IEEE Transactions on Sustainable Energy*, vol. 8, pp. 496–504, April 2017.
- [2] J. Varela, L. J. Puglisi, T. Wiedemann, U. Ysberg, D. Stein, Z. Pokorna, C. Arnoult, R. Garaud-Verdier, and L. Consiglio, “Show me!: Large-scale smart grid demonstrations for european distribution networks,” *IEEE Power and Energy Magazine*, vol. 13, pp. 84–91, Jan 2015.
- [3] P. F. Ribeiro, B. K. Johnson, M. L. Crow, A. Arsoy, and Y. Liu, “Energy storage systems for advanced power applications,” *Proceedings of the IEEE*, vol. 89, pp. 1744–1756, Dec 2001.
- [4] H. Zhou, T. Bhattacharya, D. Tran, T. S. T. Siew, and A. M. Khambadkone, “Composite energy storage system involving battery and ultra-capacitor with dynamic energy management in microgrid applications,” *IEEE Transactions on Power Electronics*, vol. 26, pp. 923–930, March 2011.
- [5] M. T. Zareifard and A. V. Savkin, “Model predictive control for wind power generation smoothing with controlled battery storage based on a nonlinear battery mathematical model,” in *2015 10th Asian Control Conference (ASCC)*, pp. 1–6, May 2015.
- [6] A. D. Giorgio, F. Liberati, and A. Lanna, “Electric energy storage systems integration in distribution grids,” in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, pp. 1279–1284, June 2015.
- [7] F. Luo, K. Meng, Z. Y. Dong, Y. Zheng, Y. Chen, and K. P. Wong, “Coordinated operational planning for wind farm with battery ener-

- gy storage system,” *IEEE Transactions on Sustainable Energy*, vol. 6, pp. 253–262, Jan 2015.
- [8] K. M. Chandy, S. H. Low, U. Topcu, and H. Xu, “A simple optimal power flow model with energy storage,” in *49th IEEE Conference on Decision and Control (CDC)*, pp. 1051–1057, Dec 2010.
- [9] D. Gayme and U. Topcu, “Optimal power flow with large-scale storage integration,” *IEEE Transactions on Power Systems*, vol. 28, pp. 709–717, May 2013.
- [10] K. Worthmann, C. M. Kellett, P. Braun, L. Grüne, and S. R. Weller, “Distributed and decentralized control of residential energy systems incorporating battery storage,” *IEEE Transactions on Smart Grid*, vol. 6, pp. 1914–1923, July 2015.
- [11] “Consumi giornalieri stimati e reali raccolti da terna s.p.a, gestore della rete di trasmissione italiana.” <https://www.terna.it/it-it/sistemaelettrico/dispacciamento/datiesercizio/datigiornalieri.aspx>. Accessed: 2017-07-04.