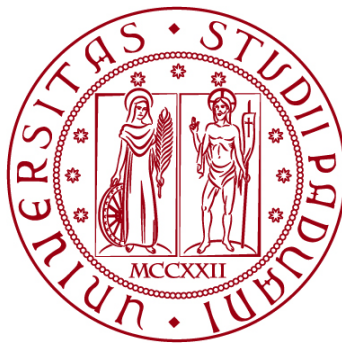


UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE DII
Department Of Industrial Engineering DII

Corso di Laurea Magistrale in Ingegneria dell'energia Elettrica



Probabilistic Power Flow con Algoritmo Patavino

Relatore: Prof. Roberto Benato

Correlatore: Ing. Giovanni Gardan

Laureando: Niccolò Osti

Matricola: 202689

*“Colui che non sa niente, non ama niente.
Colui che non fa niente, non capisce niente.
Colui che non capisce niente è spregevole.
Ma colui che capisce, ama, vede, osserva...
La maggiore conoscenza
È congiunta indissolubile all’amore...
Chiunque creda che tutti i frutti
maturino contemporaneamente come le fragole,
non sa nulla dell’uva.”*

Paracelso

Sommario

Introduzione	1
1. Lo studio dei flussi di potenza nelle reti trifase simmetriche ed equilibrate	2
1.1 Introduzione.....	2
1.2 Metodo Patavino per lo studio dei flussi di Potenza.....	3
1.2.1 Algoritmo iterativo per la ricerca della soluzione	6
1.2.2 Metodo FCC.....	6
1.3 Inclusione delle linee <i>HVDC VSC/LCC</i> e Metodo <i>IRPE</i> per la stima di Potenza Reattiva...	9
1.3.1 Modellizzazione linee HVDC VSC.....	9
1.3.2 Modellizzazione linee HVDC LCC.....	11
1.4 Implementazione MatLab del metodo <i>IRPE</i>	14
1.4.1 Dati in input.....	14
1.4.2 Corpo del codice.....	18
2. Probabilistic Power Flow	23
2.1 La madre dello studio Probabilistico	23
2.2 Metodi per il Probabilistic Power Flow	25
2.2.1 Metodi Numerici	26
2.2.2 Metodi Analitici.....	28
2.2.3 Metodi ad approssimazione	30
3. Statistica e Probabilità applicata allo studio dei flussi di potenza probabilistici	31
3.1 Strumenti matematici per la statistica	31
3.1.1 Media.....	31
3.1.2 Scarto quadratico medio	32
3.1.3 Varianza.....	33
3.1.4 Campo di variazione.....	33
3.2 Funzioni di distribuzione di probabilità:	34
3.2.1 Normale	35
3.2.2 LogNormale.....	36
3.2.3 Weibull.....	38
3.2.4 Distribuzione Beta	40
3.3. Calcolo della probabilità a partire da una <i>Pdf</i> o <i>Cdf</i>	42
4. Implementazione del Probabilistic Power Flow al calcolatore.....	43
4.1 Implementazione del Probabilistic Power Flow su DigSilent.....	43
4.1.1 Analisi Probabilistica.....	43
4.1.2 Metodi per l'analisi probabilistica	44
4.1.3 Metodo di Monte Carlo (<i>MCM</i>)	44
4.1.4 Metodo Quasi-Monte Carlo (<i>QCM</i>)	45
4.1.5 Metodi a confronto	45

4.1.6 Modellizzazione dei nodi di generazione distribuita	47
4.1.7 Modellizzazione degli aerogeneratori.....	47
4.1.8 Modellizzazione degli impianti fotovoltaici	50
4.1.9 Generazione e assegnazione di distribuzioni di probabilità a partire da dati reali per la generazione da fonte rinnovabile su DigSilent.	53
4.1.10 Lancio del Probabilistic Power Flow con Metodo di Montecarlo	58
4.2 Implementazione del Probabilistic Power Flow su MatLab	62
4.2.1 Generazione e assegnazione di distribuzioni di probabilità a partire da dati reali per la generazione da fonte rinnovabile.....	63
4.2.2 Creazione delle Matrici <i>CAR_GEN</i> per ogni iterazione del <i>PFPF_PRB</i>	65
4.2.3 Calcolo preliminare del Probabilistic Power Flow	66
4.2.4 Ciclo <i>parfor (1)</i>	69
4.2.5 Situazioni patologiche di non convergenza della rete.....	73
4.3 Ottimizzazione dell'Algoritmo	75
4.3.1 Parallel Computing Toolbox	76
4.4 Flow chart dell'algoritmo <i>PFPD_PRB</i>	77
5. Confronto dei risultati tra <i>PFPD_PRB</i> e <i>DigSilent</i>	78
5.1. Rete Test AC/DC 22 nodi con collegamento <i>HVDC-VSC bipolare</i>	80
5.1.1 Simulazione della rete test AC/DC 22 nodi con collegamento <i>HVDC-VSC bipolare</i> considerando 1000 iterazioni <i>MC</i>	82
6. Applicazione del <i>PFPD_PRB</i> su reti con presenza di generazione distribuita	84
6.1 Rete 41 nodi <i>IEEE</i> modificata	84
6.1.1 Studio dell'Ampacity delle linee elettriche	92
6.1.2 Studio del superamento della corrente nominale dei trasformatori	98
6.2 Rete Siciliana con presenza di generazione distribuita	102
Conclusione.....	108
Bibliografia	109
Appendice	111
Guida alla creazione del foglio <i>Excel</i> per l'importazione dei dati da <i>DigSilent</i> a <i>MatLab</i>	111
<i>A_INTERFACCIA_DS_MT.m</i>	115
<i>A_INTERFACCIA_IRPE_PRB.m</i>	120
<i>Power_curve_wind.m</i>	124
<i>Power_solar.m</i>	124
Ringraziamenti	Errore. Il segnalibro non è definito.

Introduzione

Negli ultimi anni, l'aumento dell'utilizzo di fonti energetiche rinnovabili per la produzione di energia elettrica ha portato alla necessità di nuovi strumenti e metodi di analisi per studiare il comportamento delle reti elettriche sotto condizioni di incertezza e variabilità. In questo contesto, il *Probabilistic Power Flow (PPF)* rappresenta una soluzione efficace per gestire la presenza di variabili aleatorie che influenzano il funzionamento delle reti di trasmissione. Nell'elaborato verrà illustrato l'algoritmo *PFPD_PRB*, una soluzione innovativa per analizzare i flussi di potenza utilizzando un approccio probabilistico. Grazie all'unione dell'algoritmo *PFPD* (sviluppato presso l'università di Padova) e delle tecniche di analisi probabilistica descritte in letteratura, messe a punto fin dal 1974, è stato possibile svilupparlo in ambiente *MatLab*. L'applicazione dell'algoritmo al campo delle fonti energetiche rinnovabili permette di valutare il loro impatto sull'integrità della rete elettrica e di ottimizzare la loro gestione in modo da ridurre le perdite di potenza e migliorare la sicurezza del sistema. Inoltre, l'utilizzo del *PPF* rappresenta un importante strumento di supporto decisionale per i gestori di rete, in quanto consente di prevedere l'andamento del sistema in condizioni di instabilità, di fornire un importante strumento decisionale per eventuali investimenti e intervenire tempestivamente per evitare eventuali blackout. La tesi si focalizzerà su una serie di casi di studio riguardanti reti elettriche in cui sono presenti fonti rinnovabili di energia, allo scopo di verificare l'effettiva applicabilità dell'algoritmo proposto e valutare le prestazioni rispetto ad altre tecniche di analisi. Sarà dedicata particolare attenzione alla verifica della correttezza dei risultati ottenuti e alla loro validità in situazioni reali, allo scopo di fornire un valido contributo alla ricerca nel campo dell'energia elettrica.

1. Lo studio dei flussi di potenza nelle reti trifase simmetriche ed equilibrate

1.1 Introduzione

Con il crescente sviluppo dell'infrastruttura elettrica e l'incessante progresso di tecnologie che ne permettono l'esercizio, lo studio delle varie grandezze in essa presenti diventano sempre più fondamentali e complicate, per tale ragione lo studio dei flussi di potenza sta diventando un cavallo di battaglia della ricerca elettrotecnica sempre più prorompente. Tra le valutazioni ottenibili attraverso questo studio troviamo:

- Livelli di tensione che si stabiliscono nella rete;
- Studio dello stress sui vari componenti che costituiscono la rete stessa;
- Considerazioni sulle condizioni di esercizio migliori per una data struttura;
- Analisi dei punti deboli della rete e valutazione di eventuali soluzioni per il rafforzamento;

A partire dai quattro punti sopra esposti è deducibile che lo studio dei flussi di potenza è fondamentale per un corretto esercizio e controllo della rete elettrica, ma anche per effettuare future previsioni in materia di ampliamento e aggiornamento delle attuali infrastrutture, il quale risulta essere necessario in vista della sempre più elevata penetrazione di energie rinnovabili e rimarcata aleatorietà dei carichi presenti nella rete. Al di là di una componente fisica analitica tali studi intercorrono anche in altri ambiti:

- Economico, attraverso pianificazioni strategiche di dispacciamento dell'energia elettrica;
- Politico, studio dell'acquisizione di energia elettrica da parte di nazioni estere e della possibilità dell'autoproduzione e consumo;
- Ambientale, riduzione delle perdite lungo la rete elettrica e sviluppo di centrali di generazione sostenibili, solare, idroelettrico, eolico ecc...;

1.2 Metodo Patavino per lo studio dei flussi di Potenza

Se la rete elettrica trifase che prendiamo in considerazione è di una struttura che possiamo ritenere simmetrica (nelle tensioni) ed è interessata da un normale funzionamento che ammettiamo equilibrato (nelle correnti), possiamo ricondurci allo studio del regime permanente della corrispondente rete monofase alla sequenza diretta [8]. Tale rete è costituita da nodi e rami, in particolar modo i nodi possono essere di tre diverse tipologie:

- Nodi PV;
- Nodi PQ e/o di transito;
- Nodo di Slack;

tale suddivisione è fondamentale per l'impostazione dei vari vincoli di esercizio all'interno della rete:

- \bar{u}_a , fasore di tensione per il nodo di Slack;
- $\bar{u}_b \dots \bar{u}_g$, modulo delle tensioni per i generatori;
- $\bar{p}_b \dots \bar{p}_g$, potenze attive erogate dai generatori;
- $\bar{p}_h + j\bar{q}_h \dots \bar{p}_l + j\bar{q}_l$, potenze complesse assorbite dai carichi;

la rete elettrica, tipologicamente e topologicamente parlando può essere rappresentata attraverso lo schema seguente:

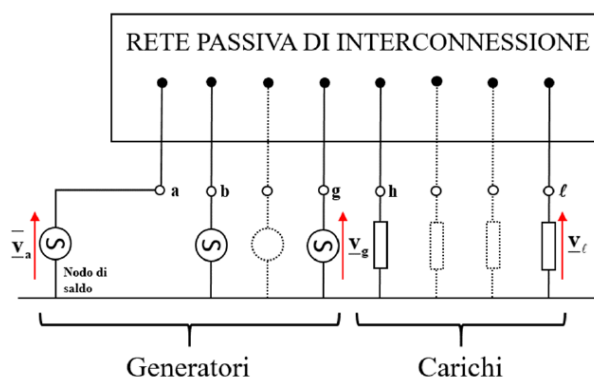


Figura 1 Rappresentazione schematica della rete elettrica italiana attraverso la rete passiva e gli elementi attivi della rete. [10]

Rete Passiva (\underline{Y}_n) e Attiva (\underline{Y}_{sh}), vengono rappresentate entrambe come ammettenze, calcolate una sola volta durante il Power Flow e con espressioni fisiche differenti. Vediamole di seguito:

- \underline{Y}_n , racchiude le informazioni topologiche della rete, ovvero come i vari nodi sono interconnessi tra di loro, a partire dalla rappresentazione a Π degli elementi della rete come rappresentato in figura.

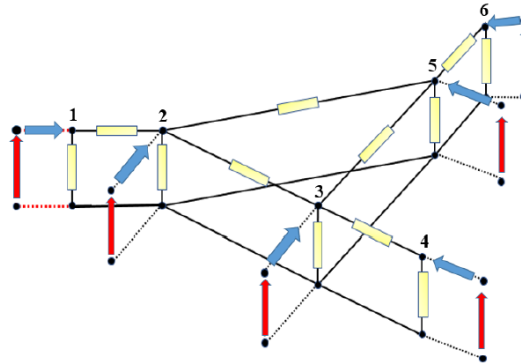


Figura 2 Rete elettrica con gli elementi (doppi bipoli) rappresentati attraverso lo schema equivalente a Π . [10]

È possibile effettuare il calcolo della matrice primitiva (contenente le relazioni tra le correnti di ramo e le tensioni concatenate di ramo) i cui elementi sono ammettenze e della matrice di incidenza (contenente le relazioni tra tensioni concatenate e nodali) costituita invece da 1 e 0. Attraverso opportune operazioni matematiche sarà possibile determinare a partire dalle due matrici sopracitate la definitiva \underline{Y}_n ;

- \underline{Y}_{sh} , racchiude le informazioni tipologiche della rete, essa è una matrice quadrata diagonale la quale presenta al suo interno tutte le ammettenze riguardanti nodi di saldo, nodi PV e nodi PQ che vengono calcolate e/o indicate nel modo seguente:

1. $\underline{y}_a = 0$, nodo di Saldo;

1. 1

2. $\underline{y}_g = -\frac{(p_g - jq_g)}{v_g^2}$, nodi PV di generazione;

1. 2

3. $\underline{y}_l = \frac{(p_l - jq_l)}{v_l^2}$, nodi PQ di carico;

1. 3

per una miglior chiarezza rappresentiamo tale matrice nella figura seguente:

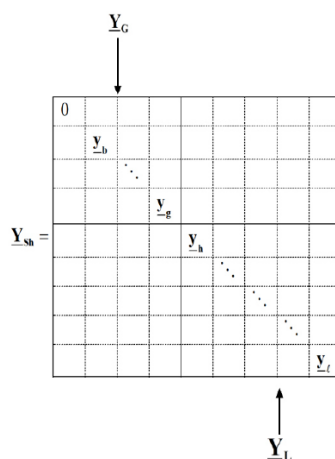


Figura 3 Inserimento delle ammettenze di carico e generazione all'interno della matrice \underline{Y}_{sh} . [10]

a differenza della \underline{Y}_n tale matrice è diagonale essendo che tutte le ammettenze che la compongono sono derivate e non anche longitudinali (ciò causato dal modello a π);

Entrambe le matrici sopra descritte sono sparse, le quali sono una classe speciale di matrici con la caratteristica di contenere un significativo numero di elementi nulli, il che permette una miglior gestione della memoria. Attraverso poi una successiva somma tra la matrice alle ammettenze derivate rappresentativa di carichi e generatori e la matrice alle ammettenze nodali otteniamo:

$$\underline{Y} = \underline{Y}_{sh} + \underline{Y}_n$$

1.4

la quale a valle della determinazione dei moduli e argomenti delle tensioni di tutti i nodi PQ e degli argomenti dei nodi PV e il calcolo del Power Flow sarà possibile calcolare le correnti entranti nei nodi grazie alla seguente formula [10]:

$$\underline{i} = \underline{Y} \underline{u}$$

1.5

1.2.1 Algoritmo iterativo per la ricerca della soluzione

Essendo il problema del power flow non lineare, esso sarà linearizzato e, volta per volta risolto iterativamente fino al raggiungimento della convergenza. L'algoritmo precedentemente utilizzato dall'università degli studi di Padova prende il nome di *Admittance Matrix Correction (AMC)* [10] e consisteva nella continua modifica della matrice alle ammettenze nodali che comportava inversioni di matrici di grandi dimensioni ed un elevato costo computazionale. Per risolvere tale problematica si è deciso di passare all'utilizzo del metodo *Fringing Currents Correction (FCC)*, esso prevede di far assumere alla matrice \underline{Y} gli stessi valori iniziali delle ammettenze calcolate al paragrafo precedente senza però in questo caso effettuare un continuo aggiornamento di quest'ultime per ogni iterazione, ciò che invece viene aggiornato è un set di correnti entranti nei nodi che concorrono a variare la potenza nodale fino al raggiungimento della convergenza (vincoli rispettati). Ciò ha permesso al *metodo FCC* di essere molto più prestante in confronto al *metodo AMC*, si è dimostrato che per le reti più disparate che vanno da 18 a 450 nodi gli abbassamenti percentuali avvenuti della *CPU-time* sfiorano il 94%. [21]

1.2.2 Metodo FCC

A valle di una stima iniziale delle potenze reattive dei nodi di generazione, delle tensioni ai nodi di carico e delle ammettenze dei carichi e dei generatori ottenute secondo le eq. (1.1), (1.2) e (1.3) sarà possibile esprimere la relazione (1.5) che esplicitata in forma matriciale sarà pari a:

$$\begin{array}{c} \underline{i}_g \\ \underline{\Delta i}_l \end{array} \begin{array}{c} \left[\begin{array}{c} \underline{i}_a \\ \underline{\Delta i}_b \\ \vdots \\ \underline{\Delta i}_g \end{array} \right] \\ \left[\begin{array}{c} \underline{\Delta i}_h \\ \vdots \\ \vdots \\ \underline{\Delta i}_f \end{array} \right] \end{array} = \begin{array}{cc} \left[\begin{array}{cc} \underline{Y}_{GG} & \underline{Y}_{GL} \\ \underline{Y}_{LG} & \underline{Y}_{LL} \end{array} \right] & \begin{array}{c} \left[\begin{array}{c} \underline{v}_a \\ \underline{v}_b \\ \vdots \\ \underline{v}_g \end{array} \right] \\ \left[\begin{array}{c} \underline{v}_h \\ \vdots \\ \vdots \\ \underline{v}_f \end{array} \right] \end{array} \end{array} \begin{array}{c} \underline{u}_g \\ \underline{u}_l \end{array}$$

Figura 4 Matrice complessa alle ammettenze totale. Si noti il partizionamento in quattro blocchi, fondamentale per portare alla scrittura di relazioni vettoriali e subvettoriali per il calcolo di tutte le tensioni nodali della generica rete oggetto di studio.

e sotto forma di sistema a:

$$\begin{cases} \underline{i}_g = \underline{Y}_{GG} \underline{u}_g + \underline{Y}_{GL} \underline{u}_l \\ \underline{\Delta i}_l = \underline{Y}_{LG} \underline{u}_g + \underline{Y}_{LL} \underline{u}_l \end{cases}$$

1.6

A partire dalla seconda equazione della (1.6) esplicitiamo il fasore delle tensioni sui nodi di carico ottenendo:

$$\underline{u}_l = \underline{Y}_{LL}^{-1} \underline{Y}_{LG} \underline{u}_g + \underline{Y}_{LL}^{-1} \underline{\Delta i}_l$$

1. 7

Sostituendo la (1.7) nella prima equazione della (1.6) otteniamo:

$$\underline{i}_g = \left(\underline{Y}_{GG} - \underline{Y}_{GL} \underline{Y}_{LL}^{-1} \underline{Y}_{LG} \right) \underline{u}_g + \underline{Y}_{GL} \underline{Y}_{LL}^{-1} \underline{\Delta i}_l$$

1. 8

L'equazione (1.8) può essere scritta in forma matriciale e a partire da quest'ultima sarà possibile determinare la relazione che lega le correnti incrementali di carico con le correnti incrementali dei generatori:

$$\underline{\Delta i}_g = \underline{C} \underline{u}_a + \underline{D} \underline{u}_g + \underline{L}_{xL} \underline{\Delta i}_l$$

1. 9

Dalla (1.9) esprimo l'equazione in funzione della \underline{u}_g ottenendo:

$$\underline{u}_g = \underline{D}^{-1} \underline{C} \underline{u}_a + \underline{D} \left(\underline{\Delta i}_g - \underline{L}_{xL} \underline{\Delta i}_l \right)$$

1. 10

Una volta ricavate le tensioni dei nodi di generazione, a partire dalla (1.7) si ricavano invece i rimanenti fasori relativi ai nodi di carico. Si evince dunque che le condizioni necessarie per ricavare tutte le tensioni è la conoscenza per ogni ciclo iterativo delle correnti di iniezione ai nodi, a tal proposito per iniziare il ciclo verranno poste le seguenti condizioni iniziali a quest'ultime:

$$\underline{\Delta i}_{l1} = \underline{\Delta i}_{g1} = 0$$

1. 11

A seguito della determinazione di una prima approssimazione delle tensioni dei vari nodi attraverso le (1.7) e la (1.10), si procederà con il calcolo delle potenze apparenti dei nodi con il fine di confrontare potenze attive e reattive determinatesi dall'algoritmo con i vincoli posti in sede di progetto, tale confronto attuato attraverso una differenza espressa dalle seguenti formule:

$$\underline{\Delta S}_l = \underline{S}_{l1} - \overline{S}_l$$

1. 12

Con \underline{S}_n indicheremo il vettore contenente le potenze apparenti dei nodi PQ calcolati alla prima iterazione, con \overline{S}_l il vettore delle potenze apparenti di vincoli settate per i nodi PQ e con $\underline{\Delta S}_l$ il vettore degli scarti di potenza che verrà scomposto in parte reale ed immaginaria e poi confrontato con i valori di tolleranza precedentemente fissati (usualmente attorno a 10^{-8}) come di seguito rappresentato:

$$|\underline{\Delta S}_{lreale}| < toll$$

1. 13

$$|\underline{\Delta S}_{limmaginario}| < toll$$

1. 14

Nel caso la tolleranza non venga rispettata si procederà con il ricalcolo della corrente incrementale di carico a partire dalla formula seguente:

$$\underline{v}_l \Delta \underline{i}_l^* = \left| \underline{v}_l \right|^2 \underline{y}_{ll}^* - 1^2 \underline{y}_{ll}^*$$

1. 15

Da cui si ricava:

$$\underline{\Delta i}_l = \underline{y}_{ll} \frac{\left| \underline{v}_l \right|^2 - 1}{\underline{v}_l^*}$$

1. 16

La quale è valida per ogni nodo PQ . Una volta determinata la (1.16) sarà possibile attraverso la (1.9) calcolare la $\underline{\Delta i}_g$; questi due nuovi vettori di correnti incrementali appena calcolate saranno poi utilizzati per procedere con il secondo passo del ciclo e così iterativamente, sino al raggiungimento della convergenza. Racchiudiamo nel flow chart di seguito rappresentato i vari passaggi che avvengono all'interno di ogni ciclo [21]:

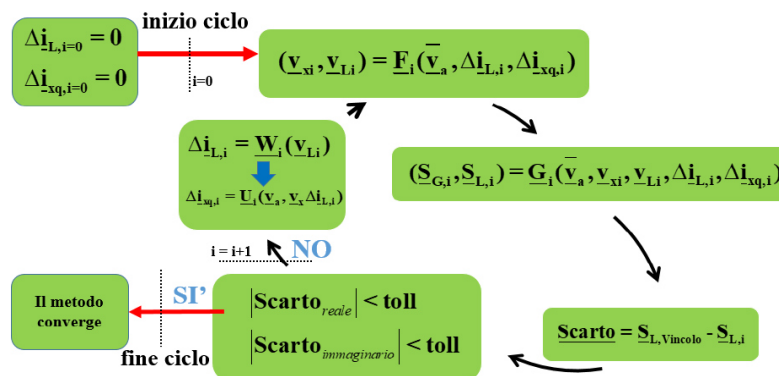


Figura 5 Schematizzazione dell' algoritmo FCC. [34]

1.3 Inclusione delle linee HVDC VSC/LCC e Metodo IRPE per la stima di Potenza Reattiva

Tale algoritmo si basa su un'estensione del metodo patavino, attraverso la modellizzazione sotto forma di nodi PV/PQ delle reti HVDC-VSC/LCC e la loro successiva inclusione all'interno della matrice Y_{sh} (in quanto modellizzati come carichi e generatori) lo rende così in grado di risolvere reti di trasmissione AC/DC. Le motivazioni che hanno spinto ad una integrazione delle reti HVDC sono, oltre ad una più generalità e versatilità raggiunta da parte del codice una grande motivazione in ambito economico e ambientale che spesso spingono alla scelta delle tecnologie stesse. La tecnica che permette tutto ciò viene definita in letteratura come *DC Elimination Technique* questo perché la rete reale da essere definite come AC/DC verrà trasformata dall'algoritmo in una rete completamente AC senza però perdere le caratteristiche fisiche della rete stessa (trasmissione di Potenza, collegamento tra i nodi e le perdite lungo tali reti) includendo tali informazioni all'interno dei nodi PV/PQ opportunamente modellizzati come presentato di seguito [12].

1.3.1 Modellizzazione linee HVDC VSC

Le linee HVDC VSC sono costituite da convertitori IGBT di potenza completamente controllabili (in apertura e chiusura) e che permettono la regolazione della Potenza e tensione trasmessa. Vediamo nella figura seguente la casistica più semplice nell'ambito della trasmissione dell'energia elettrica e analizziamola:

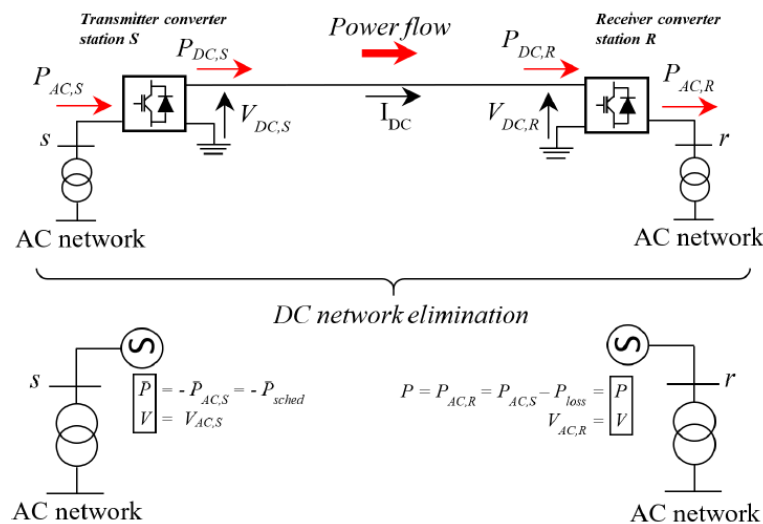


Figura 6 Trasformazione di una linea HVDC-VSC monopolare a due avvolgimenti con dei nodi PV. [12]

Il flusso di Potenza va dalla stazione trasmittente S alla ricevente R, modellizzati entrambi come nodi PV come anticipato nella sezione precedente, i vincoli tra le due stazioni sono differenti e sono assegnate nel seguente modo:

- **Stazione trasmittente (S)**, il convertitore è controllato per assorbire la potenza schedulata e per controllare il fasore della tensione rispetto alla rete a cui è connessa, i vincoli sono quindi:

$$P_{ACS} = -P_{sched}$$

1. 17

$$V_{ACS} = V_{sched}$$

1. 18

- **Stazione ricevente (R)**, per quando riguarda la tensione essa è completamente controllata sia lato AC che lato DC:

$$V_{ACR} = V_{AC,sched} \text{ e } V_{DCR} = V_{DC,sched}$$

1. 19

la potenza all'arrivo risulta essere invece dipendente dalle perdite riscontrate lungo la trasmissione e viene quindi espressa secondo tale formulazione:

$$P_{ACR} = P_{sched} - P_{loss}$$

1. 20

dove con P_{loss} si indica:

$$P_{loss} = P_{loss_{conv,S}} + P_{loss_{conv,R}} + P_{loss_{line}}$$

1. 21

dove $P_{loss_{conv,S,R}}$ sarà pari a:

$$P_{loss_{conv,S,R}} = P_{loss_{noload}} + P_{loss_{switch}} + P_{loss_{cond}}$$

1. 22

e con $P_{loss_{line}}$:

$$P_{loss_{line}} = r l I_{DC}^2$$

1. 23

tutte le perdite sono legate alla I_{DC} che può essere determinate a priori dalle caratteristiche della linea e dei convertitori di potenza da noi conosciuti a partire da tale sistema:

$$\begin{cases} P_{AC,S} - P_{loss_{conv,S}} = I_{DC} V_{DC,S} \\ V_{DC,S} = V_{DC,R} + r l I_{DC} \end{cases}$$

1. 24

da cui otteniamo l'equazione seguente per la determinazione della corrente DC:

$$I_{DC}^2 (r l + k'') + I_{DC} (V_{DC,R} + V_{drop}) + P_{loss,S} - P_{AC,S}$$

1. 25

le espressioni sopra espote per una linea HVDC-VSC monopolare possono essere dunque generalizzate per linee con un maggior numero di convertitori e configurazioni differenti ma stessa tecnologia IGBT per la trasmissione di potenza [12].

1.3.2 Modellizzazione linee HVDC LCC

Le linee *HVDC-LCC* sono basate sulla tecnologia a tiristori, tali dispositivi sono controllabili in chiusura ma non in apertura, la tensione V_{AC} non è dunque completamente regolabile, sarà dunque necessario modellizzare tali linee attraverso nodi *PQ*, rappresentiamo di seguito il caso di una linea *HVDC-LCC* monopolare con trasformatori a due avvolgimenti:

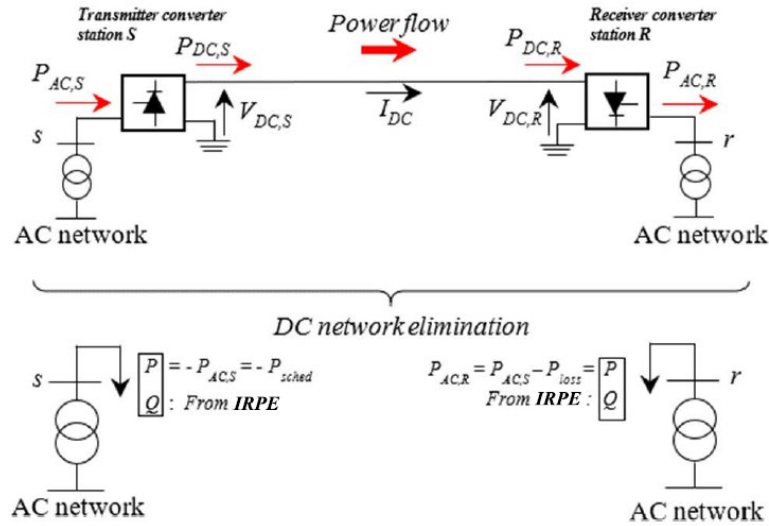


Figura 7 Modellizzazione di una linea monopolare HVDC-LCC con nodi PQ. [12]

Il flusso di potenza va dalla stazione trasmittente *S* alla ricevente *R*, modellizzati entrambi, i vincoli tra le due stazioni sono differenti e sono assegnate nel seguente modo:

- *Stazione Trasmittente (S)*, il convertitore trasmittente controlla la $V_{DC,S}$ attraverso l'angolo di ritardo di commutazione $0^\circ < \alpha < 90^\circ$ con la formulazione seguente:

$$V_{DC,S} = \frac{3\sqrt{2}}{\pi} V_{LL,S} \cos(\alpha) - \frac{3\omega L_s}{\pi} I_{DC} + k'' I_{DC} + V_{drop}$$

1. 26

per quanto riguarda invece la I_{DC} trasmessa lungo la linea, si determinerà a partire dal seguente sistema:

$$\begin{cases} P_{sched} = I_{DC} V_{DC,S} \\ V_{DC,S} = V_{DC,R} + r I_{DC} \end{cases}$$

1. 27

da cui otteniamo:

$$I_{DC}^2 (r l) + I_{DC} (V_{DC,R}) - P_{DC,sched} = 0$$

1. 28

una volta determinate dalle equazioni superiori Tensione e corrente DC , possiamo porre il vincolo di Potenza pari a:

$$P_{AC,S} = P_{DC,sched} + P_{loss_conv,S}$$

1. 29

dove $P_{loss_conv,S}$ sarà sempre determinabile attraverso (1.22).

- *Stazione Ricevente (R)*, converte la Potenza in arrivo dal bus DC in Potenza AC controllando il funzionamento dei tiristori in modalità inverter, utilizzando dunque un angolo di ritardo pari a $90^\circ < \alpha < 180^\circ$, la $V_{DC,R}$ sarà calcolata attraverso l'eq. (1.26) modificando l'angolo e la I_{DC} sarà pari a quella calcolata precedentemente per la stazione trasmittente. Detto questo abbiamo che il vincolo di Potenza è pari a:

$$P_{AC,R} = -(P_{DC,sched} - P_{loss_conv,R})$$

1. 30

dove $P_{loss_conv,R}$ sarà sempre determinabile attraverso (1.22). Le espressioni sopra esposte per una linea $HVDC-LCC$ monopolare possono essere dunque generalizzate per linee con un maggior numero di convertitori e configurazioni differenti ma stessa tecnologia a *TIRISTORI* per la trasmissione di potenza, per una trattazione più precisa si consiglia la visione di [12];

Per la scelta dei vincoli Q si opta per una nuova tecnica sviluppata interamente presso l'università degli studi di Padova, il metodo **IRPE (Iterative Reactive Power Estimation)** valido per il funzionamento degli LCC sia da inverter che da raddrizzatore. Per prima cosa assumiamo che la reattanza di commutazione e le perdite del convertitore siano trascurabili, ricordiamo poi che la Potenza reattiva Q assorbita dal convertitore a tiristori dipende dall'angolo di ritardo che indicheremo con α per il funzionamento da raddrizzatore e β per il funzionamento da inverter e che nella formula seguente indicheremo con θ :

$$Q_{conv} \cong |P| \tan(\theta)$$

1. 31

con P indichiamo la potenza assorbita dal sistema AC o iniettata dallo stesso. Differentemente dalla I_{DC} la Q non può essere calcolata a priori in quanto i tiristori di potenza non possono controllare la tensione lato AC in quanto essa stessa è soluzione del Power Flow e la potenza attiva P dipende da θ . Viene sviluppato dunque il ciclo iterativo seguente. Consideriamo la potenza reattiva assorbita da un LCC come:

$$Q_{conv} = \sqrt{S^2 - P_{sched}^2}$$

1. 32

e riscritta come:

$$Q_{conv} = \sqrt{(\sqrt{3} V_{AC} I_{AC})^2 - P_{sched}^2}$$

1. 33

considerando poi la relazione che lega I_{AC} e I_{DC} derivante dalle leggi dei convertitori a tiristori:

$$I_{AC} = \frac{\sqrt{6}}{\pi} I_{DC}$$

1. 34

e sostituendola nella (1.33) otteniamo:

$$Q_{conv} = V_{AC} \sqrt{\frac{18}{\pi^2} I_{DC}^2 - \left(\frac{P_{sched}}{V_{AC}} \right)^2}$$

1. 35

la quale lega la Q_{conv} con I_{DC} e P_{sched} che conosciamo a priori e con V_{AC} che è invece soluzione stessa del Power Flow. A tal proposito i valori iniziali di tensione dei nodi AC a cui sono connessi i convertitori saranno $V_{AC,conv} = 1$ [p.u.]. Una volta sostituito all'interno dell'eq. (1.35) sarà calcolato Q_{conv} . Ora che abbiamo la conoscenza totale dei vincoli PQ possiamo lanciare l'algoritmo PFPD e ottenere il risultato di esso ovvero V_{AC} , verranno poi di nuovo sostituite dentro l'eq. (1.35) e lanciati conseguentemente ancora una volta il PFPD, si procederà così iterativamente fintanto che il vincolo non sarà soddisfatto:

$$\Delta V_k = V_k - V_{k-1} < tol = 100[V]$$

1. 36

la procedura precedente può essere riassunta nella figura seguente:

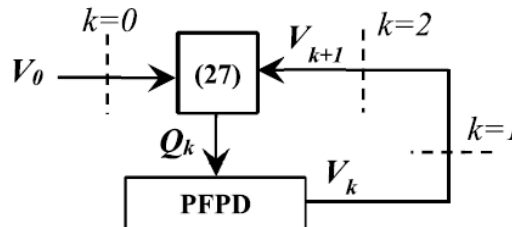


Figura 8 Flow chart per il metodo IRPE. [12]

I nodi PV e PQ relativi ai convertitori verranno inseriti nella matrice come trattato nella sezione *Metodo Patavino per lo studio dei flussi di Potenza*. Tale inclusione non comporta quindi un ingrandimento della matrice che rimane di dimensioni (nodi x nodi), pertanto il PFPD non subisce modifiche sostanziali che ne intaccano il buon condizionamento ma ne deriva la possibilità di renderlo adatto allo studio di reti AC/DC. [12]

1.4 Implementazione MatLab del metodo IRPE

Come indicato nel titolo, tale sezione prenderà in considerazione le sole modifiche da apportare all'algoritmo patavino (*PFPD*) per l'implementazione del metodo *IRPE*, per una descrizione dettagliata dell'algoritmo *PFPD* in ambiente *MatLab* sarà più opportuno fare riferimenti a gli articoli citati in bibliografia. [10] [11] [12]

1.4.1 Dati in input

L'importazione dei dati riferiti alla rete presa in analisi avviene attraverso la function "*A_INTERFACCIA_IRPE.m*" (la quale può essere visionata all'interno dell'appendice) essa fa da tramite tra l'ambiente *EXCEL* e *MatLab*, le leggi che invece regolano l'importazione tra *DIGsilent* ed *EXCEL* sono raccolte nell'appendice in "*Guida importazione dati Digsilent-Matlab*". E' giusto e opportuno porre una particolare attenzione all'importazione delle linee *HVDC* da *Digsilent* ad *EXCEL*, di seguito le accortezze per ogni "sheet" del file *EXCEL*:

1. Terminal, eliminare quei terminali associati ai convertitori *AC/DC* e *DC/AC* e alle loro relative terre ponendo:
Energized = 0;
OutofService = 1;
2. Linee *VSC*, le celle *EXCEL* sono create a mano direttamente sul foglio stesso, in quanto non è stata implementata una possibilità di importazione adeguata, l'intestazione è la seguente:

Tabella 1 Intestazione Linee *VSC* foglio *EXCEL* per l'importazione.

Name	Grid	Terminal i	Terminal j	Out of Service	Energized	Length	Unom	r	P1	P2	Type_HVDC
------	------	---------------	---------------	-------------------	-----------	--------	------	---	----	----	-----------

particolare attenzione va data al numero di righe di questa colonna, nel caso di linee monopolari le quali presentano una sola linea di collegamento attiva si compileranno nel modo seguente:

Tabella 2 Rappresentazione di una linea monopolare *HVDC-VSC*.

Name	Grid	Terminal i	Terminal j	Out of Service	Energized	Length	Unom	r	P1	P2	Type_HVDC
		Busbar	Busbar			km	kV	Ohm/km	kW		
Line	Grid	Terminal(2)	Terminal(3)	0	1	200	150	0,0085	250	0	1

nel caso di linee *VSC* bipolari esse hanno sempre e comunque un'andata e un ritorno, di conseguenza, anche se in sostanza essa è una linea bipolare dovrà essere rappresentata nel foglio Excel *LINEE VSC* come due linee separate che immettono la stessa potenza dallo stesso lato, è come se fossero considerate come due linee *VSC* monopolari che però fanno capo allo stesso terminale a monte dei trasformatori *AC*, di seguito la tabella rappresentativa di una linea presa in esempio:

Tabella 3 Rappresentazione di una linea bipolare HVDC-VSC come due linee monopolari HVDC-VSC.

Name	Grid	Terminal i	Terminal j	Out of Service	Energized	Length	Unom	r	P1	P2	Type_HVDC
		Busbar	Busbar			km	kV	Ohm/km	kW		
Line	Grid	Terminal(10)	Terminal	0	1	200	150	0,0085	250	0	1
Line(1)	Grid	Terminal(12)	Terminal(2)	0	1	200	150	0,0085	250	0	1

In questo caso secondo quanto detto precedentemente, la linea presa in analisi trasmetterà una potenza totale di $500 [kW]$ ma sarà considerata come due linee monopolari separate che tramettono rispettivamente $250 [kW]$. L'indicazione "Type_HVDC" fa riferimento al tipo di trasformatore che troviamo a monte di ogni rispettiva linea monopolare, indicheremo:

- Type_HVDC = 1, nel caso di trasformatore a 2 avvolgimenti;
- Type_HVDC = 2, nel caso di trasformatore a 3 avvolgimenti;

3. Linee LCC, le celle EXCEL hanno l'intestazione seguente:

Tabella 4 Intestazione Linee LCC foglio EXCEL per l'importazione.

Name	Grid	Type	Terminal i	Terminal j	Out of Service	Energized	Length	U1 nom	r	P1	P2	Q1	Q2	Type_HVDC
------	------	------	------------	------------	----------------	-----------	--------	--------	---	----	----	----	----	-----------

le considerazioni fatte in precedenza per la questione delle linee monopolari e bipolari è la stessa usata per il caso "Linee VSC", di seguito riportiamo le tabelle esempio:

Tabella 5 Rappresentazione di una linea monopolare HVDC-LCC.

Name	Grid	Type	Terminal i	Terminal j	Out of Service	Energized	Length
			Busbar	Busbar			km
Line	Grid	LINE_DC	Terminal	Terminal(1)	0	1	200
Unom	r	P1	P2	Q1	Q2	Type_HVDC	
kV	Ohm/km	kW	kW	kVAR	kVAR		
150	0,0085	300	0	500	500	3	

Tabella 6 Rappresentazione di una linea bipolare HVDC-LCC come due linee monopolari HVDC-LCC.

Name	Grid	Type	Terminal i	Terminal j	Out of Service	Energized	Length
			Busbar	Busbar			Km
Line	Grid	LINE_ DC	Terminal	Terminal(1)	0	1	200
Line	Grid	LINE_ DC	Terminal(2)	Terminal(3)	0	1	200
Unom	r	P1	P2	Q1	Q2	Type_ HVDC	
kV	Ohm/ km	kW	kW	kVAR	kVAR		
150	0,0085	150	0	100	100	1	
150	0,0085	150	0	100	100	1	

4. VSC, le celle EXCEL presentano l'intestazione seguente e sono ottenute tramite l'operazione manuale di copia e incolla da DigSilent, l'intestazione risulta essere la seguente:

Tabella 7 Intestazione convertitori VSC foglio EXCEL per l'importazione.

Name	Grid	Terminal AC	No-load-losses	Switching-loss-factor	Resistive-loss-factor	Outof Service	Energized	Type_Conv
------	------	-------------	----------------	-----------------------	-----------------------	---------------	-----------	-----------

particolare attenzione va data al numero di righe di questa colonna, nel caso di linee monopolari a 2 avvolgimenti avremmo conseguentemente due convertitori in totale, uno trasmittente e l'altro ricevente, saranno compilati nel modo seguente:

Tabella 8 Rappresentazione dei convertitori di una linea monopolare HVDC-VSC con trasformatore a due avvolgimenti.

Name	Grid	Terminal AC	No-load-losses	Switching-loss-factor
		Busbar	kW	kW/A
PWM	Grid	Terminal(2)	0,2	0,2
PWM	Grid	Terminal(3)	0,2	0,2
Resistive-loss-factor	OutofService	Energized	Type_Conv	Rated VoltageAC
Ohm				kV
0,2	0	1	1	195
0,2	0	1	1	195

è dunque opportuno inserire come sopra fatto per primi i convertitori lato trasmittente e successivamente quelli lato ricevente, per ogni linea, volta per volta. Inoltre, è necessario inserire per primi i convertitori relativi ai collegamenti $Type_{HVDC} = 1$ e poi quelli relativi ai collegamenti $Type_{HVDC} = 2$. Le perdite (*No-load-losses*, *Switching-loss-factor*, *Resistive-loss-factor*) se presenti vengono poste con lo stesso identico valore e pari a 0.2. Particolare attenzione va posta nel caso di linee bipolari, esse vengono considerate come due monopolari e per tale ragione inseriremo prima i dati di tutta una monopolare (raddrizzatori + inverter) e poi tutti i dati dell'altra.

5. *LCC*, le celle *EXCEL* presentano la medesima intestazione già presentata in *Tabella 7*, i fattori di perdita sono gli stessi e presentati nello stesso identico modo, per completezza viene rappresentata la tabella:

Tabella 9 Rappresentazione dei convertitori di una linea monopolare HVDC-VSC con trasformatore a due avvolgimenti.

Name	Grid	Terminal AC	No-load-losses	Switching-loss-factor
		Busbar	kW	kW/A
Rectifier/Inverter/2 DC-Connection	Grid	Terminal(2)	0,2	0,2
Rectifier/Inverter/2 DC-Connection(1)	Grid	Terminal(3)	0,2	0,2
Resistive-loss-factor	OutofService	Energized	Type_Conv	Rated VoltageAC
Ohm				kV
0,2	0	1	1	195
0,2	0	1	1	195

6. *TRAF02*, attraverso il comando di copia/incolla è possibile passare tutte le informazioni necessarie senza alcuna modifica.
7. *TRAF03*, attraverso il comando di copia/incolla è possibile passare tutte le informazioni necessarie senza alcuna modifica.
8. *LINEE*, eliminiamo le linee *HVDC* che non esistono in *PFPD* essendo tutto modellizzato all'interno dei nodi fittizi o dei bilanci di potenza finali, le uniche linee che devono essere considerate in questo foglio *EXCEL* sono quelle di trasmissione *AC*.
9. *SHUNT FILTER*, attraverso il comando di copia/incolla è possibile passare tutte le informazioni necessarie senza alcuna modifica.
10. *GENERATORI*, attraverso il comando di copia/incolla è possibile passare tutte le informazioni necessarie senza alcuna modifica.
11. *CARICHI*, attraverso il comando di copia/incolla è possibile passare tutte le informazioni necessarie senza alcuna modifica.

1.4.2 Corpo del codice

Una volta caricate nel *workspace* di *MatLab* tutte le variabili attraverso “*A_INTERFACCIA_IRPE.m*” lanciamo il codice *PPFD.m* che riporto di seguito:

```
format long
path(path, 'B_POWER_FLOW_ns_PQ')
A00_Pulizia_ns_PQ
global N_nodi_gen N_nodi_tot Nbase omegan
load('C:\Users\ostinic48425\Desktop\DEFINITIVO\Dati\Dati.mat');

C=(table2array(RETE_LCC(:, {'Type_HVDC'})));
C(C(:,1)~=1,:)=[];
c=size(C,1);

E=(table2array(RETE_LCC(:, {'Type_HVDC'})));
E(E(:,1)~=3,:)=[];
e=size(E,1);

    if isempty (CONV_VSC) == 1 && isempty (CONV_LCC) == 1
        PFPD_AC
    elseif isempty (CONV_LCC) == 1
        PFPD_VSC
    else
        Metodo_IRPE
    end

save('Data_from_LF_ott.mat');
save('C_POST_PROCESSING_PF_ott\Data_from_LF_ott_SOGNO.mat');
```

Figura 9 Codice di inizializzazione e scelta PFPD.

Lo scopo principale di tale codice è quello di indirizzare il *PPFD* verso l’algoritmo adatto alla risoluzione della rete tramite la valutazione delle variabili *CONV_VSC* e *CONV_LCC*. Nel caso in cui non vi siano sia reti *LCC* che *VSC* verrà lanciata la function *PPFD_AC* che racchiude in sé l’algoritmo *PPFD* classico trattato nella sezione precedente. Nel caso invece in cui vi fosse la presenza di reti *VSC*, sarà lanciato l’algoritmo *PPFD_VSC* il quale rappresentiamo in parte di seguito:

```
Risp = input ('Premere 1 per considerare le perdite. Premere 2 per considerare il caso ideale');
if (Risp == 1)

    PerditeVSC = ones(size(VSC,1),1)*0.2;
    LossesVSC = array2table(PerditeVSC);

    VSC(:,4) = LossesVSC;
    VSC(:,5) = LossesVSC;
    VSC(:,6) = LossesVSC;
end

if (Risp == 2)

    No_PerditeVSC = zeros(size(VSC,1),1);
    No_LossesVSC = array2table(No_PerditeVSC);

    VSC(:,4) = No_LossesVSC;
    VSC(:,5) = No_LossesVSC;
    VSC(:,6) = No_LossesVSC;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SOLO VSC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[Ploss_1,Ploss_2] = A07_PERDITE_VSC(VSC, CONV_VSC, LINEE_VSC, RETE_VSC);

% PFPD_AC
```

Figura 10 Fase iniziale dell’algoritmo *PPFD_VSC*.

Come già anticipato, l'algoritmo in *figura 9* non è completo in quanto dove viene indicato con *% PFPD_AC* abbiamo il lancio dell'algoritmo classico, particolare attenzione va dedicata invece alla function *A07_PERDITE_VSC* in essa verranno racchiuse tutte quelle considerazioni che accomunano la presenza delle linee *VSC* nel codice; l'output di tale algoritmo è *Ploss_1* e *Ploss_2* le quali sono le perdite totali di una linea *HVDC* con trasformatore rispettivamente a due o tre avvolgimenti, esse verranno successivamente inserite nel bilancio di potenza come segue:

```
In = (Yretebase)*En_sogno;
Nn = En_sogno.*conj(In)*Nbase;
Ntotg = - sum(Nn(1:N_nodi_gen));
Ncarichi = sum(Nn(N_nodi_gen+1:N_nodi_tot));

Pn=real(Nn);
Qn=imag(Nn);

P_loss = real(Ntotg) - real(Ncarichi);
Q_loss = imag(Ntotg) - imag(Ncarichi);
modulo_tensioni=abs(En_sogno);
angolo_tensioni=angle(En_sogno)*180/pi;

Pn(1)=Pn(1)+Ploss_1+Ploss_2;
```

Figura 11 Inserimento delle perdite dovute alle linee HVDC nella variabile Pn.

nel caso di presenza di sole linee *LCC* o di linee *LCC* e *VSC* verrà lanciata la function *Metodo_IRPE*, teoricamente essa è stata trattata precedentemente, il codice presenta una grande generalità, in quanto riprendendo i concetti trattati nella sezione **1.4.1 Dati input** le linee *HVDC* monopolari o bipolari che siano vengono trattate nello stesso modo e l'unica cosa che conta è la suddivisione e il corretto inserimento in *EXCEL*, a tal proposito vengono nel codice *IRPE* sviluppate tre casistiche:

- Linea HVDC monopolare con trasformatore a due avvolgimenti;
- Linea HVDC monopolare con trasformatore a tre avvolgimenti;
- Linea HVDC monopolare con trasformatore a due e tre avvolgimenti;

i convertitori *LCC* che vengono considerati come nodi *PQ* saranno trattati differentemente e gli verranno associati determinati vincoli, è quindi opportuno creare un vettore che nella function si chiamerà "*pos*" esso permette di identificare tutti i nodi *PQ* fittizi, la creazione del vettore viene rappresentata nel codice seguente ed è differente per ogni casistica sopra anticipata.

```
Posizione_PQ_fittizi = (size(CAR_GEN,1) - size(OTHER,1) - num_PQ_fittizi);
N = num_radd_lm;
pos_radd_l = zeros(N,1);
for i=1:N
    pos_radd_l(i) = (Posizione_PQ_fittizi+1) + 2*(i-1);
end
pos_radd_lm = [pos_radd_l];
pos_inv_l = pos_radd_l + ones(N,1);
pos_inv_lm = [pos_inv_l];
pos = [pos_radd_lm;pos_inv_lm];
```

Figura 12 Creazione del vettore pos per una linea monopolare con trasformatore a 2 avvolgimenti.

Prendiamo in considerazione per la spiegazione il caso di due linee monopolari con trasformatori a 2 avvolgimenti (che è lo stesso di una linea bipolare con trasformatori a 2 avvolgimenti vista dal codice). La prima parte è di inizializzazione, con la variabile *Posizione_PQ_fittizi* si identifica la posizione

precedente ai nodi PQ fittizi, con N invece il numero di raddrizzatori totale (somma dei raddrizzatori di ogni linea), che sarà uguale a quello degli inverter. Si entrerà poi all'interno del ciclo `for` che numera i raddrizzatori, esso andrà da uno fino al numero di raddrizzatori $N=2$, ed effettuerà una numerazione associando il valore 1 al primo raddrizzatore e 3 al secondo, questo è dovuto al fatto che la posizione 2 e 4 sarà lasciata agli inverter così da poter avere gli elementi della prima linea monopolare come 1 e 2 e della seconda come 3 e 4, vediamo una rappresentazione grafica nella figura seguente:

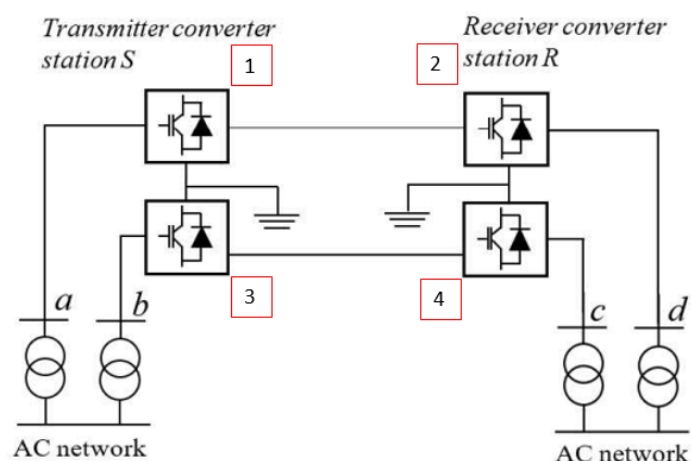


Figura 13 Numerazione di raddrizzatori ed inverter.

Una volta calcolato il vettore pos_radd_IM verrà sommato con un vettore costituito solo da uni così da poter associare la corretta posizione ad ogni inverter relativo ad ogni raddrizzatore. Verrà infine creato il vettore pos costituito dalle posizioni dei raddrizzatori e poi quelle degli inverter, infine tramite il comando `sort` riordino numericamente il vettore.

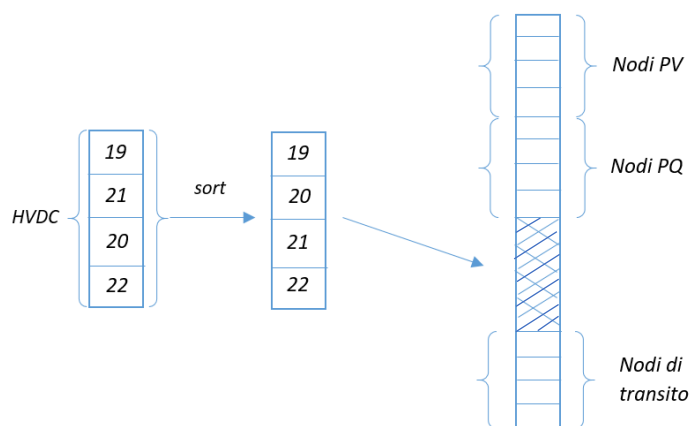


Figura 14 Posizionamento dei nodi relativi al sistema HVDC nella matrice CAR_GEN .

Prendiamo ora in considerazione il caso di una linea $HVDC-LCC$ monopolare a tre avvolgimenti, per una completa generalizzazione il codice è contenuto all'interno di un ciclo `if` che contraddistingue il caso in cui vi sia una sola linea o ve ne siano di più in quanto la variabile `if` contiene il numero di linee presenti. Un'importante differenza rispetto al codice precedente sono le posizioni dei raddrizzatori e degli

inverter in quanto vi sono sempre due raddrizzatori e due inverter per ogni linea, di conseguenza vi sarà la variabile pos_radd_1 per il convertitore più in alto e pos_radd_2 per quello più in basso, lo stesso ragionamento vale anche per le posizioni degli inverter. La numerazione segue la legge della figura seguente:

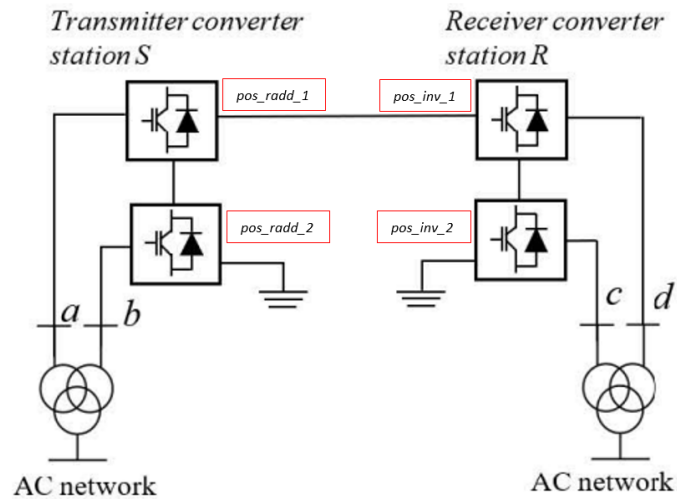


Figura 15 Assegnazione vettori di posizionamento ai convertitori.

Il principio di numerazione è lo stesso del caso precedente.

```

if e == 1
N = num_radd_2M/2;
    pos_radd_1 = zeros(N,1);
    for i=1:N
        pos_radd_1(i) = (Posizione_PQ_fittizi+1) + 4*(i-1);
    end
    pos_radd_2 = pos_radd_1 + ones(N,1);
    pos_radd_2M = [pos_radd_1;pos_radd_2];
    pos_inv_1 = pos_radd_1 + 2*ones(N,1);
    pos_inv_2 = pos_inv_1 + ones(N,1);
    pos_inv_2M = [pos_inv_1;pos_inv_2];

pos = [pos_radd_2M;pos_inv_2M];
pos = sort(pos);
else
    N = num_radd_2M/e;
    pos_radd_1 = zeros(N,1);
    for i=1:N
        pos_radd_1(i) = (Posizione_PQ_fittizi+1) + 4*(i-1);
    end
    pos_radd_2 = pos_radd_1 + ones(N,1);
    pos_radd_2M = [pos_radd_1;pos_radd_2];
    pos_inv_1 = pos_radd_1 + 2*ones(N,1);
    pos_inv_2 = pos_inv_1 + ones(N,1);
    pos_inv_2M = [pos_inv_1;pos_inv_2];

pos = [pos_radd_2M;pos_inv_2M];
pos = sort(pos);
end

```

Figura 16 Creazione del vettore pos per una linea monopolare con trasformatore a 3 avvolgimenti.

Nel caso misto in cui ritroviamo sia linee a due avvolgimenti che linee a tre avvolgimenti, il codice è l'unione dei due precedentemente spiegati con la piccola differenza che la numerazione nel caso di tre avvolgimenti deve partire non dalla $Posizione_PQ_fittizi$ ma da $Posizione_PQ_fittizi + size(pos,1)$,

dove $size(pos,1)$ è il numero di elementi della/e linee HVDC con trasformatori a due avvolgimenti. Il modus operandi rimane lo stesso.

```

N = num_radd_1M;
    pos_radd_1 = zeros(N,1);
    for i=1:N
        pos_radd_1(i) = (Posizione_PQ_fittizi+1) + 2*(i-1);
    end
    pos_radd_1M = [pos_radd_1];
    pos_inv_1 = pos_radd_1 + ones(N,1);
    pos_inv_1M = [pos_inv_1];
    pos=[pos_radd_1M;pos_inv_1M];
% Monopolare 3 avvolgimenti
if e == 1
N = num_radd_2M/2;
    pos_radd_1 = zeros(N,1);
    for i=1:N
        pos_radd_1(i) = (Posizione_PQ_fittizi+1+size(pos,1)) + 4
4*(i-1);
    end
    pos_radd_2 = pos_radd_1 + ones(N,1);
    pos_radd_2M = [pos_radd_1;pos_radd_2];
    pos_inv_1 = pos_radd_1 + 2*ones(N,1);
    pos_inv_2 = pos_inv_1 + ones(N,1);
    pos_inv_2M = [pos_inv_1;pos_inv_2];

pos = [pos_radd_2M;pos_inv_2M];
pos = sort(pos);
else
    N = num_radd_2M/e;
    pos_radd_1 = zeros(N,1);
    for i=1:N
        pos_radd_1(i) = (Posizione_PQ_fittizi+1+size(pos,1)) + 4
4*(i-1);
    end
    pos_radd_2 = pos_radd_1 + ones(N,1);
    pos_radd_2M = [pos_radd_1;pos_radd_2];
    pos_inv_1 = pos_radd_1 + 2*ones(N,1);
    pos_inv_2 = pos_inv_1 + ones(N,1);
    pos_inv_2M = [pos_inv_1;pos_inv_2];

pos = [pos;pos_radd_2M;pos_inv_2M];
pos = sort(pos);

```

Figura 17 Creazione del vettore pos per una linea monopolare con trasformatore a due e tre avvolgimenti.

2. Probabilistic Power Flow

L'utilizzo del *Probabilistic Load Flow (PLF)* è stato fin dai primi tempi dedicato allo studio sugli errori di previsione dell'assorbimento di energia da parte dei carichi e alla probabilità di disservizi della rete e su come quest'ultime considerazioni potessero nuocere al sistema elettrico. Negli ultimi decenni il focus si è nettamente spostato sull'interazione tra la produzione di energia elettrica da fonti di energie rinnovabili e la rete di distribuzione e trasmissione. Il *Deterministic Load flow (DLF)* ignora l'aleatorietà presente in diverse grandezze contenute all'interno dell'architettura elettrica, esso effettua una fotografia del sistema nell'istante t con le determinate produzioni $P = P \dots P_i$ di ogni generatore e l'assorbimento $P_u = P_u \dots P_{ui}$ di ogni carico; possiamo con certezza affermare che nel caso di un campo fotovoltaico in produzione connesso alla rete di media tensione (MT) e interfacciato alla rete di alta tensione (AT) o altissima tensione (AAT) per mezzo di un trasformatore elevatore avrà una produzione che dipenderà dall'istante del giorno in cui verrà effettuata la fotografia, all'istante t la produzione risulterà pari a P , mentre nel caso dell'istante t_n la produzione sarà P_n , in funzione dell'impatto che tale generatore ha sulla rete può provocare grandi o piccole variazioni di risultato sul *DLF*. Per le seguenti considerazioni un approccio deterministico non risulterà più soddisfacente ai fini dell'analisi di una rete con tali caratteristiche. [23]

2.1 La madre dello studio Probabilistico

Barbara Borkowska docente universitaria del Politecnico di Varsavia fece nel 1974 una pubblicazione scientifica nominata "*Probabilistic Load Flow*" [22] che portò per la prima volta alla luce lo studio di reti elettriche di trasmissione e distribuzione di energia contenente carichi che presentavano variabili assorbimenti di energia elettrica tali per cui l'utilizzo di un singolo Load Flow avrebbe portato ad una scorretta analisi della rete presa in questione. *Borkowska* applicò per la prima volta tale metodo sulla rete seguente:

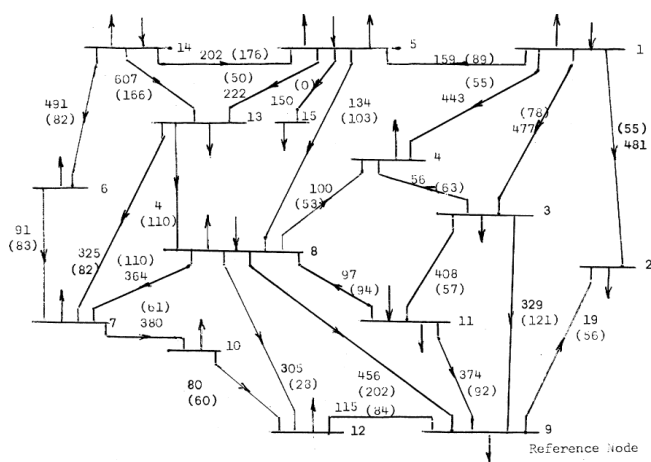


Figura 18 Rete elettrica presa in analisi da Borkowska per lo studio del Probabilistic Load Flow. [22]

essa presenta una configurazione e dei parametri di linea fissi ma con valori di potenze in ingresso ed in uscita nei nodi variabili che conseguentemente provocavano variabilità dei valori di flussi di potenza nei rami. Il metodo più immediato per la risoluzione di una rete di tale tipo sarebbe stato quello di eseguire più *Load Flow* per ogni valore di carico preso in esame; tuttavia, in quegli anni i computer non

erano così performanti da poter soddisfare un costo computazionale così elevato ed una analisi a posteriori così precisa, motivo per cui *Borkowska* propose un metodo basato sull'uso di calcoli probabilistici. La formulazione principale di tale metodo è la seguente:

$$P_B = AP_W$$

2.1

ricordando $P_W = (P_N - \varphi(S))$ otteniamo:

$$P_B = A(P_N - \varphi(S))$$

2.2

la quale ci permette di determinare il vettore delle potenze di ramo P_B a partire dal vettore casuale dei carichi nodali P_N e $P_L = \varphi(S)$ che risulta essere il vettore di variazione della potenza dei carichi controllati secondo il dispacciamento. Tutto questo attraverso A ovvero la matrice di trasformazione. Analizziamo di seguito membro a membro gli elementi dell'eq. (2.1):

- $P_N = T P_n$, con T matrice di trasformazione la quale può assumere valori pari a 1 o a 0 e P_n vettore di potenze casuali in ingresso e uscita;
- $A = [A_o|0]$, matrice di trasformazione la quale lega le potenze nodali con le potenze di ramo con $A_o = Z_B^{-1}C(C_t Z_B^{-1}C)^{-1}$, dove C è la matrice di connessione e Z_B la matrice contenente i moduli delle impedenze di ramo.
- $\varphi(S) = P_L$, è il vettore che in funzione del vettore S può esprimere la perdita o il surplus di potenza;

L'eq. (2.2) sviluppata al calcolatore attraverso ulteriori considerazioni che possono essere consultate nel documento [22] trovabile in bibliografia permette la risoluzione del Probabilistic Load Flow, i risultati ottenuti saranno i seguenti:

- Media e deviazione standard di P_B ;
- Media e deviazione standard di P_N ;
- Funzione di distribuzione di probabilità di P_B e di P_N ;

Tali risultati ci forniscono delle risposte interessanti a quesiti che vengono posti prima dell'analisi vera e propria della rete, quali ad esempio la probabilità che la potenza ecceda i valori massimi o che sia maggiore e/o minore di un determinato valore, quale sia la gamma di oscillazione delle potenze dei carichi e qual è il più probabile valore ottenibile. [22]

2.2 Metodi per il Probabilistic Power Flow

Il PLF risulta essere fondamentale nella determinazione dello stato del sistema elettrico di potenza con presenza di indeterminanze nella rete. L'efficacia di un metodo di studio probabilistico viene confrontata in funzione di: tempo di esecuzione, accuratezza e semplicità. Le diverse metodologie possono essere raggruppate in tre sottogruppi [25]:

- *Metodi Numerici e di Campionamento*, dove troviamo come primo tra questi il *Metodo di Monte Carlo (MCM)* e il *Metodo Quasi - Monte Carlo (QMC)* con le loro relative tipologie di generazione di numeri casuali e di campionamento, le due più usate sono il *Simple Random Sampling (SRS)* e il *Latin Hypercube Sampling (LHS)*;
- *Metodi Analitici*, tra cui ritroviamo il *Convolution Method* e il *Cumulant Method*;
- *Metodi ad Approssimazione*, come il *Point estimate Method (PEM)*, il *Unscented Transformation Method (UTM)* ecc...

di seguito una figura rappresentativa con tutte le metodologie:

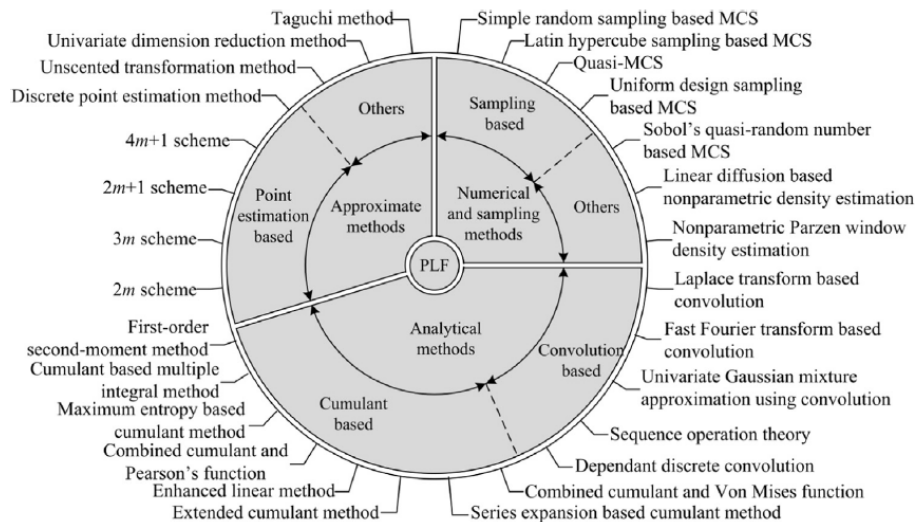


Figura 19 PLF classificati secondo il metodo di analisi. [25]

2.2.1 Metodi Numerici

Metodi numerici, come, ad esempio, il *Metodo di Monte Carlo (MCM)* consistono nel lanciare un gran numero di *DLF* con combinazioni di valori nodali differenti. Ad ogni *DLF* vi sarà la risoluzione delle cinque equazioni non lineari caratteristiche dei flussi di potenza definite come *Load flow equations (LF's)* [23]:

$$P_i = U_i \sum_{k=1}^n U_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \quad 2.3$$

$$Q_i = U_i \sum_{k=1}^n U_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad 2.4$$

$$P_{ik} = -t_{ik} G_{ik} U_i^2 + U_i U_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \quad 2.5$$

$$Q_{ik} = t_{ik} B_{ik} U_i^2 - B_{ik} U_i^2 + U_i U_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad 2.6$$

$$Q_{i(sh)} = U_i^2 B_{i(sh)} \quad 2.7$$

Dove P_i e Q_i sono le potenze attive e reattive nette iniettate dai nodi i , P_{ik} e Q_{ik} sono le potenze attive e reattive che circolano nelle linee ik nel lato del bus i ; U_i e U_k sono le tensioni rispettivamente al nodo i e al nodo k ; θ_{ik} è la differenza di angolo che c'è tra la tensione al nodo i e la tensione al nodo k ; G_{ik} e B_{ik} sono la parte reale e immaginaria della matrice alle ammettenze. *MCM* è costituito da due caratteristiche principali, il generatore di numeri casuali e il campionamento casuale, la prima tra queste è facilmente costruibile usando tecniche matematiche di base a partire dalla *Probability density function (PDF)* mentre vi sono diverse tecniche di campionamento esistenti, la più comune tra queste è l'*SRS* grazie alla sua elevata semplicità, la combinazione di *MCM* e *SRS* viene definita *MCM crude*. Il grande vantaggio dell'*MCM crude* e dell'*MCM* in generale rispetto agli altri metodi è che le equazioni non devono essere linearizzate, vengono infatti mantenute le equazioni classiche del *Load Flow* (2.3 – 2.7), questo ci permette di eliminare l'operazione di linearizzazione che in quanto approssimazione delle equazioni stesse porta ad ottenere un risultato più impreciso. Data l'elevatissima accuratezza nei risultati di tale metodo grazie alle considerazioni precedenti, esso sarà utilizzato come base di confronto per ulteriori metodi che svolgono la stessa funzione, come l'algoritmo sviluppato in tale tesi *PFPD_PRB* e gli ulteriori metodi, analitici e approssimativi. Tuttavia, uno dei maggiori svantaggi di *MCM crude* è che il numero di campioni necessari deve per forza incrementare se l'accuratezza vuole essere alta ciò rende questo metodo computazionalmente molto costoso e con elevato spazio di memoria richiesto. [26] In particolare se noi consideriamo una rete test semplice costituita da 6 nodi di cui 3 con distribuzione normale, ed una varianza del 5 [%] si avrà la convergenza a partire da 20000 campioni e un tempo di simulazione di 150 [s] come possiamo vedere nella figura sottostante:

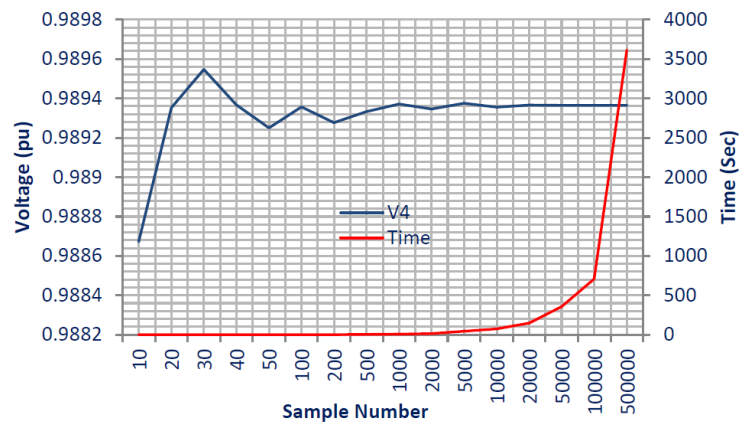


Figura 20 Tasso di convergenza e tempo computazionale rispetto al numero di campioni utilizzati nel caso di MCM crude. [26]

Con il fine di diminuire il numero di campioni necessari per la convergenza e quindi di migliorare il costo computazionale, la precisione e la casualità del campionamento vengono introdotti metodi *pseudo-randomici* rispetto al classico *SRS*, colui che ha trovato maggior sviluppo all'interno del *Power System Analysis* è il *Latin Hypercube Sampling (LHS)* ovvero una combinazione di campionamento stratificato e aleatorio che campiona la variabile in tutta la distribuzione e che procede suddividendo l'insieme di numeri possibili in intervalli, genera un numero a partire da quest'ultimi e poi andrà ad eliminare l'intervallo contenente il numero appena scelto prima di effettuare la scelta successiva; tuttavia, tale tecnica genera indesiderati fenomeni di dipendenza tra gli elementi campionati. Rispetto al classico *SRS*, tuttavia, presenta i seguenti vantaggi:

- A parità di campioni sfruttati può ricoprire per ognuno di questi e per quindi ogni variabile causale uno spettro di elementi più ampio;
- È un metodo più robusto;

2.2.2 Metodi Analitici

Metodi analitici, tali metodologie sono state sviluppate con lo scopo di avere un metodo per la risoluzione del *PLF* con un costo computazionale inferiore, il primo tra questi è il *Convolution Method* che si basa sull'utilizzo di tecniche di convoluzione a partire da input di potenza descritti da *funzioni di densità di probabilità (PDFs)* ottenendo quindi variabili di output che sono esse stesse ancora *PDFs*. Vista però la difficoltà nell'effettuare l'operazione di convoluzione con le *PDFs* si dovranno apportare le quattro seguenti approssimazioni [23]:

- Linearizzazione delle *LFs* (2.3 – 2.7);
- Totale indipendenza o dipendenza lineare tra le variabili di potenza;
- Assunzione di distribuzioni normali o discrete per le variabili in analisi;
- Configurazione della rete e parametri costanti;

la risoluzione di tali metodi richiedono tre passaggi fondamentali, per prima cosa si esegue un *DLF* nel quale le grandezze aleatorie di input non vengono considerate come *PDFs* ma gli viene associato il valore medio $\bar{\mu}_x$ della funzione stessa, una volta ricavata la soluzione, essa verrà utilizzata al passo successivo per linearizzare le *LFs* attorno alle soluzioni appena determinate attraverso la serie di Taylor fino al primo ordine, di seguito un esempio generale:

$$Y = f(X)$$

2. 8

la cui forma generalizzata può essere espressa come:

$$X \cong \mu_x + A(Y - \mu_y)$$

2. 9

dove,

$$A = \left(\frac{\partial f}{\partial X} \Big|_{X=\mu_x} \right)^{-1}$$

2. 10

e ottenere le *PDFs* di output ricercate. Il fenomeno di linearizzazione attorno ad un solo termine causa grandi errori nei risultati, vengono dunque adottati metodi di multi-linearizzazione. Come terzo ed ultimo punto, infine, si procederà con l'operazione di convoluzione come di seguito rappresentato:

$$f(X_i) = f(Y_1 - \mu_{y1}) * f(Y_2 - \mu_{y2}) * \dots * f(Y_n - \mu_{yn})$$

2. 11

essa può essere effettuata attraverso la trasformata di *Laplace* oppure, in modo molto più efficiente in termini di accuratezza e onere computazionale utilizzando la *Fast Fourier Transform (FFT)*, nel dominio della frequenza si ha la proprietà tale per cui un prodotto di convoluzione diventerà un normale prodotto algebrico. Attraverso l'anti-trasformata si otterrà poi il risultato della convoluzione ovvero la *PDFs* finale. [24] Recentemente un ulteriore metodo analitico nominato *Cumulant Method* ha acquisito particolare attenzione grazie alle sue alte velocità di calcolo. Tale metodo permette di evitare i

complicati calcoli di convoluzione tra le *PDFs* per procedere alla spiegazione di tale metodo consideriamo il sistema di equazioni:

$$\begin{cases} W = f(X) \\ Z = g(X) \end{cases}$$

2.12

Dove W è il vettore delle potenze nodali attive e reattive immesse, Z è il flusso di potenza nei rami e X il vettore contenente tensioni nodali e relative fasi (tensioni complesse) tutte e tre espresse in condizioni di funzionamento normale; mentre f e g sono le funzioni che legano rispettivamente le tensioni nodali con il vettore delle potenze immesse e con il vettore del flusso di potenza nei rami. In condizioni operative normali la (2.12) diventa la (2.13):

$$\begin{cases} W_o = f(X_o) \\ Z_o = g(X_o) \end{cases}$$

2.13

Assumendo che le potenze iniettate nel sistema subiscano un piccolo disturbo ΔW , ci sarà un cambiamento del tipo ΔX e ΔZ , il disturbo in quanto piccolo può essere sviluppato in serie di Taylor con approssimazione attorno al primo ordine, ottenendo:

$$\begin{cases} W = \Delta W + W_o \cong f(X_o) + f'(X_o)\Delta X \\ Z = \Delta Z + Z_o \cong g(X_o) + g'(X_o)\Delta X \end{cases}$$

2.14

con

$$f'(X_o) = \left. \frac{df(X)}{dX} \right|_{X=X_o} = J_o$$

2.15

$$g'(X_o) = \left. \frac{dg(X)}{dX} \right|_{X=X_o} = G_o$$

2.16

Da cui otteniamo che

$$\Delta X = J_o^{-1}\Delta W = S_o\Delta W$$

2.17

Ricordando che

$$\Delta Z = G_o\Delta X$$

2.18

E sostituendo la (2.17) nella (2.18) otteniamo:

$$\Delta Z = G_o S_o \Delta W$$

2.19

Attraverso le equazioni sovrastanti ogni variazione in termini di potenza in ingresso può essere trasformata in output ($\Delta Z, \Delta W$) attraverso la conoscenza dei cumulanti del vettore delle potenze iniettate. [26] Un trattamento più dettagliato può essere consultato nel Paper [27] della bibliografia.

2.2.3 Metodi ad approssimazione

Metodi ad approssimazione, lo scopo di tali metodi è quello di calcolare i momenti statistici delle *PDFs* in uscita rispetto ai momenti delle funzioni di distribuzione in ingresso, tra i metodi più utilizzati troviamo *First order second moment method (FOSMM)*, *point estimate method (PEM)* e il più recente è *l'unscented transform method (UT)*. Questi metodi approssimativi hanno gradualmente acquisito importanza nell'esecuzione del *PLF* a causa della loro robusta capacità di utilizzare direttamente le equazioni del flusso di carico non lineare come il metodo *MCS*, eliminando così gli errori di linearizzazione associati ai metodi analitici. [26]

3. Statistica e Probabilità applicata allo studio dei flussi di potenza probabilistici

3.1 Strumenti matematici per la statistica

L'analisi statistica è fondamentale per l'applicazione del *Probabilistic Power Flow* in quanto il metodo stesso è basato su essa, conseguentemente sarà opportuno dare una panoramica generale sugli strumenti matematici e statistici che caratterizzano tale applicazione in modo tale da poterne garantire una migliore comprensione.

3.1.1 Media

In statistica, la media (*Average, avg*) è un singolo valore numerico che descrive sinteticamente un insieme di dati. Varie tipologie di media possono essere scelte per descrivere un fenomeno, quelle più comunemente impiegate sono le tre cosiddette medie pitagoriche: aritmetica, geometrica e armonica. Nel linguaggio ordinario, con il termine media si intende comunemente la media aritmetica essa viene usata per riassumere con un solo numero un insieme di dati su un fenomeno misurabile (per esempio, nel caso da noi posto in analisi le tensioni medie su un nodo in funzione di una generazione variabile distribuita). La formula della media aritmetica semplice per n elementi è [13]:

$$\mu_x = \frac{1}{n} \sum_{i=1}^n x_i$$

3.1

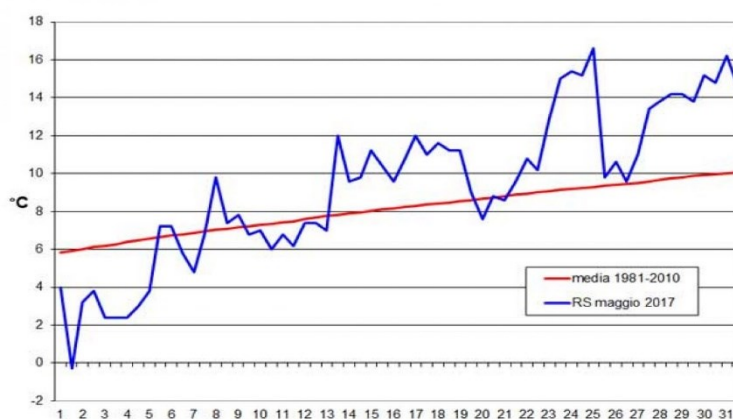


Figura 21 media dei valori.[13]

3.1.2 Scarto quadratico medio

Lo scarto quadratico medio (o deviazione standard, o scarto tipo, o scostamento quadratico medio) è un indice di dispersione statistico, vale a dire una stima della variabilità di una serie di dati o di una variabile casuale. È uno dei modi per esprimere la dispersione (e dunque in statistica la precisione) dei dati intorno ad un indice di posizione, quale può essere, ad esempio, la media aritmetica o una sua stima. Ha pertanto la stessa unità di misura dei valori osservati. In statistica descrittiva lo scarto quadratico medio di un carattere rilevato su una popolazione di unità statistiche si definisce nel seguente modo:

$$\sigma_x = \sqrt{\frac{1}{n} \sum_{i=1}^N (x_i - \mu_x)^2}$$

3. 2

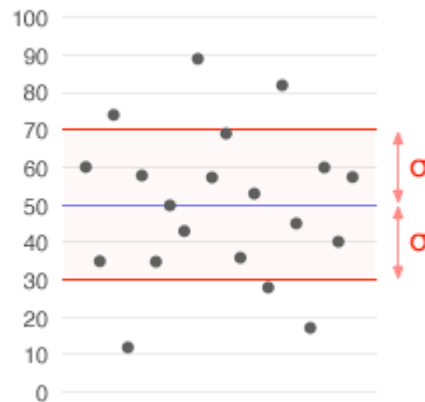


Figura 22 Deviazione standard. [14]

dove μ_x è la media aritmetica del dataset di valori X . A partire dallo scarto quadratico medio si definisce anche il coefficiente di variazione o la deviazione standard relativa come il rapporto tra lo scarto tipo e il valore assoluto della media aritmetica della variabile in esame sempreché quella media sia non nulla [14]:

$$\sigma_x^* = \frac{\sigma_x}{|\mu_x|}$$

3. 3

3.1.3 Varianza

In statistica e in teoria della probabilità la varianza di una variabile statistica o di una variabile aleatoria è una funzione, indicata con σ^2 che fornisce una misura della variabilità dei valori assunti dalla variabile stessa; nello specifico, la misura di quanto essi si discostino quadraticamente dalla media aritmetica o dal valore atteso μ_x .

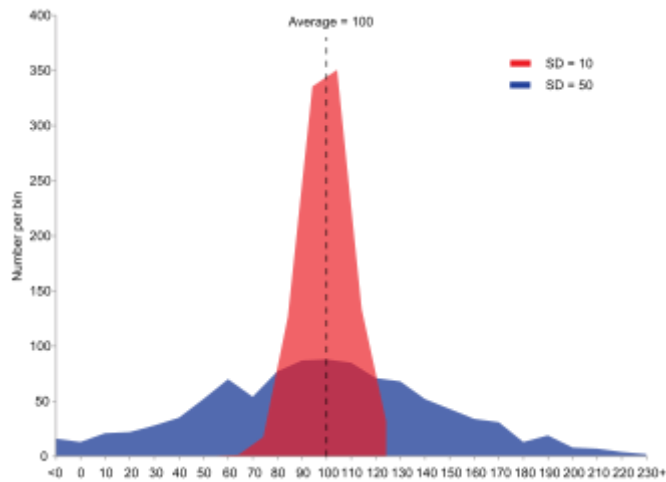


Figura 3 Esempio di due campioni con la stessa media ma diversa varianza, i campioni di colore blu hanno varianza più alta.[17]

La varianza può essere determinata in due modi, o a partire dalla deviazione standard facendone per l'appunto il quadrato, oppure con la formula seguente:

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu_x)^2}{n}$$

3. 4

La quale corrisponde alla media aritmetica dei quadrati delle differenze tra ogni valore della distribuzione e il valore medio della distribuzione [17].

3.1.4 Campo di variazione

In statistica, il campo di variazione è il più semplice indice di variabilità ed è dato dalla differenza tra il valore massimo di una distribuzione ed il valore minimo. Esso può essere definito anche come intervallo di variabilità o gamma [15].

$$W = x_{max} - x_{min}$$

3. 5

3.2 Funzioni di distribuzione di probabilità:

In questa sezione si andranno a trattare le diverse tipologie di funzioni di distribuzioni di probabilità utilizzate per modellizzare le fonti climatiche aleatorie (sole, vento) e conseguentemente la produzione di energia elettrica derivante proprio da esse. Tali funzioni saranno poi utilizzate anche per l'analisi dei risultati, quali ad esempio, sovraccarico dei generatori, *ampacity* delle linee eventuali situazioni patologiche e fattibilità stessa della rete. Oltre ad essere rappresentate sotto forma di curve di distribuzione di probabilità (*Probability density function Pdf*), ogni distribuzione può essere espressa come funzione di ripartizione (*Cumulative distribution function Cdf*). Quest'ultima è ottenuta rappresentando in ordine crescente i valori campionati e conseguentemente le proprie probabilità sull'asse y . Di seguito un esempio applicato alla distribuzione di probabilità normale. [4]

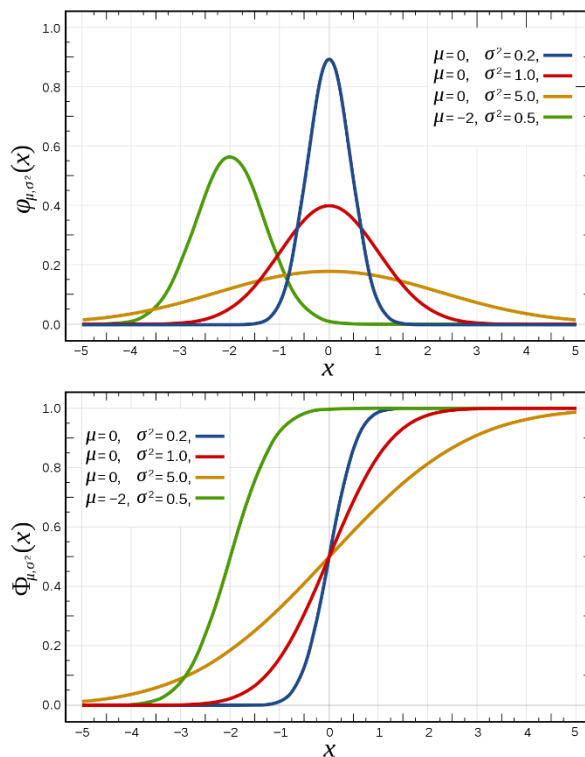


Figura 23 Probability density function (sopra) e Cumulative density function (sotto).[18]

3.2.1 Normale

La distribuzione normale (o distribuzione di *Gauss* dal nome del matematico tedesco *Carl Friedrich Gauss*), nella teoria della probabilità, è una distribuzione continua che è spesso usata come prima approssimazione per descrivere variabili casuali a valori reali che tendono a concentrarsi attorno a un singolo valor medio. Il grafico che la rappresenta ha una forma a campana ed è simmetrico, per tale ragione viene definito come *curva gaussiana* o *curva normale*. Essa è caratterizzata dalla seguente Pdf [5]:

$$h(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

3. 6

con i parametri μ e σ che rappresentano media e deviazione standard. Essa è rappresentata nella *figura sovrastante*. La Cdf è rappresentata dalla formula seguente [18]:

$$H(x) = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)$$

$$\text{con } \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

3. 7

3.2.2 LogNormale

In teoria delle probabilità la distribuzione *LogNormale*, è la distribuzione di probabilità di una variabile aleatoria il cui rispettivo logaritmo segue una distribuzione normale. Quest'ultima può approssimare il prodotto di molte variabili aleatorie positive indipendenti. La sua *Pdf* è pari a:

$$f(x) = \frac{e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}}{x\sqrt{2\pi}(\sigma)} \text{ per } x > 0$$

3. 8

la quale presenta il seguente andamento:

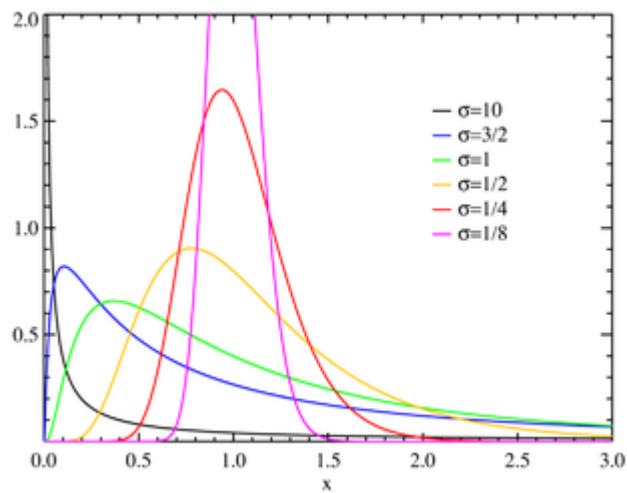


Figura 24 Probability density function di una distribuzione LogNormale con valore costante di media e deviazione standard variabile. [35]

A partire da un Dataset i parametri della distribuzione sono ottenuti attraverso le seguenti formulazioni:

$$\mu = \log \frac{(\mu^2)}{\sqrt{v + \mu^2}}$$

3. 9

$$\sigma = \sqrt{\frac{\log(v)}{\mu^2 + 1}}$$

3. 10

Dove con μ (3. 1) e v (3. 4) indichiamo rispettivamente la media e la varianza calcolata sulla base del Dataset di partenza. Per quanto riguarda la *Cdf* essa viene rappresentata dalla formulazione seguente:

$$F(x) = \frac{1}{\sigma \sqrt{2\pi}} \int_0^x \frac{1}{t} e^{-\frac{(\log(t)-\mu)^2}{2\sigma^2}} dt \text{ per } x > 0$$

3. 11

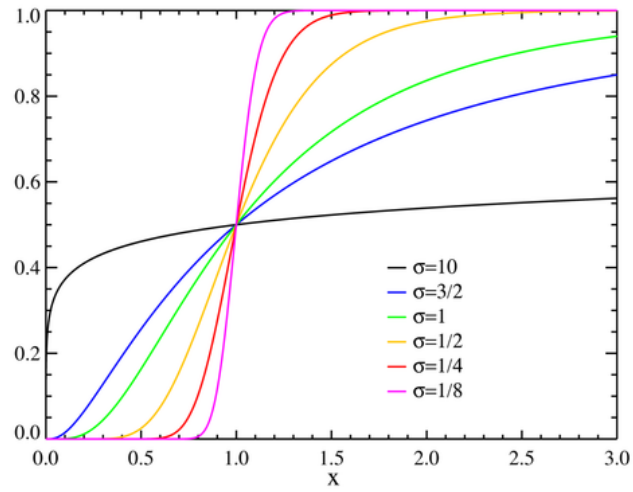


Figura 25 Cumulative density function di una distribuzione LogNormale con valore costante di media e deviazione standard variabile. [35]

3.2.3 Weibull

In teoria delle probabilità la distribuzione di Weibull è una distribuzione continua e descritta dai parametri c (parametro di scala) e k (parametro di forma). Per valori di $c > 0$ e $k > 0$ è definita sul piano dei numeri reali positivi con Pdf pari a:

$$f_w(x) = \frac{k}{c} \left(\frac{x}{c}\right)^{k-1} e^{-\left(\frac{x}{c}\right)^k}$$

3. 12

essa assume poi il nome di distribuzione esponenziale ($k=1$) o distribuzione Rayleigh ($k=2$) in funzione del parametro di forma [4].

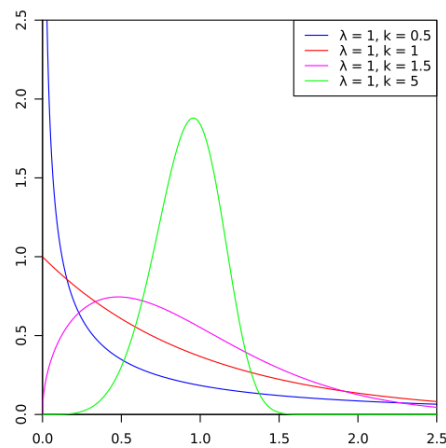


Figura 26 Probability density function di Weibull. [19]

I parametri di scala c e di forma k possono essere calcolati rispettivamente come segue:

$$k = \left(\frac{\sigma}{\mu}\right)^{-1.086} \quad \text{e} \quad c = \frac{\mu}{\text{gamma}\left(1 + \frac{1}{k}\right)}$$

3. 13

La Cdf è rappresentata dalla formula seguente:

$$F(x) = 1 - e^{-\left(\frac{x}{c}\right)^k}$$

3. 14

e assume la caratteristica seguente:

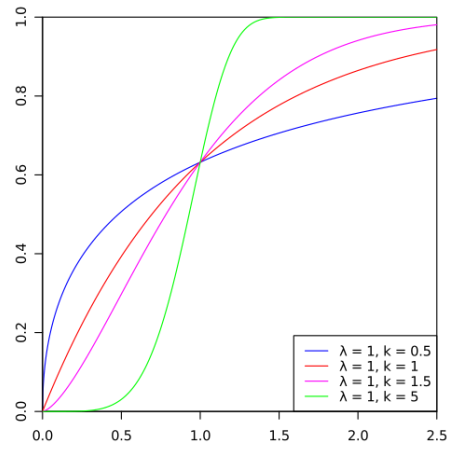


Figura 27 Cumulative distribution function di Weibull[19].

3.2.4 Distribuzione Beta

La distribuzione *Beta* è una distribuzione di probabilità continua definita sull'intervallo unitario $[0, 1]$ da due parametri α e β entrambi maggiori di 0. Essa è descritta dalla seguente Pdf:

$$f_b(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

3. 15

In altri termini la funzione di densità di probabilità è proporzionale alla funzione:

$$x^{\alpha-1}(1-x)^{\beta-1}$$

3. 16

riscalata per un fattore dato dalla funzione Beta:

$$B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx$$

3. 17

essa assume la seguente caratteristica:

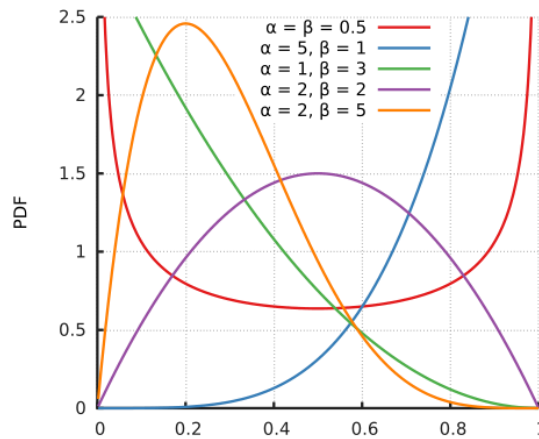


Figura 28 Probability density function della distribuzione Beta [20].

La Cdf è la funzione *Beta* incompleta regolarizzata ed è rappresentata dalla formula seguente:

$$F(x) = \frac{\int_0^x t^{\alpha-1}(1-t)^{\beta-1} dt}{\int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt}$$

3. 18

e assume la caratteristica:

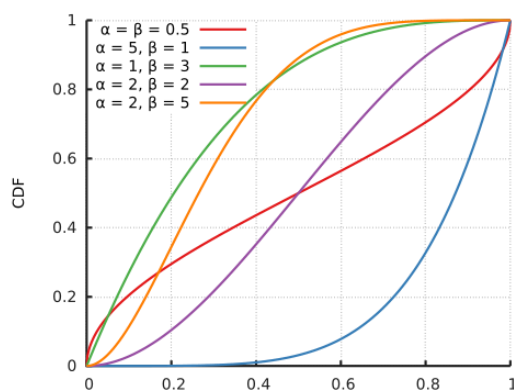


Figura 29 Cumulative distribution function della distribuzione Beta [20].

3.3. Calcolo della probabilità a partire da una Pdf o Cdf

Come già largamente esposto nella sezione (3.2) le Pdf e le Cdf sono funzioni continue di cui non ha senso esprimere la probabilità puntuale che un determinato valore assume poiché quest'ultima risulterebbe sicuramente pari a 0. È corretto invece parlare di probabilità che la variabile casuale assuma valori in un intervallo, anche piccolissimo. La probabilità che la variabile casuale continua X cada in un intervallo $[a, b]$ è data dall'area sottesa alla curva Pdf nell'intervallo stesso e può essere calcolata secondo le seguenti formulazioni:

$$P(X \leq x) = \int_{-\infty}^x f(x) dx \tag{3.19}$$

$$P(a \leq X \leq b) = \int_a^b f(x) dx \tag{3.20}$$

$$P(X \geq x) = \int_x^{\infty} f(x) dx \tag{3.21}$$

Ricordiamo che la probabilità sarà pari a 1, ovvero al 100% solamente quando si integrerà tutta l'area sottesa dalla curva. Nel caso di una Cdf la probabilità dipende dalla variabile aleatoria e può essere calcolata come la differenza tra i valori della Cdf in due punti specifici. In generale, la Cdf di una variabile aleatoria X è definita come:

$$F(x) = P(X \leq x) \tag{3.22}$$

Dove $F(x)$ rappresenta la probabilità che X assuma un valore minore o uguale a x . La Cdf è una funzione crescente, limitata superiormente da 1 e inferiormente da 0. Per calcolare la probabilità associata alla Cdf in un intervallo specifico $[a, b]$, si può utilizzare la seguente formula:

$$P(a \leq x \leq b) = F(b) - F(a) \tag{3.23}$$

Dove $F(a)$ e $F(b)$ sono i valori della Cdf nei punti a e b , rispettivamente.

4. Implementazione del Probabilistic Power Flow al calcolatore

4.1 Implementazione del Probabilistic Power Flow su DigSilent

DigSilent è un software indipendente specializzato nei sistemi elettrici per la trasmissione, distribuzione, generazione, l'impiantistica industriale e le energie rinnovabili. Tra i suoi servizi troviamo *PowerFactory*, *StationWare* e *Monitoring Systems*. *PowerFactory* è lo strumento di *DigSilent* che useremo, facendo particolare attenzione all'aspetto della trasmissione dell'energia elettrica. Lo studio dei flussi di potenza viene utilizzato per analizzare le prestazioni del sistema elettrico, comprendere le perdite di energia e valutare la stabilità del sistema. Inoltre, lo studio dei flussi di potenza può essere utilizzato per analizzare i problemi di congestione della rete e per valutare la capacità della rete di supportare l'interconnessione di nuove fonti di energia rinnovabile. Per eseguire uno studio dei flussi di potenza in *PowerFactory*, è necessario creare un modello completo del sistema elettrico, che includa le reti di trasmissione, le stazioni di generazione, i carichi, i dispositivi di regolazione e protezione e gli elementi di connessione. Una volta creato il modello, si può eseguire una simulazione del flusso di potenza per analizzare la distribuzione delle tensioni, delle correnti e delle perdite di potenza in tutto il sistema. *PowerFactory* fornisce una vasta gamma di strumenti per eseguire lo studio dei flussi di potenza, tra cui il calcolo delle matrici di ammettenza, il calcolo delle potenze reattive, la simulazione delle condizioni di contingenza e il calcolo della sensibilità del sistema. Inoltre, *PowerFactory* può essere utilizzato per valutare l'impatto di scenari di generazione e carico alternativi sul sistema elettrico. [7]

4.1.1 Analisi Probabilistica

L'analisi probabilistica della rete elettrica sta diventando un argomento di studio sempre più interessante in quanto essa permette di effettuare studi di fattibilità di progetto ed economica considerando la presenza di fonti di energie rinnovabile all'interno di sistemi di trasmissione e distribuzione. In genere, questo tipo di analisi caratterizza gli elementi della rete attraverso curve di distribuzione di probabilità, tra cui *Normale*, *Weibull*, *Uniforme* e *LogNormale* che possono modellizzare la produzione di un generatore da fonte rinnovabile o la variabilità dei carichi. Dai dati di input stocastici si ottengono conseguentemente dati di output stocastici, e quindi anche curve di distribuzione di probabilità. I risultati vengono espressi sotto forma di media μ , deviazione standard σ , valore massimo, valore minimo, eccetera, come riportato nel Capitolo 3.1 [7].

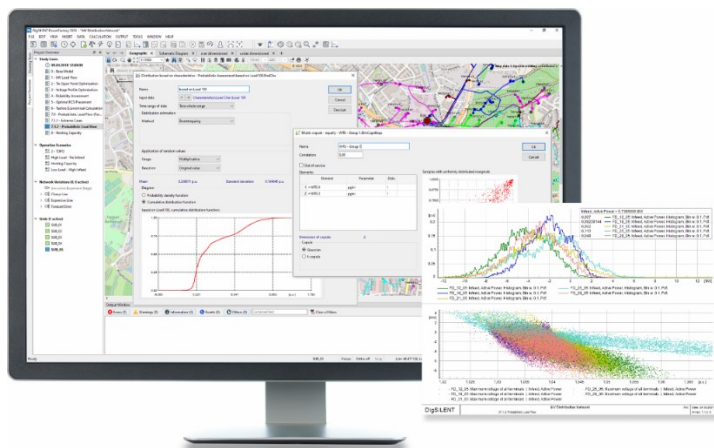


Figura 30 Analisi probabilistica DigSilent.

4.1.2 Metodi per l'analisi probabilistica

All'interno dell'interfaccia *Probabilistic Analysis* di *PowerFactory* ritroviamo due metodologie di calcolo per la rete posta in analisi:

- *Load Flow (LF)*, è una tecnica di analisi utilizzata nell'ambito dei sistemi di trasmissione e distribuzione dell'energia elettrica per calcolare i flussi di potenza e le tensioni presenti nella rete elettrica in condizioni di regime permanente. In pratica, il Load flow permette di determinare la distribuzione di tensione e potenza attiva e reattiva in ogni punto della rete elettrica, in modo da garantire un'adeguata fornitura di energia elettrica ai carichi e prevenire eventuali problemi di stabilità e sicurezza del sistema. È una tecnica di analisi utilizzata nell'ambito dei sistemi di trasmissione e distribuzione dell'energia elettrica per calcolare i flussi di potenza e le tensioni presenti nella rete elettrica in condizioni di regime permanente. In pratica, il Load flow permette di determinare la distribuzione di tensione e potenza attiva e reattiva in ogni punto della rete elettrica, in modo da garantire un'adeguata fornitura di energia elettrica ai carichi e prevenire eventuali problemi di stabilità e sicurezza del sistema;
- *Optimal Power Flow (OPF)*, è una tecnica di ottimizzazione utilizzata nell'ambito dei sistemi di trasmissione e distribuzione dell'energia elettrica per determinare la combinazione di generazione e di consumo di energia elettrica che minimizza i costi e/o massimizza il rendimento del sistema. In pratica, l'OPF calcola la combinazione ottimale di potenza attiva e reattiva da distribuire ai generatori e ai carichi in modo da soddisfare i requisiti di potenza e tensione del sistema, rispettando contemporaneamente i vincoli imposti dalla rete elettrica (come, ad esempio, i limiti di trasmissione, le limitazioni di tensione e le restrizioni sulle capacità di generazione);

Gli algoritmi utilizzati per effettuare tale studio, sono:

- *Metodo di Monte Carlo (MCM)*;
- *Metodo Quasi Monte Carlo (QMCM)*;

4.1.3 Metodo di Monte Carlo (MCM)

Il *Metodo di Monte Carlo*, già precedentemente descritto, prevede la generazione di campioni pseudo-casuali attraverso l'algoritmo di generazione di numeri casuali *Mersenne Twister*. Tali campioni generati vengono definiti pseudocasuali in quanto è pressoché impossibile ottenere numeri perfettamente casuali poiché vi è un algoritmo che li genera. Inoltre, essi sono generati a partire da una distribuzione di probabilità, pertanto gli intorni di numeri con probabilità più elevata avranno conseguentemente più probabilità di essere generati rispetto ad altri numeri. Le funzioni che permettono tale generazione sono definite funzioni generatrici da distribuzioni. Ad ogni iterazione di tale metodo, viene lanciato un *Deterministic Power Flow* a cui vengono associati valori costanti agli elementi della rete privi di distribuzione, mentre agli elementi descritti da una funzione di distribuzione di probabilità viene associato uno dei valori generati a partire dalla funzione generatrice di distribuzioni [7]. Un'importante proprietà di tale metodo, che si applica molto bene al nostro caso di studio, è il fatto che la buona convergenza di esso non dipenda dal numero di elementi della rete ma solo dal numero di iterazioni del metodo stesso come indicato nell'ordine di convergenza che risulta essere pari a:

$$O\left(\frac{1}{\sqrt{N}}\right)$$

4. 1

L'ordine di convergenza di un metodo indica la velocità con cui l'errore del metodo diminuisce all'aumentare del numero di iterazioni o campioni. Un metodo con un ordine di convergenza $O\left(\frac{1}{\sqrt{N}}\right)$ significa che l'errore si ridurrà di un fattore costante diviso la radice quadrata del numero di iterazioni o campioni (N). In altre parole, all'aumentare di (N), la precisione del metodo migliora lentamente, in modo proporzionale alla radice quadrata di N . Questo tipo di convergenza è considerato relativamente lento, ma comunque migliore di un ordine di convergenza lineare $O\left(\frac{1}{N}\right)$.

4.1.4 Metodo Quasi-Monte Carlo (QCM)

Il *QCM* lavora sullo stesso principio dell'*MCM* ma utilizzando campioni distribuiti in modo simmetrico così da poter coprire in modo più uniforme lo spazio d'interesse. Il *QCM* ha un ordine di convergenza superiore rispetto al *MCM*. In particolare, l'ordine di convergenza del *QCM* è:

$$O\left(\frac{\ln(N)^d}{N}\right)$$

4. 2

dove N è il numero di campioni generati. Questo significa che, rispetto al *MCM*, il *QCM* può convergere più rapidamente per un dato numero di campioni generati. Tuttavia, il *QCM* richiede una maggiore complessità computazionale nella generazione dei campioni, rispetto al *MCM*. Per ottenere una precisione più alta il metodo necessita di un numero inferiore di iterazione, risultato eccezionale rispetto al *MCM* classico, se non fosse per il fatto che il metodo stavolta dipende dal numero di elementi della rete impedendone una convergenza per ogni rete presa in analisi [7].

4.1.5 Metodi a confronto

La questione che più differenzia tali metodi è la distribuzione dei campioni nello spazio preso in analisi il quale viene rappresentato nella figura seguente:

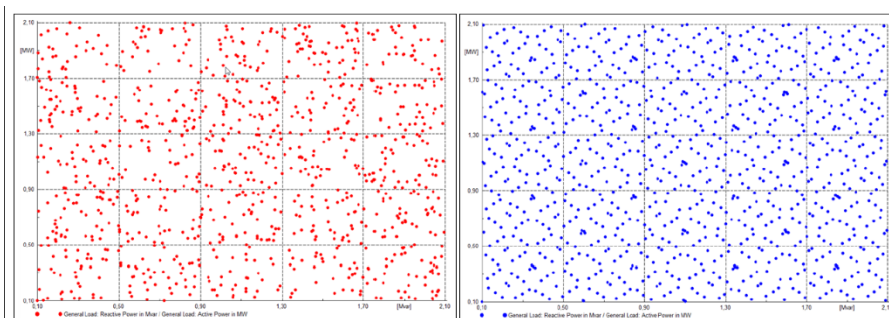


Figura 31 Confronto tra il Metodo di Monte Carlo e il Metodo Quasi Monte Carlo.

Nella tabella riassuntiva seguente i metodi vengono messi a confronto:

Tabella 5 Metodi messi a confronto.

Metodo	MCM	QCM
Tasso di convergenza	$O\left(\frac{1}{\sqrt{N}}\right)$	$O\left(\frac{\ln(N)^d}{N}\right)$
One digit more precision: x=number of samples	x=100	x=10
Convergenza dipendente dalla dimensione della rete	No	Si

Per lo studio statistico sulle reti è stato scelto il Metodo di Monte Carlo, poiché grazie alle sue qualità di convergenza sempre garantita con un elevato numero di iterazioni, può essere utilizzato per lo studio di qualsiasi tipo di rete, anche per reti di grandi dimensioni (elevato numero di nodi) [7].

4.1.6 Modellizzazione dei nodi di generazione distribuita

I nodi presenti all'interno della rete, come già discusso nel primo capitolo, sono suddivisi in tre grandi categorie: nodi di generazione, nodi di carico e nodo di saldo. La modellizzazione di un impianto ad energia rinnovabile (come l'eolico, il fotovoltaico o il biogas) può essere immediatamente associata ad un nodo di generazione *PV*. Tuttavia, nella letteratura [28], questi impianti sono trattati come veri e propri nodi *PQ* in quanto la maggior parte di essi sono dotati di sistemi di elettronica di potenza per l'interfacciamento con la rete, che permettono la regolazione della potenza attiva e reattiva in ingresso e uscita. La trattazione di questi nodi come nodi *PQ* consente di mantenere inalterato il codice principale e di effettuare valutazioni a priori, evitando di appesantire i tempi di esecuzione.

4.1.7 Modellizzazione degli aerogeneratori

La modellizzazione delle turbine eoliche come nodi *PQ* viene effettuata considerando la potenza attiva direttamente collegata alla velocità del vento, mentre la potenza reattiva dipende dalla tipologia di generatore in esse presente (fisso, semi-variabile e variabile). In particolare modo per la trattazione della potenza attiva vi sono tre valori generalmente assunti:

$$\begin{cases} P_{WTt} = 0 & 0 \leq W_t \leq W_{ci} \\ P_{WTt} = f(W_t) & W_{ci} \leq W_t \leq W_r \\ P_{WTt} = P_{max} & W_r \leq W_t \leq W_{coff} \end{cases}$$

4.3

dove $f(W_t)$ è una funzione non lineare che solitamente viene approssimata con una funzione lineare come rappresentato di seguito, mentre W_{ci} , W_r e W_{coff} sono relative la velocità del vento di *cut-in*, *rated* e di *cut-off*. Graficamente rappresentato di seguito [28]:

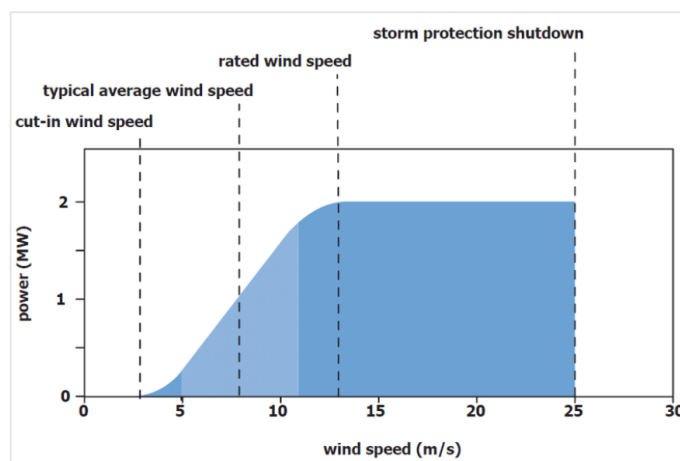


Figura 32 Curva di potenza turbina eolica. [28]

A causa dell'aleatorietà del vento la sua velocità dovrà essere espressa da una distribuzione di probabilità, quella che nel miglior modo la rappresenta è la distribuzione di Weibull la cui equazione classica è la (3.12) che riferita al caso di un aerogeneratore sarà pari a:

$$f_w(w) = \frac{k}{c} \left(\frac{w}{c}\right)^{k-1} e^{-\left(\frac{w}{c}\right)^k}$$

4. 4

Che seguirà la rappresentazione seguente:

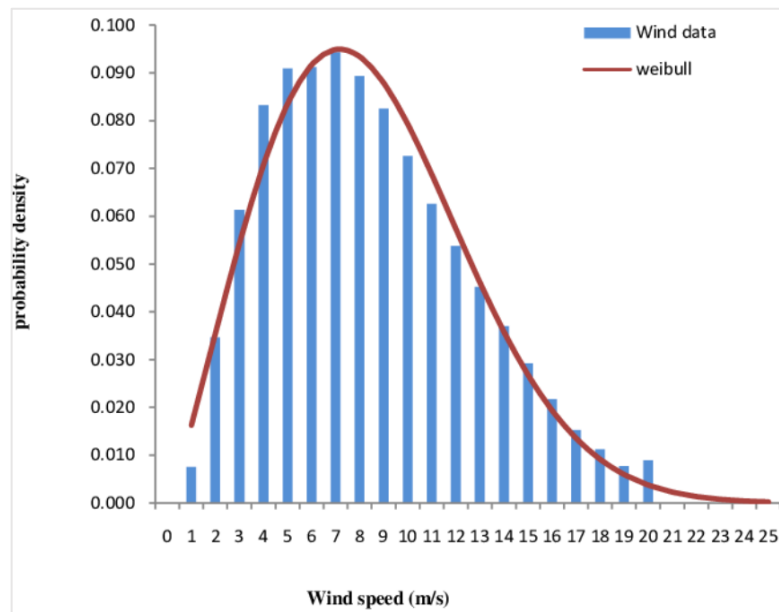


Figura 33 Distribuzione di Weibull confrontata con una serie di dati di ventosità. [28]

Per la trattazione del parametro di forma si manterrà un valore di $k=2$ mentre per il parametro di scala c esso potrà essere determinato attraverso due modalità. La prima, meno precisa ma più semplice da applicare nei casi come quelli trattati nel progetto di tesi e consiste nel considerare c pari a [31]:

$$c \cong 1.128 \mu_w$$

4. 5

nel secondo caso invece potrà essere calcolata con:

$$c = \frac{\mu_w}{\Gamma\left(1 + \frac{1}{k}\right)}$$

4. 6

Con $\Gamma(\bullet)$ pari alla *Gamma function*. L'utilizzo di una singola distribuzione di *Weibull* permette di racchiudere la maggior parte delle casistiche di ventosità presenti ma non è completamente generalizzata, per portare al minimo l'errore di stima si dovrebbe utilizzare due distribuzioni di *Weibull* oppure una distribuzione di *Weibull* e una *Distribuzione normale* troncata. La potenza reattiva come anticipatamente detto dipende dalla tipologia di generatore presente, vi sono tre diverse categorie [28]:

- *Aerogeneratori a velocità fissa*, sono *IG (Induction Generator)* direttamente connessi alla rete, alimentati dalla turbina eolica con la possibilità di variazione dell'angolo delle pale attraverso un *pitch control*, l'equazione è la seguente:

$$Q_{WT_i} = \left(\frac{V_t^2 (X_c - X_m)}{X_c X_m} + \frac{X}{V_t^2} P_{WT_i} \right)$$

4. 7

Ottenuta semplificando le perdite e considerando solo le reattanze, otteniamo che V_t , X_c , X e X_m sono rispettivamente la tensione, la reattanza del banco di condensatori (utile al miglioramento del fattore di potenza), la reattanza di dispersione (sia di statore che di rotore) e la reattanza di magnetizzazione [28];

- *Aerogeneratori a velocità semi-variabile*, la situazione risulta essere più complicata da analizzare in quanto oltre alla resistenza R abbiamo anche lo scorrimento s , fortunatamente per i dati trovati in letteratura è stato possibile semplificare la formulazione e quindi anche il controllo attraverso l'uso di una resistenza equivalente $R_{eq} = R_r / s$, tuttavia tali tipi di generatori non verranno presi in considerazione [28];
- *Aerogeneratori a velocità variabile*, la potenza reattiva risulta essere completamente controllata dal convertitore di interfacciamento e per tale ragione sarà settata a priori:

$$Q_{WT_i} = Q^{sp}$$

4. 8

Sarà proprio quest'ultima ad essere utilizzata in fase di progetto nel caso preso in analisi in tesi [28];

4.1.8 Modellizzazione degli impianti fotovoltaici

La modellizzazione di tali tipi di impianto è dettata da due indicatori principali:

- l'irraggiamento (I_β), viene espresso in $\left[\frac{W}{m^2}\right]$ e consiste nella radiazione solare incidente sul pannello fotovoltaico (o solare che sia) che cede tutta o parte della propria energia così da poter produrre energia elettrica, l'energia che non arriva alla superficie della terra viene definita estinta ed è formata dalla radiazione riflessa, riemessa e retro diffusa verso lo spazio. L'intensità di irraggiamento solare all'esterno dell'atmosfera è di $1353 \left[\frac{W}{m^2}\right]$ (Costante solare). Sulla superficie terrestre, a livello del mare, in condizioni meteorologiche favorevoli e a mezzogiorno, l'irraggiamento solare I è convenzionalmente pari a $1 \left[\frac{kW}{m^2}\right]$, rispetto alla costante solare, si ha una riduzione legata alla riflessione e al parziale assorbimento di energia da parte dell'atmosfera. [32] Secondo la legge di Lambert la quantità di radiazioni che colpisce l'unità di superficie è proporzionale al coseno dell'angolo di incidenza:

$$I_\beta = I \cos(\beta)$$

4. 9

- il valore di picco lo si avrà con incidenza perpendicolare ($\beta = 90^\circ$);
- l'indice di chiarezza oraria (k_t), viene definito come il rapporto tra l'irraggiamento solare globale sulla terra (su di una superficie orizzontale) e l'irraggiamento solare extraterrestre. Tale valore spazia tra 0 e 1 ed è adimensionale, in letteratura oltre al principale si possono ritrovare anche l'indice di chiarezza oraria medio giornaliero e mensile. [29]

Una volta poste queste premesse è possibile enunciare che quando l'impianto fotovoltaico è equipaggiato con un algoritmo MPPT (*Maximum Power Point Tracking*) l'output di potenza attiva può essere espresso come una funzione lineare che lega la potenza attiva con l'irraggiamento sulla superficie (I_β), considerando un angolo di inclinazione dei raggi incidenti β , vediamo dunque di seguito [28]:

$$P_{Pvt} = S_c \eta I_\beta$$

4. 10

dove S_c è la superficie totale degli array fotovoltaici e η il rendimento totale dell'impianto fotovoltaico. L'eq. (4. 10) risulta essere di facile applicazione in quanto la relazione che lega la potenza in uscita con la variabile aleatoria è lineare, per tale ragione essa verrà utilizzata in fase preliminare di progetto per l'analisi della potenza in output ottenibile e la creazione dei successivi modelli probabilistici. Più precisa ma anche più complicata da implementare è invece l'analisi della potenza in uscita dall'impianto basata sull'indice di chiarezza oraria (k_t). Causa maggiore di questa complicazione è la non linearità notevole nell'eq. seguente [28]:

$$P_{Pvt} = S_c \eta (T k_t - T' k_t^2)$$

4. 11

Dove con T e T' indichiamo:

$$T = \left[\left(R_b + \rho \frac{1 - \cos \beta}{2} \right) + \left(\frac{1 + \cos \beta}{2} - R_b \right) c_1 \right] r_d \left(\frac{H_o}{3600} \right)$$

4. 12

$$T' = \left(\frac{1 + \cos\beta}{2} - R_b \right) c_2 r_d \left(\frac{H_o}{3600} \right)$$

4. 13

Dove con H_o , r_d e R_b indichiamo rispettivamente la radiazione solare extraterrestre, la radiazione diffusa giornaliera e il rapporto tra la radiazione incidente sulla superficie inclinata rispetto a quella orizzontale. Per quanto riguarda invece i parametri c_1 e c_2 essi ci permettono di ottenere l'indice k a partire da k_t attraverso la formulazione seguente:

$$k = c_1 - c_2 k_t$$

4. 14

Per maggiori informazioni riguardo ai parametri c_1 e c_2 consultare in bibliografia [30]. Confrontando l'eq. (4. 10) con la (4. 11) è possibile notare che:

$$I_\beta = (T k_t - T' k_t^2)$$

4. 15

I_β e k_t sono dunque le uniche variabili che variano la generazione fotovoltaica in quanto le caratteristiche di impianto S_c e η sono determinate a priori. Lo scambio di potenza reattiva con la rete è controllata dagli inverter di interfacciamento (Impianto – Rete), essa viene trattata in fase di progetto con due modalità differenti:

- $Q_{Pvt} = Q^{sp}$, che consiste nel settare direttamente un valore di potenza da inseguire;
- $Q_{Pvt} = P_{Pvt} \frac{\sqrt{1 - (\cos\varphi^{sp})^2}}{\cos\varphi^{sp}}$, a partire da un valore di $\cos\varphi^{sp}$ che vogliamo mantenere il sistema andrà a settare la potenza reattiva;

A differenza delle classiche *pdf*, l'indice di chiarezza oraria (k_t) ne ha una in particolare proposta per la prima volta da *Hollands* e *Huget*:

$$f_{k_t}(k, C_t, \lambda_t) = \frac{C_t (\overline{k_{tu}} - k)}{\overline{k_{tu}}} e^{\lambda_t k_t}$$

4. 16

I parametri C_t e λ_t sono rispettivamente:

$$C_t = \frac{\lambda_t^2 \overline{k_{tu}}}{e^{\lambda_t k_t} - 1 - \lambda_t \overline{k_{tu}}}$$

4. 17

$$\lambda_t = \frac{2F_t - 17.519e^{-1.311F_t} - 1062e^{-5.0426F_t}}{\overline{k_{tu}}}$$

4. 18

con:

$$F_t = \frac{\overline{k_{tu}}}{\overline{k_{tu}} - \mu_{k_t}}$$

4. 19

$\overline{k_{tu}}$ e μ_{k_t} sono rispettivamente la banda superiore dell'indice di chiarezza oraria e il valore medio di k_t .
 . Dagli studi affrontati poi da parte di *Liu e Jordan* è emerso che due Pdf sono simili quando il loro valore medio è uguale [28]. Un'ulteriore metodologia per l'assunzione della potenza generata dall'impianto è attraverso l'equazione seguente:

$$P(G) = \begin{cases} P_{sr} \left(\frac{G^2}{G_{std} R_c} \right), & 0 \leq G < R_c \\ P_{sr} \left(\frac{G}{G_{std}} \right), & G \geq R_c \end{cases}$$

4. 20

Dove con G_{std} , R_c e P_{sr} indichiamo rispettivamente l'irraggiamento standard pari a $1 \left[\frac{kW}{m^2} \right]$, un particolare set-point di irraggiamento pari a $120 \left[\frac{W}{m^2} \right]$ e con P_{sr} la potenza nominale erogabile dall'impianto preso in analisi. L'equazione ci dice che al di sotto di un livello di irraggiamento specifico la proporzione non risulta più lineare, portando complicazioni di progetto, conseguentemente ci si ricondurrà alla seguente formula approssimata:

$$P(G) \cong P_{sr} \left(\frac{G}{G_{std}} \right)$$

4. 21

L'irraggiamento G verrà estratto da una distribuzione Log-normale [33] in quanto quest'ultima permette di descrivere molto bene la probabilità di avere un determinato valore di irraggiamento nell'arco della giornata, come possiamo vedere nella figura seguente:

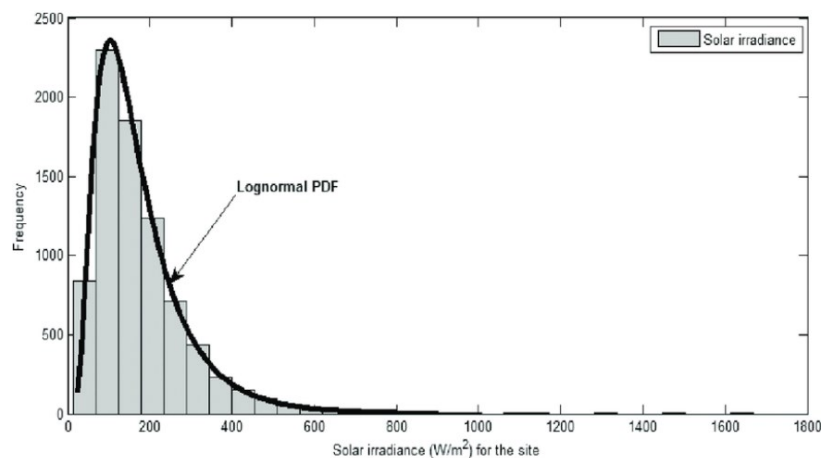


Figura 34 Distribuzione LogNormale per irraggiamento solare. [33]

Per il settaggio dei parametri della distribuzione, essa è determinata dalla media e dalla deviazione standard logaritmica, i quali vengono ottenuti a partire da media e deviazione standard classici dei dati prelevati con opportuni calcoli matematici descritti in precedenza nelle equazioni (3. 9) e (3. 10).

4.1.9 Generazione e assegnazione di distribuzioni di probabilità a partire da dati reali per la generazione da fonte rinnovabile su DigSilent.

Nell'interfaccia di *DigSilent*, la modellizzazione degli impianti di produzione di energia da fonte rinnovabile viene costruita attraverso l'utilizzo di *Static Generator*, l'esempio preso in analisi per la spiegazione è una rete a 41 nodi *IEEE modificata*.

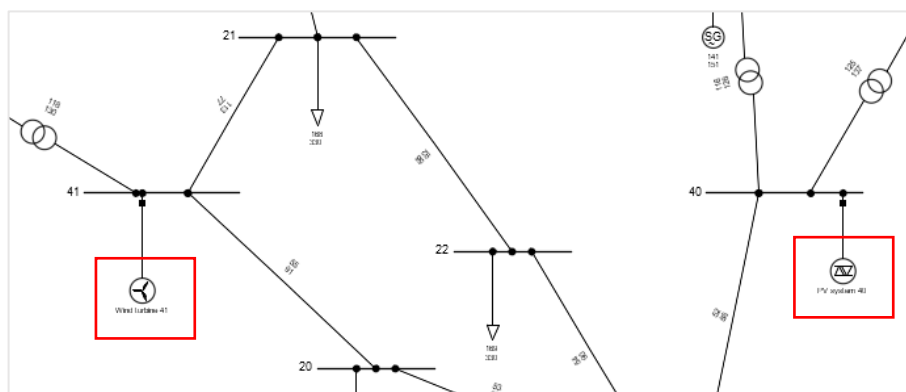


Figura 35 Rappresentazione grafica di generatori distribuiti, nel riquadro a Sx una Wind Turbine (*Static Generator*) e nel riquadro a Dx uno *Static Generator* che modella un PV System.

Iniziamo analizzando la modellizzazione degli aerogeneratori. Inizialmente, nel *Local Controller*, viene introdotto il parametro *Const. Q* che consente di considerare il nodo come *PQ* anziché *PV* durante il calcolo del *Power Flow*, nonostante esso sia un generatore. Per quanto riguarda invece il parametro *Model*, essendo desiderabile l'utilizzo di una funzione di distribuzione di *Weibull* per descrivere il profilo del vento, i dati di input saranno rappresentati dalla velocità del vento (*WS*) e dalla potenza reattiva (*Reactive Power*).

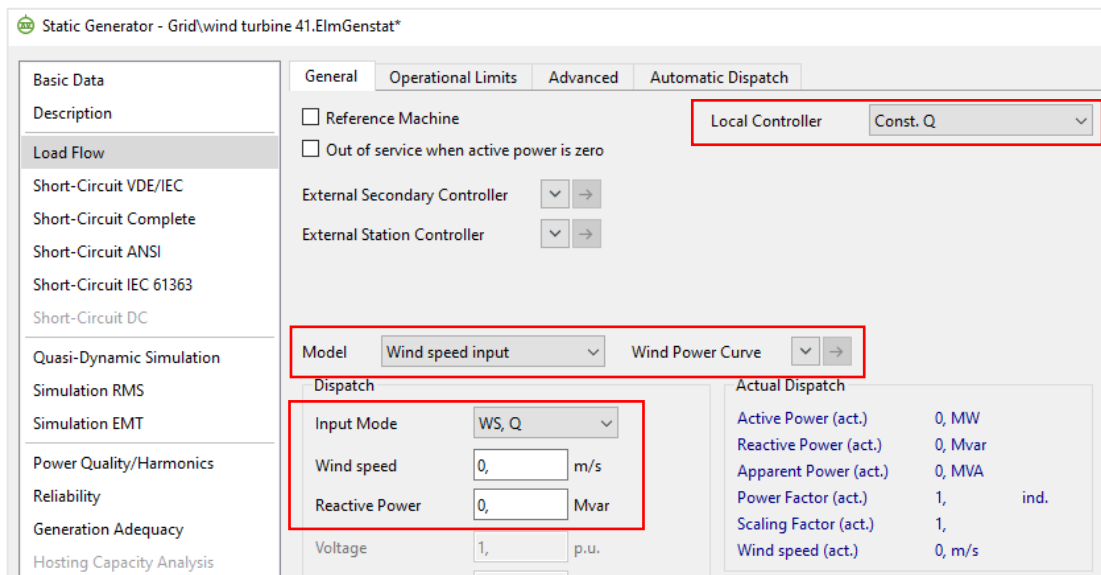
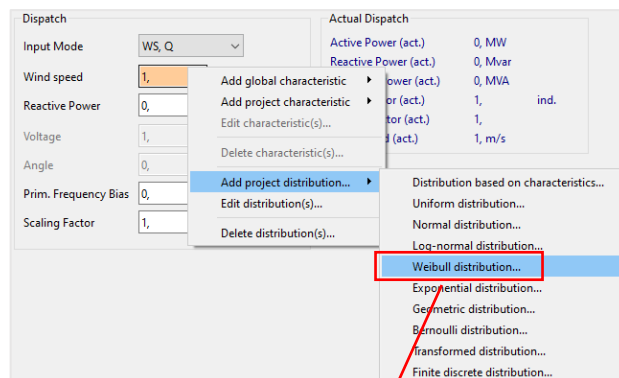


Figura 36 Menu di configurazione di una Wind Turbine (Static generator).

Per associare una distribuzione al nodo si dovrà fare click con il tasto destro del mouse all'interno della finestra *Wind speed [m/s]*, e vi si aprirà un ulteriore menu di selezione sulla destra, il quale ci fornirà diverse possibilità:

- *Add global characteristic*;
- *Add project characteristic*;
- *Add project distribution*, permette di associare al carico preso in analisi una funzione di distribuzione di probabilità;
- *Edit distribution*, permette di modificare nel carico selezionato le caratteristiche della funzione di distribuzione di probabilità associata nel campo *Add project distribution*;

Per prima cosa sceglieremo *Add project distribution* e successivamente si aprirà una larga scelta di distribuzioni utilizzabili, nelle casistiche prese da noi in analisi la scelta ricadrà sulla *Weibull distribution* e la *Log-normal distribution* motivo di tale scelta è dovuto alla possibilità di descrivere le fonti di energie rinnovabili eolica e fotovoltaica grazie a queste due [33] inoltre ci permette di effettuare un ottimo confronto con il codice da noi progettato su *MatLab*, di seguito viene rappresentato il processo di selezione della curva di distribuzione per la velocità del vento:



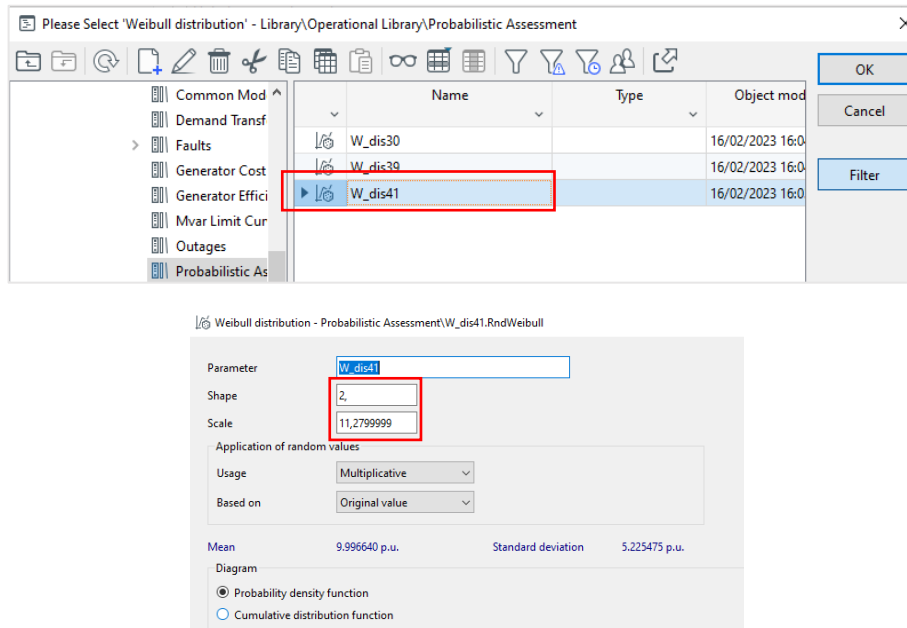


Figura 37 Processo di selezione della curva di distribuzione di Weibull per la velocità del vento di una turbina eolica.

Attraverso la formulazione (4.6), a partire dalla velocità media del vento, sarà possibile determinare il parametro di scala c della distribuzione di probabilità. Successivamente, si selezionerà il comando *multiplicative* in modo tale da avere i valori ottenuti dalle distribuzioni in $[p.u.]$, cosicché questi vengano moltiplicati con il valore indicato all'interno della finestra WS che sarà pari a 1. Tuttavia, non avremo ancora espresso la potenza in uscita dal generatore, poiché l'unità di misura è $[m/s]$ in quanto stiamo parlando di velocità del vento. A questo punto, sarà opportuno associare una *Wind Power Curve*, la quale funzionerà secondo il seguente processo:

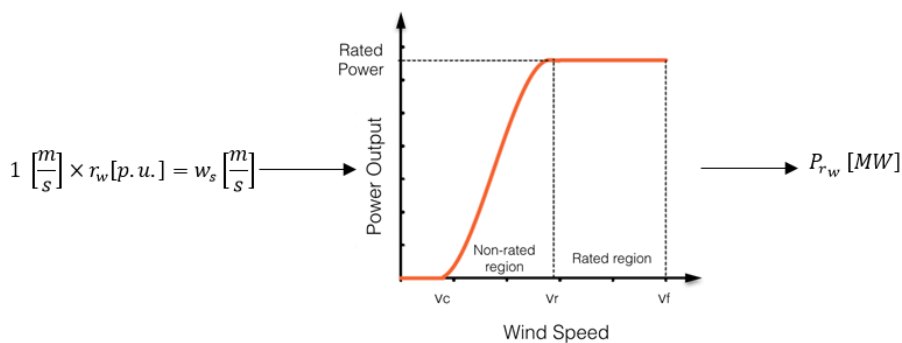


Figura 38 Processo di ottenimento dei valori di potenza in uscita da un aerogeneratore a partire dalle velocità del vento.

Facendo click con il tasto destro nel riquadro *Power Curve* sarà possibile aggiungere righe che sono costituite da coordinate $(ws [m/s], P [MW])$ in modo tale da poter descrivere l'intera curva. Risulterà fondamentale indicare *linear* come parametro di *approximation*.

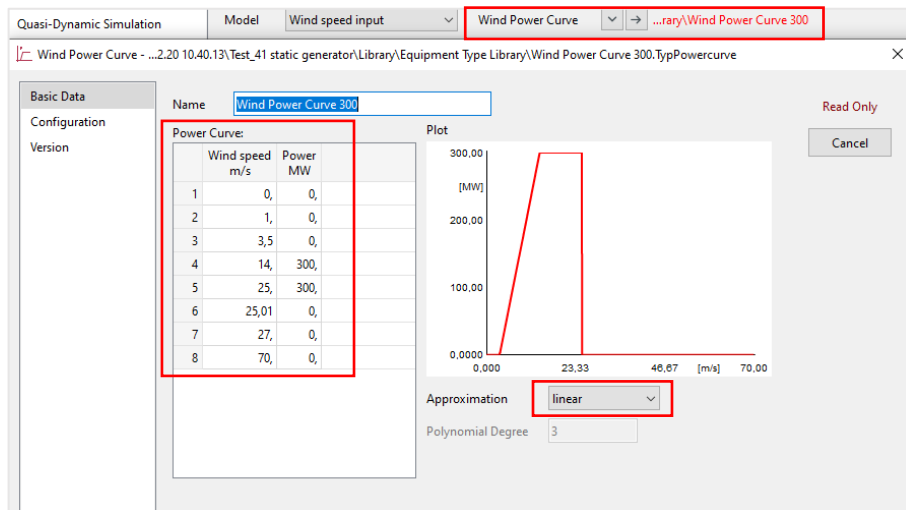


Figura 39 Power Curve di una turbina eolica.

Prendiamo ora in considerazione il caso di uno *Static Generator* che andrà a modellizzare un *PV System*, il *Local Controller* sarà configurato su *Const.Q* così da modellizzarlo come un nodo *PQ* come nel caso precedente, verranno inseriti i valori di potenza attiva e reattiva, in questa casistica la potenza attiva sarà divisa per mille e i valori in uscita dalla distribuzione non saranno variati in quanto già espressi nell'unità di misura corretta. Facendo click con il tasto destro sulla sezione *Active Power* si aprirà la finestra di selezione delle distribuzioni e per la casistica presa in analisi si sceglierà la *Log Normal distribution*.

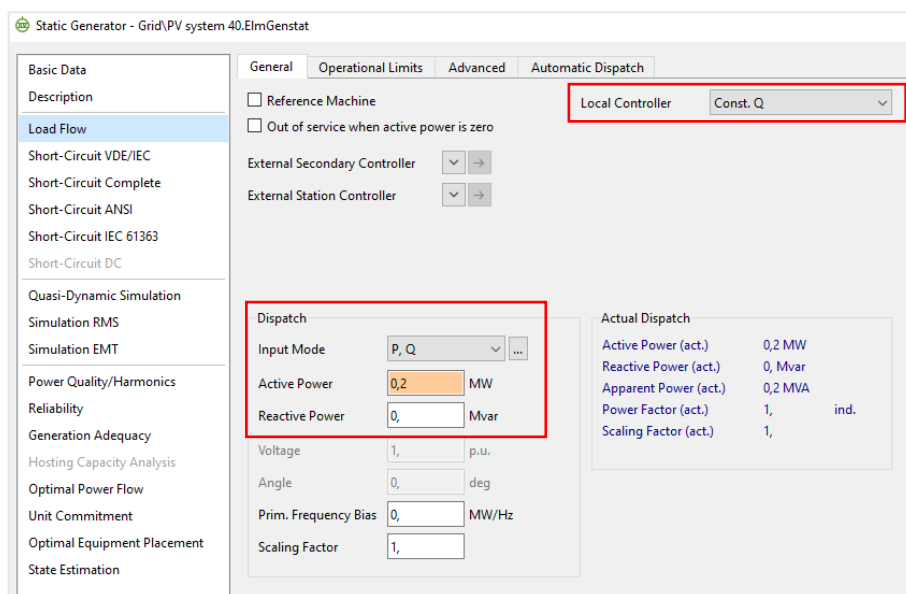


Figura 40 Menu di selezione del PV System per lo Static Generator.

A quel punto cliccandoci sopra si aprirà la finestra nella figura seguente, il riquadro in alto a sinistra più piccolo indica la possibilità di creare una nuova distribuzione di tale tipo mentre nella finestra principale si hanno le distribuzioni *Log-normali* fino ad ora create.

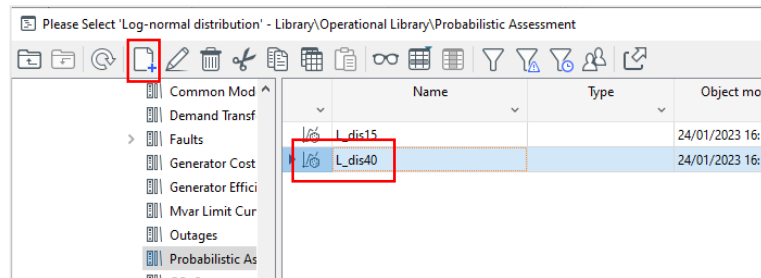


Figura 41 Menu di selezione della distribuzione Log-normale.

Cliccando con il tasto destro su una di queste due e selezionando *Edit* si passerà alla finestra di configurazione della distribuzione stessa, *Mean of logarithm* e *Std. of logarithm* sono determinate attraverso calcoli a priori a partire dai dati in analisi e vengono inseriti nel menu seguente come rappresentato di seguito:

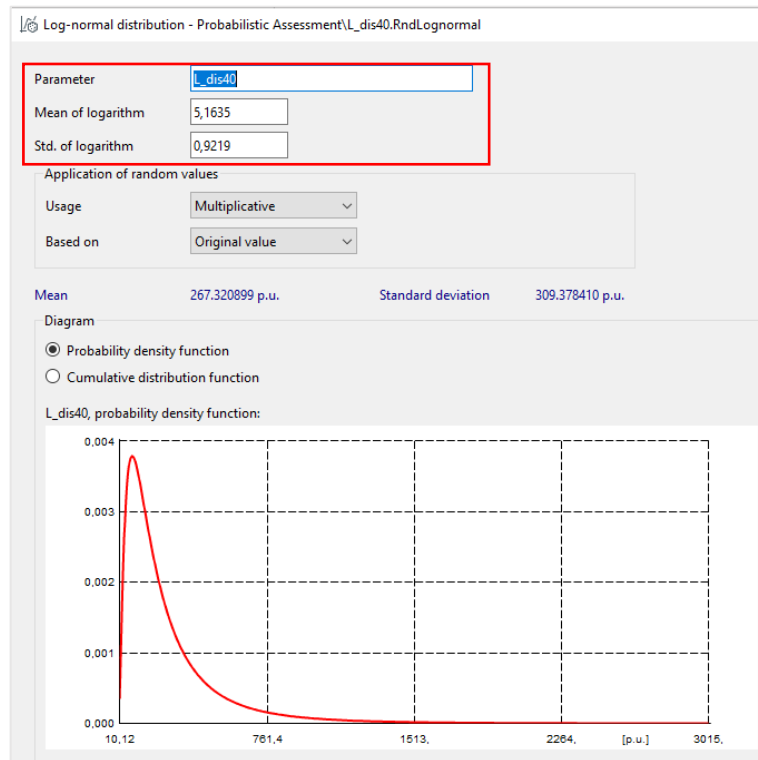


Figura 42 Menu di configurazione di una distribuzione Log-Normale.

in tal modo i valori in output dalla distribuzione saranno espressi in *[p.u.]* così che possano poi essere moltiplicati per i valori nominali di potenza dell'impianto, nel modo seguente:

$$P_{r_nodo} = r_s \times \frac{-P_{rated}}{100}$$

4.1.10 Lancio del Probabilistic Power Flow con Metodo di Montecarlo

Una volta configurata la rete con le impostazioni probabilistiche descritte nel paragrafo precedente, si può procedere alla simulazione attraverso l'utilizzo dell'analisi probabilistica. Per selezionare tale opzione, occorre cliccare sulla freccia posta in alto a destra, che mostrerà un menu a tendina con tutti gli algoritmi aggiuntivi disponibili in *DigSilent*. Scegliere l'algoritmo desiderato, che verrà indicato con un riquadro rosso nella figura seguente, per poi avviare la simulazione.

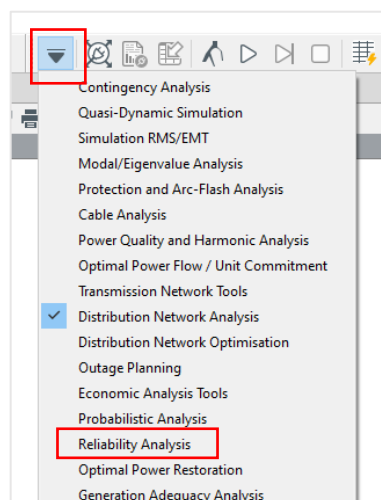


Figura 43 Menu di selezione a tendina per algoritmi aggiuntivi DigSilent.

Una volta selezionato *Probabilistic Analysis* il menu precedente verrà modificato assumendo la forma della figura seguente nella quale a sua volta selezioneremo il dado per procedere con il calcolo.



Figura 40 Comandi della Probabilistic Analysis.

La finestra che si aprirà di seguito sarà quella definitiva prima del lancio dell'algoritmo stesso e in cui dovremo definire il tipo di calcolo tra:

- *Load Flow (LF)*;
- *Optimal Power Flow (OPF)*;

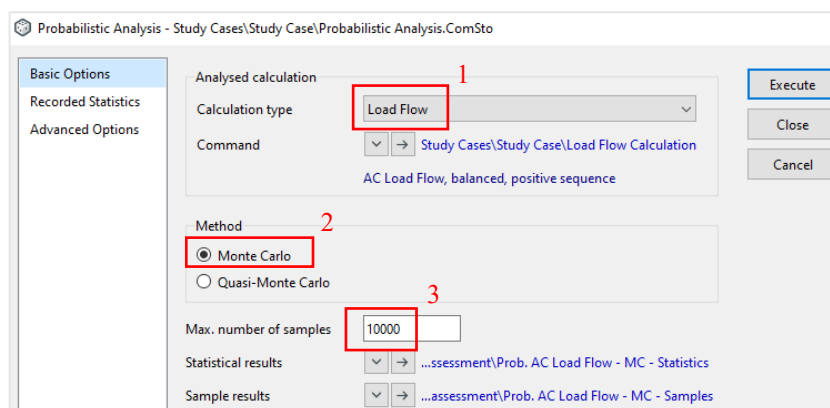


Figura 44 Parametri da selezionare nel caso di PPF.

Inoltre, sarà definita la tipologia di metodo, tra quelli disponibili, che andrà a determinare la distribuzione dei campioni casuali nella rete:

- *Monte Carlo*, distribuzione aleatoria;
- *Quasi-Monte Carlo*, distribuzione simmetrica;

Infine, si definirà il numero di iterazioni necessarie per eseguire la simulazione, che varierà da un minimo di *1000* a un massimo di *10000*, in modo da garantire una buona precisione senza però

appesantire eccessivamente i tempi di calcolo. Inoltre, per poter confrontare successivamente i risultati ottenuti con il software *DigSilent* con quelli ottenuti tramite l'algoritmo Patavino per la risoluzione dei flussi di potenza probabilistici (*PFPD_PRB*), si imposterà *Seeding type* su *Automatic* in modo che ogni lancio del *PPF* sia indipendente dal precedente. In questo modo, con 20 simulazioni *MCM* otterremo 20 risultati differenti (la motivazione di ciò è consultabile nella sezione "*Confronto dei risultati tra PFPD_PRB e DigSilent*").

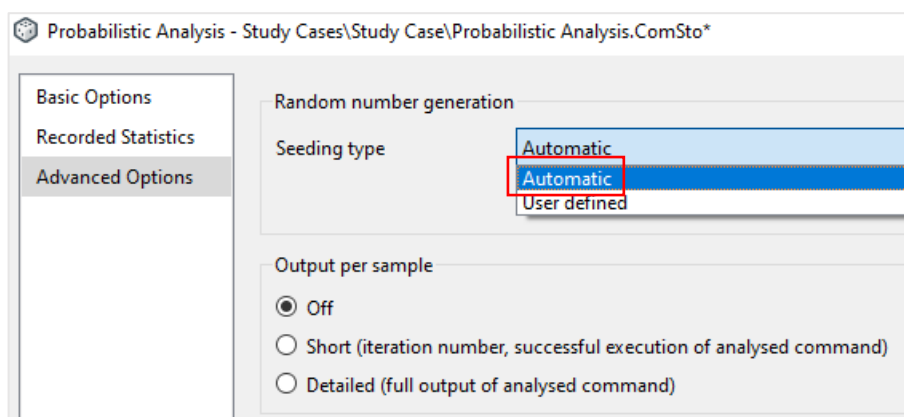


Figura 45 Selezione del tipo di generatore di numeri casuali.

I due diversi tipi di settaggi sono esposti di seguito:

- *User Defined*, il programma fornisce all'utente la possibilità di scegliere il *seed* da cui verrà generato il Dataset dei valori casuali di partenza, stesso *seed* corrisponde allo stesso Dataset;
- *Automatic*, il programma definisce autonomamente un *seed* diverso ad ogni *Probabilistic Analysis* cosicché ad ogni lancio l'algoritmo utilizzi un Dataset di valori diversi.

Un altro metodo per avviare la simulazione probabilistica in *DigSilent* è utilizzando il programma *Run_LF_&_Time_gen&car_Probabilistic*. Questo programma, oltre ad avviare la simulazione, fornisce anche il tempo di esecuzione dell'analisi. Per utilizzare questo programma, bisogna per prima cosa aprire il progetto di interesse in *DigSilent* (ad esempio *Test_41*, che corrisponde alla rete IEEE standard con 41 nodi), quindi aprire una volta ancora il menu *Project* e cliccare col tasto destro del mouse su *Run_LF_&_Time_gen&car_Probabilistic* per aprire il menu a tendina. Infine, selezionare l'opzione *Execute*, il codice avvierà la simulazione e fornirà il tempo impiegato per completarla. Il menu di selezione sarà mostrato nella figura seguente:

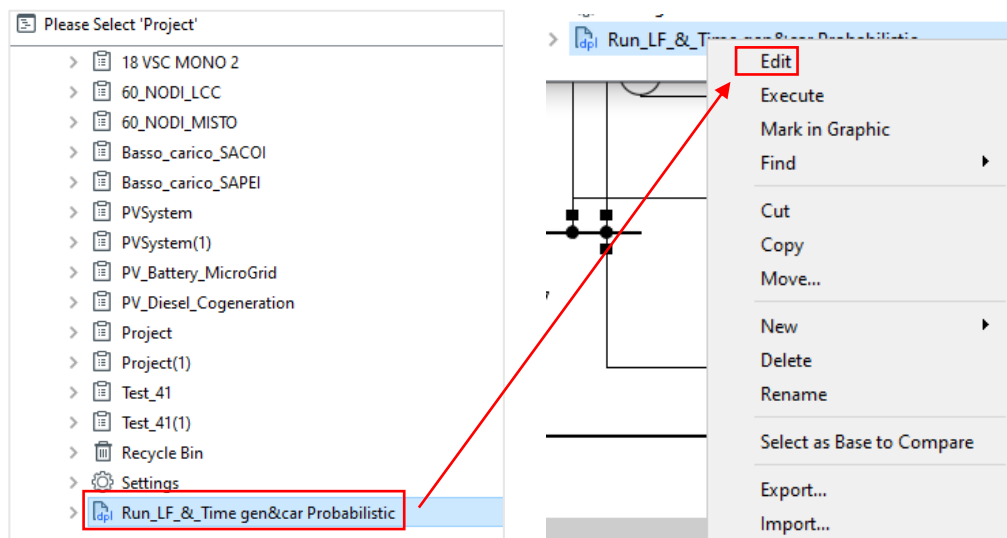


Figura 46 Menu di selezione del codice Run_LF_&_Time gen&car Probabilistic.

Cliccando su *Edit* sarà possibile accedere alla sezione di modifica del codice, che è suddivisa in due parti: la prima è la fase di inizializzazione e non richiede modifiche, mentre nella seconda è possibile modificare alcuni parametri come il *numSample* (che indica il numero di iterazioni che la *Probabilistic Analysis* eseguirà, di default impostato su *10000*), il *seedingType* (che di default è impostato su automatico attraverso il parametro *0*), il *method* (che di default è impostato sul metodo di Monte Carlo ovvero *0*, ma può essere cambiato in quello di Quasi Monte Carlo impostando il valore *1*) e il *calcAnalysed* (che indica il tipo di analisi che si vuole eseguire, con *0* che sta per *Load Flow* e *1* per *Optimal Power Flow*).

Con il comando *printf*, simile a quello presente in *MatLab* e altri linguaggi di programmazione, è possibile stampare a video il risultato desiderato. È importante notare che i termini sopracitati sono differenti per ogni tipo di simulazione (ad esempio, studio di guasti nelle reti, Load flow normali, ecc.) e vanno modificati opportunamente. Per conoscere il nome delle variabili, è possibile selezionare la simulazione desiderata e avvicinare il cursore sopra ogni parametro modificabile.

DPL Command - Run_LF_&_Time gen&car Probabilistic.ComDpl

Selecting a remote script destroys all current settings

Remote script [v] [→]

Code 1

```
double t;  
string str;  
string str1;  
double a;  
double final;  
a=GetTime(5);  
ExecuteCmd('sto calcAnalysed=0 method=0 numSamples=10000 seedingType=0');  
t=GetTime(5);  
final=t-a;  
printf('Probabilistic Power Flow calculation time, in seconds: %.4f',final);
```

2

Figura 47 Script del progetto di lancio della simulazione e della stampa a video dei risultati.

4.2 Implementazione del Probabilistic Power Flow su MatLab

L'implementazione del metodo patavino per il calcolo probabilistico segue il suo sviluppo a partire dal principio del *Metodo di Monte Carlo (MCM)*, quest'ultimo ampiamente citato nei capitoli precedenti consiste nel lanciare innumerevoli volte il codice Patavino con valori delle potenze associate ai generatori distribuiti differenti, presi a partire da curve di distribuzione di probabilità ad esse associate, se saranno 10000 i DLF all'interno di un solo PLF saranno anche 10000 differenti valori quelli associati

allo stesso elemento per ogni LF . Di seguito sono rappresentate le relative parti di codice con le opportune spiegazioni.

4.2.1 Generazione e assegnazione di distribuzioni di probabilità a partire da dati reali per la generazione da fonte rinnovabile

Per prima cosa verranno generate a partire da dati reali prelevati di ventosità e irraggiamento le potenze in [p.u.] dei generatori, rispettivamente in funzione del tipo di fonte di energia rinnovabile presa in considerazione eolico o fotovoltaico che sia, le function che effettueranno tale processo, sono le seguenti, r_w e r_s che esporremo di seguito, le motivazioni che hanno spinto alla scelta di distribuzioni di *Weibull* e *Log-Normali* sono citate nei capitoli precedenti (3.3.1 e 3.3.2):

- r_w , prende in input il numero di generatori eolici presenti nella rete (N_{gen_w}), i dati di ventosità del sito preso in analisi ($Data_wind_gen$) e il numero di iterazioni MC che il codice andrà a svolgere (set), fornendo in output i valori in [p.u.] delle potenze generabili dalle turbine eoliche prese in analisi attraverso un vettore di nome r_w per l'appunto con dimensione [$set \times 1$] e i valori dei parametri c della funzione di distribuzione di *Weibull*.

```
function [r_w,c_parameter] = r_w(N_gen_w,Data_wind_gen,set)
% Calcolo automatico dei parametri di scala delle distribuzioni di Weibull
% a partire dai dati del vento
c_parameter=[];
k=2; % Rimane costante uguale a 2 per garantire la forma della distribuz
% di Weibull.
for i=1:size(Data_wind_gen,1)
c=1.168*Data_wind_gen(i,1);
c_parameter(i,1)=c;
cas=wblrnd(c_parameter(i,1),k,set,1);
r_w(:,i)=cas;
end
```

Figura 48 Codice generatrice da distribuzione di Weibull per turbine eoliche.

la function che ricopre la parte più importante del codice è *wblrnd* (*Weibull random*) è una funzione generatrice di valori casuali a partire da una distribuzione di *Weibull*. Una volta ottenuti i valori aleatori di ventosità essi verranno dati in pasto alla function *power_curve_wind* che rappresenta la curva di potenza di un aerogeneratore:

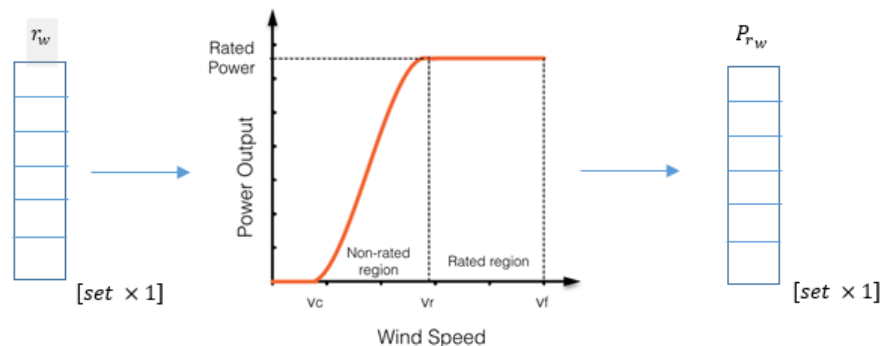


Figura 49 Metodo di creazione del vettore delle potenze a partire dal vettore di ventosità.

Il seguente script `power_curve_wind` prende in input un vettore di ventosità aleatorie e restituisce un vettore di potenze associate alla curva di potenza del generatore eolico utilizzato. Lo script di `power_curve_wind` è riportato nella figura seguente:

```
function [P_r_w] = power_curve_wind(set, r_w, P_rated_1, P_rated_2, P_rated_3)

% Inizializza P_r_w a zero
P_r_w = zeros(set, 3);

% Applica le condizioni sulla funzione di potenza in un unico passaggio vettoriale
r_w(r_w < 3.5) = 0;
r_w(r_w > 3.5 & r_w < 14) = 0.095*r_w(r_w > 3.5 & r_w < 14) - 0.3333333;
r_w(r_w > 14 & r_w < 25) = 1;
r_w(r_w > 25) = 0;

% Calcola la funzione di potenza
P_r_w(:,1) = P_rated_1 .* r_w(:,1);
P_r_w(:,2) = P_rated_2 .* r_w(:,2);
P_r_w(:,3) = P_rated_3 .* r_w(:,3);
```

Figura 50 Script di generazione di P_{r_w} a partire da r_w attraverso l'utilizzo di una curva di potenza.

- r_s , prende in input il numero di generatori fotovoltaici presenti nella rete N_{gen_s} , i dati di irraggiamento del sito preso in analisi Ig_{hourly} e il numero di iterazioni MC che il codice andrà a svolgere set fornendo in output i valori in $[p.u.]$ delle potenze generabili dai campi fotovoltaici di nome r_s per l'appunto con dimensione $[set \times 1]$ e i valori dei parametri μ e σ della funzione di distribuzione *LogNormale*.

```
function [r_s,mu,sigma] = r_s(Ig_hourly,N_gen_s,set)
% Calcolo automatico dei parametri di scala delle distribuzioni di Weibull
% a partire dai dati del vento
for i=1:size(Ig_hourly,2)
    m(i,1) = mean(Ig_hourly(:,i)); % mean
    v(i,1) = var(Ig_hourly(:,i)); % variance
    mu(i,1) = log((m(i,1)^2)/sqrt(v(i,1)+m(i,1)^2)); % parametri della dist
    sigma(i,1) = sqrt(log(v(i,1)/(m(i,1)^2+1))); % parametri della dist
    r_s(:,i) = lognrnd(mu(i,1),sigma(i,1),set,1);
end
```

Figura 51 Codice generatrice da distribuzione *LogNormale* per impianti fotovoltaici.

A partire dai dati dell'irraggiamento verrà calcolata la media e la deviazione standard dei dati in input, successivamente saranno calcolati i parametri μ e σ della distribuzione stessa, quest'ultimi saranno poi forniti alla function `lognrnd` (*LogNormal random*) che rappresenta la funzione generatrice di valori casuali a partire da una distribuzione *LogNormale*.

Dopo questa prima fase di generazione si passerà all'associazione dei valori appena determinati ai relativi generatori, si creeranno a partire dai valori casuali generati precedentemente i vettori contenenti le potenze possibilmente generabili come di seguito:

$$P_{r_{-n.nodo}_{[set \times 1]}} = r_{w[set \times 1]} \times P_{rated}$$

$$P_{r_n.nodo_{[set \times 1]}} = r_{s[set \times 1]} \times \frac{-P_{rated}}{100}$$

4. 24

Con P_{rated} indichiamo la potenza nominale dell'impianto con un segno meno davanti in quanto nodi PQ ma che generano potenza attiva invece che assorbirla, l'eq. (4. 3), contiene una divisione per 100 per motivi di convalidazione con *DigSilent*. Di seguito la rappresentazione su *MatLab*, il caso preso in esempio è quello di una rete a 41 nodi *IEEE* standard con 5 carichi convertiti in elementi di generazione distribuita.

```

%% Calcolo valore eolico
N_gen_w = 3; % Numero di generatori eolici
Data_wind_gen=[4.47 5.49 3.24]'./10;
[r_w,c_parameter] = r_w(N_gen_w,Data_wind_gen,set);
%% Calcolo valore fotovoltaico
N_gen_s=2; % Numero di generatori solari
Ig_hourly=[0 0 0 0 23.52 88.11 228.91 404.02 568.12 697.02 772.09 801.23 775.58 686.18 570.23 419.16 247.44 101.06 33.10 0 0 0 0 0
0 0 0 0 26.24 152.81 324.41 498.29 632.62 720.37 783.24 746.49 668.85 526.47 370.39 200.39 48.39 0.02 0 0 0 0 0]';
[r_s,mu,sigma] = r_s(Ig_hourly,N_gen_s,set);
%% Dichiarazione valori dei generatori distribuiti
% Numeri casuali generati a partire dalle distribuzioni sopracalcolate
P_r_41=r_w(:,1).*(-300e6);
P_r_40=r_s(:,1).*(-0.2e6);
P_r_39=r_w(:,2).*(-332e6);
P_r_15=r_s(:,2).*(-0.35e6);
P_r_30=r_w(:,3).*(-0.150e6);

```

Figura 52 Funzioni di generazione e creazione dei relativi vettori di potenza nel caso preso in esempio (Rete a 41 nodi *IEEE*).

4.2.2 Creazione delle Matrici CAR_GEN per ogni iterazione del $PFPF_PRB$

La fase precedentemente esposta era necessaria per progredire ora alla creazione delle matrici CAR_GEN per ogni iterazione MC , l'assegnazione viene fatta attraverso un processo di vettorializzazione, con 41, 40, 39, 15 e 30 indicheremo i nodi dei generatori distribuiti (e le rispettive righe nella matrice), mentre con la terza colonna della matrice CAR_GEN la potenza attiva che andremo volta per volta a modificare come svolto di seguito:

```

%% Calcolo della matrice multidim CAR_GEN_mdin
% Tale matrice è costituita da un numero di colonne pari a set(n.sim MC)
% attraverso l'operazione di vettorializzazione seguente potremmo avere
% l'associazione
CAR_GEN_mdin(:, :, 1:set) = repmat(CAR_GEN, 1, 1, set);
CAR_GEN_mdin(41, 3, :) = reshape(P_r_41, 1, 1, set);
CAR_GEN_mdin(40, 3, :) = reshape(P_r_40, 1, 1, set);
CAR_GEN_mdin(39, 3, :) = reshape(P_r_39, 1, 1, set);
CAR_GEN_mdin(15, 3, :) = reshape(P_r_15, 1, 1, set);
CAR_GEN_mdin(30, 3, :) = reshape(P_r_30, 1, 1, set);

```

Figura 53 Ciclo for per la creazione di tutte le CAR_GEN della rete, di dimensioni pari al numero di iterazioni di una simulazione MC.

Il comando *repmat* permette di ripetere la matrice CAR_GEN generata attraverso il codice $A_INTERFACCIA_IRPE.m$ attraverso la terza dimensione pari a *set* della nostra matrice CAR_GEN_mdim , successivamente i vettori P_r_x creati nelle sezioni precedenti attraverso il comando *reshape* verranno modificati di dimensione e resi dei vettori tridimensionali $[1,1,set]$ in modo tale da poter essere immagazzinati all'interno della matrice CAR_GEN_mdim .

4.2.3 Calcolo preliminare del Probabilistic Power Flow

Una volta calcolata la matrice multidimensionale CAR_GEN_mdim , che contiene tutte le CAR_GEN di ogni singola rete e ha dimensioni $[N_nodi_tot \times 6 \times set]$, si procederà ad utilizzare la funzione $A_Yretebase_nolosses$. Questa funzione permetterà la creazione della matrice topologica di rete senza considerare le perdite Y_n , indicata nel codice come $Yretebase$, che risulterà essere costante in quanto la rete non presenterà variazioni geometriche.

```

%% Stima della Yrete_base senza perdite
[Yretebase] = A_Yretebase_nolosses(RETE, TRASF, TRASF_TRE, SHUNT, omegan, Nbase, N_nodi_tot);

```

Figura 54 Function per la creazione della matrice topologica senza perdite $Yretebase$.

In seguito, verrà eseguito il calcolo della stima delle potenze reattive dei generatori. Questo processo è stato vettorializzato mediante l'utilizzo di matrici/vettori, indicati come *mdim* quando possibile. Quando ciò non era possibile, è stato utilizzato un ciclo *parfor*, che comunque risulta più efficiente di un ciclo for normale.

```

%% Stima della potenza reattiva da assegnare ai nodi di generazione
V_assegnate_pu_mdim = CAR_GEN_mdim(1:N_nodi_gen,5,:)/CAR_GEN_mdim(1:N_nodi_gen,
6,:);
V_assegnate_tot_pu_mdim = [V_assegnate_pu_mdim;ones(N_nodi_tot - N_nodi_gen,1,
set)];
% Stima iniziale matrice Ysh
S_compl_pu_mdim = zeros(N_nodi_tot,1,set);
S_compl_pu_mdim(CAR_GEN_mdim(:,2),1,:) = conj((CAR_GEN_mdim(:,3,:)+1i*CAR_GEN_mdim
(:,4,:))/Nbase);
S_compl_pu_mdim(1:N_nodi_gen,1,:) = zeros(N_nodi_gen,1,set);
Y_car_gen_mdim=zeros(N_nodi_tot);
parfor i=1:set
Y_car_gen_mdim(:,i) = diag(S_compl_pu_mdim(:,i)/V_assegnate_tot_pu_mdim(:,i),
i).^2);
end
Y_car_gen_mdim = (Y_car_gen_mdim);
Segni = diag([-ones(N_nodi_gen,1);ones((N_nodi_tot-N_nodi_gen),1)]);
Segni(:,1:set)=repmat(Segni,1,1,set);
Y_car_gen_mdim = bsxfun(@times, Y_car_gen_mdim, Segni);
%Partizione della matrice Y_rete_base in quattro quadranti:
Ygg = Yretebase(1:N_nodi_gen,1:N_nodi_gen);
Ygl = Yretebase(1:N_nodi_gen, (N_nodi_gen+1):N_nodi_tot);
Ylg = Yretebase((N_nodi_gen+1):N_nodi_tot,1:N_nodi_gen);
Yll = Yretebase((N_nodi_gen+1):N_nodi_tot, (N_nodi_gen+1):N_nodi_tot);
Yll_mdim(:,1:set)=repmat(Yll,1,1,set);
YLL_mdim = Y_car_gen_mdim((N_nodi_gen+1):N_nodi_tot, (N_nodi_gen+1):N_nodi_tot,:);
%Somma delle due matrici sopra partizionate:
Ytotll_mdim = Yll_mdim + YLL_mdim;
ig_mdim=zeros(N_nodi_gen,1,set);
REA_val_mdim=zeros(N_nodi_gen,1,set);
parfor i=1:set
ig_mdim(:,i) = (Ygg - Ygl*(Ytotll_mdim(:,i)\Ylg))*V_assegnate_pu_mdim(:,i);
%Potenze reattive generatori
REA_val_mdim(:,i) = imag(V_assegnate_pu_mdim(:,i).*conj(ig_mdim(:,i)))
*Nbase);
end
% Sostituzione delle potenze reattive appena stimate
CAR_GEN_mdim(2:N_nodi_gen,4,:) = REA_val_mdim(2:N_nodi_gen,1,:);

```

Figura 55 Stima della potenza reattiva dei generatori attraverso l'uso della matrice topologica senza perdite.

Una volta stimati i valori delle potenze reattive della rete, verrà ricalcolata la matrice Y_n considerando anche le perdite, tramite la funzione $A_Yretebase$. Queste potenze reattive saranno poi utilizzate, come descritto nel primo capitolo, per il calcolo della potenza apparente al fine di creare la matrice Y_{sh} . Dopo di che, saranno effettuati i successivi calcoli matriciali multidimensionali fino ad arrivare alla determinazione della matrice $Zequi_mdim$.

```

%% Calcolo della Y_retebase considerando le perdite
[Yretebase_mdin,Yretebase]=A_Yretebase(RETE,TRASF,TRASF_TRE,SHUNT,omegan,Nbase,
N_nodi_tot,set);
% --- end ---
%% [Y_car_gen,Ylg,YLL,Yll_1,YglYll_1,Yequi,Zequi] = A03_Y_car_gen(CAR_GEN,
Yretebase,Ea);
V_assegnate_tot_pu_mdin = [V_assegnate_pu_mdin;ones(N_nodi_tot-N_nodi_gen,1,set)];
S_compl_pu_mdin = zeros(N_nodi_tot,1,set); % Crea un vettore con numero di righe
pari al numero di nodi totali del sistema
S_compl_pu_mdin(CAR_GEN_mdin(:,2),1,:) = conj((CAR_GEN_mdin(:,3,:)+i*CAR_GEN_mdin
(:,4,:))/Nbase); % Associa ad ogni nodo se presente il valore espresso in p.u. della
potenza complessa
Y_car_gen_mdin=zeros(N_nodi_tot,N_nodi_tot,set);
parfor i=1:set
    Y_car_gen_mdin(:, :, i) = diag(S_compl_pu_mdin(:, :, i) ./ V_assegnate_tot_pu_mdin
(:, :, i).^2);
end
Y_car_gen_mdin = (Y_car_gen_mdin);
Segni = diag([-ones(N_nodi_gen,1);ones((N_nodi_tot-N_nodi_gen),1)]);
Segni(:, :, 1:set)=repmat(Segni,1,1,set);
Y_car_gen_mdin = bsxfun(@times, Y_car_gen_mdin, Segni);
Y_car_gen_mdin(1,1,:) = 1e5 - 1i*1e5;
Ygg = Yretebase(1:N_nodi_gen,1:N_nodi_gen);
Ygg_mdin(:, :, 1:set)=repmat(Ygg,1,1,set);
Ygl = Yretebase(1:N_nodi_gen, (N_nodi_gen+1):N_nodi_tot);
Ygl_mdin(:, :, 1:set)=repmat(Ygl,1,1,set);
Ylg = Yretebase((N_nodi_gen+1):N_nodi_tot,1:N_nodi_gen);
Ylg_mdin(:, :, 1:set)=repmat(Ylg,1,1,set);
Yll = Yretebase((N_nodi_gen+1):N_nodi_tot, (N_nodi_gen+1):N_nodi_tot);
Yll_mdin(:, :, 1:set)=repmat(Yll,1,1,set);
YGG_mdin = Y_car_gen_mdin(1:N_nodi_gen,1:N_nodi_gen,:);
YLL_mdin = Y_car_gen_mdin((N_nodi_gen+1):N_nodi_tot, (N_nodi_gen+1):N_nodi_tot,:);
Ytotgg_mdin = YGG_mdin + Ygg_mdin;
Ytotll_mdin = Yll_mdin + YLL_mdin;
inv_A = arrayfun(@(x) inv(Ytotll_mdin(:, :, x)), 1:set, 'UniformOutput', false);
Yll_1_mdin = cat(3, inv_A{:});
YglYll_1_mdin=zeros(N_nodi_gen,N_nodi_car,set);
Yequi_mdin=zeros(N_nodi_gen,N_nodi_gen,set);
parfor i=1:set
    YglYll_1_mdin(:, :, i) = Ygl_mdin(:, :, i)/Ytotll_mdin(:, :, i);
    Yequi_mdin(:, :, i) = Ytotgg_mdin(:, :, i) - YglYll_1_mdin(:, :, i)*Ylg;
end
inv_B = arrayfun(@(x) inv(Yequi_mdin(:, :, x)), 1:set, 'UniformOutput', false);
Zequi_mdin = cat(3, inv_B{:});
% --- end ---

```

Figura 56 Generazione dei parametri Z_{equi_mdim} , $Y_{glYll_1_mdim}$, Y_{lg_mdim} , $V_{assegnate_pu_mdim}$, I_{x_mdim} , I_{l_mdim} , S_{carto_mdim} , Y_{LL_mdim} , Y_{equi_mdim} , CAR_GEN_mdim e $Y_{retebase_mdim}$ da utilizzare successivamente nel ciclo `parfor(2)`.

4.2.4 Ciclo *parfor* (1)

All'interno del ciclo *parfor* (1) viene effettuato prima il recupero delle matrici a partire da quelle multidimensionali in funzione del n. di iterazione che stiamo prendendo in considerazione (processo effettuato grazie all'utilizzo dell'indice *monti*).

```
parfor monti=1:size(r,1)
%% Recupero delle matrici usate per ogni ciclo dalle matrici mdim
Zequi=Zequi_mdin(:,:,monti);
YglYll_1=YglYll_1_mdin(:,:,monti);
Yll_1=Yll_1_mdin(:,:,monti);
Ylg=Ylg_mdin(:,:,monti);
V_assegnate_pu=V_assegnate_pu_mdin(:,:,monti);
Ix=Ix_mdin(:,:,monti);
Il=Il_mdin(:,:,monti);
Scarto=Scarto_mdin(:,:,monti);
YLL=YLL_mdin(:,:,monti);
Yequi=Yequi_mdin(:,:,monti);
CAR_GEN=CAR_GEN_mdin(:,:,monti);
Yretebase=Yretebase_mdin(:,:,monti);
En_sogno=[];
%% Ciclo patavino
tol = 1e-8;
tic
ciclo=0;
EN=[];
Eng_c=ones(N_nodi_gen,1);
% Precalcolo di una matrice che verrà utilizzata nel ciclo
Yll_1Ylg=Yll_1*Ylg;
% --- PFPD ---
```

Figura 57 Recupero delle matrici a partire dalle matrici multidimensionali per ogni iterazione.

Vi sarà poi il ciclo *PPFD* stesso che riportiamo di seguito con le opportune modifiche per l'adattamento al *PPF*:

```

%% Ciclo patavino
tol = 1e-8;
tic
ciclo=0;
Eng_c=ones(N_nodi_gen,1);
% Precalcolo di una matrice che verrà utilizzata nel ciclo

Yll_1Ylg=Yll_1*Ylg;
% --- PFPD ---
while any(abs(real(Scarto))>tol | abs(imag(Scarto))>tol)
ciclo = ciclo + 1;
%% [Eng_c,Eload] = tensioni_pre_ciclo(Zequi,YglYll_1,Yll_1,Ylg,V_assegnate_pu,Ix,
I1);
Eng = Zequi*(Ix - YglYll_1*I1);
Eng_c(1:N_nodi_gen,1) = V_assegnate_pu(1:N_nodi_gen,1).*exp(1i*angle(Eng(1:...
N_nodi_gen,1)));
Eload = -Yll_1Ylg*Eng_c + Yll_1*I1;
% --- end ---
En_sogno = [Eng_c;Eload];
%% [Ix,I1] = A_rinnovo_FCC(Eng_c,Eload,YLL,Yequi,YglYll_1);
I1 = -(diag(YLL)).*(1 - abs(Eload).^2)./conj(Eload);
Ix = Yequi*Eng_c + YglYll_1*I1;
Ixc_q = -1i*(imag(Eng_c.*conj(Ix))./conj(Eng_c));
Ix(2:N_nodi_gen,1) = Ixc_q(2:N_nodi_gen,1);
% --- end ---
%% [Scarto] = A06_SCARTI(CAR_GEN,En_sogno,Yretebase);
cccc = (CAR_GEN(:,1) == abs('C'));
Nciclo = En_sogno.*conj(Yretebase*En_sogno);
Pcar = CAR_GEN(cccc,3)/Nbase;
Qcar = CAR_GEN(cccc,4)/Nbase;
Scarto = Nciclo(cccc) + (Pcar + 1i*Qcar);
% --- end ---
if ciclo > 50
canc(monti,1)=0;
disp(['La simulazione n. ' num2str(monti) ' ha fallito il calcolo del Power
Flow']);
break;
end
T_loadflow = toc;
In = (Yretebase)*En_sogno; % Correnti
entranti/uscenti nei nodi (Vettore colonna), con tali nodi modellizzo la potenza
entrante e uscente dai nodi che viene data rispettivamente da carichi e generatori.
Nn = En_sogno.*conj(In)*Nbase; % Potenze complesse
entranti ai nodi (Vettore colonna).
Ntotg = - sum(Nn(1:N_nodi_gen)); % Potenza complessa
totale generata dai nodi.
Ncarichi = sum(Nn(N_nodi_gen+1:N_nodi_tot)); %#ok<PFGV> % Potenza
complessa totale assorbita dai nodi.
Pn=real(Nn);
Qn=imag(Nn);
P_loss = real(Ntotg) - real(Ncarichi);
Q_loss = imag(Ntotg) - imag(Ncarichi);
angolo_tensioni=angle(En_sogno)*180/pi;
Pn(1)=Pn(1)+Ploss_1+Ploss_2;
% Salvataggio dei dati ti Output
In_monti(:,monti)=In;
Nn_monti(:,monti)=Nn;
En_monti(:,monti)=En_sogno;
end
% --- end ---

```

Figura 58 Codice PFPD completamente esplicitato per renderlo adatto al PPF e all'utilizzo della function parfor.

Una volta concluso il ciclo PFPD i risultati di tensioni (En), correnti (In) e potenza apparente (Sn) dei nodi verranno salvate per ogni iterazione del PLF effettuata.

```

% Salvataggio dei dati ti Output
In_monti(:,monti)=In;
Nn_monti(:,monti)=Nn;
En_monti(:,monti)=En_sogno;
% --- end ---
end
En_monti_modulo=abs(En_monti);
In_monti_modulo=abs(In_monti);
Pn_monti=real(Nn_monti);
Qn_monti=imag(Nn_monti);
T_interfaccia = toc(start);
disp(['Il tempo impiegato per eseguire il Probabilistic Power flow è: ',...
      num2str(T_interfaccia)])
clear T_interfaccia
% Salvataggio di tutti i dati di Output nel percorso seguente
save('C:\Users\Nicco\OneDrive\Desktop\Doppio-parfor\Risultati prob\Risultati prob.
mat');

```

Figura 59 Immagazzinamento dei dati all'interno della matrice X_monti e operazioni matriciali per successive analisi.

La creazione di queste matrici viene effettuato seguendo un processo colonna per colonna, dove la prima colonna corrisponderà alla prima iterazione, la seconda colonna alla seconda e così proseguendo fino a raggiungere il numero di iterazioni definitivo e quindi il numero di colonne finale. Una volta svolto tale processo per En , In e Nn , verranno eseguite fuori dal ciclo direttamente sulle matrici come l'operazione che calcola il modulo del numero complesso (abs) e di estrazione della parte reale ed immaginaria della potenza apparente rendendo il costo computazionale del codice ancora più basso.

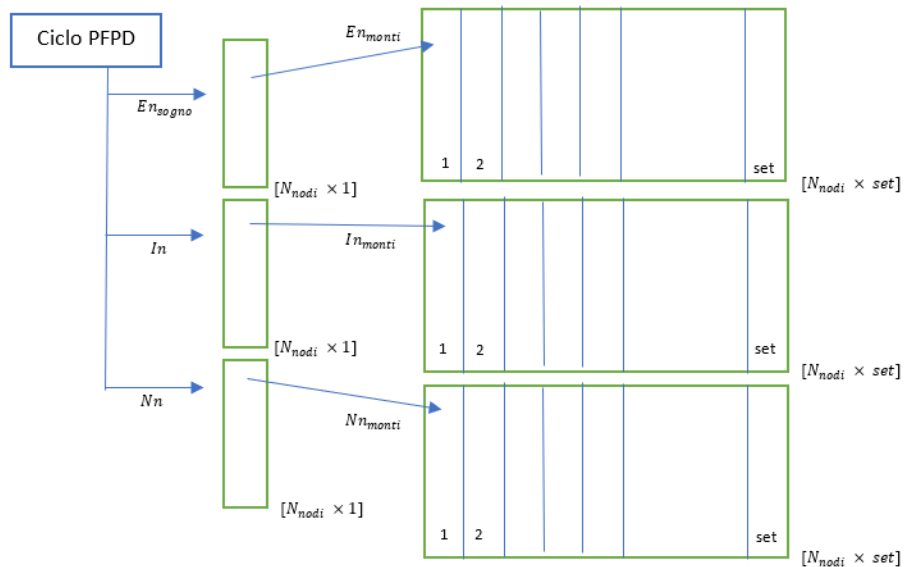


Figura 60 Creazione delle matrici En_monti , In_monti e Nn_monti costituite dai risultati.

Attraverso la function *mu_std*, saranno calcolate le medie e le deviazioni standard delle tensioni ai nodi e delle correnti sui rami della rete. I risultati saranno poi salvati in file Excel per poter essere confrontati con i risultati di DigSilent o elaborati per analisi successive.

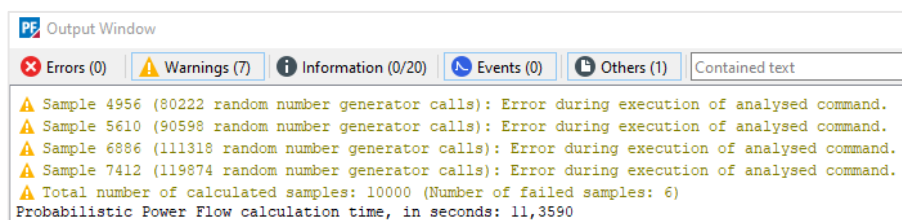
```
function [media_En_monti_modulo,media_In_monti_modulo,dev_En_monti_modulo, dev_In_monti_modulo] = mu_std(En_monti_modulo, In_monti_modulo)
% Calcolo delle medie e delle deviazioni standard
media_En_monti_modulo = mean(En_monti_modulo,2);
dev_En_monti_modulo = std(En_monti_modulo,0,2);
media_In_monti_modulo = mean(In_monti_modulo,2);
dev_In_monti_modulo = std(In_monti_modulo,0,2);

% Salvo su file excel le medie e le deviazioni standard
writematrix(media_En_monti_modulo, 'media_Tensioni.xlsx');
writematrix(dev_En_monti_modulo, 'deviazione_Tensioni.xlsx');
writematrix(media_In_monti_modulo, 'media_Correnti.xlsx');
writematrix(dev_In_monti_modulo, 'deviazione_Correnti.xlsx');
```

Figura 61 Codice di analisi probabilistica per lo studio e immagazzinamento dei risultati ottenuti.

4.2.5 Situazioni patologiche di non convergenza della rete

Importante verifica che deve essere presente nel codice *MatLab* è quella riguardante la convergenza e la non convergenza dell'algoritmo per ogni rete analizzata, tale processo è necessario poiché nel caso di reti che presentano una grande quantità di generatori da fonte rinnovabile può accadere che vi siano delle situazioni patologiche in cui nello stesso istante temporale vi sia o un sovraccarico di potenza o un abbassamento ingente della produzione di energia che il sistema stesso non riesce a reggere, tali situazioni causano una non convergenza dell'algoritmo della rete presa in analisi in quell'istante, l'insorgere di queste situazioni patologiche è direttamente proporzionale all'aleatorietà del sistema (n. di generatori distribuiti presenti nella rete) e al numero di iterazioni che vengono effettuate per ogni simulazione. Un esempio di tale fenomeno avviene simulando la rete *41 nodi IEEE modificata* inserendo 5 generatori da fonte rinnovabile, 3 eolici e 2 fotovoltaici. Per fare un parallelismo con *DigSilent* avremo che la simulazione fornirà i seguenti output:



```
PF Output Window
Errors (0) Warnings (7) Information (0/20) Events (0) Others (1) Contained text
Sample 4956 (80222 random number generator calls): Error during execution of analysed command.
Sample 5610 (90598 random number generator calls): Error during execution of analysed command.
Sample 6886 (111318 random number generator calls): Error during execution of analysed command.
Sample 7412 (119874 random number generator calls): Error during execution of analysed command.
Total number of calculated samples: 10000 (Number of failed samples: 6)
Probabilistic Power Flow calculation time, in seconds: 11,3590
```

Figura 62 Fallimento di 6 iterazioni su un totale di 10000 per la rete 41 nodi IEEE modificata.

DigSilent però non interrompe del tutto il calcolo probabilistico in quanto è consapevole che vi sarà la possibilità di avere delle situazioni patologiche e ne tiene semplicemente conto. La stessa cosa faremo noi nel nostro codice Matlab, attraverso la creazione del vettore *canc* costituito da un numero di righe pari al n. di simulazioni del metodo montecarlo e costituito solamente da 0 e 1, tale vettore verrà utilizzato poi successivamente all'interno di un ciclo *for* dove attraverso un ciclo *if* annidato ci permetterà di eliminare quelle colonne il cui numero corrisponde al numero di riga corrispondente all'operatore logico zero. Se effettuiamo 10000 simulazioni, la matrice *En_monti_modulo* sarà costituita da 10000 colonne se però avremo come nell'esempio superiore 6 simulazioni fallite allora quest'ultima sarà costituita invece che da 10000 da 9994 colonne e su di essere saranno calcolate medie e deviazioni standard.

(Aggiungi immagine codice)

(Inserisci immagine eliminazione colonne)

Come possiamo notare *DigSilent* non interromperà completamente la simulazione alla prima casistica di non convergenza ma passerà al campione successivo concludendo il *PLF* ed evidenziando le iterazioni per cui non è stato possibile raggiungere la convergenza ed eliminando quest'ultime dai risultati ottenibili cosicché i calcoli possano avvenire solo su quelle reti non patologiche ed effettivamente possibili. Lo stesso principio lo applicheremo sul codice *MatLab* attraverso le seguenti modifiche:

```

if ciclo > 50
    canc(monti,1)=0;
    disp(['La simulazione n. ' num2str(monti) ' ha fallito il calcolo del Power
Flow']);
    break;
end

```

Figura 63 Ciclo *if* annidato all'interno del ciclo *while* dell'algoritmo *PFPD* che mostra a video la non convergenza dell'algoritmo.

Come mostra la figura superiori si andrà ad inserire all'interno del ciclo *PFPD* un ciclo annidato *if* che verrà eseguito solo in caso di un numero di cicli interni per far convergere l'algoritmo superiore a 50 (solitamente la convergenza del *PF* è garantita per un numero di cicli pari a 10), una volta superati i cinquanta cicli verrà modificato all'interno del vettore $canc_{[set \times 1]}$ costituito solo da 1 l'elemento che si trova nella posizione *monti* in quella iterazione all'interno del ciclo, ponendolo a 0 invece che a 1. Attraverso il comando *disp* poi si mostrerà a video il seguente testo "La simulazione n. ha fallito il calcolo del Power Flow", il comando *break* ci porterà poi fuori dal ciclo *while* che contiene il *PFPD* stesso riportandoci al ciclo *parfor* (2) esterno contenente l'iterazione del metodo *MC* successiva e di conseguenza alla rete successiva.

```

function [En_monti,In_monti,Nn_monti] = elimina_colonne(En_monti,In_monti,Nn_monti,
canc)
% Elimina le colonne di En_monti, In_monti e Nn_monti di numero uguale alla
posizione degli eventuali zeri in canc

idx = find(canc == 0); % Trova gli indici degli zeri in canc
if ~isempty(idx)
    En_monti(:, idx) = []; % Elimina le colonne corrispondenti
    In_monti(:, idx) = []; % Elimina le colonne corrispondenti
    Nn_monti(:, idx) = []; % Elimina le colonne corrispondenti
end
end

```

Figura 64 Eliminazione a valle del calcolo di tutti i Power flow dei risultati della rete per cui non vi è stata la convergenza.

A valle di tutto l'algoritmo poi si richiamerà la function *elimina_colonne*, la quale attraverso l'uso del vettore $canc_{[set \times 1]}$ precedentemente creato permetterà di eliminare quelle colonne appartenenti alle matrici X_monti relative alle reti patologiche di cui non è stato possibile avere la convergenza. Per completezza riportiamo di seguito l'esempio di una simulazione *MC* con mille iterazioni:

```

La simulazione n. 400 ha fallito il calcolo del Power Flow
Il tempo impiegato per eseguire il Probabilistic Power flow è:      2.4592

```

Figura 65 Output all'interno della command window di MatLab dell'algoritmo *PFPD_PRB* in caso di non convergenza di una rete.

Se andremo ad analizzare il vettore $canc_{[set \times 1]}$ noteremo che per la riga 480 si avrà il valore zero e che le dimensioni delle matrici X_monti saranno conseguentemente diminuite di uno.

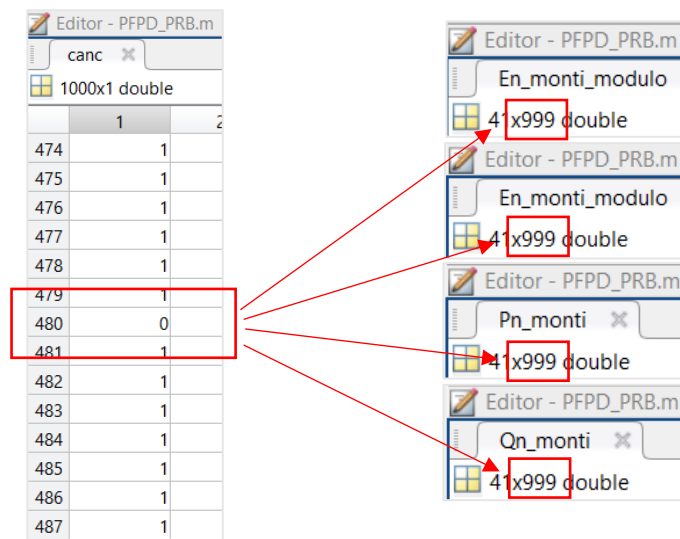


Figura 66 Assegnazione del valore zero al vettore *canc* in caso di non convergenza e correzione della dimensione matriciale.

4.3 Ottimizzazione dell'Algoritmo

L'ottimizzazione di un algoritmo *MatLab* è un processo importante che può migliorare notevolmente la velocità e l'efficienza di un programma. Ci sono alcune strategie generali che possono essere eseguite per migliorare le prestazioni del codice:

- *Vettorizzazione*: uno dei modi più efficienti per migliorare le prestazioni di un codice Matlab è quello di sfruttare l'elaborazione vettoriale. Matlab è progettato per elaborare array e matrici, e utilizzando la vettorializzazione è possibile evitare *cicli for* e rendere il codice più efficiente.
- *Funzioni built-in di MatLab*: *MatLab* fornisce molte funzioni built-in ottimizzate. Per esempio, è possibile utilizzare la funzione *inv* per calcolare l'inverso di una matrice, invece di scrivere il codice autonomamente;
- *Calcolo parallelo*: se è presente l'accesso a un'architettura di calcolo parallelo, si può utilizzare il *Parallel Computing Toolbox* di *MatLab* per parallelizzare il codice e utilizzare più core del processore;
- *Preallocazione della memoria*: la preallocazione degli array è un altro modo per migliorare le prestazioni di un codice *MatLab*. Invece di creare gli array man mano che si esegue il codice, è possibile preallocare gli array con le dimensioni adeguate in modo che non sia necessario ridimensionarli durante l'esecuzione del programma;
- *Profiler di Matlab*: la profilazione del codice può essere utilizzata per individuare le parti del codice che richiedono più tempo di esecuzione e che potrebbero essere ottimizzate ulteriormente. Matlab dispone di strumenti di profilazione integrati che possono aiutare a individuare queste sezioni del codice;
- *Riduzione della dimensione delle Matrici*: ridurre le dimensioni delle matrici presenti nel codice permette di ridurre il tempo di esecuzione del codice stesso;
- *Ottimizzazione dell'accesso alla memoria*: si cerca di accedere alla memoria in modo sequenziale così da ridurre il numero di volte in cui la *cache* del processore deve essere aggiornata;

4.3.1 Parallel Computing Toolbox

Sfruttando le tecniche di ottimizzazione menzionate in precedenza, è possibile raggiungere importanti miglioramenti nelle prestazioni del codice. È particolarmente importante prestare attenzione all'utilizzo del calcolo parallelo, che può essere implementato attraverso il tool *Parallel Computing Toolbox*TM [36], disponibile gratuitamente nella libreria. La function principale di questo processo è *parfor*, che può essere applicata con la seguente struttura:

```
parfor loopVar = initVal : endVal  
  
    statements;  
  
end
```

MatLab eseguirà il corpo del ciclo espresso sotto forma di *statements* per i valori dell'indice *loopVar* che andranno da *initVal* fino a *endVal*. Tale tool consente di acquisire il controllo dei processori *multicore* locali o della GPU per velocizzare il lavoro, sarà quindi possibile sfruttare tutte le potenze di elaborazione dei desktop *multicore* eseguendo applicazioni sui *worker* (motori di calcolo *MatLab*) in esecuzione localmente.

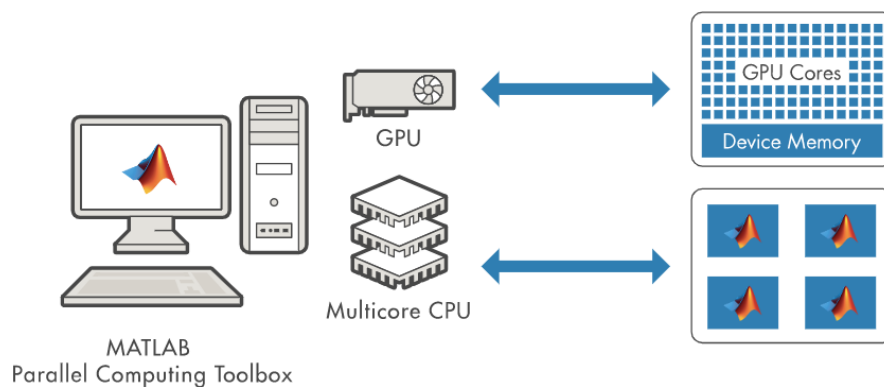


Figura 67 Processo di Parallel Computing. [36]

In particolare, come anticipato nella sezione precedente, saranno sviluppati due cicli *parfor*. Il primo ciclo *parfor* (1) verrà utilizzato per creare le matrici necessarie per eseguire il power flow. Tale processo ha un impatto ridotto sulle prestazioni per reti piccole e poche iterazioni, ma può risultare molto oneroso per reti estese e un elevato numero di iterazioni. Il secondo ciclo *parfor* (2) sarà utilizzato per eseguire il codice *PFPD*. In questo ciclo, ogni worker si occuperà delle matrici relative alla propria iterazione e eseguirà il codice *PFPD*, consentendo di eseguire più *Deterministic Load Flow* paralleli e ridurre notevolmente i tempi di esecuzione.

4.4 Flow chart dell' algoritmo PFPD_PRB

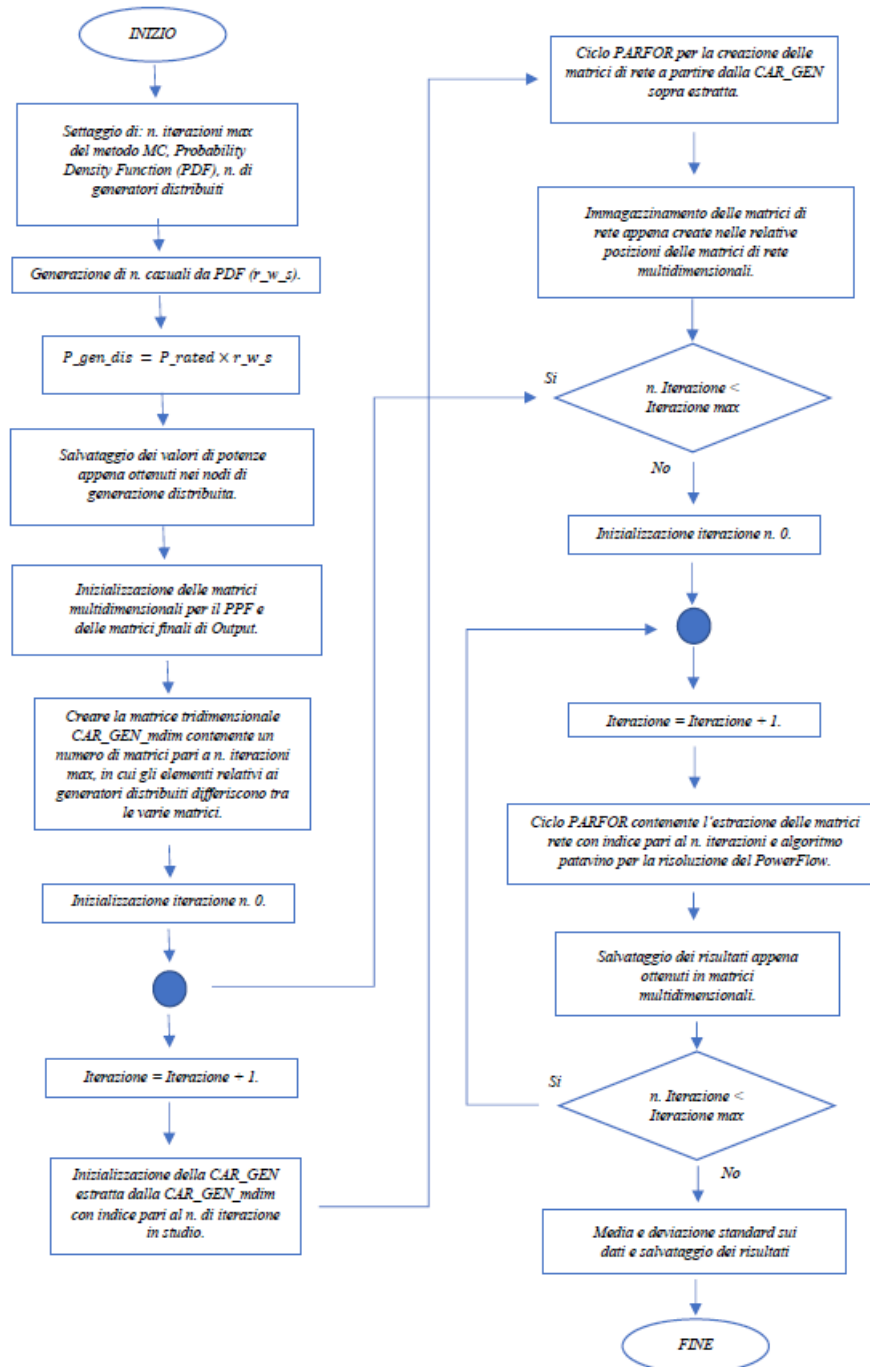


Figura 68 Flow Chart PFPD_PRB con metodo MC.

5. Confronto dei risultati tra *PFPD_PRB* e *DigSilent*

In conformità al titolo, il confronto si svolgerà tra *PFPD_PRB* e *DigSilent*, escludendo l'algoritmo *Open Source MatPower* a causa dell'impossibilità di condurre analisi probabilistiche della rete attraverso quest'ultimo. La comparazione tra algoritmi probabilistici non è un compito facile poiché ogni simulazione è indipendente dalle precedenti. Di conseguenza, anche se i risultati possono risultare simili a causa delle stesse condizioni di contorno, confrontare le due simulazioni e calcolare l'errore relativo non consente di effettuare un confronto adeguato. Prima di tutto, è necessario porre i due algoritmi nelle stesse condizioni. Ciò richiede che si rispettino i seguenti vincoli:

- Il medesimo numero di iterazioni per ogni simulazione Monte Carlo deve essere eseguito sia con *PFPD_PRB* che con *DigSilent*;
- Il medesimo algoritmo *Load Flow* o *Optimal Power Flow* e il medesimo metodo Monte Carlo o Quasi Monte Carlo devono essere impiegati;
- La stessa rete deve essere analizzata, importata in *MatLab* tramite l'uso del codice *A_INTERFACCIA_IRPE.m*, insieme ai relativi file di importazione da *DigSilent* a Excel;
- Le stesse funzioni di distribuzione devono essere applicate ai carichi/generatori tramite l'utilizzo degli stessi parametri di configurazione, ovvero il parametro di forma e di scala per la distribuzione di *Weibull* e la media e la deviazione standard logaritmica per la distribuzione *LogNormale*;

Dopo che i due programmi sono stati posti nelle medesime condizioni, si eseguiranno 20 simulazioni ciascuno. I risultati che saranno confrontati consistono nei valori di μ e σ dei moduli delle tensioni di tutti i nodi ottenuti ad ogni simulazione. Sarà necessario calcolare le medie di ogni simulazione e successivamente la media di questi valori per ottenere un vettore di valori per *MatLab* (*A*) e uno per *DigSilent* (*B*):

$$\begin{pmatrix} \mu_1^A \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mu_n^A \end{pmatrix} \quad \begin{pmatrix} \mu_1^B \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mu_n^B \end{pmatrix}$$

4. 25

Lo stesso processo verrà ripetuto per le deviazioni standard di ogni nodo, ottenendo così il seguente vettore:

$$\begin{pmatrix} \sigma_1^A \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \sigma_n^A \end{pmatrix} \quad \begin{pmatrix} \sigma_1^B \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \sigma_n^B \end{pmatrix}$$

4. 26

Dopo la creazione di questi due vettori, verrà effettuato il calcolo dell'errore relativo utilizzando la seguente formula:

$$E_{rel\%} = \frac{a-b}{a} \times 100$$

4. 27

che applicata allo studio da noi preso in esame risulta pari a:

$$E_{rel\%,\mu} = \frac{\mu_1^B - \mu_1^A}{\mu_1^B} \times 100 \quad E_{rel\%,\sigma} = \frac{\sigma_1^B - \sigma_1^A}{\sigma_1^B} \times 100$$

4. 28

I risultati verranno poi immagazzinati nei seguenti vettori il primo per le medie e il secondo per le deviazioni standard:

$$\begin{pmatrix} E_{rel\%,\mu_1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E_{rel\%,\mu_n} \end{pmatrix} \quad \begin{pmatrix} E_{rel\%,\sigma_1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E_{rel\%,\sigma_n} \end{pmatrix}$$

4. 29

Gli errori relativi saranno espressi in percentuale rispetto alle medie di μ e σ di ogni nodo. La scelta di utilizzare l'errore relativo è motivata dalla sua capacità di considerare l'ordine di grandezza delle grandezze confrontate, semplificando la valutazione delle differenze. Un'importante parametro di confronto sarà il tempo di elaborazione del codice, si utilizzerà per entrambe le simulazioni un PC con le seguenti caratteristiche:

- Processore, Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz;
- Memoria installata (RAM), 32.0 GB (31.9 GB utilizzabile);

Per concludere verrà dunque proposto di seguito una rete test su cui effettuare la simulazione e verificare l'affidabilità del codice.

5.1. Rete Test AC/DC 22 nodi con collegamento HVDC-VSC bipolare

La rete test utilizzata per il confronto iniziale degli algoritmi è costituita da una rete AC/DC di 22 nodi con un collegamento HVDC-VSC bipolare, illustrata nella figura seguente:

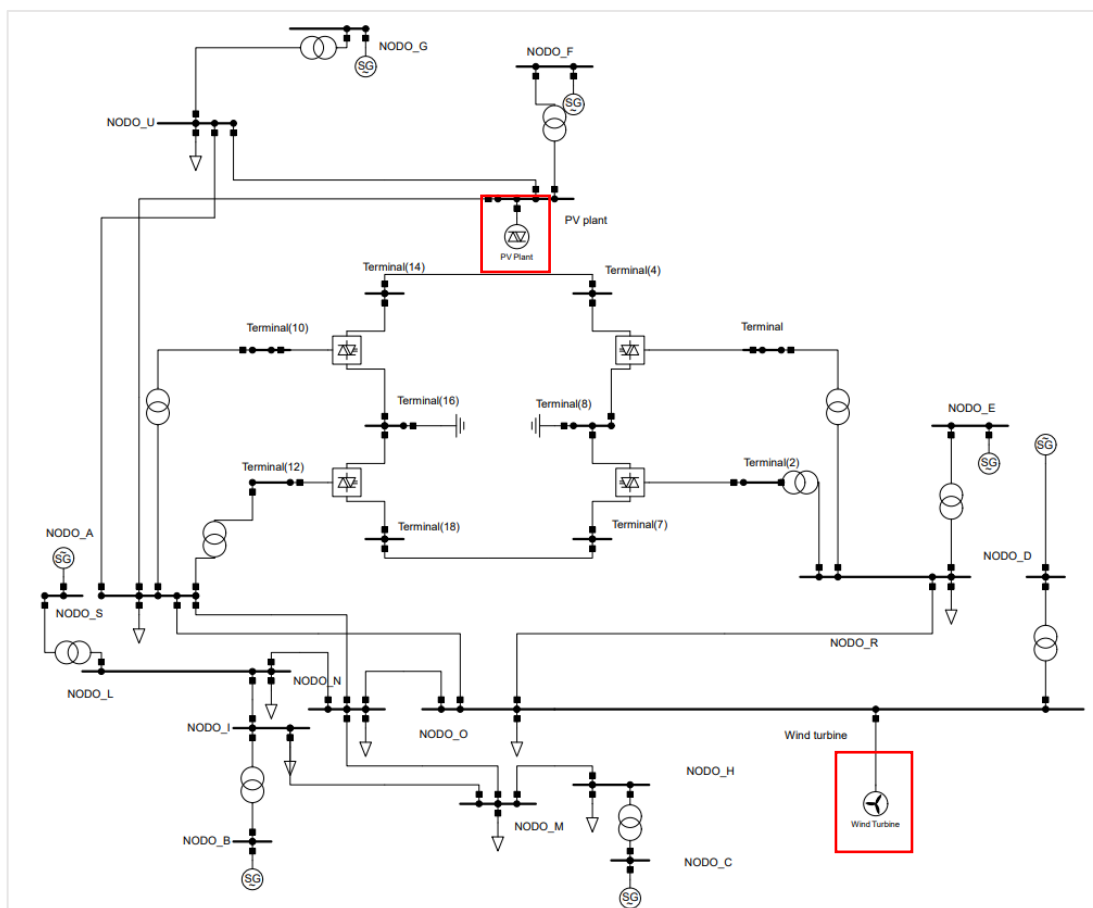


Figura 69 Rete fittizia a 22 nodi AC/DC con presenza di collegamento HVDC-VSC bipolare.

Essa presenta le seguenti caratteristiche generali:

Tabella 10 Principali caratteristiche della rete di riferimento a 18 nodi test AC/DC.

n. di sbarre	22
n. di generatori sincroni da fonte fossile	7
n. di generatori sincroni da fonte rinnovabile	2
n. di carichi elettrici	9
n. di linee elettriche	14
n. di trasf. a 2 avvolgimenti	11
n. di trasf. a 3 avvolgimenti	0
n. di shunt/filtri	0

I due nodi, nominati *gen_dis_1* e *gen_dis_2* riquadrati in rosso sono rispettivamente una turbina eolica e un impianto fotovoltaico e presentano le seguenti caratteristiche riportate in tabella:

Tabella 11 Caratteristiche principali dei generatori distribuiti presenti nella rete.

N. nodo	Type	Pr [MW]	Pdf	c	k	mu	sigma
21	Wind	200	Weibull	0.8466	2	/	/
22	Solar	300	LogNormal	/	/	5.1635	0.9219

5.1.1 Simulazione della rete test AC/DC 22 nodi con collegamento HVDC-VSC bipolare considerando 1000 iterazioni MC

Riportiamo di seguito il confronto tra algoritmi utilizzando un numero di iterazioni del metodo Monte Carlo pari a 1000 e un *Seeding Type* di generazione di numeri casuali impostato su *Automatic*:

Tabella 12 Tabella di confronto tra algoritmi MatLab – DigSilent sulle medie dei risultati con un numero di risultati pari a 1000.

Nome Nodo	Matlab (A)	DigSilent (B)	$E_{rel\%,\sigma}$
NODO_A	1,03	1,03	0 %
NODO_B	1,03	1,03	0 %
NODO_C	1,03	1,03	0 %
NODO_D	1,03	1,03	0 %
NODO_E	1,03	1,03	0 %
NODO_F	1,03	1,03	0 %
NODO_G	1,03	1,03	0 %
Terminal	1	1	0 %
Terminal (10)	1	1	0 %
Terminal (2)	1	1	0 %
Terminal (12)	1	1	0 %
NODO_H	1,011464	1,011465	0,0003%
NODO_I	1,000356	1,000362	0,0002%
NODO_L	1,011612	1,011625	0,0007%
NODO_M	1,002061	1,002063	0,0005%
NODO_N	1,002872	1,002876	0,0010%
NODO_O	0,903773	0,903822	0,0024%
NODO_R	0,999031	0,999037	0,0002%
NODO_S	0,983823	0,983854	0,0014%
NODO_U	1,002196	1,00221	0,0005%
Gen_dis_1	1,029625	1,029623	0,0002%
Gen_dis_2	1,014551	1,014565	0,0004%

Il massimo errore relativo ottenuto durante la valutazione delle medie è del 0,0024%, il che rende i due risultati praticamente identici. Per quanto riguarda il confronto delle deviazioni standard, la situazione è altrettanto positiva con l'unica precisazione che si starà confrontando la dispersione dei dati ottenuti come risultati di diverse simulazioni, che partono quindi da *dataset* di dati differenti. Per questo motivo, un margine di errore ammissibile più elevato può essere accettabile. Nella rete presa in esame, l'errore relativo massimo ottenuto è del 1,0763%. Per i nodi di generazione PQ che si estendono da A a G e che considerano tutti i Terminal, l'errore sulla σ sarà praticamente zero. Infatti, sebbene questi valori siano leggermente differenti, sono di ordini di grandezza così piccoli da poter essere approssimati a zero. Ciò è dovuto al fatto che la tensione in un nodo PV è un vincolo e per tale ragione sarà sempre mantenuta pari al valore ad essa assegnato a differenza dei nodi PQ.

Tabella 13 Tabella di confronto tra algoritmi MatLab – DigSilent sulle deviazioni standard dei risultati con un numero di iterazioni pari a 1000.

Nome Nodo	Matlab (A)	DigSilent (B)	$E_{rel\%,\sigma}$
NODO_A	0	0	0 %
NODO_B	0	0	0 %
NODO_C	0	0	0 %
NODO_D	0	0	0 %
NODO_E	0	0	0 %
NODO_F	0	0	0 %
NODO_G	0	0	0 %
Terminal	0	0	0 %
Terminal (10)	0	0	0 %
Terminal (2)	0	0	0 %
Terminal (12)	0	0	0 %
NODO_H	0,000347	0,000343	0,9387%
NODO_I	0,000407	0,000405	0,3457%
NODO_L	0,000567	0,000568	0,1132%
NODO_M	0,000631	0,000625	0,9391%
NODO_N	0,001156	0,001145	0,9347%
NODO_O	0,002486	0,002509	0,9339%
NODO_R	0,000286	0,000289	0,9408%
NODO_S	0,001801	0,001817	0,8781%
NODO_U	0,000983	0,000992	0,9059%
Gen_dis_1	0,00024	0,000243	1,0763%
Gen_dis_2	0,000902	0,000909	0,7267%

Il tempo di esecuzione necessario per effettuare le simulazioni utilizzando l'algoritmo *PFPD_PRB* di *MatLab* sarà di 0.9901 [s], mentre nel caso di *DigSilent* sarà di 1.1150 [s].

6. Applicazione del *PFPD_PRB* su reti con presenza di generazione distribuita

Dopo le verifiche di conformità dei risultati dell'algoritmo *PFPD_PRB* attraverso il confronto effettuato nella sezione precedente è possibile procedere alla simulazione su *Reti IEEE* e reti reali di grande estensione.

6.1 Rete 41 nodi *IEEE* modificata

La rete seguente viene descritta dalla Tabella (15) e rappresentata attraverso il suo schema unifilare in figura (61):

Tabella 14 Principali caratteristiche della rete di riferimento a 41 nodi.

n. di sbarre	41
n. di generatori sincroni da fonte fossile	10
n. di generatori sincroni da fonte rinnovabile	5
n. di carichi elettrici	26
n. di linee elettriche	36
n. di trasf. a 2 avvolgimenti	12
n. di trasf. a 3 avvolgimenti	0
n. di shunt/filtri	0

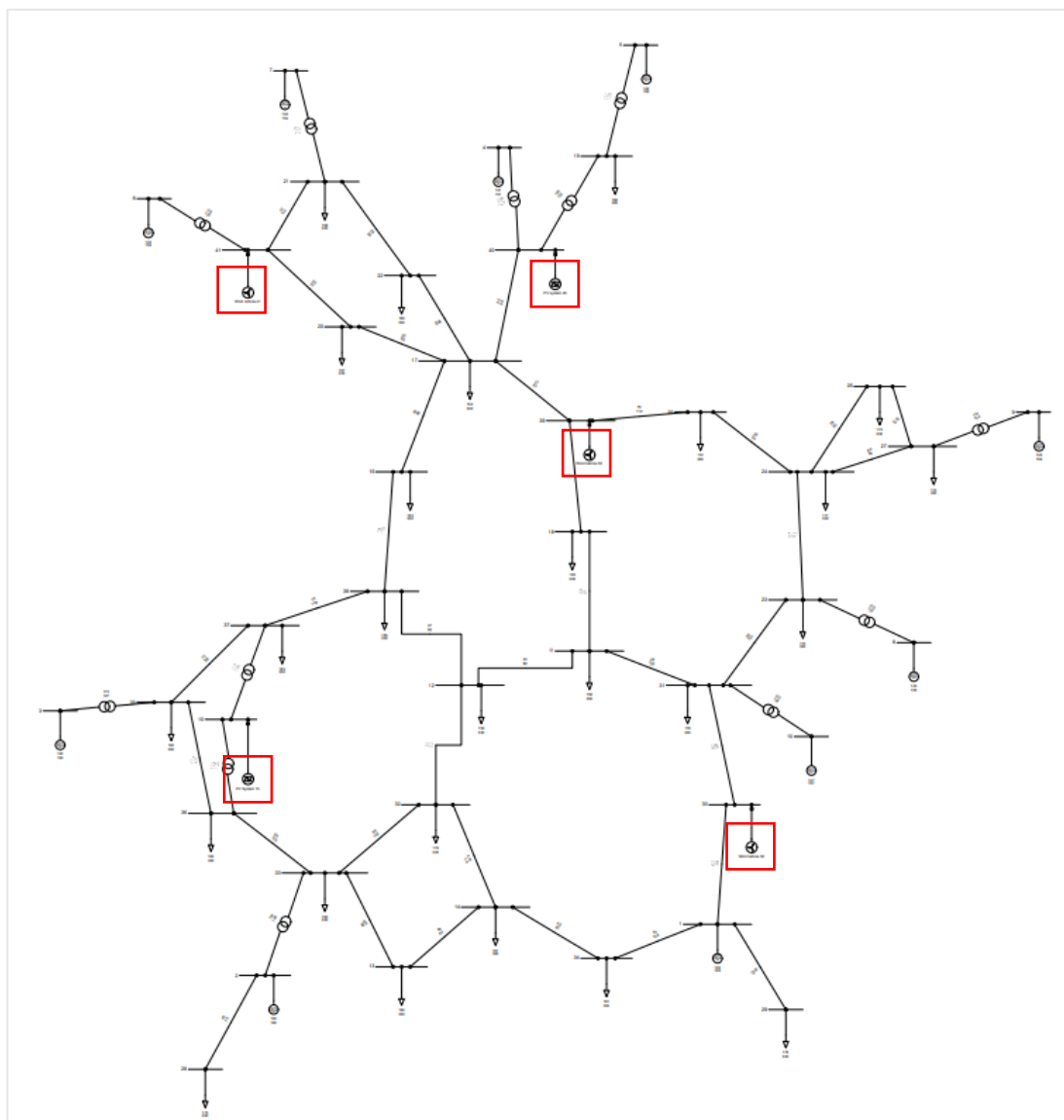


Figura 70 Rete 41 nodi IEEE modificata con la presenza di 5 fonti di generazione da energia rinnovabile.

Nella figura sovrastante vengono evidenziate con riquadri rossi gli *Static Generator* a cui sono associati i generatori distribuiti, per completezza nella tabella seguente vengono riportate le caratteristiche di ogni generatore.

Tabella 15 Caratteristiche di identificazione della funzione di distribuzione associata alla fonte di generazione e della relativa taglia della macchina.

N.nodo	Type	Pr [MW]	Pdf	c	k	mu	sigma
41	Wind	300	Weibull	12.2799	2	/	/
39	Wind	300	Weibull	13.5359	2	/	/
30	Wind	140	Weibull	15.7919	2	/	/
40	Solar	200	LogNormal	/	/	5.1635	0.9219
15	Solar	350	LogNormal	/	/	5.0007	0.9688

Inoltre, per quanto riguarda i tre Aerogeneratori dei nodi 41, 39, 30 vi andremo ad associare anche due *Power Wind Curve*, la prima (sopra) relativa agli impianti di taglia pari a 300 [MW] mentre la seconda (sotto) relativa agli impianti da 140 [MW], le caratteristiche sono mostrate nella figura seguente:

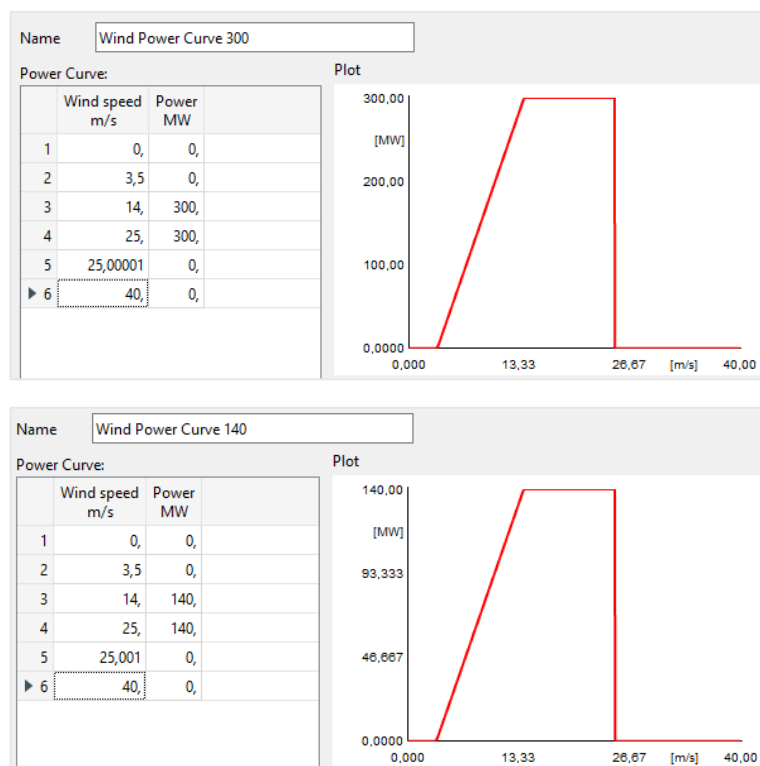


Figura 71 Power Wind Curve degli Aerogeneratori.

Una volta descritte le caratteristiche principali della rete presa in analisi prima di effettuare ulteriori considerazioni confrontiamo i risultati ottenuti tra *DigSilent* e *MatLab* così da validificare la rete attraverso il confronto tra valori attesi e deviazioni standard sulla base dei ragionamenti affrontati nella sezione (4.4). I risultati sono ottimi in quanto gli errori relativi sulle misure sono molto bassi e i due algoritmi possono essere ben paragonati, di seguito riportate tabelle e figure per quanto detto anticipatamente.

Tabella 16 Tabella di confronto tra algoritmi MatLab – DigSilent sulle medie delle medie e sulle medie delle deviazioni dei risultati, con un numero di iterazioni MC pari a 1000.

Nome Nodo	MatLab (A)	DigSilent (B)	$E_{rel\%,\mu}$	MatLab (A)	DigSilent (B)	$E_{rel\%,\sigma}$
1	1,03	1,03	0,0000%	0,000000	0,000000	0,0000%
2	0,982	0,982	0,0000%	0,000000	0,000000	0,0000%
3	0,9831	0,9831	0,0000%	0,000000	0,000000	0,0000%
4	0,9972	0,9972	0,0000%	0,000000	0,000000	0,0000%
5	1,0123	1,0123	0,0000%	0,000000	0,000000	0,0000%
6	1,0493	1,0493	0,0000%	0,000000	0,000000	0,0000%
7	1,0635	1,0635	0,0000%	0,000000	0,000000	0,0000%
8	1,0278	1,0278	0,0000%	0,000000	0,000000	0,0000%
9	1,0265	1,0265	0,0000%	0,000000	0,000000	0,0000%
10	1,0475	1,0475	0,0000%	0,000000	0,000000	0,0000%
11	0,993525	0,993697	0,0172%	0,005139	0,005071	1,3416%
12	0,96173	0,96188	0,0156%	0,005027	0,00497	1,1516%
13	0,944585	0,944836	0,0266%	0,006935	0,006861	1,0653%
14	0,942148	0,94243	0,0299%	0,007692	0,007611	1,0645%
15	0,968679	0,968799	0,0124%	0,004369	0,004481	2,5162%
16	0,980098	0,980138	0,0041%	0,003187	0,003127	1,9450%
17	0,997574	0,997572	0,0002%	0,002609	0,002544	2,5339%
18	0,996825	0,996926	0,0101%	0,004436	0,004366	1,5951%
19	0,981788	0,981789	0,0001%	0,000846	0,000822	2,8800%
20	1,003922	1,003856	0,0066%	0,001736	0,001694	2,4328%
21	1,027185	1,027155	0,0029%	0,000995	0,000973	2,3245%
22	1,006177	1,006141	0,0036%	0,002323	0,002265	2,5709%
23	1,018107	1,018248	0,0138%	0,003927	0,003884	1,1154%
24	1,022883	1,022888	0,0005%	0,003499	0,003456	1,2531%
25	1,007585	1,007606	0,0021%	0,003701	0,003645	1,5437%
26	1,028808	1,028781	0,0027%	0,001843	0,001813	1,6658%
27	1,028565	1,02854	0,0025%	0,001286	0,001264	1,7806%
28	0,981993	0,981993	0,0000%	7,95E-15	2,08E-08	0,0000%
29	1,029999	1,029999	0,0000%	1,52E-14	7,5E-08	0,0000%
30	1,021292	1,021707	0,0406%	0,007103	0,00703	1,0303%
31	1,016589	1,016786	0,0194%	0,004934	0,00488	1,0942%
32	0,955923	0,956122	0,0209%	0,00568	0,005622	1,0438%
33	0,957634	0,957819	0,0193%	0,005284	0,005231	1,0041%
34	0,975142	0,975626	0,0496%	0,009827	0,009742	0,8765%
35	0,970129	0,970247	0,0122%	0,003738	0,003721	0,4323%
36	0,966243	0,966384	0,0146%	0,004288	0,004265	0,5461%
37	0,970599	0,970716	0,0121%	0,003908	0,003896	0,2880%
38	0,971088	0,9712	0,0115%	0,004139	0,004101	0,9287%
39	1,000814	1,000861	0,0048%	0,003856	0,00379	1,7443%
40	0,992	0,992001	0,0002%	0,001148	0,001116	2,8784%

Tabella 17 Tabella di confronto tra algoritmi MatLab – DigSilent sulle medie delle medie e sulle medie delle deviazioni dei risultati, con un numero di iterazioni MC pari a 10000.

Nome Nodo	MatLab (A)	DigSilent (B)	$E_{rel\%,\mu}$	MatLab (A)	DigSilent (B)	$E_{rel\%,\sigma}$
1	1,03	1,03	0,0000%	0,000000	0,000000	0,0000%
2	0,982	0,982	0,0000%	0,000000	0,000000	0,0000%
3	0,9831	0,9831	0,0000%	0,000000	0,000000	0,0000%
4	0,9972	0,9972	0,0000%	0,000000	0,000000	0,0000%
5	1,0123	1,0123	0,0000%	0,000000	0,000000	0,0000%
6	1,0493	1,0493	0,0000%	0,000000	0,000000	0,0000%
7	1,0635	1,0635	0,0000%	0,000000	0,000000	0,0000%
8	1,0278	1,0278	0,0000%	0,000000	0,000000	0,0000%
9	1,0265	1,0265	0,0000%	0,000000	0,000000	0,0000%
10	1,0475	1,0475	0,0000%	0,000000	0,000000	0,0000%
11	0,993507	0,99368	0,0174%	0,005111	0,005103	0,1477%
12	0,961708	0,961852	0,0149%	0,005011	0,005021	0,2016%
13	0,944563	0,944802	0,0252%	0,006906	0,006911	0,0762%
14	0,942126	0,942394	0,0285%	0,007659	0,007662	0,0434%
15	0,968637	0,968741	0,0107%	0,004608	0,004919	6,3338%
16	0,980081	0,980118	0,0038%	0,003166	0,003183	0,5255%
17	0,997561	0,997559	0,0003%	0,002578	0,002592	0,5574%
18	0,996806	0,996911	0,0105%	0,004407	0,004400	0,1624%
19	0,981787	0,981784	0,0003%	0,000833	0,000849	1,8560%
20	1,003915	1,003845	0,0069%	0,001716	0,001726	0,5651%
21	1,027182	1,027149	0,0031%	0,000985	0,000990	0,5410%
22	1,006166	1,006129	0,0037%	0,002295	0,002308	0,5532%
23	1,018096	1,018239	0,0140%	0,003909	0,003906	0,0913%
24	1,022869	1,022878	0,0009%	0,00348	0,003477	0,0824%
25	1,007569	1,007594	0,0025%	0,003677	0,003672	0,1289%
26	1,028801	1,028776	0,0024%	0,001833	0,001824	0,4903%
27	1,02856	1,028536	0,0023%	0,001279	0,001272	0,6039%
28	0,981993	0,981993	0,0000%	0,000000	0,000000	0,0000%
29	1,029999	1,029999	0,0000%	0,000000	0,000000	0,0000%
30	1,021283	1,021671	0,0380%	0,007084	0,007076	0,1197%
31	1,016576	1,016775	0,0196%	0,004912	0,004908	0,0740%
32	0,955901	0,95609	0,0197%	0,005662	0,005673	0,1904%
33	0,957613	0,957787	0,0181%	0,00527	0,005285	0,2823%
34	0,975128	0,975598	0,0481%	0,009792	0,009813	0,2125%
35	0,970106	0,970216	0,0113%	0,003763	0,003821	1,5139%
36	0,966219	0,96635	0,0135%	0,004308	0,004364	1,2828%
37	0,970573	0,970682	0,0112%	0,003944	0,004016	1,7910%
38	0,971064	0,971169	0,0108%	0,004145	0,004180	0,8399%
39	1,000794	1,000847	0,0053%	0,003828	0,003826	0,0558%
40	0,991998	0,991994	0,0004%	0,001131	0,001152	1,8494%
41	1,02721	1,027188	0,0022%	0,000953	0,000957	0,4632%

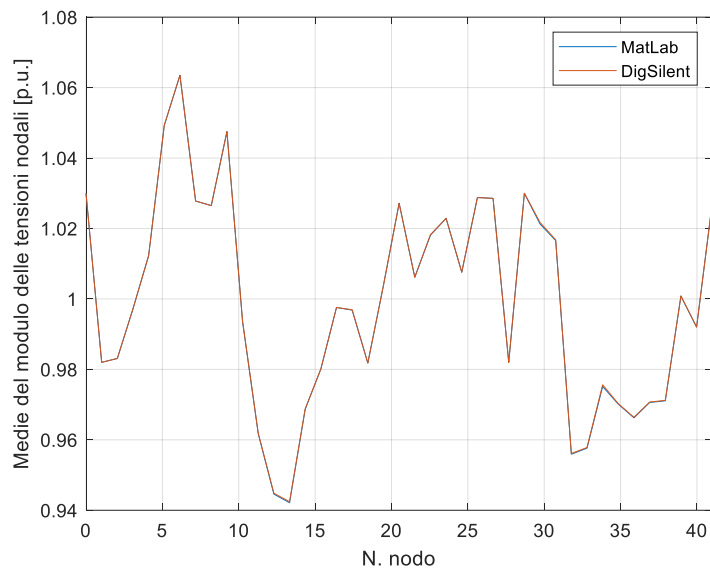


Figura 72 Confronto medie delle tensioni nodali in [p.u.] tra MatLab e DigSilent.

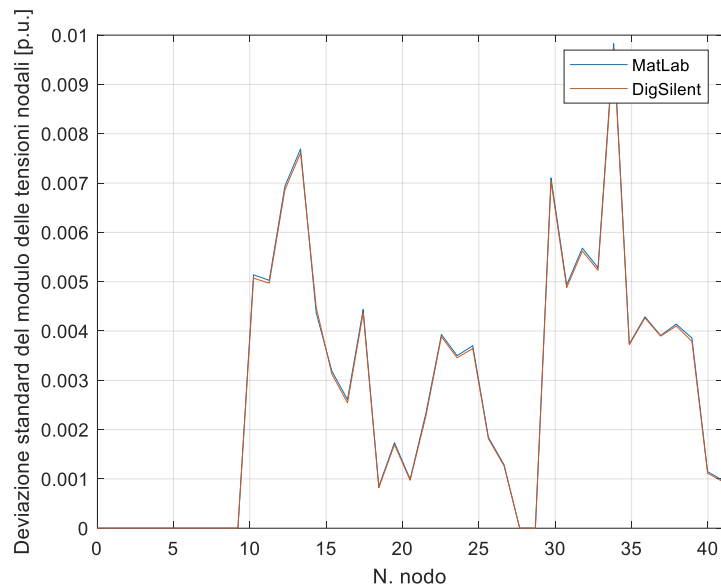


Figura 73 Confronto deviazioni standard delle tensioni nodali in [p.u.] tra MatLab e DigSilent.

Il tempo di esecuzione necessario per effettuare le simulazioni utilizzando l'algoritmo *PFPD_PRB* di *MatLab* sarà di $1.2598 [s]$, mentre nel caso di *DigSilent* sarà di $1.3520 [s]$ per 1000 iterazioni mentre nel caso di 10000 iterazioni per effettuare le simulazioni utilizzando l'algoritmo *PFPD_PRB* di *MatLab* impiegheremo $11.794 [s]$, mentre nel caso di *DigSilent* sarà di $20.1678 [s]$. Dai grafici evidenziati sopra, è possibile osservare che le differenze tra le medie e le deviazioni standard delle tensioni non sono distinguibili, indicando che i due algoritmi sono completamente comparabili. È interessante analizzare le tensioni dei nodi PQ e come queste variano in condizioni di ventosità ed irraggiamento differenti. Per tale analisi, si considerano tre diversi scenari in cui si evidenziano i parametri fondamentali (c , k , μ_{log} , σ_{log}). Per effettuare il confronto, si analizzano i nodi 11, 17, 24 e 37 e si mostra la

distribuzione normale della tensione nodale confrontandola con le bande ammissibili di sovra e sotto tensione accettabili lungo una linea di trasmissione ($u \pm 10\%$). In questo modo, si dimostra come i fenomeni climatici imprevedibili possono causare situazioni critiche per i nodi della rete di trasmissione. Lo studio considera tre scenari differenti, presentati nella tabella seguente, che raccoglie i parametri delle distribuzioni ottenuti a partire da dati climatici.

Tabella 18 Caratteristiche di identificazione della funzione di distribuzione associata alla fonte di generazione per ogni scenario evidenziato.

Scenario 1	c	k	mu	sigma
41	12.2799	2	/	/
39	13.5359	2	/	/
30	15.7919	2	/	/
40	/	/	5.1635	0.9219
15	/	/	5.0007	0.9688
Scenario 2				
41	15.7920	2	/	/
39	18.0480	2	/	/
30	20.3040	2	/	/
40	/	/	6.1000	1.1200
15	/	/	6.0000	1.2200
Scenario 3				
41	18.0480	2	/	/
39	20.3040	2	/	/
30	22.5600	2	/	/
40	/	/	6.6	1.25
15	/	/	6.5	1.32

Le *Probability Density Function* relative alle tensioni dei nodi 17, 19, 36 e 37 mostrano come all'aumentare della variabilità delle condizioni climatiche le tensioni presentano una *PDF* con picchi più bassi e curve più espanse questo è dovuto all'incremento della deviazione standard delle tensioni nei nodi, fenomeno che ci aspettavamo in quanto la generazione risulta più variabile, la deviazione standard sulle tensioni risulta quindi essere proporzionale alla deviazioni sulla generazione sulla potenza generata dai nodi. Per i nodi 17, 19, 36 e 37 la variazione di tensione non supera le bande limite che portano alla creazione di situazioni patologiche, la probabilità che non vi sia una sovra/sotto tensione è del 100% dunque. Per calcolare dunque la probabilità che una sovra o sotto tensione accada bisognerà applicare le seguenti formulazioni:

$$P(X \leq x) = \int_{-\infty}^{0.85} f(x) dx$$

$$P(X \geq x) = \int_{1.05}^{\infty} f(x) dx$$

Di seguito vengono riportati i grafici con gli andamenti delle tensioni e le relative bande di sovra e sotto tensione presenti all'interno dei nodi della rete elettrica di trasmissione

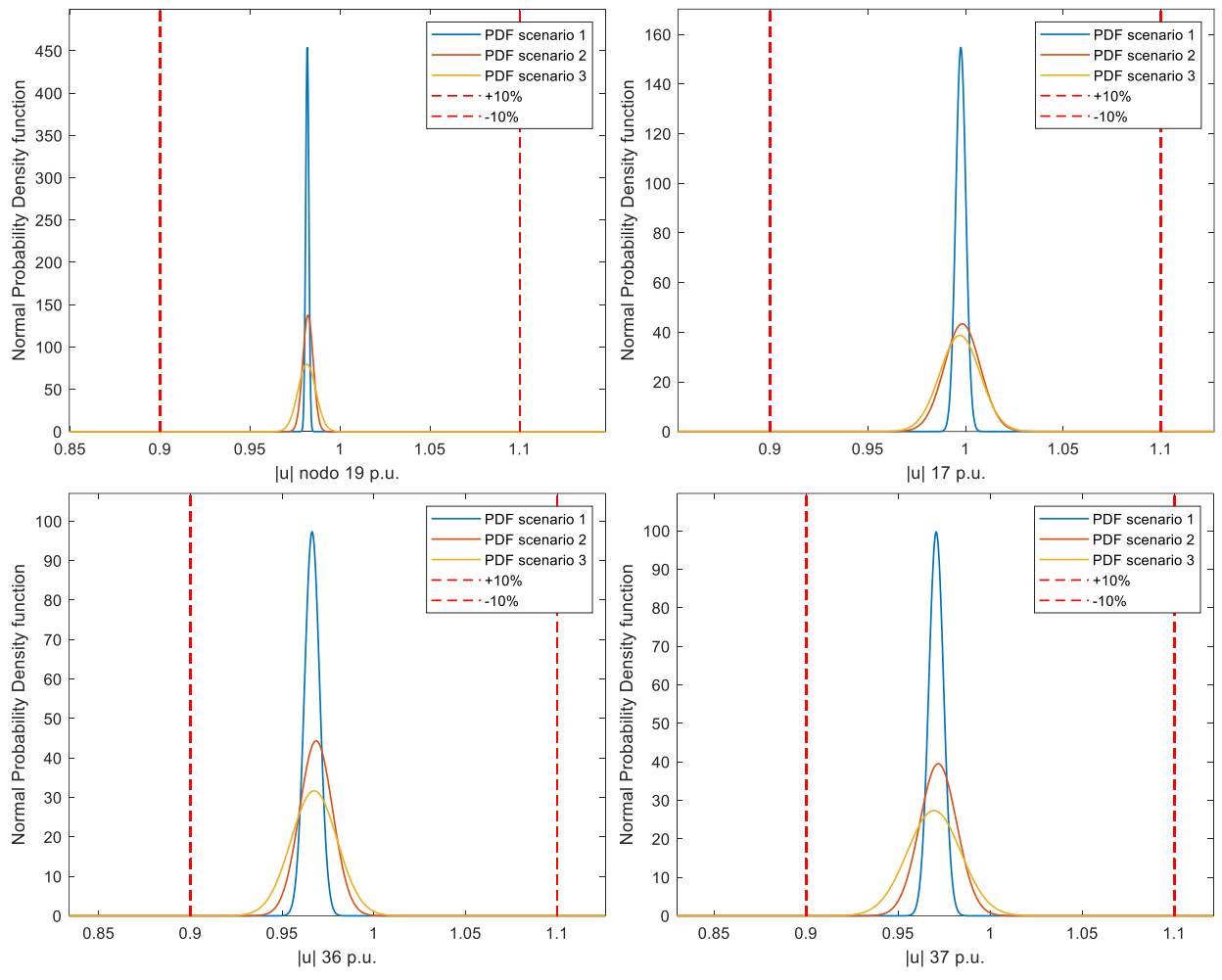


Figura 74 Confronto dell'andamento delle tensioni nodali espresse in [p.u.] per i nodi 17, 19, 36 e 37 rispetto ai tre diversi scenari.

6.1.1 Studio dell'Ampacity delle linee elettriche

Lo studio della portata delle linee elettriche di trasmissione è essenziale per garantire la sicurezza e l'affidabilità del sistema elettrico, prevenendo guasti ed eventuali danni ai componenti dell'infrastruttura elettrica. La portata al limite termico (*Ampacity*) indica il valore di corrente massimo che la linea può sopportare per un tempo illimitato, con una probabilità minima di scariche e un invecchiamento accettabile degli elementi della linea. Nel nostro caso, abbiamo analizzato tre linee aeree trinate in alluminio acciaio (*All-Acc/ACSR*), caratterizzate da leggerezza e resistenza meccanica, con un nucleo centrale in acciaio e una corona esterna in alluminio che trasmette l'energia elettrica. Considerando una frequenza di rete di esercizio di 50 Hz, una temperatura di esercizio di 50 °C e una insolazione di 1000 [W/m²], al fine di adottare precauzioni estreme, abbiamo riportato i seguenti parametri di esercizio nella tabella:

Tabella 19 Parametri di esercizio e portate di corrente delle linee di trasmissione elettrica utilizzate per il modello a 41 nodi, (valore espressi per subconduttore).

Diametro esterno [mm]	$r_{50^{\circ}\text{C}}$ [Ω/km]	l [mH/km]	c [$\mu\text{F}/\text{km}$]	I_a [kA]
31.5	0.0210	0.8592	0.013	0.62
34.6	0.0229	0.8580	0.013	0.68
37.7	0.0195	0.8591	0.012	0.74

La verifica classica per lo studio della portata di una linea consiste nel confronto tra la corrente massima circolante all'interno e la portata stessa della linea, riportata nell'ultima colonna della tabella (I_a), nel seguente modo:

$$I_a \geq I_M$$

6. 1

In altre parole, la portata deve essere maggiore della corrente massima che circola lungo la linea. Nel caso di DigSilent, e quindi dei ragionamenti che andremo a fare su MatLab, per studiare l'Ampacity calcoleremo due parametri definiti come $loading_{\%(i)}$ e $loading_{\%(j)}$, uno per ogni estremità della linea.

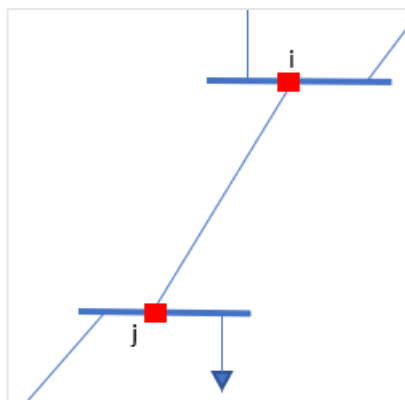


Figura 75 Modellizzazione di una linea elettrica connessa a due sbarre con estremi i e j.

Il massimo tra i due sarà definito come il loading della linea stessa. Questo perché conosciamo la corrente alle estremità della linea elettrica, in entrata al nodo (i) ed in uscita al nodo (j).

Per porci nel caso più cautelativo, prenderemo il massimo tra i due valori. Le formule da utilizzare sono le seguenti:

$$loading_{\%(i)} = 100 \times \frac{I_{bus(i)}}{I_a} \quad 6.2$$

$$loading_{\%(j)} = 100 \times \frac{I_{bus(j)}}{I_a} \quad 6.3$$

$$loading_{\%} = \max\{loading_{\%(i)}, loading_{\%(j)}\} \quad 6.4$$

Quindi, quello che otteniamo non è un confronto diretto con il valore di *Ampacity*, ma piuttosto quanto la linea è caricata, rispetto al valore massimo accettabile corrispondente all'*Ampacity*. Nel nostro caso di studio probabilistico, non studieremo un solo valore di loading, ma la media, la deviazione standard e il valore massimo raggiunto. Questi tre modelli matematici verranno confrontati per tutti e tre gli scenari, al fine di valutare come le linee vengono sollecitate all'aumentare della variabilità delle condizioni climatiche. Tra tutte le 36 linee, ne analizzeremo alcune per studiare il loro comportamento:

- *Linea 43*, è isolata, infatti, permette di connettere un carico localizzato in una posizione molto periferica della rete ad un gruppo di generazione non rinnovabile. Il flusso di potenza lungo la linea sarà dunque unidirezionale in quanto il carico che si sta alimentando è isolato ed il flusso di potenza è unidirezionale. Di seguito riportiamo graficamente il collegamento e i risultati ottenuti a partire dai tre scenari:

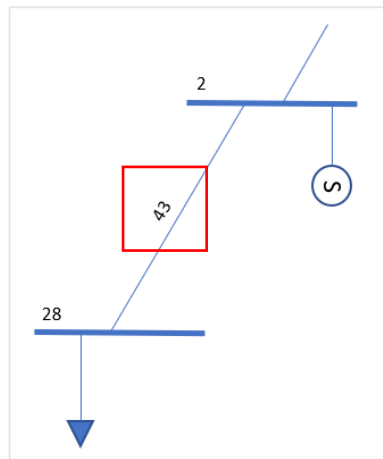


Figura 76 Schema elettrico di collegamento della linea 43.

Tabella 20 Risultati statistici di loading ottenuti per i tre scenari per la linea numero 43.

N. Linea 43	$\mu_{Loading\%}$	$\sigma_{Loading\%}$	$max_{Loading\%}$
Scenario 1	73,30267	5,38E-06	73,3027
Scenario 2	73,30267	4,36E-06	73,30271
Scenario 3	73,30267	6,39E-06	73,3027

A causa delle ragioni topografiche precedentemente rappresentate ed esposte, si può notare che il valore del carico sulla linea è pressoché costante, poiché la deviazione è praticamente nulla.

- *Linea 69*, essa si trova in stretta prossimità di una fonte di generazione fotovoltaica, quest'ultima ha una produzione descritta da una distribuzione *Log-Normale*, l'aleatorietà della fonte si incrementa man mano che passiamo dallo *scenario 1* allo *scenario 3*, il sovraccarico medio della linea per l'appunto passerà da un valore di circa 18.5 fino a raggiungere un valore pari a 35.7, parallelamente ad esso seguirà anche un incremento della deviazione standard e del valore massimo che il parametro *loading* potrà assumere.

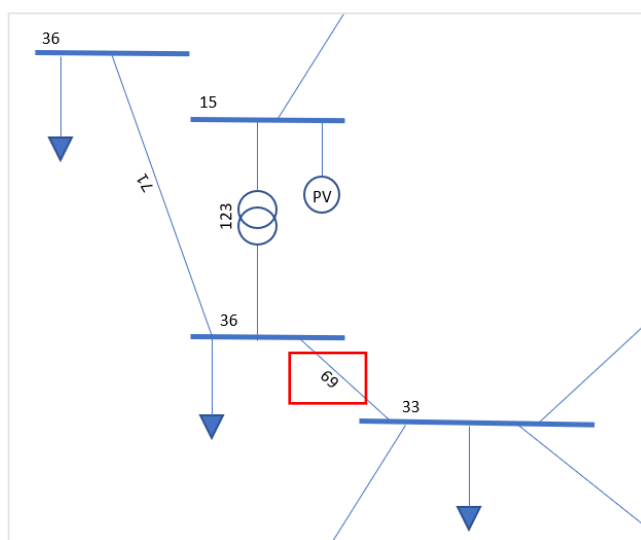


Figura 77 Schema elettrico di collegamento della linea 69.

l'aumento dell'aleatorietà nelle condizioni operative della linea, in particolar modo negli ultimi due scenari, ci porta a un maggiore rischio di superamento dell'ampacity, il che potrebbe causare problemi di sicurezza e affidabilità nella rete elettrica. In questi casi, possono essere necessari interventi sulle protezioni o sui dispositivi della rete, come l'installazione di dispositivi di limitazione della corrente o l'aumento della sezione trasversale della linea, per prevenire o prontamente risolvere eventuali problemi. Tuttavia, tali interventi potrebbero comportare costi elevati, pertanto è necessario valutare attentamente i trade-off tra la sicurezza e la affidabilità della rete e i costi degli interventi.

Tabella 21 Risultati statistici di loading ottenuti per i tre scenari per la linea numero 69.

N. Linea 69	$\mu_{Loading\%}$	$\sigma_{Loading\%}$	$max_{Loading\%}$
Scenario 1	18.45206	4.597382	63.1716
Scenario 2	27.53323	15.22462	112.1577
Scenario 3	35.64683	20.56366	130.8553

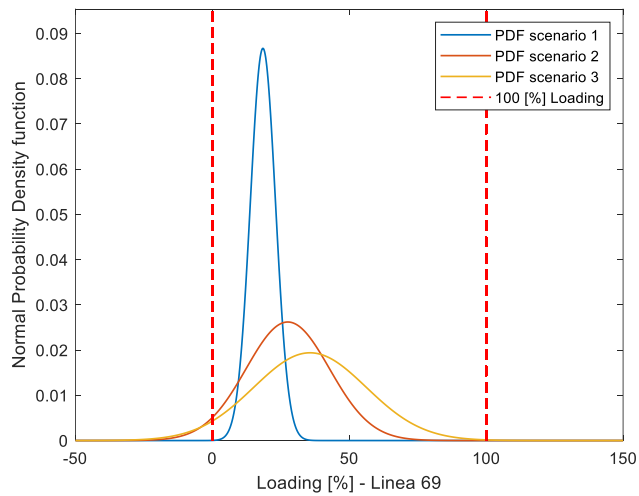


Figura 78 Rappresentazione grafica delle Normal PDF riguardo il Loading [%] della linea elettrica 69.

Risulterà opportuno a tale punto descrivere la probabilità che il Loading [%] massimo ammissibile (100%) venga superato o che la linea non sia caricata (Loading [%] = 0); attraverso l'applicazione delle formule (3.19) e (3.21) sarà possibile ottenere i seguenti risultati:

N. Linea 69	Probabilità di avere un valore di Loading inferiore allo 0%	Probabilità di avere un valore di Loading superiore al 100%
Scenario 1	0.00 %	0.00 %
Scenario 2	3.53 %	0.00 %
Scenario 3	4.15 %	0.00 %

- Linea 52, per questa è possibile fare gli stessi ragionamenti effettuati precedentemente per la linea numero 69., di seguito i risultati ottenuti:

Tabella 22 Risultati statistici di loading ottenuti per i tre scenari per la linea numero 52.

N. Linea 52	$\mu_{Loading\%}$	$\sigma_{Loading\%}$	$max_{Loading\%}$
Scenario 1	11.49331	5.724825	84.82293
Scenario 2	20.64773	18.76608	210.3702
Scenario 3	31.52485	34.15658	280.0935

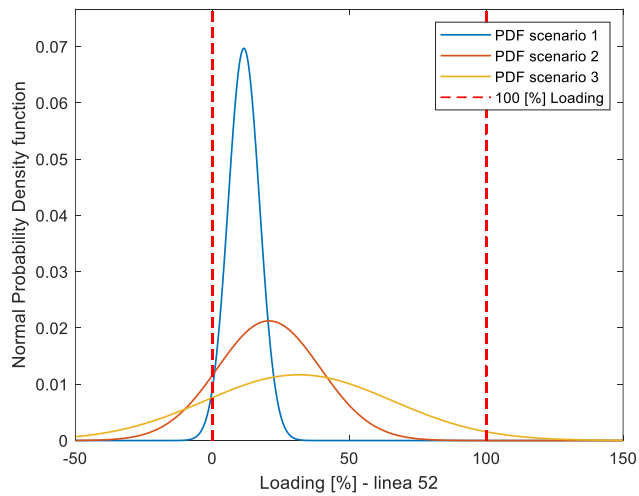


Figura 79 Rappresentazione grafica delle Normal PDF riguardo il Loading [%] della linea elettrica 52.

Risulterà opportuno a tale punto descrivere la probabilità che il *Loading [%]* massimo ammissibile (100%) venga superato o che la linea non sia caricata (*Loading [%] = 0*); attraverso l'applicazione delle formule (3.19) e (3.21) sarà possibile ottenere i seguenti risultati:

N. Linea 52	Probabilità di avere un valore di Loading inferiore allo 0%	Probabilità di avere un valore di Loading superiore al 100%
Scenario 1	2.23 %	0.00 %
Scenario 2	13.56 %	0.00 %
Scenario 3	17.81 %	2.25 %

- *Linea 75*, come per le *linee 52 e 69*, anche in questo caso le variazioni che si avranno sui parametri analizzati saranno le stesse. Tuttavia, l'unica differenza da notare è che tali variazioni saranno meno evidenti per le linee vicine a una fonte di generazione eolica, poiché la *Power Wind Curve* limita l'erogazione di potenza. Questo è diverso dall'impianto fotovoltaico, il quale ha come unica fonte di limitazione un valore massimo di erogazione stabilito dall'inverter di collegamento che non può essere superato. Qui di seguito rappresentiamo la linea di collegamento e i risultati ottenuti a partire dai tre scenari:

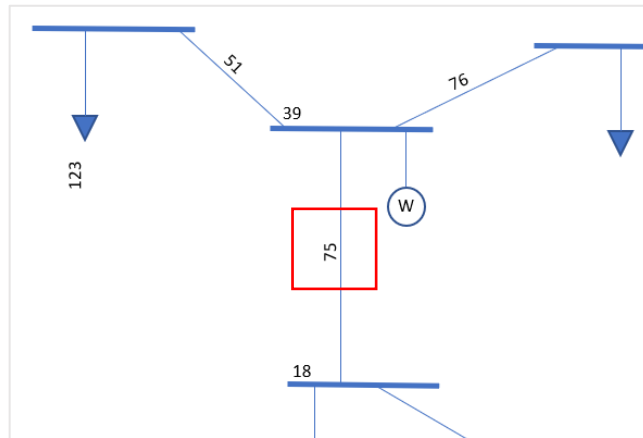


Figura 80 Schema elettrico di collegamento della linea 75.

Tabella 23 Risultati statistici di loading ottenuti per i tre scenari per la linea numero 75.

N. Linea 75	$\mu_{Loading\%}$	$\sigma_{Loading\%}$	$max_{Loading\%}$
Scenario 1	8.89	3.98	22.67
Scenario 2	9.34	6.25	77.89
Scenario 3	12.54	11.59	100.45

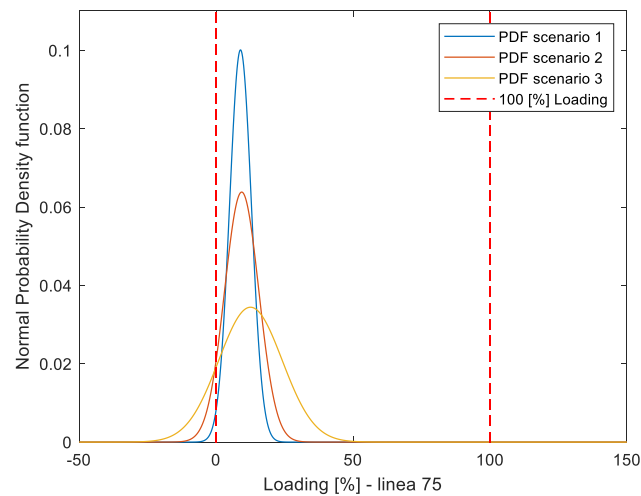


Figura 81 Rappresentazione grafica delle Normal PDF riguardo il Loading [%] della linea elettrica 75.

Risulterà opportuno a tale punto descrivere la probabilità che il Loading [%] massimo ammissibile (100%) venga superato o che la linea non sia caricata (Loading [%] = 0); attraverso l'applicazione delle formule (3.19) e (3.21) sarà possibile ottenere i seguenti risultati:

N. Linea 75	Probabilità di avere un valore di Loading inferiore allo 0%	Probabilità di avere un valore di Loading superiore al 100%
Scenario 1	1.28 %	0.00 %
Scenario 2	6.76 %	0.00 %
Scenario 3	13.95 %	0.00 %

6.1.2 Studio del superamento della corrente nominale dei trasformatori

Il superamento della corrente nominale di un trasformatore si riferisce alla massima corrente elettrica che il trasformatore può trasportare in modo sicuro e continuativo senza surriscaldarsi e senza causare danni al sistema elettrico a cui è collegato. Essa dipende principalmente dalle dimensioni del trasformatore, dalle proprietà del materiale isolante utilizzato nei suoi componenti e dalle condizioni ambientali in cui il trasformatore è installato, in particolar modo, le temperature ambientali elevate possono ridurre la capacità di un trasformatore di trasportare corrente elettrica, in quanto possono compromettere l'isolamento elettrico dei suoi componenti. Di seguito lo schema elettrico rappresentativo:

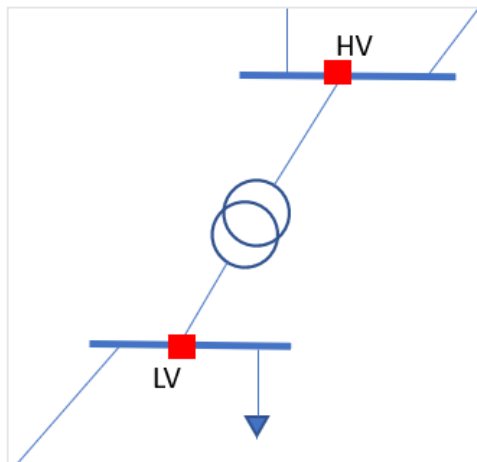


Figura 82 Modellizzazione di un trasformatore a due avvolgimenti connesso a due sbarre con estremi HV ed LV.

Per calcolare la corrente nominale e il *Loading [%]* vengono utilizzate le formule seguenti che sono sviluppate per ogni estremo del trasformatore (lato alta tensione (HV) e lato bassa tensione (LV)).

$$I_{rated}(HV) = \frac{S_{rated}}{\sqrt{3}U_n(HV)} \quad 6.5$$

$$loading_{\%}(HV) = \frac{I(HV)}{I_{rated}(HV)} \quad 6.6$$

$$I_{rated}(LV) = \frac{S_{rated}}{\sqrt{3}U_n(LV)} \quad 6.7$$

$$loading_{\%}(LV) = \frac{I(LV)}{I_{rated}(LV)} \quad 6.8$$

Il valore massimo tra i due verrà identificato come valore di loading del trasformatore in quella condizione ed istante.

$$loading_{\%} = \max\{loading_{\%}(HV), loading_{\%}(LV)\}$$

6.9

Per garantire il funzionamento sicuro e affidabile di un trasformatore, è importante che la corrente nominale del trasformatore sia adeguato alla corrente di carico massima prevista per il sistema elettrico a cui è collegato. Se la corrente nominale del trasformatore è insufficiente per gestire la corrente di carico richiesta, possono verificarsi surriscaldamenti e guasti al sistema elettrico. Nel nostro caso di studio probabilistico, non studieremo un solo valore di Loading, ma la media, la deviazione standard e il valore massimo raggiunto. Questi tre modelli matematici verranno confrontati per tutti e tre gli scenari, al fine di valutare come le linee vengono sollecitate all'aumentare della variabilità delle condizioni climatiche. Tra tutti i 12 trasformatori a 2 avvolgimenti, ne analizzeremo alcuni per studiarne il comportamento:

- *Trasformatore 123-124*: Questi due trasformatori consentono di trasportare l'energia elettrica generata da un campo fotovoltaico all'esterno. Il loro carico è direttamente proporzionale alla produzione e alla variabilità della fonte. Di seguito è riportato lo schema di collegamento:

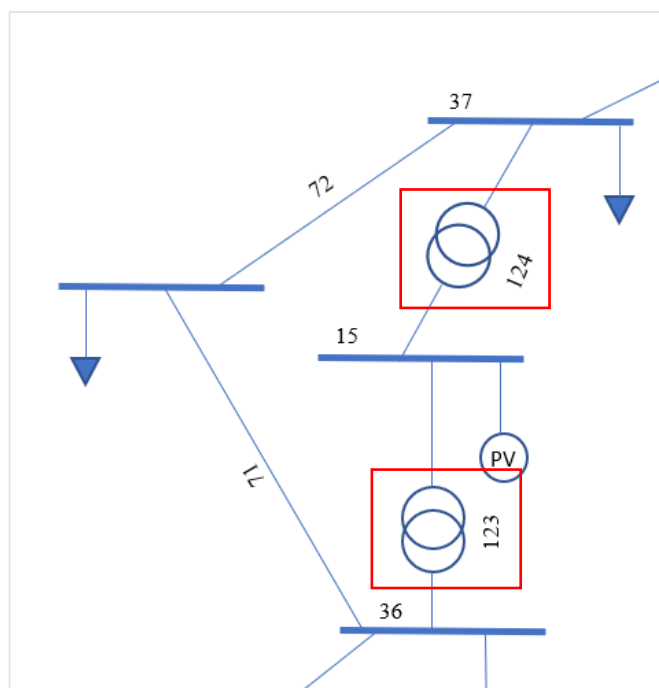


Figura 83 Schema elettrico di collegamento dei trasformatori 123 e 124.

Come si può evincere dai risultati riportati di seguito, il carico medio ha subito un notevole incremento, con una conseguente crescita sia della deviazione standard che del valore massimo raggiunto. Di seguito sono presentati i risultati:

Tabella 24 Risultati statistici di loading ottenuti per i tre scenari per i trasformatori numero 123 e 124.

N. Trasformatore			
Scenario 1	$\mu_{Loading\%}$	$\sigma_{Loading\%}$	$max_{Loading\%}$
123	12,98237	16,34826	170,8982
124	14,97034	16,04201	172,0945
Scenario 2			
123	41,56864	52,00945	366,7899
124	41,45893	50,66478	358,322
Scenario 3			
123	59,40957	64,78888	362,0971
124	57,4505	63,00279	355,4073

Riportiamo quindi graficamente i risultati appena ottenuti:

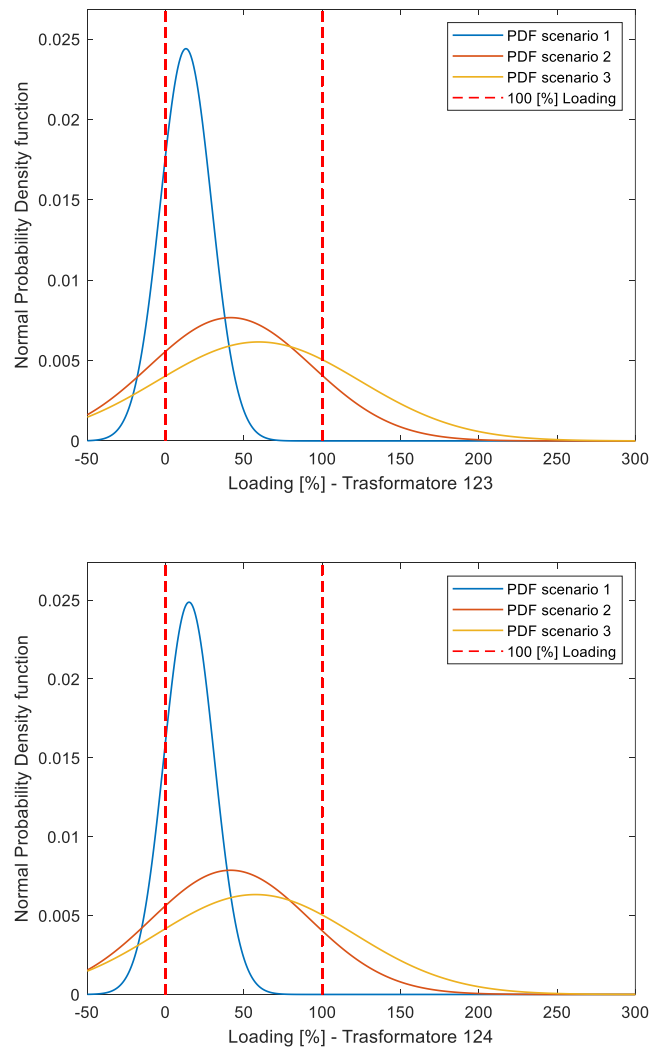


Figura 84 Rappresentazione grafica delle Normal PDF riguardo il Loading [%] del Trasformatore 123 e 124.

Le curve mostrano un aumento del valore medio di carico dovuto all'incremento della produzione della fonte rinnovabile, nonché un appiattimento della curva a causa dell'aumento della variabilità della stessa. Inoltre, mediante l'utilizzo della teoria degli integrali e l'applicazione dell'equazione (3.12) al nostro caso di studio è possibile calcolare la probabilità di eccedere il 100% (*funzionamento in sovraccarico*):

$$P(x)_{\%} = \int_{100}^{\infty} pdf(x) dx$$

6.10

Trasformatore - 123	Probabilità di avere un valore di Loading superiore al 100%	Trasformatore - 124	Probabilità di avere un valore di Loading superiore al 100%
Scenario 1	0.00 %	Scenario 1	0.00 %
Scenario 2	13.06 %	Scenario 2	12.39 %
Scenario 3	26.55 %	Scenario 3	24.97 %

- *Trasformatore 122*, non è direttamente collegato ad una fonte di energia rinnovabile, topologicamente è collocato nel modo seguente.

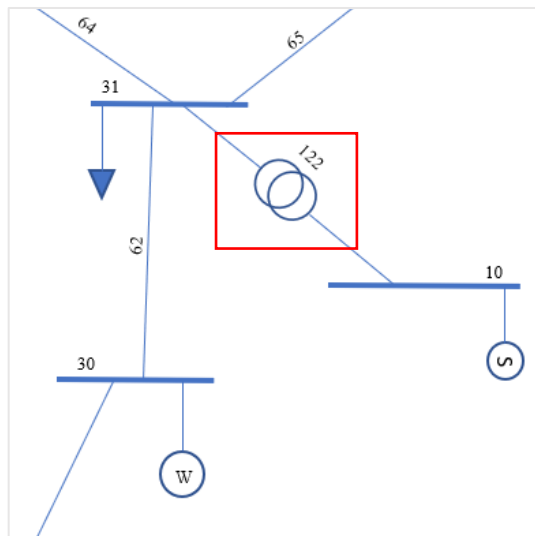


Figura 85, schema elettrico del collegamento del trasformatore 122.

Tuttavia, quest'ultimo (si riferisce al trasformatore menzionato precedentemente) subisce, seppur in maniera più moderata, le variazioni dovute alle condizioni di produzione del nostro sistema. Di seguito, sono riportati i risultati in tabella:

Tabella 25, risultati statistici di loading ottenuti per i tre scenari per il trasformatore 122.

N. Trasformatore 122	$\mu_{Loading\%}$	$\sigma_{Loading\%}$	$max_{Loading\%}$
Scenario 1	29.75	1.57	34.79
Scenario 2	28.39	1.68	34.78
Scenario 3	28.06	2.06	52.82

6.2 Rete Siciliana con presenza di generazione distribuita

La rete seguente viene descritta dalla *Tabella (26)* e rappresentata attraverso il suo schema di collegamento in *Figura (86)*:

Tabella 26 Principali caratteristiche della rete di riferimento Siciliana.

n. di sbarre	467
n. di generatori sincroni da fonte fossile	34
n. di generatori sincroni da fonte rinnovabile	58
n. di carichi elettrici	355
n. di linee elettriche	452
n. di trasf. a 2 avvolgimenti	148
n. di trasf. a 3 avvolgimenti	13
n. di shunt/filtri	9



Figura 86 Rete Siciliana di trasmissione.

Lo studio di una rete di così vaste dimensioni ha permesso la creazione del processo di automatizzazione dell'algoritmo *PPFD_PRB* permettendo una importazione generale per n. fonti di generazione rinnovabile, una modifica dell'interfaccia di importazione tra *Excel* e *MatLab* e lo sfruttamento di due nuove function nominate *power_curve_wind.m* e *power_solar.m*. Gli sviluppi futuri dell'algoritmo sono dunque focalizzati sulla sua applicazione in reti di interconnessione sempre più grandi ad alta e altissima tensione. Quest'ultimo non ci ha permesso fino ad ora di ottenere i medesimi risultati confrontabili con DigSilent e per tale ragione verrà ripreso a valle di questo elaborato in modo tale da poter dare una prova su una rete reale delle sue potenzialità. Di seguito riportiamo il codice attuale per lo studio di tale rete:

```
load('C:\Users\Nicco\OneDrive\Desktop\tesi magistrale\ALGORITMO\Sicilia\DATI\Dat1.
mat');
% global omegan Nbase N_nodi_gen N_nodi_tot
% Settaggio numero di iterazioni Monte Carlo (MC).
set=1000;
%% Dati in input per far funzionare il codice
% Le uniche parti di codice da modificare sono quelle riguardanti il numero
% di fonti di generazione da energie rinnovabili presenti nella rete "N_gen_w"
% e N_gen_s, ed i dati di input di ventosità (Data_wind_gen) e di
% irraggiamento (Ig_hourly).
%% Calcolo valore eolico
% Potenze di taglia dei campi eolici espresse in [MW]
Data_wind_gen=[10];
[r_w,c_parameter] = r_w(Data_wind_gen,set);
[P_r_w,prova_w] = power_curve_wind(set,r_w,PLANT);
%% Calcolo valore fotovoltaico
% Potenze di taglia dei campi fotovoltaici espresse in [MW].
Ig_hourly=[0 0 0 0 0 26.24 152.81 324.41 498.29 632.62 720.37 783.24 746.49 668.85
526.47 370.39 200.39 48.39 0.02 0 0 0 0];
[r_s,mu,sigma] = r_s(Ig_hourly,set);
% Aggiungo la colonna a PLANT contenente le potenze attive di tutti gli
% impianti fotovoltaici, poichè nel caso di fotovoltaico multiplico la
% attiva.
[P_r_s,prova_s]=power_solar(set,r_s,PLANT,CAR_GEN);
%% Dichiarazione valori dei generatori distribuiti
% Numeri casuali generati a partire dalle distribuzioni sopracalcolate.
%% Calcolo della matrice multidim CAR_GEN_mdin
% Tale matrice è costituita da un numero di colonne pari a set
% attraverso l'operazione di vettorializzazione seguente potremmo avere
% l'associazione.
CAR_GEN_mdin(:, :, 1:set) = repmat(CAR_GEN, 1, 1, set);
parfor i=1:size(PLANT,1)
    if PLANT(i,2)==2
        CAR_GEN_mdin(i, 3, :) = reshape(P_r_s(:,i), 1, 1, set);
    elseif PLANT(i,2)==1
        CAR_GEN_mdin(i, 3, :) = reshape(P_r_w(:,i), 1, 1, set);
    end
end
%% Creazione delle matrici utilizzate nel ciclo PFPD classico.
% Le matrici create in questa sezione di codice verranno salvate
% all'interno di matrici multidimensionali indicate con Nomefile_mdin e
% ripescate volta per volta in ogni ciclo successivo.
start=tic;
if size(LINEE_VSC) ~= [0 0]
% Calcolo delle perdite dovute alle Linee VSC (Vengono prese in
% considerazione solo in questo modo).
[Ploss_1,Ploss_2] = A07_PERDITE_VSC(VSC,CONV_VSC,LINEE_VSC,RETE_VSC);
else
    Ploss_1=0;
    Ploss_2=0;
end
end
```

```

%% Stima della Yrete_base senza perdite
[Yretebase] = A_Yretebase_nolosses (RETE, TRASF, TRASF_TRE, SHUNT, omegan, Nbase, N_nodi_tot);
%% Stima della potenza reattiva da assegnare ai nodi di generazione
V_assegnate_pu_mdim = CAR_GEN_mdim(1:N_nodi_gen,5,:)./CAR_GEN_mdim(1:N_nodi_gen,6,:); %Tensioni in p.u. dei PV.
V_assegnate_tot_pu_mdim = [V_assegnate_pu_mdim;ones(N_nodi_tot - N_nodi_gen,1, set)]; %Tensioni in p.u. di tutti i nodi.
% Stima iniziale matrice Ysh
S_compl_pu_mdim = zeros(N_nodi_tot,1,set); %Vettore di zeri pari al Numero totale di nodi
S_compl_pu_mdim(CAR_GEN_mdim(:,2),1,:) = conj((CAR_GEN_mdim(:,3,:)+li*CAR_GEN_mdim(:,4,:))/Nbase);
S_compl_pu_mdim(1:N_nodi_gen,1,:) = zeros(N_nodi_gen,1,set);
Y_car_gen_mdim=zeros(N_nodi_tot);
parfor i=1:set
Y_car_gen_mdim(:,i) = diag(S_compl_pu_mdim(:,i)./V_assegnate_tot_pu_mdim(:,i).^2);
end
Y_car_gen_mdim = (Y_car_gen_mdim);
Segni = diag([-ones(N_nodi_gen,1);ones((N_nodi_tot-N_nodi_gen),1)]);
Segni(:,1:set)=repmat(Segni,1,1,set);
%RICORDA tale matrice non contiene le perdite e va sommata a quella di rete
%calcolata precedentemente.
Y_car_gen_mdim = bsxfun(@times, Y_car_gen_mdim, Segni);
%Partizione della matrice Y_rete_base in quattro quadranti:
Ygg = Yretebase(1:N_nodi_gen,1:N_nodi_gen);
Ygl = Yretebase(1:N_nodi_gen,(N_nodi_gen+1):N_nodi_tot);
Ylg = Yretebase((N_nodi_gen+1):N_nodi_tot,1:N_nodi_gen);
Yll = Yretebase((N_nodi_gen+1):N_nodi_tot,(N_nodi_gen+1):N_nodi_tot);
Yll_mdim(:,1:set)=repmat(Yll,1,1,set);
%ESTRAPOLAZIONE DI Y_car_gen dei soli carichi, essendo che ciò che mi serve
%è solo quella contenente le ammettenze di carico siccome il resto nella
%matrice è 0
YLL_mdim = Y_car_gen_mdim((N_nodi_gen+1):N_nodi_tot,(N_nodi_gen+1):N_nodi_tot,:);
%Somma delle due matrici sopra partizionate:
Ytotll_mdim = Yll_mdim + YLL_mdim; %è quella che in letteratura la chiamo come Y_LL
%Calcolo correnti erogate generatori
ig_mdim=zeros(N_nodi_gen,1,set);
REA_val_mdim=zeros(N_nodi_gen,1,set);
parfor i=1:set
ig_mdim(:,i) = (Ygg - Ygl*(Ytotll_mdim(:,i)\Ylg))*V_assegnate_pu_mdim(:,i);
%V_assegnate_pu è vettore di numeri reali(tutti su asse reale) in pu
%Potenze reattive generatori
REA_val_mdim(:,i) = imag(V_assegnate_pu_mdim(:,i).*conj(ig_mdim(:,i))) * Nbase; %Pot. reattive in valori assoluti
end
% --- end ---
% Sostituzione delle potenze reattive appena stimate
CAR_GEN_mdim(2:N_nodi_gen,4,:) = REA_val_mdim(2:N_nodi_gen,1,:);
%% Calcolo della Y_retebase considerando le perdite

```

```

[Yretebase_mdim,Yretebase]=A_Yretebase(RETE,TRASF,TRASF_TRE,SHUNT,omegan,Nbase,
N_nodi_tot,set);
% --- end ---
%% [Y_car_gen,Ylg,YLL,Yll_1,YglYll_1,Yequi,Zequi] = A03_Y_car_gen(CAR_GEN,
Yretebase,Ea);
V_assegnate_tot_pu_mdim = [V_assegnate_pu_mdim;ones(N_nodi_tot-N_nodi_gen,1,
set)];
S_compl_pu_mdim = zeros(N_nodi_tot,1,set); % Crea un vettore con numero di
righe pari al numero di nodi totali del sistema
S_compl_pu_mdim(CAR_GEN_mdim(:,2),1,:) = conj((CAR_GEN_mdim(:,3,:)
+li*CAR_GEN_mdim(:,4,:))/Nbase); % Associa ad ogni nodo se presente il valore
espresso in p.u. della potenza complessa
Y_car_gen_mdim=zeros(N_nodi_tot,N_nodi_tot,set);
parfor i=1:set
    Y_car_gen_mdim(:,i) = diag(S_compl_pu_mdim(:,i)./V_assegnate_tot_pu_mdim
(:,i).^2);
end
Y_car_gen_mdim = (Y_car_gen_mdim);
Segni = diag([-ones(N_nodi_gen,1);ones((N_nodi_tot-N_nodi_gen),1)]);
Segni(:,1:set)=repmat(Segni,1,1,set);
Y_car_gen_mdim = bsxfun(@times, Y_car_gen_mdim, Segni);
Y_car_gen_mdim(1,1,:) = 1e5 - li*1e5;
Ygg = Yretebase(1:N_nodi_gen,1:N_nodi_gen);
Ygg_mdim(:,1:set)=repmat(Ygg,1,1,set);
Ygl = Yretebase(1:N_nodi_gen,(N_nodi_gen+1):N_nodi_tot);
Ygl_mdim(:,1:set)=repmat(Ygl,1,1,set);
Ylg = Yretebase((N_nodi_gen+1):N_nodi_tot,1:N_nodi_gen);
Ylg_mdim(:,1:set)=repmat(Ylg,1,1,set);
Yll = Yretebase((N_nodi_gen+1):N_nodi_tot,(N_nodi_gen+1):N_nodi_tot);
Yll_mdim(:,1:set)=repmat(Yll,1,1,set);
YGG_mdim = Y_car_gen_mdim(1:N_nodi_gen,1:N_nodi_gen,:);
YLL_mdim = Y_car_gen_mdim((N_nodi_gen+1):N_nodi_tot,(N_nodi_gen+1):
N_nodi_tot,:);
Ytotgg_mdim = YGG_mdim + Ygg_mdim;
Ytotll_mdim = Yll_mdim + YLL_mdim;
inv_A = arrayfun(@(x) inv(Ytotll_mdim(:,x)), 1:set, 'UniformOutput', false);
Yll_1_mdim = cat(3, inv_A{:});
YglYll_1_mdim=zeros(N_nodi_gen,N_nodi_car,set);
Yequi_mdim=zeros(N_nodi_gen,N_nodi_gen,set);
parfor i=1:set
    YglYll_1_mdim(:,i) = Ygl_mdim(:,i)/Ytotll_mdim(:,i);
    Yequi_mdim(:,i) = Ytotgg_mdim(:,i) - YglYll_1_mdim(:,i)*Ylg;
end
inv_B = arrayfun(@(x) inv(Yequi_mdim(:,x)), 1:set, 'UniformOutput', false);
Zequi_mdim = cat(3, inv_B{:});
% --- end ---
%% Fase di inizializzazione per il ciclo PFPD.
% V_assegnate_pu = [CAR_GEN(1:N_nodi_gen,5)./CAR_GEN(1:N_nodi_gen,6)];
Ix_mdim = zeros(N_nodi_gen,1,set);
Ix_mdim(1,1,:) = Ea/Zequi_mdim(1,1,:); % Assegno il valore al solo nodo di
Saldo
Il_mdim = zeros(N_nodi_tot - N_nodi_gen,1,set);
Scarto_mdim = zeros(N_nodi_tot,1,set);

```

```

    Scarto_mdin(1,1,:) = 1 + li;
% Preallocazione degli array che verranno aggiornati durante il ciclo
In_monti = zeros(N_nodi_tot, set);
Nn_monti = zeros(N_nodi_tot, set);
En_sogno = zeros(N_nodi_tot, set);
En_monti = zeros(N_nodi_tot, set);
canc=ones(set,1);
parfor monti=1:set
%% Recupero delle matrici usate per ogni ciclo dalle matrici mdim
    Zequi=Zequi_mdin(:, :,monti);
    YglYll_1=YglYll_1_mdin(:, :,monti);
    Yll_1=Yll_1_mdin(:, :,monti);
    Ylg=Ylg_mdin(:, :,monti);
    V_assegnate_pu=V_assegnate_pu_mdin(:, :,monti);
    Ix=Ix_mdin(:, :,monti);
    Il=Il_mdin(:, :,monti);
    Scarto=Scarto_mdin(:, :,monti);
    YLL=YLL_mdin(:, :,monti);
    Yequi=Yequi_mdin(:, :,monti);
    CAR_GEN=CAR_GEN_mdin(:, :,monti);
    Yretebase=Yretebase_mdin(:, :,monti);
%% Ciclo patavino
    tol = 1e-8;
    tic
    ciclo=0;
    Eng_c=ones(N_nodi_gen,1);
    % Precalcolo di una matrice che verrà utilizzata nel ciclo
    Yll_1Ylg=Yll_1*Ylg;
    % --- PFPD ---
    while any(abs(real(Scarto))>tol | abs(imag(Scarto))>tol)
        ciclo = ciclo + 1;
%% [Eng_c,Eload] = tensioni_pre_ciclo(Zequi,YglYll_1,Yll_1,Ylg,V_assegnate_pu,Ix,
Il);
        Eng = Zequi*(Ix - YglYll_1*Il);
        Eng_c(1:N_nodi_gen,1) = V_assegnate_pu(1:N_nodi_gen,1).*exp(li*angle(Eng(1: ...
N_nodi_gen,1)));
        Eload = -Yll_1Ylg*Eng_c + Yll_1*Il;
    % --- end ---
        En_sogno = [Eng_c;Eload];
%% [Ix,Il] = A_rinnovo_FCC(Eng_c,Eload,YLL,Yequi,YglYll_1);
        Il = -(diag(YLL)).*(1 - abs(Eload).^2)./conj(Eload);
        Ix = Yequi*Eng_c + YglYll_1*Il;
        Ixc_q = -li*(imag(Eng_c.*conj(Ix))./conj(Eng_c));
        Ix(2:N_nodi_gen,1) = Ixc_q(2:N_nodi_gen,1);
    % --- end ---
%% [Scarto] = A06_SCARTI(CAR_GEN,En_sogno,Yretebase);
        cccc = (CAR_GEN(:,1) == abs('C'));
        Nciclo = En_sogno.*conj(Yretebase*En_sogno);
        Pcar = CAR_GEN(cccc,3)/Nbase;
        Qcar = CAR_GEN(cccc,4)/Nbase;
        Scarto = Nciclo(cccc) + (Pcar + li*Qcar);
    % --- end ---
        if ciclo > 300

```

```

        canc(monti,1)=0;
        disp(['La simulazione n. ' num2str(monti) ' ha fallito il calcolo del Power
Flow']);
        break;
    end
    T_loadflow = toc;
    In = (Yretebase)*En_sogno; % Correnti
    entranti/uscenti nei nodi (Vettore colonna), con tali nodi modellizzo la potenza
    entrante e uscente dai nodi che viene data rispettivamente da carichi e generatori.
    Nn = En_sogno.*conj(In)*Nbase; % Potenze complesse
    entranti ai nodi (Vettore colonna).
    Ntotg = - sum(Nn(1:N_nodi_gen)); % Potenza complessa
    totale generata dai nodi.
    Ncarichi = sum(Nn(N_nodi_gen+1:N_nodi_tot)); % Potenza complessa
    totale assorbita dai nodi.
    Pn=real(Nn);
    Qn=imag(Nn);
    P_loss = real(Ntotg) - real(Ncarichi);
    Q_loss = imag(Ntotg) - imag(Ncarichi);
    angolo_tensioni=angle(En_sogno)*180/pi;
    Pn(1)=Pn(1)+Ploss_1+Ploss_2;
% Salvataggio dei dati ti Output
    In_monti(:,monti)=In;
    Nn_monti(:,monti)=Nn;
    En_monti(:,monti)=En_sogno;
    end
% --- end ---
end
T_interfaccia = toc(start);
disp(['Il tempo impiegato per eseguire il Probabilistic Power flow è: ',...
num2str(T_interfaccia)])
clear T_interfaccia
% Function che mi permette l'eliminazione di quei power flow che non hanno
% raggiunto la convergenza.
[En_monti,In_monti,Nn_monti] = elimina_colonne(En_monti,In_monti,Nn_monti, canc);
% - end - %
% Estrazione dei valori finali di Output
En_monti_modulo=abs(En_monti);
In_monti_modulo=abs(In_monti);
Pn_monti=real(Nn_monti);
Qn_monti=imag(Nn_monti);
[media_En_monti_modulo,media_In_monti_modulo,dev_En_monti_modulo,
dev_In_monti_modulo] = mu_std(En_monti_modulo, In_monti_modulo);

```

Figura 87 Codice MatLab per lo studio della rete Siciliana.

Conclusione

Lo sviluppo del *PFPD_PRB* ha ulteriormente dimostrato le potenzialità applicative dell'algoritmo *PFPD*, di origine Patavina: l'uso di un metodo matriciale vettorializzato rispetto ad uno non lineare, fornisce all'algoritmo una maggior rapidità di calcolo rispetto agli ulteriori algoritmi presenti sul mercato utilizzati per lo studio dei flussi di potenza probabilistici (*PPF*). L'algoritmo è stato sottoposto a test e confrontato con DigSilent, dimostrando la sua efficacia e utilizzabilità per le linee di trasmissione in Corrente Alternata (*AC*), in Corrente Continua (*DC*) e per le linee di trasmissione miste Alternata/Continua (*AC/DC*), reti che presentano sia elementi di generazione fossile che elementi di generazione da fonte rinnovabile. In questo modo, è stato possibile integrare la sfera probabilistica che caratterizza la produzione eolica e fotovoltaica, dipendente dalle condizioni climatiche, nello studio dei flussi di potenza. Grazie ad una ricerca approfondita e utilizzando le ultime conoscenze in letteratura, è stato possibile applicare il *Metodo di Monte Carlo* e modellizzare in modo accurato i nodi di generazione rinnovabile della rete, rendendola comparabile con la realtà. Uno studio Probabilistico di tale tipo è fondamentale sia per l'organizzazione dell'architettura delle reti elettriche future sempre più complesse ma anche per lo studio del caricamento delle linee e degli elementi con presenza di generazione (da fonte rinnovabile) e assorbimento (da parte di carichi come batteria e automobili elettriche) aleatorio.

L'applicazione dell'algoritmo *PFPD_PRB* e del *PPF* in generale permette una gestione più efficiente ed affidabile delle reti di trasmissione, migliorando l'orientamento delle scelte per la progettazione e la gestione delle infrastrutture di trasmissione dell'energia elettrica.

Bibliografia

- [1] Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, and R. Seethapathy, "Optimal Renewable Resources Mix for distribution system energy loss minimization," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 360–370, 2010.
- [2] Energy Assessment of Photovoltaic Conversion. Systems. Pedro Vale pedro.vale@ist.utl.pt. Instituto Superior Técnico, Lisboa, Portugal.
- [3] P. Wais, "A review of Weibull functions in wind sector," *Renewable and Sustainable Energy Reviews*, vol. 70, pp. 1099–1107, 2017.
- [4] W. Weibull, "A statistical distribution function of wide applicability," *Journal of Applied Mechanics*, vol. 18, no. 3, pp. 293–297, 1951.
- [5] D. S., M. Abramowitz, and I. A. Stegun, "Handbook of Mathematical Functions with formulas, graphs, and mathematical tables.," *Mathematics of Computation*, vol. 20, no. 93, p. 167, 1966.
- [6] NIST/SEMATECH "- beta distribution," *Handbook of Statistical Distributions with Applications*, pp. 274–285, 2016.
- [7] PowerFactory 2020 User Manual.
- [8] A. Paolucci, *Lezioni di Trasmissione dell'Energia Elettrica*. Padova: Cleup, 1998.
- [9] M. Guarnieri, "Elementi di elettrotecnica circuitale", terza edizione, Edizioni Progetto-Padova, 2014.
- [10] R. Benato, A. Paolucci, and R. Turri, "Power flow solution by a complex admittance matrix method," *European Transactions on Electrical Power*, vol. 11, no. 3, pp. 181–188, 2001.
- [11] R. Benato, "Tecniche matriciali avanzate per l'analisi statica e dinamica nelle reti elettriche di potenza", Tesi di Dottorato all'Università degli Studi di Padova, 1998.
- [12] R. Benato and G. Gardan, "A Novel AC/DC Power Flow: HVDC-LCC/VSC inclusion into the PFPD bus admittance matrix," *IEEE Access*, vol. 10, pp. 38123–38136, 2022.
- [13] Media (statistica). (20 novembre 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6 dicembre 2022, 15:21 da [//it.wikipedia.org/w/index.php?title=Media_\(statistica\)&oldid=130589925](https://it.wikipedia.org/w/index.php?title=Media_(statistica)&oldid=130589925).
- [14] Scarto quadratico medio. (5 marzo 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6/12/22 [//it.wikipedia.org/w/index.php?title=Scarto_quadratico_medio&oldid=126096902](https://it.wikipedia.org/w/index.php?title=Scarto_quadratico_medio&oldid=126096902).
- [15] Campo di variazione. (20 novembre 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6/12/22, 15:24 da [//it.wikipedia.org/w/index.php?title=Campo_di_variazione&oldid=130590375](https://it.wikipedia.org/w/index.php?title=Campo_di_variazione&oldid=130590375).
- [16] Intervallo di confidenza. (30 giugno 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6/12/22 [//it.wikipedia.org/w/index.php?title=Intervallo_di_confidenza&oldid=128153292](https://it.wikipedia.org/w/index.php?title=Intervallo_di_confidenza&oldid=128153292).
- [17] Varianza. (28 maggio 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6 dicembre 2022, 15:27 da [//it.wikipedia.org/w/index.php?title=Varianza&oldid=127565216](https://it.wikipedia.org/w/index.php?title=Varianza&oldid=127565216).
- [18] Distribuzione normale. (10 settembre 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6/12/22 [//it.wikipedia.org/w/index.php?title=Distribuzione_normale&oldid=129288750](https://it.wikipedia.org/w/index.php?title=Distribuzione_normale&oldid=129288750).
- [19] Distribuzione di Weibull. (4 novembre 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6/12/22 [//it.wikipedia.org/w/index.php?title=Distribuzione_di_Weibull&oldid=130275570](https://it.wikipedia.org/w/index.php?title=Distribuzione_di_Weibull&oldid=130275570).

- [20] Distribuzione Beta. (20 novembre 2022). *Wikipedia, L'enciclopedia libera*. Tratto il 6/12/22 //it.wikipedia.org/w/index.php?title=Distribuzione_Beta&oldid=130585252.[20]
- [21] Benato, R. (2022) “A basic AC power flow based on the bus admittance matrix incorporating loads and generators including Slack Bus,” *IEEE Transactions on Power Systems*, 37(2), pp. 1363–1374. Available at: <https://doi.org/10.1109/tpwrs.2021.3104097>.
- [22] Borkowska, B. (1974) “Probabilistic load flow,” *IEEE Transactions on Power Apparatus and Systems*, PAS-93(3), pp. 752–759. Available at: <https://doi.org/10.1109/tpas.1974.293973>.
- [23] P. Chen, Z. Chen, and B. Bak-Jensen, “Probabilistic load flow: A Review,” 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, 2008.
- [24] F. Mišurović and S. Mujović, “Numerical Probabilistic Load Flow Analysis in modern power systems with intermittent energy sources,” *Energies*, vol. 15, no. 6, p. 2038, 2022.
- [25] B. R. Prusty and D. Jena, “A critical review on Probabilistic Load Flow Studies in uncertainty constrained power systems with photovoltaic generation and a new approach,” *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 1286–1302, 2017.
- [26] O. A. Oke and D. W. P. Thomas, “Enhanced cumulant method for probabilistic power flow in systems with wind generation,” 2012 11th International Conference on Environment and Electrical Engineering, 2012.
- [27] X.-F. Wang, Y. Song, and M. Irving, “Modern Power Systems Analysis,” 2008.
- [28] G. Carpinelli, P. Caramia, and P. Varilone, “Multi-linear Monte Carlo simulation method for probabilistic load flow of distribution systems with wind and Photovoltaic Generation Systems,” *Renewable Energy*, vol. 76, pp. 283–295, 2015.
- [29] C. Nemes, R. Ciobanu, and C. Rugina, “Probabilistic analysis of Sky Clearness index for Solar Energy Systems Planning,” *2018 Smart City Symposium Prague (SCSP)*, 2018.
- [30] J. F. Orgill and K. G. T. Hollands, “Correlation equation for hourly diffuse radiation on a horizontal surface,” *Solar Energy*, vol. 19, no. 4, pp. 357–359, 1977.
- [31] Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, and R. Seethapathy, “Optimal Renewable Resources Mix for distribution system energy loss minimization,” *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 360–370, 2010.
- [32] R. Caldon and F. Bignucolo, *Impianti di produzione dell'energia elettrica: Criteri di Scelta e Dimensionamento*. Bologna: Esculapio, 2018.
- [33] P. P. Biswas, P. N. Suganthan, B. Y. Qu, and G. A. J. Amaratunga, “Multiobjective Economic-environmental power dispatch with stochastic wind-solar-small hydro power,” *Energy*, vol. 150, pp. 1039–1057, 2018.
- [34] “Flussi di potenza della rete elettrica italiana calcolati con algoritmo basato sulla matrice complessa alle ammettenze nodali” Giovanni Gardan.
- [35] Distribuzione LogNormale. (10 febbraio 2023). *Wikipedia, L'enciclopedia libera*. Tratto il 6/03/2022//it.wikipedia.org/w/index.php?title=Distribuzione_lognormale&oldid=131992512.
- [36] “Parallel computing toolbox,” *Parallel Computing Toolbox Documentation*. [Online]. Available: <https://www.mathworks.com/help/parallel-computing/>. [Accessed: 05-Apr-2023].

Appendice

Guida alla creazione del foglio *Excel* per l'importazione dei dati da *DigSilent* a *MatLab*.

Tale documento ha lo scopo di fornire la lista delle variabili da inserire all'interno di *Flexible Data* → *Variable Selection* → *Editor* di *DigSilent* con lo scopo di ottenere come risultati i dati utili all'importazione della rete su *Matlab* e alla successiva analisi tramite lo sfruttamento di algoritmi Patavini per il calcolo dei flussi di Potenza. Di seguito riportati *sheet for sheet* con eventuali correzioni così da adattare l'importazione all'algoritmo.

TERMINAL

Eliminare dai *Terminal* i nodi *HVDC* compresi quelli riferiti alla messa a terra dei convertitori, tale processo, va fatto a mano e ponendo *Energized* → 0 ed *OutofService* → 1.

e:ciEnergized
e:outserv
e:uknom

TRAFO 2

e:bushv_bar
e:buslv_bar
e:outserv
t:curmg
t:pfe
t:uktr
t:strn
e:ciEnergized
t:nt2ag
t:utrn_1
t:utrn_h
t:pcutr

TRAFO 3

t:strn3_h
t:strn3_m
t:strn3_l
t:utrn3_h
t:utrn3_m
t:utrn3_l
t:nt3ag_h
t:nt3ag_l
t:uktr3_h
t:uktr3_m
t:uktr3_l
t:pcut3_h
t:pcut3_m
t:pcut3_l
e:ciEnergized
e:outserv
e:bushv_bar
e:busmv_bar
e:buslv_bar

LINEE

Eliminiamo le linee *HVDC* che non esistono in *PPFD* essendo un algoritmo per la risoluzione di reti *AC*, le informazioni riguardanti le reti in corrente continua sono contenute all'interno dei nodi fittizi e negli altri *sheet* (*LCC*, *VSC*, *LINEE LCC*, *LINEE VSC*).

e:bus1_bar
e:bus2_bar
e:dline
r:typ_id:e:uline
t:cline
e:outserv
t:lline
t:rline
t:gline
e:ciEnergized

LINEE VSC

Inserimento fatto a mano, si tratta di pochi valori, le colonne sono le seguenti:

Name	Grid	Terminal i	Terminal j	Outof service	Energized	Length	Unom	R	P1	P2	Type_HVDC
------	------	------------	------------	---------------	-----------	--------	------	---	----	----	-----------

Spiegazione trattata nei particolari nel Capitolo 1.3.

VSC

Spiegazione trattata nei particolari nel Capitolo 1.3.

e:busac_bar
e:Pnold
e:swtLossFactor
e:resLossFactor
e:outserv
e:ciEnergized
t:Unom

LINEE LCC

Inserimento fatto a mano, si tratta di pochi valori, le colonne sono le seguenti:

Name	Grid	Terminal i	Terminal j	Outof service	Energized	Length	Unom	R	P1	P2	Q1	Q2	Type_HVDC
------	------	------------	------------	---------------	-----------	--------	------	---	----	----	----	----	-----------

Spiegazione trattata nei particolari nel Capitolo 1.3.

LCC

Spiegazione trattata nei particolari nel Capitolo 1.3.

e:busac_bar
e:outserv
e:ciEnergized
t:Unom

Mancano le colonne *No_loadlosses* (colonna 3), *SwitchingLossFactor* (colonna 4), *ResistiveLossFactor* (colonna 5), vanno aggiunti a mano e il valore che va indicato su di essi è pari a 0.2 per tutti, inoltre vanno inserite in una determinata posizione essendo che la function *Metodo_IRPE.m* è fatta in modo tale che essa vada a prendere quelle determinate colonne.

SHUNT FILTER

e:outserv
e:ushnm
e:ccap
e:rlrea
e:bus1_bar
e:shtype
e:gparac
e:ciEnergized
e:rrea
e:ncapa

GENERATORI

e:bus1_bar
e:pgini
e:qgini
t:sgn
e:outserv
t:Q_max
t:Q_min
t:ugn
e:av_mode
e:ciEnergized
e:usetp

STATIC

L'importazione dei generatori statici presentano alcune vicissitudini, essi vanno inseriti nello stesso sheet *Excel* riferito ai generatori, una volta svolta tale operazione bisognerà sistemare le colonne facendole combaciare in quanto nel caso degli Static generator mancherà la colonna Rtd. Voltage, la si aggiunge a mano e le si associa il valore 0.

e:bus1_bar
e:pgini
e:qgini
e:sgn
e:outserv
e:q_max
e:q_min
e:av_mode
e:ciEnergized
e:usetp
e:cosn

copio e incollo a mano la categoria "*Plant*" dove saranno di interesse solo gli impianti con "*Wind*" e "*Photovoltaic*", tutto il resto (compresi i generatori sincroni normali), gli elementi categorizzati come "*Storage*" vengono trattati come degli Static generator senza distribuzione applicata; nel caso di generatori sincroni gli verrà associato il valore "*Syncro*" così da ricordarci l'origine da fonte fossile.

CARICHI

e:bus1_bar
e:plini
e:qlini
e:outserv
t:tpf
t:kpf
e:ciEnergized
e:u0

OUT OF SERVICE / ENERGIZED

Nei fogli Excel nel caso non fossero presenti andranno inserite sempre le due colonne seguenti:

- *Out of service*, che assume valore 1 quando il terminale è fuori servizio e 0 quando è in funzione.
- *Energized*, assume valore 1 quando la linea è energizzata e 0 quando invece non lo è.

Lo scopo di tali colonne è quello di poter escludere dal Load Flow determinati elementi che normalmente possono venir prese in analisi.

ALTRO

Per quanto riguarda i fogli Switch, Shunt Filter o quei fogli che nel caso di studio non vengono presi in considerazione è obbligatorio inserirli comunque all'interno del documento Excel, compilando tutti i campi ma inserendo nelle colonne *Out of service* il parametro 1 e nelle colonne *Energized* il parametro 0. Questo perché la function *A_INTERFACCIA_IRPE.m* è stata costruita per prendere in analisi tutte le categorie di elementi all'interno di una rete e di conseguenza se omettiamo qualche valore esso non funziona.

SELEZIONE DEI VALORI OTTENUTI DAL PROBABILISTIC POWER FLOW DA AVERE IN OUTPUT.

Andando ad inserire tali valori all'interno degli output ricercati nel calcolo del Probabilistic Power Flow, si avranno per parametro i seguenti risultati:

m:u:std, modulo delle tensioni nodali deviazione standard [p.u.];
m:u:avg, modulo delle tensioni nodali media [p.u.];
m:u, modulo delle tensioni nodali [p.u.];
m:u:max, modulo delle tensioni nodali massimo [p.u.];
m:u:min, modulo delle tensioni nodali minimo [p.u.];
m:phiu:avg, angolo delle tensioni nodali media [deg];
m:phiu:std, angolo delle tensioni nodali deviazione standard [deg];

A_INTERFACCIA_DS_MT.m

Codice classico per il trasferimento dei dati da *Excel* a *MatLab* ottenuti a partire dall'operazione precedente descritta in "Guida alla creazione del foglio Excel per l'importazione dei dati da DigSilent a MatLab".

```
%% _____ INTERFACCIA_DATI_DIGSILENT_MATLAB _____
format long
path(path, 'A_INTERFACCIA_DS_MT')
E00_Pulizia_ott

Nbase = 100e6;
omegan = 2*pi*50;

%% DEFINIRE FILE DI INPUT

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILE AC/DC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_VSC_MONO_2.xlsx';           %Rete✓
A
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_VSC_BIPO_2_NEW.xlsx';       %Rete✓
B
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_VSC_MONO_3.xlsx';           %Rete✓
C
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_VSC_BIPO_3_NEW.xlsx';       %Rete✓
D

% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_LCC_MONO_2_NEW.xlsx';       %Rete✓
E
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_LCC_BIPO_2_NEW.xlsx';       %Rete✓
F
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_LCC_MONO_3_NEW.xlsx';       %Rete✓
G
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_LCC_BIPO_3_NEW.xlsx';       %Rete✓
H

% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_VSC_EXTERNAL_GRID_NEW.xlsx';
% FILE_INPUT = '000_DATA_excel_from_DS_ott\18_NODI_LCC_EXTERNAL_GRID_NEW.xlsx';

% FILE_INPUT = '000_DATA_excel_from_DS_ott\60_NODI_VSC_NEW.xlsx';             %Rete✓
I
% FILE_INPUT = '000_DATA_excel_from_DS_ott\60_NODI_LCC_NEW.xlsx';             %Rete✓
L
% FILE_INPUT = '000_DATA_excel_from_DS_ott\60_NODI_MISTO_NEW.xlsx';           %Rete✓
M

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[A,B] = xlsfinfo(FILE_INPUT);
sheetValid = any(strcmp(B, 'LCC'));

if sheetValid == 1

tic

%% IMPORTAZIONE DATI
% Mi importa i dati da Excel a Matlab sotto forma di Tabelle con i
% nominativi di output della function seguente "E01_0_IMPORTAZIONE_ott_HVDC"
```

```

% che risultano essere differenti da quelli presenti nel file excel -->
% forse lo modifico se ho tempo devp valutare.
[TERMINAL, TRAF02, TRAF03, RETE, FILTER, CARICHI, GENERATORI, RETE_VSC, VSC, RETE_LCC, LCC] = ✓
...
E01_0_IMPORTAZIONE_ott_HVDC(FILE_INPUT);

%% ESTRAPOLAZIONE DATI NODI
[NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, NOMI_DIGSILENT_GEN] = E02_NOME_BUS_ott(TERMINAL, ✓
GENERATORI);

%% ESTRAZIONE DATI TRASFORMATORI
%Estrapolazione dati trasformatore a 2 avv. _____
[TRASF, control_nod, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS] = E03_FORM_TRASF_2W_ott_HVDC...
(TRAFO2, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS);
%Estrapolazione dati trasformatore a 3 avv. _____
[TRASF, TRASF_TRE, control_nod] = E03_FORM_TRASF_3W_ott(TRASF, TRAF03, ✓
NOMI_DIGSILENT_BUS, ...
NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI LINEE
[LINEE, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_ott_HVDC(...
RETE, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI LINEE_DC
[LINEE_VSC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_VSC(...
RETE_VSC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI CONVERTITORI VSC
[CONV_VSC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_VSC(...
VSC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

[LINEE_LCC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_LCC(...
RETE_LCC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI CONVERTITORI LCC
[CONV_LCC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_LCC(...
LCC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI SHUNT
[SHUNT] = E05_FORM_SHUNT_ott_HVDC(Nbase, FILTER, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, ✓
omegan);

[Ea, CAR_GEN, GEN, Qlim, GENasPQ, CAR_S, CAR_D, GEN_FIT, GEN_FIT_1, GEN_FIT_2, GEN_TOT, ✓
CAR_FIT_1M, ...
CAR_FIT_1B, CAR_FIT_2M, CAR_FIT_2B, CAR_TOT, CAR_FIT, OTHER] = ✓
E06_FORM_CAR_GEN_ott_HVDC(CARICHI, GENERATORI, ...
NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod, LINEE_VSC, CONV_VSC, ✓
RETE_VSC, VSC, RETE_LCC, LINEE_LCC, CONV_LCC);

```



```

E06_2_ACCORPAMENTO_CARICHI

%% CREAZIONE LINEE
[COR_NODI,CAR_GEN,LINEE,Qlim] = E07_CREAZIONE_DI_LINEE_ott_HVDC(LINEE,CAR_GEN,Qlim);

%% N9 RIORDINO NODI CON CAMBIO NUMERAZIONE
[SHUNT,nodi_originali,Qlim,CAR_GEN,RETE,TRASF,TRASF_TRE,GENasPQ] ✓
=E08_CAMBIO_NUM_ott_HVDC...
(SHUNT,CAR_GEN,LINEE,TRASF,TRASF_TRE,GEN,Qlim,NOMI_DIGSILENT_BUS,COR_NODI,GENasPQ);

%% OPERAZIONI FINALI
N_nodi_tot = size(CAR_GEN,1);
N_nodi_gen = size(CAR_GEN(CAR_GEN(:,1)=='G'),1);
N_nodi_car = N_nodi_tot-N_nodi_gen;

clear control_nod COR_NODI
clear GEN LINEE
clear NOMI_DIGSILENT_BUS NOMI_MATLAB_BUS

T_interfaccia = toc;
disp(['Il tempo impiegato per acquisire i dati da PowerFactory è:      ',...
      num2str(T_interfaccia)])

%% Cancellare per non creare confusione. Cancellazione valida solo
% per fare il solo Power Flow va bene.
clear CAR_DIN
clear CAR_ROT
clear CARICHI
clear dop
clear FILE_INPUT
clear FILTER
clear GEN_DIN
% clear GENasPQ
clear GENERATORI
clear k
clear T_interfaccia
clear CAR_DIN
clear TERMINAL
clear TRAFO2
clear TRAFO3
clear ccc
clear fff

%% Salvataggio di tutti i dati di output nei percorsi seguenti:
save('C:\Users\ostinic48425\Desktop\ALGORITMI\PFPPD metodo IRPE\Dati\Dati.mat');

else

tic

```

```

%% IMPORTAZIONE DATI
[TERMINAL, TRAF02, TRAF03, RETE, FILTER, CARICHI, GENERATORI] = ...
    E01_0_IMPORTAZIONE_ott(FILE_INPUT);

%% ESTRAPOLAZIONE DATI NODI
[NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, NOMI_DIGSILENT_GEN] = E02_NOME_BUS_ott(TERMINAL, ✓
GENERATORI);

%% ESTRAZIONE DATI TRASFORMATORI
%Estrapolazine dati trasformatore a 2 avv. _____
[TRASF, control_nod, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS] = E03_FORM_TRASF_2W_ott...
    (TRAF02, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS);
%Estrapolazione dati trasformatore a 3 avv. _____
[TRASF, TRASF_TRE, control_nod] = E03_FORM_TRASF_3W_ott (TRASF, TRAF03, ✓
NOMI_DIGSILENT_BUS, ...
    NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZINE DATI LINEE
[LINEE, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_ott(...
    RETE, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI SHUNT
[SHUNT] = E05_FORM_SHUNT_ott (Nbase, FILTER, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, ...
    omegan);

%% ESTRAPOLAZIONE DATI CARICHI E GENERATORI
[Ea, CAR_GEN, GEN, Qlim, GENasPQ] = E06_FORM_CAR_GEN_ott (CARICHI, GENERATORI, ...
    NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

E06_2_ACCORPAMENTO_CARICHI
%% CREAZIONE LINEE
[COR_NODI, CAR_GEN, LINEE, Qlim] = E07_CREAZIONE_DI_LINEE_ott (LINEE, CAR_GEN, Qlim);

%% N9 RIORDINO NODI CON CAMBIO NUMERAZIONE
[SHUNT, nodi_originali, Qlim, CAR_GEN, RETE, TRASF, TRASF_TRE, GENasPQ]-E08_CAMBIO_NUM_ott...
(SHUNT, CAR_GEN, LINEE, TRASF, TRASF_TRE, GEN, Qlim, NOMI_DIGSILENT_BUS, COR_NODI, GENasPQ);

%% ESTRAPOLAZIONE MATRICI FINALI
[CAR_GEN, GEN_DIN, CAR_ROT, CAR_DIN]=E09_FORMAZIONE_MATRIX_DINAMICA_ott (CAR_GEN);

%% OPERAZIONI FINALI
N_nodi_tot = size(CAR_GEN,1);
N_nodi_gen = size(CAR_GEN(CAR_GEN(:,1)=='G'),1);
N_nodi_car = N_nodi_tot-N_nodi_gen;
Nomi_gen=NOMI_DIGSILENT_GEN;
% clear control_nod COR_NODI
clear GEN
clear LINEE
%% RIMUOVERE IL SEGUENTE APPUNTO PER TORNARE COME IN ORIGINE
% clear NOMI_DIGSILENT_BUS NOMI_MATLAB_BUS

```

```

T_interfaccia = toc;
disp(['Il tempo impiegato per acquisire i dati da PowerFactory è: ',...
      num2str(T_interfaccia)])

%% Cancellare per non creare confusione. Cancellazione valida solo
% per fare il solo Power Flow va bene.
clear CAR_DIN
clear CAR_ROT
clear CARICHI
clear dop
clear FILE_INPUT
clear FILTER
clear GEN_DIN
% clear GENasPQ
% clear GENERATORI
clear k
clear T_interfaccia
clear CAR_DIN
clear TERMINAL
clear TRAF02
clear TRAF03
clear ccc
clear fff

%% Salvataggio di tutti i dati di output nei percorsi seguenti:
save('C:\Users\ostinic48425\Desktop\ALGORITMI\PFPPD metodo IRPE\Dati\Dati.mat');
end

```

A_INTERFACCIA_IRPE_PRB.m

Questo codice è stato modificato rispetto al precedente in quanto permette la creazione della matrice *PLANT* che trasporta in *MatLab* le informazioni riguardanti la posizione e la tipologia degli static generator associati alla fonte rinnovabile.

```
%% _____ INTERFACCIA_DATI_DIGSILENT_MATLAB _____
format long
path(path, 'A_INTERFACCIA_DS_MI')
E00_Pulizia_ott

Nbase = 100e6;
omegan = 2*pi*50;

%% DEFINIRE FILE DI INPUT
%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% FILE DI AC %%%%%%%%%%%%%%%%%%%%%%%%%

FILE_INPUT = '000_DATA_EXCEL_from_DS_ott_MONTECARLO\Sicilia.xlsx';

%%%%%%%%%%%%%%%%%%%%%%%%

[A,B] = xlsfinfo(FILE_INPUT);
sheetValid = any(strcmp(B, 'LCC'));

if sheetValid == 1

tic

%% IMPORTAZIONE DATI
% Mi importa i dati da Excel a Matlab sotto forma di Tabelle con i
% nominativi di output della function seguente "E01_0_IMPORTAZIONE_ott_HVDC"
% che risultano essere differenti da quelli presenti nel file excel -->
% forse lo modifico se ho tempo devp valutare.
[TERMINAL, TRAF02, TRAF03, RETE, FILTER, CARICHI, GENERATORI, RETE_VSC, VSC, RETE_LCC, LCC] = ✓
...
                                                    E01_0_IMPORTAZIONE_ott_HVDC(FILE_INPUT);

%% ESTRAPOLAZIONE DATI NODI
[NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, NOMI_DIGSILENT_GEN] = E02_NOME_BUS_ott(TERMINAL, ✓
GENERATORI);

%% ESTRAZIONE DATI TRASFORMATORI
%Estrapolazione dati trasformatore a 2 avv. _____
[TRASF, control_nod, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS] = E03_FORM_TRASF_2W_ott_HVDC...
(TRAF02, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS);
%Estrapolazione dati trasformatore a 3 avv. _____
[TRASF, TRASF_TRE, control_nod] = E03_FORM_TRASF_3W_ott(TRAF03, ✓
NOMI_DIGSILENT_BUS, ...
                                                    NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI LINEE
[LINEE, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_ott_HVDC(...
RETE, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI LINEE_DC
[LINEE_VSC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_VSC(...
RETE_VSC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);
```

```

%% ESTRAPOLAZIONE DATI CONVERTITORI VSC
[CONV_VSC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_VSC(...
    VSC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

[LINEE_LCC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_LCC(...
    RETE_LCC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI CONVERTITORI LCC
[CONV_LCC, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_LCC(...
    LCC, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

%% ESTRAPOLAZIONE DATI SHUNT
[SHUNT] = E05_FORM_SHUNT_ott_HVDC(Nbase, FILTER, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, ✓
    omegan);

[PLANT, Ea, CAR_GEN, GEN, Qlim, GENasPQ, CAR_S, CAR_D, GEN_FIT, GEN_FIT_1, GEN_FIT_2, GEN_TOT, ✓
    CAR_FIT_1M, ...
    CAR_FIT_1B, CAR_FIT_2M, CAR_FIT_2B, CAR_TOT, CAR_FIT, OTHER] = ✓
E06_FORM_CAR_GEN_ott_PLANT(CARICHI, GENERATORI, ...
    NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod, LINEE_VSC, CONV_VSC, ✓
    RETE_VSC, VSC, RETE_LCC, LINEE_LCC, CONV_LCC);

E06_2_ACCORPAMENTO_CARICHI_PLANT

%% CREAZIONE LINEE
[COR_NODI, CAR_GEN, LINEE, Qlim] = E07_CREAZIONE_DI_LINEE_ott_HVDC_PLANT(LINEE, CAR_GEN, ✓
    Qlim);

%% N9 RIORDINO NODI CON CAMBIO NUMERAZIONE
[PLANT, SHUNT, nodi_originali, Qlim, CAR_GEN, RETE, TRASF, TRASF_TRE, GENasPQ] ✓
=E08_CAMBIO_NUM_ott_HVDC_PLANT...
(SHUNT, CAR_GEN, LINEE, TRASF, TRASF_TRE, GEN, Qlim, NOMI_DIGSILENT_BUS, COR_NODI, GENasPQ, ✓
    PLANT);

%% OPERAZIONI FINALI
N_nodi_tot = size(CAR_GEN, 1);
N_nodi_gen = size(CAR_GEN(CAR_GEN(:, 1) == 'G'), 1);
N_nodi_car = N_nodi_tot - N_nodi_gen;

clear control_nod COR_NODI
clear GEN LINEE
clear NOMI_DIGSILENT_BUS NOMI_MATLAB_BUS

T_interfaccia = toc;
disp(['Il tempo impiegato per acquisire i dati da PowerFactory è: ', ...

```

```

num2str(T_interfaccia)])

%% Cancellare per non creare confusione. Cancellazione valida solo
% per fare il solo Power Flow va bene.
clear CAR_DIN
clear CAR_ROT
clear CARICHI
clear dop
clear FILE_INPUT
clear FILTER
clear GEN_DIN
% clear GENasPQ
clear GENERATORI
clear k
clear T_interfaccia
clear CAR_DIN
clear TERMINAL
clear TRAF02
clear TRAF03
clear ccc
clear fff

%% Salvataggio di tutti i dati di output nei percorsi seguenti:
save('C:\Users\ostinic48425\Desktop\ALGORITMI\Sicilia\Dati\Dati.mat');

else

    tic

    %% IMPORTAZIONE DATI
    [TERMINAL, TRAF02, TRAF03, RETE, FILTER, CARICHI, GENERATORI] = ...
        E01_0_IMPORTAZIONE_ott(FILE_INPUT);

    %% ESTRAPOLAZIONE DATI NODI
    [NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, NOMI_DIGSILENT_GEN] = E02_NOME_BUS_ott(TERMINAL, ✓
GENERATORI);

    %% ESTRAZIONE DATI TRASFORMATORI
    %Estrapolazine dati trasformatore a 2 avv. _____
    [TRASF, control_nod, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS] = E03_FORM_TRASF_2W_ott...
        (TRAF02, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS);
    %Estrapolazione dati trasformatore a 3 avv. _____
    [TRASF, TRASF_TRE, control_nod] = E03_FORM_TRASF_3W_ott(TRASF, TRAF03, ✓
NOMI_DIGSILENT_BUS, ...
        NOMI_MATLAB_BUS, control_nod);

    %% ESTRAPOLAZINE DATI LINEE
    [LINEE, control_nod, NOMI_MATLAB_BUS, NOMI_DIGSILENT_BUS] = E04_FORM_RETE_ott(...
        RETE, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

    %% ESTRAPOLAZIONE DATI SHUNT
    [SHUNT] = E05_FORM_SHUNT_ott(Nbase, FILTER, NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, ...

```

```

omegan);

%% ESTRAPOLAZIONE DATI CARICHI E GENERATORI
[Ea, CAR_GEN, GEN, Qlim, GENasPQ] = E06_FORM_CAR_GEN_ott(CARICHI, GENERATORI, ...
    NOMI_DIGSILENT_BUS, NOMI_MATLAB_BUS, control_nod);

E06_2_ACCORPAMENTO_CARICHI
%% CREAZIONE LINEE
[COR_NODI, CAR_GEN, LINEE, Qlim] = E07_CREAZIONE_DI_LINEE_ott(LINEE, CAR_GEN, Qlim);

%% N9 RIORDINO NODI CON CAMBIO NUMERAZIONE
[SHUNT, nodi_originali, Qlim, CAR_GEN, RETE, TRASF, TRASF_TRE, GENasPQ]=E08_CAMBIO_NUM_ott...
(SHUNT, CAR_GEN, LINEE, TRASF, TRASF_TRE, GEN, Qlim, NOMI_DIGSILENT_BUS, COR_NODI, GENasPQ);

%% ESTRAPOLAZIONE MATRICI FINALI
[CAR_GEN, GEN_DIN, CAR_ROT, CAR_DIN]=E09_FORMAZIONE_MATRIX_DINAMICA_ott(CAR_GEN);

%% OPERAZIONI FINALI
N_nodi_tot = size(CAR_GEN,1);
N_nodi_gen = size(CAR_GEN(CAR_GEN(:,1)=='G'),1);
N_nodi_car = N_nodi_tot-N_nodi_gen;
Nomi_gen=NOMI_DIGSILENT_GEN;
% clear control_nod COR_NODI
clear GEN
clear LINEE
%% RIMUOVERE IL SEGUENTE APPUNTO PER TORNARE COME IN ORIGINE
% clear NOMI_DIGSILENT_BUS NOMI_MATLAB_BUS

T_interfaccia = toc;
disp(['Il tempo impiegato per acquisire i dati da PowerFactory è: ',...
    num2str(T_interfaccia)])

%% Cancellare per non creare confusione. Cancellazione valida solo
% per fare il solo Power Flow va bene.
clear CAR_DIN
clear CAR_ROT
clear CARICHI
clear dop
clear FILE_INPUT
clear FILTER
clear GEN_DIN
% clear GENasPQ
% clear GENERATORI
clear k
clear T_interfaccia
clear CAR_DIN
clear TERMINAL
clear TRAF02
clear TRAF03
clear ccc
clear fff

%% Salvataggio di tutti i dati di output nei percorsi seguenti:
save('C:\Users\ostinic48425\Desktop\ALGORITMI\Sicilia\Dati\Dati.mat');

```

Power_curve_wind.m

Codice automatizzato, per l'associazione e il calcolo delle potenze da associare ai nodi *PQ* di generazione eolica.

```
function [P_r_w,prova_w] = power_curve_wind(set, r_w,PLANT)

% Inizializza P_r_w a zero
P_r_w = zeros(set, size(PLANT,1));

% Applica le condizioni sulla funzione di potenza in un unico passaggio
% vettoriale
r_w(r_w < 3.5) = 0;
r_w(r_w > 3.5 & r_w < 14) = 0.095*r_w(r_w > 3.5 & r_w < 14) - 0.33333333;
r_w(r_w > 14 & r_w < 25) = 1;
r_w(r_w > 25) = 0;

% Calcola la funzione di potenza
wind=PLANT(:,2);
wind(wind < 1 )=0;
wind(wind >1 )=0;
if ~isempty(wind)
for i=1:size(PLANT,1)
    P_r_w(:,i)=-r_w.*PLANT(i,3).*PLANT(i,4);
end
end
prova_w=P_r_w;
P_r_w=P_r_w.*wind';
```

Power_solar.m

Codice automatizzato, per l'associazione e il calcolo delle potenze da associare ai nodi *PQ* di generazione fotovoltaica.

```
function [P_r_s,prova_s]=power_solar(set,r_s,PLANT,CAR_GEN)
% Calcola la funzione di potenza
P_r_s=zeros(set,size(PLANT,1));
PLANT=[PLANT,CAR_GEN(:,3)];
solar=PLANT(:,2);
solar(solar < 2 )=0;
solar(solar > 2 )=0;
solar(solar > 1 & solar < 3)=1;
if ~isempty(solar)
for i=1:size(PLANT,1)
    P_r_s(:,i)=-r_s.*PLANT(i,5);
end
end
PLANT(:,5)=[];
prova_s=P_r_s;
P_r_s=P_r_s.*solar';
```