



UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE

FACOLTÀ DI INGEGNERIA  
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E  
DELL'AUTOMAZIONE

---

# Studio e implementazione di modelli matematici di elettrolizzatori per la produzione di idrogeno a fini diagnostici

Study and implementation of mathematical models of electrolysers  
for the production of hydrogen for diagnostic purposes

Candidato:  
**Francesco Silvi**

Relatore:  
**Prof. Alessandro Freddi**

Correlatore:  
**Luca Mazzuferi**

Anno Accademico 2022-2023



# Sommario

La sostenibilità rappresenta una delle sfide più importanti per l'umanità al giorno d'oggi, negli ultimi anni si sta spingendo sempre di più verso delle fonti di energia che non provengano dai combustibili fossili. Una delle alternative è rappresentata dall'idrogeno, utilizzabile sia come combustibile che come forma in cui immagazzinare energia elettrica. Essendo però l'idrogeno un gas non naturalmente presente in atmosfera è necessario produrlo, a questo scopo sono utilizzate diverse metodologie, ognuna con una sua produzione di CO<sub>2</sub>. Fra le tecniche che vantano le minori emissioni di CO<sub>2</sub> c'è l'elettrolisi abbinata all'utilizzo di sorgenti di energia rinnovabile.

Nel presente studio si è implementato un modello matematico di un elettrolizzatore a membrana polimerica, che tramite la calibrazione dei parametri è in grado di allinearsi alle misurazioni presenti su dei dataset acquisiti in un banco di misura. Partendo dal modello allineato ad un elettrolizzatore in salute, attraverso l'errore presente fra l'uscita del modello e le misurazioni sul dispositivo, si punta ad ottenere delle informazioni sullo stato di salute del componente. Essere a conoscenza di quella che è la condizione del dispositivo può aiutare a prendere provvedimenti al fine di prolungarne l'utilizzo e programmarne la sostituzione.



# Indice

|   |           |
|---|-----------|
| <b>1. Introduzione</b>  | <b>1</b>  |
| <b>2. Elettrolizzatori e celle a combustibile</b>                   | <b>3</b>  |
| 2.1. Differenze, tipologie ed utilizzi . . . . .                    | 3         |
| 2.2. Elettrolizzatori PEM . . . . .                                 | 4         |
| 2.2.1. Composizione . . . . .                                       | 4         |
| 2.2.2. Modellazione matematica . . . . .                            | 6         |
| <b>3. Implementazione del modello su Simulink</b>                   | <b>11</b> |
| 3.1. Dataset a disposizione . . . . .                               | 11        |
| 3.2. Simulink . . . . .   | 11        |
| 3.2.1. Ingressi e uscite . . . . .                                  | 12        |
| 3.2.2. Parametri . . . . .  | 14        |
| 3.2.3. Produzione gas e conversione in NL/min . . . . .             | 15        |
| 3.2.4. Tensione . . . . .   | 16        |
| 3.3. Calibrazione dei parametri . . . . .                           | 22        |
| 3.3.1. Prima fase: calibrazione manuale . . . . .                   | 22        |
| 3.3.2. Seconda fase: Calibrazione tramite PSO . . . . .             | 27        |
| <b>4. Degradazione e guasti di un elettrolizzatore</b>              | <b>35</b> |
| 4.1. Tipologie di degradazione in un elettrolizzatore PEM . . . . . | 35        |
| 4.2. Possibili approcci alla prognosi . . . . .                     | 35        |
| 4.3. Utilizzo del modello . . . . .                                 | 36        |
| 4.3.1. Selezione delle finestre . . . . .                           | 40        |
| 4.3.2. Calcolo degli indici . . . . .                               | 44        |
| 4.3.3. Stima parametrica inversa . . . . .                          | 52        |
| <b>5. Conclusioni e sviluppi futuri</b>                             | <b>59</b> |
| 5.1. Conclusioni . . . . .  | 59        |
| 5.2. Sviluppi futuri . . . . .                                      | 60        |
| <b>A. Codice Matlab</b>   | <b>63</b> |
| A.1. Importazione dei dati dal dataset . . . . .                    | 63        |
| A.2. Formazione delle finestre . . . . .                            | 63        |
| A.3. Aggiornamento della finestra . . . . .                         | 68        |
| A.4. Calcolo degli indici sulla finestra . . . . .                  | 69        |

*Indice*

|  |    |
|--|----|
| A.5. Stima parametrica inversa . . . . .                         | 70 |
| A.6. Inversione equazioni per il calcolo dei parametri . . . . . | 71 |
| A.7. Algoritmo PSO . . . . .                                     | 72 |

## Elenco delle figure

|   |    |
|---|----|
| 2.1. Schema funzionale di un elettrolizzatore PEM [1] . . . . .   | 5  |
| 3.1. Vista complessiva del modello Simulink . . . . .   | 12 |
| 3.2. Elaborazione dei segnali . . . . .   | 13 |
| 3.3. Caricamento dal workspace della tensione e della produzione dei gas<br>contenute nel dataset . . . . .                           | 13 |
| 3.4. Sottosistema per l'inserimento dei parametri . . . . .   | 14 |
| 3.5. Sottosistema per il calcolo della produzione di idrogeno e ossigeno in<br>mol/s . . . . .  | 15 |
| 3.6. Sottosistema per l'elaborazione dei gas prodotti . . . . .   | 17 |
| 3.7. Funzione di trasferimento applicata al flusso dell'idrogeno . . . . .  | 17 |
| 3.8. Calcolo $\eta_F$ in funzione della densità di corrente . . . . .   | 17 |
| 3.9. Calcolo della tensione diviso per i quattro contributi . . . . .   | 18 |
| 3.10. Opzioni per il calcolo della tensione a circuito aperto . . . . .   | 19 |
| 3.11. Prima opzione per il calcolo della tensione a circuito aperto con<br>selezione del calcolo della tensione reversibile . . . . . | 19 |
| 3.12. Opzioni per il calcolo della tensione reversibile . . . . .   | 20 |
| 3.13. Calcolo delle perdite ohmiche . . . . .   | 21 |
| 3.14. Calcolo perdite di attivazione . . . . .  | 21 |
| 3.15. Calcolo delle perdite di concentrazione . . . . .   | 21 |
| 3.16. Calcolo degli indici di fit . . . . .   | 22 |
| 3.17. Calcolo dell'efficienza dell'elettrolizzatore . . . . .   | 23 |
| 3.18. Curve di polarizzazione dell'elettrolizzatore . . . . .   | 25 |
| 3.19. Ingressi del dataset utilizzato per la calibrazione manuale . . . . .   | 26 |
| 3.20. Uscite prima della calibrazione manuale . . . . .   | 26 |
| 3.21. Uscite dopo la calibrazione manuale . . . . .   | 27 |
| 3.22. Uscite dopo la prima calibrazione con PSO . . . . .   | 29 |
| 3.23. Miglioramento dell'IAE durante la calibrazione con PSO . . . . .  | 30 |
| 3.24. Movimento dello sciame durante la calibrazione . . . . .  | 32 |
| 3.25. Confronto fra la tensione stimata dal modello e le misurazioni del<br>dataset a seguito della calibrazione in 3 fasi . . . . .  | 34 |
| 4.1. Corrente, temperature e pressioni di due dataset BoT . . . . .   | 38 |
| 4.2. Esempi di misurazioni di flusso non utilizzabili . . . . .   | 39 |
| 4.3. Corrente, temperature e pressioni del Dataset EoT . . . . .  | 40 |

*Elenco delle figure*

|   |    |
|---|----|
| 4.4. Blu: Valore dei parametri seguito delle calibrizioni sui 4 dataset BoT; Rosso: Valore medio delle prima 4 colonne; Giallo: Valore dei parametri seguito della calibrazione sul dataset EoT; . . . . .  | 41 |
| 4.5. Variazione dei parametri calibrati per il dataset EoT in relazione alla media di quelli calibrati sui 4 dataset BoT . . . . .  | 42 |
| 4.6. Modulo dell'errore fra il modello calibrato con i parametri medi BoT e le misurazioni dei 4 dataset BoT . . . . .  | 42 |
| 4.7. Modulo dell'errore fra il modello calibrato con i parametri medi BoT e la tensione del dataset EoT . . . . .   | 43 |
| 4.8. Diagramma di flusso dell'algoritmo per la formazione delle finestre e calcolo degli indici . . . . .   | 45 |
| 4.9. Finestre temporali selezionate sulla corrente e la temperatura del dataset EoT 111115 Le coppie di rette, tracciate sopra i due segnali di ingresso, rappresentano le zone in cui sono state riconosciute delle finestre, ogni colore rappresenta una finestra differente. L'estensione orizzontale delle rette rappresenta gli istanti di tempo che fanno parte della finestra, mentre l'altezza alla quale sono graficate è pari al valor medio del segnale di ingresso in quell' intervallo di tempo. . . . . | 46 |
| 4.10. Tipologie di andamenti identificati a seconda della posizione della finestra (rappresentata in rosso) rispetto alle posizioni $x_1$ e $x_2$ dei flessi del polinomio . . . . .  | 48 |
| 4.11. Classificazione delle zone A, B, C e D . . . . .  | 49 |
| 4.12. Calcolo del tempo residuo al raggiungimento della soglia . . . . .  | 50 |
| 4.13. Polinomio di terzo grado che approssima l'andamento del modulo dell'errore in una finestra in cui sono presenti oscillazioni . . . . .  | 52 |
| 4.14. Stima del tempo residuo al raggiungimento della soglia per il dataset BoT 150234 . . . . .  | 53 |
| 4.15. Stima del tempo residuo al raggiungimento della soglia per il dataset EoT 111115 . . . . .  | 54 |
| 4.16. Scostamenti normalizzati per i dataset BoT . . . . .  | 56 |
| 4.17. Scostamenti normalizzati per i dataset EoT . . . . .  | 57 |
| 5.1. Sfasamento effetti temperatura . . . . .   | 61 |

## Elenco delle tabelle

|  |    |
|--|----|
| 3.1. Parametri stack . . . . .                                 | 23 |
| 3.2. Parametri produzione H <sub>2</sub> . . . . .             | 23 |
| 3.3. Parametri perdite ohmiche . . . . .                       | 24 |
| 3.4. Parametri perdite di attivazione . . . . .                | 24 |
| 3.5. Calibrazione manuale . . . . .                            | 27 |
| 3.6. Prima calibrazione PSO . . . . .                          | 29 |
| 3.7. Suddivisione della calibrazione in 3 fasi . . . . .       | 33 |
| 3.8. Risultati calibrazione in 3 fasi . . . . .                | 33 |
| 4.1. Ripartizione delle Tipologie 1÷7 nelle 4 classi . . . . . | 47 |
| 4.2. Scostamento normalizzato dei 5 parametri . . . . .        | 57 |



# Capitolo 1.

## Introduzione

Negli ultimi anni sta crescendo sempre di più la necessità di ottenere energia da fonti alternative ai classici combustibili fossili. Una di queste alternative è rappresentata dall'idrogeno, il quale può essere in seguito impiegato direttamente come combustibile o stoccato per la successiva produzione di energia elettrica. L'idrogeno infatti è da diversi anni utilizzato anche come combustibile in applicazioni automobilistiche e solo più recentemente inizia ad essere impiegato anche come risorsa per la successiva produzione di energia elettrica.

Per la riconversione dell'idrogeno in elettricità sono utilizzate le celle a combustibile (o fuel cells), che dall'immissione di idrogeno e ossigeno sono in grado di produrre energia elettrica e uno scarto d'acqua. Considerando che l'idrogeno non è un gas presente in natura come molecola pura, si rende necessaria la sua produzione, che fra le varie metodologie include l'utilizzo degli elettrolizzatori. Gli elettrolizzatori, in contrapposizione alle celle a combustibile partendo da una fonte di energia elettrica e dell'acqua (a volte abbinata ad altre sostanze), attraverso l'elettrolisi producono idrogeno e ossigeno.

Sia elettrolizzatori che fuel cells, sono soggetti nel corso della loro vita a diverse tipologie di usura, che oltre a diminuirne l'efficienza ne possono aumentare la pericolosità [2]. A questo proposito si sono sviluppate delle tecniche volte alla diagnosi e alla prognosi dei guasti.

L'obiettivo di questo studio è stato quello di sviluppare un modello matematico dal quale fosse possibile procedere con una diagnosi dei guasti o avere un'indicazione dello stato di degradazione di un elettrolizzatore. Nello specifico si è sviluppato il modello matematico di un elettrolizzatore dotato di elettrolita a membrana polimerica, che a seguito di una fase di tuning dei parametri, fosse in grado di allinearsi con il comportamento degli elettrolizzatori appartenenti alla stessa tipologia. Si è poi considerato un elettrolizzatore presente presso il gruppo Loccioni e si è allineato il modello a questo componente, dall'errore misurato fra il modello e le misurazioni sull'elettrolizzatore, si è cercato di estrarre degli indici utilizzabili per la stima della degradazione o per la previsione di un guasto.

Nel Capitolo 2 si analizzano elettrolizzatori e fuel cells, evidenziandone le differenze e dandone una panoramica delle loro applicazioni. Successivamente si approfondisce

## *Capitolo 1. Introduzione*

l'elettrolizzatore a membrana polimerica (PEM), mostrandone gli aspetti tecnici e ne segue la modellazione matematica.

Il Capitolo 3 inizia con la descrizione del dataset a disposizione e segue con l'implementazione del modello su Simulink, dividendola nelle diverse componenti. Successivamente è riportato come si sia implementata una calibrazione dei parametri attraverso l'algoritmo di ottimizzazione a sciame di particelle, al fine di allineare il modello alle misurazioni.

Il Capitolo 4 introduce le tipologie di degradazione a cui va incontro un elettrolizzatore e alcune delle tecniche di prognosi presenti in letteratura. A seguire sono presenti degli utilizzi fatti del modello implementato, per ottenere degli indici dai quali distinguere una tendenza al guasto o una degradazione del componente. Come conclusione del capitolo sono riportati dei possibili sviluppi futuri

Nell'Appendice A sono presenti i codici degli script e delle funzioni Matlab utilizzate durante le analisi del modello.

## Capitolo 2.

# Elettrolizzatori e celle a combustibile

### 2.1. Differenze, tipologie ed utilizzi

Come introdotto, lo scopo degli elettrolizzatori è inverso a quello delle celle a combustibile, ma il loro funzionamento è basato sullo stesso fenomeno chimico, tant'è che alcune tipologie di dispositivi permettono di essere utilizzati in entrambe le modalità. Il principio sul quale si basano questi componenti è l'elettrolisi: nel caso dell'elettrolizzatore viene trasformata l'energia elettrica in energia chimica per scindere le molecole di acqua ed ottenere molecole di idrogeno e di ossigeno, mentre in una fuel cell sono gli spostamenti delle cariche che avvengono durante la reazione chimica a generare la corrente elettrica.

Le celle a combustibile possono essere distinte fra di loro in base all'elettrolita utilizzato, che ne determina poi i range di temperature, pressioni e potenze ai quali possono lavorare. Secondo questa suddivisione si possono trovare le celle alcaline, a membrana polimerica, ad acido fosforico, a carbonati fusi e ad ossido solido [3].

Come per le fuel cells, anche le diverse tipologie di elettrolizzatori sono discriminate dalla tipologia di elettrolita, che ne determina le condizioni operative del dispositivo e quali sono le molecole addette al trasporto delle cariche. Fra le più comuni tipologie di elettrolizzatori si possono trovare: elettrolizzatori alcalini, a membrana polimerica, a membrana anionica e ad ossidi solidi (o ad alta temperatura) [1].

Fra gli utilizzi delle fuel cells è presente il settore automobilistico con la generazione di elettricità a bordo dei veicoli [4] ed ultimamente anche il settore aerospaziale si sta avvicinando al mondo dell'idrogeno [5]. Le fuel cell, a differenza delle batterie, richiedono una continua fornitura dei reagenti, sono di fatto dei convertitori di energia, questo le avvantaggia dal punto di vista dei tempi di ricarica. Un altro campo di utilizzo è quello della produzione di elettricità per l'uso domestico, in De Giorgi [6] viene illustrato un esempio applicativo nel quale sfruttando l'idrogeno prodotto dai gas di scarico di altri processi presenti in una palazzina, si possa generare dell'energia elettrica attraverso una fuel cell.

La produzione di idrogeno si diversifica in base alla produzione di CO<sub>2</sub> che ne consegue, attualmente la maggior parte avviene a partire da fonti non rinnovabili, il cui prodotto finale può essere distinto in: idrogeno nero, marrone, grigio o blu, a seconda della risorsa utilizzata e la conseguente CO<sub>2</sub> rilasciata. Gli elettrolizzatori

rappresentano la tecnologia dalla quale è possibile la produzione di idrogeno a partire dall'acqua, se questo procedimento avviene tramite l'utilizzo di fonti di energia rinnovabili, allora quello che si ottiene è denominato idrogeno verde, che vanta una delle più basse emissioni di anidride carbonica fra tutte le altre tipologie [1]. Questa tecnologia, paragonata alle altre, ha dei costi più elevati e per questo risulta la meno utilizzata, ma con un'ottica rivolta al minor impatto ambientale si sta continuando la ricerca e lo sviluppo su questo campo<sup>1</sup>.

## **2.2. Elettrolizzatori PEM**

### **2.2.1. Composizione**

La struttura comune ai vari tipi di elettrolizzatori è quella dello stack, ossia una serie di celle, ognuna costituita da quattro principali componenti: due elettrodi indicati come anodo e catodo, l'elettrolita che li separa e l'alimentazione esterna in corrente continua. L'elettrolizzatore che si prende in considerazione in questo documento è un elettrolizzatore a membrana polimerica, anche detto elettrolizzatore PEM (Proton Exchange Membrane). Come il nome può far intuire, l'elettrolita di questa tipologia di elettrolizzatori è una membrana polimerica che consente il passaggio degli ioni fra anodo e catodo, il materiale più comunemente utilizzato per sua costruzione è il Nafion, ma ce ne sono diversi altri in commercio o in via di sviluppo, allo scopo di ridurre gli elevati costi e le problematiche che il Nafion porta con sé [8]. Gli elettrodi, sulle quali superfici avvengono le reazioni, sono realizzati in materiali metallici, come possono essere titanio o acciaio inossidabile, spesso rivestiti per resistere ai fenomeni corrosivi [9].

### **Condizioni operative**

Gli elettrolizzatori PEM più diffusi prevedono un intervallo di temperature di lavoro di  $30 \div 80^\circ\text{C}$ , mentre le pressioni, grazie alla membrana solida, possono arrivare a circa 40 bar. Avendo un elettrolita solido che consente di lavorare a pressioni più elevate delle altre tipologie di elettrolizzatori, è possibile avere in uscita idrogeno pressurizzato anche fino a 70 bar, come riportato su Medina [10]. Gli elettrolizzatori PEM vantano anche un tempo di vita superiore alle altre tipologie, raggiungendo le  $60.000 \div 80.000$  ore. La purezza dell'idrogeno prodotto è molto elevata, ma può subire alterazioni in caso di pressioni elevate per le quali si verificano fenomeni di permeazione attraverso la membrana.

---

<sup>1</sup>In Incer-Valverde [7] è presente un'interessante panoramica sulla produzione dell'idrogeno nel corso degli ultimi anni.

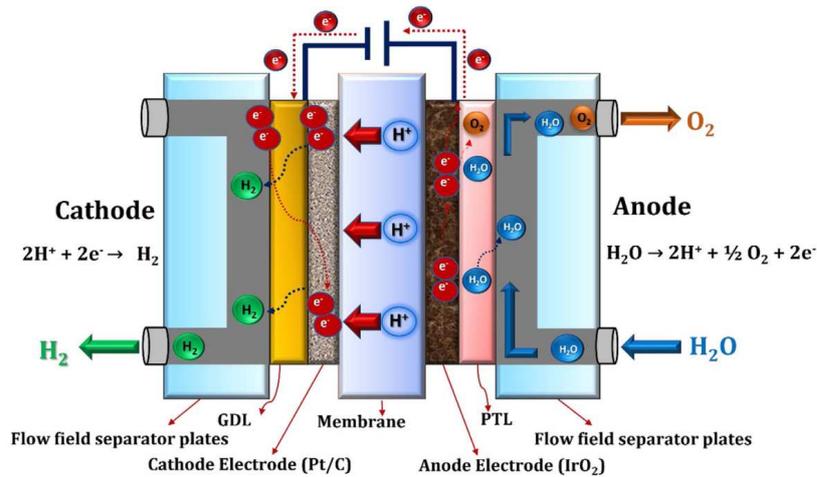
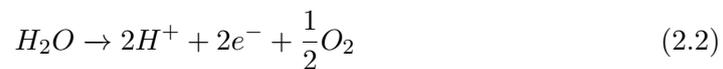


Figura 2.1.: Schema funzionale di un elettrolizzatore PEM [1]

### Funzionamento

La reazione complessiva che avviene può essere descritta dall'Eq.2.1, ed è frutto della somma della reazione anodica con quella catodica (Eq.2.2 e Eq.2.3). Dal lato dell'anodo, sulla superficie dell'elettrodo, grazie all'energia fornita dall'alimentazione esterna, le molecole di acqua ( $H_2O$ ) in ingresso vengono separate, liberando gli idroni ( $H^+$ ) e gli anioni ( $e^-$ ) dagli atomi di ossigeno. Gli atomi di ossigeno liberati dall'idrogeno si ricombinano fra loro formando le molecole di ossigeno ( $O_2$ ), che sono portate fuori dall'elettrolizzatore attraverso il flusso d'acqua che scorre dal lato dell'anodo. Le cariche positive giungono al catodo attraverso la membrana, mentre le cariche negative attraverso il circuito elettrico, a questo punto sulla superficie del catodo le cariche si ricombinano formando le molecole di idrogeno ( $H_2$ ). Dal punto di vista globale per ogni due molecole di acqua in ingresso all'anodo si ottengono due molecole di idrogeno e una di ossigeno. L'intero processo è schematizzato in Figura 2.1<sup>2</sup>.



<sup>2</sup>Nella figura è rappresentata l'acqua in ingresso all'anodo, infatti è da lì che inizia la reazione e l'elettrolizzatore considerato presenta la stessa struttura, tuttavia è bene sapere che sono presenti elettrolizzatori PEM che prevedono l'ingresso di acqua anche da lato del catodo. L'acqua dal lato del catodo può entrare sotto forma di vapore per agire sull'efficienza [11], o come riportato in Tugirumubano [12] entrare dal catodo per poi reagire sempre dal lato dell'anodo

### 2.2.2. Modellazione matematica

L'elettrolizzatore PEM del quale si è costruito il modello matematico è inserito in un banco nel quale a fronte di diversi ingressi in corrente si vogliono osservare gli effetti sulle altre grandezze. Si ha un ingresso in corrente e non in tensione, perché come è visibile in Eq.2.6 la produzione di idrogeno è direttamente proporzionale ad essa, ed essendo gli elettrolizzatori dispositivi utilizzati allo scopo di produrre questo gas la maggior parte dei dispositivi è controllata proprio con un riferimento di corrente. Da questo ne deriva che anche la tensione sarà una conseguenza della corrente in ingresso, osservare il suo comportamento è di fondamentale importanza, in quanto va a impattare sull'efficienza del dispositivo.

#### Produzione $H_2$ e $O_2$

La quantità dei gas prodotti è direttamente proporzionale alla corrente ( $I$ ), in quanto essa rappresenta la quantità di carica positiva che attraversa la membrana e la quantità di carica negativa che percorre il circuito elettrico. Più precisamente, presa la legge di Faraday (Eq.2.4), in cui  $Q$  è la quantità totale di carica in Coulomb (C),  $n$  sono il numero di moli,  $z$  il numero di cariche elementari scambiate e  $F=96485.3365$  C la costante di Faraday, è possibile ricavare che per ogni due moli di acqua la reazione scambia una carica pari a  $4F$  Coulomb e per il bilanciamento dell'Eq.2.1 saranno prodotte una mole di ossigeno e due di idrogeno. Invertendo la proporzione si ottiene la produzione dei due gas espressa in moli al secondo, queste grandezze vengono successivamente moltiplicate per il numero di celle presenti nell'elettrolizzatore ( $N_c$ ) e per l'efficienza di Faraday  $\eta_F$  ottenendo le equazioni Eq.2.5 e Eq.2.6 [13, 14].  $\eta_F$  potenzialmente può assumere valori da 0 a 1, ma solitamente è compreso fra 0.9 e 1.

$$Q = nzF \quad (2.4)$$

$$O_{2,g} = \frac{N_c I}{4F} \eta_F \quad (2.5)$$

$$H_{2,g} = \frac{N_c I}{2F} \eta_F \quad (2.6)$$

#### Tensione a circuito aperto (OCV)

La reazione di elettrolisi non è una reazione spontanea, infatti necessita di un'energia pari a 236.483 kJ/mol che viene richiesta alla sorgente di alimentazione esterna. Alle condizioni di riferimento, ossia pressione atmosferica e temperatura 25°C, questa energia dalla teoria può essere generata da un potenziale di 1.229V, anche se sperimentalmente si è verificato che questo valore si può alzare fino a raggiungere 1.48V. Questa grandezza è denominata tensione reversibile e viene indicata con  $E_{rev}^0$ . Sono presenti diverse formule ottenute empiricamente che ne modellano la dipendenza

dalla temperatura: Eq.2.7 ÷ Eq.2.9.

$$E_{rev}^0 = 1.229 - 0.9 \cdot 10^{-3}(T - 298) \quad (2.7)$$

$$E_{rev}^0 = 1.5184 - 1.5421 \cdot 10^{-3}T + 9.523 \cdot 10^{-5}T \ln(T) + 9.84 \cdot 10^{-8}T^2 \quad (2.8)$$

$$E_{rev}^0 = 1.5241 - 1.2261 \cdot 10^{-3}T + 1.8558 \cdot 10^{-5}T \ln(T) + 5.6692 \cdot 10^{-7}T^2 \quad (2.9)$$

La tensione di circuito a aperto od OCV (Open Circuit Voltage) riportata in Eq.2.10, si ottiene utilizzando l'equazione di Nernst e indica la tensione alla quale lavorerebbe la cella in assenza delle perdite. Le  $p$  rappresentano le pressioni parziali delle sostanze indicate al pedice, mentre  $R = 8.31446 \text{ J/molK}$  è la costante universale dei gas.

$$E_{Cell} = E_{rev}^0 + \frac{RT}{2F} \ln \left( \frac{p_{H_2} p_{O_2}^{0.5}}{p_{H_2O}} \right) \quad (2.10)$$

### Perdite ohmiche

Le perdite a carattere resistivo della cella si possono dividere fra le tre componenti: la membrana e i due elettrodi (Eq.2.11). La sovratensione può essere calcolata attraverso la legge di Ohm riportata in Eq.2.12, nella quale  $\sigma$  rappresenta la conducibilità espressa in S/m,  $\delta$  lo spessore del conduttore,  $i$  la densità di corrente espressa in A/m<sup>2</sup> e l'indice  $k$  distingue membrana (*mem*), anodo (*an*) e catodo (*cat*). Le perdite degli elettrodi sono solitamente trascurabili se messe a confronto con quella della membrana.

$$V_{\Omega} = V_{\Omega,mem} + V_{\Omega,an} + V_{\Omega,cat} \quad (2.11)$$

$$V_{\Omega,k} = \frac{\delta_k}{\sigma_k} i \quad (2.12)$$

Approfondendo, nel caso del Nafion la conducibilità della membrana può essere espressa secondo l'Eq.2.13, nella quale la si esprime in funzione della temperatura e di un termine  $\lambda$ , ovvero un indice di umidificazione della membrana [15]. Questo valore può essere calcolato a sua volta dall'attività dell'acqua all'anodo  $a_a$ , tramite l'Eq.2.14 [16].

$$\sigma_{mem} = (0.5139\lambda - 0.326) \exp \left[ 1268 \left( \frac{1}{303} - \frac{1}{T} \right) \right] \quad (2.13)$$

$$\begin{aligned} \lambda &= 0.043 + 17.81a_a - 39.85a_a^2 + 36a_a^3; \quad 0 < a_a \leq 1 \\ \lambda &= 14 + 1.4(a_a - 1); \quad 1 < a_a \leq 3 \end{aligned} \quad (2.14)$$

### Perdite di attivazione

Per far sì che la reazione avvenga e le molecole di acqua siano scisse, è necessario oltrepassare una barriera energetica. Le perdite di attivazione sono le sovratensioni

necessarie affinché si verifichi questo salto. La modellazione più comune è derivata dall'equazione di Butler-Volmer ed è riportata in Eq.2.15.  $\alpha_a$  e  $\alpha_c$  rappresentano i coefficienti di trasferimento di carica,  $i_{0,a}$  e  $i_{0,c}$  le densità di correnti di scambio<sup>3</sup>. Questi parametri in letteratura sono riportati in diverse combinazioni, in Falcão [13] è presente una panoramica con alcuni esempi.

$$V_{Act} = \frac{RT_a}{\alpha_a F} \sinh^{-1} \left( \frac{i}{2i_{0,a}} \right) + \frac{RT_c}{\alpha_c F} \sinh^{-1} \left( \frac{i}{2i_{0,c}} \right) \quad (2.15)$$

In letteratura è possibile trovare alcune modellazioni per il calcolo delle densità di correnti di scambio, su García-Valverde [18] tramite l'applicazione di un'equazione di Arrhenius è proposta l'Eq.2.16, nella quale  $i_{0,ref}$  rappresenta la densità di corrente di scambio alla temperatura di riferimento  $T_{ref}$  e  $E_{act}$  è l'energia di attivazione, ossia l'energia da superare per innescare la reazione.

$$i_0 = i_{0,ref} \exp \left[ -\frac{E_{act}}{R} \left( \frac{1}{T} - \frac{1}{T_{ref}} \right) \right] \quad (2.16)$$

### Perdite di concentrazione

Le perdite di concentrazione, o di diffusione, sono delle sovratensioni che si generano a causa del rallentamento dovuto all'aumentare dei flussi all'interno dell'elettrolizzatore. Questo tipo di perdite si verificano solo per densità di correnti elevate, in linea generale, per gli elettrolizzatori PEM è riportato che per densità di correnti inferiori a 1.6A/cm<sup>2</sup> possono essere trascurate [13]. Una formulazione per calcolare questa sovra tensione è riportata in Eq.2.17, nella quale si considerano il rapporto fra le concentrazioni  $C_{O_2}$  e  $C_{H_2}$  dei due gas all'interfaccia elettrodo-membrana e la loro concentrazione in una condizione considerata di riferimento  $C_{0,O_2}$  e  $C_{0,H_2}$  [19, 20].

$$V_{Diff} = \frac{RT}{2F} \ln \left( \frac{C_{O_2}}{C_{0,O_2}} \right) + \frac{RT}{4F} \ln \left( \frac{C_{H_2}}{C_{0,H_2}} \right) \quad (2.17)$$

Questa modellazione richiede una conoscenza specifica delle caratteristiche costruttive dell'elettrolizzatore, per cui si può ricorrere ad un'alternativa più semplice (Eq.2.18) che utilizza come parametro  $i_{lim}$ , ossia la massima densità di corrente applicabile all'elettrolizzatore [13, 21].

$$V_{Diff} = -0.75 \frac{RT}{F} \ln \left( 1 - \frac{i}{i_{lim}} \right) \quad (2.18)$$

---

<sup>3</sup>Le correnti di scambio rappresentano la velocità di scambio delle cariche degli elettrodi nella condizione di equilibrio, idealmente all'equilibrio sono uguali e di segno opposto e fanno in modo che la corrente netta dell'elettrolizzatore sia nulla. I coefficienti di trasferimento di carica fungono come fattore di simmetria [17].

### Somma dei contributi

Una volta ottenuti i vari contributi di tensione possono essere sommati e moltiplicati per il numero di celle presenti nell'elettrolizzatore per ottenere la tensione totale dello stack:

$$V = N_c(E_{Cell} + V_{\Omega} + V_{Act} + V_{Diff}) \quad (2.19)$$

### Efficienza di un elettrolizzatore

Per determinare l'efficienza di un elettrolizzatore, si debbono dichiarare quali sono le potenze da considerare. Per quanto riguarda la potenza di ingresso è chiaro che questa sia rappresentata dalla potenza elettrica  $P_e$  (Eq.2.21), mentre per la potenza di uscita si considera quella che verrebbe prodotta dalla combustione dell'idrogeno. Questo valore si ottiene moltiplicando la quantità di idrogeno prodotta al secondo per il suo potere calorifico, di cui solitamente si considera il valore superiore ( $HHV = 141.8 MJ/Kg$ ), ma in altri casi si utilizza quello inferiore ( $LHV = 120 MJ/Kg$ ). Il potere calorifico superiore rispetto a quello inferiore tiene conto anche del calore di vaporizzazione dell'acqua, ossia dell'energia spesa durante la combustione per far passare l'acqua dallo stato liquido a quello gassoso.

L'efficienza di un elettrolizzatore quindi è data dal rapporto di queste due potenze (Eq.2.20)<sup>4</sup> [22, 23, 24]:

$$\eta = \frac{H_{2,g}HHV}{P_e} \quad (2.20)$$

$$P_e = VI \quad (2.21)$$

---

<sup>4</sup>La produzione di idrogeno in 2.2.2 è calcolata in mol/s, per il calcolo dell'efficienza è necessario convertirla in Kg/s)



## Capitolo 3.

# Implementazione del modello su Simulink

### 3.1. Dataset a disposizione

L'elettrolizzatore su cui si è basato lo studio è inserito all'interno di un banco che acquisisce diverse misure su tutto l'impianto circostante, ad essere prese in considerazione sono state le seguenti:

- Corrente [A]
- Tensione dello stack [V]
- Pressione di uscita dell'anodo [Bar]
- Pressione di uscita del catodo [Bar]
- Temperatura di entrata dell'anodo [°C]
- Temperatura di entrata del catodo [°C]
- Flusso di ossigeno [Nl/min]
- Flusso di idrogeno [Nl/min]

Si avevano a disposizione delle misurazioni effettuate nell'arco temporale che va da fine ottobre 2022 a inizio aprile 2023, tutti i dati sono stati campionati ad una frequenza di 4Hz.

### 3.2. Simulink

La vista generale del modello è mostrata in Figura 3.1, il sistema è diviso in più blocchi per gestire i diversi aspetti. All'interno di alcuni sottosistemi, è possibile andare a selezionare quale fra le equazioni implementate si vuole utilizzare. Lo scopo del modello, una volta ricevuti gli ingressi provenienti da un dataset, è quello di confrontare i suoi risultati con quelli presenti nel dataset analizzato. Per eseguire il modello è necessario caricare i dati dal workspace tramite uno script, dal quale si specifica il codice del dataset da considerare e si impostano i valori dei parametri con i quali si desidera lavorare<sup>1</sup>.

<sup>1</sup>Tutti i parametri possono essere specificati direttamente dal modello, ma alcuni si è preferito collegarli a delle variabili del workspace

## Capitolo 3. Implementazione del modello su Simulink

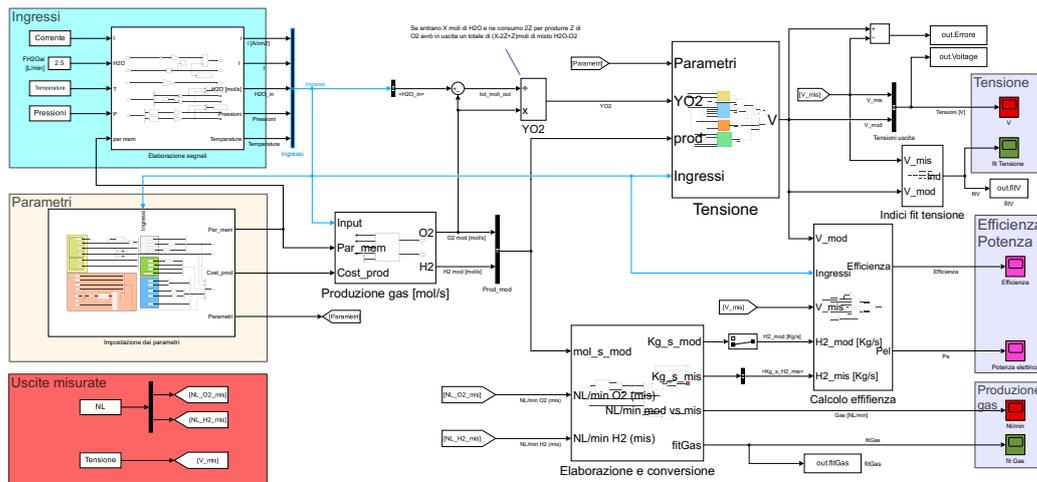


Figura 3.1.: Vista complessiva del modello Simulink

---

```

1  %% CARICAMENTO DATASET E PARAMETRI %%

3  % Caricamento dati da file.xlsx
   DataLog='121203';
5  DataSet=['DataLog_' DataLog '.xlsx'];
   [time, Temperature,NL,Pressioni,Tensione,Corrente] = ...
7                                     ...EstraiDati(DataSet);

9  % Parametri tensione
   alpha_a=2.5;
11  alpha_c=0.5698;
   i0a=2.3301*1e-4;
13  i0c=0.8753;
   lambda=8.9354;

15

17  % Parametri gas
   H2_Gain=0.9661;
   H2_Tau=1.4885;
19  H2_Offset=0.8797;

```

---

### 3.2.1. Ingressi e uscite

Il sottosistema Elaborazione segnali in Figura 3.2 elabora e converte i segnali riferendoli al sistema internazionale della misura.

Della corrente viene effettuato il modulo per eliminare le fluttuazioni negative dovute dal rumore quando la corrente è nulla, viene poi divisa per l'area della membrana in  $\text{cm}^2$  (unica grandezza non espressa nel SI) per ottenere la densità di corrente.

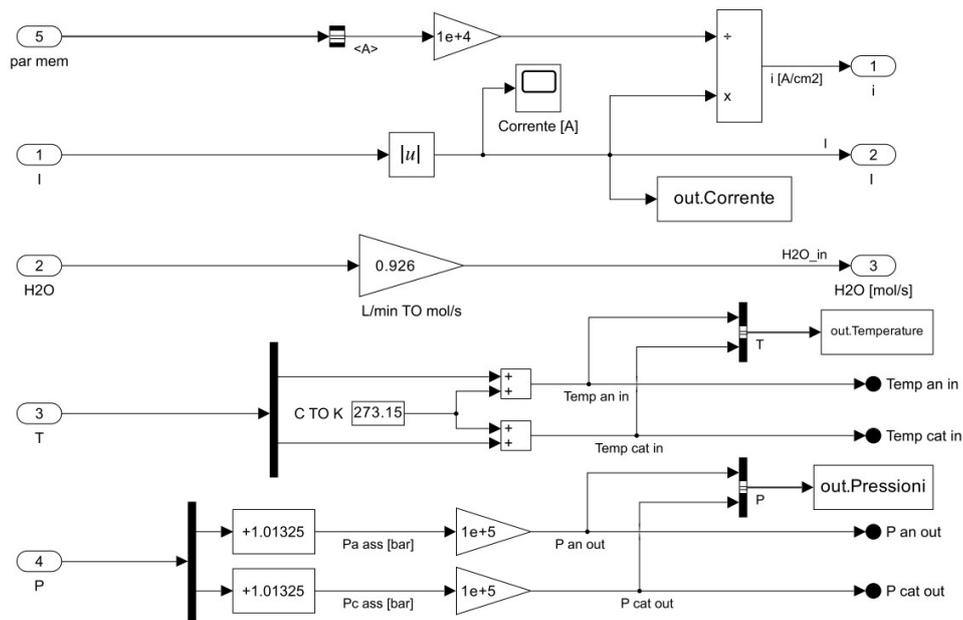


Figura 3.2.: Elaborazione dei segnali

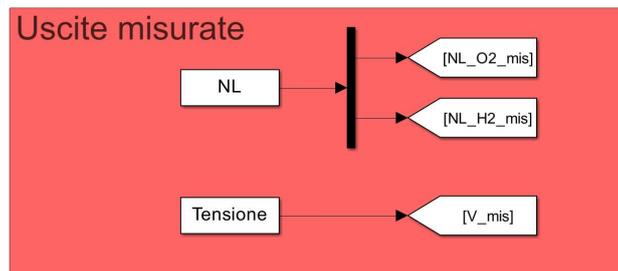


Figura 3.3.: Caricamento dal workspace della tensione e della produzione dei gas contenute nel dataset

Il flusso di acqua in L/min viene moltiplicato per il numero di moli di  $H_2O$  presenti in un litro (55.56 mol/l) e divisi per 60 per ottenere il valore in moli al secondo ( $55.56/60 = 0.926$ ).

Le temperature vengono espresse in K sommando 273.15, mentre alle pressioni viene aggiunto l'equivalente della pressione atmosferica in bar e il risultato è trasformato in Pascal. L'aggiunta della pressione atmosferica serve per passare dalle pressioni relative rilevate dei sensori alle pressioni assolute.

Come uscite si prendono in considerazione la tensione complessiva dello stack e la produzione dei due gas, questi valori vengono caricati dal workspace, tramite lo script citato in precedenza e non necessitano di conversioni (Figura 3.3).

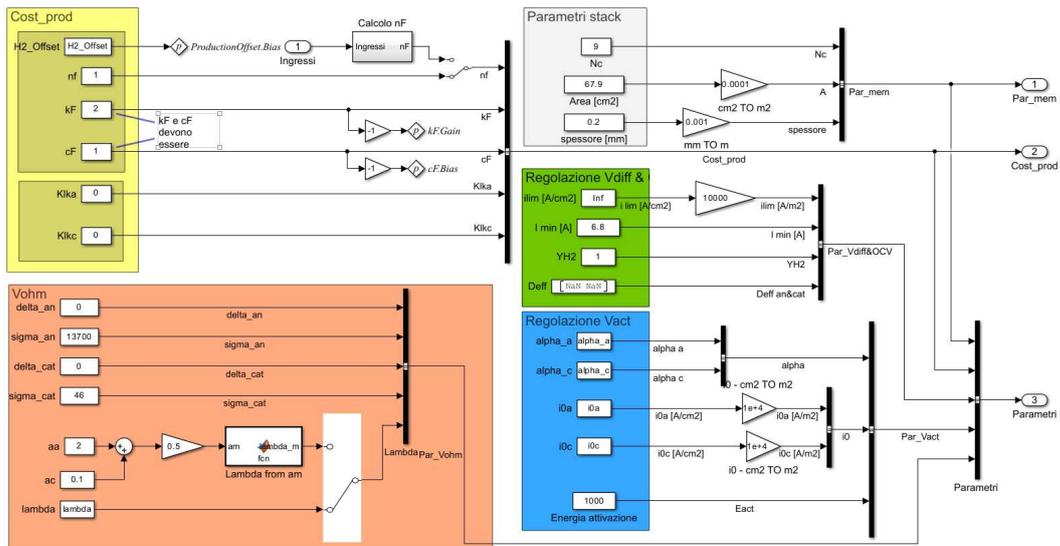


Figura 3.4.: Sottosistema per l'inserimento dei parametri

### 3.2.2. Parametri

Il blocco Impostazione dei parametri (Figura 3.4) al suo interno è diviso in più aree, ognuna di esse comprende un gruppo di parametri che va ad influenzare un aspetto del sistema. Non tutti i parametri vengono utilizzati, questo dipende dalle equazioni che vengono selezionate nel resto del modello.

In alto a sinistra, riquadrati in giallo, sono presenti i parametri che agiscono sulla produzione di idrogeno e ossigeno, il loro effetto è spiegato nella sezione 3.2.3. In basso a sinistra si trovano gli spessori e la conducibilità dei due elettrodi e il lambda visto in Eq.2.13. Il coefficiente  $\lambda$  può anche essere calcolato tramite i due coefficienti  $aa$  e  $ac$ , i quali rappresentano l'attività dell'acqua alla membrana dal lato anodico e da quello catodico. In Eq.2.14 si è fatto riferimento solo ad  $a_a$ , ma nel modello Simulink si è preso in considerazione un caso generale, in cui può essere presente dell'acqua anche al catodo, pertanto si effettua la media dell'attività dell'acqua fra anodo e catodo e si calcola  $\lambda$  come indicato nell'Eq.2.14.

In alto a destra  $N_c$ , Area [cm<sup>2</sup>] e spessore [mm] indicano rispettivamente il numero di celle presenti nello stack, l'area e lo spessore della membrana espresse secondo l'unità di misura riportata. Subito sotto sono presenti i parametri che influenzano le perdite per diffusione (ilim, YH2 e Deff) e una corrente minima per l'attivazione dell'OCV  $I_{min}$ .

Come ultimi troviamo i parametri per la regolazione della sovratensione di attivazione, sono presenti i coefficienti di trasporto di carica  $\alpha_a$  e  $\alpha_c$ , le densità di correnti di scambio  $i_{0a}$  e  $i_{0c}$  e l'energia di attivazione della reazione.

Tutti i parametri sono etichettati e raccolti in una struttura di bus per agevolarne l'utilizzo.

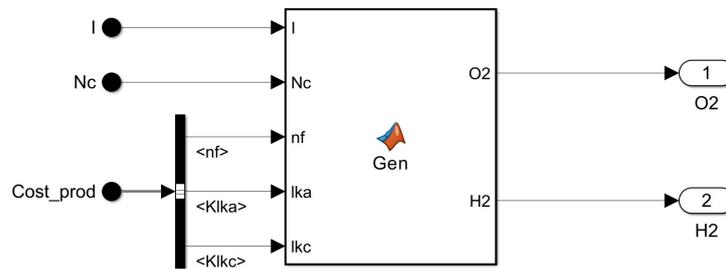


Figura 3.5.: Sottosistema per il calcolo della produzione di idrogeno e ossigeno in mol/s

### 3.2.3. Produzione gas e conversione in NL/min

#### Produzione

Nel blocco Produzione gas [mol/s] (Figura 3.5) si implementano le equazioni 2.5 e 2.6, alle quali sono aggiunte 2 costanti: Klka e Klkc. Queste costanti servono a modellare il leakage, che seppur trascurabile nella maggior parte delle applicazioni, è presente in un elettrolizzatore. Il leakage è il fenomeno che avviene quando uno o entrambi i gas, appena formati attraversano la membrana finendo dal lato opposto. Si verifica soprattutto ad alte pressioni e in presenza di membrane usurate, è un fenomeno pericoloso, in quanto una concentrazione di idrogeno nell'aria a partire dal 4%, può causare esplosioni [25]. Con la costante Klka si modella la percentuale di ossigeno prodotto che penetra la membrana finendo dal lato del catodo, mentre con Klkc la percentuale di idrogeno prodotto, che sempre attraverso la membrana, arriva all'anodo. Entrambi i parametri hanno un range da 0 a 1, ma il loro valore sarà vicino allo 0.

---

```

1 function [O2, H2] = Gen(I,Nc,nf,lka,lkc)
3 % Produzione dei gas
  O2g = Nc*I*nf/(4*96485.3365);
5 H2g = Nc*I*nf/(2*96485.3365);
7 %Fenomeni di leakage
  O2 = O2g*(1-lka) + H2g*lkc;
9 H2 = H2g*(1-lkc) + O2g*lka;

```

---

#### Elaborazione e conversione

La produzione dei gas è calcolata in moli al secondo, ma i sensori presenti nell'impianto riportano una misura in normo-litri al minuto (NL/min), pertanto nel blocco Elaborazione e conversione di Figura 3.6 si effettua questa conversione. La

MatlabFunction Litri\_Normolitri\_minuto (mol\_s\_mod) ricava la produzione dei gas in NL/min e g/min.

---

```
function [g_min, NL_min] = Litri_Normolitri_minuto (mol_s_mod)
2
Nm3_s = [mol_s_mod(1)*8.314*273.15/101325;...
4         mol_s_mod(2)*8.314*273.15/101325];
6
NL_min = Nm3_s*1e+3*60;
8
g_min = [mol_s_mod(1)*32*60; mol_s_mod(2)*2.016*60];
```

---

Si deve tenere in considerazione che i sensori di flusso non sono posizionati direttamente sull'elettrolizzatore, ma a valle di altri processi, come per esempio la fase di de-umidificazione. Infatti l'ossigeno deve essere separato dall'acqua assieme alla quale esce dall'anodo, mentre l'idrogeno porta con se del vapore, dovuto dall'acqua che riesce ad attraversare la membrana (sono quantità trascurabili a livello di calcolo, ma nel sistema reale vanno tenute in considerazione ed eliminate). Questa sequenza fa in modo che, a fronte di una variazione di corrente, i sensori avvertano la variazione della produzione di gas con un certo ritardo. Al fine di modellare questo aspetto si è introdotta una funzione di trasferimento (Figura 3.7a), che utilizza i parametri H2\_Offset (impostato dalla sezione parametri in Figura 3.4), H2\_Tau e H2\_Gain (caricati direttamente dal workspace nel blocco ProductionDynamic, come mostrato in Figura 3.7b)<sup>2</sup>.

Sempre nella sezione parametri in Figura 3.7 fra i parametri dei gas compaiono anche cF e kF, questi vengono utilizzati se si vuole considerare il parametro nF (corrispondente di  $\eta_F$  in Eq.2.5 e Eq.2.6), non come valore costante, ma come funzione della densità di corrente. In Tijani [26] è riportato che l'efficienza di Faraday assume un andamento di tipo esponenziale, nel modello questo aspetto è stato modellato nell'impostazione dei parametri dando la possibilità di selezionare fra il valore costante nF o l'uscita del blocco Calcolo nF, nel quale è inserita l'Eq.3.1 implementata con la sequenza di operazioni mostrata in Figura 3.8.

$$\eta_F = 1 - e^{i \cdot k_F + c_F} \quad (3.1)$$

### 3.2.4. Tensione

Il sottosistema per il calcolo della tensione dello stack effettua la somma dei quattro contributi e moltiplica il risultato per il numero di celle.

---

<sup>2</sup>Si è aggiunta la dinamica solamente per la produzione dell'idrogeno dato che è il gas di maggior interesse in questo processo, per l'ossigeno si dovrebbero considerare parametri differenti, infatti, se l'idrogeno viene semplicemente deumidificato, l'ossigeno viene prima estratto dall'acqua.

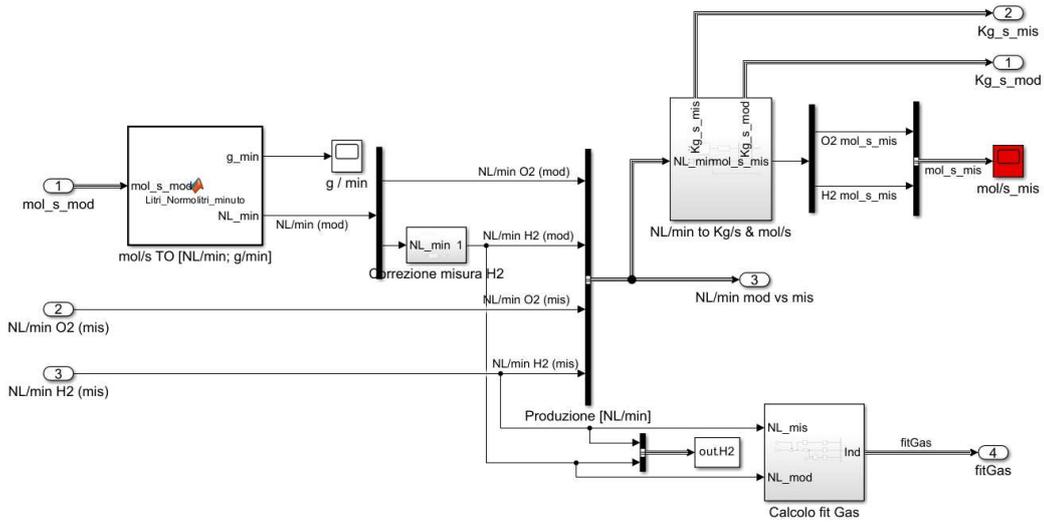
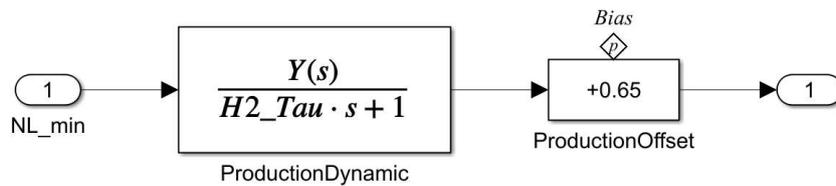
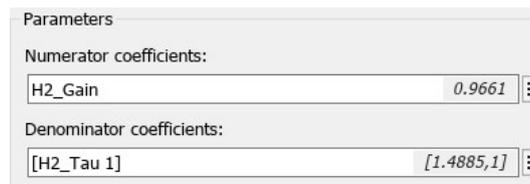


Figura 3.6.: Sottosistema per l'elaborazione dei gas prodotti



(a) Funzione di trasferimento e offset



(b) Parametri funzione di trasferimento

Figura 3.7.: Funzione di trasferimento applicata al flusso dell'idrogeno

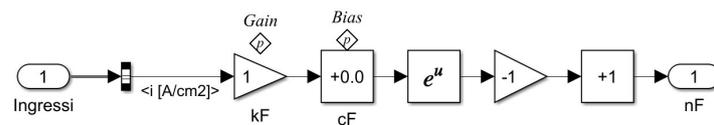


Figura 3.8.: :Calcolo  $\eta_F$  in funzione della densità di corrente

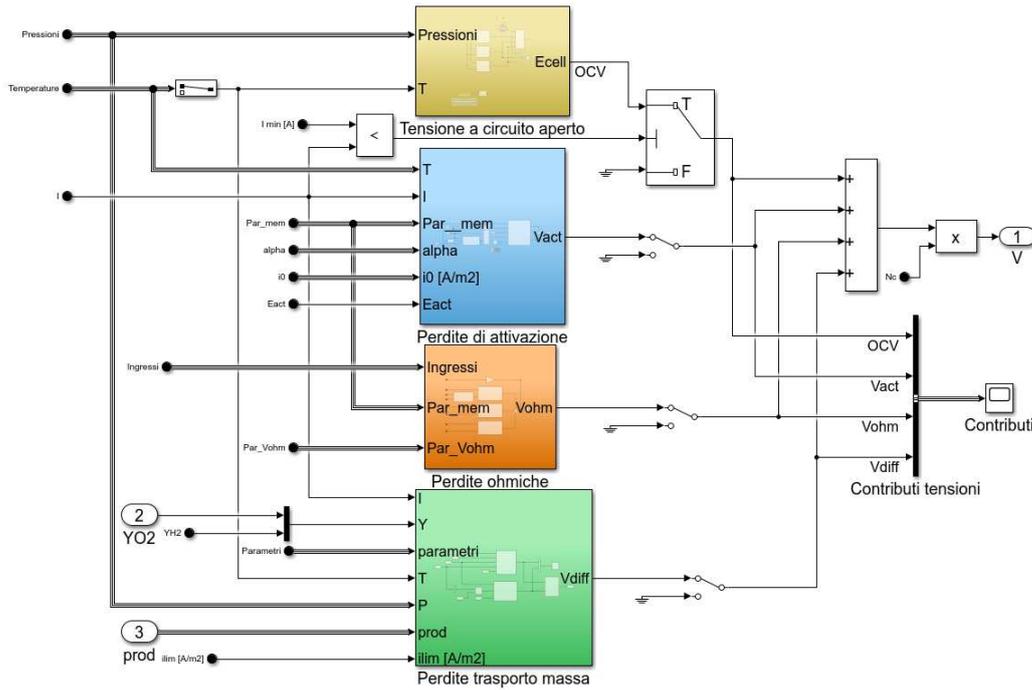


Figura 3.9.: Calcolo della tensione diviso per i quattro contributi

**Tensione a circuito aperto** Nel modello sono riportate tre opzioni per il calcolo della OCV (Figura 3.10), la prima (Figura 3.11) è con l'Eq.2.10, nella quale sono state effettuate delle approssimazioni ottenendo l'Eq.3.2<sup>3</sup>. In questo primo blocco anche il calcolo della tensione reversibile può essere scelto da quattro differenti opzioni (Figura 3.12), la prima implementa l'Eq.2.9, la seconda l'Eq.2.7, la terza l'Eq.2.8 e la quarta considera la  $E_{rev}^0$  costante, il valore della costante può essere scelto fra 1.229V e 1.48V (come trattato in 2.2.2).

$$E_{Cell} = E_{rev}^0 + \frac{RT}{2F} \ln(P_{H_2} P_{H_2O}^{0.5}) \quad (3.2)$$

La seconda e terza opzione per il calcolo della OCV portano a risultati equiparabili: utilizzano entrambe la tensione reversibile calcolata con Eq.2.7 la differenza fra le due è che nella prima si è effettuato un cambio di base del logaritmo (Eq.3.3), mentre nella seconda si inverte l'argomento del logaritmo, che quindi viene sottratto.

$$E_{rev}^0 = 1.229 - 0.9 \cdot 10^{-3}(T - 298) + 2.3 \frac{RT}{4F} \log(P_{H_2}^2 P_{O_2}) \quad (3.3)$$

La OCV a differenza degli altri contributi non è in funzione della corrente, perché non rappresenta una perdita, ma una quantità minima di potenziale per poter innescare la reazione. Se la OCV non viene superata non può quindi esserci corrente, poiché la reazione non ha inizio e non si formano le cariche che attraversano il circuito elettrico.

<sup>3</sup>Avendo l'acqua allo stato liquido il denominatore si semplifica e la pressione parziale dell'ossigeno è approssimata a quella dell'acqua presente nell'anodo.

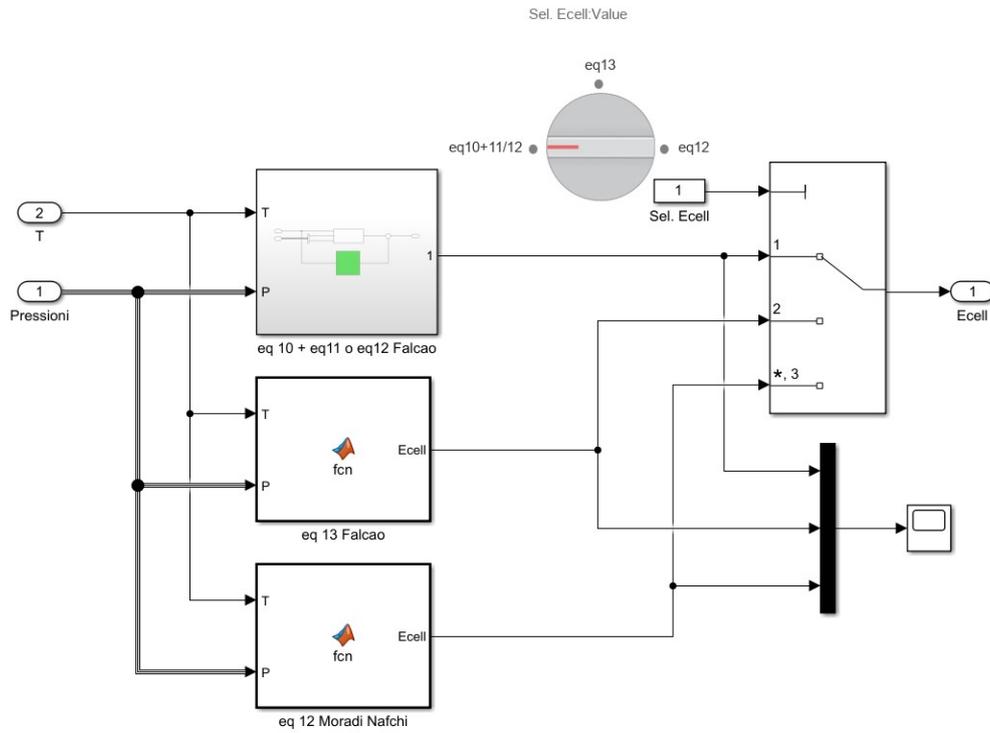


Figura 3.10.: Opzioni per il calcolo della tensione a circuito aperto

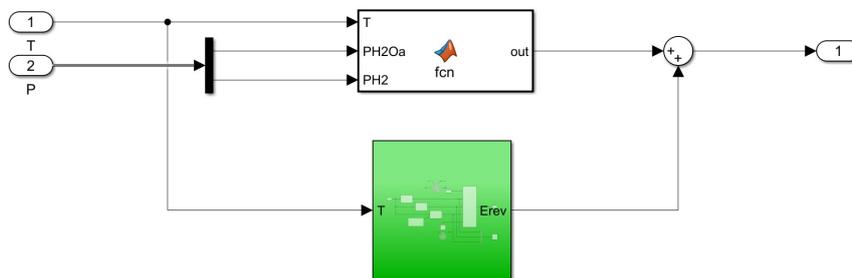


Figura 3.11.: Prima opzione per il calcolo della tensione a circuito aperto con selezione del calcolo della tensione reversibile

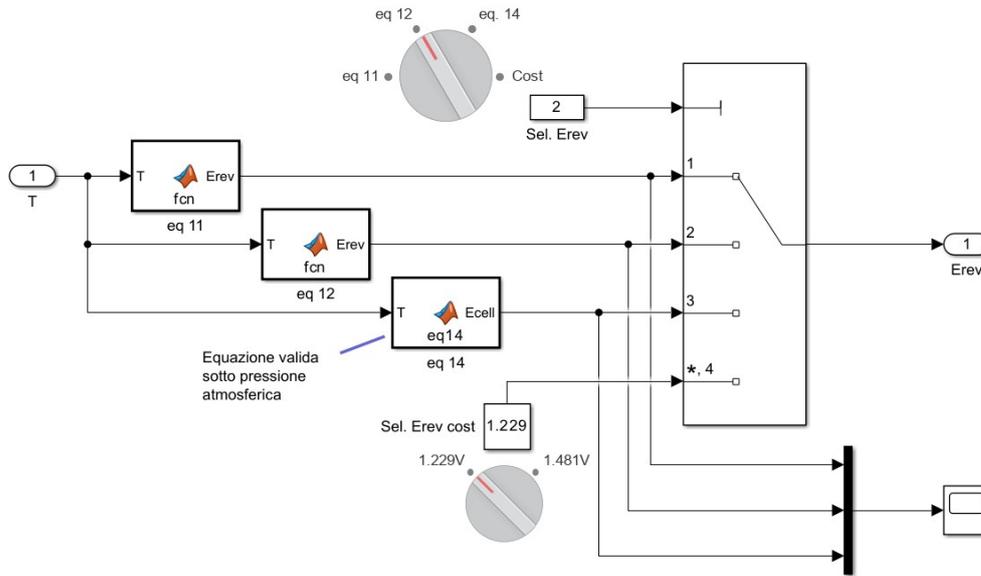


Figura 3.12.: Opzioni per il calcolo della tensione reversibile

Per modellare questo effetto si utilizza un comparatore che annulla l'effetto della OCV per correnti al di sotto di quella minima, impostata attraverso il parametro  $I_{min}$  [A] nella sezione parametri (3.2.2).

**Perdite ohmiche** Le perdite ohmiche vengono calcolate come la somma delle perdite sulla membrana e sugli elettrodi, si calcolano le tre resistività, e alla somma di queste si moltiplica la densità di corrente per ottenere la sovratensione Figura 3.13.

**Sovratensioni di attivazione** Le perdite di attivazione sono calcolate secondo l'Eq.2.15 (Figura 3.14<sup>4</sup>), è possibile scegliere se utilizzare le densità di correnti di scambio come valori costanti impostati dai parametri  $i_{0a}$  e  $i_{0c}$  dalla sezione parametri, o calcolarle con l'equazione di Arrhenius (Eq.2.16). Nel secondo caso i parametri  $i_{0a}$  e  $i_{0c}$  rappresentano le  $i_{0,ref}$  e l'energia di attivazione è data dal parametro **Energia attivazione**, passato in ingresso attraverso il canale **Eact** del bus dei parametri. È stato aggiunto anche un selettore per aggiungere un fattore 2 al denominatore, in quanto in Falcão [13] è riportato che alcuni autori utilizzano questa versione.

**Sovratensione di concentrazione** Le perdite di diffusione presentano due opzioni per il loro calcolo (Figura 3.15), con la prima si utilizza l'Eq.2.17 che considera le concentrazioni dei gas all'interfaccia membrana-elettrodi, con la seconda si utilizza la versione semplificata Eq.2.18 che utilizza la corrente limite dell'elettrolizzatore.

<sup>4</sup>L'output è stato diviso per tensioni di attivazione anodiche e catodiche per osservarne le differenze al variare dei parametri.

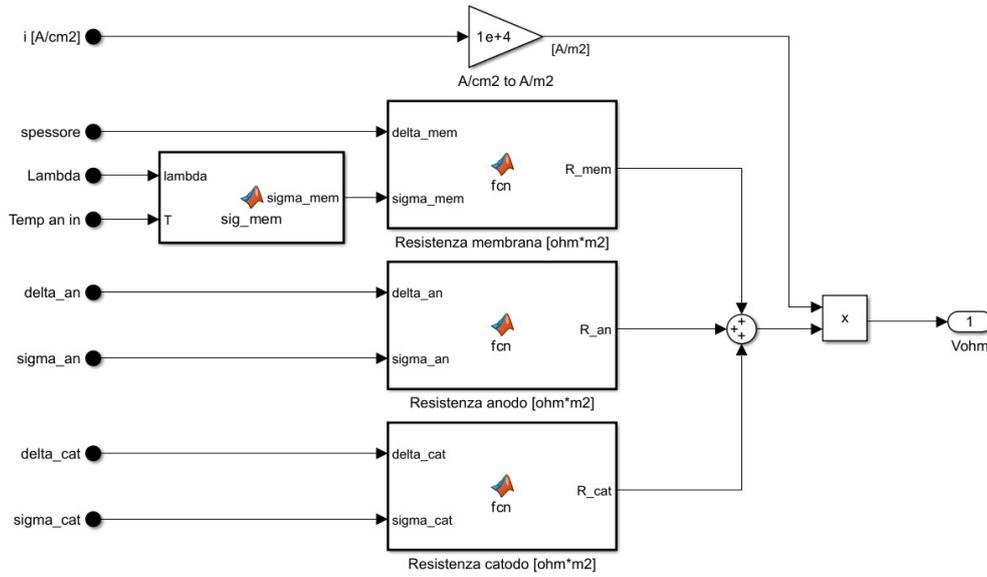


Figura 3.13.: Calcolo delle perdite ohmiche

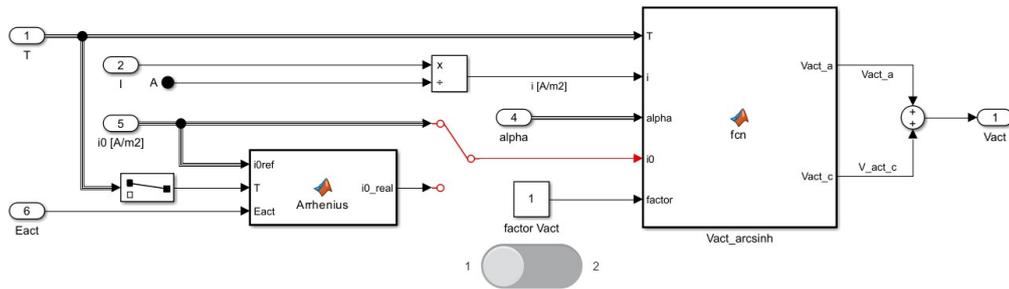


Figura 3.14.: Calcolo perdite di attivazione

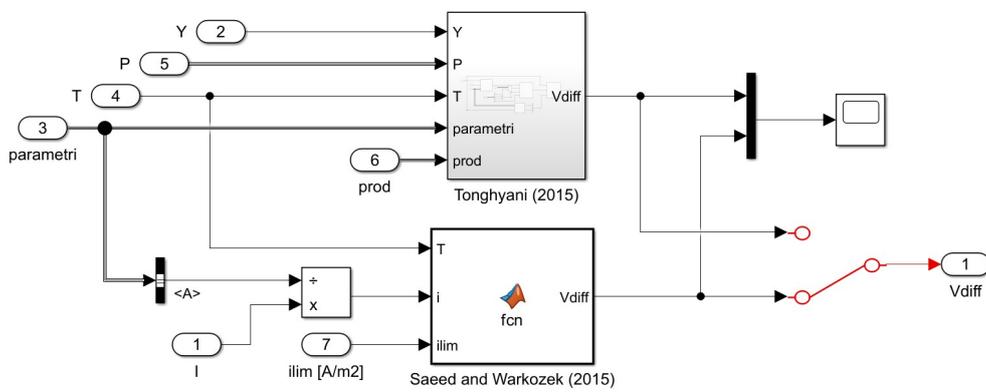


Figura 3.15.: Calcolo delle perdite di concentrazione

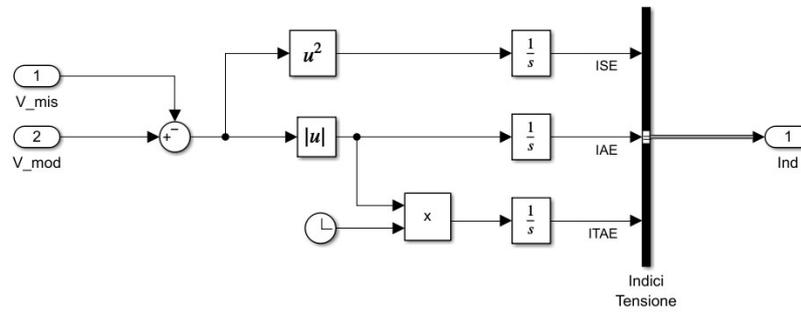


Figura 3.16.: Calcolo degli indici di fit

### Calcolo indici di fitting

Per valutare in modo quantitativo la capacità del modello di approssimare il reale funzionamento dell'elettrolizzatore si sono calcolati, sia per la tensione che per la produzione di idrogeno in  $\text{Nl/min}$ , tre diversi indici:

- ISE: Integral of Square Error
- IAE: Integral of the Absolute value of Error
- ITAE: Integral of the Time-weighted Absolute Error

Il calcolo è stato effettuato con due sottosistemi identici sia per la tensione che per l'idrogeno (Figura 3.16).

### Efficienza

L'efficienza dell'elettrolizzatore è calcolata secondo l'Eq.2.20, sia con le misurazioni presenti nel dataset che con i risultati del modello. Si sono anche considerati entrambi i valori calorifici ( $HHV$  e  $LHV$ ) per osservare le differenze.

## 3.3. Calibrazione dei parametri

### 3.3.1. Prima fase: calibrazione manuale

Fin'ora si è visto come il modello matematico sia stato implementato su Simulink e in quali porzioni vanno ad agire i vari parametri, a questo punto si deve trovare una combinazione di quest'ultimi che consenta al modello di allinearsi alle misurazioni presenti nei dataset. In primo luogo viene fatta una calibrazione manuale per comprendere in quali range possono ricadere i valori dei parametri.

Si parte dai parametri dell'elettrolizzatore che sono presenti sul suo datasheet, ossia un numero di celle pari a 9 e una membrana con un'area di  $67.9 \text{ cm}^2$  e uno spessore di  $0.2 \text{ mm}$ .

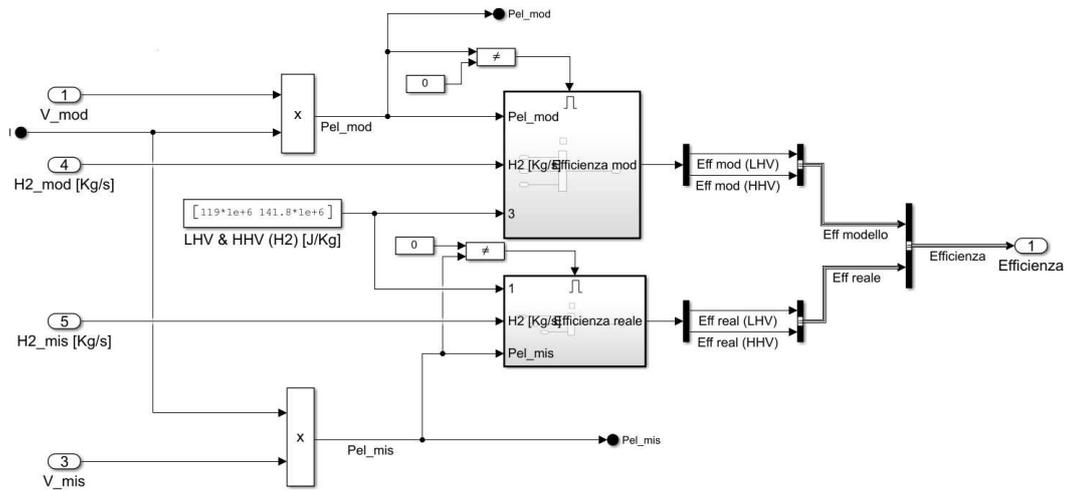


Figura 3.17.: Calcolo dell'efficienza dell'elettrolizzatore

Tabella 3.1.: Parametri stack

|                   |                      |
|-------------------|----------------------|
| Numero di celle   | 9                    |
| Area membrana     | 67.9 cm <sup>2</sup> |
| Spessore membrana | 0.2 mm               |

Passando ai parametri relativi alla produzione dei gas si è deciso: di trascurare i fenomeni di leakage impostando i parametri  $Klka$  e  $Klkc$  a 0, di considerare un  $\eta_F$  costante (annullando l'effetto di  $cF$  e  $kF$ ) e di tenere in considerazione  $H2\_Tau$ ,  $H2\_Gain$  e  $H2\_Offset$  in modo da poter modellare la dinamica introdotta in Figura 3.7. Dato che a moltiplicare la quantità di gas prodotta è già presente il parametro  $H2\_Gain$  della funzione di trasferimento, il parametro  $nF$  viene lasciato a 1. Per il momento si lascia a 1 anche  $H2\_Gain$ , mentre  $H2\_Tau$  e  $H2\_Offset$  sono impostati a 0. La combinazione appena descritta è riportata in Tabella 3.2.

Tabella 3.2.: Parametri produzione H<sub>2</sub>

|                       |                |
|-----------------------|----------------|
| Costante tempo        | 0              |
| Guadagno              | 1              |
| Offset                | 0              |
| Efficienza di Faraday | 1              |
| cF, kF                | Non utilizzati |

Passando all'OCV si debbono selezionare quali equazioni si intende utilizzare, in questo caso si sono scelte l'opzione 1 per la OCV, accompagnata dall'opzione 2 per la tensione reversibile, così facendo sono selezionate l'Eq.2.7 e l'Eq.3.2. La differenza fra le varie implementazioni, al variare di temperature e pressioni, resta comunque inferiore a 10 mV.

Come corrente minima per calcolare la OCV si è impostata una corrente di 6.8A,

in modo che fosse di poco inferiore alla minima utilizzata durante le simulazioni (7.5 A).

Per le perdite ohmiche si sono trascurati i contributi degli elettrodi, in quanto non se ne conoscevano conducibilità e spessore e il loro effetto è trascurabile rispetto a quello della membrana. Per annullare la perdita sugli elettrodi si è impostato il loro spessore a 0. Della membrana invece si è impostato direttamente il  $\lambda$  senza calcolarlo attraverso i parametri di attività dell'acqua  $aa$  e  $ac$ , che comunque non sarebbero stati noti o facilmente ottenibili<sup>5</sup>. Il valore preso in considerazione per  $\lambda$  è 21, un valore plausibile poiché ricade all'interno dei range riportati in letteratura [19].

Tabella 3.3.: Parametri perdite ohmiche

|                    |                |
|--------------------|----------------|
| $\lambda$          | 21             |
| $aa$ e $ac$        | Non utilizzati |
| Spessore elettrodi | 0              |
| Cond. elettrodi    | Non utilizzati |

Per le perdite di attivazione si è deciso di considerare le densità di correnti di scambio come valori costanti, dunque non viene utilizzato il blocco contenente l'equazione di Arrhenius (Eq.2.16) in Figura 3.14. I parametri utilizzati restano quindi i coefficienti di trasferimento di carica e le densità di corrente di scambio, i cui valori vengono presi dalla panoramica presente in Falcão [13].

Tabella 3.4.: Parametri perdite di attivazione

|                         |                |
|-------------------------|----------------|
| $\alpha_a$ e $\alpha_c$ | 0.5            |
| $i_{0,a}$ e $i_{0,c}$   | $10^{-6}$      |
| $E_{act}$               | Non utilizzato |

In ultimo si impostano i parametri per le perdite di concentrazione, in cui si hanno: la densità di corrente limite imposta all'elettrolizzatore  $ilim$  [A/cm<sup>2</sup>], la frazione molare dell'idrogeno dal lato del catodo  $YH_2$ <sup>6</sup> e il coefficiente di diffusione per anodo e catodo  $Deff$ . Di questi parametri si utilizza solamente  $ilim$  [A/cm<sup>2</sup>], poiché si considera l'equazione semplificata per il calcolo della sovratensione di diffusione. Tuttavia, osservando in Figura 3.18 le curve di polarizzazione costruite sui dati forniti dal costruttore, non si osservano i tipici andamenti presenti in caso di perdite di diffusione. Da questo si è deciso di semplificare il modello impostando  $ilim$  [A/cm<sup>2</sup>] a  $\infty$ , annullando il contributo delle perdite di concentrazione. Un andamento tipico di questa sovratensione si ha quando per alte densità di corrente la tensione subisce un aumento di tipo esponenziale, nel caso di questo elettrolizzatore le curve tendono a mantenere un andamento rettilineo. Si consideri inoltre che nelle simulazioni non è

<sup>5</sup>È possibile trovare l'equazione per il calcolo delle attività dell'acqua sulla membrana in Dutta [16].

<sup>6</sup>Pari a 1 se si trascura il passaggio di una piccola parte di vapore acqueo attraverso la membrana, altrimenti è di poco inferiore a 1

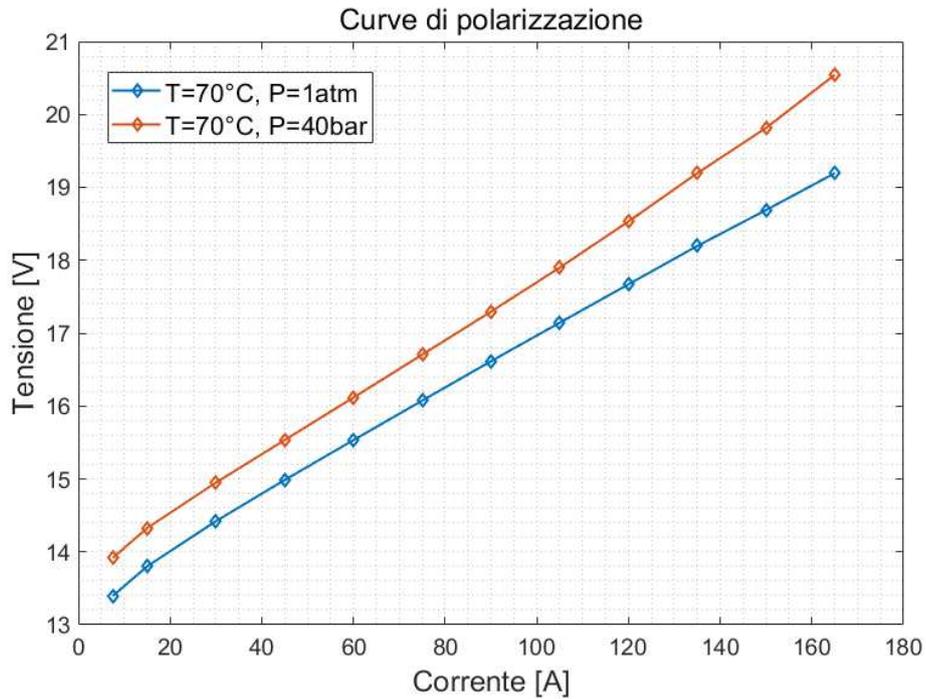


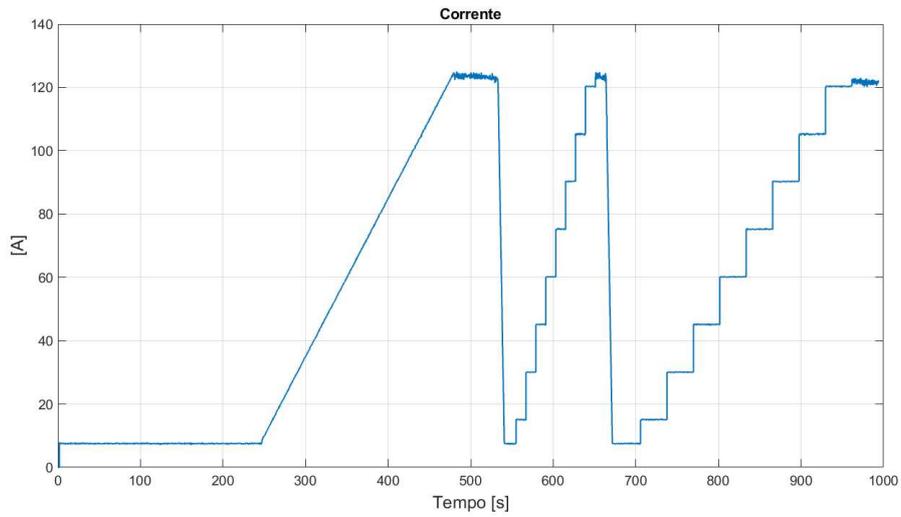
Figura 3.18.: Curve di polarizzazione dell'elettrolizzatore

mai stata superata una corrente di 135 A, non coinvolgendo quindi gli ultimi due tratti della curva di polarizzazione mostrata.

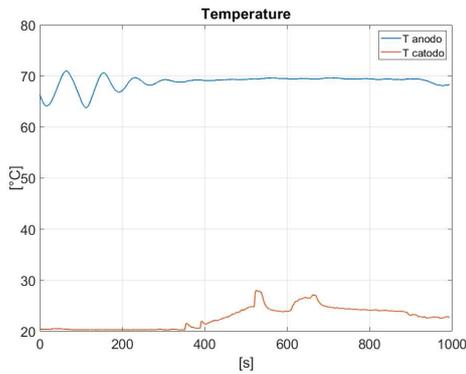
Come dataset per eseguire il primo tuning se ne è selezionato uno fra i disponibili che non presentasse anomalie nelle misurazioni, i suoi ingressi sono mostrati in Figura 3.19: la corrente presenta una rampa seguita da due serie di scalini, la temperatura dell'anodo è controllata intorno ai 70°C, mentre la temperatura del catodo non è controllata e si aggira fra i 20 e i 30°C, così come le pressioni che assumono un valore vicino alla pressione atmosferica.

Mettendo a confronto le uscite del dataset con quelle del modello, utilizzando i parametri elencati in precedenza si ottengono i risultati in Figura 3.20 con un IAE finale di 10781 per la tensione e 752.8204 per l'idrogeno. Dopo alcune variazioni manuali dei parametri si trova la combinazione riportata in Tabella 3.5, ottenendo le uscite in Figura 3.21 e un indice IAE di 187 per la tensione e 241 per la produzione di idrogeno. Si è tenuto in considerazione l'IAE, ma le considerazioni sarebbero analoghe se si fosse utilizzato l'ISE. L'ITAE non è stato preso in considerazione perché avrebbe pesato maggiormente lo scostamento delle misure ad un tempo più alto, ma non è corretto dal momento che non si sta valutando l'efficienza con cui un controllore riporta un segnale al valore di riferimento.

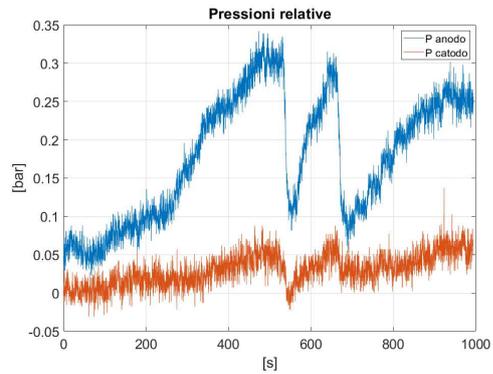
Capitolo 3. Implementazione del modello su Simulink



(a) Corrente dataset calibrazione manuale

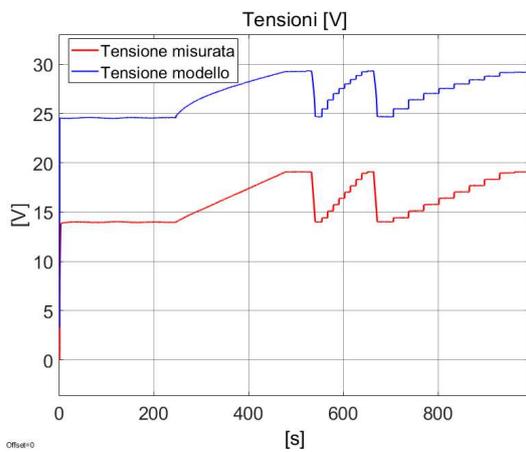


(b) Temperature dataset calibrazione manuale

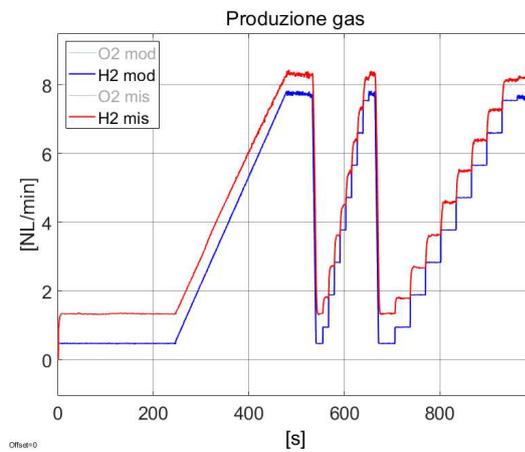


(c) Pressioni dataset calibrazione manuale

Figura 3.19.: Ingressi del dataset utilizzato per la calibrazione manuale



(a) Confronto tensioni



(b) Confronto produzione idrogeno

Figura 3.20.: Uscite prima della calibrazione manuale

Tabella 3.5.: Calibrazione manuale

| Parametri tensione |     | Parametri Gas |   |
|--------------------|-----|---------------|---|
| $\alpha_a$         | 0.5 | H2_Tau        | 1 |
| $\alpha_c$         | 0.5 | H2_Gain       | 1 |
| $i_{0,a}$          | 0.1 | H2_Offset     | 1 |
| $i_{0,c}$          | 0.1 |               |   |
| $\lambda$          | 15  |               |   |

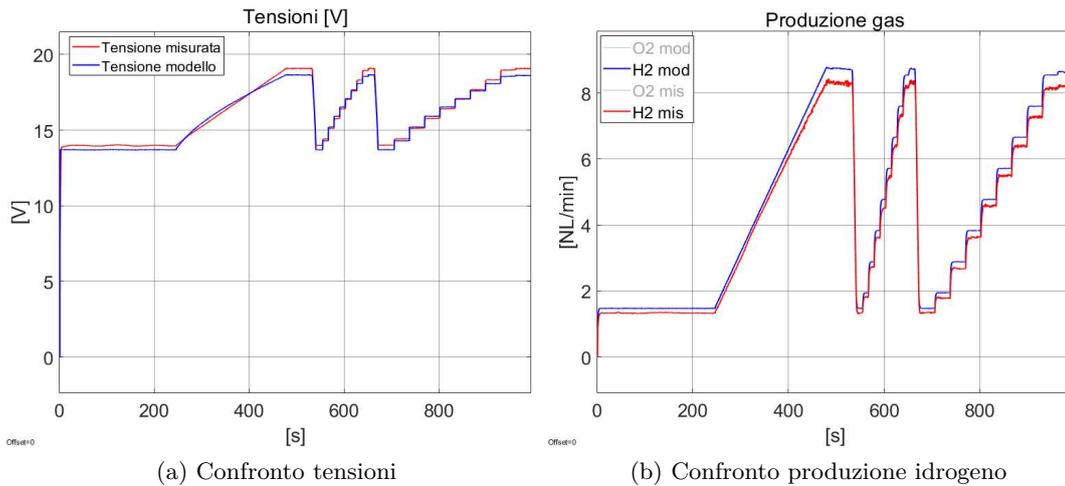


Figura 3.21.: Uscite dopo la calibrazione manuale

### 3.3.2. Seconda fase: Calibrazione tramite PSO

Avendo a disposizione più parametri, la calibrazione manuale può risultare complessa, inoltre, se si volessero verificare i parametri ottimi per un altro dataset si dovrebbe ripetere l'intera procedura manualmente. Per automatizzare la calibrazione si è scelto l'algoritmo PSO (Particle Swarm Optimization) [27], che simula il comportamento di uno sciame, in cui le singole componenti, dette particelle, si muovono nello spazio secondo delle precise regole. Lo sciame è formato da  $N$  particelle, queste si muovono in uno spazio di  $N_{var}$  dimensioni, con  $N_{var}$  pari al numero delle variabili che si stanno ricercando. La valutazione della bontà della soluzione è data da un indice di affinità, come può essere l'IAE in questo caso. Lo spostamento delle particelle è determinato dalla velocità che possiedono all'istante precedente e dalla loro conoscenza, che comprende: la migliore soluzione trovata dalla particella stessa fino a quel momento e la migliore soluzione ottima dal punto di vista globale trovata dallo sciame. L'equazione che descrive questo funzionamento e regola lo spostamento delle particelle è riportata in Eq.3.4, in cui  $v$  è la velocità della particella,  $i$  sono gli istanti di tempo,  $P$  la posizione della particella,  $P_{P_{Best}}$  l'ultima posizione migliore conosciuta dalla particella,  $P_{G_{Best}}$  l'ultima posizione migliore trovata a livello di sciame e seguono una serie di parametri:

### Capitolo 3. Implementazione del modello su Simulink

- $w$ : peso inerziale della particella
- $w_{damp}$ : smorzamento dell'inerzia
- $c_1$ : peso associato alla conoscenza personale della particella
- $c_2$ : peso associato alla conoscenza globale dello sciame
- $r_1$  e  $r_2$ : numeri casuali compresi fra 0 e 1 per introdurre stocasticità

$$v(i+1) = w(i+1) \cdot v(i) + c_1 \cdot r_1 \cdot (P(i) - P_{P_{Best}}) + c_2 \cdot r_2 \cdot (P(i) - P_{G_{Best}}) \quad (3.4)$$

$$w(i+1) = w(i) \cdot w_{damp} \quad (3.5)$$

**Funzionamento dell'algoritmo** Lo sciame è formato da una popolazione di  $N$  particelle, ogni particella ha un *Personal Best*, che contiene il miglior valore di fit da lei raggiunto e la posizione alla quale lo ha trovato. Mentre, tutte le particelle dello sciame sono a conoscenza del *Global Best*, ovvero il valore del miglior fit raggiunto e le sue coordinate. Ad ogni iterazione la posizione delle particelle è aggiornata con l'Eq.3.4, se le particelle trovano un *Personal Best* migliore del precedente lo aggiornano, lo stesso vale se una particella dello sciame trova un nuovo miglior *Global Best*. Inizialmente i *Personal Best* e il *Global Best*, se il fit migliora in senso decrescente, sono impostati a  $\infty$ , altrimenti sono impostati a 0, le posizioni iniziali delle particelle sono casuali e le loro velocità nulle. L'algoritmo si arresta se sono raggiunte un limite massimo di iterazioni, o se non ci sono miglioramenti al di sopra di una certa soglia nel *Global Best* per più di un certo numero di iterazioni consecutive, o se la velocità delle particelle cala sotto una determinata soglia.

**Applicazione dell'algoritmo** In questo caso si è deciso di effettuare la ricerca della combinazione dei parametri in maniera simultanea sia per i parametri della tensione che per quelli dell'idrogeno, ma si sono tenuti gli sciami separati. Questa separazione è giustificata dal fatto che i parametri per il calcolo della tensione e quelli per la produzione di idrogeno non influenzano gli uni il risultato degli altri. Avendo due sciami separati si possono gestire le due ricerche in maniera differente, impostando ad ognuna i propri pesi  $w$ ,  $c_1$ ,  $c_2$  e  $w_{damp}$ .

Per iniziare si impostano gli intervalli all'interno dei quali ricercare i parametri, si sono presi dei range attorno ai valori trovati con la calibrazione manuale che siano compatibili con la letteratura (gli intervalli utilizzati sono riportati in Tabella 3.22). Per la prima calibrazione i pesi assegnati ai due sciami sono gli stessi, si decide di pesare maggiormente la conoscenza collettiva, dunque vengono assegnati:  $w = 1$ ,  $c_1 = 1.5$ ,  $c_2 = 2$  e  $w_{damp} = 0.99$ . La dimensione degli sciami utilizzati è di 100

### 3.3. Calibrazione dei parametri

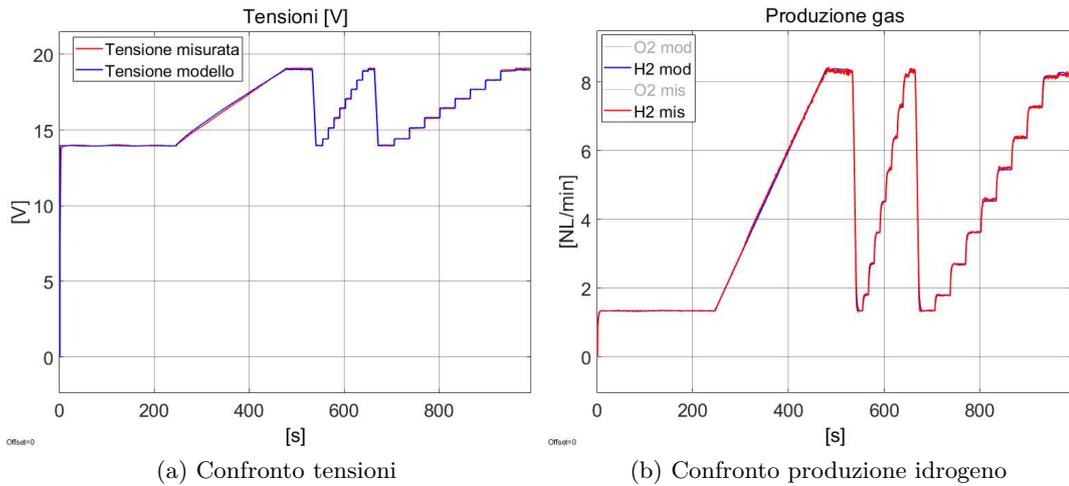


Figura 3.22.: Uscite dopo la prima calibrazione con PSO

particelle ognuno. La velocità<sup>7</sup> massima delle particelle non è la stessa lungo tutte le direzioni, ma è impostata ad un decimo degli intervalli sui quali sono limitati i parametri.

All'avvio, l'algoritmo genera due sciami di 100 particelle, uno con particelle di 5 dimensioni, l'altro di 3. Queste particelle vengono posizionate in maniera casuale nel loro spazio di ricerca con velocità nulle, i *Personal Best* delle particelle e i *Global Best* degli sciami sono inizializzati a  $\infty$ . Ad ogni nuova posizione di una particella, si carica la sua combinazione di parametri sul modello Simulink, lo si esegue, e si ottengono i valori dell'IAE per tensione e idrogeno in quella posizione. Nel caso riportato, dopo 60 cicli l'algoritmo si è arrestato per mancanza di miglioramenti dell'IAE al di sopra di 0.001 per più di 10 iterazioni (Figura 3.23). Dai due *Global Best* si estraggono le combinazioni di parametri riportati in Tabella 3.6 con in IAE di 62.3393 per la tensione e di 36.4492 per la produzione di idrogeno.

Tabella 3.6.: Prima calibrazione PSO

| Parametri tensione |                  |                                       | Parametri Gas |             |               |
|--------------------|------------------|---------------------------------------|---------------|-------------|---------------|
| $\alpha_a$         | [0.1 1]          | <b>0.9491</b>                         | H2_Tau        | [0.75 1.25] | <b>0.9671</b> |
| $\alpha_c$         | [0.1 1]          | <b>0.9992</b>                         | H2_Gain       | [0 5]       | <b>1.5419</b> |
| $i_{0,a}$          | [ $10^{-3}$ 0.5] | <b>0.5</b>                            | H2_Offset     | [0 1.5]     | <b>0.8802</b> |
| $i_{0,c}$          | [ $10^{-3}$ 0.5] | <b><math>6.4 \cdot 10^{-3}</math></b> |               |             |               |
| $\lambda$          | [5 25]           | <b>9.7989</b>                         |               |             |               |

**Modifiche dell'algoritmo** La combinazione dei parametri per il calcolo della produzione di idrogeno è stata trovata senza difficoltà dall'algoritmo, riportando lo stesso

<sup>7</sup>Nell'algoritmo, lavorando a tempo discreto, la velocità di una particella non è altro che il suo spostamento all'iterazione successiva

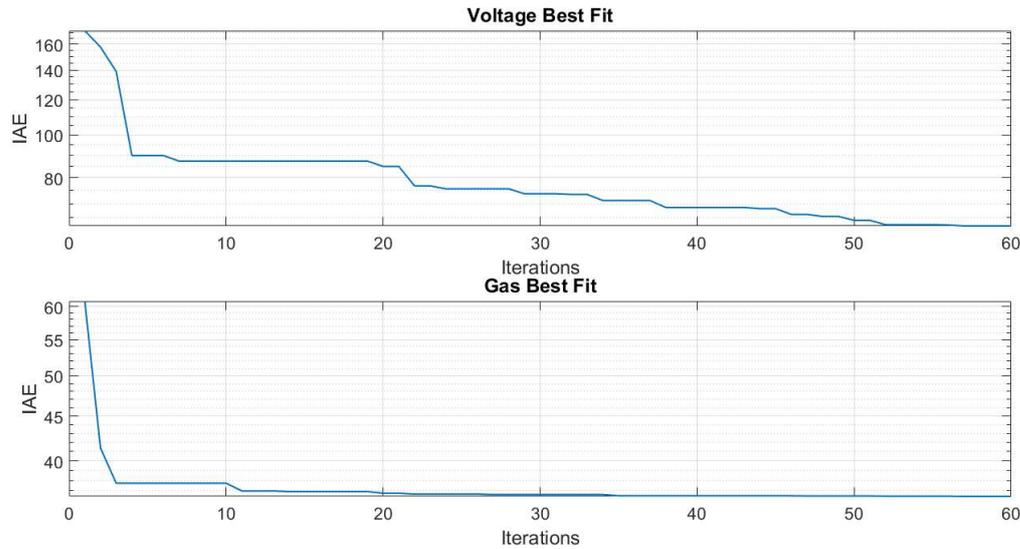


Figura 3.23.: Miglioramento dell'IAE durante la calibrazione con PSO

risultato per diverse esecuzioni, anche con pesi del PSO differenti. Per i parametri della tensione invece sorgono delle problematiche: il fit è migliorato notevolmente, ma soffermandosi al significato fisico del risultato, si nota che la  $i_{0,a}$  risulta molto più alta della  $i_{0,c}$ , al contrario di quanto riportato spesso in letteratura [13]. È opportuno anche osservare lo spostamento della popolazione mostrato in Figura 3.24<sup>8</sup>, dal quale si denota che i parametri dopo alcune iterazioni tendono ad annullare reciprocamente i loro contributi, tendendo ad andare incontro alla frontiera (tornado all'Eq.2.15 se ne comprendono le ragioni). Le proposte per correggere questo comportamento sono stati le seguenti:

- Variazione degli intervalli all'interno di cui cercare i parametri
- Variazione dei pesi del PSO
- Ipotizzare la condizione di simmetria nell'Eq.2.15:  $i_{0,a}=i_{0,c}$  e  $\alpha_a=\alpha_c$
- Scomporre la ricerca dei parametri in più fasi

La variazione degli intervalli, è stata fatta con l'obiettivo di ampliare ed esplorare altre zone dello spazio di ricerca, si sono aumentati i margini superiori dei parametri  $\alpha_a$  e  $\alpha_c$ , mentre per  $i_{0,a}$  si è diminuito quello inferiore. A questo punto è sorta un'altra problematica: durante la calibrazione si è notata la difficoltà a lavorare con i valori estremamente piccoli che poteva raggiungere  $i_{0,a}$ . Questo parametro

<sup>8</sup>La figura mostra lo spostamento delle particelle durante la ricerca: sull'asse delle x sono riportati i 5 o 3 parametri che ogni particella sta cercando, sull'asse y è riportato il valore assunto dai parametri normalizzato in un range da 0 a 1 rispetto al loro intervallo di ricerca, infine l'asse z rappresenta il numero di iterazioni. Con i cerchi neri è rappresentata la media dei parametri considerando l'intera popolazione in una specifica iterazione, mentre in rosso sono rappresentati i punti in cui è stato trovato un nuovo *Global Best*

è inserito al denominatore di un arcoseno iperbolico (Eq.2.15), per cui anche una sua piccola variazione ha effetti che possono essere molto evidenti sulla tensione. Una volta raggiunti valori di  $i_{0,a}$  inferiori a  $10^{-1}$ , l'IAE diventa molto sensibile alle variazioni del parametro e l'algoritmo impiega più tempo per ridurre la velocità attorno al punto ottimo. Per aumentare il grado di precisione con cui si cerca questo parametro si è inserita una conversione dalla scala lineare del PSO ( $\Delta_{PSO}$ ) a una scala logaritmica sulla quale viene riportata la  $i_{0,a}$  ( $\Delta_{i_0}$ ), dunque se si vuole imporre un intervallo di  $[10^{-6} \ 10^{-3}]$ , secondo la conversione riportata in Eq.3.6 i nuovi limiti da imporre nel PSO saranno  $[0 \ 4]$ . Il minimo dell'intervallo  $\Delta_{PSO}$  è impostato a 0 per semplificare la conversione, mentre le incognite da ricavare con l'Eq.3.6 sono: il massimo valore dell'intervallo di ricerca  $p_{max}$  e il termine  $c$  all'esponente.

$$\begin{aligned}
 1) \quad & \Delta_{PSO} = [0 \ p_{max}]; \\
 2) \quad & \Delta_{i_0} = [i_{0,min} \ i_{0,max}]; \\
 3) \quad & \Delta_{i_0} = 10^{\Delta_{PSO}-c}; \\
 4) \quad & c = -\log_{10}(i_{0,min}); \\
 5) \quad & p_{max} = \log_{10}(i_{0,max}) + c;
 \end{aligned} \tag{3.6}$$

A seguito delle variazioni sugli intervalli dei parametri si sono trovate combinazioni in grado di raggiungere un IAE più basso del risultato precedente (l'IAE per la tensione è sceso a 39.23), ma rimane il problema della divergenza dei parametri. Sono stati fatti dei tentativi variando i pesi del PSO, ma senza risolvere questo aspetto. Si è osservato che aumentando  $c_1$  rispetto a  $c_2$  rallenta velocità con cui lo sciame si sposta verso la soluzione ottima e in alcune situazioni potrebbe aiutare a non cadere in un minimo locale. Attraverso  $w_{damp}$  invece si può regolare il collasso dello sciame attorno al *Global Best* trovato dall'algoritmo,  $w_{damp}$  infatti va ad attenuare iterazione dopo iterazione l'inerzia della particella, lasciando solamente i contributi del *Personal Best* e *Global Best*.

Un altro tentativo è stato supporre per l'Eq.2.15 la condizione di simmetria, per cui i coefficienti  $\alpha_a$  e  $\alpha_c$  sono stati imposti a 0.5 e si sono cercati il  $\lambda$  e le densità di corrente  $i_{0,a}$  e  $i_{0,c}$ , supponendo  $i_{0,a}=i_{0,c}$ . I risultati però hanno riportato un IAE per la tensione di 194, molto più alto degli altri casi, facendo intuire che per allineare il modello non si potesse considerare la simmetria delle correnti di scambio.

L'ultima strada percorsa è stata la suddivisione della ricerca in 3 fasi: in una prima si ricercano tutti e 5 i parametri della tensione e ci si arresta al raggiungimento del fit oltre il quale si prevede che i parametri inizierebbero a divergere, nella seconda fase si fissano 2 dei 5 parametri e si cercano gli altri 3 (tra cui il  $\lambda$ ), una volta terminata anche la seconda fase, nella terza si vanno a cercare i 2 parametri che si erano fissati al passo precedente e ancora una volta il  $\lambda$ . Ad ogni passo vengono ristretti gli intervalli nei quali si cercano i parametri attorno al valore ottimo trovato al passo precedente, in questo caso si è preso il 25% del range originale centrato sull'ottimo del passo precedente. Il  $\lambda$  è l'unico parametro ad essere cercato in tutte le 3 fasi, poiché nelle calibrazioni precedenti si era assestato sempre attorno agli stessi valori.

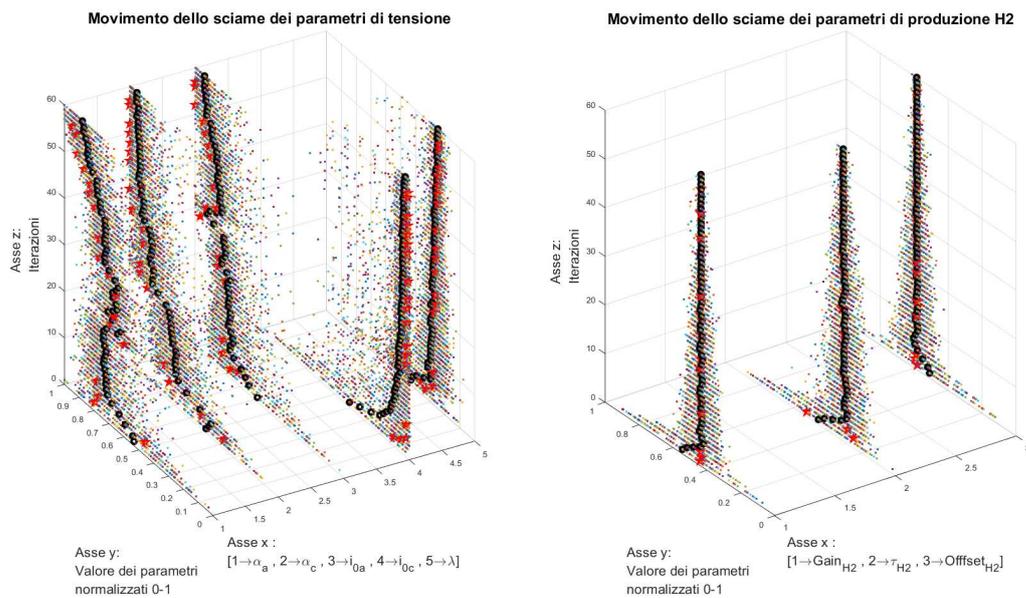


Figura 3.24.: Movimento dello sciame durante la calibrazione

Il procedimento è schematizzato in Tabella 3.7, nella quale i  $\Delta$  rappresentano gli intervalli di ricerca, i  $\Delta'$  e i  $\Delta''$  i loro progressivi restringimenti e i risultati delle ottimizzazioni sono indicati anch'essi con uno o più apici.

A seguito di questo procedimento si è migliorato ulteriormente l'IAE calcolato sull'errore della tensione raggiungendo un 39.09. Il miglioramento è piuttosto piccolo rispetto al precedente 39.23 ottenuto con una fase soltanto, ma se si osservano i risultati di questa ottimizzazione a Tabella 3.8 e Figura 3.25, si può comprendere perché la variazione di solamente lo 0.36% sia dovuta alla già ottima stima effettuata dal modello.

Tabella 3.7.: Suddivisione della calibrazione in 3 fasi

| <b>Step 1</b>  |                |                       |                |
|----------------|----------------|-----------------------|----------------|
| Nome parametro | Valore fissato | Intervallo di ricerca | Ottimo trovato |
| $\alpha_a$     | -              | $\Delta_{\alpha_a}$   | $\alpha'_a$    |
| $\alpha_c$     | -              | $\Delta_{\alpha_c}$   | $\alpha'_c$    |
| $i_{0,a}$      | -              | $\Delta_{i_{0,a}}$    | $i'_{0,a}$     |
| $i_{0,c}$      | -              | $\Delta_{i_{0,c}}$    | $i'_{0,c}$     |
| $\lambda$      | -              | $\Delta_\lambda$      | $\lambda'$     |

| <b>Step 2</b>  |                |                       |                |
|----------------|----------------|-----------------------|----------------|
| Nome parametro | Valore fissato | Intervallo di ricerca | Ottimo trovato |
| $\alpha_a$     | $\alpha'_a$    | -                     | $\alpha'_a$    |
| $\alpha_c$     | $\alpha'_c$    | -                     | $\alpha'_c$    |
| $i_{0,a}$      | -              | $\Delta'_{i_{0,a}}$   | $i''_{0,a}$    |
| $i_{0,c}$      | -              | $\Delta'_{i_{0,c}}$   | $i''_{0,c}$    |
| $\lambda$      | -              | $\Delta'_\lambda$     | $\lambda''$    |

| <b>Step 3</b>  |                |                       |                |
|----------------|----------------|-----------------------|----------------|
| Nome parametro | Valore fissato | Intervallo di ricerca | Ottimo trovato |
| $\alpha_a$     | -              | $\Delta'_{\alpha_c}$  | $\alpha''_a$   |
| $\alpha_c$     | -              | $\Delta'_{\alpha_c}$  | $\alpha''_c$   |
| $i_{0,a}$      | $i''_{0,a}$    | -                     | $i''_{0,a}$    |
| $i_{0,c}$      | $i''_{0,c}$    | -                     | $i''_{0,c}$    |
| $\lambda$      | -              | $\Delta''_\lambda$    | $\lambda'''$   |

Tabella 3.8.: Risultati calibrazione in 3 fasi

| <b>Parametri tensione</b> |                     |  |
|---------------------------|---------------------|--|
| $\alpha_a$                | [0.1 2]             | <b>2</b>                               |
| $\alpha_c$                | [0.1 1.5]           | <b>1.038</b>                           |
| $i_{0,a}$                 | $[10^{-6} 10^{-2}]$ | <b><math>7.67 \cdot 10^{-4}</math></b> |
| $i_{0,c}$                 | [0.1 1]             | <b>0.969</b>                           |
| $\lambda$                 | [5 25]              | <b>8.779</b>                           |

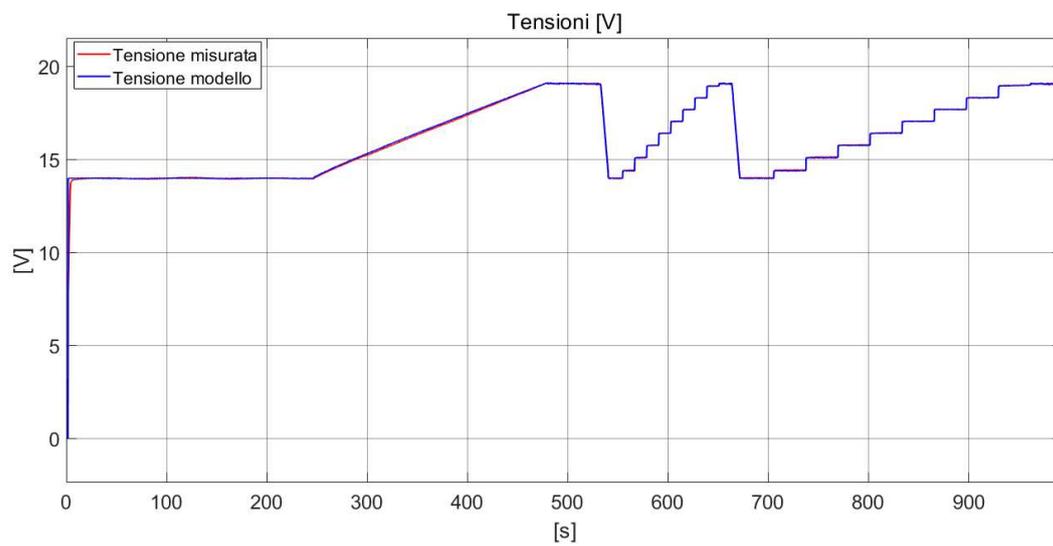


Figura 3.25.: Confronto fra la tensione stimata dal modello e le misurazioni del dataset a seguito della calibrazione in 3 fasi

## Capitolo 4.

# Degradazione e guasti di un elettrolizzatore

### 4.1. Tipologie di degradazione in un elettrolizzatore PEM

I componenti di un elettrolizzatore sono soggetti a diverse tipologie di degradazione, che se non considerate, oltre a cali di prestazione, comportano l'aumento del rischio di potenziali guasti. Fra i componenti sottoposti a usura di un elettrolizzatore PEM, vi è sicuramente la membrana, questa può logorarsi sia dal punto di vista meccanico per via delle alte pressioni, che dal punto di vista termico o chimico. Una membrana degradata, oltre alla riduzione di conducibilità può soffrire di fenomeni di cross-leak, o nel peggiore dei casi, subire una foratura. Una foratura, a causa della composizione a stack della maggior parte degli elettrolizzatori, comporta l'arresto dell'intero sistema per poter sostituire la membrana fuori uso.

Come la membrana, anche gli elettrodi ed il catalizzatore sono soggetti a diverse tipologie di usura, queste ne riducono la capacità di scambiare le cariche elettriche, sfavorendo la reazione di elettrolisi. [9, 25, 28]

Il possedere una conoscenza sull'evolversi della degradazione dell'elettrolizzatore può aiutare a prolungare la vita del dispositivo e nel mentre pianificare una sostituzione. Per esempio, pagando con un calo della produzione, si potrebbero modificare le condizioni di lavoro, come temperature, pressioni e corrente per rallentare i fenomeni di usura.

### 4.2. Possibili approcci alla prognosi

Gli effetti visibili delle degradazioni sull'elettrolizzatore possono essere:

- Variazione della quantità di gas prodotti e del loro rapporto [28]
- Incremento dell'impurità dei gas prodotti dovuto dal cross-leak
- Aumento delle sovratensioni
- Variazioni nella dinamica della tensione

In Kheirrouz [9] sono presentate una serie di possibili strade per affrontare la diagnosi dello stato di salute di un elettrolizzatore, le alternative offrono possibilità per la diagnosi guasti o la prognosi del tempo di vita rimanente. Queste tecniche sono basate su metodi differenti e non sempre risultano applicabili, alcune richiedono dei test specifici da compiere sull'elettrolizzatore, come la spettroscopia di impedenza o la voltammetria, altre si basano sull'analisi del segnale o sull'intelligenza artificiale, per le quali si richiede una gran mole di dati con specifiche caratteristiche che siano ripetute nel tempo. Altre metodologie riportate, come l'osservatore, le equazioni di parità e l'identificazione parametrica invece fanno utilizzo di un modello matematico.

Nella pubblicazione di Boulevard [29] si analizza il comportamento della tensione al variare del riferimento di corrente. Viene osservato che a fronte di variazioni a scalino della corrente la tensione tende ad avere delle sovra elongazioni, variabili in ampiezza a seconda del valore della corrente. L'analisi di questi aspetti viene proposta come un possibile punto di partenza per la valutazione dello stato di salute dell'elettrolizzatore.

Un approccio presentato da Parakash [30], è basato sull'ottenimento di residui provenienti da un modello grafico (ottenuto attraverso un Bond graph), con i quali si addestra un modello che possa predire il tempo di vita rimanente dell'elettrolizzatore e isolarne il tipo di degradazione presente.

### **4.3. Utilizzo del modello**

Per poter scegliere un metodo con il quale proseguire si deve valutare che tipo di dati si hanno a disposizione: quelli che si possiedono sono dei test di vario tipo, tutti effettuati a scopo sperimentale su di una nuova apparecchiatura, le acquisizioni non sono state fatte con il principale scopo di ottenere dati per una precisa metodologia di diagnosi. Si vanno quindi ad escludere le tecniche ad approccio statistico perché non si hanno dati a sufficienza e le reti neurali perché non si dispone di simulazioni omogenee. Il campionamento del segnale effettuato a 4 Hz preclude l'utilizzo di metodologie basate sull'analisi dello spettro del segnale.

Fatte queste osservazioni si è optato per l'utilizzo di metodologie euristiche basate sull'errore fra la predizione del modello e i valori misurati. L'idea è quella di allineare il modello Simulink, a dei dataset acquisiti sull'elettrolizzatore ad inizio vita, successivamente andare ad utilizzare questo modello su un dataset in cui il componente si sia degradato o si stia degradando per ottenere la differenza, a fronte degli stessi ingressi, fra le uscite del modello e quelle del dataset. Da questa differenza si vogliono estrarre delle informazioni che possano determinare le condizioni di salute dell'elettrolizzatore.

Come dataset per la calibrazione si sono selezionate 4 acquisizioni effettuate i giorni immediatamente successivi all'istallazione del componente (BoT: Begining of Time), questi 4 dataset sono differenti fra loro per la pressione a cui è sottoposto il catodo e per il tipo di curve di corrente imposte in ingresso. Sono indicate come

polarizzazioni di tipo "A" le curve di corrente con rampe come in Figura 4.1a, mentre con "B" quelle in cui le rampe sono intervallate da delle zone a valore costante, come in Figura 4.1b.

Nei dataset si è riscontrato un problema sull'acquisizione del flusso dei gas prodotti, su 3 dei 4 dataset presi come BoT, il controllo in pressione sul catodo ha impedito la corretta misurazione del flusso d'idrogeno, poiché la valvola che era addetta al controllo della pressione, aprendosi e richiudendosi causava delle brusche fluttuazioni nel flusso di idrogeno. In Figura 4.2a si nota come il flusso registrato per l'idrogeno è nullo fino al raggiungimento dei 20 bar, poiché la valvola resta chiusa per poter pressurizzare il gas, dopodiché il suo stato viene commutato più volte per mantenere la pressione, causando una sequenza di spike. Il dataset a pressione atmosferica non presenta la problematica introdotta dal controllo di pressione, ma non rispetta il rapporto 2 a 1 fra l'idrogeno e l'ossigeno prodotti, che normalmente regola un processo di elettrolisi (Figura 4.2b), questo può essere dovuto ad un errore di calibrazione dei sensori.

A seguito di queste osservazioni sui dataset, lo studio si concentrerà sulle misure della tensione in uscita, perciò la calibrazione dei parametri e l'errore fra modello e dataset considereranno solamente la tensione dell'elettrolizzatore.

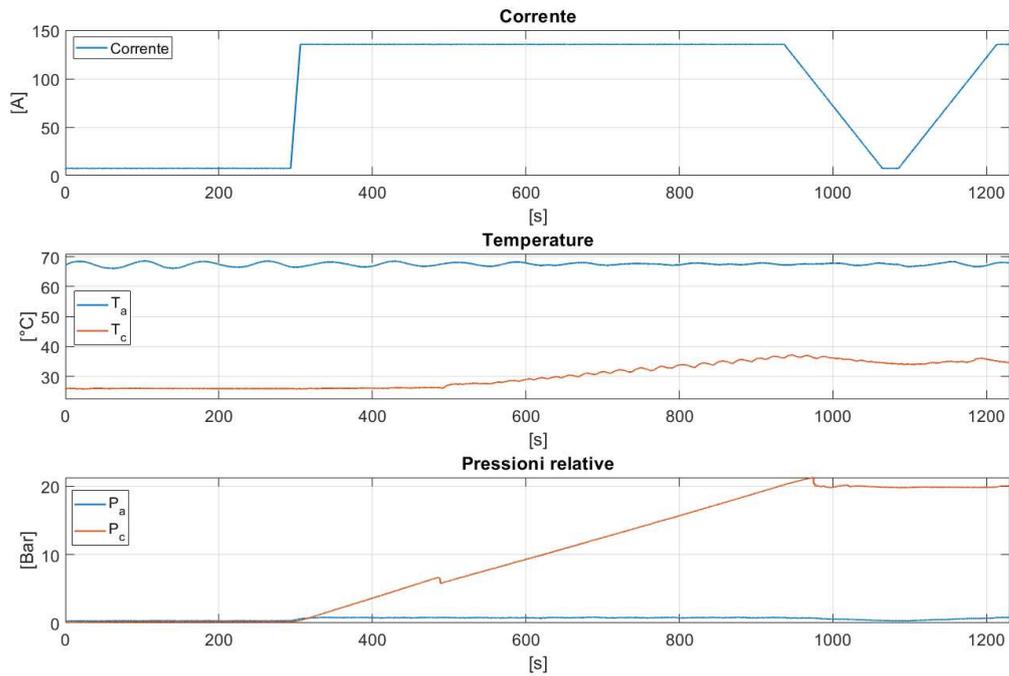
Come dataset di fine vita se n'è scelto uno acquisito poco tempo prima della foratura della membrana, questo verrà indicato come dataset EoT (End of Time). Il dataset in questione ha codice "111115" ed è stato selezionato perché contiene un'acquisizione in cui si è mantenuto per un lungo periodo di tempo un valore costante di corrente (Figura 4.3). È stata ricercata questa caratteristica per evidenziare comportamenti di degradazione a fronte di una condizione di ingresso costante.

Si sono quindi presi i 4 dataset BoT e per ognuno è stata effettuata la calibrazione dei parametri, al fine di effettuare un confronto si è eseguita la calibrazione anche sul dataset EoT. In Figura 4.4, per ogni parametro è riportato: il risultato del parametro a seguito della calibrazione sui quattro dataset BoT, la media di questi quattro risultati e il valore del parametro a seguito della calibrazione sul dataset EoT. In Figura 4.5 sono riportati gli scostamenti netti e percentuali dei parametri del dataset EoT rispetto alla media dei dataset BoT. Quello che risulta evidente è che il parametro  $\lambda$  è quello che nel caso del dataset EoT presenta una maggiore variazione rispetto alla media BoT, mentre la maggior variazione percentuale è di  $i_{0,a}$ .

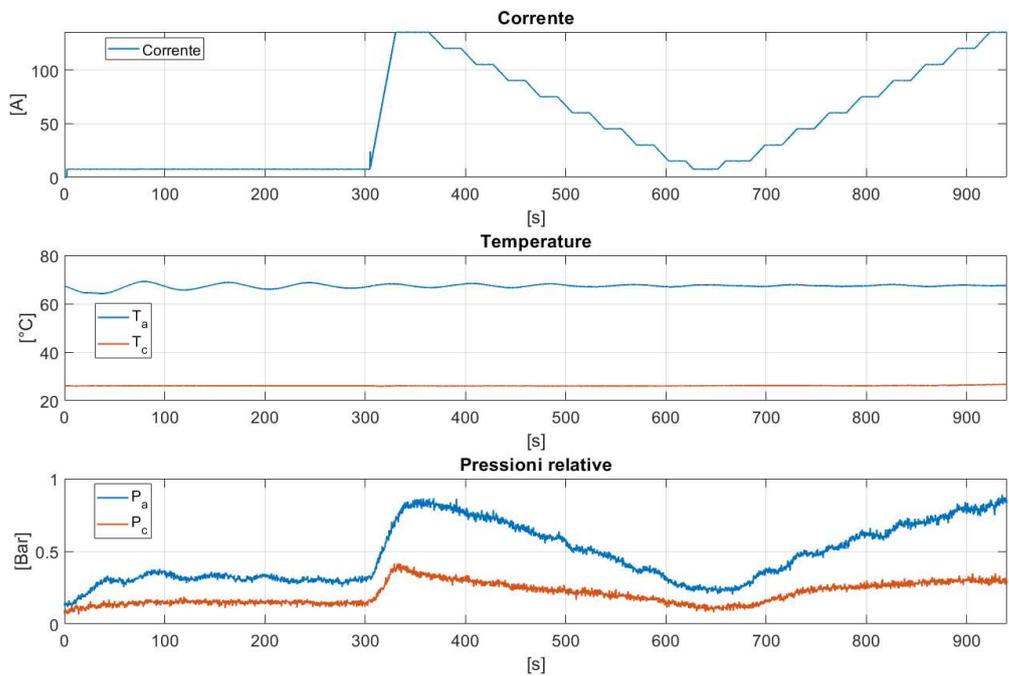
Prendendo la media dei parametri ottenuti per i dataset BoT (che da ora saranno indicati come "parametri BoT"), caricandoli nel modello e lanciando le simulazioni con gli ingressi dei quattro dataset, il modulo dell'errore fra la tensione del modello e quella dei dataset non supera gli 0.3 V (Figura 4.6), al contrario, se si considerano gli ingressi e le uscite del dataset EoT e si esegue il modello con i parametri BoT, si supera abbondantemente la precedente soglia. Altra caratteristica interessante è l'ultimo tratto, nel quale a fronte di una corrente costante il modulo dell'errore mostra un andamento crescente (Figura 4.7).

Si vuole estrarre da un comportamento come quello presente sul dataset 111115

Capitolo 4. Degradazione e guasti di un elettrolizzatore

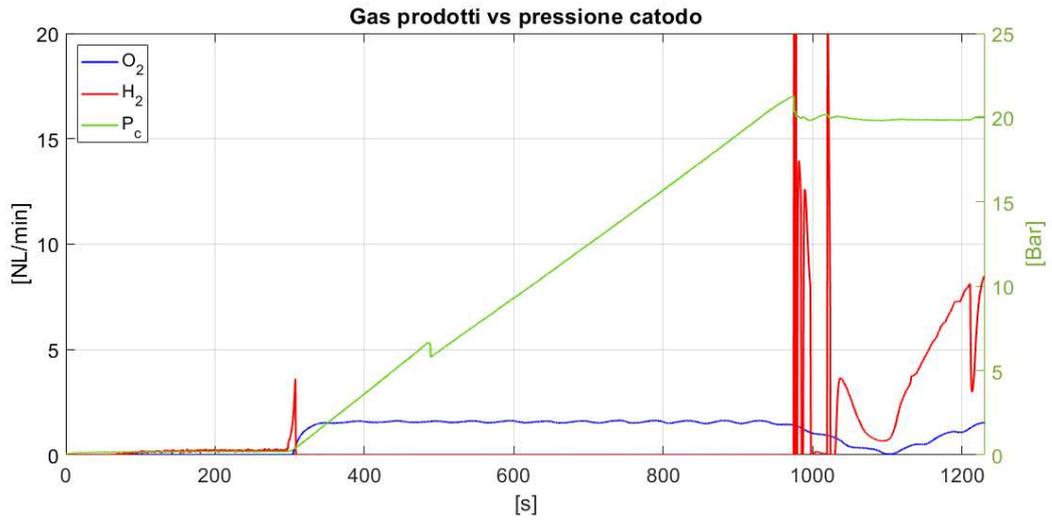


(a) Dataset 175526 (Tipo A, 20 Bar)

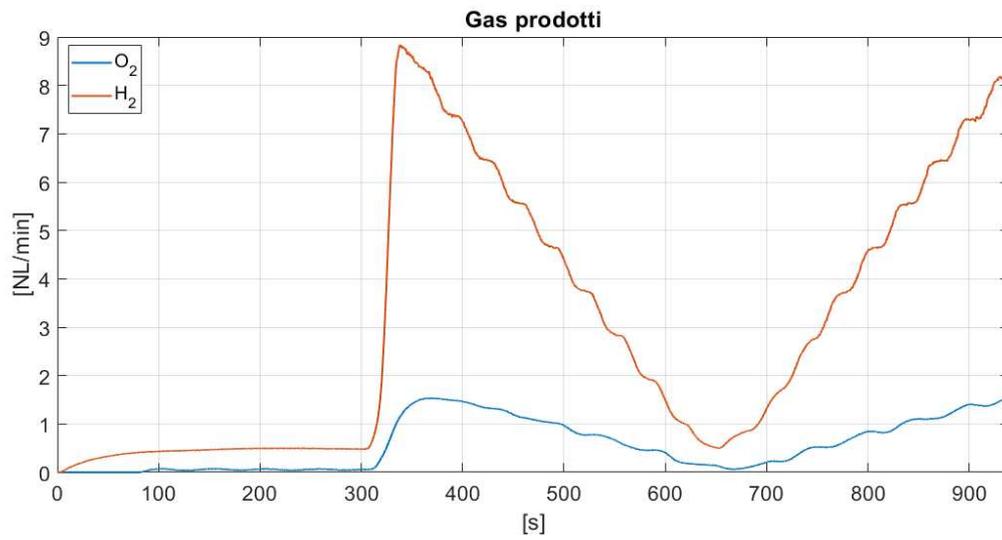


(b) Dataset 150234 (Tipo B, P atm)

Figura 4.1.: Corrente, temperature e pressioni di due dataset BoT



(a) Effetto del controllo in pressione sulla misurazione del flusso d'idrogeno



(b) Mancanza del rapporto 2:1 fra idrogeno e ossigeno prodotti

Figura 4.2.: Esempi di misurazioni di flusso non utilizzabili

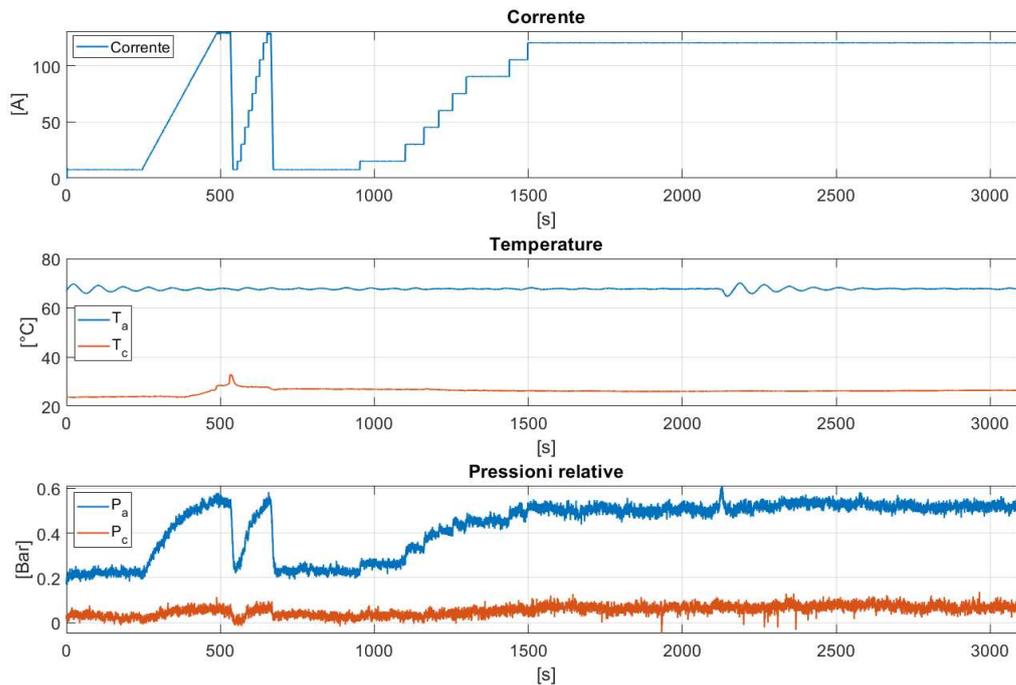


Figura 4.3.: Corrente, temperature e pressioni del Dataset EoT

una grandezza che possa determinare se l'elettrolizzatore si stia degradando o se stia divergendo verso una soglia oltre la quale non si ritiene sicuro, o conveniente, lavorare.

#### 4.3.1. Selezione delle finestre

Si vogliono selezionare delle finestre temporali in cui le condizioni di ingresso si possono considerare costanti, sulle quali valutare l'andamento del modulo dell'errore per cercare delle informazioni che indichino la presenza o meno della degradazione del componente. Come ingressi si sono considerati quelli che maggiormente mostrano influenza nei dataset, ovvero la corrente e la temperatura dell'anodo. Per considerare costante un tratto di segnale, per quell'insieme di campioni si calcolano, relativamente ai due input di corrente e temperatura anodica, i seguenti indicatori:

- **Deviazione standard** La deviazione standard dei campioni presi in considerazione. Controllo che mira a scartare le porzioni dei segnali in cui è presente più rumore.
- **Pendenza della retta interpolata** La pendenza della retta che meglio approssima la serie di campioni considerati. Controllo che mira a scartare le zone in cui i segnali tendono ad avere un andamento crescente o decrescente troppo accentuato.

### 4.3. Utilizzo del modello

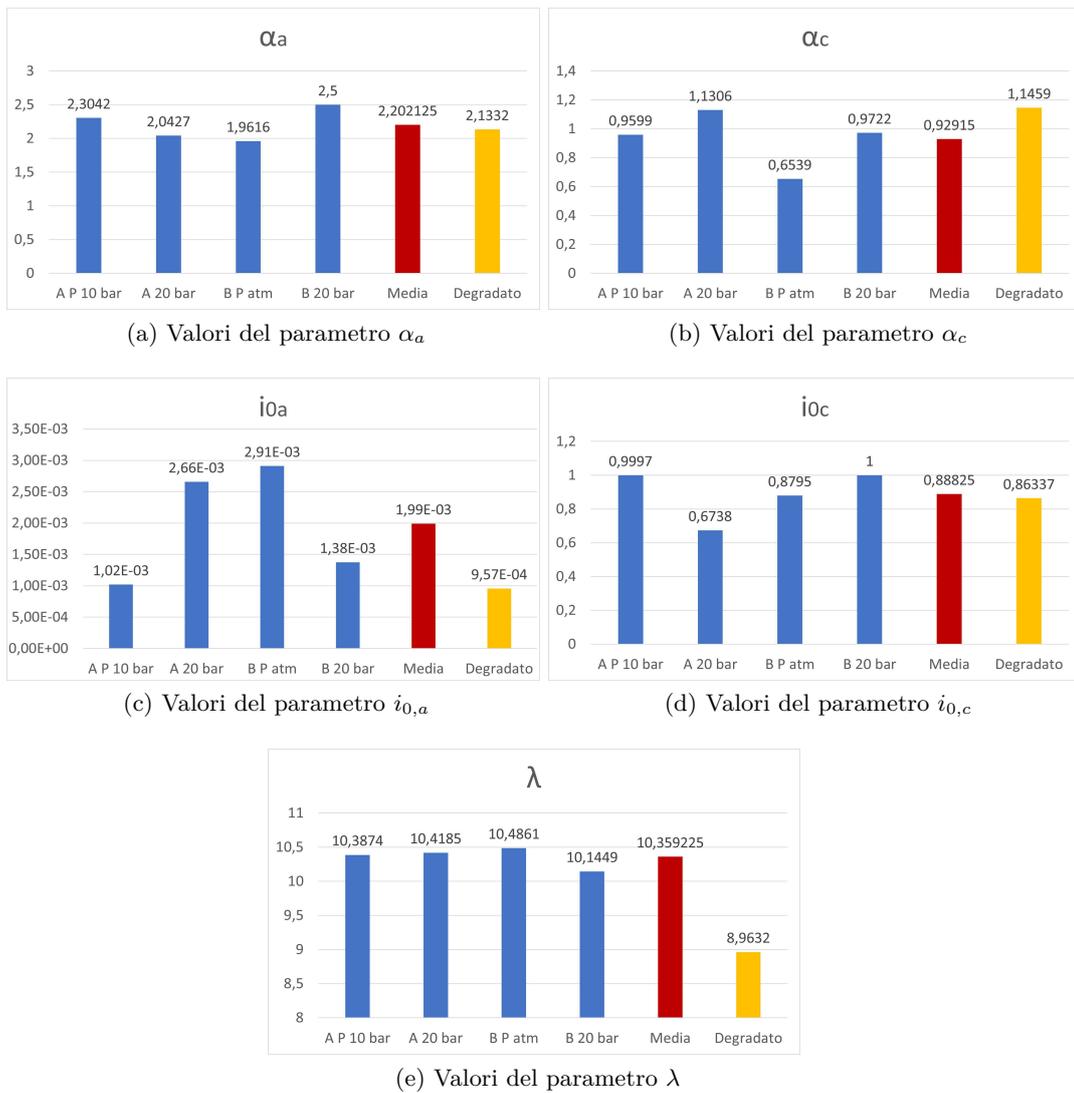


Figura 4.4.: Blu: Valore dei parametri seguito delle calibrazioni sui 4 dataset BoT;  
 Rosso: Valore medio delle prima 4 colonne;  
 Giallo: Valore dei parametri seguito della calibrazione sul dataset EoT;

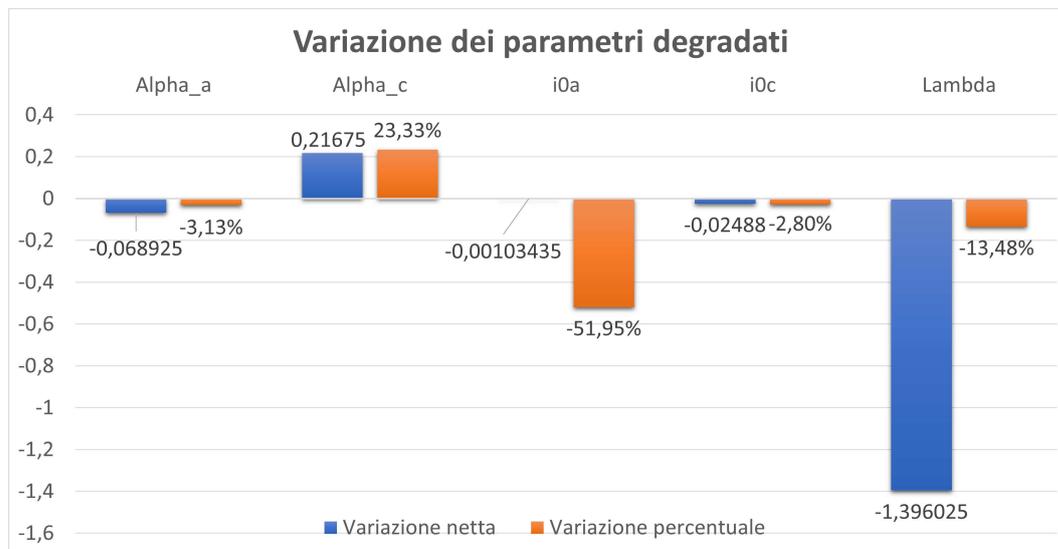


Figura 4.5.: Variazione dei parametri calibrati per il dataset EoT in relazione alla media di quelli calibrati sui 4 dataset BoT

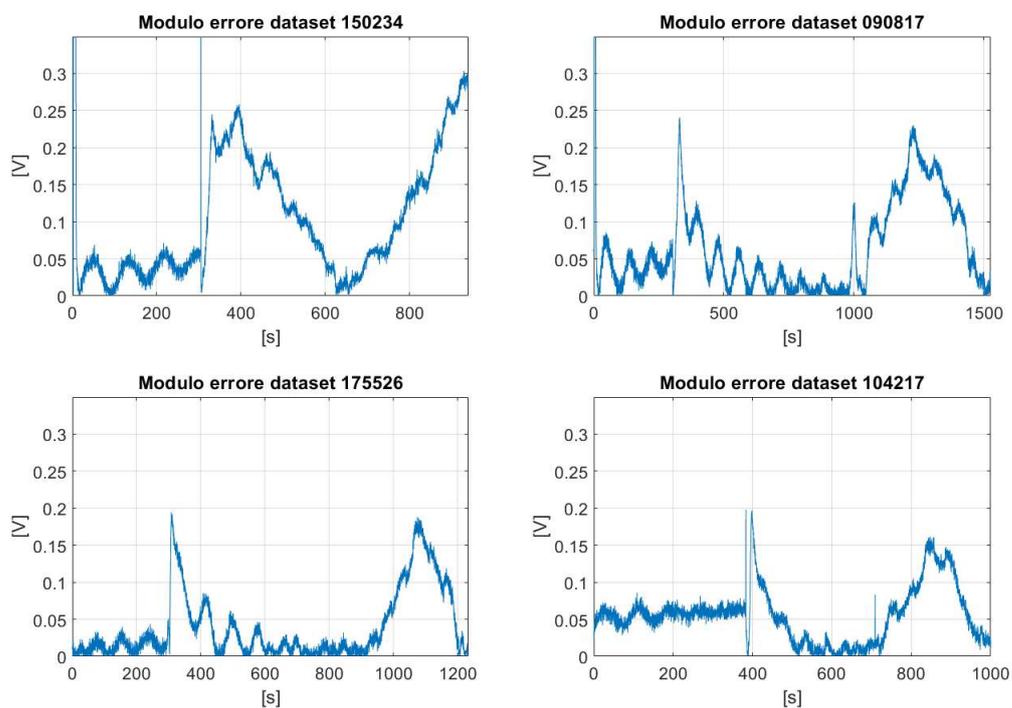


Figura 4.6.: Modulo dell'errore fra il modello calibrato con i parametri medi BoT e le misurazioni dei 4 dataset BoT

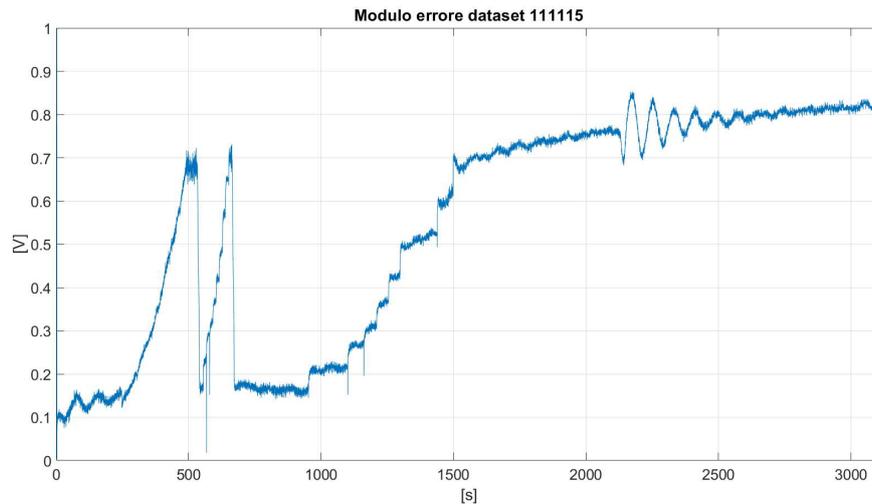


Figura 4.7.: Modulo dell'errore fra il modello calibrato con i parametri medi BoT e la tensione del dataset EoT

- **Differenze fra i campioni alle estremità e la media dei restanti** La differenza fra il primo campione tra quelli considerati (testa) e la media dei restanti; la stessa differenza viene calcolata fra l'ultimo campione di quelli considerati (coda) e la media dei restanti. Rende la selezione delle finestre più reattiva alla presenza di scalini, di contro uno spike può spezzare una finestra. Nell'algoritmo riportato in Appendice A.2, il parametro `fuoriMedia` va a modificare il numero di campioni di testa e di coda che vengono considerati, se pari a 1 il funzionamento è come quello appena descritto, se pari a 0 questo controllo viene a meno, se maggiore di 1 vengono considerati più campioni. Se si aumenta questo parametro, è più probabile che spike di lieve entità non causino l'interruzione di una finestra.

A seguito del calcolo di queste misure si effettua un controllo per verificare se siano al di sotto di alcune soglie prestabilite: `sogliaDev` per la deviazione standard, `sogliaPend` per la pendenza della retta e `sogliaMod` per i campioni di testa e di coda che ricadono fuori dalla media.

L'algoritmo per la formazione delle finestre scansiona i segnali di ingresso campione per campione, a simulare l'ingresso di dati a run-time. Si può considerare come se i due input fossero scansionati da due puntatori: uno di testa che indica il primo campione che si sta considerando e uno di coda che rappresenta il campione considerato nell'istante corrente. La sequenza di operazioni è mostrata nel diagramma di flusso in Figura 4.8. La dimensione delle finestre non è fissa, una volta iniziata una finestra questa viene espansa fino a che i campioni successivi soddisfano le condizioni di verifica. È possibile assegnare un limite alla dimensione massima di una finestra tramite la variabile `maxSizeF`, oppure lasciarlo a `Inf`.

Le operazioni di verifica elencate in precedenza vengono effettuate solamente se si sono accumulati almeno un numero di campioni pari a `sizeF`, con la variabile `succ` si

tiene il conto di quante volte consecutive la verifica ha dato esito positivo. Quando `succ` raggiunge il valore `delay`, si "Conferma" una finestra, ossia si considera quel tratto di segnale come costante e si calcolano gli indici mostrati in 4.3.2, se la finestra era già stata confermata in una fase precedente si calcolano direttamente gli indici omettendo l'operazione di conferma. Nel confermare una finestra si aggiorna la testa spostando il puntatore in avanti di una percentuale di `delay`. Se la verifica delle condizioni di segnale costante ha esito negativo, la variabile `succ` viene resettata e se era presente una finestra valida il puntatore di testa viene eguagliato a quello di coda, altrimenti, il puntatore di testa viene spostato avanti di un campione.

In Figura 4.9 è mostrato l'output dell'algoritmo per il dataset EoT 111115, in base alle soglie che si vanno ad impostare si può determinare la sensibilità alle variazioni di temperatura e corrente. Mentre con le variabili `sizeF`, `delay` e `coda` si regola la quantità di tempo per la quale si vuole che il segnale risulti costante prima di confermare la finestra. In particolare, la variabile `coda` rappresenta la percentuale di campioni da mantenere nella fase di conferma della finestra, serve ad escludere dalla finestra sfasamenti dell'errore, dovuti ad un precedente cambio dei riferimenti in ingresso.

### 4.3.2. Calcolo degli indici

A partire dal modulo dell'errore, fra tensione del modello e del dataset, calcolato in corrispondenza delle finestre precedentemente ottenute, sono stati calcolati diversi indici. Si è osservato che la deviazione standard e la pendenza di una retta che approssimasse il modulo dell'errore all'interno della finestra, non fornivano risultati che distinguessero un caso BoT da uno EoT. Si è presa in considerazione anche la monotonicità del modulo dell'errore, ma anche in questo caso senza ottenere distinzioni fra le casistiche, poiché questo indice non tiene in considerazione la magnitudo della pendenza del segnale, ma solo se si mantiene un andamento monotono e proprio per questa ragione è fortemente intaccato da eventuali oscillazioni.

Un altro tentativo si è fatto andando ad interpolare un polinomio di terzo grado sulle porzioni di errore contenute nelle finestre, con questo polinomio si sono estratte due informazioni: la stima del tempo che il modulo dell'errore impiegherebbe per raggiungere una soglia preimpostata e la classificazione del tipo di andamento che l'errore assume nella finestra.

In ultimo si è implementata una stima parametrica inversa per verificare quanto i parametri si stessero discostando dai valori presenti nel modello.

**Classificazione dell'andamento dell'errore all'interno della finestra** Si è preso un polinomio di terzo grado, come quello descritto nell'Eq.4.1, per approssimare l'andamento del modulo dell'errore, dunque la  $x$  rappresenta il tempo e la  $y$  il valore del modulo dell'errore. Per un polinomio di questo tipo si sono distinte sette tipologie di andamento per  $a > 0$  (Figura 4.10) e altre sette, speculari rispetto l'asse orizzontale, per  $a < 0$ . Le sette tipologie vengono individuate confrontando la posizione dei margini

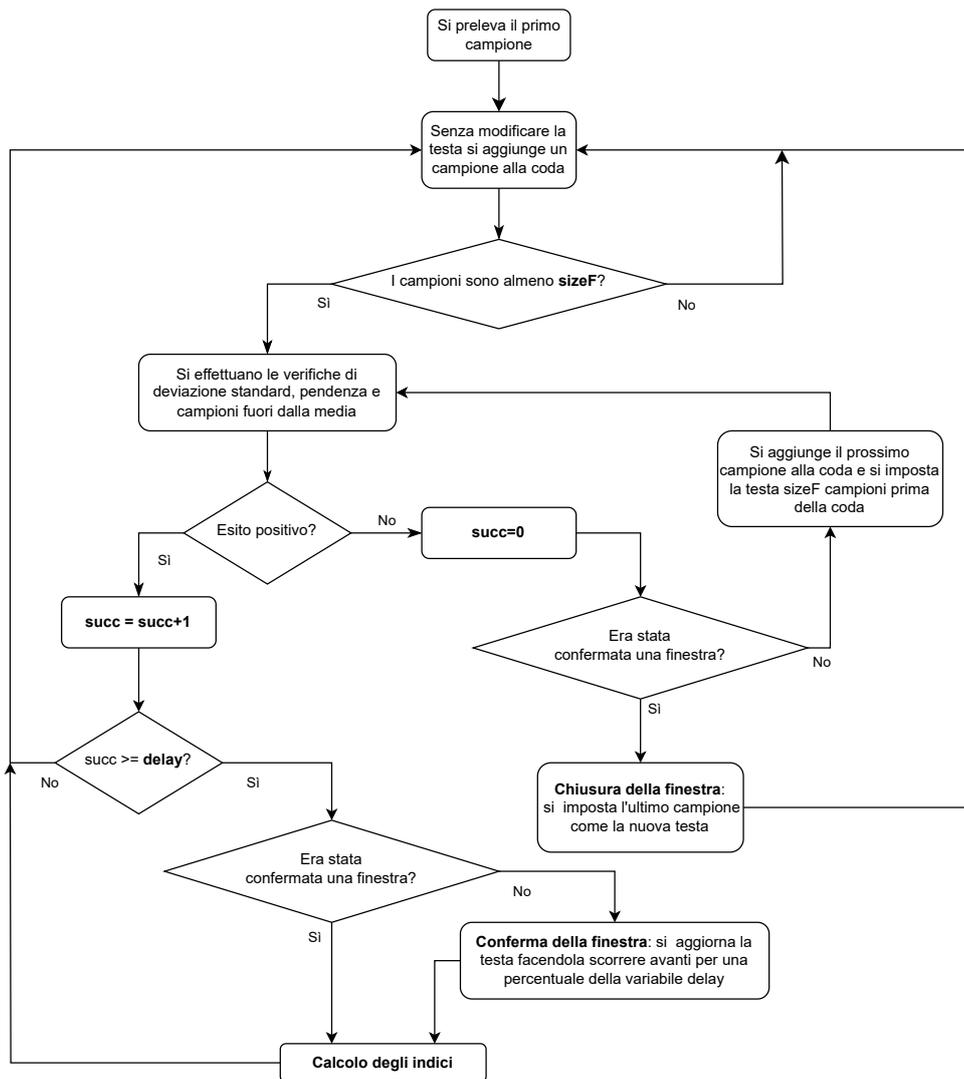


Figura 4.8.: Diagramma di flusso dell’algoritmo per la formazione delle finestre e calcolo degli indici

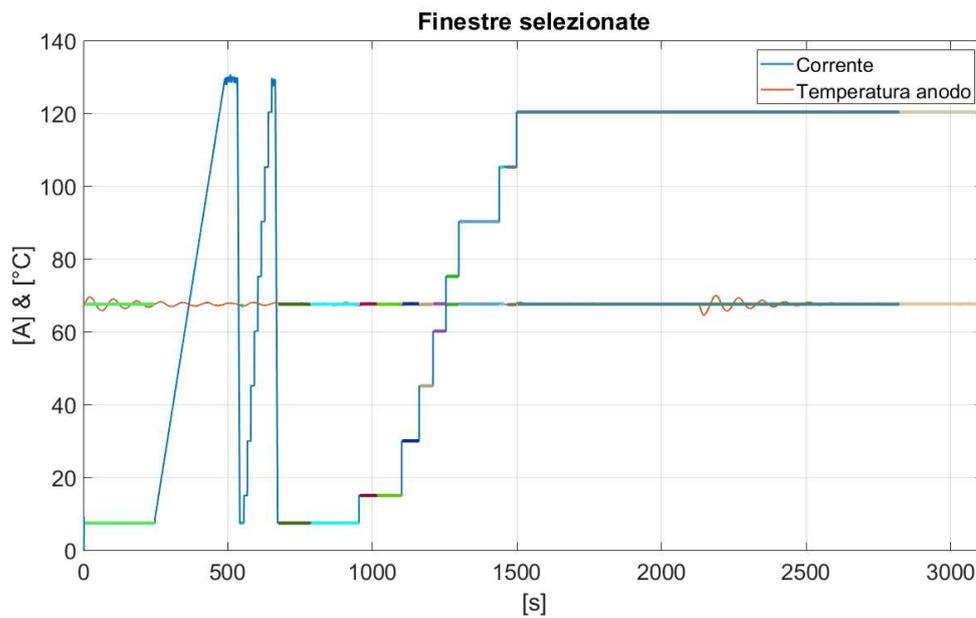


Figura 4.9.: Finestre temporali selezionate sulla corrente e la temperatura del dataset EoT 111115

Le coppie di rette, tracciate sopra i due segnali di ingresso, rappresentano le zone in cui sono state riconosciute delle finestre, ogni colore rappresenta una finestra differente. L'estensione orizzontale delle rette rappresenta gli istanti di tempo che fanno parte della finestra, mentre l'altezza alla quale sono graficate è pari al valor medio del segnale di ingresso in quell'intervallo di tempo.

della finestra con la posizione dei flessi del polinomio, se i flessi non sono presenti si ha una tipologia 7. Le tipologie vengono poi ripartite in quattro classi secondo la Tabella 4.1.

$$y = ax^3 + bx^2 + cx + d \quad (4.1)$$

Tabella 4.1.: Ripartizione delle Tipologie 1÷7 nelle 4 classi

| Classe | $a > 0$ | $a < 0$ | Descrizione classe              |
|--------|---------|---------|---------------------------------|
| A      | 1,5,7   | 3       | Andamento crescente             |
| B      | 3       | 1,5,7   | Andamento decrescente           |
| C      | 2,4     |         | 1 cambio di pendenza (parabola) |
| D      | 6       |         | 2 cambi di pendenza             |

Il polinomio è ottenuto tramite la funzione `polyfit` di Matlab, alla quale vengono passati i valori delle ordinate, quelli delle ascisse e il grado del polinomio che si vuole ottenere (in questo caso 3), in uscita si ottiene il vettore con i 4 coefficienti  $a$ ,  $b$ ,  $c$  e  $d$  dell'Eq.4.1 [31].

Dato che la stima del polinomio deve essere effettuata in regioni dell'asse dei tempi che possono andare oltre i 2000 secondi e all'interno delle finestre la pendenza del modulo dell'errore non supera lo 0.01, i coefficienti di terzo e secondo grado uscenti risultano estremamente piccoli. Al fine di non rischiare di incorrere in problemi nella successiva risoluzione numerica, si trasla l'inizio della finestra in corrispondenza dell'origine dell'asse tempi. Tradotto: nel codice, al vettore delle ascisse, ossia il vettore dei tempi, passato alla funzione `polyfit`, è sottratto il suo primo elemento.

Per trovare la posizione dei flessi del polinomio si considerano i parametri  $a$ ,  $b$  e  $c$  e si moltiplicano rispettivamente per 3, 2 e 1, ottenendo in questo modo la derivata del polinomio descritta in Eq.4.2. I parametri della derivata sono a loro volta utilizzati dalla funzione `risolvi`, che applicando le formule per la risoluzione di un polinomio del secondo ordine (Eq.4.3), restituisce la posizione dei flessi del polinomio di partenza. Queste due soluzioni dovranno essere traslate in avanti di un tempo pari a quello dell'inizio della finestra per riallineare il polinomio con i segnale d'errore, a questo punto i due flessi traslati possono essere confrontati con gli istanti di tempo corrispondenti ai margini della finestra. In base ai risultati di questo confronto si classifica l'andamento secondo la Tabella 4.1.

$$y = 3ax^2 + 2bx + c \quad (4.2)$$

$$y = 0 \iff x_{1,2} = \frac{-2b \pm \sqrt{(2b)^2 - 12ac}}{6a} \quad (4.3)$$

La classe sulla quale si ripone la maggiore attenzione è la classe A, poiché si vogliono individuare zone in cui la tensione tende a discostarsi dal modello, e quindi a far aumentare il modulo dell'errore. Le altre classi identificano zone in cui prevalgono

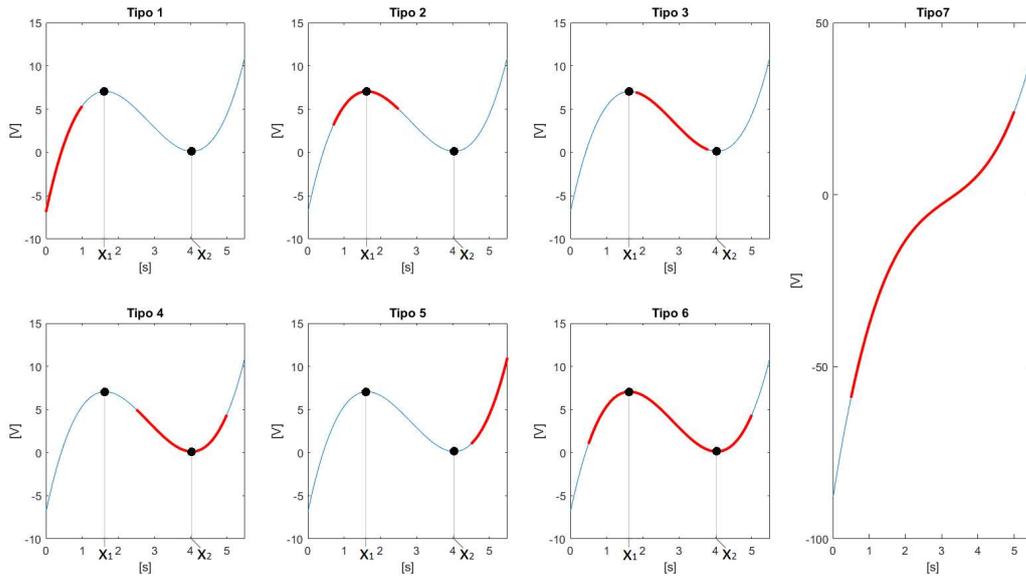


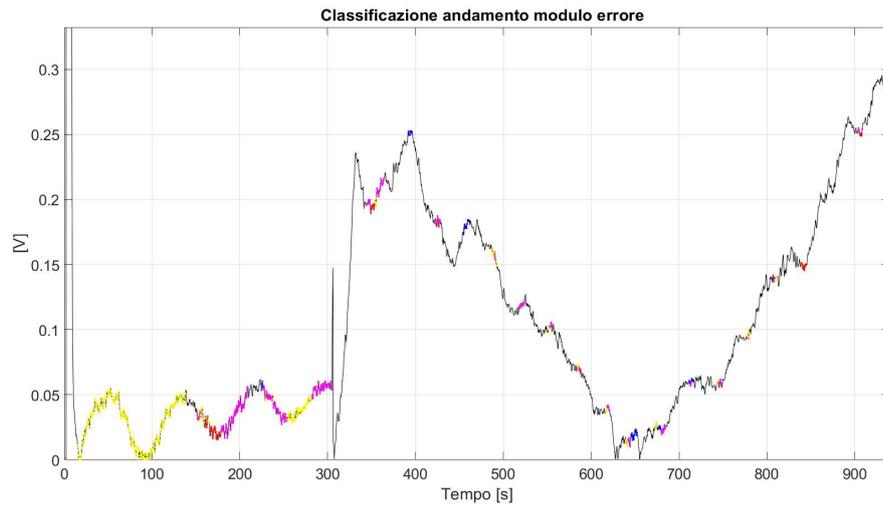
Figura 4.10.: Tipologie di andamenti identificati a seconda della posizione della finestra (rappresentata in rosso) rispetto alle posizioni  $x_1$  e  $x_2$  dei flessi del polinomio

le oscillazioni o nelle quali l'errore sta diminuendo in modulo.

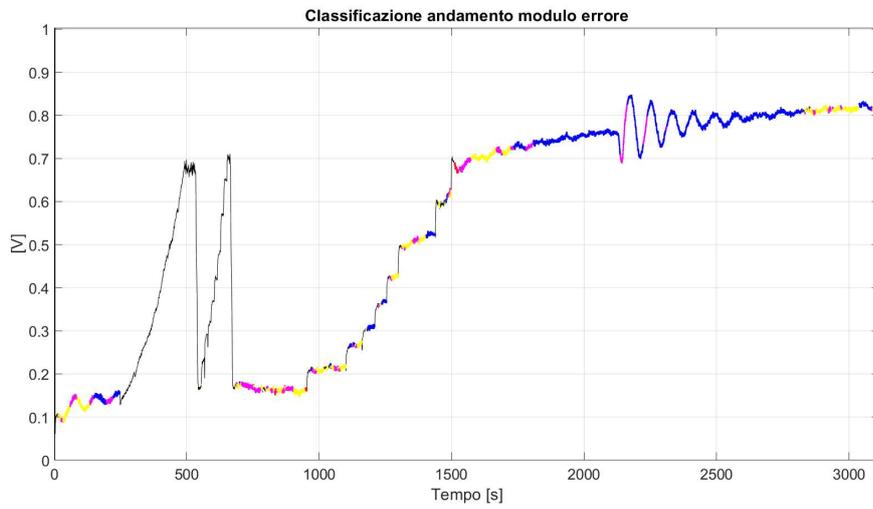
In Figura 4.11 sono riportati i risultati della classificazione sul modulo dell'errore per due dataset, uno BoT e l'altro EoT: la classificazione avviene solamente nelle zone temporali che ricadono all'interno delle finestre discusse nella sezione 4.3.1, sul modulo dell'errore vengono evidenziate con i colori della Tabella 4.1 le 4 classi. Il risultato mostra che nel dataset BoT sono presenti poche zone di carattere "A", al contrario del dataset EoT, questo risultato è analogo anche per gli altri BoT.

**Tempo rimanente al raggiungimento della soglia** Si imposta un valore che il modulo dell'errore è preferibile che non superi (soglia), a questo punto si può andare a stimare quanto tempo impiegherebbe il polinomio ottenuto al passo precedente a raggiungere questa soglia. Per effettuare questa stima si estende il polinomio al di fuori della finestra temporale che si sta esaminando e si calcola l'istante di tempo al quale il polinomio raggiunge il valore della soglia (in Figura 4.12<sup>1</sup> è mostrato un esempio), il tempo residuo sarà dato dalla differenza fra l'istante di tempo appena trovato e il limite superiore della finestra (nonché l'istante di tempo del campione in esame). Per determinare a che istante di tempo il polinomio eguaglia la soglia si risolve l'Eq.4.4 attraverso la funzione `vpsolve` di Matlab [32], in output si ottengono le soluzioni in formato simbolico. Le soluzioni vengono riconvertite in double e necessitano anche in questo caso di essere traslate in avanti per essere riallineate

<sup>1</sup>I valori numerici riportati sugli assi sono arbitrari, l'immagine ha il solo scopo di chiarire il meccanismo che c'è dietro al calcolo effettuato.



(a) Classificazione per dataset BoT (150234)



(b) Classificazione per dataset BoT (111115)

Figura 4.11.: Classificazione delle zone A, B, C e D

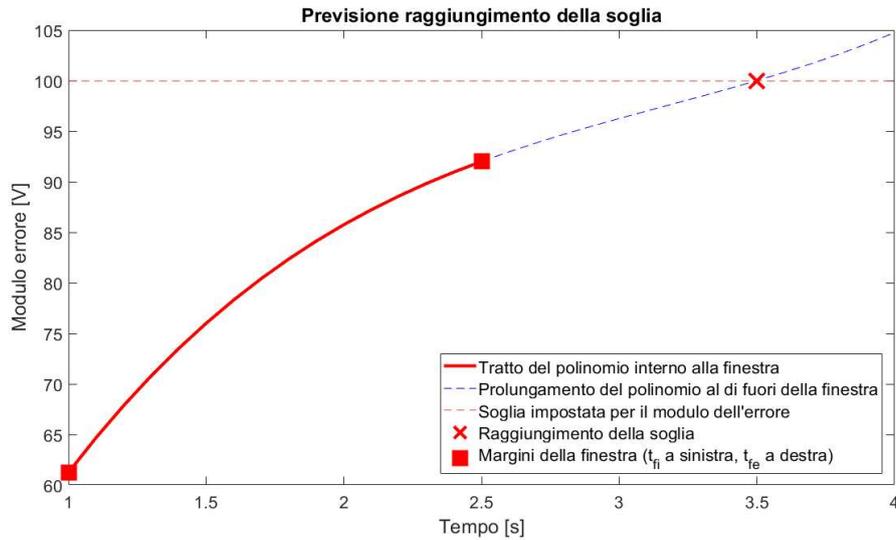


Figura 4.12.: Calcolo del tempo residuo al raggiungimento della soglia

con la finestra che si sta considerando. Le soluzioni, che possono essere multiple, sono considerate solo se maggiori del margine superiore della finestra, altrimenti significherebbe che il modulo dell'errore ha già superato la soglia prima o all'interno della finestra. Nel caso di più soluzioni accettabili si considera solo la più vicina alla finestra e facendo la differenza fra la soluzione selezionata e il margine della finestra si ottiene la stima del tempo che impiegherebbe il polinomio, e quindi il modulo dell'errore, a raggiungere la soglia.

$$ax^3 + bx^2 + cx + d = soglia \quad (4.4)$$

L'Eq.4.5 riporta il calcolo effettuato per ottenere la stima del tempo rimanente al raggiungimento della soglia,  $t_{reach}$  è la soluzione dell'Eq.4.4 più prossima al margine superiore della finestra  $t_{fe}$ . All'interno di questa operazione è compresa anche la traslazione temporale effettuata su  $t_{reach}$  di ampiezza pari al primo istante di tempo della finestra  $t_{fi}$ . Il calcolo non viene effettuato se non si ha alcuna soluzione  $t_{reach}$  che a seguito della traslazione risulta maggiore di  $t_{fe}$ .

$$t_{p3} = t_{reach} - (t_{fe} - t_{fi}) \quad (4.5)$$

La stima viene effettuata solamente in corrispondenza dei punti nei quali la finestra temporale è stata classificata di tipo "A". All'interno della stessa finestra infatti possono esserci punti classificati in modo disomogeneo, poiché, come visto nella sezione 4.3.1, alle finestre vengono aggiunti campioni fino a che gli ingressi si mantengono costanti e ad ogni nuovo punto il polinomio di terzo grado viene ricalcolato. Questo meccanismo fa in modo che, a seconda dell'andamento dell'errore,

il polinomio possa variare le sue caratteristiche all'interno della finestra, ciò si riflette sulla classificazione del suo andamento.

Non si calcola il tempo  $t_{p_3}$  per zone classificate diversamente da "A", perché si avrebbe un modulo dell'errore in diminuzione o con oscillazioni, l'indice  $t_{p_3}$  in questi casi perderebbe di significato e attendibilità.

Nei grafici nelle Figure 4.14 e 4.15, è riportato il modulo dell'errore, sul quale sono evidenziate le classificazioni A, B, C e D (come in Figura 4.11), in aggiunta, dove è stata possibile effettuarla, è presente la stima del tempo  $t_{p_3}$ . La soglia in questo caso è stata impostata a 0.75 per osservare il comportamento dell'indice nel tratto di errore del dataset EoT che parte dai 1500 secondi.

Sempre osservando le Figure 4.14 e 4.15, si colgono le evidenti limitazioni di questo indice dovute alle caratteristiche dei segnali di ingresso: se questi non possiedono dei lunghi periodi a condizione costante, risulta difficile effettuare una previsione accurata. Per esempio, nel caso dei dataset a disposizione, la corrente presenta curve di polarizzazione sia a rampa che a scalino e la temperatura contiene delle oscillazioni, le conseguenze di ciò possono essere:

- Nel caso di scalini o rampe di corrente, la finestra che viene a formarsi subito dopo potrebbe ancora includere una variazione dell'errore dovuta al precedente aumento o decremento della corrente. Questa variazione dell'errore non può essere comparata a quella che si ha ad una condizione di ingresso costante e vuole essere evitata. Per limitare questo fenomeno si impostano i parametri delay e coda nella fase di creazione delle finestre (Sezione 4.3.1).
- Nel caso di oscillazioni della temperatura, si forma uno sfasamento fra la tensione del modello e quella misurata che introduce un'oscillazione dell'errore. Questa oscillazione non è dovuta a sintomi di degradazione, pertanto introduce un fattore di disturbo nella stima di  $t_{p_3}$ .

In entrambi i casi descritti la stima di  $t_{p_3}$  è resa poco attendibile, se ad esempio si decide una soglia di tempo minimo oltre la quale il tempo  $t_{p_3}$  non deve scendere, si potrebbero generare dei falsi allarmi. Un esempio può essere quello in Figura 4.14a nella quale un'oscillazione del modulo dell'errore all'inizio della finestra fa scendere  $t_{p_3}$  a 15 secondi, ma esaminando quella porzione più in dettaglio in Figura 4.14b, si nota che la zona classificata di tipo "A" sulla quale si calcola l'indice è più breve della stima stessa e la medesima cosa si ripete dopo circa 60 secondi, nel successivo tratto di errore classificato come "A". Si potrebbe quindi impostare un intervallo di tempo da attendere prima di convalidare l'attendibilità di  $t_{p_3}$ .

In Figura 4.15c, nella quale si sta effettivamente superando la soglia impostata, nel tratto precedente al superamento,  $t_{p_3}$  decresce fino a che non arriva in prossimità dello zero, una volta superata la soglia il calcolo dell'indice è sospeso. Si nota anche che c'è una larga porzione classificata come "A" fra i 1850 e i 1920 secondi nella quale la stima non avviene, succede questo perché la pendenza del modulo dell'errore sta

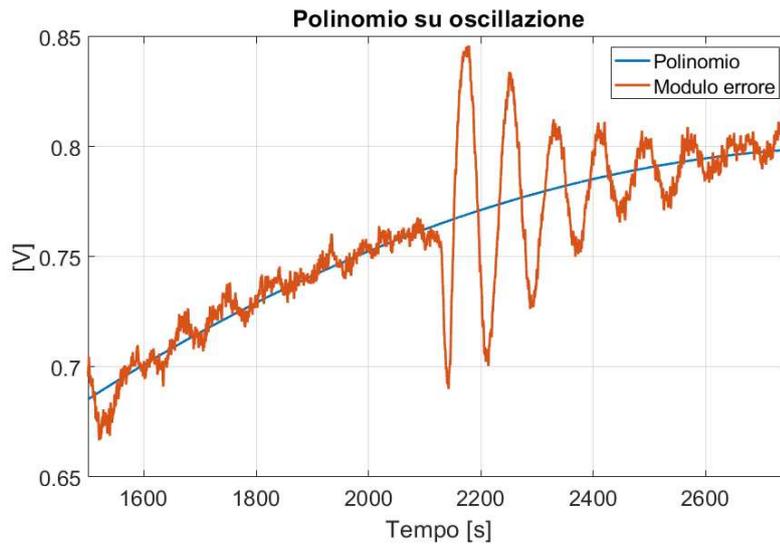


Figura 4.13.: Polinomio di terzo grado che approssima l'andamento del modulo dell'errore in una finestra in cui sono presenti oscillazioni

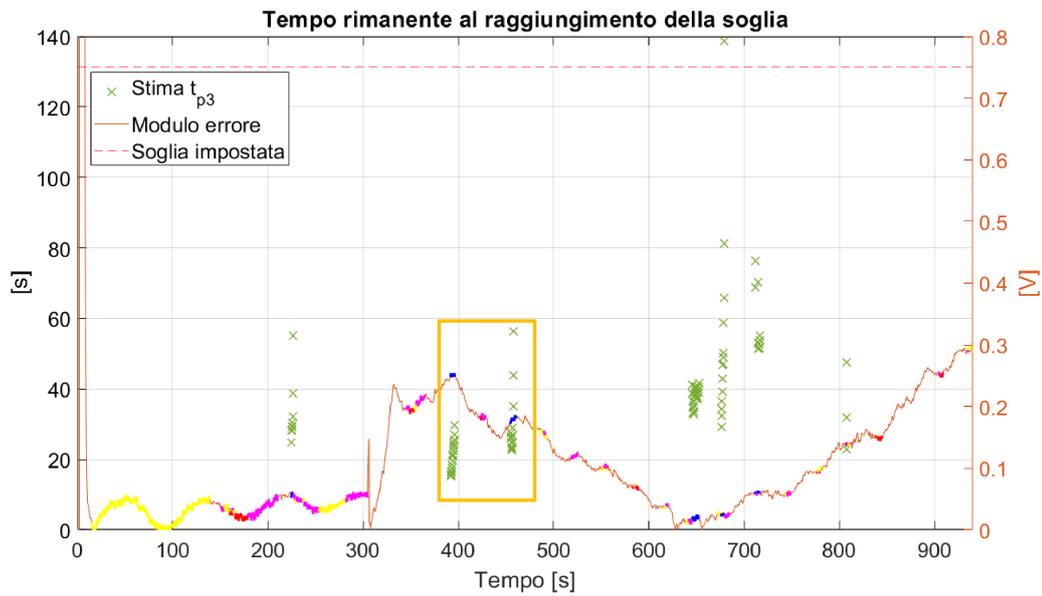
decrescendo e l'estensione del polinomio oltre la finestra non raggiungere la soglia impostata a 0.75. Questa caratteristica potrebbe essere fonte di mancati allarmi in particolari condizioni.

Nella stima di  $t_{p3}$  l'oscillazione della temperatura presente fra i 2000 e i 2500 secondi è in parte filtrata dal polinomio, che tenendo in considerazione tutti i campioni precedenti contenuti nella finestra, attenua l'effetto delle oscillazioni. Se però la finestra non comprende molti campioni, il polinomio sarebbe influenzato maggiormente e non approssimerebbe con precisione il reale andamento del modulo dell'errore. In Figura 4.13 è presente il comportamento del polinomio all'interno di una finestra in cui l'oscillazione compare dopo più di 500 secondi della formazione della finestra, in questo caso l'effetto dell'oscillazione è stato filtrato.

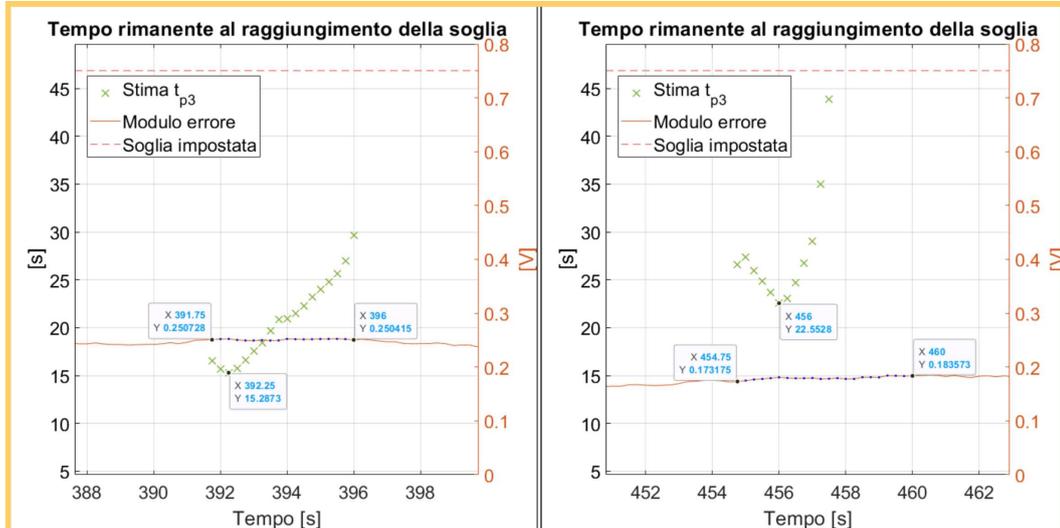
Una tecnica con la quale andare a valutare lo stato di salute dell'elettrolizzatore che non risenta dalle problematiche appena descritte può essere quella della stima parametrica trattata nella Sezione 4.3.3.

### 4.3.3. Stima parametrica inversa

Per avere un'informazione sullo stato di salute dell'elettrolizzatore si vuole osservare come varia il valore assunto dai parametri nel corso della vita del dispositivo. La degradazione dei componenti causa un aumento delle sovratensioni dell'elettrolizzatore, in particolare, la degradazione della membrana può portare ad un aumento della sua resistività, dipendente dal parametro  $\lambda$ , la degradazione del catalizzatore invece rende necessaria maggiore energia per l'innescio della reazione, ne consegue un aumento della sovratensione di attivazione, dipendente dai coefficienti  $\alpha_a$ ,  $\alpha_c$ ,  $i_{0,a}$  e  $i_{0,c}$  [30]. Presi questi cinque parametri, si può confrontare il valore che assumono

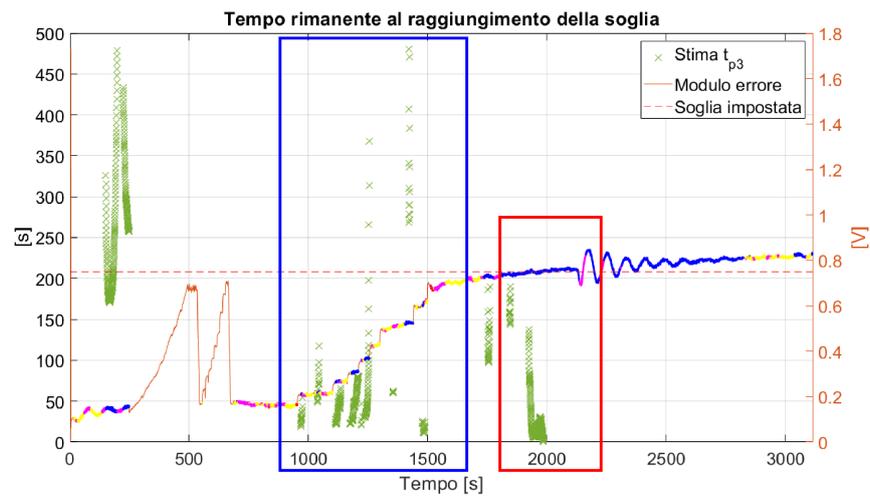


(a) Stima  $t_{p3}$  su dataset EoT 111115

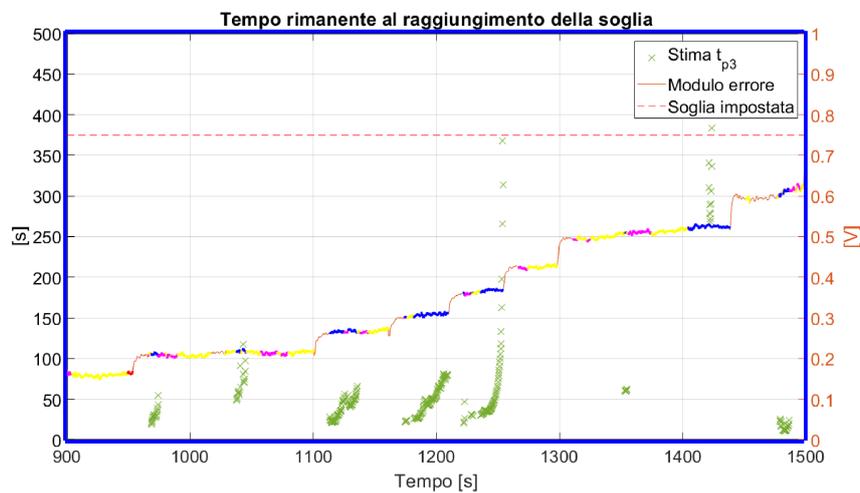


(b) Dettaglio della stima riquadrata nella sotto-figura a

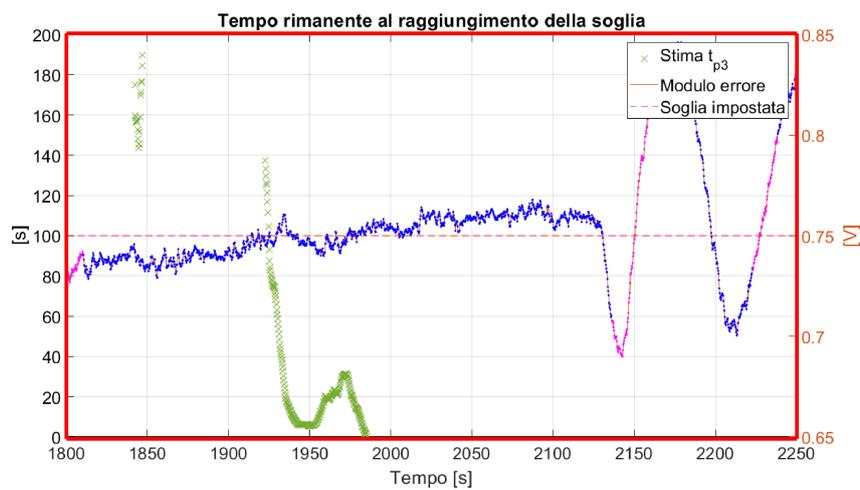
Figura 4.14.: Stima del tempo residuo al raggiungimento della soglia per il dataset BoT 150234



(a) Stima  $t_{p3}$  su dataset EoT 111115



(b) Dettaglio della stima durante una salita a scalino della corrente



(c) Dettaglio della stima in prossimità del raggiungimento della soglia

Figura 4.15.: Stima del tempo residuo al raggiungimento della soglia per il dataset EoT 111115

in un caso BoT con la stima del loro valore ottenuta a partire dalle misurazioni del dataset esaminato.

La stima è stata effettuata per un parametro alla volta, fissando gli altri 4 al valore assunto in un caso BoT, successivamente a partire dalla misurazione di tensione presente in un dataset, si sono invertite e applicate le equazioni viste nella sezione 2.2.2. Come riportato nell'Eq.4.6, la stima del parametro ( $p_{st}$ ) viene sottratta al valore medio del parametro nel caso BoT ( $\bar{p}_{BoT}$ ) e si divide questa differenza sempre per  $\bar{p}_{BoT}$ , così facendo si ottiene la variazione del parametro normalizzata rispetto alle condizioni BoT ( $\delta_{\%}$ ).

$$\delta_{\%} = \frac{\bar{p}_{BoT} - p_{st}}{\bar{p}_{BoT}} \quad (4.6)$$

Quello che ci si aspetta è che in un dataset in cui è presente degradazione i cinque parametri risultino inferiori a quelli della media, poiché per tutti e cinque una diminuzione comporta un aumento della tensione (Eq.2.13, Eq.2.15). Quanto appena detto si traduce in un  $\delta_{\%}$  positivo per un elettrolizzatore degradato, tanto più grande quanto è il grado della degradazione.

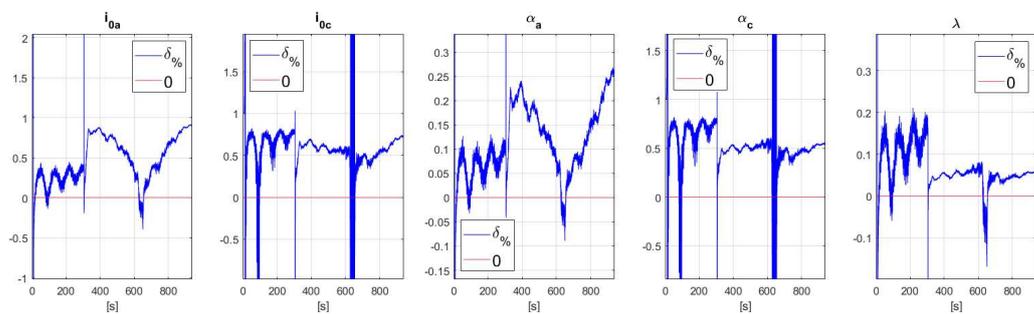
Passando ai risultati (Figure 4.16 e 4.17), si osserva che il  $\delta_{\%}$  dei parametri non rimane costante nel tempo, ma varia a seconda degli ingressi presenti nel dataset, fra le cause di ciò c'è il metodo con cui avviene la stima dei parametri. Andando a stimare i parametri singolarmente, considerando i restanti al valore assunto nel caso BoT, il parametro preso in considerazione deve compensare anche l'effetto dei restanti, che non hanno però la stessa influenza nelle diverse zone di lavoro dell'elettrolizzatore. Prendendo come esempio le perdite ohmiche e quelle di attivazione, all'aumentare della corrente, si ha un andamento lineare nel primo caso e logaritmico nel secondo.

Per i quattro dataset BoT, i  $\delta_{\%}$  dei cinque parametri (Figura 4.16) assumono in alcuni casi valori molto vicino allo 0, mentre in altri si discostano maggiormente, ma pur sempre mantenendone uno in prossimità dello 0. Sempre nel caso dei dataset BoT, i  $\delta_{\%}$  assumono sia valori negativi che positivi, mentre per i dataset EoT<sup>2</sup> il valore del  $\delta_{\%}$  rimane sempre positivo per tutti i parametri, e nella maggior parte dei casi risulta maggiore rispetto ai valori del caso BoT (Figura 4.17).

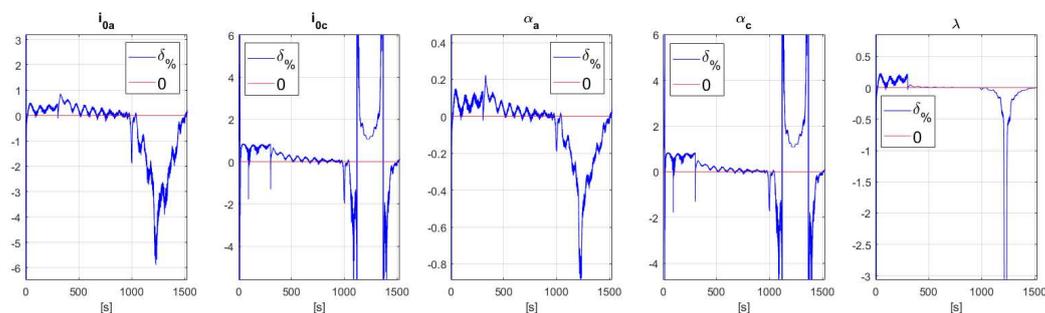
In Tabella 4.2 sono riportati, per ogni dataset, la media dell  $\delta_{\%}$  di ogni parametro e la media dei 5 risultati. Si osserva che il  $\delta_{\%}$  per i parametri del dataset EoT è sempre maggiore rispetto a quello dei dataset BoT, questo conferma l'ipotesi di un elettrolizzatore, che degradandosi, ha subito un aumento delle perdite con conseguente aumento delle sovratensioni.

---

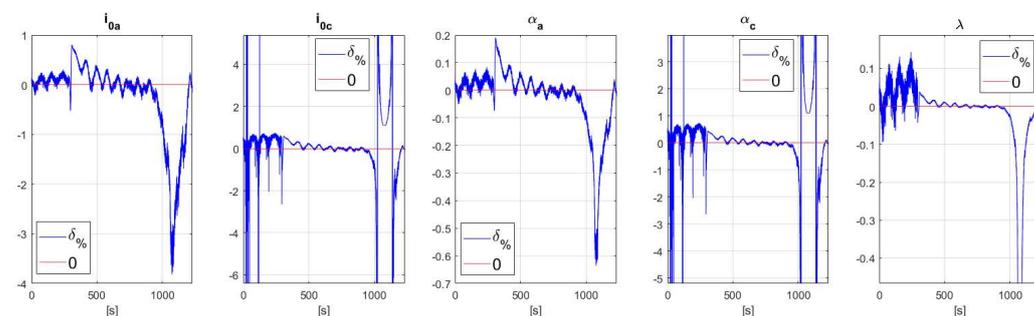
<sup>2</sup>Per avere un ulteriore riscontro è stato preso un altro dataset EoT, il 121203, acquisito qualche giorno dopo del 111115. La differenza principale fra i due dataset EoT è la tipologia di test che è stata effettuata con essi, nel 121203 l'elettrolizzatore non è sottoposto ad un lungo periodo a corrente costante come avviene nel 111115



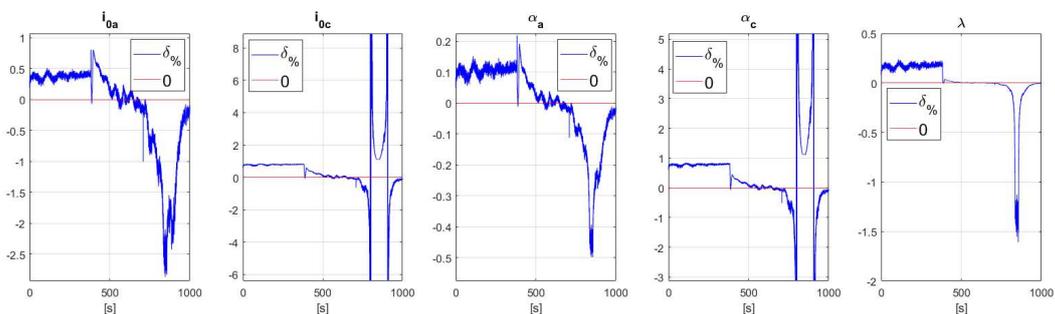
(a) Scostamento normalizzato per il dataset BoT 150234



(b) Scostamento normalizzato per il dataset BoT 090817



(c) Scostamento normalizzato per il dataset BoT 175526

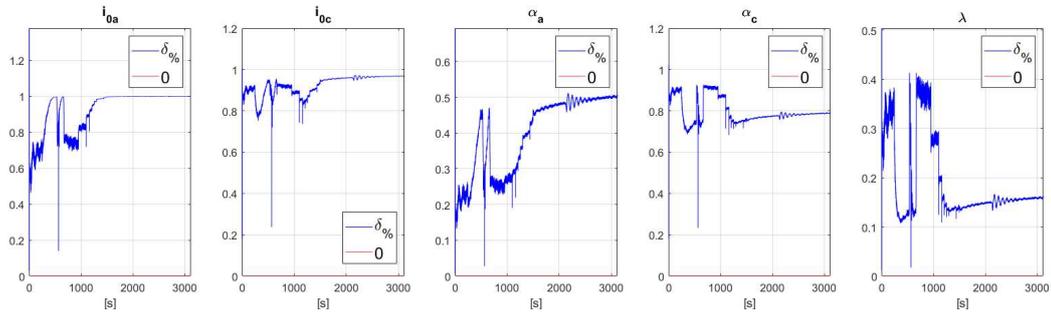


(d) Scostamento normalizzato per il dataset BoT 104217

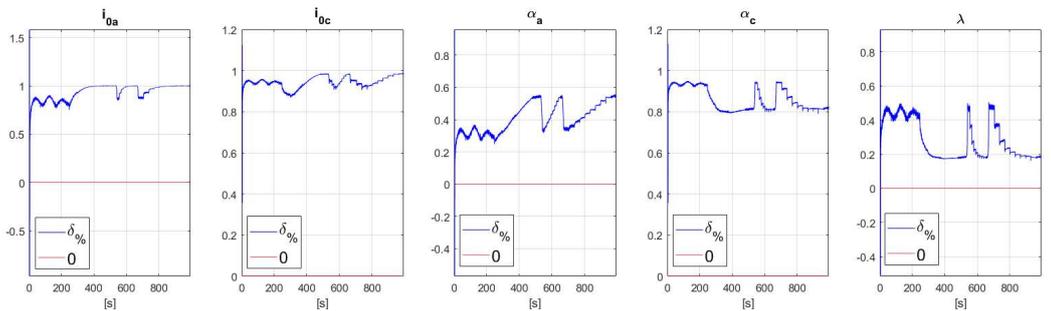
Figura 4.16.: Scostamenti normalizzati per i dataset BoT

Tabella 4.2.: Scostamento normalizzato dei 5 parametri

| Dataset |               | $\delta\%$ |           |            |            |           | Media          |
|---------|---------------|------------|-----------|------------|------------|-----------|----------------|
|         |               | $i_{0,a}$  | $i_{0,c}$ | $\alpha_a$ | $\alpha_c$ | $\lambda$ |                |
| BoT     | <b>104217</b> | -0.1027    | 0.6929    | 0.0079     | 0.6871     | 0.0207    | <b>0.2612</b>  |
|         | <b>150234</b> | 0.5050     | 0.5731    | 0.12222    | 0.5226     | 0.0621    | <b>0.3570</b>  |
|         | <b>090817</b> | -0.4801    | 0.4425    | -0.0428    | 0.4397     | -0.1103   | <b>0.0498</b>  |
|         | <b>175526</b> | -0.2282    | 0.1264    | -0.0254    | 0.1276     | -0.0457   | <b>-0.0091</b> |
| EoT     | <b>111115</b> | 0.9253     | 0.9264    | 0.3973     | 0.7974     | 0.1901    | <b>0.6473</b>  |
|         | <b>121203</b> | 0.9303     | 0.9450    | 0.4089     | 0.8592     | 0.2818    | <b>0.6851</b>  |



(a) Scostamento normalizzato per il dataset EoT 111115



(b) Scostamento normalizzato per il dataset EoT 121203

Figura 4.17.: Scostamenti normalizzati per i dataset EoT



## Capitolo 5.

# Conclusioni e sviluppi futuri

### 5.1. Conclusioni

Il modello sviluppato, tramite la selezione delle equazioni e la calibrazione dei parametri, è stato in grado di approssimare fedelmente il funzionamento dell'elettrolizzatore a disposizione. Questo fa supporre che questa elasticità si possa estendere anche agli altri elettrolizzatori a membrana polimerica. Per avere successo, nella fase di calibrazione è importante conoscere gli intervalli nei quali si vanno a ricercare i parametri per trovare la loro combinazione ottima. Questa ricerca è stata affidata all'algoritmo di PSO, si è notato come alcuni parametri tendessero a divergere, per cui l'algoritmo è stato diviso in tre fasi.

Ottenere un modello allineato ad un componente in condizioni di salute offre il vantaggio di poter confrontare come, al trascorrere del tempo, il comportamento dell'elettrolizzatore si discosti da quello iniziale. A tal proposito si sono selezionati quattro dataset acquisiti su un elettrolizzatore appena installato e in salute, su questi si è effettuata la calibrazione dei parametri ed è stata calcolata la media dei risultati nei quattro casi, così da ottenere dei parametri di riferimento per un elettrolizzatore ad inizio vita. In questa fase di selezione dei dataset, si è osservato che le misurazioni effettuate sulla produzione dei gas contenevano degli errori, per questa ragione è stato deciso di proseguire tenendo in considerazione solamente gli aspetti legati alla tensione. È stato in seguito considerato un dataset acquisito poco tempo prima della rottura della membrana, nel quale la tensione, a fronte di ingressi costanti, tende ad aumentare: supponendo che questo potesse essere il sintomo di un logoramento, sono stati passati gli ingressi di questo dataset al modello, facente utilizzo della media dei parametri ottenuti in precedenza. Dalla differenza fra la tensione del modello e quella del dataset si è ottenuto l'errore e nel suo modulo si sono cercate delle informazioni che potessero notificare l'imminenza di un possibile guasto o rilevare lo stato della degradazione.

Al fine di ottenere queste informazioni, si è scelto di valutare il modulo dell'errore in zone in cui i segnali di ingresso risultassero costanti, così da evidenziare delle variazioni non dovute dagli ingressi, ma dalla degradazione dei componenti. Quindi si sono selezionate delle finestre temporali in cui gli ingressi del dataset si mantengono costanti, e all'interno di queste si è calcolato il polinomio di terzo grado che approssima

meglio l'andamento del modulo dell'errore contenuto in quell'intervallo temporale. Nelle finestre temporali in cui il polinomio approssima il modulo dell'errore con un andamento identificato come strettamente crescente, si è calcolato il tempo che il polinomio avrebbe impiegato a raggiungere un valore di soglia preimpostato. Questo indice, indicato con  $t_{p3}$ , ha mostrato una maggiore affidabilità per condizioni di ingresso stabili in un lungo periodo, condizioni spesso non verificate per la tipologia di dataset a disposizione.

Per ottenere un'informazione sulla degradazione che coinvolga tutte le zone del dataset si è effettuata una stima parametrica a partire dalle misurazioni presenti nel dataset. Il valore stimato dei parametri è stato confrontato con la media dei valori che questi assumono nei quattro casi di inizio vita e si è osservato che nel caso degradato i parametri tendono ad assumere valori più bassi. Questi risultati sono in linea con quanto riportato in letteratura: a fronte di una diminuzione dei parametri  $\alpha_a$ ,  $\alpha_c$ ,  $i_{0,a}$ ,  $i_{0,c}$  e  $\lambda$ , utilizzati per la tensione, corrispondono dei fenomeni di degradazione dell'elettrolizzatore e un conseguente aumento delle perdite, alle quali corrisponde un aumento di tensione e quindi di energia spesa.

## 5.2. Sviluppi futuri

In questo studio i parametri analizzati per la tensione sono stati: le due densità di corrente di scambio, i due coefficienti di trasferimento di carica e il contenuto di acqua alla membrana. Mentre per la produzione di idrogeno: la costante tempo, il guadagno e l'offset della funzione di trasferimento. In uno sviluppo futuro, per allineare il modello, si potrebbero prendere in considerazione altri parametri o selezionare delle equazioni che in questo caso non sono state utilizzate. Per esempio, con l'equazione di Arrhenius (Eq.2.16), le  $i_0$  impostate come parametro assumono il ruolo delle  $i_{0,ref}$  ed entra in gioco il parametro  $E_{act}$ , con questa equazione le correnti di scambio non sono più considerate come costanti, ma variano con la temperatura. Altre due equazioni di cui non si è fatto utilizzo sono l'Eq.2.14 e l'Eq.2.17, che a loro volta coinvolgono dei parametri che richiedono una conoscenza più approfondita dell'elettrolizzatore. L'utilizzo di altre equazioni e parametri può garantire l'ottenimento di un modello più preciso, andando a modellare aspetti che in questo studio sono stati approssimati.

La modellazione della produzione dei gas può essere migliorata attraverso l'aggiunta di una funzione di trasferimento sulla produzione di ossigeno, come è stato fatto per l'idrogeno. Le due funzioni di trasferimento debbono avere coefficienti separati, dato che i due gas subiscono processi di deumidificazione differenti. Questo aspetto è stato tralasciato per prioritizzare la modellazione della tensione, dato che nella sezione 4.3 con la Figura 2.2.2, si è determinato di non avere a disposizione misurazioni consistenti di idrogeno e ossigeno per basare uno studio diagnostico. Tuttavia, dataset successivi contengono misurazioni nelle quali i sensori sono stati ricalibrati, se anche gli errori di misurazione legati al controllo di pressione fossero risolti, si potrebbe procedere con lo studio dell'impatto della degradazione sulla produzione dei gas. Ciò

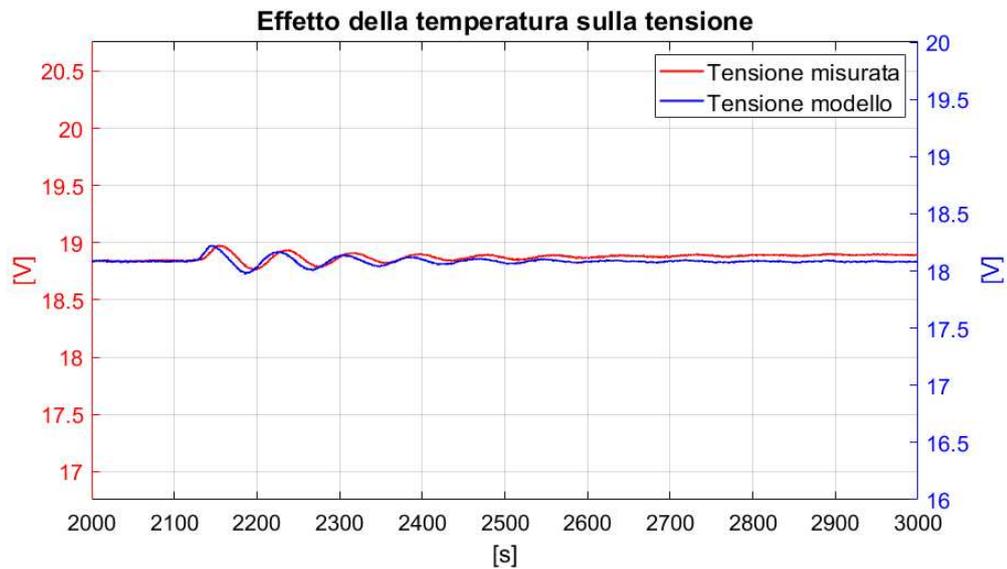


Figura 5.1.: Sfasamento effetti temperatura

che ci si può aspettare è un calo del flusso di idrogeno in uscita, lo stesso varrebbe per il flusso dell'ossigeno, che però potrebbe subire un calo minore a causa del cross-leak dell'idrogeno verso l'anodo.

Sull'errore del dataset EoT 111115 (Figura 4.7), sono ben visibili delle oscillazioni sul modulo dell'errore: queste sono principalmente dovute dallo sfasamento presente fra il segnale di tensione del modello e quello presente nel dataset (visibile in dettaglio a Figura 5.1). Ciò avviene perché la temperatura è misurata sull'acqua in ingresso all'anodo, ma affinché questa influenzi le tensioni dell'elettrolizzatore, anche le componenti del dispositivo devono essere riscaldate, questa dinamica però non è presente all'interno del modello. Le variazioni di temperatura vengono dunque trasferite istantaneamente sulle equazioni per il calcolo della tensione, questo crea un anticipo del segnale del modello rispetto a quello del dataset. Questo sfasamento provoca delle oscillazioni nella differenza dei due segnali (l'errore) e dal momento che queste sono dovute ad un'imperfezione del modello, e non da effetti dovuti alla degradazione, è preferibile che non siano presenti. Per eliminare queste oscillazioni si potrebbe aggiungere una funzione di trasferimento che vada ad introdurre un ritardo nelle variazioni di temperatura in ingresso, così che tramite la calibrazione dei parametri si possa minimizzare lo sfasamento fra i due segnali di tensione.

Le prossime acquisizioni, allo scopo di poter effettuare analisi nel campo della frequenza, saranno effettuate con un campionamento più veloce. In questo modo si potranno tentare degli approcci, che analizzando lo spettro del segnale, possano estrarre dei comportamenti distintivi di un elettrolizzatore ad inizio vita e di uno a fine vita.



# Appendice A.

## Codice Matlab

### A.1. Importazione dei dati dal dataset

La funzione EstraiDati qui riportata, è utilizzata nel codice per il caricamento dei dati dal workspace a Simulink visto nella sezione 3.2 e nei codici a seguire nelle appendici A.2, A.5 e A.7.

---

```
%% FUNZIONE PER L'ESTRAZIONE DEI DATI DAL DATASET
2 function [time, Temperature, NL, Pressioni, Tensione, Corrente]=...
    EstraiDati(DataSet)
4
M=readmatrix(DataSet);
6 time=M(:,1);
Temperature=M(:,1:3);
8 NL=M(:,[1 4 5]);
Pressioni=M(:,[1 6 7]);
10 Tensione=M(:,[1 8]);
Corrente=M(:,[1 9]);
```

---

### A.2. Formazione delle finestre

A seguire è riportato il codice per la formazione delle finestre e il calcolo degli indici di cui si parla nella sezione 4.3.1. Gli indici calcolati con la funzione Calcolo\_indici\_online presente al punto A.4. Per eliminare parte del rumore presente nei segnali si sono filtrati i campioni facendo la media di un numero di campioni pari a quelli indicati dal vettore sampleFilter, i suoi 3 elementi sono utilizzati rispettivamente per: corrente, temperatura dell'anodo ed errore. La variabile sogliaModulo indica la soglia che si imposta per il modulo dell'errore, con la quale si va a calcolare  $t_{p3}$ .

In questo codice la stima parametrica è stata effettuata solamente nelle zone identificate come "A", la stima per tutto l'intervallo temporale è stata effettuata nel codice riportato in A.5.

---

```
1 %% SELEZIONE FINESTRE %%
% Pulizia ambiente
3 clc
clear
```

## Appendice A. Codice Matlab

```
5 close all

7 %% Selezione del dataset
  EoT='DataLog_104217.xlsx';

9
  %% Inizializzazione parametri
11 sizeF=4;
  delay=[48 8];
13 maxSizeF=Inf;

15 sogliaDev=[0.1 Inf];
  sogliaPend=[0.4 0.8];
17 sogliaMod=[0.15 Inf];

19 sogliaModulo=0.75;

21 coda=0.9;          % numero >=0 e <=1
  fuoriMedia=2;     % primi&ultimi N camp. confrontati con la media
23                  % (deve essere minore della meta' di sizeF)
  sampleFilter=[4 4 4]; % impostare a 0 se non si vuole filtrare

25
  Area=67.9*0.0001; % [m2]
27 NumCelle=9;
  spessore=0.0002; % [m]
29 %%
  % Esecuzione modello BoT
31 load('Parametri\Parametri_media.mat')

33 % Caricamento dati da Excel per Simulink EoT
  [time, Temperature, NL, Pressioni, Tensione, Corrente] = ...
35   EstraiDati(EoT);
  % conversioni unita' di misura
37 Temperature_K=Temperature+273.15;
  Pressioni_Pa=(Pressioni+1.01325)*1e5;
39 i0a_m=i0a*1e4; % conversione da A/cm2 a A/m2
  i0c_m=i0c*1e4;

41
  % Estrazione dati per creazione finestre
43 IEOt=Corrente(:,2);
  TEOt=Temperature(:,2);
45 n = length(IEOt);

47 IfEOt=filtra_segnaled(IEOt, sampleFilter(1));
  TfEOt=filtra_segnaled(TEOt, sampleFilter(2));
49 outEOt=sim("PEM_PSO_v006.slx");
  ErroreEOt=outEOt.Errore.Data;

51
```

```

%%
53 Finestra=[];
   timeFinestra=[];
55 started=0; suc=[0 0]; andamento=[];Pendenza=[];
   CronologiaFinestre=[];Monotonicita=[];
57 MonotonicitaPol3=[];Parabola=[];TS=[];
   TP=[];TP3=[];ParametriStimati=[];syms x
59
   contatore=-1;
61 for k=1:n
   if k>sampleFilter(3)
63     Errore(k)=abs(mean(ErroreEoT(k-sampleFilter(3):k)));
   else
65     Errore(k)=abs(mean(ErroreEoT(k)));
   end
67 [LF,~]=size(Finestra);
   if LF>=sizeF-1
69     Finestra=[Finestra; IfEoT(k) TfEoT(k)];
   timeFinestra=[timeFinestra time(k)];
71 [Finestra,suc,started,timeFinestra]=...
   aggiornaFinestra(suc,started,timeFinestra,delay,...
73     sogliaDev,sogliaPend,sogliaMod,fuoriMedia,coda,...
   Finestra,sizeF,maxSizeF);
75 else
   Finestra=[Finestra;IfEoT(k) TfEoT(k)];
77 timeFinestra=[timeFinestra time(k)];
   end
79 if started==true
   if contatore===-1
81     contatore=0;
   else
83     contatore=min(contatore+1,maxSizeF);
   end
85 [LF,~]=size(Finestra);
   ErroreFinestra=Errore(end-LF+1:end);
87 [Pol1, Pol2, Pol3, Mono]=...
   Calcolo_indici_online_v03(ErroreFinestra,...
89     timeFinestra);

91 % elaborazione dati
   % rilevazione andamento
93 d_Pol3=[3 2 1].*Pol3(1:3);
   [x1,x2]=risolvi(d_Pol3);
95 t=timeFinestra-timeFinestra(1);
   I = t(end)<x1;
97 II = t(1)<x1 && t(end)>x1 && t(end)<x2;
   III = t(1)>x1 && t(end)<x2;

```

Appendice A. Codice Matlab

```

99     IV = t(1)>x1 && t(1)<x2 &&t(end)>x2;
      V  = t(1)>x2;
101    VI = t(1)<x1 && t(end)>x2;
      VII = isnan(x1);
103    zona=[I,II,III,IV,V,VI,VII];
      ABCD=[((I||V||VII)&& Pol3(1)>0)||((III && Pol3(1)<0),...
105          ((I| V||VII)&& Pol3(1)<0)||((III && Pol3(1)>0),...
          II || IV ...
107          VI)];

109    % fintraggio nelle zone A
      if ABCD(1)==0
111          m=NaN;
          Sqr=NaN;
113          MonoPol3=NaN;
          ts=NaN;
115          tp=NaN;
          tp3=NaN;
117          stima=NaN(1,5);
      else
119          m=Pol1(1);
          Sqr=Pol2(1);
121          C={[(timeFinestra-timeFinestra(1))'...
                polyval(Pol3,timeFinestra-timeFinestra(1))' ]};
123          MonoPol3=monotonicity(C);
          ts=(sogliaModulo-ErroreFinestra(end))/m;
125          Delta=Pol2(2)^2-4*Pol2(1)*(Pol2(3)-sogliaModulo);
          if Delta>=0
127              xs=(-Pol2(2)+sqrt(Delta))/(2*Pol2(1));
              tp=timeFinestra(1)+xs-timeFinestra(end);
129          else
              tp=NaN;
131          end
          if polyval(Pol3,timeFinestra(end)-...
133              timeFinestra(1)) > sogliaModulo
              tp3=NaN;
135          else
              sol=vpasolve(Pol3(1)*x^3+...
137                  Pol3(2)*x^2+Pol3(3)*x+Pol3(4)...
                  == sogliaModulo,[0 Inf]);
139              if isempty(sol)
                  tp3=NaN;
141              else
                  tempo_superamento=...
143                  double(sol)+timeFinestra(1);
                  sol_val=find(tempo_superamento>=...
145                  timeFinestra(1));

```

## A.2. Formazione delle finestre

```

147         tp3=min(tempo_superamento(sol_val))-...
            timeFinestra(end);
149     end
151     end
153     itc=timeFinestra(end)*4+1;
155     [stima(1),stima(2),stima(3),stima(4),stima(5)]=...
        inverse(Tensione(itc,2),Corrente(itc,2),...
157         Area,NumCelle,spessore,alpha_a,alpha_c,...
159         i0a_m,i0c_m,lambda,Temperature_K(itc,2:3),...
161         Pressioni_Pa(itc,2:3));
163     end
165     %salvataggio dati
167     andamento=[andamento;time(k),ABCD];
169     Pendenza=[Pendenza;time(k),m];
171     CronologiaFinestre=...
        [CronologiaFinestre;timeFinestra(1),...
173         timeFinestra(1)*4+1 , k];
175     Monotonicita=[Monotonicita; time(k), Mono];
177     MonotonicitaPol3=[MonotonicitaPol3; time(k), MonoPol3];
179     Parabola=[Parabola; time(k) Sqr];
181     TS=[TS; time(k) ts];
183     TP=[TP; time(k) tp];
185     TP3=[TP3; time(k) tp3];
187     ParametriStimati=[ParametriStimati; time(k) stima];
189     else
191         contatore=-1;
193     end
195 end
197 %% Compressione
199 CronologiaCompatta=CronologiaFinestre(1,:);
201 cci=1;
203 for i=2:length(CronologiaFinestre)
205     if CronologiaFinestre(i,2)~=CronologiaFinestre(i-1,2)
207         CronologiaCompatta(cci,3)=CronologiaFinestre(i-1,3);
209         CronologiaCompatta=[CronologiaCompatta;...
211             CronologiaFinestre(i,:)];
213         cci=cci+1;
215     end
217 end
219 CronologiaCompatta(end,3)=CronologiaFinestre(end,3);

```

---

### A.3. Aggiornamento della finestra

Funzione utilizzata nel codice per la formazione delle finestre per effettuare le valutazioni sui campioni selezionati e aggiornare i vettori di tempo e modulo dell'errore che rappresentano la finestra attualmente considerata.

---

```

1      %% FUNZIONE PER L'AGGIORNAMENTO DELLE FINESTRE
function [Fout, sucNew, startedNew, timeNew]=...
3          aggiornaFinestra(sucOld,startedOld,timeOld,...
        delay,sogliaDev,sogliaPend,sogliaMod,...
5          fuoriMedia,coda,Fin,sizeF,maxSize)
    I=Fin(:,1);
7    T=Fin(:,2);

9    pol_I=polyfit(timeOld,I,1);
    pol_T=polyfit(timeOld,T,1);
11   pend_I=abs(pol_I(1));
    pend_T=abs(pol_T(1));
13
    step=...
15     [abs(mean(I(end-fuoriMedia+1:end))-...
        mean(I(1:end-fuoriMedia)))...
17     abs(mean(I(1:fuoriMedia))-...
        mean(I(fuoriMedia+1:end)))];...
19     abs(mean(T(end-fuoriMedia+1:end))-...
        mean(T(1:end-fuoriMedia)))...
21     abs(mean(T(1:fuoriMedia))-...
        mean(T(1:end-fuoriMedia-1)))];

23
    soglieOK = [std(I)          <  sogliaDev(1)    &&  ...
25                 pend_I      <  sogliaPend(1)  &&  ...
                max(step(1,:)) <  sogliaMod(1)    ,   ...
27                 std(T)      <  sogliaDev(2)    &&  ...
                pend_T        <  sogliaPend(2)  &&  ...
29                 max(step(2,:)) <  sogliaMod(2)];

31   first_suc=not(sucOld) & soglieOK;
    sucNew=(sucOld+soglieOK + first_suc*(length(I)-1)).*soglieOK;
33
    if isequal( sucNew>=delay , ones(1,2))
35         if startedOld==false
                Fout=[I(end-round(min(sucNew)*coda)+1:end) , ...
37                 T(end-round(min(sucNew)*coda)+1:end)];
                timeNew=timeOld(end-round(min(sucNew)*coda)+1:end);
39                 startedNew=true;
        else
41                 if length(I)<=maxSize
                        Fout=[I , T];

```

```

43         timeNew=timeOld;
         startedNew=startedOld;
45     else
         Fout=[I(2:end) , T(2:end)];
47         timeNew=timeOld(2:end);
         end
49         startedNew=startedOld;

51     end
else
53     if startedOld==true
         Fout=[I(end) , T(end)];
55         timeNew=timeOld(end);
         else
57         if min(sucNew)==0
                 Fout=[I(end-sizeF+2:end) , T(end-sizeF+2:end)];
59                 timeNew=timeOld(end-sizeF+2:end);
                 else
61                 Fout=[I , T];
                 timeNew=timeOld;
63         end
         end
65         startedNew=false;
end

```

---

## A.4. Calcolo degli indici sulla finestra

La funzione `Calcolo_indici_online_v03` è utilizzata per il calcolo degli indici sulla finestra di errore passata in ingresso, in questa versione della funzione si calcolano solamente i polinomi di primo, secondo e terzo grado e la monotonicità, la varianza è stata rimossa per cessato utilizzo.

---

```

        %% FUNZIONE CALCOLO INDICI SU FINESTRA %%
2     function [pol1, pol2, pol3, mono] =...
           Calcolo_indici_online_v03(Errore, time)
4
   % in questa versione si calcolano i polinomi di
6 % I, II e III grado e la monotonicita' all'interno
   % della finestra passata in ingresso
8
   % polinomio di primo grado
10  pol1=polyfit(time-time(1),Errore,1);
   % polinomio di primo grado
12  pol2=polyfit(time-time(1),Errore,2);
   % polinomio di terzo grado
14  pol3=polyfit(time-time(1),Errore,3);

```

```

16
    % monotonicita'
18 if time(end)-time(1)>=0
        mono=table2array(monotonicity(table(time','Errore')));
20 else
        mono=0;
22 end

```

---

## A.5. Stima parametrica inversa

Il seguente codice è quello utilizzato per effettuare la stima dei parametri come trattato nella sezione 4.3.3. Come nel codice riportato in A.2, il calcolo utilizza la funzione *inverse* (A.6), ma la stima non si limita solamente alle zone di tipo "A". In questo script non viene utilizzato il modello Simulink, ma solamente i dati provenienti dal dataset caricati con la funzione *EstraiDati* vista in A.1.

---

```

1     %% STIMA PARAMETRICA
    clc
3    clear
    close all
5    EoT='DataLog_111115.xlsx';
    load('Parametri\Parametri_media.mat')
7    [time, Temperature,NL,Pressioni,Tensione,Corrente]=...
        EstraiDati(EoT);
9
    Area=67.9*0.0001;    %[m2]
11   Nc=9;                % numero celle
    spessore=0.0002;    %[m]
13
    % conversioni unita' di misura
15   Temperature_K=Temperature+273.15;
    Pressioni_Pa=(Pressioni+1.01325)*1e5;
17   i0a_m=i0a*1e4; % conversione da A/cm2 a A/m2
    i0c_m=i0c*1e4;
19
    alpha_a_new=[];alpha_c_new=[];
21   i0a_new=[];i0c_new=[];lambda_new=[];
    for i=1:length(time)
23       if time(i)==900
            pause(0)
25       end
        [alpha_a_new(end+1,1),alpha_c_new(end+1,1),...
27         i0a_new(end+1,1),i0c_new(end+1,1),...
            lambda_new(end+1,1)]=...
29         inverse(Tensione(i,2),Corrente(i,2),...

```

## A.6. Inversione equazioni per il calcolo dei parametri

```
Area,Nc,spessore,alpha_a,alpha_c,i0a_m,...
31 i0c_m,lambda,Temperature_K(i,2:3),...
    Pressioni_Pa(i,2:3));
33 end

35 MediaScostamPerc=...
    mean([(i0a-i0a_new(~isnan(i0a_new)))/i0a...
37 (i0c-i0c_new(~isnan(i0c_new)))/i0c ...
    (alpha_a-alpha_a_new(~isnan(alpha_a_new)))/alpha_a...
39 (alpha_c-alpha_c_new(~isnan(alpha_c_new)))/alpha_c...
    (lambda-lambda_new(~isnan(lambda_new)))/lambda]);
41 MediaTot=mean(MediaScostamPerc);
```

---

## A.6. Inversione equazioni per il calcolo dei parametri

La funzione inverse applica le equazioni inverse attraverso le quali a partire dalle misurazioni del dataset e quattro dei cinque parametri considerati si ricava il parametro incognito.

---

```
1    %% INVERSIONE EQUAZIONI TENSIONE
function [alpha_a_s, alpha_c_s, i0a_s, i0c_s, lambda_s]=...
3    inverse(Vmis,I,A,Nc,delta,alpha_a,alpha_c,i0a,i0c,...
    lambda,T,P)
5
6    F= 96485.3365;
7    R=8.314;
8
9    i=I/A; % [A/m2]
10
11 % tensione a circuito aperto
12 if i>0
13     Erev = 1.229-0.9*(T(1)-298)*10^-3;
14     Vocv = Erev + R*T(1)*log(P(2)*sqrt(P(1)))/(2*F);
15 else
16     Vocv=0;
17     i=0;
18 end
19 % perdita ohmica calcolata
20 sigma=100*(0.005139*lambda-0.00326)*...
21 exp(1268*(1/303-1/T(1))); % [S/m]
22 Rmem = delta/sigma; % [m/S/m]=[Ohm*m2]
23 Vohm = Rmem*i; % [Ohm*m2*A/m2]=[V]
24
25 % perdita di attivazione per differenza
26 Vact_diff=Vmis/Nc-Vocv-Vohm;
27
```

## Appendice A. Codice Matlab

```
% stima i0a e i0c
29 factor=1;
   C1=R*T(1)/(factor*alpha_a*F);
31 C2=R*T(2)/(factor*alpha_c*F);
   Vact_a=C1*asinh(0.5*i/i0a);
33 Vact_c=C2*asinh(0.5*i/i0c);
   i0a_s=(0.5*i/sinh((Vact_diff-Vact_c)/C1))*1e-4;
35 i0c_s=(0.5*i/sinh((Vact_diff-Vact_a)/C2))*1e-4;

37 %stima alpha_a e alpha_c
   C1_s=(Vact_diff-C2*asinh(0.5*i/i0c))/asinh(0.5*i/i0a);
39 alpha_a_s=R*T(1)/(F*C1_s);
   C2_s=(Vact_diff-C1*asinh(0.5*i/i0a))/asinh(0.5*i/i0c);
41 alpha_c_s=R*T(2)/(F*C2_s);

43 % perdita di attivazione calcolata
   Vact=Vact_a+Vact_c;
45
   % perdita ohmica per differenza
47 Vohm_diff=Vmis/Nc-Vocv-Vact;

49 % stima di lambda
   sigma_s=delta*i/Vohm_diff;
51 lambda_s=(sigma_s*0.01/exp(1268*(1/303-1/T(1))))...
   +0.00326)/0.005139;
```

---

## A.7. Algoritmo PSO

Il codice riportato di seguito descrive la prima fase della ricerca PSO, al termine della quale si lancia la seconda. La seconda e terza fase si distinguono solamente nella fase di determinazione degli intervalli di ricerca, che vengono ristretti per come descritto in Tabella 3.7. Per interrompere le tre fasi ci si basa sul rapporto del fit sul tempo di durata del dataset, il rapporto serve a omogenizzare il fit sui dataset di diversa durata. Ogni fase ha una soglia di interruzione diversa per questo rapporto, la prima fase ha la soglia più alta, la terza la più bassa. La ricerca si interrompe anche se non ci sono miglioramenti maggiori di 0.001 per 10 iterazioni o la velocità delle particelle è scesa sotto la soglia impostata all'inizio dello script.

Il codice è basato di quello presente sul sito della Mathworks [33] al quale sono state apportate diverse modifiche.

---

```
%% ALGORITMO PSO %%
2 % Copyright (c) 2015, Yarpiz (www.yarpiz.com)
   % All rights reserved. Please read the
4 % "license.txt" for license terms.
   %
```

```

6  % Project Code: YPEA102
   % Project Title: Implementation of
8  % Particle Swarm Optimization in MATLAB
   %
10 % Publisher: Yarpiz (www.yarpiz.com)
   %
12 % Developer: S. Mostapha Kalami Heris (Member of Yarpiz Team)
   %
14 % Contact Info: sm.kalami@gmail.com, info@yarpiz.com
   %
16
   clc;
18 clear;
   close all;
20
   %% Problem Definition
22 DataLog = '104217';
24
   % Funzione di costo
   CostFunction=@() sim("PEM_PSO_v006.slx",'FastRestart','on');
26
   % Number of Decision Variables
28 nVarVol = 5;
30
   % Size of Decision Variables Matrix
   VarSizeVol = [1 nVarVol];
32
   % Lower/Upper bounds Voltage's parameters
34 VarsMinVol = [0.1 0.1 0 1e-3 5];
   VarsMaxVol = [2.5 1.5 4 1 20];
36
   PercVel2Stop = 0.25;
38 tlim=Inf;
40
   %% PSO Parameters
   MaxIt=100;      % Maximum Number of Iterations
42 nPop=100;      % Population Size (Swarm Size)
44
   % PSO Parameters
   wv=1;          % Inertia Weight
46 wdampv=0.99;   % Inertia Weight Damping Ratio
   c1v=1.5;       % Personal Learning Coefficient
48 c2v=2;         % Global Learning Coefficient
50
   % Velocity Limits
   VelMaxVol=0.1*(VarsMaxVol-VarsMinVol);
52 VelMinVol=-VelMaxVol;

```

## Appendice A. Codice Matlab

```
54 %% Estrazione dati
    [time, Temperature, NL, Pressioni, Tensione, Corrente]=...
56         EstraiDati(['DataLog_' DataLog]);

58 %% Initialization
    tstart=datetime;
60 disp([datestr(tstart) '□-□Inizializzazione'])

62 H2_Gain=1; H2_Tau=1; H2_Offset=1;

64 %% SE SI CARICANO I DATI DI UNA PRECEDENTE
    %% RICERCA COMMENTARE...
66 % caricamento vecchia popolazione e global best
    % load('filename.mat')
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ...DA QUI %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Particle structure
70 empty_particle.Position=[];
    empty_particle.Cost=[];
72 empty_particle.Velocity=[];
    empty_particle.Best.Position=[];
74 empty_particle.Best.Cost=[];

76 particle_Vol= repmat(empty_particle, nPop, 1);

78 % Global best initial value
    GlobalBest_Vol.Cost=inf;          % prima ricerca
80
82 for i=1:nPop
84     % Initialize Position
        particle_Vol(i).Position = unifrnd(VarsMinVol, VarsMaxVol);

86     % Initialize Velocity
        particle_Vol(i).Velocity = zeros(VarSizeVol);

88     % Parameter update, si assegnano i parametri per Simulink
90     [alpha_a, alpha_c, i0a, i0c, lambda] = ...
        parameterAssign(particle_Vol(i).Position);
92     i0a = 10^(i0a-6); % conversione in scala logaritmica

94     % Fit calculation from model
        fit = CostFunction();
96     Vol_IAE = fit.fitV.signals.values(end, 2);

98     particle_Vol(i).Cost=Vol_IAE;
```

```

100     % Update Personal Best
        particle_Vol(i).Best.Position = particle_Vol(i).Position;
102     particle_Vol(i).Best.Cost = particle_Vol(i).Cost;

104     % Update Global Best
        if particle_Vol(i).Best.Cost < GlobalBest_Vol.Cost
106         GlobalBest_Vol = particle_Vol(i).Best;
            end
108 end
        %%%%%%%%%%%%%%% FINO A QUI %%%%%%%%%%%%%%%
110
        %% Inizializzazione strutture di salvataggio dati
112 BestCost=zeros(MaxIt,1);

114 SinglePop_Vol = zeros(nPop,nVarVol);
        PastPopulations.Vol = SinglePop_Vol;
116 PastPopulations = repmat(PastPopulations,MaxIt,1);

118 WinPop = zeros(MaxIt,1);

120 %% Inizializzazione grafici
        figure (1)
122 hold on
        grid on
124 title('Voltage_Best_Fit')

126 figure(2)
        axis([0 1525 0 20])
128 hold on
        grid on
130 title('Confronto_tensioni')
        ylabel('V')
132 xlabel('tempo_[s]')
        xlim([0 time(end)])
134
        disp([datestr(datetime) ' Fine_inizializzazione'])
136
        %% PSO Main Loop
138
        for it=1:MaxIt
140             MaxMeanPercVel=0;
                    for i=1:nPop
142
                            % Update Velocity
144                             particle_Vol(i).Velocity = ...
                                    ww*particle_Vol(i).Velocity ...
146                             +c1v*rand(VarSizeVol).*...

```

## Appendice A. Codice Matlab

```

    (particle_Vol(i).Best.Position-...
148     particle_Vol(i).Position)...
    +c2v*rand(VarSizeVol).*...
150     (GlobalBest_Vol.Position-...
    particle_Vol(i).Position);
152

154     % Apply Velocity Limits
    particle_Vol(i).Velocity = ...
156         max(particle_Vol(i).Velocity, VelMinVol);
    particle_Vol(i).Velocity = ...
158         min(particle_Vol(i).Velocity, VelMaxVol);

160     % Update Position
    particle_Vol(i).Position = ...
162         particle_Vol(i).Position + ...
        particle_Vol(i).Velocity;
164

166     % Velocity Mirror Effect
    IsOutside_Vol=...
        (particle_Vol(i).Position < VarsMinVol |...
168         particle_Vol(i).Position > VarsMaxVol);
    % si invertono le vellocita' dei parametri che escono
170     particle_Vol(i).Velocity(IsOutside_Vol)=...
        -particle_Vol(i).Velocity(IsOutside_Vol);
172

174     % Apply Position Limits
    particle_Vol(i).Position = ...
        max(particle_Vol(i).Position, VarsMinVol);
176     particle_Vol(i).Position = ...
        min(particle_Vol(i).Position, VarsMaxVol);
178

180     % Fit calculation from model
    [alpha_a, alpha_c, i0a, i0c, lambda] = ...
        parameterAssign(particle_Vol(i).Position);
182     i0a = 10^(i0a-6); % conversione in scala logaritmica

184     fit = CostFunction(); % si lancia la simulazione
    Vol_IAE = fit.fitV.signals.values(end,2);
186

188     % Evaluation
    particle_Vol(i).Cost=Vol_IAE;

190     % Update Personal Best Voltage
    if particle_Vol(i).Cost < particle_Vol(i).Best.Cost
192         particle_Vol(i).Best.Position = ...
```

```

194         particle_Vol(i).Position;
particle_Vol(i).Best.Cost = ...
196         particle_Vol(i).Cost;

198         % Update Global Best
if particle_Vol(i).Best.Cost < GlobalBest_Vol.Cost
200             GlobalBest_Vol = particle_Vol(i).Best;
WinPop(it,1) = i;

202
figure(2)
204     cla
plot(fit.Voltage.V_mis)
206     plot(fit.Voltage.V_mod)
legend('V_m_i_s', 'V_m_o_d', ...
208         'Location', 'southeast')
% xlim([0 time(end)])
210     end

212     end

214
% Salvataggio popolazioni
216     PastPopulations(it).Vol(i,:) = particle_Vol(i).Position;

218     % aggiornamento max velocita' particelle
if mean(abs(particle_Vol(i).Velocity)./(VelMaxVol))...
220         >MaxMeanPercVel
MaxMeanPercVel = ...
222         mean(abs(particle_Vol(i).Velocity)./(...
(VelMaxVol)));
224     end

226     end

228     % Aggiornamento della posizione migliore finora trovata
BestCost(it,:) = GlobalBest_Vol.Cost;

230
% Visualizzazione fit
232     disp([datestr(datetime) ' _Iteration_ ' ...
num2str(it) ' :_Voltage_Best_Cost_ =_ ' ...
234         num2str(BestCost(it,1))]);
disp(['Ultima_posizione_migliore:_ [' ...
236         num2str(GlobalBest_Vol.Position) ' ]'])
disp(['MaxMeanPercVel:_ ' num2str(MaxMeanPercVel)])
238
% Plot fit
240     figure(1)

```

Appendice A. Codice Matlab

```
semilogy(it,GlobalBest_Vol.Cost,'x')
242 hold on

244 wv=wv*wdampv;

246 % Condizioni di terminazione della ricerca:
if GlobalBest_Vol.Cost/fit.tout(end)<0.045 ||...
248 (it>9&&(BestCost(it-9,1)-BestCost(it,1))<1e-3)||...
MaxMeanPercVel<PercVel2Stop
250 disp('Termine della ricerca')
break
252 end
end
254 BestSol = GlobalBest_Vol;

256 %% Compressione e salvataggio
258 PastPopulations = PastPopulations(1:it);
WinPop = WinPop(1:it);
260 BestCost = BestCost(1:it);

262 save([num2str(year(tstart)) '_' num2str(month(tstart))...
 '_' num2str(day(tstart)) '_h' num2str(hour(tstart))...
264 'm' num2str(minute(tstart)) '_DL' DataLog '_D'])

266 %% Lancio dello script della seconda fase
PSO_Voltage_Delta_p;
```

---

## Bibliografia

- [1] S Shiva Kumar and Hankwon Lim. An overview of water electrolysis technologies for green hydrogen production. *Energy reports*, 8:13793–13813, 2022.
- [2] Cheng Haoran, Yanghong Xia, Wei Wei, Zhou Yongzhi, Zhao Bo, and Zhang Leiqi. Safety and efficiency problems of hydrogen production from alkaline water electrolyzers driven by renewable energy sources. *International Journal of Hydrogen Energy*, 2023.
- [3] Inc. EG&G Technical Services. *Fuel Cell Handbook (Seventh Edition)*. U.S. Department of Energy Office of Fossil Energy National Energy Technology Laboratory, 2004.
- [4] G Ciancia, A Pede and R Marmigi. Fuel-cell technology for vehicular applications; l'introduzione delle pile a combustibile nel settore dei trasporti: proposte, standard e normative per sistemi di bordo ed impianti, Dec 1998.
- [5] Ahmad Baroutaji, Tabbi Wilberforce, Mohamad Ramadan, and Abdul Ghani Olabi. Comprehensive investigation on hydrogen and fuel cell technology in the aviation and aerospace sectors. *Renewable and sustainable energy reviews*, 106:31–40, 2019.
- [6] Maria Grazia De Giorgi, Paolo Maria Congedo, Antonio Ficarella, et al. Prestazioni di una fuel cell a servizio di un edificio civile. In *64° CONGRESSO NAZIONALE ATI*, pages 1–12. Libreria Universitaria Benedetti, 2009.
- [7] Jimena Incer-Valverde, Amira Korayem, George Tsatsaronis, and Tatiana Morosuk. “colors” of hydrogen: Definitions and carbon intensity. *Energy Conversion and Management*, 291:117294, 2023.
- [8] Shahbaz Ahmad, Tahir Nawaz, Asghar Ali, Mehmet Fatih Orhan, Ayesha Samreen, and Arunachala M Kannan. An overview of proton exchange membranes for fuel cells: Materials and manufacturing. *International Journal of Hydrogen Energy*, 47(44):19086–19131, 2022.
- [9] Mahdi Kheirrouz, Francesco Melino, and Maria Alessandra Ancona. Fault detection and diagnosis methods for green hydrogen production: A review. *International Journal of Hydrogen Energy*, 47(65):27747–27774, 2022.
- [10] P Medina and M Santarelli. Analysis of water transport in a high pressure pem electrolyzer. *International Journal of Hydrogen Energy*, 35(11):5173–5186, 2010.

## Bibliografía

- [11] Scott D Greenway, Elise B Fox, and Amy A Ekechukwu. Proton exchange membrane (pem) electrolyzer operation under anode liquid and cathode vapor feed configurations. *International journal of hydrogen energy*, 34(16):6603–6608, 2009.
- [12] Alexandre Tugirumubano, Hee-Jae Shin, Sun-Ho Go, Min-Sang Lee, Lee Ku Kwac, and Hong-Gun Kim. Electrochemical performance analysis of a pem water electrolysis with cathode feed mode based on flow passage shape of titanium plates. *International Journal of Precision Engineering and Manufacturing*, 17:1073–1078, 2016.
- [13] DS Falcão and AMFR Pinto. A review on pem electrolyzer modelling: Guidelines for beginners. *Journal of cleaner production*, 261:121184, 2020.
- [14] Haluk Görgün. Dynamic modelling of a proton exchange membrane (pem) electrolyzer. *International journal of hydrogen energy*, 31(1):29–38, 2006.
- [15] Haluk Görgün, Murat Arcaç, and Frano Barbir. An algorithm for estimation of membrane water content in pem fuel cells. *Journal of power sources*, 157(1):389–394, 2006.
- [16] Sandip Dutta, Sirivatch Shimpalee, and JW Van Zee. Numerical prediction of mass-exchange between cathode and anode channels in a pem fuel cell. *International Journal of Heat and Mass Transfer*, 44(11):2029–2042, 2001.
- [17] Daoping Tang, Juntao Lu, Lin Zhuang, and Peifang Liu. Calculations of the exchange current density for hydrogen electrode reactions: A short review and a new equation. *Journal of Electroanalytical Chemistry*, 644(2):144–149, 2010.
- [18] Rafael García-Valverde, Nieves Espinosa, and Antonio Urbina. Simple pem water electrolyser model and experimental validation. *international journal of hydrogen energy*, 37(2):1927–1938, 2012.
- [19] Somayeh Toghyani, Soheil Fakhradini, Ebrahim Afshari, Ehsan Baniasadi, Mohammad Yaghoob Abdollahzadeh Jamalabadi, and Mostafa Safdari Shadloo. Optimization of operating parameters of a polymer exchange membrane electrolyzer. *International Journal of Hydrogen Energy*, 44(13):6403–6414, 2019.
- [20] Z Abdin, CJ Webb, and E MacA Gray. Modelling and simulation of a proton exchange membrane (pem) electrolyser cell. *International journal of hydrogen energy*, 40(39):13243–13257, 2015.
- [21] Eng Waseem Saeed and Eng Ghaith Warkozek. Modeling and analysis of renewable pem fuel cell system. *Energy Procedia*, 74:87–101, 2015.
- [22] Ángel Hernández-Gómez, Victor Ramirez, and Damien Guilbert. Investigation of pem electrolyzer modeling: Electrical domain, efficiency, and specific energy

- consumption. *International journal of hydrogen energy*, 45(29):14625–14639, 2020.
- [23] Mousa Meratizaman, Sina Monadzadeh, and Majid Amidpour. Simulation, economic and environmental evaluations of green solar parking (refueling station) for fuel cell vehicle. *International journal of hydrogen energy*, 39(5):2359–2373, 2014.
- [24] Dmitry Pashchenko. Green hydrogen as a power plant fuel: What is energy efficiency from production to utilization? *Renewable Energy*, page 120033, 2024.
- [25] Norafneeza Norazahar, Faisal Khan, Nazmul Rahmani, and Arshad Ahmad. Degradation modelling and reliability analysis of pem electrolyzer. *International Journal of Hydrogen Energy*, 50:842–856, 2024.
- [26] Alhassan Salami Tijani and AH Abdol Rahim. Numerical modeling the effect of operating variables on faraday efficiency in pem electrolyzer. *Procedia Technology*, 26:419–427, 2016.
- [27] Michela Trapani. *Algoritmi di dinamiche di sciame di particelle per l’ottimizzazione del Machine Learning= Algorithms for swarm dynamics employed to optimization problems*. PhD thesis, Politecnico di Torino, 2021.
- [28] Chang Liu, Jacob Wrubel, Elliot Padgett, and Guido Bender. The impacts of membrane pinholes on pem water electrolysis. *Journal of Power Sources*, 581:233507, 2023.
- [29] S Boulevard, JJA Kadjo, A Thomas, B Grondin Perez, and S Martemianov. Characterization of aging effects during pem electrolyzer operation using voltage instabilities evolution. *Russian Journal of Electrochemistry*, 58(4):258–270, 2022.
- [30] Om Prakash, Sumit Sood, Mahdi Boukerdja, Belkacem Ould-Bouamama, Jean-Yves Dieulot, Anne-Lise Gehin, and Mathieu Bressel. A model-based prognosis approach to proton exchange membrane water electrolysis system. In *2021 European Control Conference (ECC)*, pages 2133–2138. IEEE, 2021.
- [31] Mathworks website, polyfit function. <https://it.mathworks.com/help/matlab/ref/polyfit.html>.
- [32] Mathworks website, vpsolve function. [https://it.mathworks.com/help/symbolic/sym.vpsolve.html?searchHighlight=vpsolve&s\\_tid=srchtitle\\_support\\_results\\_1\\_vpsolve](https://it.mathworks.com/help/symbolic/sym.vpsolve.html?searchHighlight=vpsolve&s_tid=srchtitle_support_results_1_vpsolve).
- [33] Mathworks website, particle swarm optimization (pso). <https://it.mathworks.com/matlabcentral/fileexchange/52857-particle-swarm-optimization-pso>.



# Ringraziamenti

Ringrazio il gruppo Loccioni e Luca Mazzuferi, per avermi dato la possibilità di svolgere l'attività di tirocinio sulla quale ho potuto sviluppare questa tesi, in particolare ci tengo a ringraziare Emiliano, che si è dimostrato sempre pronto a dare un consiglio e disponibile per un confronto.

Ringrazio il professor Nobili, che più volte ha saputo chiarire i dubbi legati agli aspetti elettrochimici, e il professor Freddi, relatore di questa tesi, per avermi guidato durante questo progetto.

Ringrazio Enrico e Riccardo, compagni straordinari dell'intero percorso e sempre pronti a fare squadra. Il loro contributo è stato fondamentale fino all'ultimo esame.

Ringrazio tutti coloro che hanno mostrato un attivo interesse nei confronti di ciò che stavo svolgendo, dando occasione di spendere due parole e condividere riflessioni.

Ringrazio la mia famiglia, che mi ha costantemente motivato e sostenuto durante tutto il periodo di studi.

Per concludere ringrazio Chiara, un grazie per starmi vicina e supportarmi sempre, un grazie per gli elogi e per le critiche, un grazie per darmi la possibilità di osservare le cose da altri punti di vista.

*Ancona, 16 Febbraio 2024*

Francesco Silvi