



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E
DELL'AUTOMAZIONE

**STUDIO E IMPLEMENTAZIONE DI ALGORITMI DI
SUPERVISIONE E CONTROLLO PER UN IMPIANTO DI
COGENERAZIONE MICRO SOLARE**

*Study and development of supervision and control
algorithms for a micro solar cogeneration plant*

RELATORE

DOTT. ALESSANDRO FREDDI

CORRELATORE

ING. MATTEO PIRRO

TESI DI LAUREA DI

SIMONE DE GRANDIS

MATR. S1091457

ANNO ACCADEMICO 2019/2020

Indice

1	Introduzione	1
2	Descrizione del sistema	3
2.1	<i>Linear Fresnel Reflector, LFR</i>	5
2.2	<i>Organic Rankine Cycle, ORC</i>	7
2.3	<i>Phase change material thermal storage unit, TES</i>	10
2.4	<i>Operative mode, OM</i>	11
3	Richiami sulle metodologie impiegate	13
3.1	Logica fuzzy	14
3.1.1	Operazioni sugli insiemi fuzzy	17
3.1.2	Relazione fuzzy	20
3.1.3	Variabili linguistiche	21
3.1.4	Inferenza fuzzy	22
3.1.5	Sistema di inferenza Mamdani	24
3.2	Reti neurali	28
3.2.1	Apprendimento	32
3.2.2	<i>Multi-layer perceptron</i>	33
4	Logica di controllo	35
4.1	Controllore base	36
4.2	Nuovo controllore	38
4.2.1	Inseguimento termico	39
4.2.2	Profilo per abitazioni private	40
4.2.3	Profilo per edificio scolastico	41
4.3	Massimizzazione della produzione di energia	43
5	Controllori implementati	45
5.1	Moduli ausiliari di gestione dati	46
5.2	Controllore per l'inseguimento termico	51
5.2.1	Profilo residenziale	52
5.2.2	Profilo edificio scolastico	58
5.2.3	<i>Performance-based supervision</i>	61
5.2.4	Stima delle prestazioni del sistema in funzione dei consumi	64
5.2.5	Implementazione del controllore in MATLAB/Simulink	68
5.3	Controllore per la massimizzazione della produzione	74
5.3.1	Variazione di P_{σ}	75
5.3.2	Implementazione del controllore in MATLAB/Simulink	77
6	Risultati ottenuti	80
6.1	Controllore per l'inseguimento termico (avanzato)	81
6.2	Controllore per la massimizzazione della produzione	85
7	Applicazione per l'analisi della produzione giornaliera di energia	88
7.1	Descrizione dell'architettura	90
7.2	Descrizione dei componenti software	92
8	Conclusioni	93
9	Appendici	95
A	Manuale di uso e manutenzione del simulatore	96
B	Manuale utente dell'applicazione per l'analisi della produzione giornaliera	100
A	Bibliografia e sitografia	103

Elenco delle figure

1	Schema a blocchi del sistema	3
2	Schema logico semplificato del sistema	4
3	Lente di Fresnel in sezione	5
4	Cilindro parabolico	6
5	Campo a concentrazione solare con lenti di Fresnel	6
6	Esempio di impianto ORC	7
7	Schema funzionale di un ORC	8
8	Diagramma $T - S$ di un ORC	9
9	Curva di riscaldamento di un materiale a cambiamento di fase	10
10	Riassunto delle modalità operative	11
11	Esempio di funzione di appartenenza	14
12	Grafico d'esempio di $\mu_A(x)$	15
13	Risultato dell'intersezione	18
14	Risultato dell'unione	18
15	Conseguenze della complementazione	19
16	Conseguenze della complementazione	19
17	Schema logico dell'inferenza fuzzy	22
18	Schema logico di un sistema di controllo fuzzy	24
19	Esempio di inferenza fuzzy	26
20	Esempio di un neurone	29
21	Funzione di attivazione di un perceptrone	30
22	Funzioni rappresentabili con un perceptrone	30
23	Esempio di funzione rappresentabile con una rete di perceptroni	31
24	Esempio di un <i>multi-layer perceptron</i>	33
25	<i>Overfitting</i>	34
26	Parametri per la scelta dell' <i>operative mode</i>	36
27	Schema a blocchi per il profilo dei consumi privato	40
28	Curva dei consumi di un edificio scolastico	41
29	Generazione dei consumi scolastici in funzione del tempo	41
30	Generazione dei consumi scolastici in funzione del tempo	42
31	Profilo dei consumi scolastici	42
32	Andamento delle grandezze E_{LFR} , E_e ed E_t	43
33	Andamento dei pesi w_e d w_t	43
34	Implementazione in Simulink del caricamento dati	46
35	Estrazione del dato temporalmente corretto	46
36	Esempio di estrazione del dato	46
37	Salvataggio <i>feature</i> per l'addestramento	47
38	Acquisizione dati grezza	48
39	<i>Edge detector</i>	48
40	Contatore delle commutazioni	49
41	Esempio di conteggio	49
42	Interfaccia grafica in Simulink	50
43	Correzione del segnale e controllo su T_{TES}	50
44	Schema logico del controllore avanzato	52
45	Rappresentazione degli insiemi fuzzy per P_{LFR}	53
46	Rappresentazione degli insiemi fuzzy per T_{TES}	54
47	Rappresentazione degli insiemi fuzzy per $T_{LFR_{out}}$	54
48	Rappresentazione degli insiemi fuzzy per $System_{info}$	55
49	Rappresentazione degli insiemi fuzzy per T_{diff}	56
50	Rappresentazione grafica delle regole fuzzy	56
51	Schema logico del controllore per il profilo scolastico	58
52	Rappresentazione degli insiemi fuzzy per <i>time</i>	58
53	Rappresentazione grafica delle regole fuzzy	59
54	Differenza di <i>operative mode</i>	60
55	Schema logico della politica di <i>switching</i>	61
56	Schema logico di un supervisore basato su prestazioni	62

57	Esempio di <i>switching signal</i>	63
58	Schema per la generazione del segnale per le prestazioni	64
59	Schema logico di una soglia adattiva	64
60	Prima parte della costruzione del segnale	65
61	Seconda parte della costruzione del segnale	65
62	Elaborazione di $\sigma(t)$	66
63	Andamento di $\sigma(t)$	66
64	Schema a blocchi del controllore per l'inseguimento termico	68
65	Schema a blocchi per la creazione dell'input della rete neurale	69
66	Rete neurale per l'identificazione	69
67	Parametri di sintesi della rete neurale	70
68	Grafico delle performance della rete neurale	71
69	Matrice di confusione della rete neurale	71
70	Primo caso di <i>switch</i>	72
71	Secondo caso di <i>switch</i>	72
72	Controllore fuzzy e blocchi aggiuntivi	73
73	Modifica della soglia di attacco	75
74	Implementazione del calcolo dei pesi	77
75	Calcolo delle derivate dei pesi	77
76	Schema a blocchi completo	78
77	Controllore per la massimizzazione dell'energia prodotta	78
78	Funzione che massimizza la produzione	79
79	Variazione percentuale di produzione tra i controllori	80
80	Ora media di inizio e fine produzione per il controllore base	83
81	Ora media di inizio e fine produzione per il controllore avanzato	83
82	Differenza temporale di produzione	84
83	Differenza di <i>operative mode</i> in funzione dei profili di consumo	84
84	Ora media di inizio e fine produzione per il controllore base	87
85	Ora media di inizio e fine produzione per il controllore per la massimizzazione della produzione	87
86	<i>Mockup</i> della finestra principale	88
87	<i>Mockup</i> della finestra per la visualizzazione dei grafici	89
88	<i>Mockup</i> della finestra per la visualizzazione tabellare	89
89	Diagramma UML delle classi	90
90	Diagramma UML delle sequenze	91
91	Blocchi di codice aggiuntivi	96
92	<i>Model Browser di Simulink</i>	97
93	Finestra grafica principale del programma	100
94	Finestra grafica principale del programma dopo il caricamento dei dati	100
95	Finestra grafica per la visualizzazione tabellare dei dati	101
96	Finestra grafica per la visualizzazione dei grafici	101
97	Finestra per la visualizzazione del grafico delle differenze di produzione	102

Elenco delle tabelle

1	Modalità operative	11
2	Esempio di relazione fuzzy	20
3	<i>Operative mode</i> e relative condizioni	37
4	Tabella esplicativa di T_{set}	37
5	Intervalli fuzzy di P_{LFR}	52
6	Intervalli fuzzy di T_{TES}	53
7	Intervalli fuzzy di $T_{LFR_{out}}$	54
8	Intervalli fuzzy di $System_{info}$	55
9	Intervalli fuzzy di T_{diff}	55
10	Regole fuzzy implementate	57
11	Intervalli fuzzy di <i>time</i>	58
12	Regole fuzzy per il profilo scolastico dei consumi	59
13	Interpretazione della soglia $\Sigma(t)$	67
14	Modalità di modifica dell' <i>operative mode</i> calcolata dal controllore avanzato	76
15	Risultati ottenuti con il controllore avanzato	81
16	Valutazione del risparmio con il controllore avanzato	81
17	Ora di inizio e fine produzione del controllore avanzato	82
18	Confronto dei risultati ottenuti	85
19	Valutazione del risparmio con il controllore per la massimizzazione della produzione	85
20	Valutazione rispetto il controllore avanzato	86
21	Ora di inizio e fine produzione con il controllo per la massimizzazione della produzione	86

1 Introduzione

L'obiettivo della seguente tesi è l'implementazione di algoritmi di controllo e supervisione per ottimizzare le performance di un sistema di cogenerazione micro solare.

In Europa, infatti, circa il 40% dei consumi elettrici ed il 36% delle emissioni di CO₂ sono imputabili al settore edilizio.

Tra le varie tecnologie sviluppate in questo campo, la produzione combinata di energia elettrica e calore (*combined heat and power*, CHP) consente di ridurre il consumo di energia primaria con un conseguente abbassamento delle emissioni di CO₂ [1].

In particolare, con impianto di cogenerazione si intende un sistema che produce in modo combinato sia energia elettrica che termica.

Nel caso preso in analisi, si ha la produzione combinata di elettricità e calore da un impianto di piccola taglia. Si parla in questi casi di microcogenerazione (μ CHP).

Per la sola produzione di energia elettrica si utilizzano generalmente centrali termoelettriche che disperdono una parte dell'energia termica nell'ambiente che, essendo a bassa temperatura, ha scarso valore termodinamico.

Per produrre solo energia termica, invece, solitamente si adottano caldaie che convertono l'energia primaria ad alto valore termodinamico in calore a temperatura più bassa. La cogenerazione si basa sulla considerazione che, come precedentemente illustrato, in qualunque ciclo termodinamico diretto dal quale è possibile estrarre lavoro (in questo caso sotto forma di energia elettrica), una parte del calore a temperatura maggiore fornito al sistema viene ceduto a temperatura minore. La parte di energia così ceduta è quindi una perdita nel processo che, negli impianti di cogenerazione, viene almeno in parte recuperata [S1].

Una soluzione di questo tipo comporta:

- costi minori ed un conseguente risparmio economico dovuto al ridotto consumo di combustibili;
- riduzione dell'inquinamento atmosferico e termico a seguito delle emissioni e rilascio di calore nell'ambiente contenuti;
- minore dispersione nel sistema elettrico nazionale dovuta alla trasmissione e alla distribuzione (si avranno impianti localizzati o autoconsumo dell'energia prodotta);
- sostituzione delle attuali modalità di fornitura del calore, meno efficienti e più inquinanti.

A livello normativo è utile specificare la diversa definizione di un impianto di cogenerazione in base alla capacità produttiva dello stesso [S2].

Si avranno infatti:

- Impianti di micro cogenerazione, per unità con capacità di generazione inferiori ai 50 kWe;
- Impianti di piccola cogenerazione, per unità con capacità di generazione inferiori ad 1 MWe.

Tipicamente, un impianto di cogenerazione è costituito da:

- Motore primo (ad esempio motori alternativi o impianti turbogas);
- Generatore elettrico;
- Impianto motore;
- Recuperatori di calore.

Un esempio di cogenerazione è dato dall'automobile. Infatti, la potenza presa all'albero motore è in parte usata per la trazione ed in parte convertita in energia elettrica, mentre il calore sottratto ai cilindri è utilizzato per riscaldare l'abitacolo e la pressione dei gas di scarico per movimentare la turbina di sovralimentazione (introduzione forzata di miscela combustibile-comburente nei cilindri dell'automobile per aumentare le prestazioni in termini di potenza e coppia generate) [S3]. L'utilizzo di questi elementi consente di convertire l'energia primaria immessa (ovvero l'energia chimica ottenuta dalla combustione) in altre forme di energia secondaria prodotte (in questo caso meccanica e termica) senza comportare un aumento dei consumi in quanto si utilizzano scarti del processo di conversione di energia chimica a energia meccanica attuato dal motore.

Questa tesi si inserisce all'interno del progetto *Innova MicroSolar (Innovative Micro Solar Heat and Power System for Domestic and Small Business Residential Buildings)* [S5]. Il progetto, finanziato dall'Unione Europea, si inserisce nell'ambito di *Horizon 2020*, ovvero un programma di finanziamenti per ricerca ed innovazione [S6]. L'obiettivo del progetto *Innova* è quello di incentivare e sviluppare l'utilizzo dell'energia termica solare in ambito domestico o di piccole imprese per la produzione di energia elettrica e termica.

Come precedentemente accennato, l'obiettivo di questo lavoro è quello di sviluppare algoritmi di ottimizzazione e supervisione per l'impianto oggetto del progetto *Innova MicroSolar*. Tale impianto di cogenerazione, ancora in fase di costruzione, è stato dapprima modellato e poi tradotto nel simulatore (sviluppato in MATLAB e Simulink) utilizzato in questo lavoro nell'ambito della *Simulink Student Challenge* del 2018 da R. Tascioni in *Dynamic Simulation of u-solar-CHP with Simulink*¹.

Il simulatore appena citato, inoltre integra al suo interno un controllore che gestisce il comportamento del sistema. Lo scopo della presente tesi è quindi quello di sostituire e migliorare tale controllore in modo da poter assicurare prestazioni migliori e adattare il comportamento del sistema in base all'utenza e quindi al carico ed il profilo dei consumi ad esso collegati, andando quindi ad ampliare i possibili campi di applicazione che un sistema di questo tipo può avere. Infatti, tra i vari algoritmi di ottimizzazione implementati si hanno quello per l'inseguimento termico (a seconda del carico collegato all'impianto, cambierà il tipo di algoritmo e di logica utilizzati) e per la massimizzazione della produzione di energia.

Allo stato iniziale, ovvero per quanto implementato nel lavoro precedentemente citato, il sistema può lavorare in 8 diverse modalità e la commutazione tra queste viene fatta dal controllore in modo tabellare e con condizioni fisse. Utilizzando una soluzione del tipo di quella implementata in questo lavoro e successivamente discussa, si è tradotto il vecchio controllore in uno nuovo che, oltre a ridefinire le vecchie condizioni di lavoro prima introdotte, assicura anche un incremento delle prestazioni. Al nuovo controllore, quindi, sono state operate modifiche per adattarlo al meglio al caso che si vuole prendere in esame una simulazione (che può essere cambiare il tipo di utenza collegata o l'obiettivo del controllo).

In relazione a quanto detto, gli strumenti utilizzati in questa tesi sono stati MATLAB e Simulink, sviluppati dalla Mathworks. Il primo è un ambiente per il calcolo numerico che comprende anche l'omonimo linguaggio di programmazione che consente, tra l'altro, di sviluppare algoritmi, analizzare dati e creare interfacce grafiche per gli utenti, mentre il secondo è un ambiente di sviluppo con linguaggio grafico che consente la modellazione, simulazione ed analisi di sistemi dinamici di varia natura [S4].

La seguente tesi è articolata nelle seguenti sezioni: nella seconda viene riportata una descrizione del sistema di cogenerazione simulato dal programma Simulink e frutto del progetto *Innova MicroSolar*;

Nella terza e nella quarta sezione vengono riportati e trattati gli argomenti teorici alla base del lavoro svolto e di quanto implementato in questo elaborato.

Nella quinta sezione viene introdotta la politica di *switching*, ovvero in che modo le varie modalità operative in cui può lavorare il sistema vengono decise e come avviene la loro commutazione in funzione dell'obiettivo di controllo prefissato ed in base all'utenza collegata all'impianto di produzione di energia e quindi in base al tipo di consumi.

Nella sesta sezione, invece, viene riportata una panoramica sui blocchi (e quindi sulle funzioni operative) implementate. In sezione sette, invece, si avrà una descrizione dettagliata del componente principale sviluppato in questo lavoro, ovvero il controllore che sostituisce quello precedentemente sviluppato per il simulatore e che, di fatto, costituisce l'obiettivo principale della tesi.

Nell'ottava sezione viene descritta l'applicazione sviluppata in C#, come precedentemente accennato, per la visualizzazione e l'analisi dei dati ricavati da una simulazione.

Nella nona sezione vengono riassunti i risultati ottenuti da ognuno dei vari profili di controllo implementati e, infine, nella decima sezione si procederà ad una discussione degli stessi esponendone anche gli elementi più importanti.

Sono presenti anche due appendici in cui è riportato il manuale utente delle due applicazioni sviluppate, ovvero i blocchi aggiunti al simulatore e l'applicazione in C#.

¹Per maggiori informazioni su R. Tascioni - *Dynamic Simulation of u-solar-CHP with Simulink* si faccia riferimento a Mathworks - *Simulink Student Challenge* 2018.

2 Descrizione del sistema

L'obiettivo del progetto *Innova*, come anche precedentemente detto, è quello di sviluppare un sistema ad alte prestazioni di $2kWe/18kWh$ per la produzione di calore ed energia elettrica. La tecnologia proposta si basa su un riflettore di *Fresnel* (LFR) per concentrare l'energia solare e un *micro Organic Rankine Cycle* (ORC) per la conversione dell'energia (calore) a bassa temperatura in lavoro utile (in questo caso energia elettrica) in combinazione con un *thermal storage* (TES) che utilizza materiale a cambiamento di fase (PCM, *phase change material*).

In Figura 1 viene riportata una schematizzazione a blocchi con i componenti principali del sistema, oltre quelli appena introdotti.

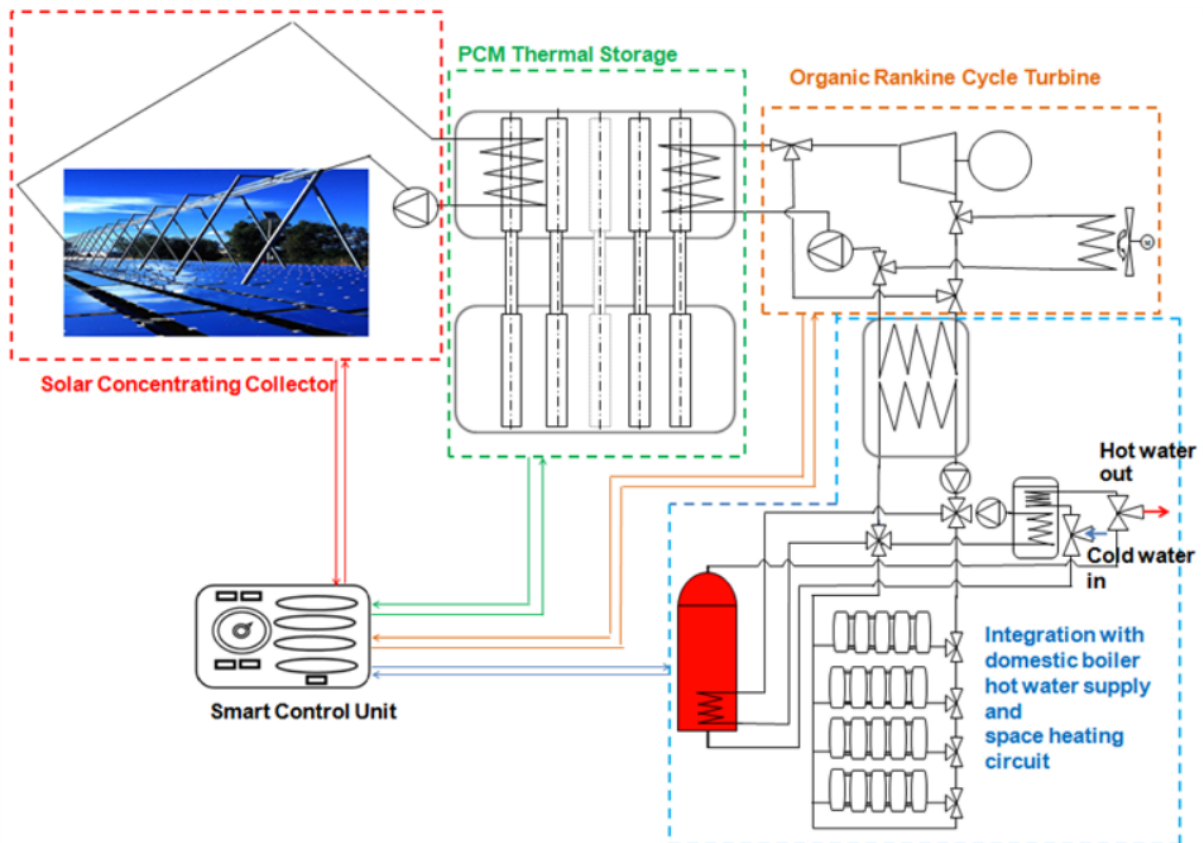


Figura 1: Schema con i principali blocchi funzionali del sistema. Per l'immagine si faccia riferimento a [F1].

Lo schema logico semplificato delle connessioni tramite valvole dei sottosistemi controllati (LFR, ORC e TES) è riportato in Figura 2.

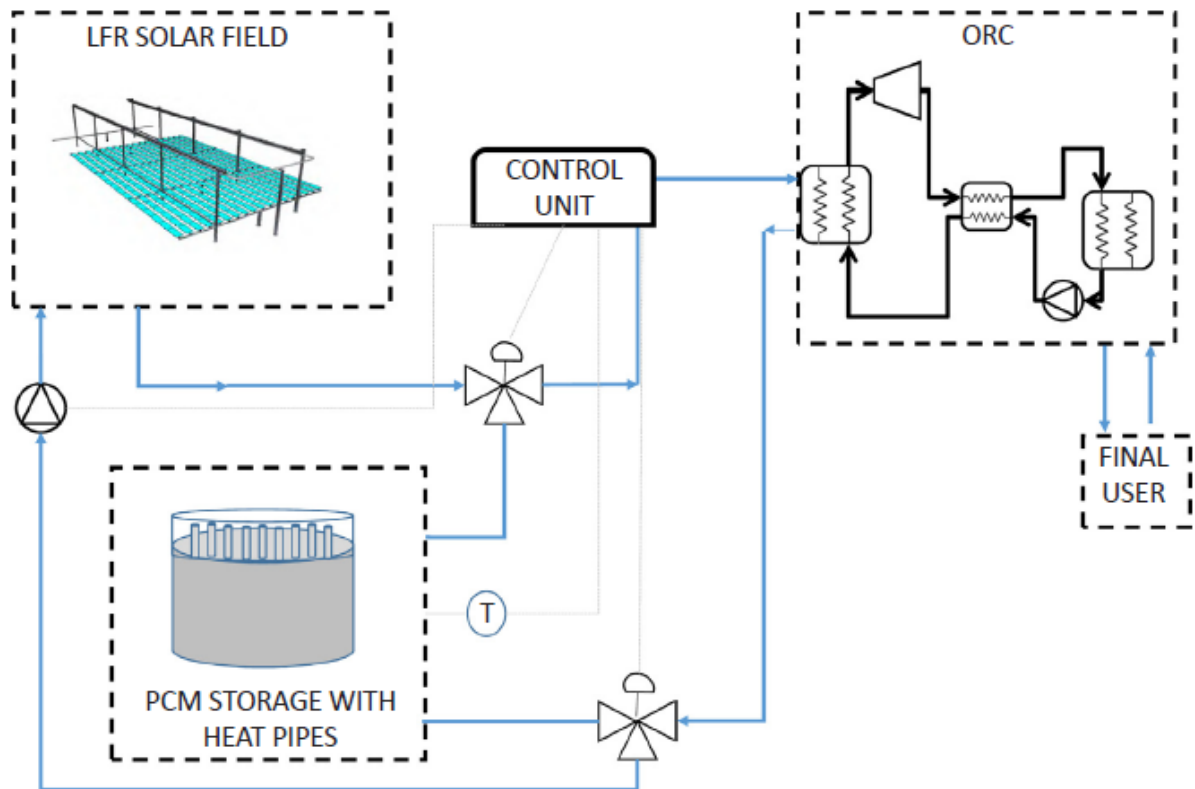


Figura 2: Schema logico semplificato della connessione di LFR, ORC e TES. Per l'immagine si faccia riferimento a [F2].

Nelle sottosezioni 2.1, 2.2 e 2.3 verranno meglio descritti i sottosistemi sopracitati (LFR, ORC e TES, rispettivamente). Il funzionamento del sistema cambia al variare delle configurazioni che assumono le varie valvole. Questo aspetto verrà illustrato nella sottosezione 2.4 in cui verranno mostrate le varie modalità operative (OM).

2.1 Linear Fresnel Reflector, LFR

Un *linear Fresnel reflector* risulta essere una forma modificata di un cilindro parabolico per la concentrazione solare che utilizza l'energia solare incidente [S7]. La particolarità di un sistema del tipo ora discusso sta nel non usare grandi cilindri per concentrare il flusso solare; il cilindro, infatti, è diviso in lunghe file di specchi piatti o leggermente curvati che, approssimando il profilo del cilindro, sono in grado di concentrare il flusso in modo simile a quello del caso cilindrico. Il flusso così concentrato è poi assorbito da un ricevitore lineare fisso (rivolto verso il basso).

L'LFR si basa sulle lenti di Fresnel, inventate dal matematico francese nel 1822 [S8]. Questo particolare tipo di lente è dato da una successione di anelli concentrici, ognuno costituito da un elemento di una lente semplice (ovvero costituita da un singolo componente) assemblati in modo da avere una superficie piatta ed una lunghezza focale ridotta. Le lenti di Fresnel sono particolarmente utilizzate per via delle loro capacità di concentrare la luce in un fascio relativamente stretto (un'applicazione tipica è l'utilizzo di questa lente nei fari) [S9].

Un'illustrazione che mostra come è costruita una lente di Fresnel è mostrata in Figura 3.

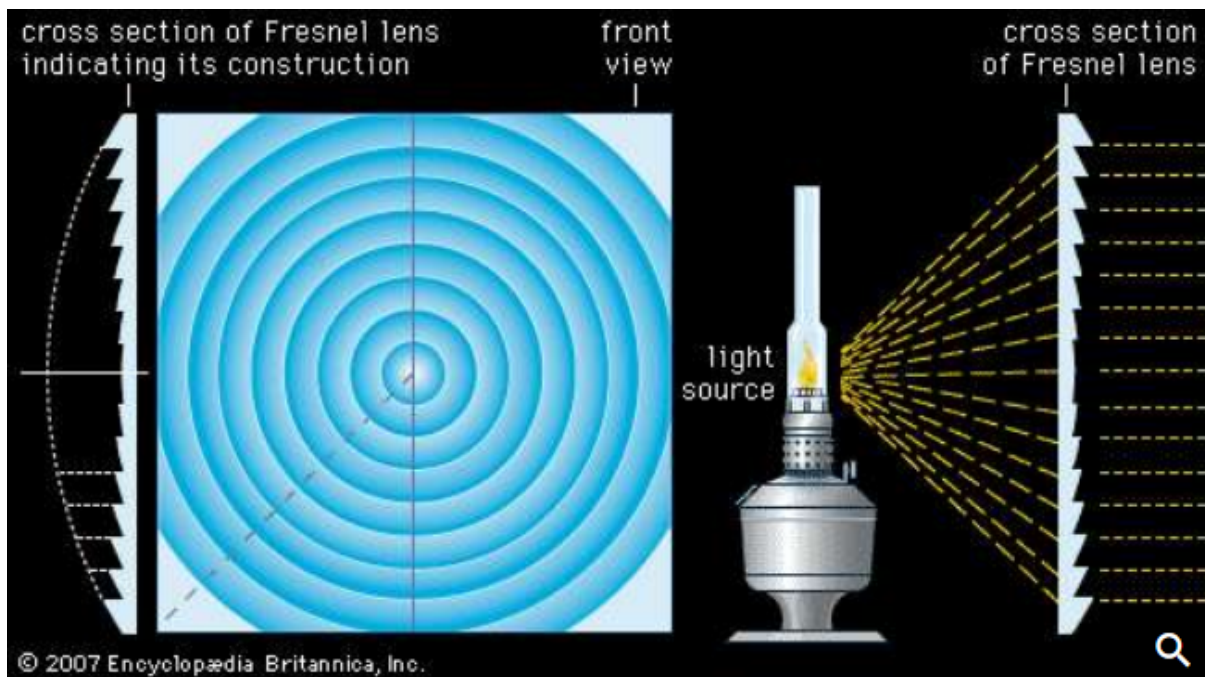


Figura 3: Lente di Fresnel in sezione. Per l'immagine si faccia riferimento a [F3].

In Figura 4 è mostrato un esempio di un impianto che utilizza un cilindro parabolico per la concentrazione del flusso solare, mentre in Figura 5 è mostrato l'impianto utilizzato nell'ambito del progetto *Innova-MicroSolar* precedentemente descritto.

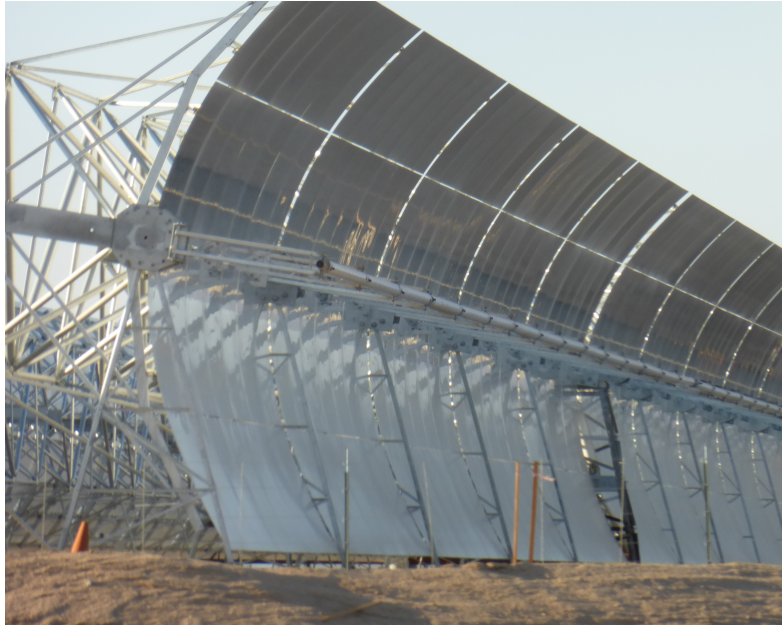


Figura 4: Cilindro parabolico in *Harper Lake*, California. Per l'immagine si faccia riferimento a [F4].



Figura 5: Campo a concentrazione solare con lenti lineari di Fresnel. Per l'immagine si faccia riferimento a [F1].

2.2 Organic Rankine Cycle, ORC

Con ciclo Rankine organico, s'intende un impianto che converte calore in energia termica, ma, al contrario di un normale ciclo di Rankine, utilizza come fluido di lavoro una sostanza organica ad alto peso molecolare (il nome di questo particolare ciclo termodinamico deriva proprio da questa caratteristica). Questa tecnologia viene sviluppata da Lucien Bronicki e Harry Zvi Tabor verso la fine degli anni '50 [S10]. Il ciclo Rankine, in tutte le sue varianti, è uno dei cicli più diffusi per la generazione di energia elettrica [S11].

In Figura 6 è riportato un esempio di ORC.



Figura 6: Impianto ORC da 75kW in Lappeenranta, Finlandia. Per l'immagine si faccia riferimento a [F5]

Un impianto ORC ha diversi campi di utilizzo: recupero del calore disperso (come nel caso qui discusso per un impianto di cogenerazione) ed in generale per il recupero di potenza in impianti di piccola taglia [2], geotermia a bassa entalpia [S11], ecc..

Il principio di funzionamento di questo ciclo termodinamico è del tutto analogo a quello del normale ciclo Rankine: il fluido di lavoro è immesso in una caldaia nel quale viene fatto evaporare (in un ciclo Rankine il fluido di lavoro è l'acqua e quindi il vapor acqueo). Il gas poi viene fatto espandere in un apposito sistema e quindi arriva in uno scambiatore di calore dove viene fatto ricondensare e quindi riportato allo stato liquido.

Riassumendo, un ORC è composto quindi da 3 scambiatori, come riportato di seguito:

- Evaporatore (scambiatore 1) che, presa l'energia termica in ingresso, la cede al fluido che evaporando entra, ad esempio, in una turbina (in questa parte ci sarà una variazione di temperatura ΔT ed una variazione di pressione ΔP , per cui si genererà lavoro utile e quindi potenza meccanica).
- Condensatore (scambiatore 2), che riporta il fluido allo stato liquido e lo reimmette nel sistema tramite una pompa (occorre infatti meno lavoro per movimentare un liquido rispetto un gas).
- Scambiatore centrale (3), utilizzato per incrementare l'efficienza. Serve per recuperare parte del calore per fornirlo all'evaporatore (infatti, il fluido in ingresso si trova ad una temperatura elevata, maggiore di quella che avrà in uscita).

In Figura 7 è riportato uno schema logico con i vari componenti impiegati in un ORC e sopra discussi. In Figura 8, invece, è mostrato il diagramma $T - S$ (temperatura - entalpia) di un ORC ideale (in blu) e reale (in rosso).

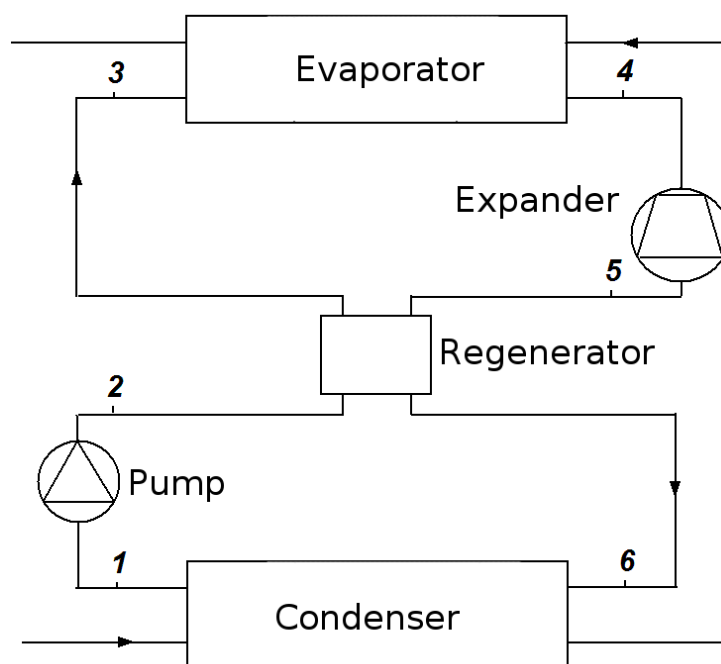


Figura 7: Schema funzionale di un ORC. Per l'immagine si faccia riferimento a [F6]

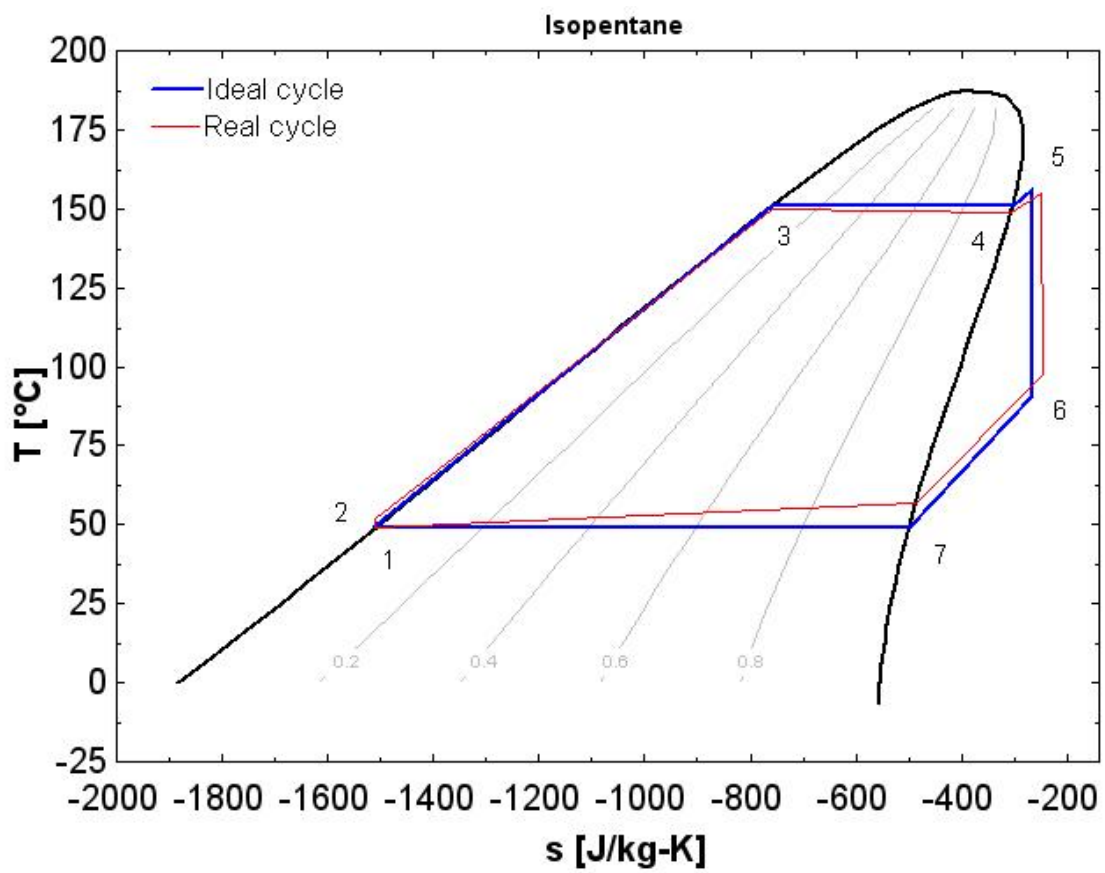


Figura 8: Diagramma $T - S$ per un ORC ideale (blu) e reale (rosso). Per l'immagine si faccia riferimento a [F7]

2.3 Phase change material thermal storage unit, TES

Un materiale a cambiamento di fase è una sostanza in grado di rilasciare (o assorbire, a seconda dei casi), abbastanza energia durante il cambiamento di fase da poter essere utilizzata per il riscaldamento o il raffreddamento. In generale, con *thermal energy storage* (TES) s'intende l'immagazzinamento di calore con l'obiettivo di utilizzarlo poi in un momento futuro [3].

In Figura 9, è mostrato l'andamento tipico del riscaldamento di un materiale a cambiamento di fase: il materiale, allo stato solido, si riscalda fino alla temperatura di fusione T_m (*melting point*); superato questo punto, continuando a riscaldare il materiale (ora allo stato liquido), si raggiungerà la temperatura di evaporazione T_e (*evaporation point*). In Figura 9, in particolare, viene mostrato come il calore necessario per far vaporizzare un materiale (ovvero il calore latente di vaporizzazione), sia significativamente maggiore di quello necessario per la fusione (calore latente di fusione). Per questo motivo, si preferisce utilizzare nelle applicazioni pratiche il passaggio di fase tra solido e liquido.

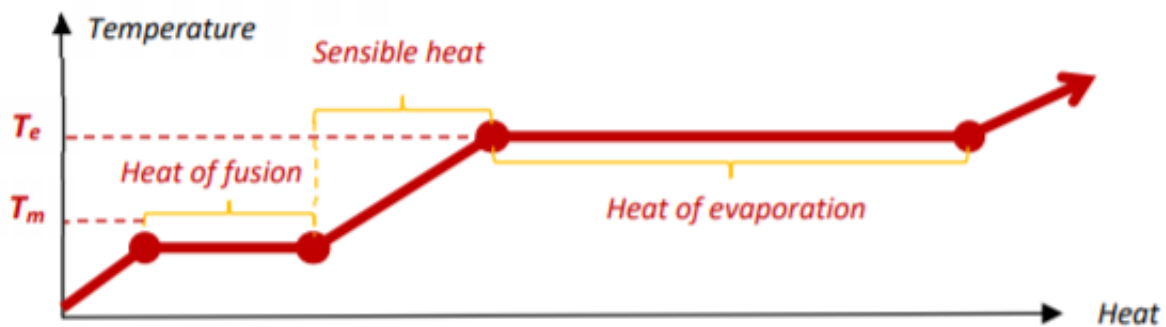


Figura 9: Curva di riscaldamento di un materiale a cambiamento di fase con la temperatura funzione del calore aggiunto, con T_m temperatura di fusione e T_e temperatura di evaporazione. Per l'immagine si faccia riferimento a [F8]

Nell'ambito del progetto, l'accumulo termico dato dal TES viene quindi utilizzato per produrre energia tramite l'ORC nei momenti in cui non c'è irraggiamento solare diretto (l'LFR non produce quindi energia) o quando questo non è abbastanza per utilizzare in modo efficiente l'ORC.

Nel momento in cui, invece, l'LFR eroga una potenza elevata, una parte di questa verrà utilizzata per produrre energia con l'ORC, mentre la restante parte servirà a riscaldare il TES, in modo da poterlo sfruttare in futuro in caso di necessità.

2.4 Operative mode, OM

L'impianto di cogenerazione considerato può lavorare in 8 diverse modalità date dalla configurazione assunta dalle valvole e quindi dalle connessioni tra *linear Fresnel reflector*, *organic Rankine cycle* ed il *thermal energy storage*. L'elenco delle OM (*operative mode*) è riportato in Tabella 1.

OM	Descrizione
OM ₋₁	$T_{TES} \geq T_{max}$, defocalizzazione dell'LFR
OM ₀	Il fluido ricircola nell'LFR per aumentare la sua temperatura
OM ₁	Il fluido va dall'LFR all'ORC che ne sfrutta il calore per produrre energia
OM ₂	Sistema spento
OM ₃	Il fluido dall'LFR circola verso il TES, in modo da fornirgli calore
OM ₄	Il fluido va dall'LFR all'ORC per produrre energia ed al TES per fornirgli calore
OM ₅	Il fluido va dal TES all'ORC per produrre energia
OM ₆	Il fluido va dall'LFR e dal TES all'ORC per produrre energia

Tabella 1: Tabella riassuntiva delle varie modalità operative.

Visivamente, uno schema riassuntivo di quanto riportato in Tabella 1 è mostrato nella Figura 10 (ad eccezione delle modalità OM₋₁ ed OM₀).

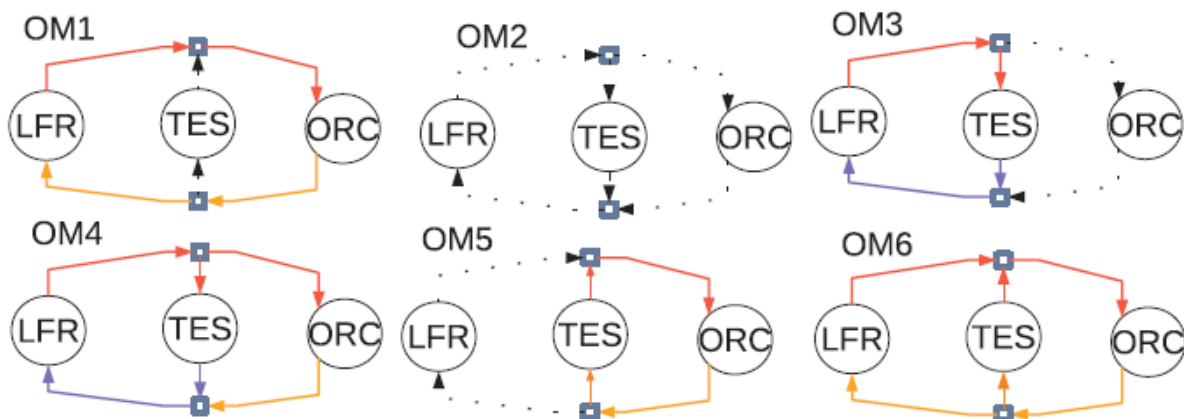


Figura 10: Schema riassuntivo delle modalità operative disponibili per il sistema. Per l'immagine si faccia riferimento a [F2]

Di seguito verrà riportata una descrizione più approfondita di ciascuna fase operativa.

OM -1 Con OM_{-1} s'intende la modalità operativa che si ha nel momento in cui la potenza erogata dall'LFR è abbastanza sia per alimentare l'ORC sia per ricaricare il TES, ma questo ha raggiunto una temperatura pari a quella massima consentita (ovvero $T_{TES} \geq 280^{\circ}\text{C}$). Per evitare danni al *thermal storage unit*, si defocalizza l'LFR fintanto a che T_{TES} non scenda a valori adeguati, ovvero si posizioneranno le lenti del *Linear Fresnel reflector* in modo che queste non siano raggiunte dai raggi solari e quindi non si riscaldi ulteriormente il fluido che circola al loro interno.

OM 0 Con OM_0 s'intende la modalità operativa che si ha nel momento in cui l'LFR genera una potenza che non permette l'utilizzo dell'ORC, che non produrrebbe in maniera efficiente. Per questo motivo, in queste condizioni lavorative del sistema, si fa in modo tramite un'opportuna configurazione delle valvole, di far ricircolare il fluido all'interno dell'LFR, affinché questo si riscaldi ed aumenti la sua temperatura.

OM 1 Con OM_1 s'intende la modalità operativa che si ha nel momento in cui l'LFR produce abbastanza energia da poter utilizzare in modo efficiente l'ORC. Questa è quindi la modalità in cui l'impianto produce energia termica ed elettrica.

OM 2 Con OM_2 s'intende la modalità operativa che si ha quando l'LFR non produce energia e non è possibile né caricare il TES (OM_3), né utilizzare quest'ultimo per produrre con l'ORC (OM_5). Il sistema, in queste condizioni, è quindi spento.

OM 3 Con OM_3 s'intende la modalità operativa per cui il fluido, circolando nell'LFR, è stato riscaldato abbastanza da poter essere utilizzato per ricaricare il TES.

OM 4 Con OM_4 s'intende la modalità operativa per cui si ha una potenza elevata dall'LFR tale da poter utilizzare l'ORC e ricaricare il TES contemporaneamente. In queste condizioni, quindi, il sistema produce energia termica ed elettrica.

OM 5 Con OM_5 s'intende la modalità operativa per cui non si ha potenza generata dall'LFR, ma la temperatura a cui si trova il TES è tale per cui possa essere utilizzata per generare energia tramite l'ORC. In queste condizioni, quindi, il sistema produce energia termica ed elettrica grazie all'unità di accumulo e non grazie all'irraggiamento solare e quindi dall'energia prodotta dall'LFR (al contrario di quanto accade per le modalità OM_1 , OM_4 e, in parte, OM_6).

OM 6 Con OM_6 s'intende la modalità operativa per cui l'energia proveniente dall'LFR non è tale da poter usare in modo efficiente l'ORC, ma il TES ha una temperatura tale da poter essere sfruttato, in combinazione con l'LFR, per utilizzare l'ORC e produrre energia termica ed elettrica.

La logica con cui verranno imposte le varie modalità operative dal controllore implementato durante la simulazione verrà esposta nella sezione 4. I vari profili realizzati per il controllore saranno poi illustrati nelle relative sottosezioni (5.2 in particolare e 5.3).

3 Richiami sulle metodologie impiegate

In questo capitolo sono fornite le base teoriche delle metodologie impiegate per la sintesi del controllore e del supervisore dell'impianto di cogenerazione. In dettaglio in 3.1 vengono richiamati i principi di logica fuzzy, mentre in 3.2 quelli relativi alle reti artificiali.

3.1 Logica fuzzy

Occorre innanzitutto osservare che quanto riportato in questa sezione è stato tratto da [4], [5], [6], [7] e appunti del corso di Metodi e Tecniche per l'Automazione, dove non diversamente specificato.

La logica fuzzy può essere ricondotta all'articolo "Fuzzy sets" scritto da Lofti Zadeh nel 1965 [5]. Da allora, la teoria degli insiemi fuzzy è stata applicata in un range di settori molto vasto oltre quello originario delle scienze umane [8]. Nello specifico, la logica fuzzy, sfocata o sfumata, è un tipo di logica a cui può essere attribuito un grado di verità diverso tra 0 e 1 (come accade con la logica crisp, booleana o esatta) e compreso tra loro, al contrario di quanto accade per gli insiemi classici, in cui la funzione caratteristica d'insieme può assumere solo valori 0 o 1.

Esempio 1 A tal proposito, si può avere come esempio il seguente insieme con relativa funzione caratteristica.

$$R = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$A = \{x \in R : x \leq 5\}$$

$$\mu_A(x) = \begin{cases} 0, & \forall x \notin A \\ 1, & \forall x \in A \end{cases}$$

La curva $\mu_A(x)$, relativamente all'esempio appena illustrato, è riportata in Figura 11.

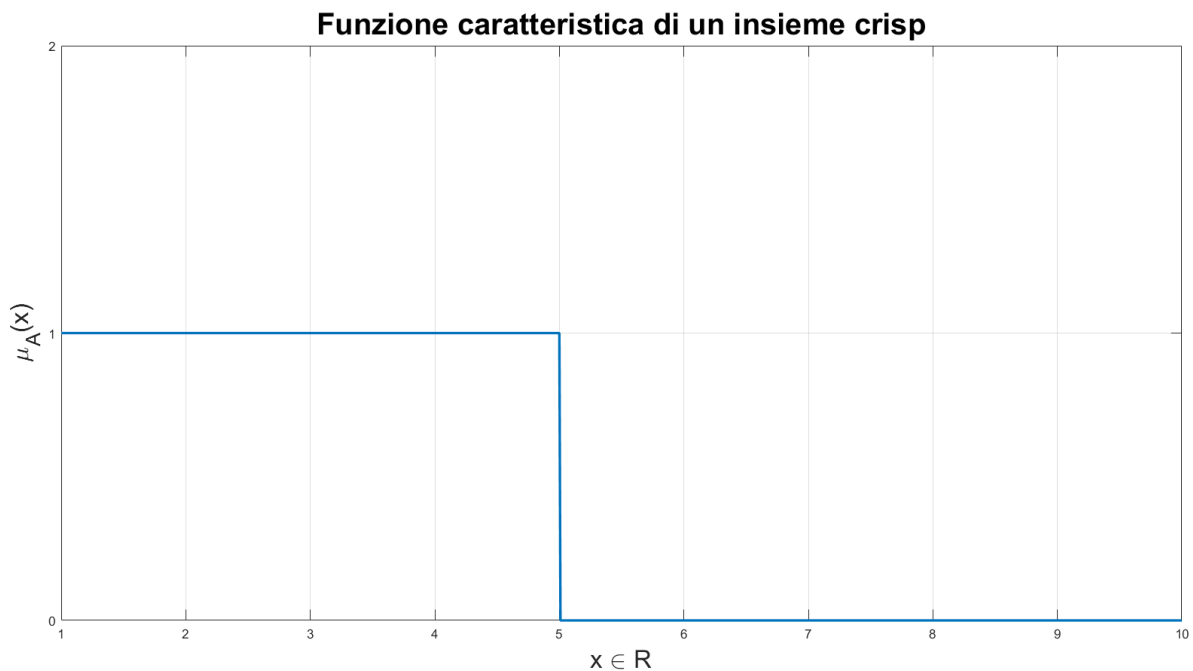


Figura 11: Grafico della funzione di appartenenza $\mu_A(x)$.

Per un insieme fuzzy, invece, la funzione di appartenenza di un elemento all'insieme stesso può assumere diversi valori.

Definizione 1 Se X è una collezione di oggetti x , l'insieme fuzzy A in X è un insieme di coppie ordinate

$$A = \{(x, \mu_A(x)), x \in X\}$$

con $\mu_A(x)$ funzione di appartenenza di x ad A tale che $\mu_A(x) : X \rightarrow M$ e M insieme di numeri reali non negativi con estremo superiore finito.

Esempio 2 Si consideri come primo esempio di logica fuzzy il seguente insieme:

$A =$ numeri reali *molto più grandi* di 10

$$A = \{(x, \mu_A(x)), x \in X \equiv \mathbb{R}\}$$

Non è possibile associare il concetto espresso dall'insieme A ad un insieme crisp. Per questo motivo si può far ricorso alla logica fuzzy. In questo caso, man mano che x aumenta, aumenterà anche il grado di verità di $\mu_A(x)$. Ne segue che esistono infiniti valori possibili tra $[0, 1]$ (in questo caso si avrà $\mu_A(x) \rightarrow 1$ per $x \rightarrow +\infty$). A tal proposito, il grado di verità della frase (o anche il grado di appartenenza di x all'insieme stesso) sarà dato da:

$$\mu_A(x) = \begin{cases} 0, & x \leq 10 \\ f(x), & x > 10 \end{cases}$$

Si può considerare, ad esempio, $f(x) = [1 + (x - 10)^{-2}]^{-1}$, ottenendo così il grafico mostrato in Figura 12.

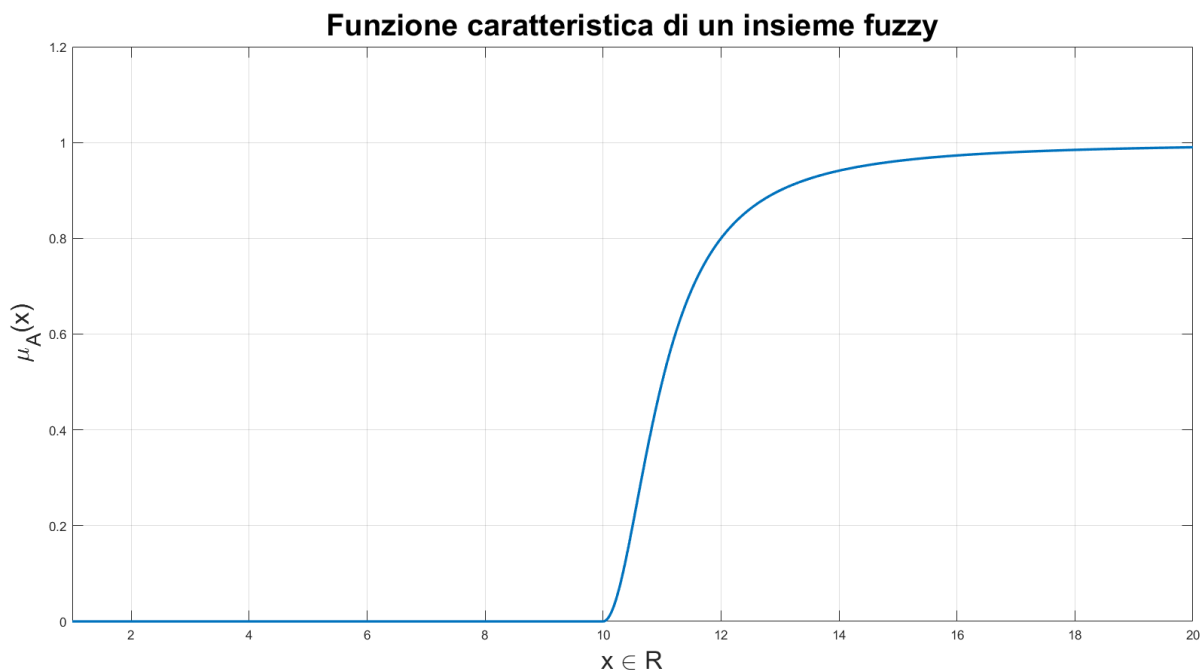


Figura 12: Grafico della funzione di appartenenza $\mu_A(x)$.

Occorre altresì notare che la curva caratteristica e quindi la funzione $\mu_A(x)$, nella logica fuzzy non è unica, ma esistono invece infinite funzioni caratteristiche (ovvero tutte quelle compatibili con il concetto espresso dall'insieme). Infatti, concetti come "molto più grande di", "vicino a", "circa uguale", ecc. possono essere espressi da diverse funzioni. Questo aspetto rappresenta un vantaggio della logica fuzzy in quanto non vincola la scelta della funzione caratteristica (sulla base di questa scelta, si avranno risultati non univoci, ma comunque corretti).

Sulla base degli esempi precedenti, si può vedere che per un insieme fuzzy $\mu_A(x)$ rappresenta il grado con cui x verifica il concetto espresso da A (così come vale anche per gli insiemi crisp, ma limitatamente ai soli valori logici di "vero" o "falso").

Mentre non esistono particolari limitazioni sulla forma delle funzioni d'appartenenza, gli insiemi fuzzy più utilizzati nel controllo sono:

- insieme singleton;
- insieme triangolare;
- insieme trapezoidale;
- insieme a campana, con

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ e^{-\frac{1}{2}(x-b)^2\sigma^{-2}}, & a < x < c \\ 0, & x \geq c \end{cases}$$

Definizione 2 Un insieme fuzzy è detto normale se $\sup_x \mu_A(x) = 1$.

Un insieme non normale può essere sempre normalizzato dividendo per $\sup_x \mu_A(x)$.

Definizione 3 Il supporto di un insieme fuzzy è l'insieme (crisp, operatore di confronto) di tutti gli elementi di X tali che $\mu_A(x) > 0$.

Più in generale, viene definito l'insieme crisp $A_\alpha = \{x \in X : \mu_A(x) \geq \alpha\}$, detto insieme di livello α corrispondente all'insieme fuzzy A .

Definizione 4 Un insieme fuzzy è convesso se la sua funzione di appartenenza $\mu_A(x)$ è convessa, ovvero $\mu_A[\lambda x_1 + (1 - \lambda)x_2] \geq \min[\mu_A(x_1), \mu_A(x_2)]$, con $x_1, x_2 \in X, \lambda \in [0, 1]$.

3.1.1 Operazioni sugli insiemi fuzzy

Nella logica fuzzy, le operazioni fondamentali (ovvero unione, intersezione e complementazione) sono definite in modo differente rispetto agli insiemi classici.

A tal proposito, sia

$$S = \{(x, \mu_S(x)), x \in X\} \text{ e } T = \{(x, \mu_T(x)), x \in X\}$$

Con

$\mu_S(\bar{x})$: grado di verità della frase " \bar{x} soddisfa il requisito espresso da S " (ovvero $\bar{x} \in S$).

$\mu_T(\bar{x})$: grado di verità della frase " \bar{x} soddisfa il requisito espresso da T " (ovvero $\bar{x} \in T$).

A questo punto, è logico considerare l'intersezione $S \cap T = V$ come l'insieme fuzzy

$$V = \{(x, \mu_{S \cap T} \triangleq f(\mu_S, \mu_T)), x \in X\}$$

dove $f(\mu_S, \mu_T)$ rappresenta il grado di verità della frase " x soddisfa contemporaneamente i requisiti espressi da S e T " (ovvero $x \in V$).

Analogamente si definisce l'unione $S \cup T = Z$ come

$$Z = \{(x, \mu_{S \cup T} \triangleq g(\mu_S, \mu_T)), x \in X\}$$

dove $g(\mu_S, \mu_T)$ rappresenta il grado di verità della frase " x soddisfa almeno uno dei requisiti espressi da S e T " (ovvero $x \in Z$).

Proprietà Le funzioni f (per l'intersezione) e g (per l'unione) devono soddisfare le seguenti limitazioni:

1. f e g sono non decrescenti e continue sia rispetto μ_S che μ_T ;
2. f e g sono simmetriche, ovvero

$$f(\mu_S, \mu_T) = f(\mu_T, \mu_S)$$

$$g(\mu_S, \mu_T) = g(\mu_T, \mu_S)$$
3. $f(\mu_S, \mu_T)$ e $g(\mu_S, \mu_T)$ sono strettamente crescenti rispetto μ_S ;
4. $f(\mu_S, \mu_T) \leq \min\{\mu_S, \mu_T\}$, ovvero accettare contemporaneamente "veri" S e T richiede di più che accettare come vero solo S o solo T ;
5. $g(\mu_S, \mu_T) \geq \max\{\mu_S, \mu_T\}$, ovvero accettare come "vero" solo S o solo T è sicuramente non inferiore al più grande tra i gradi di verità del solo S e del solo T ;
6. $f(1, 1) = 1$ e $g(0, 0) = 0$;
7. f, g, h devono essere tali che

$$S \wedge (T \vee K)$$
 Sia equivalente a

$$(S \wedge K) \vee (S \wedge T)$$
 (proprietà distributiva dell'unione rispetto l'intersezione).

Occorre osservare che $\min\{\}$ e $\max\{\}$ non sono gli unici operatori utilizzabili.

Per l'unione, ad esempio, una scelta altrettanto valida è quella di utilizzare il prodotto, ovvero:

$$S \cup T = \mu_S \mu_T$$

Sebbene accettabile dal punto di vista logico e dei risultati prodotti, tuttavia, una scelta di questo tipo non è utilizzata nel controllo in quanto potrebbe comportare una perdita di informazioni.

Ad esempio, si consideri il seguente caso:

$$\mu_S = 0.1; \mu_T = 0.2 \rightarrow S \cup T = \mu_S \mu_T = 0.02$$

Esempio 3 Di seguito sono riportati due esempi.

Si considerino gli insiemi fuzzy A e B del tipo riportato in Figura 13 e 14.

Si consideri dapprima l'insieme $C = A \cap B$ (Figura 13) e poi $C = A \cup B$ (Figura 14).

Il primo caso fa quindi riferimento all'operazione di intersezione di due insiemi fuzzy, mentre il secondo all'unione.

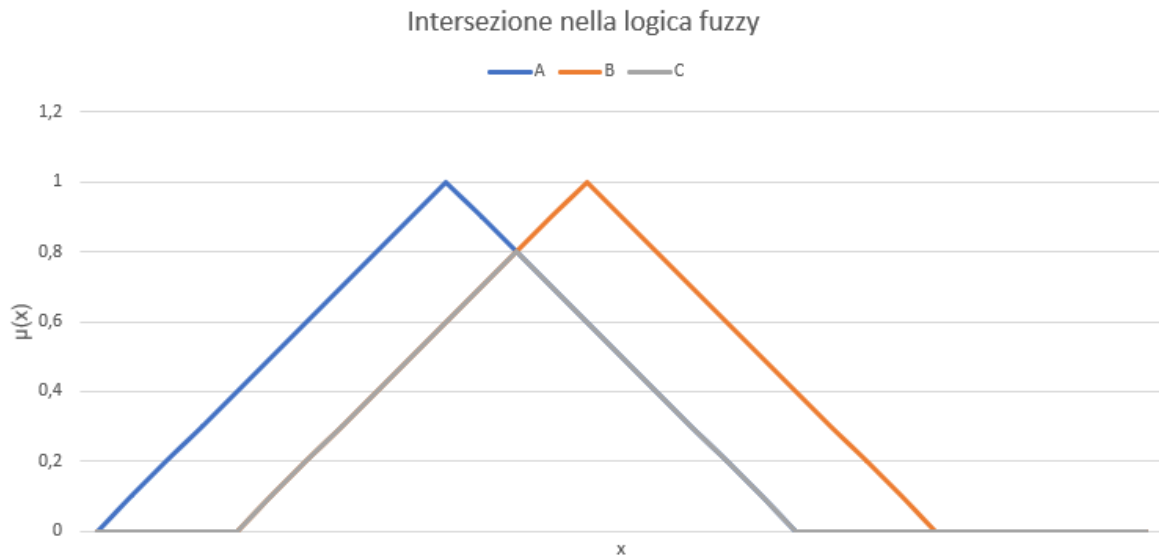


Figura 13: Risultato dell'intersezione $C = A \cap B$.

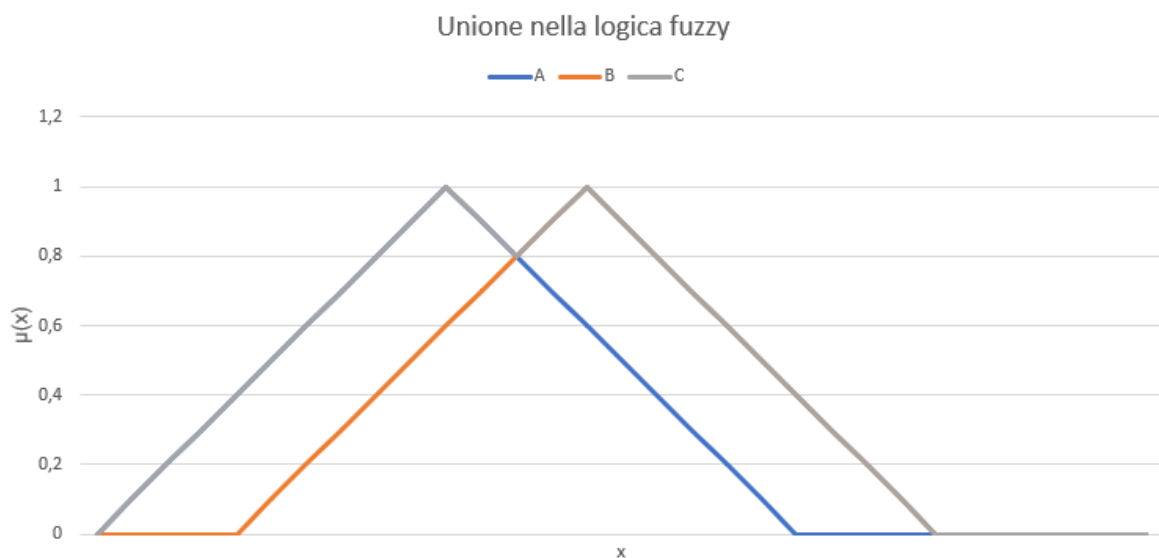


Figura 14: Risultato dell'unione $C = A \cup B$.

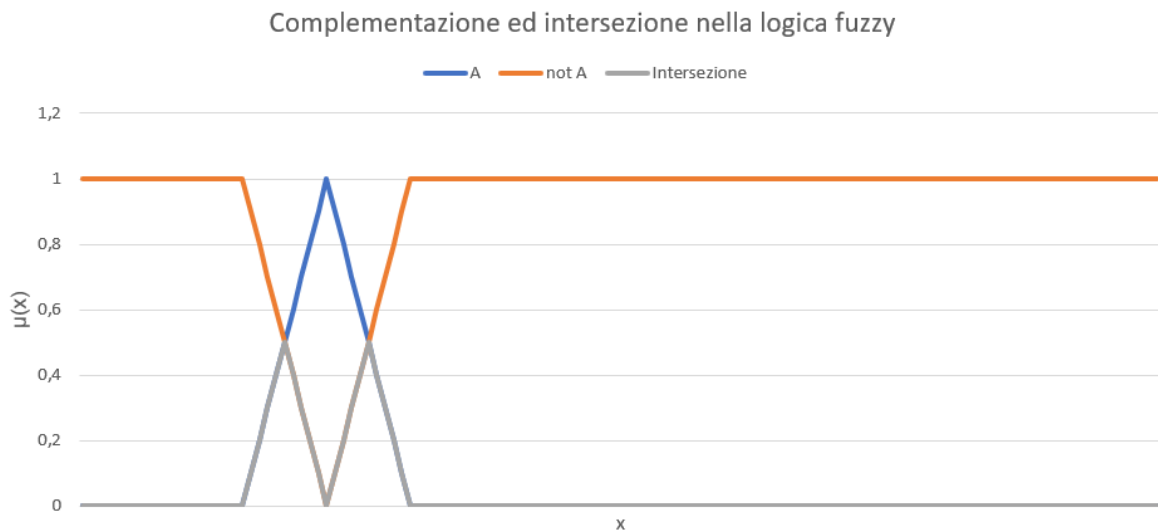
In entrambi i casi si avrà $C = \{(x, \mu_C(x)), x \in X\}$. Per l'intersezione il supporto di C è l'intervallo (c, b) , mentre per l'unione si avrà l'intervallo (a, d) .

Per la complementazione di un insieme fuzzy, si faccia riferimento ad $A = \{(x, \mu_A(x)), x \in X\}$. Si avrà che $\mu_A(x)$ rappresenta il grado con cui x soddisfa il requisito espresso da A , quindi $1 - \mu_A(x)$ rappresenta il grado con cui x non soddisfa il requisito espresso da A . Allora:

$$\bar{A} = \{(x, 1 - \mu_A(x)), x \in X\}$$

Molte delle proprietà valide per gli insiemi classici lo sono anche per gli insiemi fuzzy (utilizzando gli operatori precedenti). Occorre però considerare il caso di $A \cap \bar{A}$ e $A \cup \bar{A}$, con A un insieme fuzzy qualunque: come conseguenza del linguaggio e dei concetti sfumati di questa logica, un elemento può soddisfare, in una certa misura, sia A che \bar{A} .

Esempio 4 Siano A e B due insiemi fuzzy di elementi $x \in X$ del tipo riportato in Figura 15 e 16. Sia inoltre $C = A \cap \bar{A}$ e $C = A \cup \bar{A}$, rispettivamente in Figura 15 e 16.



Si può osservare come, al contrario di quanto accade per gli insiemi crisp, $C \neq \emptyset$ nel caso di intersezione (Figura 15) e $C \neq X$ nel caso di unione (Figura 16).

3.1.2 Relazione fuzzy

Sia per gli insiemi classici che per quelli fuzzy, la relazione è definita nel prodotto cartesiano $X \times Y$, ovvero $R : X \times Y \rightarrow [0, 1]$ ed assegna un grado di appartenenza $\mu_R(x, y)$ ad ogni coppia $(x, y) \in X \times Y$, quindi è anch'essa un insieme fuzzy:

$$R = \{(x, y), \mu_R(x, y)\}, (x, y) \in X \times Y\}$$

L'interpretazione di $\mu_R(x, y)$ è del tutto analoga a quelle di una qualsiasi altra funzione di appartenenza: tanto più il suo valore è alto, tanto più (x, y) soddisfa il requisito dato da R e quindi tanto più è vera l'esistenza di un legame tra x ed y .

Esempio 5 Si vuole ora esprimere il concetto di "x è approssimativamente uguale ad y". Si consideri a tal proposito l'insieme $U = \{1, 2, 3, 4\}$. La relazione R definita su $X \times X$ risultante è mostrata nella Tabella 2.

	1	2	3	4
1	1	0.5	0	0
2	0.5	1	0.5	0
3	0	0.5	1	0
4	0	0	0.5	1

Tabella 2: Relazione R .

3.1.3 Variabili linguistiche

La variabile linguistica è uno strumento alla base della logica fuzzy e del ragionamento approssimato in generale.

Definizione 5 Una variabile linguistica è caratterizzata dalla quintupla $(x, T(x), U, C, M)$ in cui:

- x è il nome della variabile;
- $T(x)$ è l'insieme dei valori linguistici di x ;
- U è l'universo del discorso;
- C è la regola sintattica che genera i nomi appartenenti a $T(x)$;
- M è la regola semantica per associare a ciascun nome il suo significato, ovvero un insieme fuzzy.

Alle variabili linguistiche possono venire associati anche dei modificatori linguistici, ovvero termini del tipo *leggermente*, *molto*, *circa*, ecc. Questi elementi hanno come effetto la modifica dei nomi e di $T(x)$, e quindi anche il relativo significato.

A livello di modelli matematici, i modificatori più utilizzati sono:

- "molto A " $\rightarrow A^2$;
- "più o meno A " $\rightarrow A^{1/2}$;
- "più A " $\rightarrow A^{5/4}$;
- "meno A " $\rightarrow A^{3/4}$.

3.1.4 Inferenza fuzzy

Data una legge ed osservato un fatto, si ha il meccanismo di inferenza (fuzzy in questo caso). Ovvero, l'inferenza fuzzy è il processo di formulazione del legame tra un dato ingresso ed il relativo output usando la logica fuzzy. Il legame è poi la base da cui possono essere prese decisioni [S12].

Uno schema logico di applicazione di una processo di inferenza fuzzy è mostrato in Figura 17.

Nella stessa Figura occorre inoltre osservare come gli input siano dei valori crisp (non fuzzy) e da essi, valutando in simultanea le regole ed utilizzando il ragionamento fuzzy, si giunga ad un risultato anch'esso crisp dopo una combinazione e defuzzificazione dei valori ottenuti dall'applicazione delle regole.

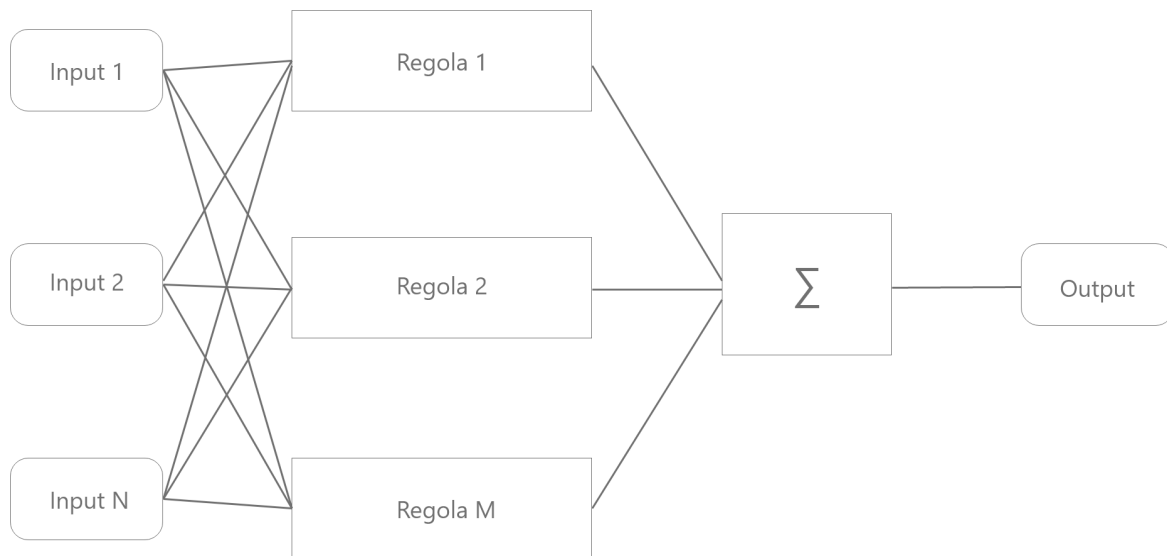


Figura 17: Schema logico del meccanismo di inferenza fuzzy.

Nella logica classica esiste il "Modus ponens" (MP) dato da una premessa ($x \in A$), un'implicazione (se $x \in A$, allora $y \in B$) e una conseguenza ($y \in B$).

Nella logica fuzzy, invece, si utilizza il "Modus ponens generalizzato" (GMP) in cui si avranno una premessa ($x \in A'$), un'implicazione (se $x \in A$, allora $y \in B$) e una conseguenza ($y \in B'$), con l'importante distinzione che $A \neq A'$ e $B \neq B'$.

Nella logica classica, una proposizione è vera o falsa, al contrario di quanto accade nella logica fuzzy, in cui può assumere anche i valori intermedi di verità. Inoltre, nella logica classica è possibile fare una deduzione solo quando il dato di ingresso soddisfa la premessa, cosa che non accade nella logica fuzzy.

Esempio di MP Di seguito è mostrato un esempio di "Modus ponens".

Premessa: *Socrate è un uomo*, $x \in A$.

Implicazione: *se Socrate è un uomo, allora è mortale*, se $x \in A$, allora $y \in B$.

Conseguenza: *Socrate è mortale*, $y \in B$.

Questo tipo di inferenza, come detto, è tipico della logica crisp ed anche i concetti espressi saranno crisp (x ed y).

Esempio di MPG Di seguito è mostrato un esempio di "Modus ponens generalizzato".

Premessa: *questo pomodoro è molto rosso*, $x \in A'$.

Implicazione: *se un pomodoro è rosso allora è maturo*, se $x \in A$, allora $y \in B$.

Conseguenza: *questo pomodoro è molto maturo*, $y \in B'$.

Dagli esempi, si può osservare come la logica fuzzy sia più elastica rispetto quella classica: non c'è più un legame stringente tra premessa ed implicazione come accade invece nella logica crisp (ma si possono comunque dedurre delle informazioni).
Si può quindi dire che il MP è un caso particolare di GMP.

3.1.5 Sistema di inferenza Mamdani

Il sistema di inferenza Mamdani (ovvero del tipo di quello implementato nella presente tesi) è stato inizialmente proposto come un primo tentativo di controllo di un motore a vapore e caldaia a partire da un insieme di regole ottenute dall'esperienza degli operatori (1973) [9]. Segue quindi che questo tipo di inferenza sintetizza un sistema di controllo a partire da un insieme di leggi di controllo linguistiche dettate dall'esperienza umana. Questo sistema d'inferenza, come tutti gli altri del resto, utilizza regole del tipo "if ... then" associate a connettori logici del tipo "and" piuttosto che "or" per determinare le regole decisionali. Prima di scendere nel dettaglio di un sistema di inferenza relativo al controllo, occorre specificare che il metodo Mamdani presenta i seguenti vantaggi rispetto altri tipi di metodi d'inferenza come quello di Takagi-Sugeno-Kang) [S13]:

- è intuitivo;
- si presta bene ad input provenienti dall'esperienza umana;
- ha una base delle regole facilmente interpretabile;
- ha un'accettazione diffusa.

In Figura 18 è riportato lo schema logico di un controllore in logica fuzzy e relativo sistema di inferenza.

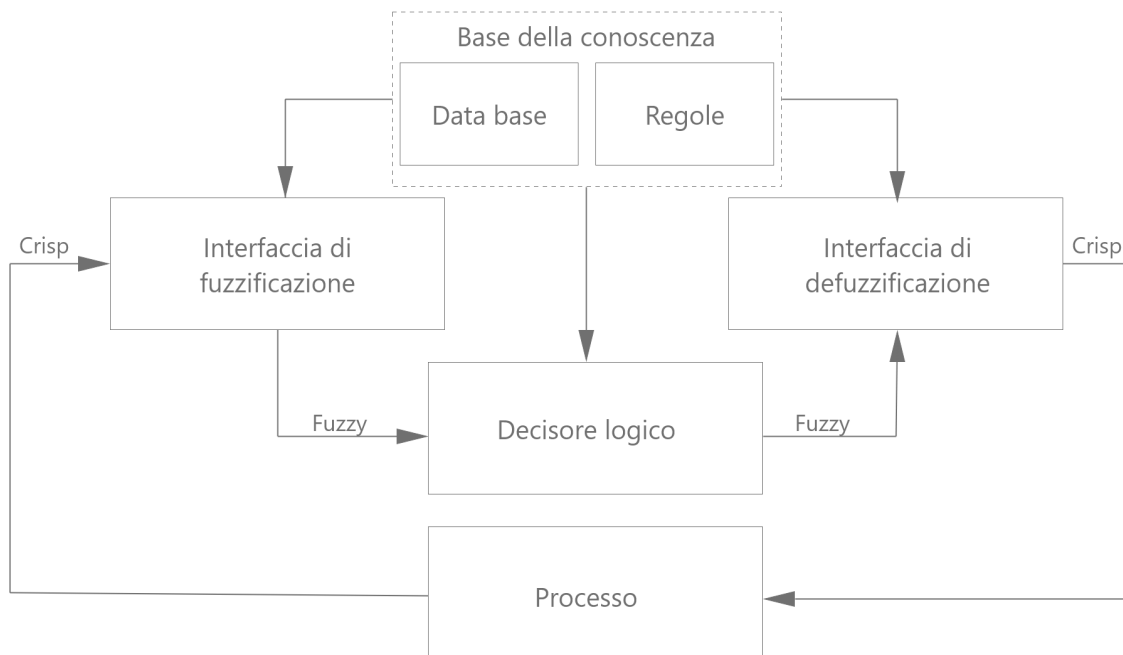


Figura 18: Schema logico di un sistema di controllo fuzzy.

Facendo sempre riferimento alla Figura 18, il blocco *base della conoscenza* è dato dalla descrizione linguistica del processo più eventuali specifiche, quello relativo all'*interfaccia di fuzzificazione* ha invece il compito di trasformare delle informazioni generiche crisp in informazioni fuzzy, mentre il blocco di *interfaccia di defuzzificazione* ha il compito inverso, ovvero trasformare quantità fuzzy in crisp; infine, il *decisore logico* rappresenta il controllore logico fuzzy con il compito di interpretare le regole fuzzy per stabilire di conseguenza le operazioni più adeguate [S14].

Di seguito sono descritte nel dettaglio le funzionalità dei blocchi di Figura 18.

- **Interfaccia di fuzzificazione**
 - Misura i valori delle variabili di ingresso (facendo riferimento ad un sistema di controllo, questo blocco conterrà la sensoristica);
 - Effettua una scalatura delle variabili;
 - Esegue la fuzzificazione, convertendo come precedentemente accennato i dati in ingresso in opportuni valori linguistici (che possono essere visti come nomi di insiemi fuzzy).
- **Base della conoscenza**, contiene la conoscenza empirica del processo e contiene a sua volta due blocchi.
 - Base dei dati, contiene le informazioni necessarie per definire le regole linguistiche per il controllo e la manipolazione dei dati;
 - Base delle regole, contiene i risultati desiderati e la politica di controllo implementata attraverso una serie di regole linguistiche (ovvero contiene le specifiche di controllo).
- **Decisore logico**, deduce le azioni di controllo a partire dalle regole e dagli insiemi fuzzy memorizzati in esso.
- **Interfaccia di defuzzificazione**
 - Converte i valori linguistici fuzzy in valori numerici (ovvero crisp);
 - Esegue la denormalizzazione delle variabili di uscita.

Di seguito sono riportate delle caratterizzazioni più dettagliate di ogni singolo blocco appena descritto.

Interfaccia di fuzzificazione L'interfaccia di fuzzificazione converte, nel caso di ingressi esatti, ovvero noti con esattezza, il valore numerico misurato in un insieme fuzzy *singleton*. Successivamente esegue una scalatura delle variabili in ingresso (passo non obbligatorio, ma utile per uniformità di linguaggio). In particolare, l'universo del discorso viene discretizzato in un insieme finito di segmenti, ad ognuno dei quali viene fatto corrispondere un appropriato segmento dell'universo normalizzato. Quindi, l'interfaccia di fuzzificazione opera una rappresentazione delle informazioni in termini di insiemi fuzzy (questo passaggio è soggettivo e basato sull'esperienza che si ha sul sistema). La procedura più utilizzata è quella di effettuare una scalatura come appena discusso ottenendo vari livelli di quantizzazione; a ciascun livello viene assegnata un'opportuna etichetta (*piccolo, molto grande, ecc.*) e, infine, si associa un insieme fuzzy ad ogni elemento dell'universo discretizzato attraverso una regola M , ovvero una funzione caratteristica.

Occorre osservare che la scelta dei livelli di quantizzazione ed il loro numero influiscono in modo importante sulla sensibilità del controllo. In genere, si può avere una risoluzione grossolana per valori grandi, mentre si avrà una maggiore risoluzione per valori piccoli. Inoltre, deve valere la proprietà di completezza, ovvero l'unione dei supporti degli insiemi fuzzy primari deve coprire l'intero universo del discorso.

Base della conoscenza Un sistema fuzzy è caratterizzato, come precedentemente detto, da un insieme di frasi condizionali del tipo "*se ... allora ...*". Risulta quindi importante scegliere le variabili linguistiche che vengono utilizzate nelle frasi condizionali. Tipicamente, le variabili possono essere lo stato del processo, l'errore sullo stato o sull'uscita, la derivata dell'errore, ecc..

Decisore logico Partendo dalle funzioni di implicazione fuzzy, dal meccanismo utilizzato per la composizione e dall'interpretazione dei connettori logici (per l'intersezione e l'unione), stabilisce l'insieme fuzzy al quale appartiene l'azione di controllo.

Interfaccia di defuzzificazione Con defuzzificazione s'intende la mappatura da uno spazio di azioni (di controllo) fuzzy in uno spazio di azioni (di controllo) non fuzzy, ovvero crisp. Il defuzzificatore produce l'azione di controllo crisp che meglio risponde alla funzione di appartenenza $C' = \cup_{i=1}^n C'_i$. Esistono diverse possibilità in questo ambito.

- Criterio del massimo, sceglie come azione di controllo il valore che, nell'universo delle uscite, corrisponde al massimo valore della funzione di appartenenza del meccanismo di inferenza (o, nel caso di più massimi, la media tra questi);
- *Mean of Maxima Method (MOM)*, genera come azione di controllo quella che corrisponde al valor medio di tutte le azioni di controllo corrispondenti a minimi locali;
- *Center of Area Method (COA)*, considera come azione di controllo l'ascissa corrispondente al centro di gravità della figura data dall'insieme fuzzy corrispondente.

Esempio 6 Come esempio di quanto appena illustrato, si faccia riferimento all'immagine riportata in Figura 19.

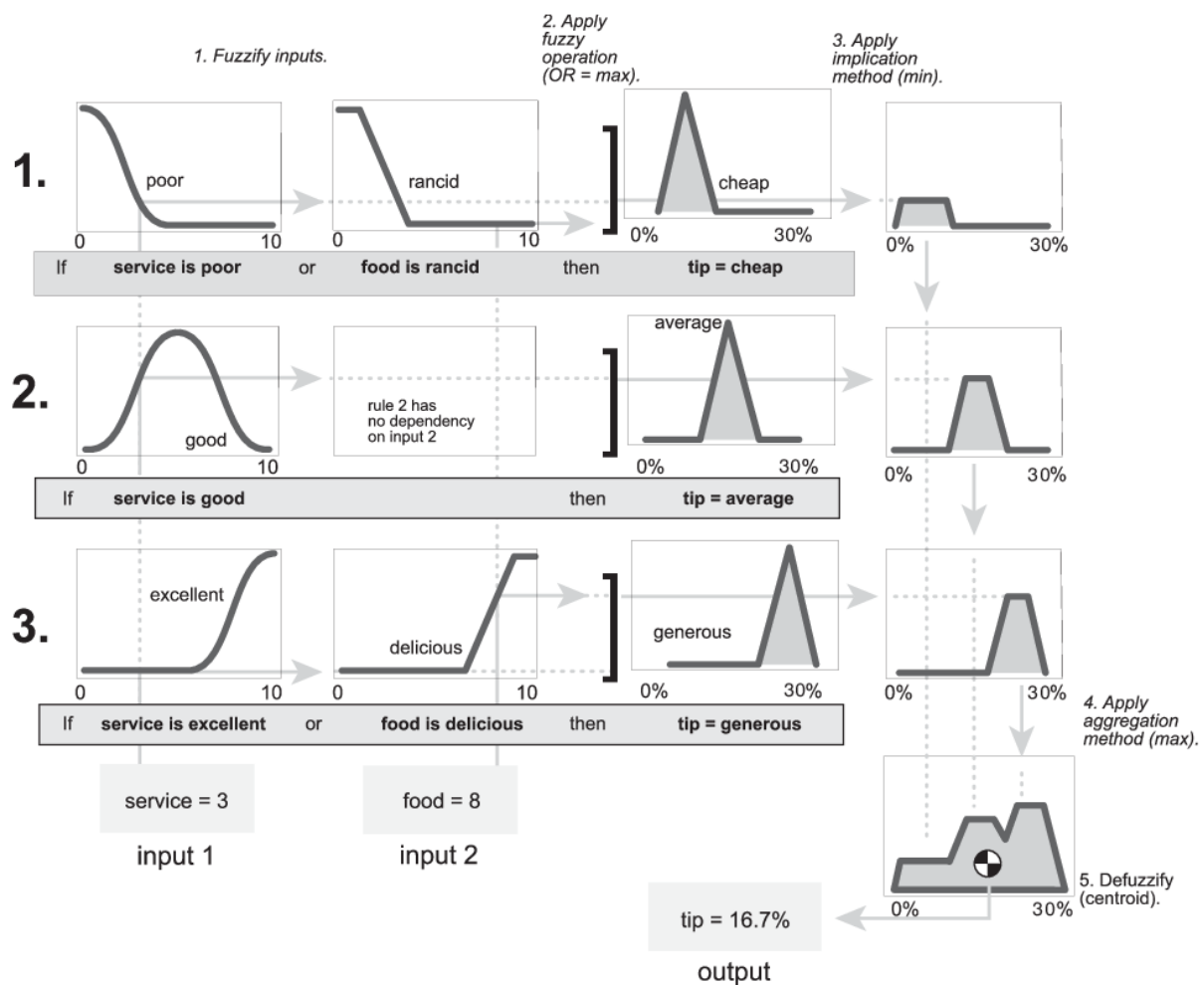


Figura 19: Esempio di applicazione del meccanismo di inferenza fuzzy. Per l'immagine si faccia riferimento a [F10]

L'esempio [S15] mostra come dapprima si proceda alla fuzzificazione dell'input (in questo caso dato da 3 variabili) per poi applicare l'operatore logico (OR) opportuno (passi 1 e 2 in Figura 19). Successivamente, si applicherà il metodi di implicazione appropriato (*min* nel caso dell'esempio). Infine, avendo che tutte e 3 le variabili di input concorrono alla decisione di quale azione scegliere, occorrerà combinare tutti i singoli insiemi fuzzy in un unico insieme fuzzy. L'operazione prende il nome di aggregazione (e corrisponde all'operatore *max*). Infine si utilizzerà uno dei metodi di defuzzificazione prima descritti per ottenere un valore crisp.

Osservazione Infine, occorre osservare che, con la logica fuzzy, non si riesce a determinare la stabilità di un sistema (si avrà un sistema altamente non lineare a cui non si può applicare la teoria di Lyapunov).

3.2 Reti neurali

Occorre innanzitutto osservare che quanto riportato in questa sezione è stato tratto da [10], [11] e appunti del corso di Intelligenza Artificiale, dove non diversamente specificato.

Nell'ambito dell'apprendimento automatico (o *machine learning*), una rete neurale (o anche ANN o NN, dall'inglese *artificial neural network* o semplicemente *neural network*) è un modello matematico composto da una struttura stratificata di neuroni artificiali che trae ispirazione dal funzionamento (semplificato) di una rete neurale biologica e quindi del cervello umano. Una rete neurale suddivide gli input in livelli di astrazione apprendendo dai dati stessi. Può quindi essere addestrata a riconoscere pattern, classificare dati o predire eventi futuri. Può essere addestrata a riconoscere immagini o i pattern del parlato (proprio come il cervello umano). Il comportamento di una rete neurale dipende dal modo in cui sono collegati i singoli elementi che la compongono, oltre che dai pesi delle connessioni stesse. Proprio i pesi sono gli elementi che vengono automaticamente modificati durante l'addestramento in base ad una specifica regola di addestramento, il tutto fino a che la rete neurale non esegue in modo soddisfacente il compito richiesto [S16].

Le reti neurali sono particolarmente adatte a svolgere compiti di riconoscimento di pattern per identificare e classificare oggetti o segnali, nella visione, nei sistemi di controllo o nella previsioni e modellazione di serie storiche. Alcuni esempi di applicazione di reti neurali sono [S16] [S17] [12]:

- servizio di raccomandazione delle piattaforme di streaming quali Netflix, Spotify, YouTube, ecc.;
- motori di ricerca come Google;
- assistenti vocali, ad esempio Google, Alexa, ecc.;
- finanza, per previsioni sulla gestione dei portafogli, valutazione dei titoli obbligazionari e azionari, ecc.;
- sistemi avanzati di assistenza alla guida, ad esempio per il riconoscimento della corsia e dei segnali stradali;
- in ambito medico per effettuare delle diagnosi.

A livello biologico, il neurone è l'unità fondamentale del tessuto nervoso, in cui avviene l'elaborazione dell'informazione. Un'illustrazione schematica di un neurone e delle sue parti essenziali è mostrata in Figura 20.

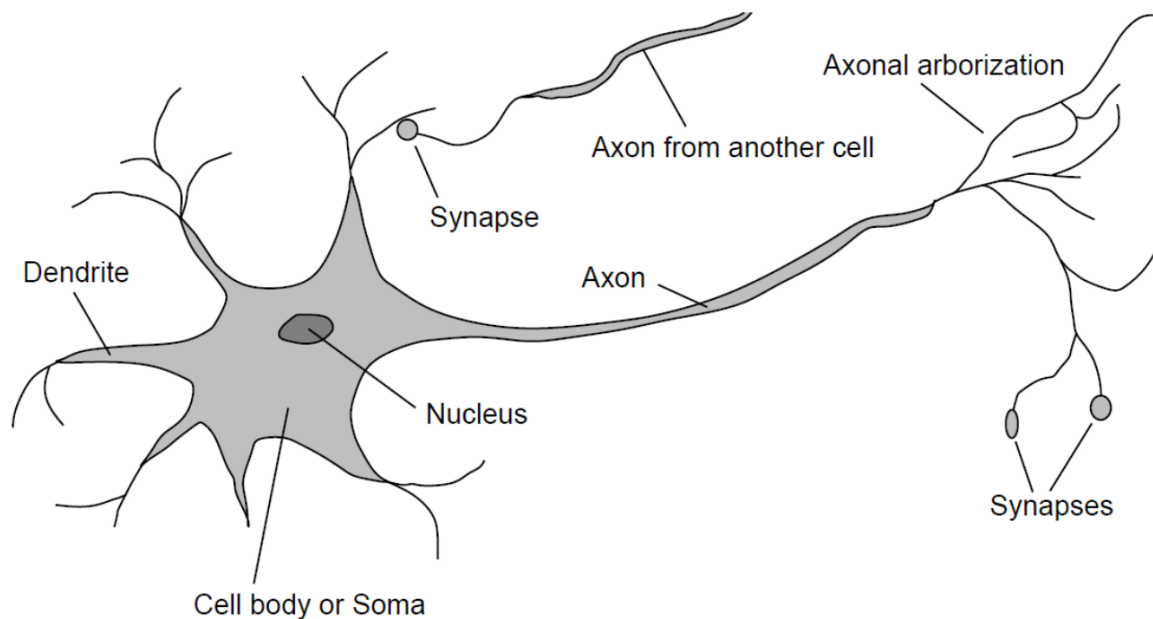


Figura 20: Esempio schematico di un neurone.

In un neurone sono presenti sostanze che fungono da trasmettitori chimici che vengono rilasciate dalle sinapsi (ovvero delle giunzioni tra neuroni) entrando poi nei dendriti (gli ingressi del neurone). Lo scopo di questi trasmettitori è quello di alzare o abbassare il potenziale elettrico del neurone. Quando il potenziale elettrico raggiunge una certa soglia, cioè il potenziale di attivazione del neurone, viene inviato un impulso elettrico lungo l'assone (l'uscita del neurone). Tale impulso corre lungo le ramificazioni dell'assone connesse tramite le sinapsi ai dendriti di altri neuroni. Questo provoca il rilascio dei trasmettitori chimici nei corpi di altre cellule.

Per quanto riguarda un neurone artificiale, invece, si può fare riferimento al suo modello matematico semplificato dato da:

$$y = f \left[\sum_{i=1}^n (w_i x_i - \theta) \right]$$

Il neurone, in questo caso, esegue la somma pesata degli n input (ovvero x_i con relativo peso w_i) che, in analogia con il neurone biologico, ne rappresentano il potenziale di attivazione. La funzione di attivazione non lineare f con ingresso a soglia θ trasforma la somma pesata degli ingressi nel corrispettivo valore di uscita y .

Il perceptrone è un modello di neurone artificiale proposto da Rosenblatt nel 1958 a partire dal lavoro di McCulloch e Pitts in cui viene definito il primo modello logico-matematico della corteccia cerebrale (1943) [13] [14]. Un perceptrone è un classificatore binario a due strati, uno di ingresso ed uno di uscita (Figura 21).

$$y = \begin{cases} 0, \sum_{i=1}^n w_i x_i \leq \theta \\ 1, \sum_{i=1}^n w_i x_i > \theta \end{cases}$$

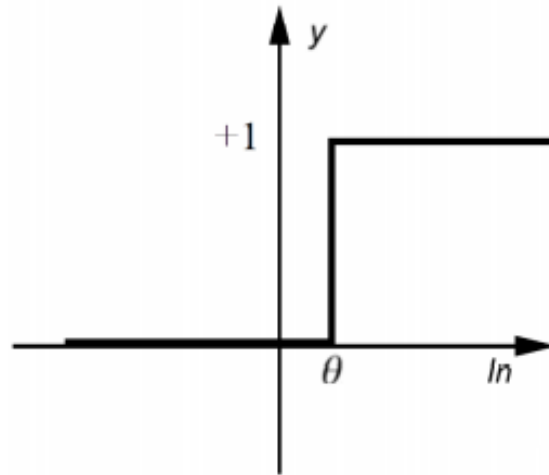


Figura 21: Funzione di attivazione un perceptrone.

A livello pratico, un perceptrone può risolvere problemi linearmente separabili, ovvero problemi per cui all'interno dello spazio di input esiste un interpiano che separa gli elementi delle due classi. A tal proposito, si faccia riferimento all'esempio mostrato in Figura 22.

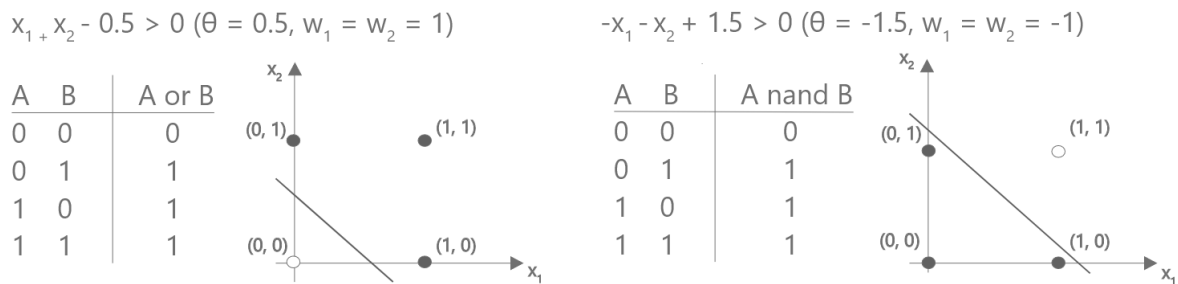


Figura 22: Esempio di funzioni rappresentabili con un perceptrone.

Il perceptrone a soglia è quindi chiamato separatore lineare e, in generale, un singolo perceptrone non può rappresentare tutte le funzioni booleane, ma solo un sottoinsieme di queste (*AND*, *OR* e *NOT*) [S19]. Per poter trattare anche problemi non lineari, occorre utilizzare una rete di perceptroni basata su tre strati diversi:

- Strato di ingresso, che contiene gli input della rete;
- Strato di uscita, che contiene le uscite della rete;
- Strato nascosto, che contiene i neuroni posizionati tra lo strato di ingresso e quello di uscita.

Un esempio è mostrato in Figura 23.

Occorre inoltre osservare che la funzione di attivazione a soglia non è l'unica utilizzabile. Altre opzioni possibili come funzione di attivazione sono:

- sigmoide, $y = \frac{1}{1+e^{-x}}$; $x = \sum_{i=1}^n (w_i x_i - \theta)$;
- tangente iperbolica, $y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$; $x = \sum_{i=1}^n (w_i x_i - \theta)$.

In particolare, per la funzione di attivazione sigmoide, si possono fare le seguenti osservazioni:

- $y \rightarrow 1$ per $x \rightarrow +\infty$. Per grandi valori di x , quindi, l'output di un neurone tende ad 1;

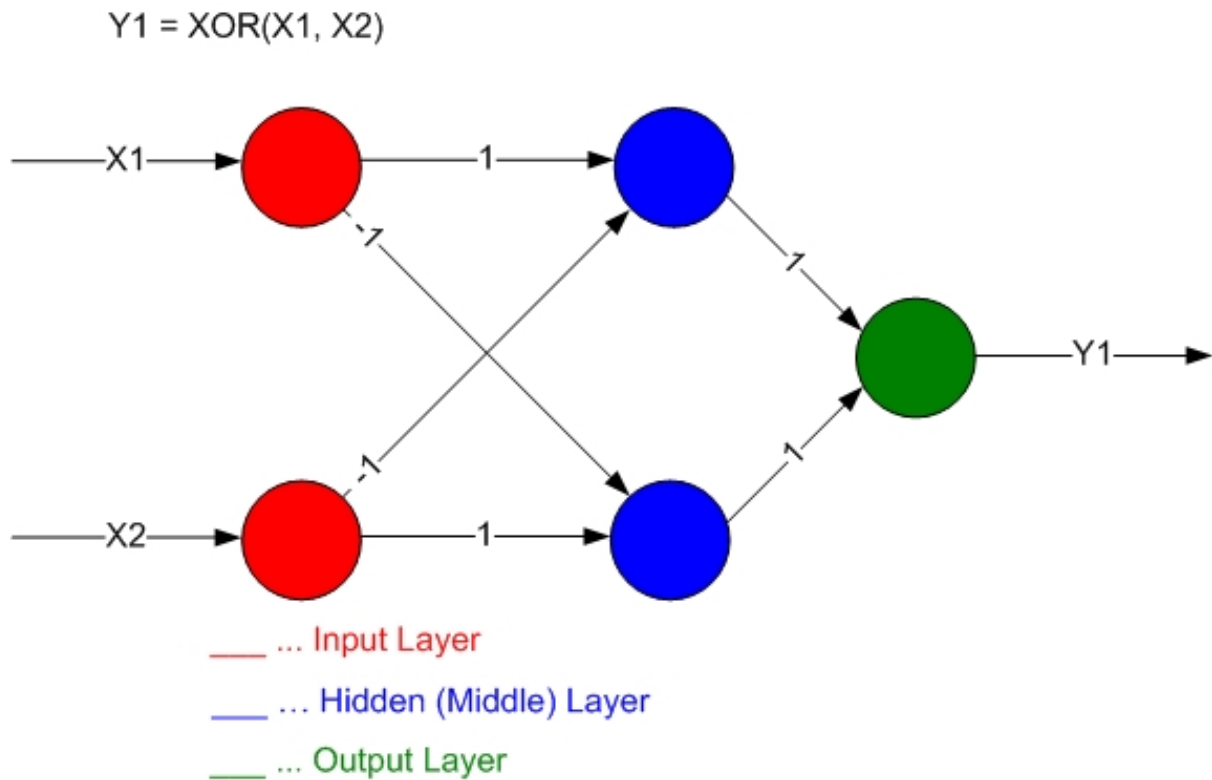


Figura 23: Funzione booleana XOR con una rete di perceptroni. Per l'immagine si faccia riferimento a [F11].

- $y \rightarrow 0$ per $x \rightarrow -\infty$. Per grandi valori di x in valore assoluto e negativi, quindi, l'output di un neurone tende a 0;

Agli estremi, quindi, un comportamento di questo tipo ben approssima quello del perceptrone con la differenza, però, di avere una funzione di attivazione e quindi un output continui rispetto al caso precedentemente discusso di funzione di attivazione a soglia del tipo $y = f[\sum_{i=1}^n (w_i x_i - \theta)]$.

In generale, comunque, le reti neurali si prestano bene a risolvere problemi che hanno le seguenti caratteristiche [S18]:

- elevata disponibilità di dati da analizzare durante la fase di addestramento della rete;
- difficoltà nell'individuare a priori un modello valido;
- necessità di elaborare anche un volume elevati di dati o comunque in tempi brevi;
- robustezza ai disturbi.

Di contro, i problemi delle reti neurali possono essere riassunti in [11]:

- bassa capacità di estrapolare un risultato quando si lavora con dati al di fuori del range del set di addestramento;
- se la rete neurale è addestrata per avere molti gradi di libertà per garantire percentuali elevate in fase di addestramento, si può avere un comportamento inatteso e risultati errati quando si ha un set di dati diverso da quello di *training*;
- *unexplainable AI*, ovvero il problema per cui le reti neurali lavorano come *black-box*, non dando quindi informazioni concrete sul loro funzionamento interno ma rispondendo semplicemente con degli output a determinati input. *L'explainable AI*, invece, comprende tutti quei modelli di intelligenza artificiale le cui soluzioni prodotte sono interpretabili e capibili da un essere umano ed è un requisito importante soprattutto nel caso di decisioni sbagliate da parte del sistema in analisi.

3.2.1 Apprendimento

Per quanto riguarda l'apprendimento utilizzando le reti neurali (e quindi i perceptron), si ha come schema generale che:

- i pesi e le soglie vengono inizializzati in modo casuale;
- vengono sottoposti alla rete degli esempi di *training*;
- i pesi e le soglie vengono aggiornati con l'obiettivo di rendere le uscite fornite dalla rete consistenti con gli esempi di *training*; le modifiche ai pesi vengono fatte con il fine di ridurre la distanza tra l'output della rete e l'uscita desiderata;
- per ogni esempio si hanno varie fasi di aggiornamento dei pesi (epoche di apprendimento).

Presi y_k come valore di uscita del percettore e t_k il valore desiderato di uscita relativi a k-esimo esempio di *training*, sia $e_k = t_k - y_k$ la regola di modifica dei pesi sarà data da:

$$w_j^{i+1} = w_j^i + \alpha x_j e_k$$

Come funzione di apprendimento si può fare riferimento al seguente pseudo-codice:

```
function nnLearning(examples) return network {  
  repeat  
    foreach ex in examples do  
      y ← nnOutput(network, ex)  
      t ← observed output values from ex  
      updateWeights(e, y, t)  
    end  
  until examples predicted or stop criteria reached  
  return network  
}
```

Teorema di convergenza del percettore (Rosenblatt) L'algoritmo di apprendimento del percettore converge sempre ad un insieme di pesi che rappresentano correttamente gli esempi, se questi sono provenienti da una funzione linearmente separabile.

Esistono diversi tipi di apprendimento, come dettagliato sotto.

- Apprendimento supervisionato, utilizzato nel momento in cui si ha a disposizione un *training set* contenente sia esempi tipici di ingresso che relative uscite. Alla rete viene quindi sottoposto un insieme di addestramento del tipo (x, t) , con x ingresso (tipicamente un vettore di *features*) e t uscita corrispondente. In questo modo la rete "impara" ad inferire la relazione che lega l'ingresso all'uscita. Il perceptrone e le reti *feed-forward* come il *multi-layer perceptron* (sezione 3.2.2) si basano sull'apprendimento supervisionato.
- Apprendimento non supervisionato, in cui gli algoritmi di addestramento modificano i pesi della rete solo sulla base delle variabili di ingresso. Questo tipo di addestramento è tipicamente utilizzato quando si vuole organizzare i dati di ingresso in *cluster* rappresentativi, quando la classe di appartenenza non è nota a priori o nella compressione dati [S20].
- Apprendimento su rinforzo nel quale la rete genera un output in modo autonomo e, un supervisore esterno interpreta tale output come corretto piuttosto che errato, incentivando o disincentivando la rete a seconda dei casi producendo una retroazione che guida l'algoritmo di apprendimento.

3.2.2 Multi-layer perceptron

Un *multi-layer perceptron* (MLP) è una particolare tipologia di rete neurale *feed-forward*. In questa architettura si possono individuare uno strato di ingresso, uno di uscita e da 1 ad n strati intermedi nascosti. Tipicamente, un MLP è completamente connesso, ovvero ogni uscita di un nodo di uno strato costituisce l'ingresso per un altro nodo dello strato successivo. A livello schematico, una rappresentazione di un *multi-layer perceptron* è mostrata in Figura 24.

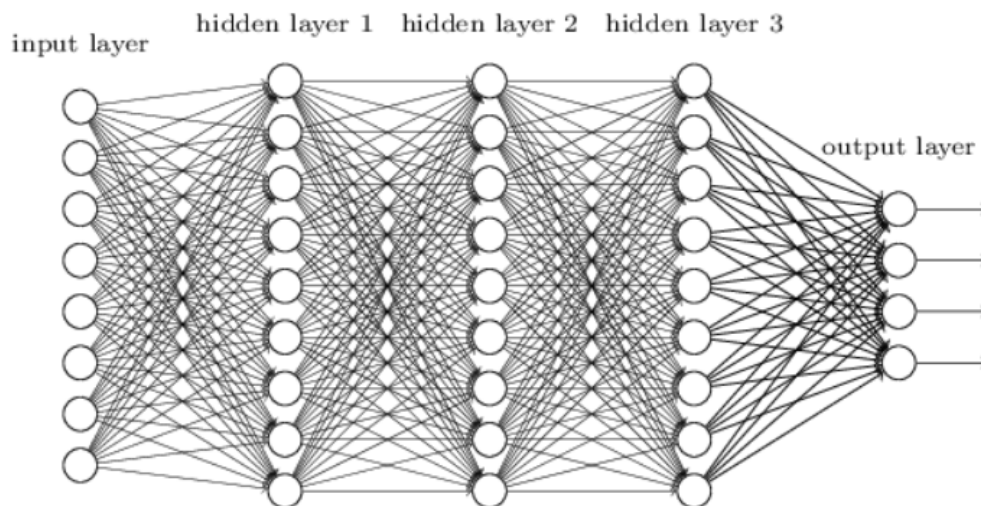


Figura 24: Rappresentazione di un *multi-layer perceptron* con 3 strati nascosti. Per l'immagine si faccia riferimento a [F12].

L'introduzione dell'MLP permette di superare le limitazioni imposte dal singolo perceptrone (in particolare quello sulla trattabilità dei soli problemi linearmente separabili). Infatti, uno o più strati nascosti possono individuare regioni arbitrarie nello spazio degli input, avendo poi di conseguenza un'intersezione di iperpiani diversi. Questo tipo di rete aggiunge nuovi strati di astrazione utilizzando gli strati intermedi. Ad esempio, nel caso si voglia utilizzare un MLP per il riconoscimento delle immagini, i neuroni del primo strato potrebbero imparare a riconoscere le linee, quelli del secondo forme più complesse come triangoli e rettangoli e così via per ogni strato aggiuntivo [S21].

Per l'addestramento di una rete di questo tipo, si utilizza il metodo di *error back propagation* (EMB), metodo che ha come scopo quello di minimizzare la somma degli errori quadratici (ovvero $\tilde{y} - y$, con \tilde{y} uscita attesa ed y uscita effettiva) [15].

Per fare quanto descritto, occorre definire la funzione errore $E(w)$ come:

$$E(w) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^L [t_j^{(i)} - y_j^{(i)}]^2$$

dove:

- $y_j^{(i)}$ è l'uscita calcolata ottenuta dal j -esimo neurone dato l' i -esimo ingresso della rete;
- $t_j^{(i)}$ rappresenta l'uscita attesa;
- N è il numero di coppie ingresso - uscita desiderata (x, t) che compongono il *training set*;
- L è il numero di neuroni che costituisce lo strato di uscita della rete.

Ad ogni nuova iterazione, i pesi dei neuroni della rete varieranno di una quantità $\Delta w = -\eta \frac{\partial E(w)}{\partial w_i}$, con $\eta \in (0, 1]$ tasso di apprendimento (o *learning rate*) della rete.

Occorre osservare che, in generale, la convergenza all'ottimo non è garantita dal metodo di *error back propagation*. In particolare, potrebbe non aversi convergenza se la funzione $E(w)$ presenta dei minimi locali. Inoltre, potrebbe verificarsi il fenomeno di *overfitting* (adattamento eccessivo o sovradattamento) per il quale la rete, in questo caso, si adatta ai dati osservati in quanto ha un numero eccessivo di parametri rispetto il numero di campioni osservati. Si faccia riferimento alla Figura 25.

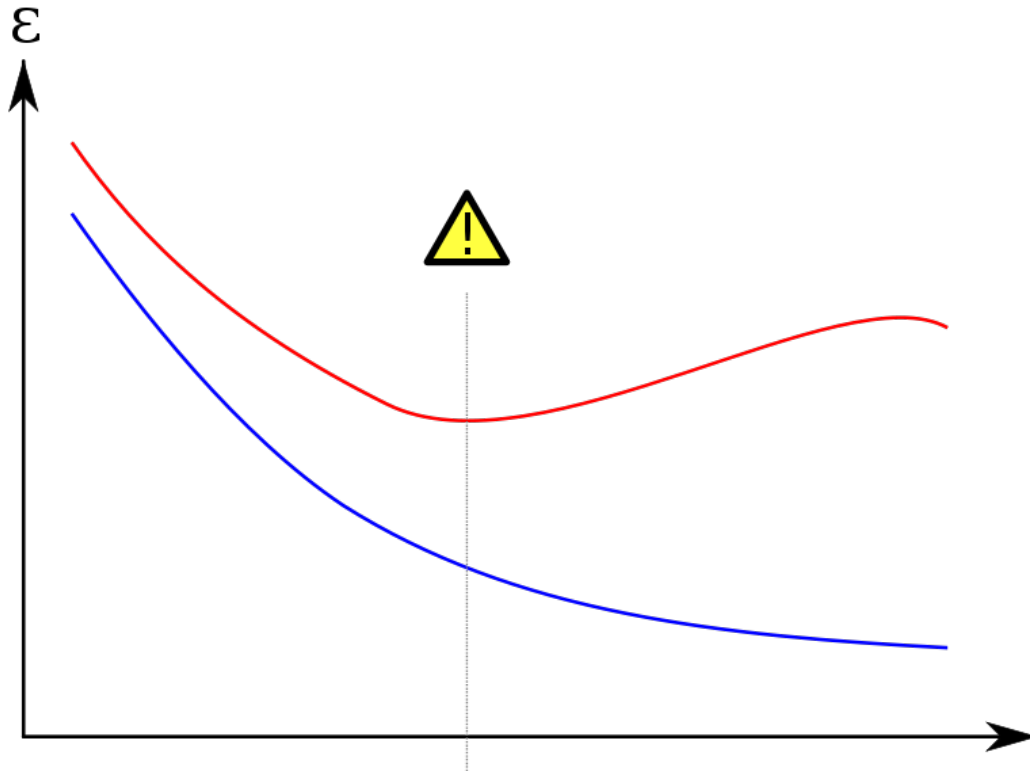


Figura 25: La curva in rosso rappresenta l'errore di classificazione dei dati di test o validazione, quella in blu l'errore di classificazione dei dati di addestramento.

Si può osservare come l'errore di classificazione aumenti mentre quello di addestramento diminuisca. Questa situazione è indice di un possibile problema di *overfitting*, ovvero il sistema ha imparato a classificare solo caratteristiche presenti nel set di *training*, ma che non sono presenti nella restante parte dei casi. Il modello di addestramento, quindi, non è in grado di generalizzare (questo avviene, ad esempio, quando il numero di campioni forniti per l'addestramento non è sufficientemente grande o l'addestramento è stato effettuato troppo a lungo) [S22].

4 Logica di controllo

In questo capitolo verrà descritta la logica di controllo implementata per la presente tesi, mentre in 4.1 è riportata la modalità di funzionamento del controllore sviluppato per il simulatore realizzato per la *Simulink Student Challenge* precedentemente introdotto.

In 4.2 viene presentata una descrizione della logica di funzionamento del nuovo controllore realizzato per questa tesi che sostituisce quello precedentemente introdotto.

Infine, in 4.3, viene riportato il funzionamento del controllore per la massimizzazione della produzione di energia.

4.1 Controllore base

Il controllore presente nel simulatore, da ora in poi chiamato *controllore base*, funge da punto di partenza per lo sviluppo dei nuovi supervisori che lo vanno di fatto a sostituire nel simulatore e, in particolare, il controllore per l'inseguimento termico (d'ora in poi *controllore avanzato*, sezione 5.2) risulta un'evoluzione ed una raffinazione di quello base. Gli altri controllori non saranno poi altro che modifiche ed accorgimenti particolari, in funzione del caso in cui ci si trova, fatti al controllore avanzato.

Il controllore base calcola l'*operative mode* a partire dalle variabili di stato del processo prese in considerazione, ovvero la potenza dell'LFR (P_{LFR}) e la temperatura a cui si trovano il TES ed il fluido di lavoro (T_{TES} e T_{LFR} rispettivamente). In base ai valori assunti dalle variabili, tramite una *lookup table*, si estrapolerà l'OM appropriata e quindi si imposterà la modalità di lavoro del sistema (come descritto nella sottosezione 2.4). Quindi, al contrario di quanto accade con il controllore avanzato che verrà successivamente descritto (sottosezione 5.2), in questo caso, la scelta dell'OM avviene mediante una serie di "if then ... else ..." codificati a priori su dei valori fissi impostati prima di avviare una simulazione. È possibile modificare tali parametri in modo da avere una OM piuttosto che un'altra. Un esempio è mostrato in Figura 26.

Block Parameters: Subsystem1

Subsystem (mask)

this block provides the actual Operation Mode (OM) based on the system status.

OM-1 = LFR ---> ORC ($T_{TES_av} > T$ hysteresis defocus)
OM0 = LFR recirculate itself
OM1 = LFR ---> ORC
OM2 = off
OM3 = LFR ---> TES
OM4 = LFR ---> TES + ORC
OM5 = TES ---> ORC
OM6 = LFR + TES ---> ORC

Please set the ORC's power features below.

Parameters

T TES max [°C] 280

P ORC max [kW] 26

P ORC min [kW] 5

Temperature ORC on [°C] 278

Temperature ORC off [°C] 200

Temperature min TES hysteresis defocus [°C] 278

Temperature max TES hysteresis defocus [°C] 280

OK Cancel Help Apply

Figura 26: Esempio di parametri utilizzati per la scelta dell'*operative mode* da parte del controllore base.

La logica con cui viene scelta una OM durante la simulazione (dati a priori i parametri prima introdotti), è mostrata nella Tabella 3. Per il parametro T_{set} introdotto nella Tabella appena citata, invece, si faccia riferimento alla Tabella 4.

OM	Condizioni operative
OM_{-1}	$T_{TES} \geq T_{TES_{max}}$
OM_0	$OM = 3 \ \& \ T_{LFR} < T_{TES} + 10C$
OM_1	$P_{LFR} \geq P_{ORC_{min}} \ \& \ P_{LFR} \leq P_{ORC_{max}}$
OM_2	$P_{LFR} = 0 \ \& \ T_{TES} \leq T_{set}$
OM_3	$P_{LFR} \geq P_{ORC_{min}} \ \& \ T_{TES} \leq T_{set}$
OM_4	$P_{LFR} \geq P_{ORC_{max}}$
OM_5	$P_{LFR} = 0 \ \& \ T_{TES} \geq T_{set}$
OM_6	$P_{LFR} \geq P_{ORC_{min}} \ \& \ T_{TES} \geq T_{set}$

Tabella 3: Tabella riassuntiva delle condizioni operative per cui si hanno le varie *operative mode*.

T_{set}	Condizione
$T_{set} = T_{off}$	$T_{TES} \geq T_{on}$
$T_{set} = T_{on}$	$T_{TES} \geq T_{off}$

Tabella 4: Tabella esplicitiva di T_{set} e di come questa varia in base al codice del simulatore. Per i valori T_{on} e T_{off} , a titolo di esempio, si faccia riferimento ai parametri della temperatura dell'ORC presenti in Figura 26.

4.2 Nuovo controllore

Nell'implementazione del nuovo controllore, si è deciso di aggiungere la possibilità di avere diversi profili e modalità di controllo e quindi diversi controllori con specifiche regole ed obiettivi. Nel dettaglio, si avranno, come profili di controllo, le seguenti modalità:

- inseguimento termico, ovvero, preso e modificato il controllore base precedentemente sviluppato per il simulatore, con questa modalità si cercherà di aumentare la produzione di energia termica del sistema anche in funzione dei consumi;
- massimizzazione della produzione, ovvero, preso il controllore per l'inseguimento termico, con questa modalità si cercherà di aumentare in generale la produzione di energia termica ed elettrica attraverso opportune modifiche al controllore prima introdotto.

Mentre le due modalità di controllo verranno descritte nel dettaglio nelle sottosezioni 4.2.1 e 4.3 rispettivamente, per una descrizione al relativo controllore che le implementa nel simulatore si rimanda alla sezione 5 e nello specifico alle sottosezioni 5.2 e 5.3.

Per quanto riguarda il riconoscimento del profilo dei consumi che di seguito verranno descritti, si è implementata una rete neurale con il compito di effettuare il riconoscimento dei segnali.

L'output della rete, quindi, determinerà poi quale controllore verrà utilizzato in funzione del tipo di utenza connessa all'impianto.

Per l'implementazione si rimanda in particolare alla sottosezione 5.2.5.

4.2.1 Inseguimento termico

Con l'inseguimento termico si cerca di fare in modo che l'utenza collegata all'impianto di generazione sia sempre coperta in termini di energia (termica in questo caso), privilegiando quindi la produzione della stessa. In altre parole, un impianto che lavora con questa logica di produzione, si attiva o incrementa la sua produzione nel momento in cui aumenta anche il carico ad esso collegato [S23].

Nell'ambito del sistema preso in analisi, si è scelto di operare cercando di inseguire il carico termico in quanto il dimensionamento dell'impianto (per 4 residenze private) è fatto in questi termini e non per l'energia elettrica. Questo fa sì che inseguire il carico elettrico risulti difficile in termini di energia massima producibile in relazione ai possibili consumi e di conseguenza sconveniente.

4.2.2 Profilo per abitazioni private

Per la generazione del profilo di consumi per abitazioni private si è utilizzato un file contenente i dati in questione relativi a 4 appartamenti con i rispettivi consumi elettrici e termici (per acqua calda domestica e riscaldamento) avuti nell'arco di un anno. A livello pratico, quindi, l'implementazione di questa parte di sistema equivale al caricamento del file e quindi l'estrapolazione di un elemento alla volta da questo, tenendo conto che il periodo di campionamento equivale a 10 minuti. Per fare questo si è implementato il sottosistema riportato in Figura 27.

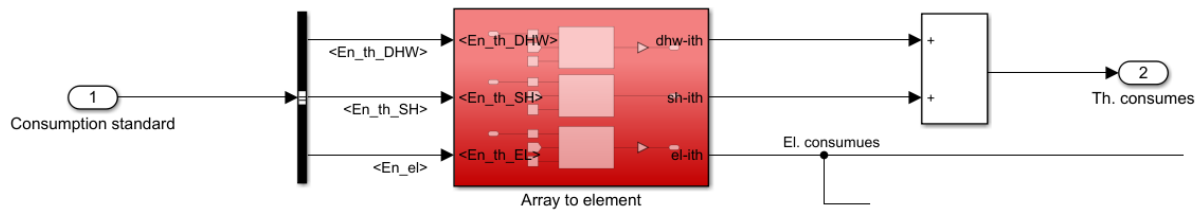


Figura 27: Schema a blocchi per la gestione del profilo dei consumi di 4 abitazioni private

Il blocco *Array to element* mostrato in Figura ha il compito di rendere disponibile l'*i*-esimo dato in accordo con il tempo della simulazione (avendo un periodo di campionamento per i dati dei consumi di 10 minuti, come precedentemente descritto).

4.2.3 Profilo per edificio scolastico

Per quanto riguarda i consumi di un edificio scolastico, si è fatto riferimento a quanto riportato in Figura 28.

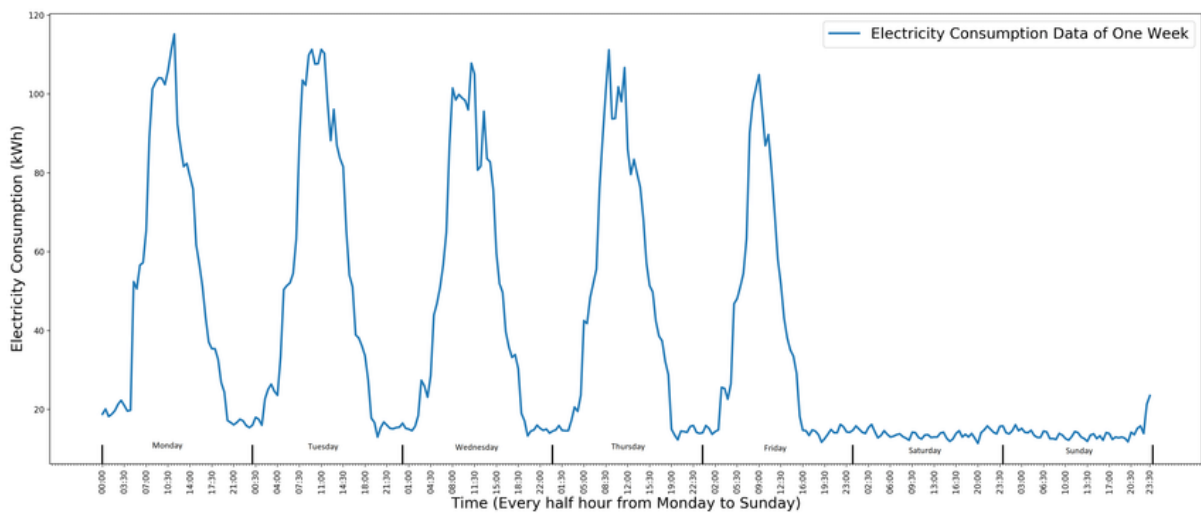


Figura 28: Profilo dei consumi di energia termica di un edificio scolastico. Per l'immagine si faccia riferimento a [F13].

Dal grafico sono stati estrapolati alcuni punti che sono stati poi analizzati in modo da ottenere una curva con un andamento in grado di approssimare in modo adeguato quello riportato in Figura 28.

Tenuto conto della dimensione del sistema di generazione di energia, infine, è stato necessario eseguire una scalatura del segnale trovato in modo da renderlo congruente con un'utenza eventualmente collegata all'impianto di cogenerazione. In particolare, trovata la media e la deviazione standard del segnale, queste verranno sommate in modo da ottenere un'ampiezza non negativa. A questo si sommerà del rumore per aggiungere una componente aleatoria. Lo schema implementativo di questa parte di sistema è mostrato in Figura 29 e 30. Nello specifico, in Figura 29 è mostrato come il segnale cambi in funzione dell'orario (sarà solo rumore di ampiezza contenuta in fascia oraria non scolastica e avrà un andamento come quello appena descritto altrimenti). In Figura 30, invece, viene mostrato la generazione del segnale dei consumi in fascia oraria scolastica.

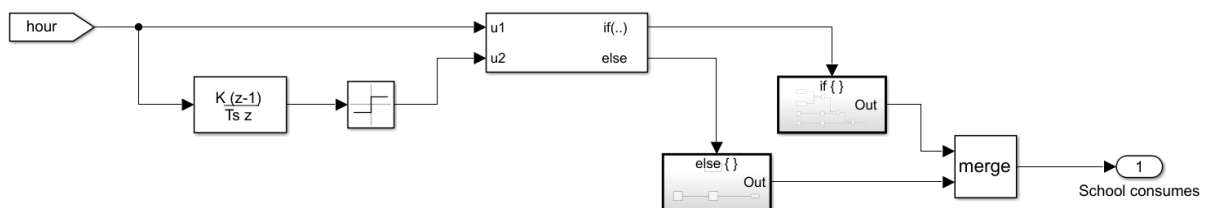


Figura 29: Schema della generazione del segnale dei consumi in funzione dell'ora.

Per l'estrapolazione dei dati dal grafico si faccia riferimento a A. Rohatgi - WebPlotDigitizer.

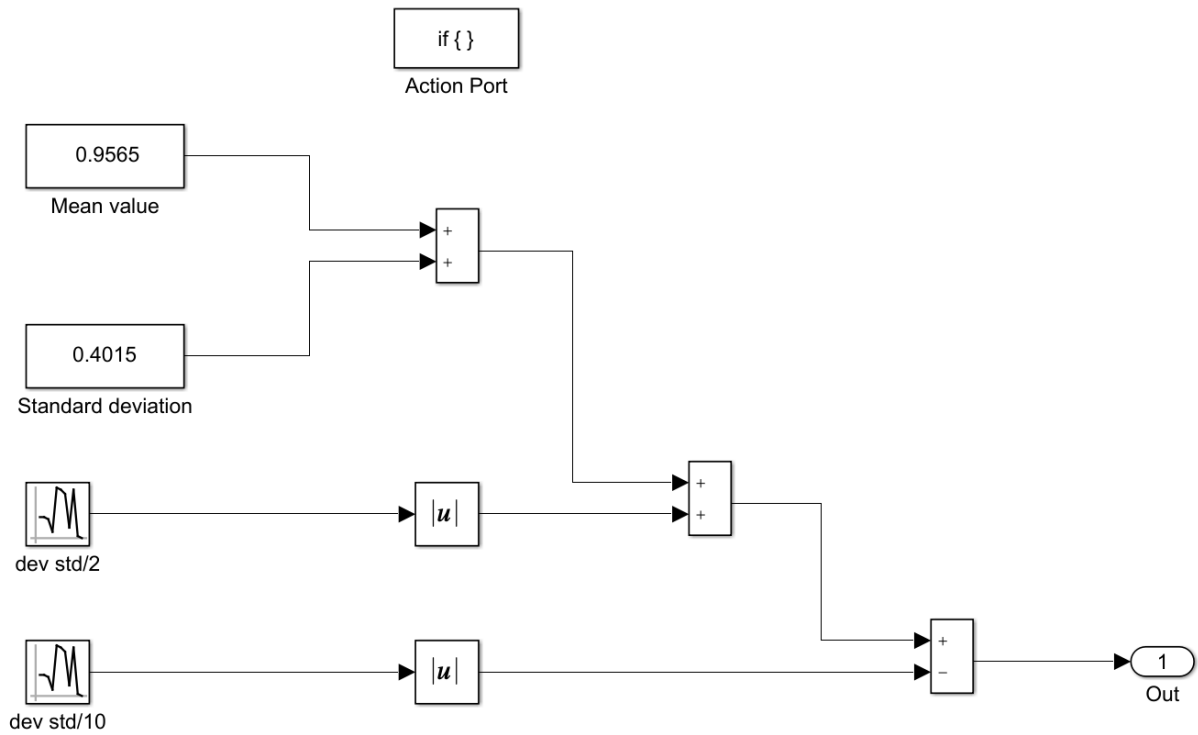


Figura 30: Generazione del profilo dei consumi scolastici in orario scolastico.

Il risultato di quanto descritto è, infine, riportato in Figura 31 in cui viene mostrato il profilo dei consumi scolastici ottenuto con l'implementazione ora presentata. Il grafico riportato è relativo ad uno *zoom* di 4 giorni fatto durante una simulazione.

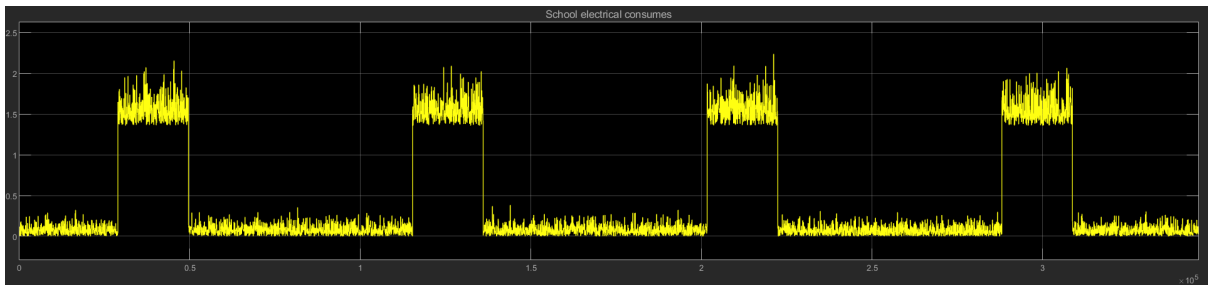


Figura 31: Profilo dei consumi scolastici ottenuto grazie all'implementazione descritta.

4.3 Massimizzazione della produzione di energia

Con questo obiettivo di controllo, si vogliono aumentare le prestazioni del sistema affinché questo produca più energia termica ed elettrica. Nel cercare di soddisfare questo requisito, si è agito andando a considerare come elemento di controllo i pesi dati dal rapporto tra l'energia proveniente dal *linear Fresnel reflector* e l'energia elettrica o termica (si sono previsti entrambi i casi, ma allo stato attuale, nel simulatore viene preso in considerazione il termine relativo all'energia termica). In particolare, si avrà $w_e = \frac{E_{LFR}}{E_e}$ il peso relativo alla produzione elettrica e $w_t = \frac{E_{LFR}}{E_t}$ quello per l'energia termica. In altre parole, la grandezza appena descritta equivale all'inverso del rendimento istantaneo η del sistema per la produzione di energia elettrica e termica, ovvero $w_e = \eta_e^{-1}$ e $w_t = \eta_t^{-1}$.

La scelta di utilizzo dell'energia come grandezza presa in considerazione è stata dettata da considerazioni di tipo implementativo. Infatti, questa presenta un valore nullo solo nell'istante iniziale della simulazione ed è una funzione monotona crescente (essendo l'integrale nel tempo della relativa potenza). Un esempio dell'andamento della grandezza è mostrato in Figura 32, mentre in Figura 33 vengono riportati gli andamenti dei relativi pesi.

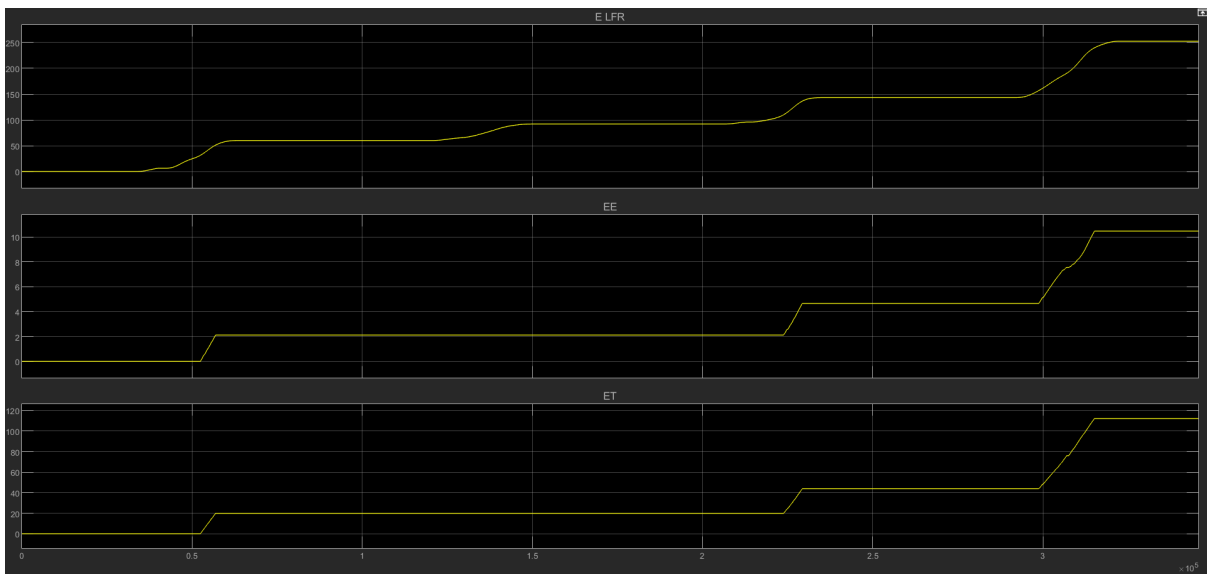


Figura 32: Andamento delle grandezze E_{LFR} , E_e ed E_t .

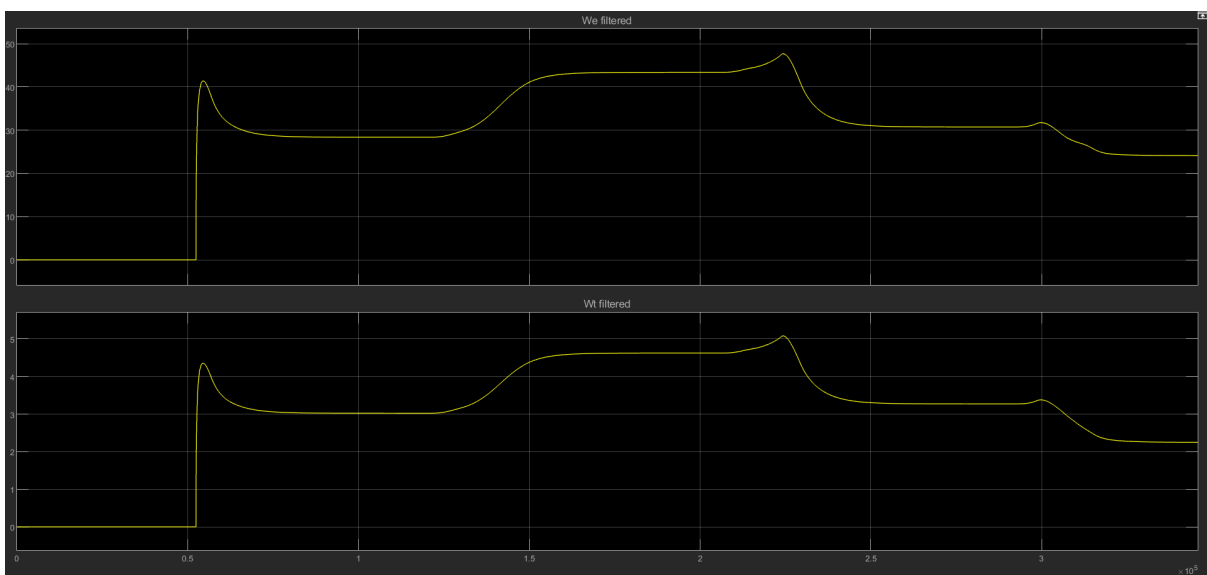


Figura 33: Andamento dei pesi w_e e w_t .

Calcolati i pesi come appena descritto, ciò che il controllore per la massimizzazione della produzione di energia prenderà in considerazione saranno le derivate di questi ultimi andando poi a modificare la potenza minima dal *linear Fresnel reflector* per cui si può produrre energia. Infatti, la soglia, da ora denominata P_σ , è inizialmente impostata a $13kWh$ così come si ha nel controllore base e, in funzione della derivata dei pesi verrà alzata o abbassata per massimizzare la produzione di energia.

Quindi, preso $\delta w_t = \frac{dw_t(t)}{dt}$, si possono avere i seguenti casi:

1. $\delta w_t \geq 0$, ovvero la potenza generata dall'LFR cresce, ma non si sta producendo energia. In questo caso si diminuisce P_σ in modo da produrre anche con potenze minori provenienti dall'LFR;
2. $\delta w_t < 0$, quindi l'energia prodotta cresce in modo maggiore rispetto l'energia dall'LFR e si sta producendo in condizioni standard e quindi, per una questione di rendimento, si aumenta P_σ in modo da non produrre sempre in condizioni svantaggiose.

Qualitativamente, infatti, il punto 1 fa riferimento al caso in cui l'LFR produce energia, ma non l'ORC e quindi il sistema in generale. Essendo quindi l'aumento della grandezza al numeratore non accompagnato da un aumento di quella al denominatore, la grandezza aumenta ed avrà una derivata non negativa. Il caso riportato nel punto 1 comprende anche la possibilità per cui né l'LFR né il TES siano in grado di alimentare l'ORC; in questo scenario si avrà derivata nulla e grandezza che rimane costante.

Il punto 2, invece, comprende i casi per cui l'LFR è in grado di alimentare l'ORC, avendo di conseguenza una produzione di energia che fa abbassare il peso w_t , la cui derivata sarà di conseguenza negativa. Nello stesso caso ricade lo scenario per cui l'ORC è alimentato dal TES (ovvero la potenza dell'LFR è bassa o nulla, ma la temperatura a cui si trova il TES fa sì che si possa comunque usare l'ORC).

Mentre in questa sottosezione si è data una prima panoramica della logica con cui si è implementato il controllo per la massimizzazione dell'energia, per una descrizione dettagliata di questo aspetto si rimanda alla sottosezione 5.3 mentre, per l'implementazione in MATLAB e Simulink si rimanda invece alla sottosezione 5.3.2.

5 Controllori implementati

In questo capitolo vengono riportati gli elementi sviluppati per il controllo e la supervisione dell'impianto di cogenerazione. In particolare, in 5.1 vengono descritti tutti i moduli ausiliari implementati per la gestione dei dati e dei segnali utilizzati nel controllore, che viene invece descritto in 5.2 nella versione per l'inseguimento termico ed in 5.3 in quella per la massimizzazione della produzione.

5.1 Moduli ausiliari di gestione dati

In questa sezione verranno descritti alcuni dei moduli software realizzati nell'ambito di questo lavoro e che presentano una certa importanza all'interno del sistema implementato.

In 5.2 e 5.3, invece, verranno trattati i componenti di maggior rilievo, ovvero i vari controllori che piloteranno l'*operative mode* del sistema in funzione delle variabili di stato e dell'obiettivo del controllo (sottosezioni 4.2.1 e 4.3).

Caricamento dei dati di consumo energetico Una delle prime funzionalità aggiunte al simulatore è stata quella di includere anche i consumi di energia elettrica e termica (in termini di energia necessaria per il riscaldamento e per l'acqua calda domestica) di una possibile utenza composta da 4 appartamenti collegata all'impianto. Per fare questo, è stato necessario caricare i dati di interesse contenuti in un file Excel all'interno del simulatore e, durante una simulazione, leggere un dato alla volta dal file in base al tempo. Infatti, i dati dei consumi hanno un periodo di campionamento di 10 minuti, mentre il simulatore opera a livello globale con un passo di 1 secondo. L'implementazione dell'operazione di caricamento dei dati è riportata in Figura 34, mentre quella di estrazione del dato temporalmente corretto è mostrata in Figura 35 per quanto riguarda il sottosistema che implementa la funzionalità ed in Figura 36 in cui è riportato un esempio. Per quanto riguarda il profilo dei consumi di un edificio scolastico, invece, si rimanda alla sottosezione 4.2.3 per una descrizione delle modalità di caricamento dei dati.

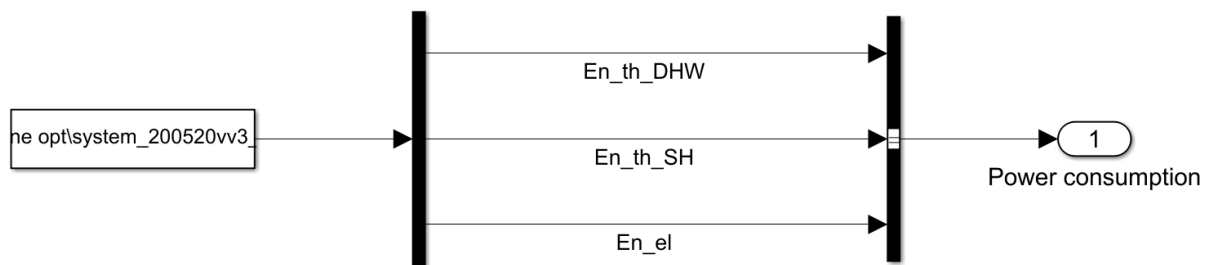


Figura 34: Implementazione in Simulink del caricamento dati.

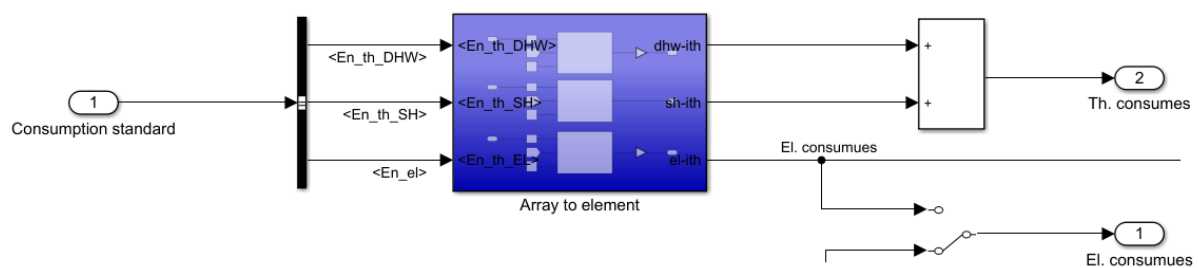


Figura 35: Implementazione in Simulink del sottosistema per la lettura del dato temporalmente corretto.

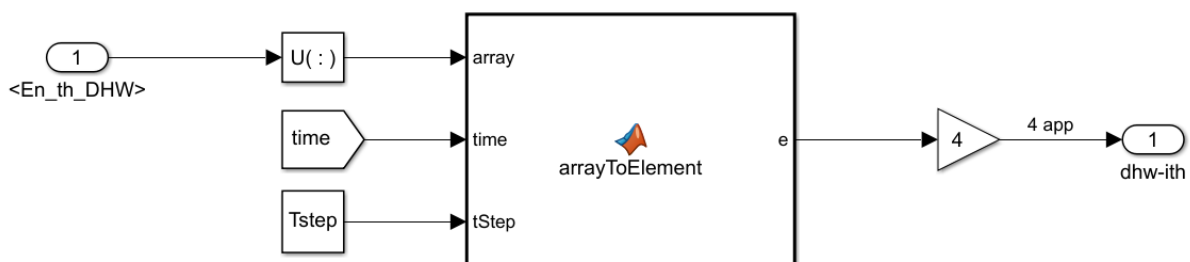


Figura 36: Esempio di estrazione del dato temporalmente corretto.

Rete neurale Per quanto riguarda la rete neurale per il riconoscimento del tipo di consumi si avrà un sottosistema che, durante la simulazione calcolerà le *feature* utilizzate in fase di addestramento (quali media, *RMS*, ecc. per una descrizione più dettagliata di questa parte si faccia riferimento alla sottosezione 5.2.5). Complessivamente, l'implementazione di questa parte è riportata in Figura 37. Per la costruzione del segnale di input della rete neurale, invece, si faccia sempre riferimento a quanto riportato nella sottosezione 5.2.5.

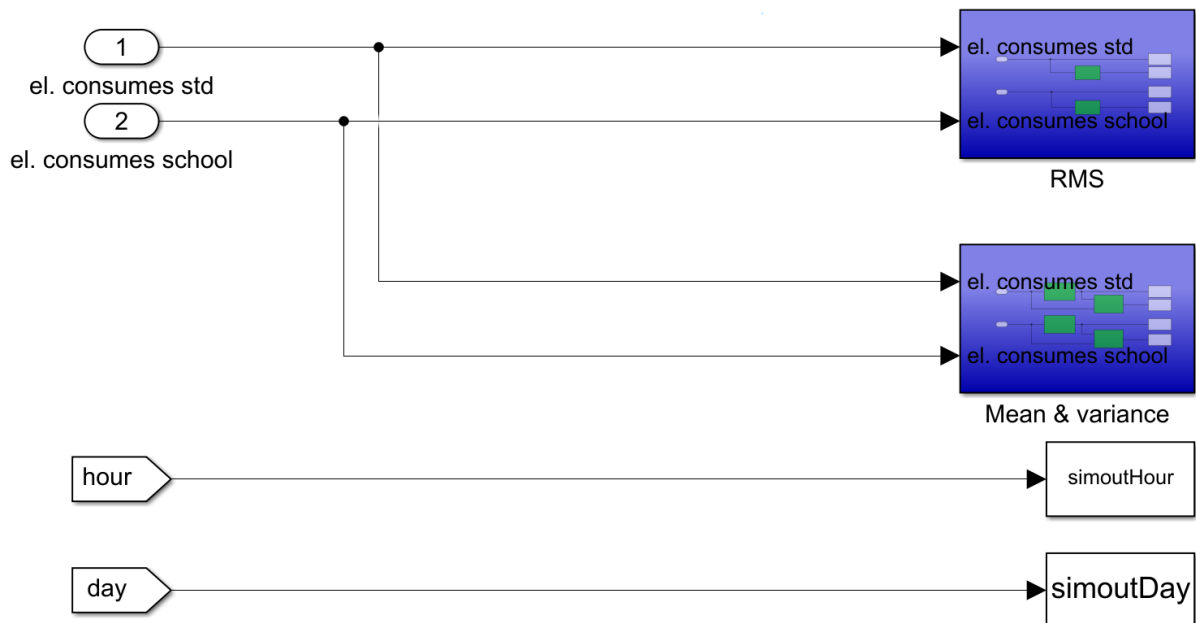


Figura 37: Sottosistema per il calcolo ed il salvataggio delle *feature* per l'addestramento della rete neurale. In blu è riportato il blocco per il calcolo delle *feature* stesse (media, dell'*RMS* e varianza online).

Calcolo dei dati (grezzi) della produzione giornaliera del sistema Per l'implementazione di questa funzionalità si è realizzato un sottosistema che, prese le informazioni temporali necessarie (ora, giorno e mese) ed il segnale relativo all'energia prodotta in kJ/h , restituisce in uscita una *struct* contenente l'insieme delle informazioni associate alla produzione di energia (in kWh) e l'intervallo di tempo in cui questa è stata generata. Si è scelto di considerare come uno degli ingressi l'energia in kJ/h in quanto, essendo questa la derivata dell'energia in kJ e, a meno di una costante, anche dell'energia in kWh , si avrà di conseguenza un segnale che sarà diverso da 0 solo nel momento in cui il sistema produce energia, al contrario di quanto accadrebbe considerando l'energia espressa in kWh che avrà valore nullo solo nell'istante iniziale e fino alla prima produzione di energia. Questo consente una maggiore semplicità a livello di implementazione e scrittura del codice ed inoltre, calcolando quando si avrà un fronte di salita e discesa nel segnale (grazie al blocco *edge detector*), si sarà anche in grado di determinare l'orario di inizio e fine produzione in modo agevole. Complessivamente, lo schema a blocchi in Simulink di questo sottosistema è riportato in Figura 38. In Figura 39, invece, è riportata l'implementazione dell'*edge detector* per il calcolo dei fronti di salita e discesa del segnale prima introdotto. Per il filtraggio (inteso come rimozione e correzione di eventuali dati ridondanti o non corretti) e la visualizzazione dei dati così acquisiti si rimanda infine a quanto riportato in sezione 7.

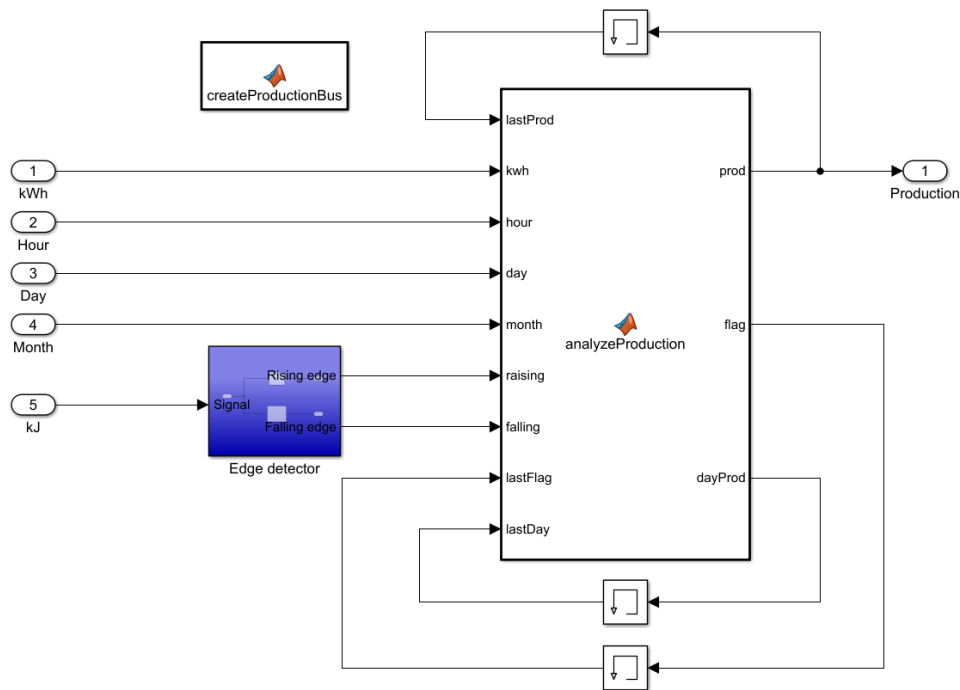


Figura 38: Implementazione in Simulink del sottosistema per l'acquisizione dei dati grezzi relativi alla produzione giornaliera.

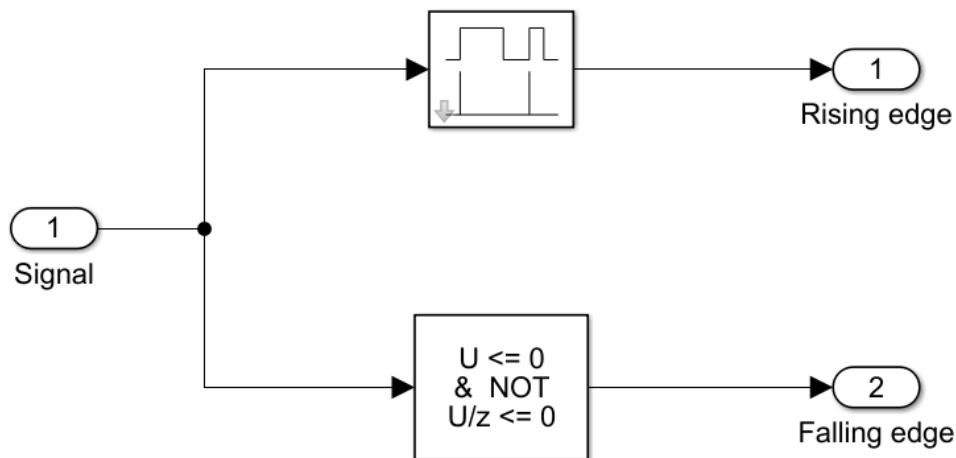


Figura 39: Implementazione dell'edge detector.

Contatore delle commutazioni di fase Il sottosistema qui descritto ha il compito di contare quante commutazioni di *operative mode* avvengono nel corso di una simulazione. In totale, si avrà il conteggio delle commutazioni che si hanno utilizzando il controllore sviluppato per il progetto *InnovaMicroSola* (sottosezione 4.1) e quello sviluppato in questo lavoro (in base alla modalità di controllo utilizzata nella simulazione, sottosezioni 5.2 e 5.3). Vengono inoltre contate anche le commutazioni che si avrebbero con un controllore per l'inseguimento del carico elettrico (non sviluppato in questo lavoro, attualmente si ha una copia del controllore per l'inseguimento del carico termico). Complessivamente, lo schema del contatore e la sua implementazione sono mostrati in Figura 40 e 41, rispettivamente.

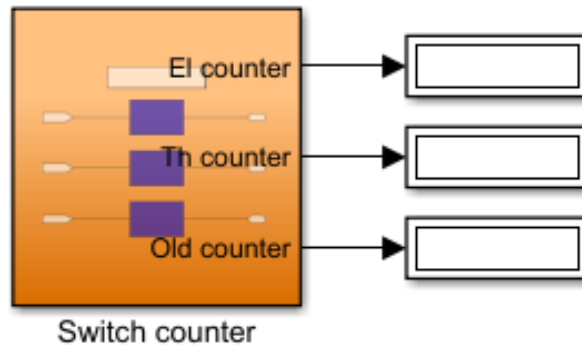


Figura 40: Schema a blocchi del contatore di commutazioni sviluppato in Simulink.

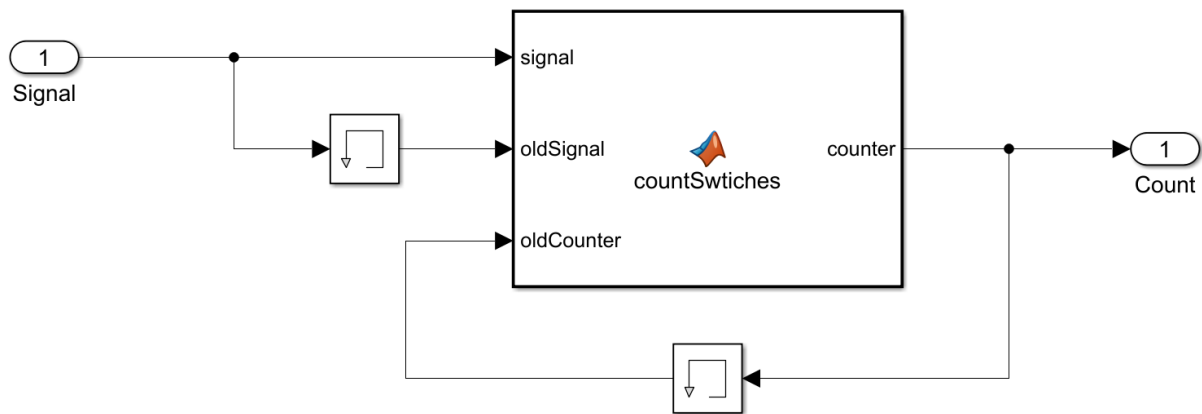


Figura 41: Esempio di implementazione del contatore di commutazioni (riportato il caso di conteggio per un singolo segnale) in Simulink.

Interfaccia per la visualizzazione delle informazioni Infine, è stata implementata un'interfaccia grafica in Simulink per la visualizzazione dell'andamento del sistema. Lo scopo di questo sistema è quello di mostrare graficamente all'utente la grandezza $System_{info}$ e quindi la produzione del sistema in relazione ai consumi. Oltre che alla visualizzazione delle performance del sistema per la produzione di energia termica, è stato prevista anche la visualizzazione delle informazioni sulla produzione di energia elettrica. Ad entrambe le informazioni è stata correlata una visualizzazione in termini di colore, oltre a quella testuale. Il sottosistema sviluppato è mostrato in Figura 42.

Number shown	Description
3	<i>Produced power greater than consumptions, increasing</i>
1	<i>Produced power greater than consumptions, decreasing</i>
-1	<i>Produced power less than consumptions, increasing</i>
-3	<i>Produced power less than consumptions, decreasing</i>

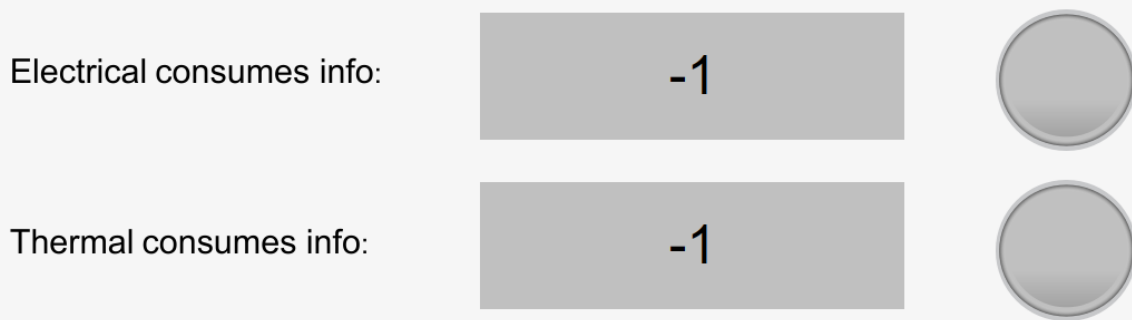


Figura 42: Implementazione in Simulink dell'interfaccia grafica per la visualizzazione delle prestazioni del sistema.

Controllo della temperatura del TES Essendo l' OM_{-1} stata rimossa dal controllore (sottosezione 5.2), è stato implementato un blocco che, oltre ad eliminare eventuali commutazioni non volute dell'*operative mode*, effettua anche un controllo su T_{TES} per evitare che questa superi il valore massimo consentito di 280°C , modificando l' OM calcolata dal controllore nel caso questa condizione si verifichi durante una simulazione. L'implementazione è mostrata in Figura 43.

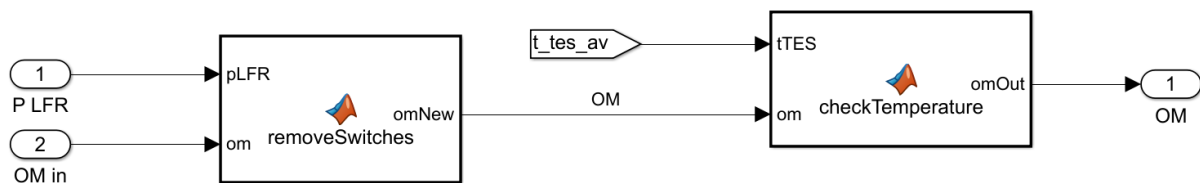


Figura 43: Implementazione in Simulink per la correzione del segnale in uscita dal controllore ed il controllo di T_{TES} .

5.2 Controllore per l'inseguimento termico

Il controllore per l'inseguimento termico ora descritto rappresenta un'evoluzione di quello base precedentemente introdotto (sottosezione 4.1) che sostituisce le vecchie regole date da una serie di "if then ... else ..." con delle nuove utilizzando la logica fuzzy (sezione 3.1). La base delle conoscenze impiegata nella sintesi del nuovo controllore è di fatto data dalle regole impiegate nel controllore base prima descritto. A questo insieme di regole ne sono state aggiunte poi di nuove introducendo altre informazioni dal sistema e quindi nuove variabili in ingresso al controllore.

5.2.1 Profilo residenziale

Lo schema di base del controllore avanzato (per il profilo di consumo di residenze private, sottosezione 4.2.2) è mostrato in Figura 44.

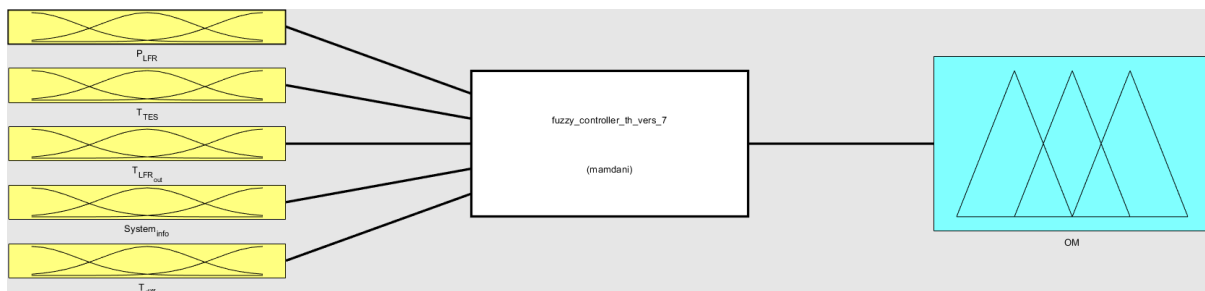


Figura 44: Schema logico della struttura del controllore avanzato. In giallo sono riportati i blocchi relativi alle variabili di ingresso, in blu quelli per l'output ed in bianco il blocco con l'insieme di regole fuzzy.

Dalla stessa Figura è possibile individuare come le grandezze in ingresso del controllore siano:

- P_{LFR} , ovvero la potenza in uscita dal *linear Fresnel reflector*;
- T_{TES} , ovvero la temperatura a cui si trova il *thermal energy storage*;
- $T_{LFR_{out}}$, ovvero la temperatura del fluido di lavoro in uscita dal *linear Fresnel reflector*;
- $System_{info}$, ovvero una grandezza che dà una stima delle prestazioni del sistema in funzione dei consumi dell'utenza collegata all'impianto (per una descrizione più dettagliata di questa grandezza e di come viene calcolata si rimanda alla sottosezione 5.2.4);
- $T_{diff} = T_{LFR_{out}} - T_{TES}$, ovvero la differenza di temperatura tra il fluido di lavoro ed il TES.

Per descrivere il controllore è opportuno introdurre gli insiemi individuati per le variabili di ingresso ed il modo in cui questi ne suddividono il range di variazione.

P_{LFR} Per la divisione in intervalli della variabile di stato P_{LFR} , si faccia riferimento alla Tabella 5 ed alla Figura 45 una rappresentazione grafica di tali insiemi.

Insieme fuzzy	Insieme di appartenenza di P_{LFR}
<i>Zero</i>	$P_{LFR} = 0$
<i>Low</i>	$P_{LFR} \in [0.01, 13]$
<i>OK</i>	$P_{LFR} \in [13, 26]$
<i>High</i>	$P_{LFR} \in [26, 150]$

Tabella 5: Divisione in intervalli della variabile P_{LFR} . Il colore blu indica un insieme singleton, mentre il colore rosso rappresenta un insieme fuzzy.

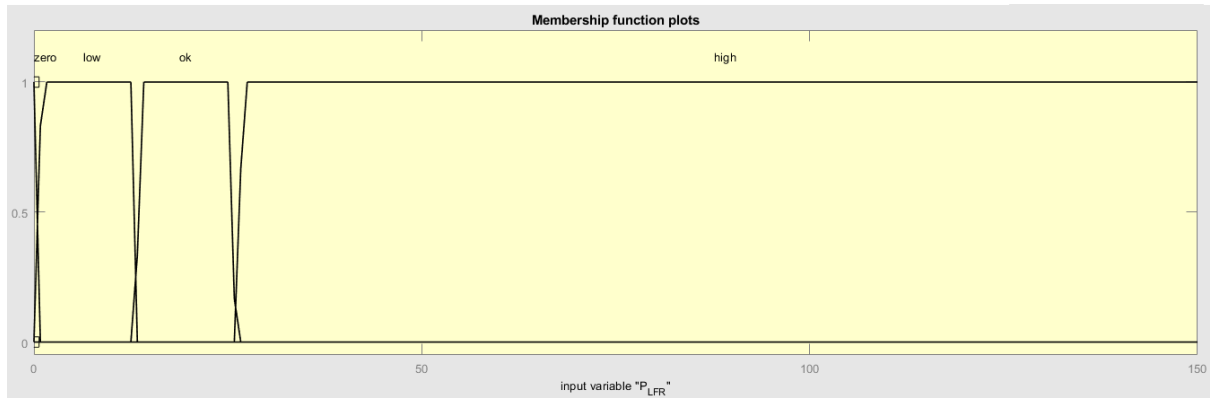


Figura 45: Rappresentazione grafica degli insiemi fuzzy individuati per la variabile P_{LFR} .

T_{TES} Per la divisione in intervalli della variabile di stato T_{TES} , si faccia riferimento alla Tabella 6 ed alla Figura 46 una rappresentazione grafica di tali insiemi.

Insieme fuzzy	Insieme di appartenenza di T_{TES}
<i>High</i> _{OM₃}	$T_{TES} \in [0, 210]$
<i>Low</i>	$T_{TES} \in [50, 180]$
<i>Normal</i>	$T_{TES} \in [180, 210]$
<i>OK</i>	$T_{TES} \in [210, 278]$
<i>High</i>	$T_{TES} \in [278, 280]$

Tabella 6: Divisione in intervalli della variabile T_{TES} . Il colore rosso indica un insieme fuzzy, mentre il colore verde rappresenta un insieme crisp.

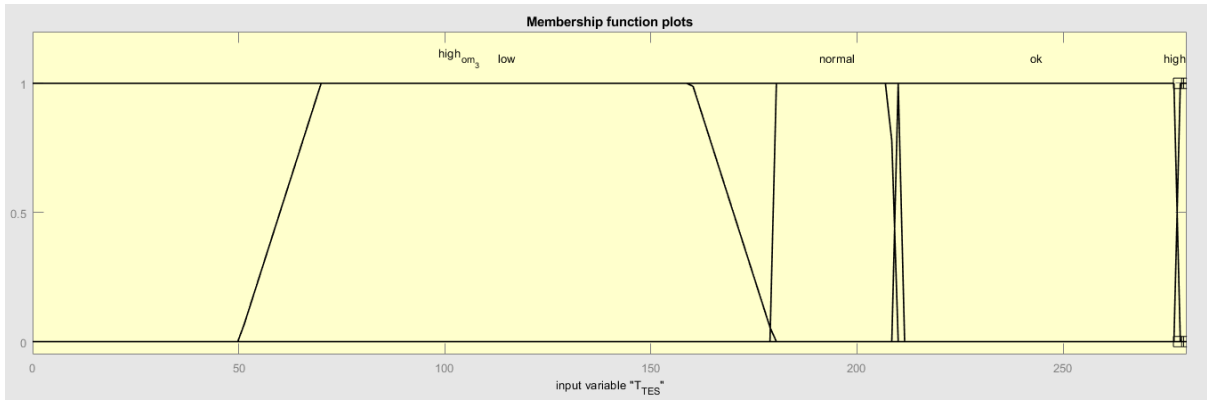


Figura 46: Rappresentazione grafica degli insiemi fuzzy individuati per la variabile T_{TES} .

$T_{LFR_{out}}$ Per la divisione in intervalli della variabile di stato $T_{LFR_{out}}$, si faccia riferimento alla Tabella 7 ed alla Figura 47 una rappresentazione grafica di tali insiemi.

Insieme fuzzy	Insieme di appartenenza di $T_{LFR_{out}}$
<i>High</i>	$T_{LFR_{out}} \in [220, 320]$

Tabella 7: Divisione in intervalli della variabile $T_{LFR_{out}}$. Il colore verde rappresenta un insieme crisp.

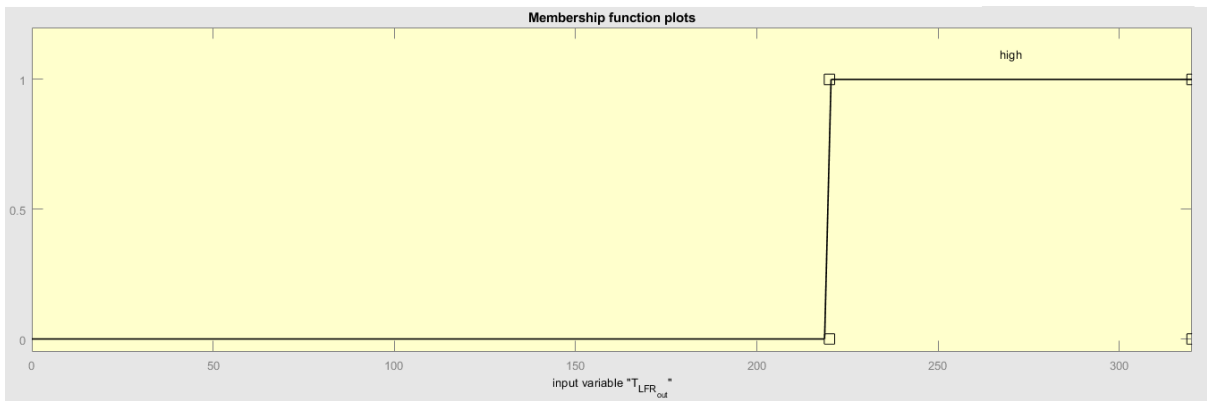


Figura 47: Rappresentazione grafica degli insiemi fuzzy individuati per la variabile $T_{LFR_{out}}$.

System_{info} Per la divisione in intervalli della variabile di stato $System_{info}$, si faccia riferimento alla Tabella 8 ed alla Figura 48 una rappresentazione grafica di tali insiemi.

Insieme fuzzy	Insieme di appartenenza di System _{info}
<i>Bad</i>	$System_{info} \in [-3, -2]$
<i>Poor</i>	$System_{info} \in [-2, 0]$
<i>OK</i>	$System_{info} \in [0, 2]$
<i>Good</i>	$System_{info} \in [2, 3]$

Tabella 8: Divisione in intervalli della variabile $System_{info}$. Il colore rosso rappresenta un insieme fuzzy.

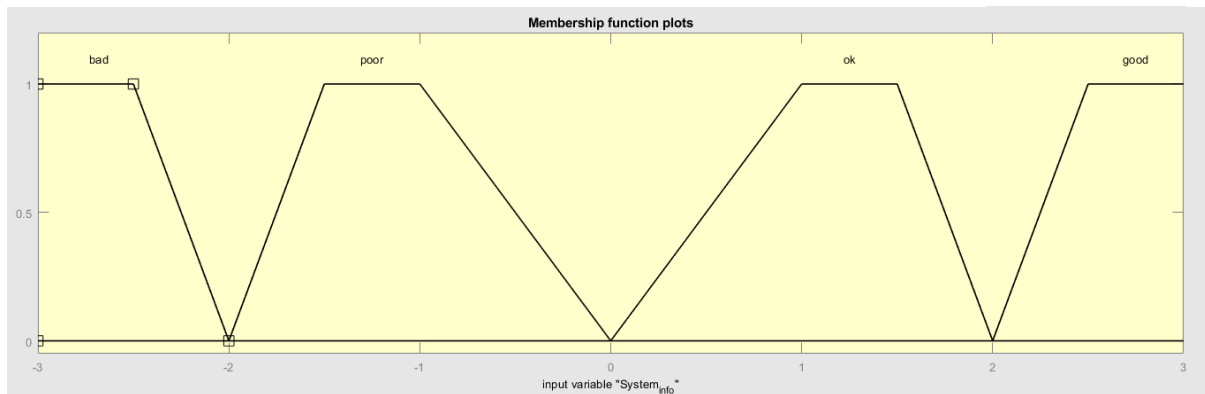


Figura 48: Rappresentazione grafica degli insiemi fuzzy individuati per la variabile $System_{info}$.

T_{diff} Per la divisione in intervalli della variabile di stato T_{diff} , si faccia riferimento alla Tabella 9 ed alla Figura 49 una rappresentazione grafica di tali insiemi.

Insieme fuzzy	Insieme di appartenenza di T _{diff}
<i>Bad</i>	$T_{diff} \in [-200, 7]$
<i>OK</i>	$T_{diff} \in [7, 200]$

Tabella 9: Divisione in intervalli della variabile T_{diff} . Il colore rosso indica un insieme fuzzy, mentre il colore verde rappresenta un insieme crisp.

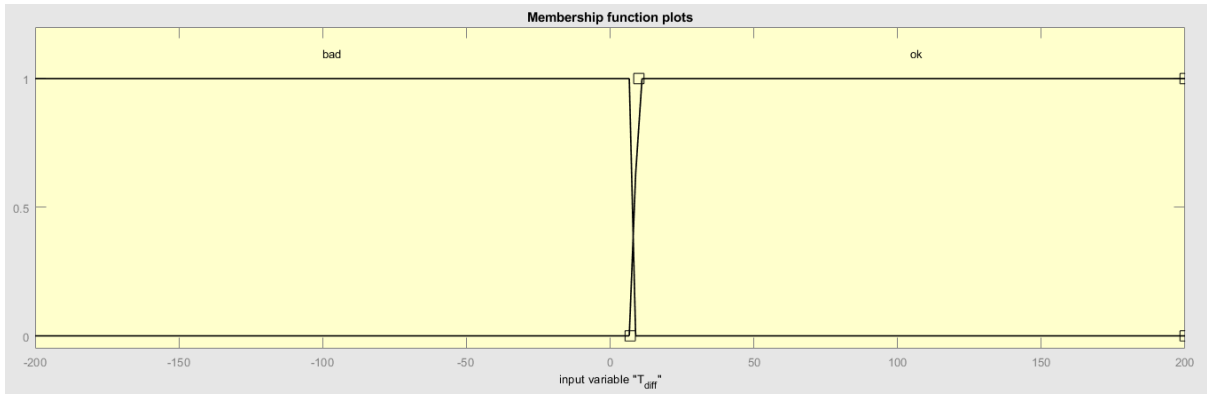


Figura 49: Rappresentazione grafica degli insiemi fuzzy individuati per la variabile T_{diff} .

Avendo definito i vari intervalli in cui sono state suddivise le diverse variabili utilizzate per il controllo, si può proseguire ad introdurre le regole implementate nel controllore fuzzy che concorrono alla scelta dell'*operative mode* in funzione delle condizioni di lavoro, ovvero i valori assunti in relazione agli intervalli dalle variabili prima definiti. A tal proposito, si faccia riferimento a quanto riportato in Tabella 10. Per una rappresentazione grafica delle stesse, invece, si rimanda alla Figura 50.

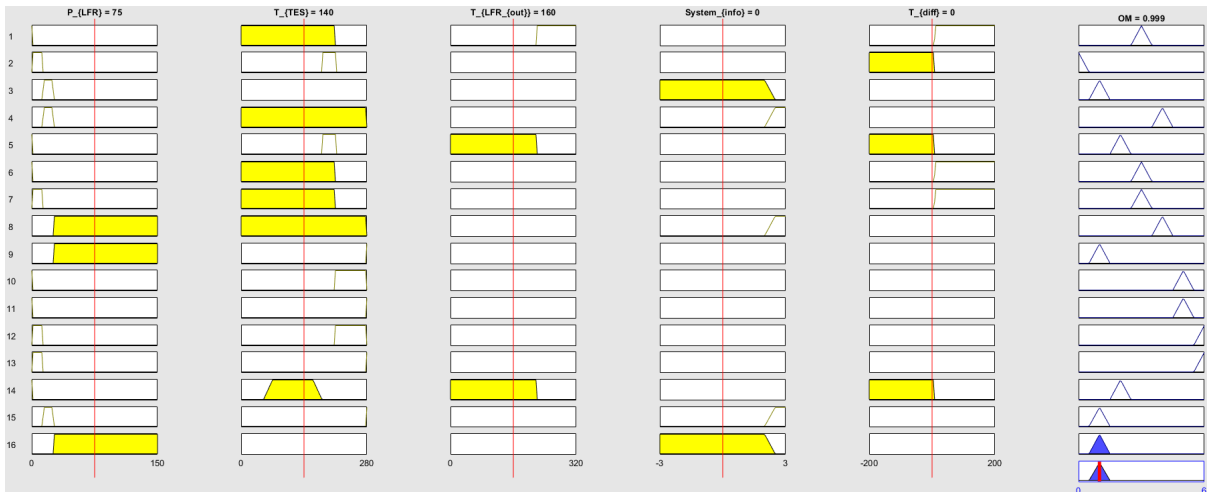


Figura 50: Rappresentazione grafica delle regole fuzzy implementate nel controllore avanzato con un esempio di scelta dell'*operative mode* in funzione dei valori assunti dalle variabili di input (regola 16, OM_1).

Regola	P_{LFR}	T_{TES}	T_{LFR_{out}}	System_{info}	T_{diff}	OM scelta
1	Zero	<i>High</i> _{OM₃}	High		OK	OM ₃
2	Low	Normal			Bad	OM ₀
3	OK			¬ Good		OM ₁
4	OK	¬ High		Good		OM ₄
5	Zero	Normal	¬ High		Bad	OM ₂
6	Zero	<i>High</i> _{OM₃}	High		OK	OM ₃
7	Low	<i>High</i> _{OM₃}	High		OK	OM ₃
8	High	¬ High		Good		OM ₄
9	High	High				OM ₁
10	Zero	OK				OM ₅
11	Zero	High				OM ₅
12	Low	OK				OM ₆
13	Low	High				OM ₆
14	Zero	Low	¬ High		Bad	OM ₂
15	OK	High		Good		OM ₁
16	High			¬ Good		OM ₁

Tabella 10: Regole fuzzy implementate per il controllore avanzato.

Il controllore che utilizza le regole prima descritte (e per la cui implementazione in Simulink si rimanda alla sottosezione 5.2.5)

5.2.2 Profilo edificio scolastico

In riferimento, invece, al profilo dei consumi che simula un edificio scolastico (sottosezione 4.2.3) si consideri lo schema logico di base riportato in Figura 51.

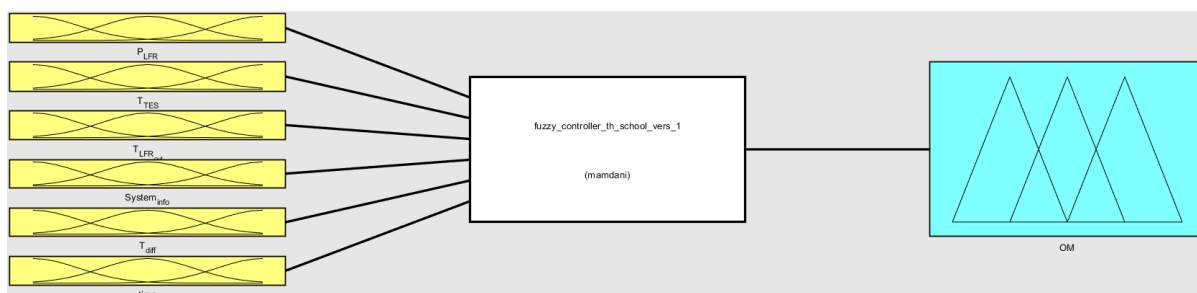


Figura 51: Schema logico della struttura del controllore avanzato per il profilo scuola. In giallo sono riportati i blocchi relativi alle variabili di ingresso, in blu quelli per l'output ed in bianco il blocco con l'insieme di regole fuzzy.

Questo controllore condivide la struttura base di quello per il profilo delle residenze private prima descritto. In aggiunta, con questo sistema, si avrà tra le variabili di ingresso anche il tempo e, in particolare, si prenderà in considerazione l'ora corrente nel calcolare l'*operative mode* più appropriata in funzione dei valori delle variabili di input. Avendo, come precedentemente detto, la struttura di base comune ai due controllori, di seguito verranno illustrate solamente le differenze tra i due, rimandando a quanto scritto precedentemente in questa sezione per il resto.

time Per la divisione in intervalli della variabile di stato *time*, si faccia riferimento alla Tabella 11 ed alla Figura 52 una rappresentazione grafica di tali insiemi.

Insieme fuzzy	Insieme di appartenenza di time
OK	$time \in [7, 14]$

Tabella 11: Divisione in intervalli della variabile *time*. Il colore verde rappresenta un insieme crisp.

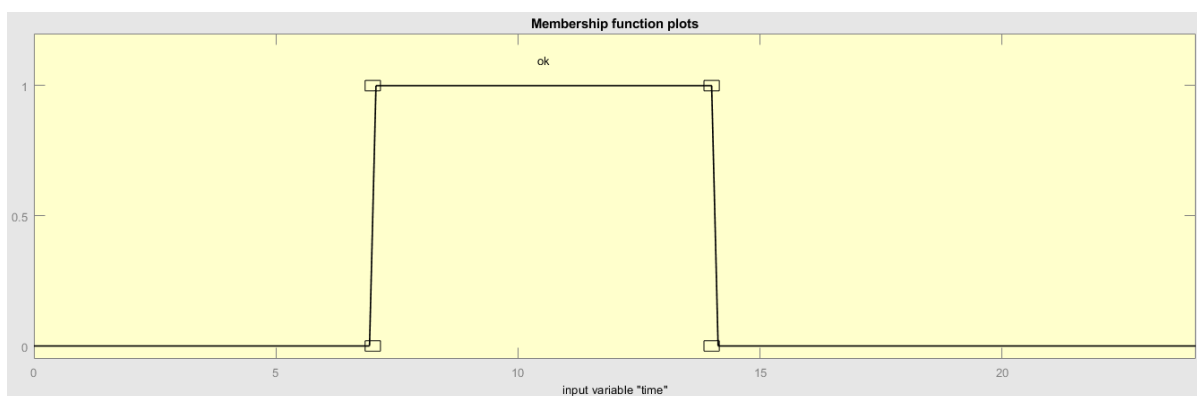


Figura 52: Rappresentazione grafica degli insiemi fuzzy individuati per la variabile *time*.

Le regole fuzzy implementate in questo controllore sono mostrate in Tabella 12. Anche in questo, avendo una struttura comune, verranno riportate solo le nuove regole aggiunte o eventuali modifiche a quelle riportate in Tabella 10. Per le regole non mostrate si rimanda dunque alla stessa. Per una rappresentazione grafica di tutte le regole, invece, si faccia riferimento alla Figura 53.

Regola	P_{LFR}	T_{TES}	$T_{LFR_{out}}$	$System_{info}$	T_{diff}	time	OM scelta
3	OK			\neg Good		OK	OM_1
10	Zero	OK			Bad	OK	OM_5
11	Zero	High				OK	OM_5
12	Low	OK		Good		OK	OM_6
17	OK			\neg Good	Bad	\neg OK	OM_4
18	Zero	High				\neg OK	OM_2
19	Zero	OK				\neg OK	OM_2
20	Low	OK				\neg OK	OM_3
21	Low	High				\neg OK	OM_6

Tabella 12: Regole fuzzy implementate per il controllore avanzato per profilo scolastico.

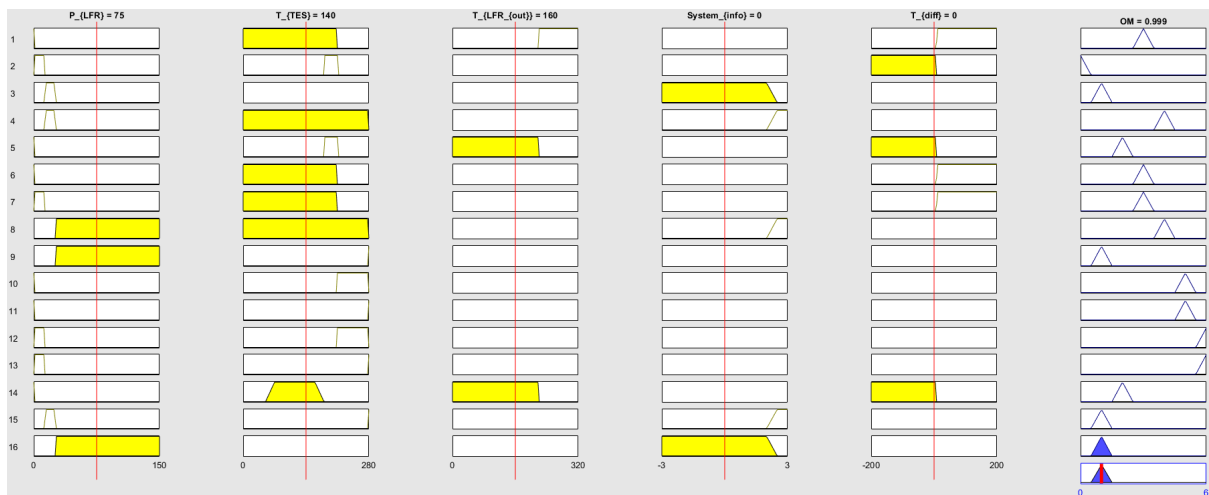


Figura 53: Rappresentazione grafica delle regole fuzzy implementate nel controllore avanzato per il profilo scolastico con un esempio di scelta dell'operative mode in funzione dei valori assunti dalle variabili di input (regola 16, OM_1).

In questa modalità di controllo, al contrario di quanto accadeva con il controllore avanzato per il profilo domestico, si cerca di produrre energia in una fascia oraria ben definita che rappresenta il possibile orario di apertura di una scuola, mentre al di fuori di questa si cerca, quando possibile di ricaricare il *thermal energy storage* in modo da averlo poi disponibile nel caso in cui non sia possibile utilizzare l'energia dal sole e quindi sfruttare il *linear Fresnel reflector* per alimentare l'*organic Rankine cycle* nella fascia oraria di apertura di una scuola.

In Figura 54 è mostrato uno *zoom* di una simulazione riferito alla differenza di *operative mode* tra controllore avanzato per il profilo di consumi privato e quello per il profilo scolastico. Il grafico rappresenta quindi la grandezza $OM_{std} - OM_{sch}$, con OM_{std} l'*operative mode* individuata dal controllore per il profilo di consumi privati ed OM_{sch} quella relativa al profilo di consumi di un edificio scolastico.

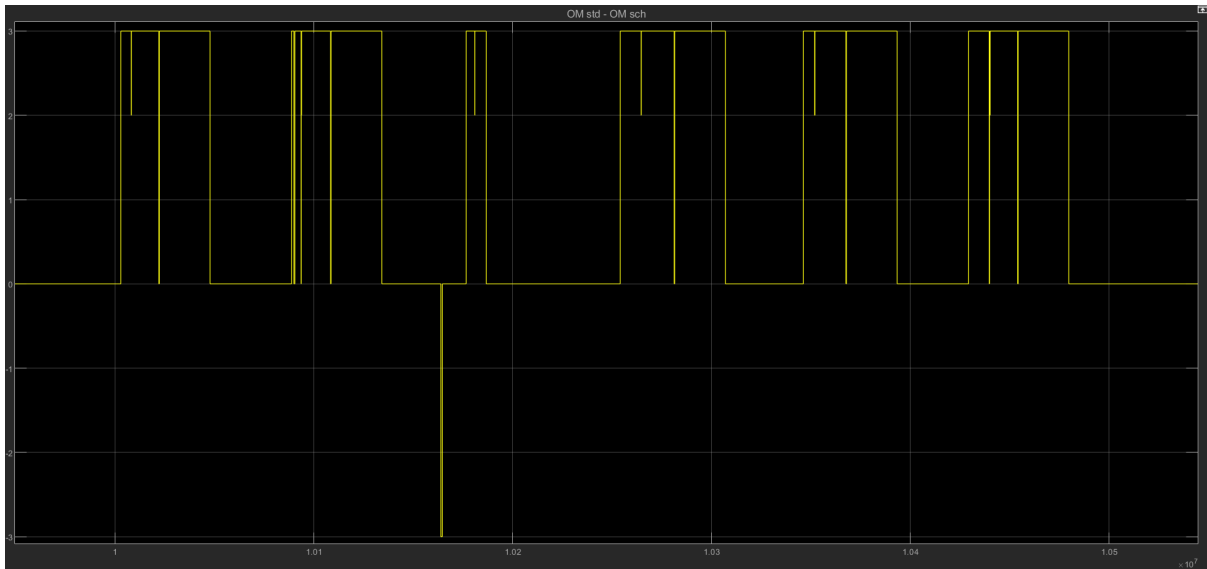


Figura 54: Differenza di *operative mode* calcolata da controllore per i diversi profili di consumi, ovvero $OM_{std} - OM_{sch}$.

Osservazione A causa del modesto irraggiamento solare che si ha nei mesi autunnali ed invernali, differenze concrete nel comportamento del controllore nei due possibili profili sono maggiormente visibili nei mesi primaverili (in cui il grafico in Figura 54 si colloca temporalmente) ed estivi, in cui l'irraggiamento solare è maggiore e quindi il sistema è in grado di produrre di più e per una finestra temporale maggiore nell'arco di una giornata.

5.2.3 Performance-based supervision

Poiché il controllore per l'inseguimento termico dipende dalla tipologia di profilo di consumo da inseguire, diventa utile poter monitorare il consumo del sistema al fine di scegliere il controllore più adatto alle esigenze. In particolare, si avranno 3 diverse modalità utilizzabili per il monitoraggio [S25]:

- **Pre-reouted supervision**, si valuta un controllore candidato dopo l'altro in sequenza (predefinita), fermandosi quando si ha un controllore che assicura delle performance accettabili. Una tecnica di questo risulta non essere efficace nel momento in cui il numero dei controllori candidati è grande;
- **Performance-based supervision** (diretta), si mantiene il controllore selezionato finché assicura buone prestazioni. Quando queste peggiorano fino a diventare inaccettabili, si cambia controllore con quello che conduce alle migliori performance attese secondo i dati disponibili;
- **Estimator-based supervision** (indiretta), si esegue una stima del processo sulla base dei dati osservati. Si seleziona quindi il controllore in relazione alla stima corrente effettuata.

Nella *performance-based supervision*, si avrà un supervisore che confronta le performance del sistema generando una misura di quanto bene il sistema si comporti in una certa situazione. Quindi, la selezione di un controllore piuttosto che un altro avverrà attraverso algoritmi basati su una logica decisionale piuttosto che attraverso un *tuning* continuo. La commutazione tra un controllore e l'altro verrà orchestrata da un sistema di più alto livello, ovvero il supervisore [16].

In particolare, la tecnica qui discussa nasce nel momento in cui il controllo classico basato su un singolo controllore non è sufficiente a garantire prestazioni adeguate. In questi casi, adottando il controllo supervisionato qui discusso, si implementerà un insieme di controllori tra i quali scegliere in base ad una serie di misure fatte online sul processo. La commutazione tra un controllore e l'altro verrà poi fatta da un'apposita logica decisionale che, in funzione delle misure fatte, valuterà le prestazioni del controllore candidato attualmente in uso e le performance potenziali che si avrebbero utilizzando uno degli altri controllori [S24]. L'architettura di base di una implementazione di questo tipo è mostrata in Figura 55.

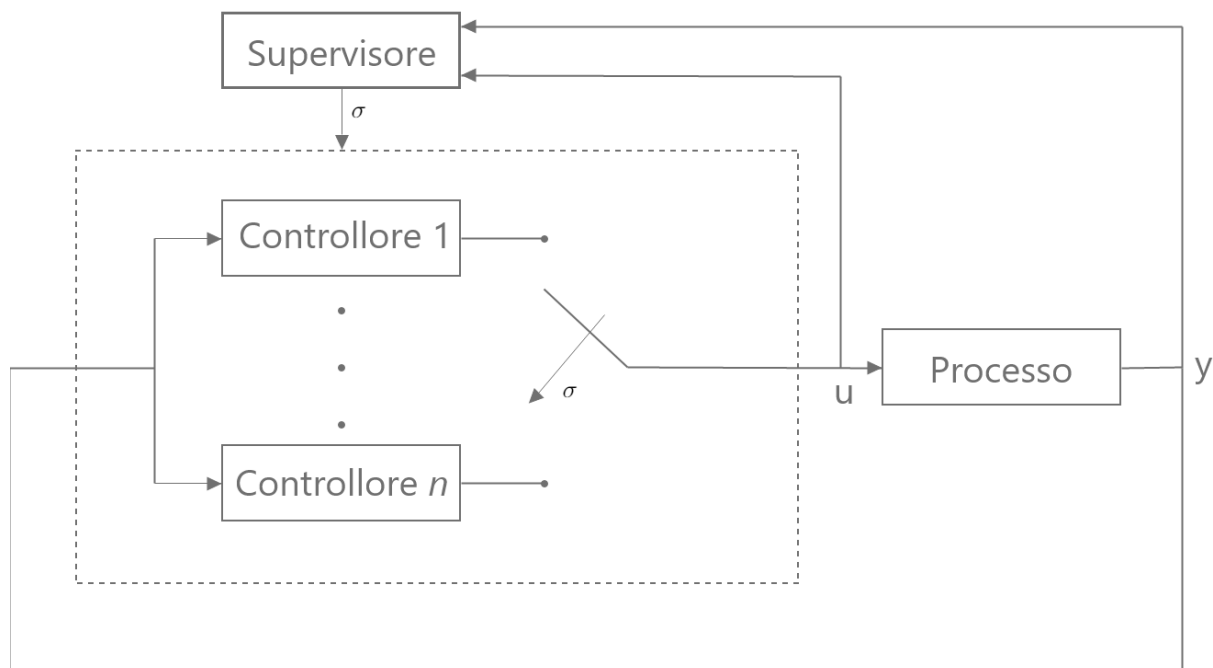


Figura 55: Schema logico di base dell'architettura utilizzata per la politica di *switching*.

La particolarità di una supervisione basata sulle performance sta nel fatto che il supervisore cerca di valutare le prestazioni dei controllori candidati senza passare attraverso una stima del modello del processo, ma tramite il *performance signal* π_q , con $q \in Q$ variabile di parametrizzazione dei controllori che indica quanto bene il controllore C_q si comporterebbe nell'ipotesi teorica in cui l'attuale segnale di controllo u sia generato da C_q in risposta alla misura y ottenuta dal processo.

Questo tipo di supervisione, quindi, consiste in un *performance monitor* che genera il segnale π_q , $q \in Q$ ed una logica decisionale (*logic-based switching*) che, sulla base del *performance signal*, genera il segnale di commutazione (*switching signal*) σ . Uno schema concettuale di quanto appena descritto è riportato in Figura 56.

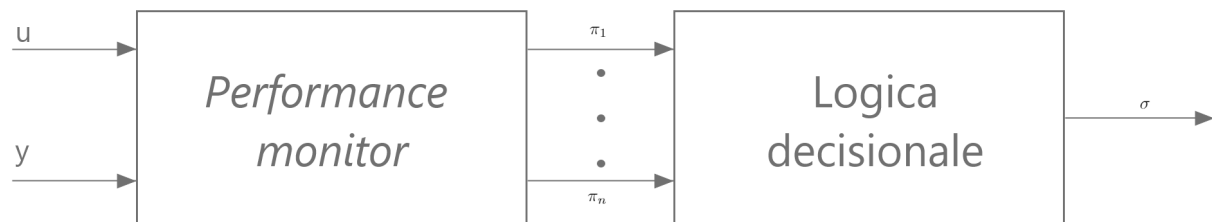


Figura 56: Schema logico di base di un supervisore basato su prestazioni. π_i rappresenta il *performance signal*, y l'uscita (misurata) del sistema, u il segnale di controllo e σ lo *switching signal*.

Il supervisore, infine, sarà dato dai seguenti blocchi funzionali considerando il caso generico di processo con modello non noto con esattezza affetto da disturbo d e rumore (limitato) n :

- **Multi-estimator**, ovvero un sistema dinamico che, presi come ingressi il segnale di controllo u e l'uscita del processo y , produce come uscita il segnale y_p , con $p \in P$ e P insieme che rappresenta il range di incertezza sul processo;
- **Monitoring signal generator**, ovvero un sistema dinamico che, preso in input l'errore di stima $e_p = y - y_p$, produce in output il *monitoring signal*;
- **Switching logic**, ovvero un sistema dinamico che, preso in input il *monitoring signal*, produce in output lo *switching signal*.

Con il *logic-based switching*, quindi, in base alle condizioni attuali del sistema vengono cambiati i controllori con il processo che rimane sempre lo stesso.

Si cerca quindi di migliorare le performance utilizzando il miglior controllore tra quelli candidati in funzione delle condizioni attuali del sistema, oppure, nel caso in cui nessuno dei controllori da solo è in grado di soddisfare i requisiti del problema si può avere una possibile connessione di questi (in serie, parallelo, ecc.). Lo *switching signal* avrà generalmente un andamento del tipo di quello mostrato in Figura 57.

L'implementazione del controllore discusso in questo elaborato trae ispirazione dall'architettura appena descritta, in particolare per la parte di stima dei consumi (variabile $System_{info}$, sottosezione 5.2). In particolare, le performance attuali del sistema concorreranno alla scelta dell'*operative mode* influenzando quindi il comportamento del sistema (al contrario di quanto descritto nella teoria appena illustrata, in questo caso si avrà un solo controllore che, in base alle prestazioni del sistema, modificherà il comportamento dello stesso).

A livello logico, nel sistema implementato, in analogia con quanto prima descritto, si avranno i seguenti blocchi funzionali:

- **Multi-estimator**, presa l'uscita del processo (potenza prodotta), fornisce una stima delle performance del sistema in relazione ai consumi dell'utenza connessa;
- **Monitoring signal generator**, ovvero un sistema dinamico che, presi i valori prodotti dal *multi-estimator*, produrrà in output il *monitoring signal*;
- **Switching logic**, rappresentata dal controllore fuzzy prima descritto (sottosezione 5.2) che, anche in funzione del *monitoring signal* generato, sceglierà l'*operative mode* più appropriata alle condizioni operative.

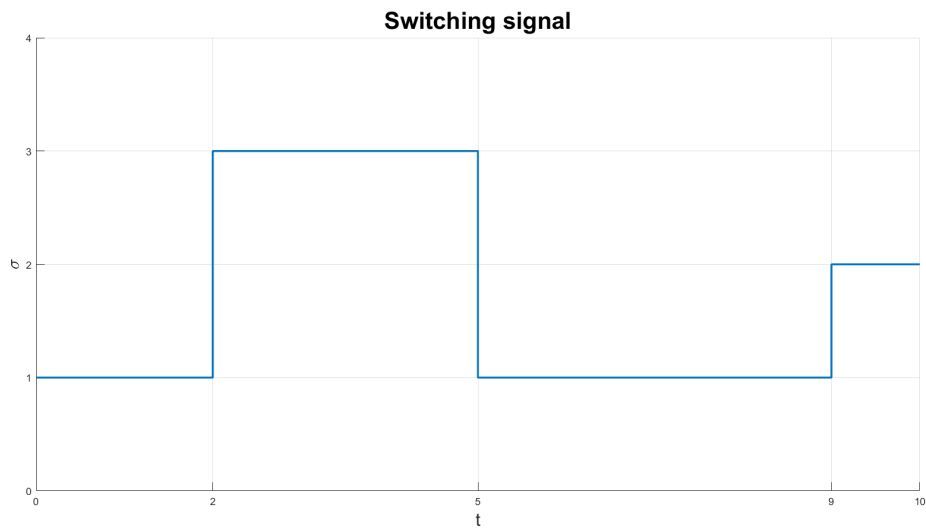


Figura 57: Esempio di *switching signal*. Per $t = 2$, $t = 5$ e $t = 9$ si hanno gli *switching time*.

Mentre la *switching logic* è stata di fatto già descritta parlando del controllore, per il *multi-estimator* ed il *monitoring signal generator*, si rimanda alla sottosezione 5.2.4.

5.2.4 Stima delle prestazioni del sistema in funzione dei consumi

Come precedentemente accennato nella sottosezione 5.2.3, l'implementazione di questa parte del sistema si basa sull'architettura di base della supervisione basata su prestazioni. In particolare, quindi, si avrà un segnale (ovvero $Sytem_{info}$) il cui valore esprime una stima delle performance del sistema e che concorrerà, insieme alle altre variabili di stato, alla scelta dell'*operative mode* (sottosezione 5.2). Lo schema implementativo per la generazione del segnale realizzato in Simulink è riportato in Figura 58. L'ingresso di tale schema è dato da $u(t) = E_p(t) - E_c(t)$, con $E_p(t)$ energia termica prodotta dall'impianto ed $E_c(t)$ energia richiesta dall'utenza (in kJ).

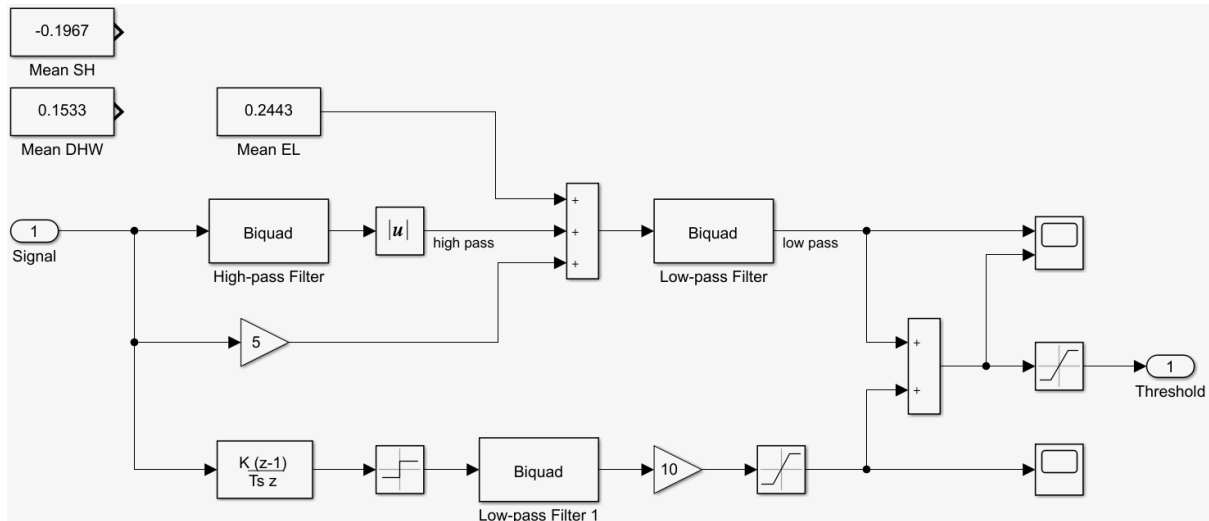


Figura 58: Schema a blocchi in Simulink per la generazione del segnale rappresentante la stima delle prestazioni del sistema.

L'architettura riprende quella per la generazione di una soglia adattiva nel campo utilizzata, ad esempio, nell'individuazione guasti con rilevamento a soglia [11]. In una possibile soluzione del problema di individuazione guasti, una soglia adattiva può essere implementata a livello logico con i blocchi mostrati in Figura 59.

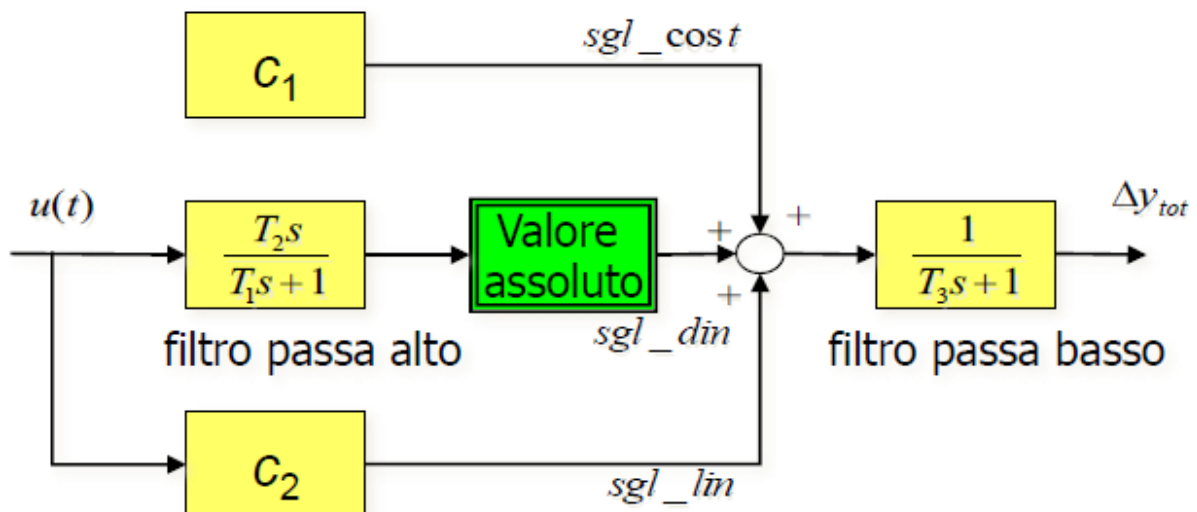


Figura 59: Schema logico dei blocchi che costituiscono una soglia adattiva.

Il filtro passa alto ha lo scopo di aumentare il valore della soglia andando a pesare le componenti ad alta frequenza del segnale e quindi considerando il contributo del rumore. La costante C_1 , invece, rappresenta il valor medio del segnale, mentre C_2 ha lo scopo di pesare la variabilità del segnale essendo moltiplicata per lo stesso. Infine, la somma dei 3 contributi appena descritti viene filtrata mediante un filtro passa basso per rendere l'andamento della soglia più regolare e smussato.

In Figura 60 è mostrata la stessa architettura implementata per la generazione del segnale di valutazione delle performance del sistema descritta per la costruzione di una soglia adattiva. In Figura 61, invece, è presente una parte aggiuntiva che concorre alla creazione del segnale che ha il compito di considerare anche il *trend* attuale del segnale. Questo viene fatto sommando a quanto ottenuto dalla prima parte del sistema il segnale dato dal segno della derivata dell'ingresso, opportunamente pesato, e quindi filtrato in modo da eliminare eventuali brusche variazioni.

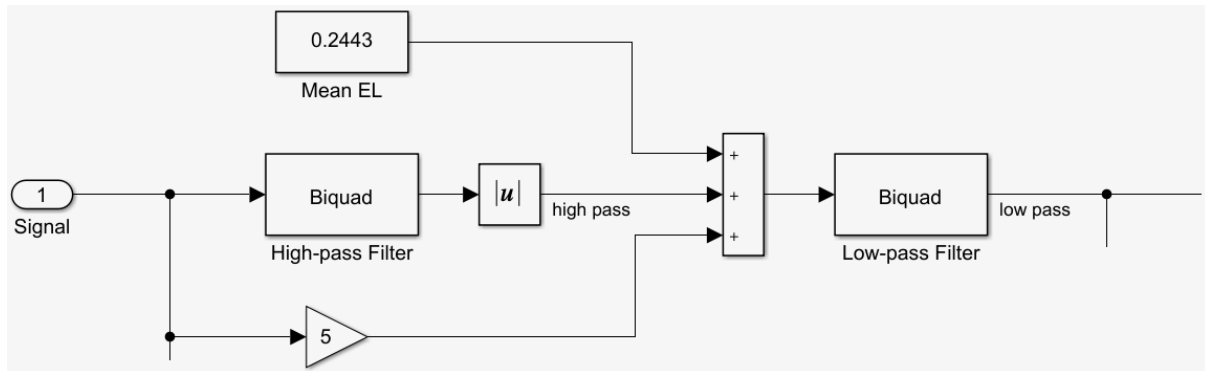


Figura 60: Schema a blocchi della prima parte di costruzione del segnale, in cui si ha una sogliatura effettuata come nel caso di soglia adattiva.

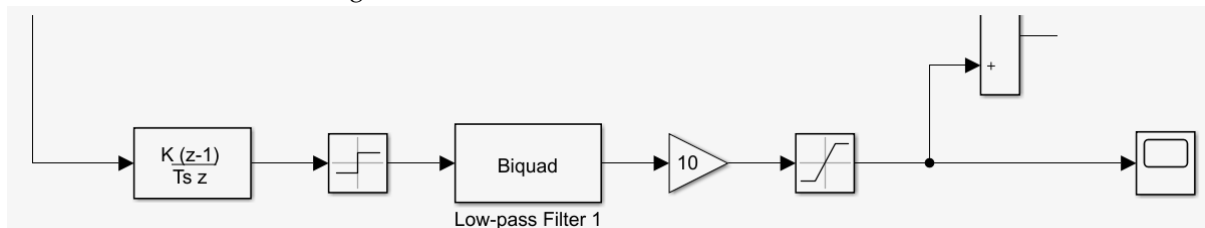


Figura 61: Schema a blocchi della seconda parte di costruzione del segnale, in cui si ha la pesatura del *trend* del segnale di ingresso.

Infine, aggiungendo un blocco saturazione al segnale così costruito, da ora $\sigma(t)$, si fa in modo che questo sia contenuto in un range ben definito, in modo da avere coerenza con l'interpretazione del significato dello stesso, avendo $\sigma(t) \in [-4, 4] \forall t \in \mathbb{R}_0^+$.

Infine, $\sigma(t)$ che, come detto appartiene all'insieme $[-4, 4] \forall t \in \mathbb{R}_0^+$, viene elaborato da un controllore in logica fuzzy utilizzato per la semplicità che garantisce nell'eseguire un'operazione di raggruppamento di valori.

Infatti, preso $\Sigma(t)$ come il segnale di uscita da questo blocco, si avrà:

$$\Sigma(t) = \begin{cases} -3, & \sigma(t) \in [-4, -1] \\ -1, & \sigma(t) \in [-1, 1] \\ 1, & \sigma(t) \in [1, 2.2] \\ 3, & \sigma(t) \in [2.2, 4] \end{cases}$$

Graficamente, l'operazione appena descritta è mostrata in Figura 62. In Figura 63, invece, è riportato l'andamento dei segnali $\sigma(t)$ e $\Sigma(t)$ appena descritti.

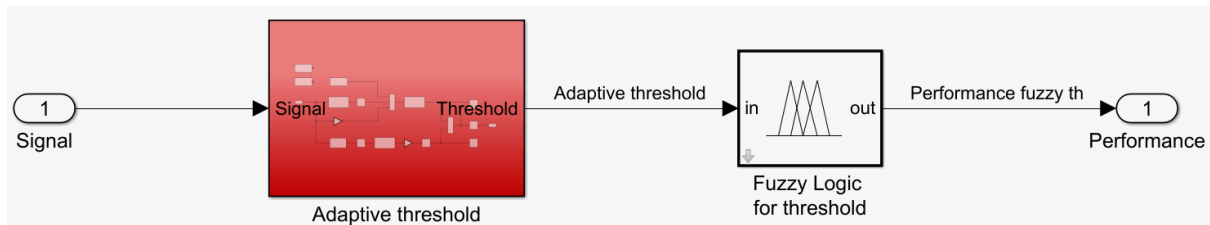


Figura 62: Schema a blocchi dell'elaborazione del segnale $\sigma(t)$. Il blocco *Adaptive threshold* contiene quanto riportato in Figura 58.

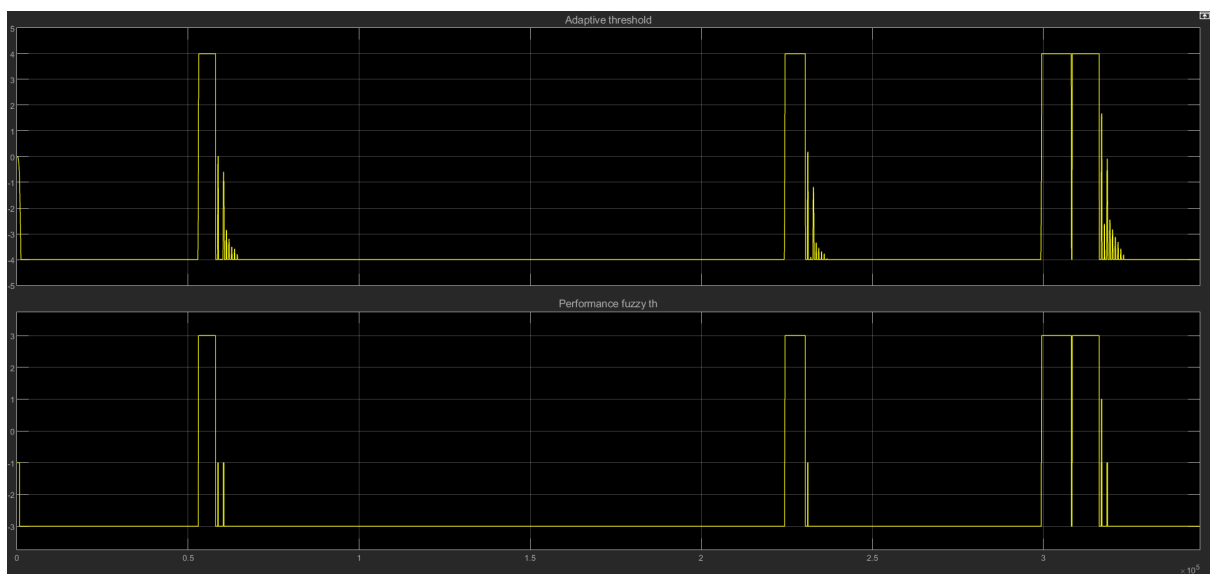


Figura 63: Andamento nel tempo dei segnali $\sigma(t)$ (in alto, indicato con *Adaptive threshold*) e $\Sigma(t)$ (in basso, *Performance fuzzy th*).

In conclusione, l'interpretazione data dai vari valori che $\Sigma(t)$ può assumere è riportata in Tabella 13.

$\Sigma(t)$	$u(t) = E_p(t) - E_c(t)$	$\frac{d}{dt}u(t)$	Descrizione
-3	< 0	< 0	Peggiori prestazioni possibili, $E_c(t) > E_p(t)$ (consumi maggiori della produzione di energia) e con <i>trend</i> decrescente
-1	< 0	> 0	Prestazioni negative, $E_c(t) > E_p(t)$ (consumi maggiori della produzione di energia), ma con <i>trend</i> crescente
1	> 0	< 0	Prestazioni $E_c(t) < E_p(t)$ (consumi minori della produzione di energia), ma con <i>trend</i> decrescente
3	> 0	> 0	Migliori prestazioni possibili, $E_c(t) < E_p(t)$ (consumi minori della produzione di energia) e con <i>trend</i> crescente

Tabella 13: Interpretazione data al segnale $\Sigma(t)$ in relazione all'ingresso $u(t)$ e alla sua derivata nel tempo.

5.2.5 Implementazione del controllore in MATLAB/Simulink

Lo schema a blocchi realizzato in Simulink che implementa il controllore avanzato per l'inseguimento termico (sottosezione 5.2) è riportato in Figura 64.

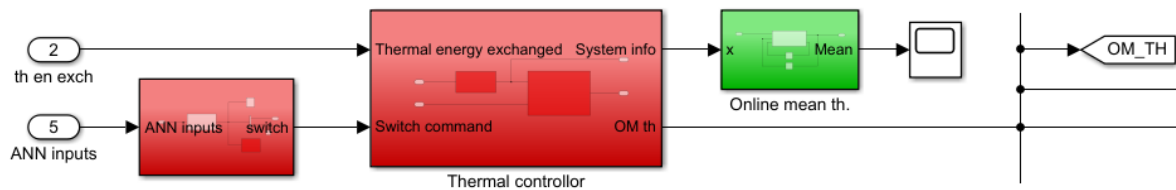


Figura 64: Schema a blocchi in Simulink del controllore per l'inseguimento termico.

I blocchi principali che lo compongono sono:

- **Blocco per l'identificazione del segnale**, che contiene la rete neurale implementata per l'identificazione del segnale;
- **Blocco che implementa il controllore**, contenente, oltre che il controllore in logica fuzzy anche il sottosistema per la generazione del segnale $System_{info}$ (sottosezione 5.2.4);
- **Blocco per calcolare la media online** (di $System_{info}$), in modo da avere una stima delle prestazioni del sistema.

Di seguito verrà data una descrizione dettagliata di ognuno di questi blocchi.

Rete neurale Per l'identificazione del segnale $u(t) = E_p(t) - E_c(t)$, ovvero la differenza tra l'energia termica prodotto e quella consumata dall'utenza (in kWh), si è implementata in MATLAB una rete neurale con i seguenti input:

- **Ora** del giorno;
- **Giorno** attuale;
- $u(t)$;
- $\bar{u}(t)$, ovvero il valor medio (calcolato online) del segnale;
- $\sigma_u^2(t)$, ovvero la varianza (calcolata online) del segnale;
- $RMS_u(t)$, ovvero il valore quadratico medio (calcolato online) del segnale.

La creazione dell'input della rete neurale è mostrato in Figura 65.

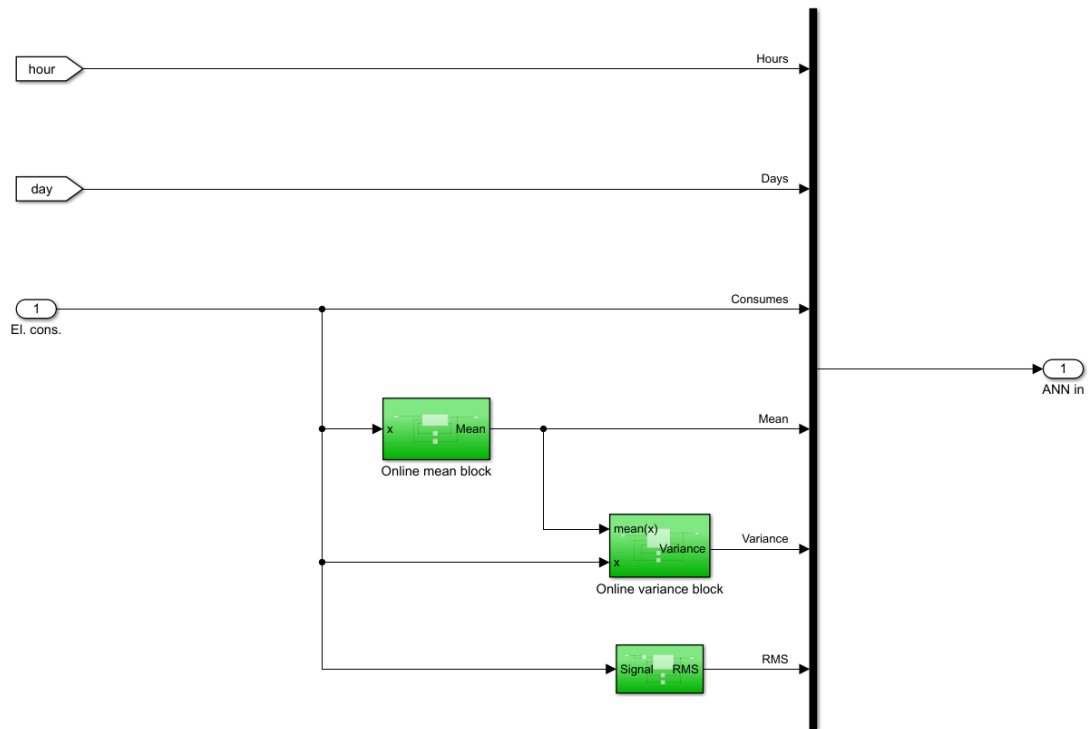


Figura 65: Schema a blocchi in Simulink per la creazione dell'input della rete neurale.

La rete neurale vera e propria (implementata come un *multi-layer perceptron*, sottosezione 3.2.2), è mostrata in Figura 66. Invece, in Figura 70 e 71 vengono mostrati i risultati dell'identificazione nel momento in cui si ha una commutazione del segnale, che passa quindi da un profilo all'altro (dal profilo per i consumi domestici a quello dei consumi scolastici e viceversa, rispettivamente).

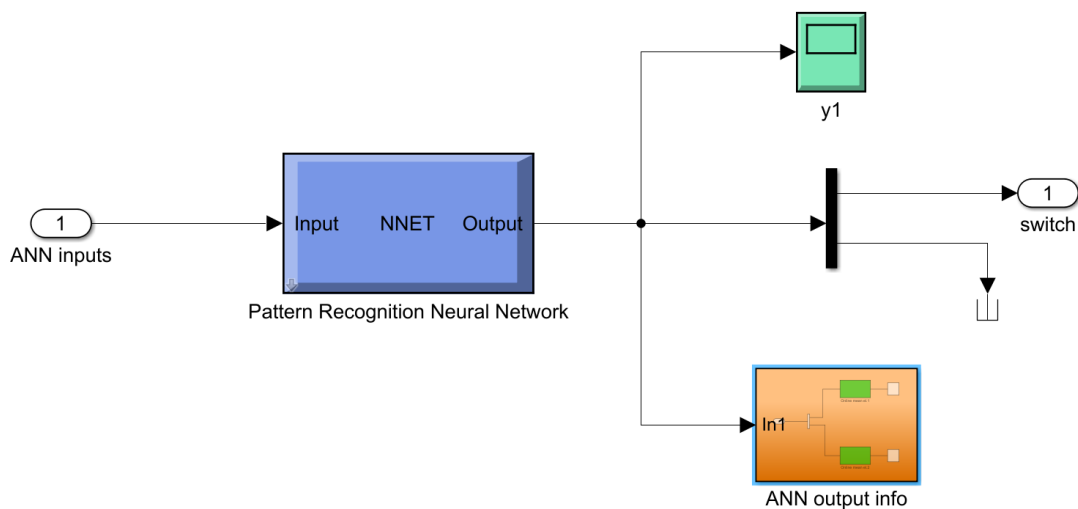


Figura 66: Schema a blocchi in Simulink della rete neurale per l'identificazione del segnale.

Per quanto riguarda i dati utilizzati per la fase di addestramento della rete, si sono utilizzati le sequenze temporali dei segnali in input precedentemente descritti ottenuti nell'arco di simulazione e relativi a diversi mesi, in modo da ottenere campioni collocabili sia in periodo invernale che autunnale/primaverile ed estivo. Per i parametri di sintesi, invece, un riassunto è mostrato in Figura 67.

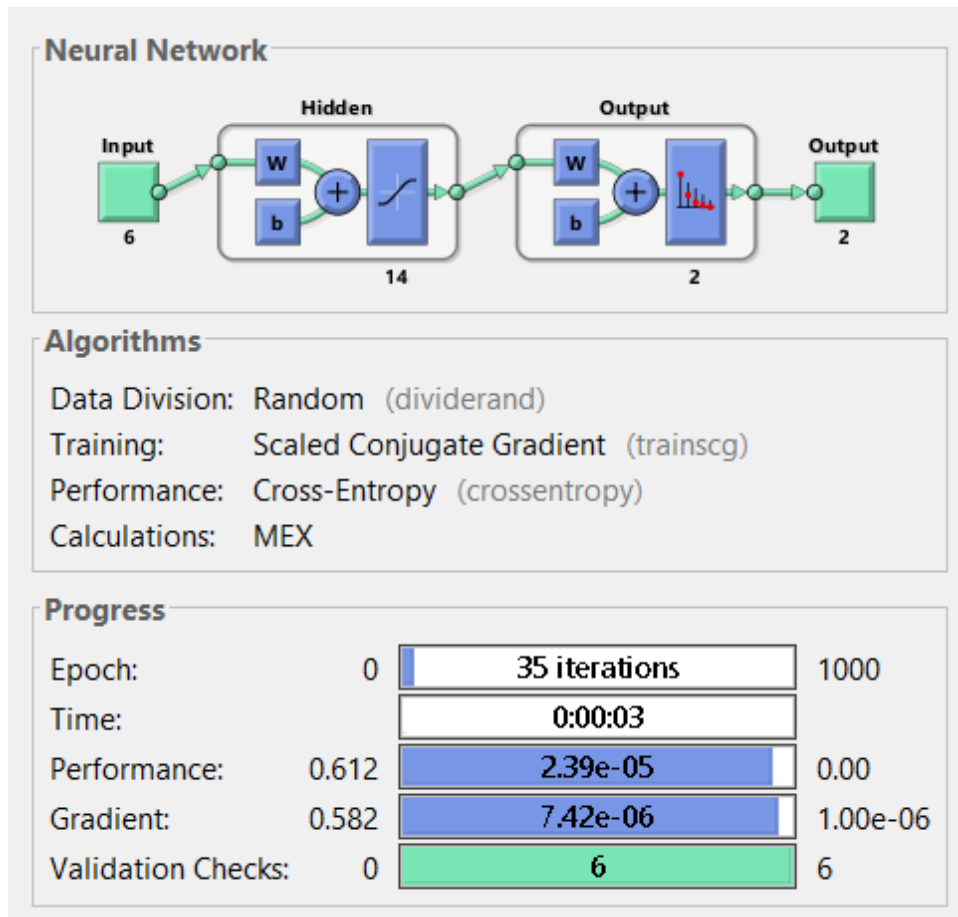


Figura 67: Parametri di sintesi principali della rete.

Dall'immagine si può osservare che:

- i dati prima descritti sono stati divisi in modo casuale per la fase di *training*, test e validazione in cui il primo insieme contiene il 70% dei campioni totali, mentre gli altri il 15% ciascuno;
- il numero di epoche di addestramento è, complessivamente, 35. Questo vuol dire che sono stati sottoposti alla rete i dati di addestramento 35 volte al fine di migliorare le prestazioni di riconoscimento del segnale;
- l'algoritmo utilizzato in fase di addestramento è quello del gradiente coniugato, ovvero una modifica al metodo descritto in 3.2.2 che consente di ridurre il numero di iterazioni dell'algoritmo.

A livello strutturale, invece, si può osservare dall'immagine nella rete quanto segue:

- la rete presenta uno strato nascosto tra input ed output;
- complessivamente, lo strato nascosto è composto da 14 neuroni;
- la funzione di attivazione dei neuroni è una sigmoide.

I risultati ottenuti con le scelte appena discusse sono visibili in Figura 68 e 69 in cui, rispettivamente sono riportati il grafico con le performance di validazione (valutando l'accuratezza della rete mediante la cross-entropia come funzionale da minimizzare) e la matrice di confusione (utile per osservare la distribuzione degli errori di classificazione).

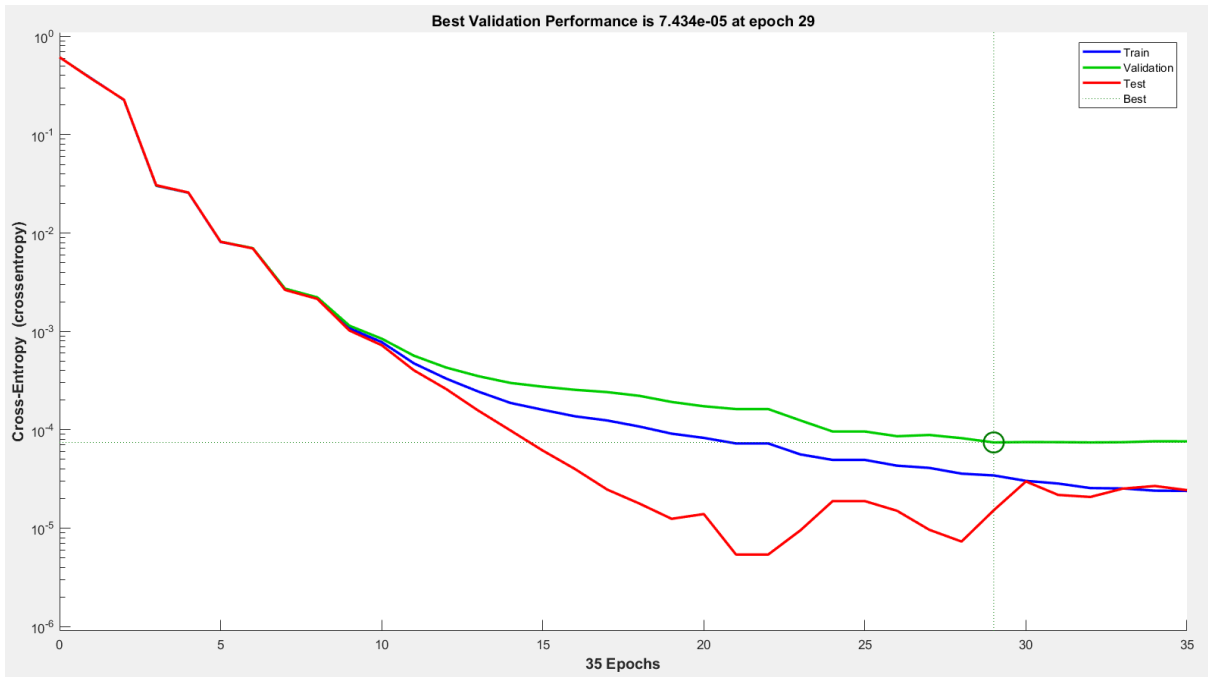


Figura 68: Grafico con le performance della rete neurale nelle varie epoche di addestramento.

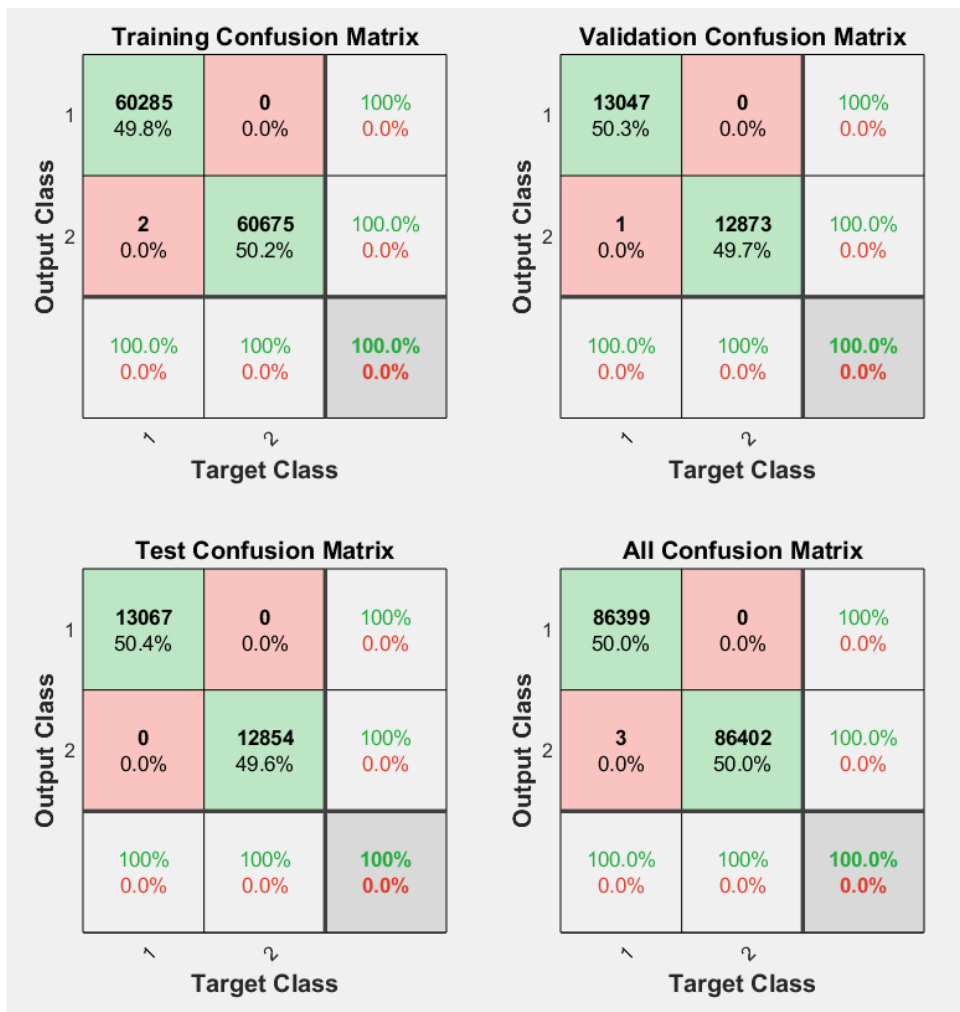


Figura 69: Matrice di confusione della rete neurale con distribuzione degli errori.

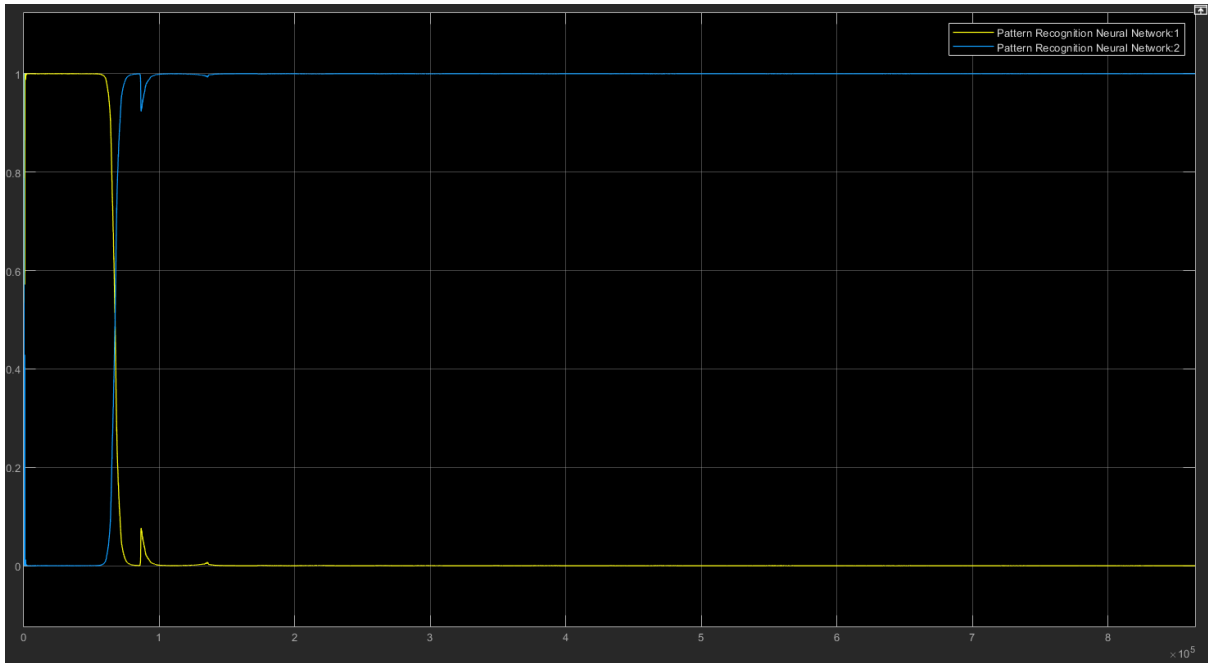


Figura 70: Identificazione del segnale in caso di *switch* dal profilo domestico a quello scolastico.

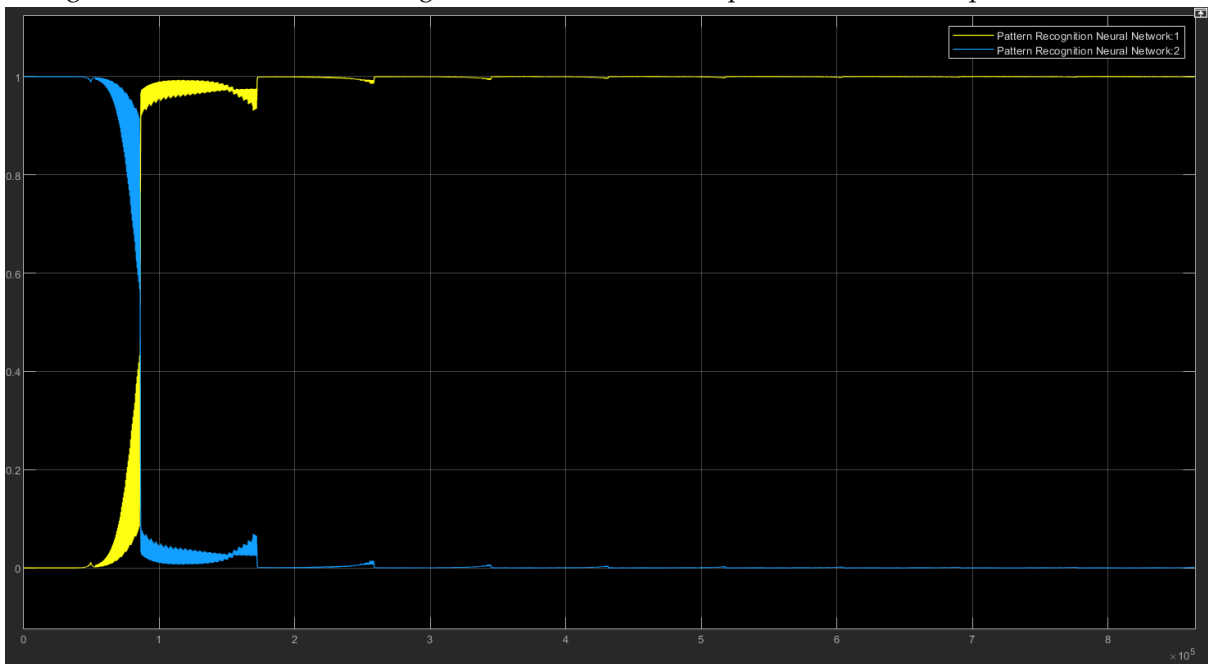


Figura 71: Identificazione del segnale in caso di *switch* dal profilo scolastico a quello domestico.

Controllore Il controllore riprende ed implementa quanto detto nella sottosezione 5.2.

In funzione, quindi, delle variabili di stato e viste le regole fuzzy prima descritte, in questo sottosistema si avrà la scelta dell'*operative mode* più adatta per il sistema.

In aggiunta a quanto detto, in questo sottosistema si avranno:

- creazione dell'ingresso per il controllore fuzzy, ovvero un segnale formato dall'insieme delle variabili di stato (quindi P_{LFR} , T_{TES} , T_{diff} , T_{LFR} e $System_{info}$);
- elaborazione dell'OM determinata dal controllore in modo da avere un segnale i cui valori siano dei numeri interi. Infatti, il controllore produce in uscita dei segnali nell'insieme dei numeri reali che sono il risultato dell'insieme delle regole fuzzy più adatte in funzione dell'ingresso del controllore. Inoltre si manipola l'OM così ottenuta in modo da evitare alcuni *switch* aggiuntivi che possono compromettere in parte le prestazioni del sistema. In particolare, infatti, potrebbe accadere che, a causa di oscillazioni nei segnali in input, l'OM passi da 1 (produzione di energia tramite l'*organic Rankine cycle*) a 3 (ricarica del TES) andando così ad abbassare le prestazioni del sistema;
- controllo per impostare eventualmente l' OM_{-1} nel caso in cui T_{TES} raggiunga una temperatura troppo elevata.

Complessivamente, lo schema a blocchi implementato in Simulink è mostrato in Figura 72.

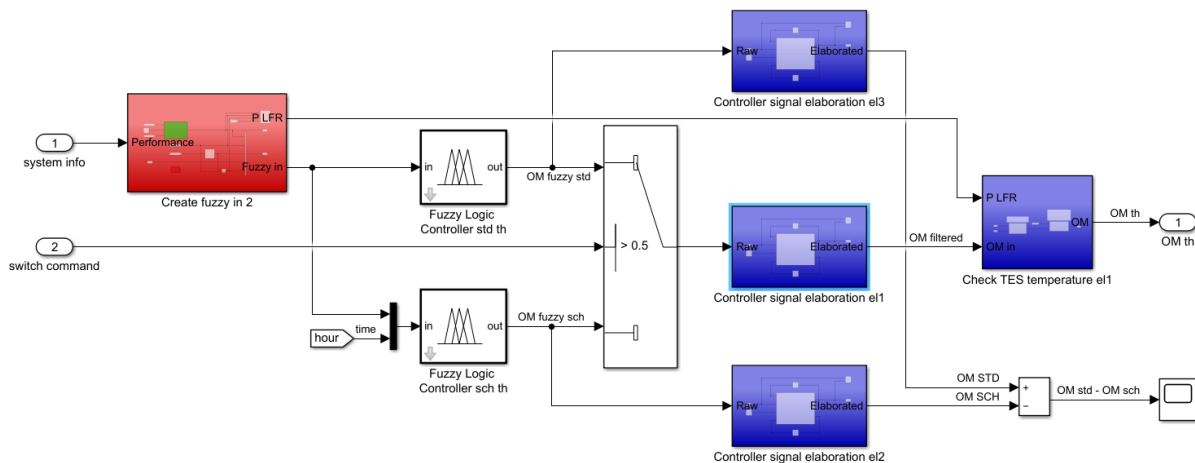


Figura 72: Schema a blocchi in Simulink della del controllore fuzzy e blocchi per l'elaborazione aggiuntivi.

La scelta di quale controllore utilizzare in funzione di quale profilo di consumi e quindi di quale tipo di utenza sia collegata al sistema viene fatto dalla rete neurale, il cui output, nel momento in cui supera o meno una soglia, determina quale uscita dei due controllori venga presa in considerazione ed utilizzata per impostare la modalità di lavoro effettiva del sistema.

Media online Il blocco per la media online della grandezza $System_{info}$ viene utilizzato per avere una stima generale per le performance del sistema.

5.3 Controllore per la massimizzazione della produzione

Mentre la logica con cui opera il controllore per la massimizzazione dell'energia prodotta è stata introdotta nella sottosezione 4.3, in questa sezione verrà descritto il modo in cui il peso w_t e la sua derivata nel tempo δw_t concorrono nel modificare il comportamento del sistema. Infatti, per produrre tramite l'ORC, si ha bisogno di una potenza minima P_σ proveniente dall'LFR (potenza di attacco dell'ORC). Questa soglia, nel controllore base ed avanzato è pari a $13kWh$ mentre, nel controllore per la massimizzazione dell'energia, verrà modificata in funzione del segno di δw_t .

5.3.1 Variazione di P_σ

Come già precedentemente illustrato, si avrà:

1. Se $\delta w_t \geq 0$, allora la potenza generata dall'LFR cresce, ma non si sta producendo energia. In questo caso si diminuisce P_σ in modo da produrre anche con potenze minori provenienti dall'LFR;
2. Se $\delta w_t < 0$, allora l'energia prodotta cresce in modo maggiore rispetto l'energia dall'LFR e si sta producendo in condizioni standard e quindi, per una questione di rendimento, si aumenta P_σ in modo da non produrre sempre in condizioni svantaggiose.

Nello specifico, dopo aver descritto come varia la soglia P_σ , l'implementazione del controllore per la massimizzazione della produzione si basa sull'*operative mode* calcolata dal controllore avanzato e quindi su quanto riportato in 5.2 e relative sottosezioni. Infatti, il controllore qui descritto modifica l'OM prodotta da quello avanzato solo sotto certe condizioni operative di seguito riportate, mentre la lascia invariata nella restante parte dei casi. Per un esempio della modalità con cui varia P_σ , si faccia riferimento a quanto riportato in Figura 73.

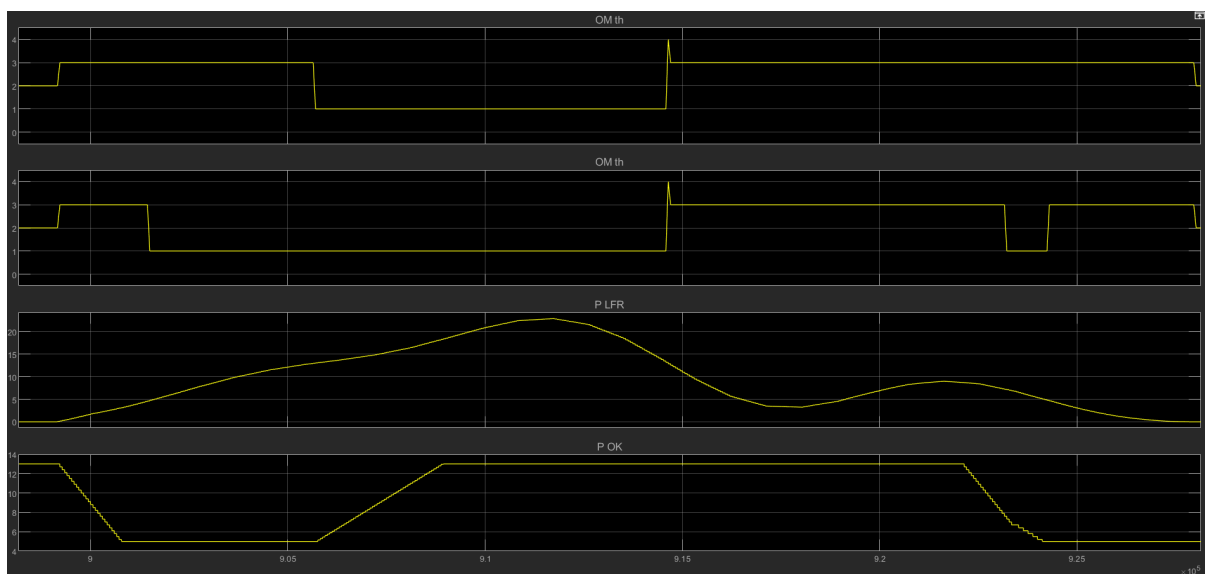


Figura 73: Zoom all'interno di una simulazione che mostra il modo in cui il controllore modifica la soglia P_σ (nell'immagine mostrata come P_{OK}) per la produzione mediante ORC.

Nel grafico appena mostrato si può vedere come, nel momento in cui si ha una potenza limitata dal *linear Fresnel reflector* la soglia P_σ si abbassi seguendo l'andamento di una rampa fino al valore minimo impostato. Nel momento in cui P_{LFR} raggiunge valori via via più alti (e maggiori della soglia massima di attacco di $13kJ$), la soglia inizia ad aumentare (sempre con un andamento a rampa) fino a raggiungere il valore massimo (sempre di $13kJ$). Infine, si può vedere come, nel momento in cui si verifica la condizione per cui P_{LFR} dapprima scende per poi risalire, la soglia si abbassa per fare in modo che si abbia una produzione aggiuntiva di energia. La produzione aggiuntiva appena citata rappresenta un *surplus* che non si avrebbe utilizzando uno degli altri controllori disponibili.

In Tabella 14, infine, vengono riportate le condizioni operative che devono essere verificate affinché venga cambiata l'*operative mode* calcolata dal controllore avanzato. Mentre la prima regola riportata in Tabella 14 ha validità generale, occorre comunque osservare come le regole 2 e 3 sono quelle che si alternano soprattutto nei mesi primaverili ed estivi in cui l'irraggiamento solare è maggiore e quindi il *thermal storage unit* può raggiungere temperature elevate, superiori di quella massima ammissibile. Le regole 5 e 6, invece, verranno utilizzate soprattutto nei mesi invernali ed autunnali, oltre che nelle fasce orarie serali e notturne nel periodo estivo, in cui l'irraggiamento solare è minore e quindi il *thermal storage unit* raggiunge temperature più contenute. In questo scenario, si è deciso di abbassare la temperatura per cui si possono avere l' OM_5 ed OM_6 dai circa 215°C ottimali a livello di rendimento a 200°C in modo da poter sfruttare maggiormente questo elemento del sistema.

Nella Tabella 18, vengono riportati i risultati ottenuti con il controllore descritto in queste sottosezioni (controllore per la massimizzazione della produzione). Per una discussione più approfondita si rimanda alla sezione 6 e relativa sottosezione.

Infine, per l'implementazione della logica di controllo descritta in questa parte dell'elaborato, si rimanda a quanto riportato nella sottosezione 5.3.2.

Osservazione Occorre comunque osservare che, mentre con il controllore avanzato il sistema lavora in condizioni pressoché nominali, con quello per la massimizzazione della produzione di energia si ha un calo del rendimento dell'*organic Rankine cycle*. Questo è dovuto al fatto che l'ORC viene fatto lavorare a potenze inferiori di quella minima consigliata per il sistema (infatti, si avranno istanti in cui $P_{\sigma} < 13kWh$). Sebbene è opportuno tenerne conto, in questo modo si assicura comunque una produzione aggiuntiva soprattutto nei mesi invernali.

Condizione operativa	Nuova OM	Descrizione
$P_{LFR} \in [P_{\sigma}, 26]$	OM_1	Il sistema produce energia tramite l'ORC se P_{LFR} è compresa nell'intervallo variabile $[P_{\sigma}, 26]$
$P_{LFR} \geq 26 \ \& \ T_{TES} \leq 278$	OM_4	Se l'LFR eroga una potenza elevata e la temperatura del TES non supera il valore massimo, si produrrà energia tramite l'ORC, caricando al tempo stesso il TES
$P_{LFR} \geq 26 \ \& \ T_{TES} > 278$	OM_1	Il TES ha una temperatura troppo elevata, per cui si produrrà energia tramite l'ORC senza che il TES venga ricaricato, in modo da non aumentarne ulteriormente la temperatura
$P_{LFR} \in (0, P_{\sigma}) \ \& \ T_{TES} > 200$	OM_5	L'LFR eroga una potenza limitata, ma il TES ha una temperatura tale da poter concorrere alla produzione di energia tramite l'ORC
$P_{LFR} = 0 \ \& \ T_{TES} > 200$	OM_6	L'LFR non eroga potenza, mentre il TES ha una temperatura per cui si può comunque produrre energia tramite l'ORC

Tabella 14: Modalità con cui viene modificata l'*operative mode* calcolata dal controllore avanzato.

5.3.2 Implementazione del controllore in MATLAB/Simulink

Per quanto riguarda il calcolo dei pesi w_e e w_t , si parte dalla considerazione che, nel caso il termine al denominatore (ovvero l'energia elettrica E_e e termica E_t , rispettivamente) sia nullo, allora sarà nullo anche il corrispettivo peso. In caso contrario questo sarà dato dal rapporto tra l'energia proveniente dal *linear Fresnel reflector* e l'energia prodotta presa in considerazione per il peso. Calcolato il peso così come appena descritto, questo verrà filtrato tramite una media mobile in modo da limitare le oscillazioni che si presentano soprattutto negli istanti iniziali della simulazione in cui $E_e, E_t \rightarrow 0$ e conseguentemente si avrebbe $w_e, w_t \rightarrow +\infty$.

Complessivamente, il blocco adibito al calcolo dei pesi è mostrato in Figura 74, in cui è presente anche uno stadio di integrazione in modo da considerare le grandezze energetiche espresse in *kWh*.

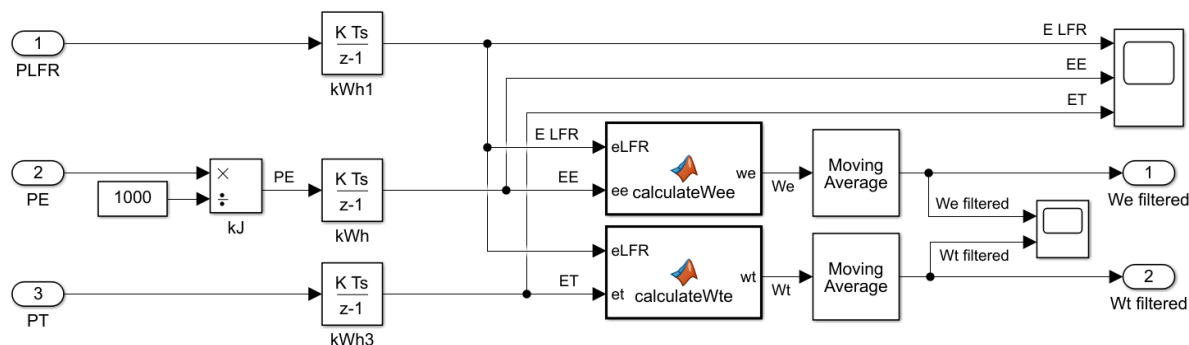


Figura 74: Schema a blocchi con l'implementazione del calcolo dei pesi w_e e w_t .

Successivamente al calcolo dei pesi, si avrà il blocco Simulink per effettuare la derivata delle grandezze ed ottenere così $\delta w_e = \frac{d}{dt}w_e(t)$ e $\delta w_t = \frac{d}{dt}w_t(t)$. A livello implementativo, il blocco che realizza questa funzionalità è mostrato in Figura 75. In tale blocco viene anche effettuata una scalatura della derivata (funzioni *analyzeWe* ed *analyzeWt*).

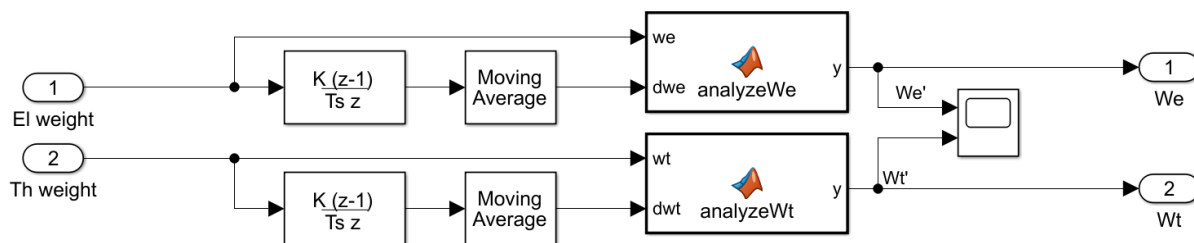


Figura 75: Schema a blocchi che calcola la derivate dei pesi w_e e w_t , effettuandone poi una scalatura.

Complessivamente, l'organizzazione dei due blocchi è mostrata in Figura 76. Il blocco *Calculate weights* effettua il calcolo di w_e e w_t convertendo il valore dell'energia da kJ/h in kWh mediante un'integrazione e moltiplicazione per il fattore di scale ed eseguendo poi il rapporto tra l'energia proveniente dall'LFR e quella elettrica e termica prodotta, rispettivamente, mentre *Analyze weights* ne calcola le derivate.

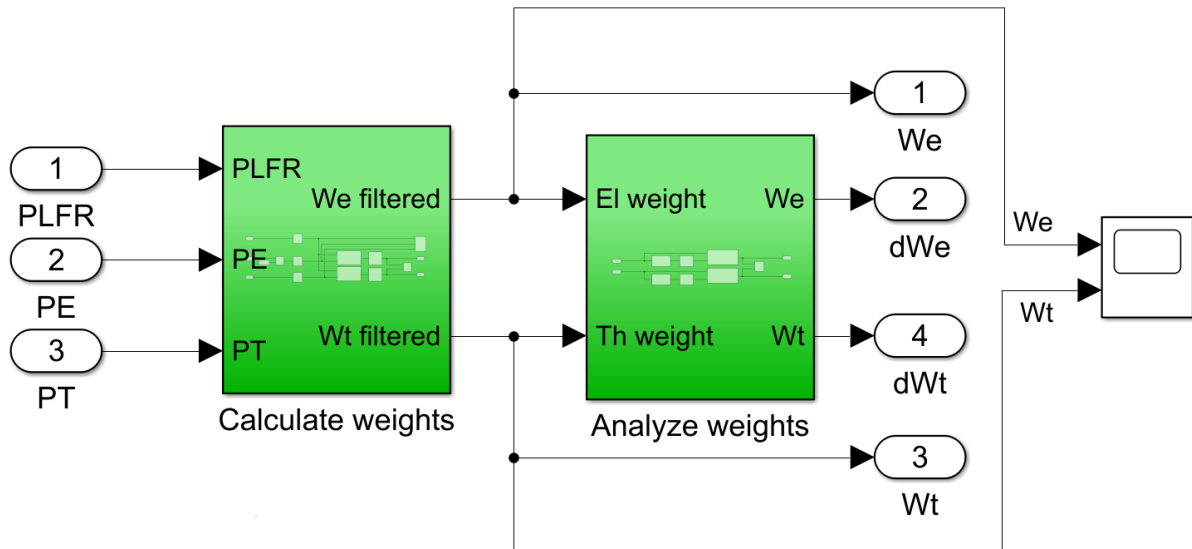


Figura 76: Schema a blocchi completo per il calcolo dei pesi e delle relative derivate.

Infine, in Figura 77 viene mostrato il blocco con il compito di modificare l'*operative mode* ottenuta dal controllore avanzato in funzione della logica descritta nelle sottosezioni 4.3 e 5.3.

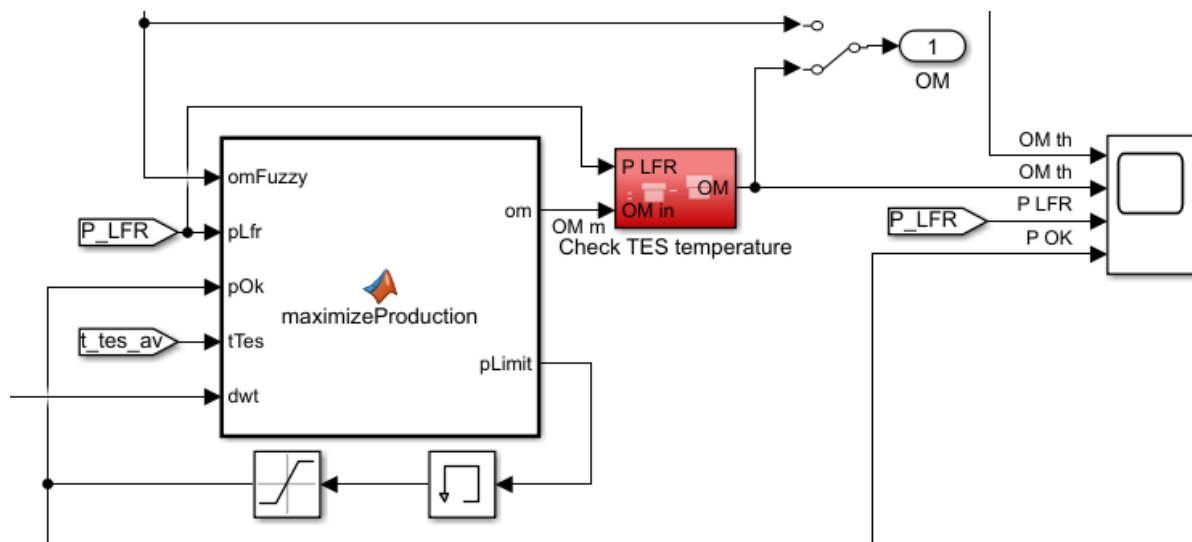


Figura 77: Schema a blocchi del controllore per la massimizzazione della produzione di energia.

Il fulcro del controllore è dato dalla funzione *maximizeProduction* che, in accordo con le condizioni operative e la logica prima descritta, effettivamente modifica l'*operative mode* determinata per incrementare la produzione di energia. Il codice implementato in questa porzione di sistema è riportato in Figura 78.

```

1  function [om, pLimit] = maximizeProduction(omFuzzy, pLfr, pOk, tTes, dwt)
2  -     pLimit = pOk;
3  -     om = omFuzzy;
4
5  -     if dwt >= 0 % Se wt' è non negativo (aumento di P_LFR e non di P_TH)
6  -         if pLfr > 0 && pLfr < pOk % Se la potenza dell'LFR è bassa
7  -             pOk = pOk - 0.3; % Diminuisco la soglia inferiore per OM1
8  -             pLimit = pOk; % Aggiorno la soglia
9  -         end
10 -    elseif pLfr > 13 % Se la potenza è sufficiente per OM1 e wt' è negativo
11 -        pOk = pOk + 0.15; % Aumento la soglia inferiore per OM1
12 -        pLimit = pOk; % Aggiorno la soglia
13 -    end
14
15 -    %% Modifica della modalità operativa in funzione di P_LFR e T_TES
16
17 -    if pLfr >= pOk && pLfr <= 26 % Range di potenza per OM1
18 -        om = 1;
19 -    elseif pLfr > 26 % Range di potenze per OM4
20 -        if tTes <= 278 % T_TES ok, riascladamento
21 -            om = 4;
22 -        else % T_TES elevata, solo ORC
23 -            om = 1;
24 -        end
25 -    elseif pLfr > 0 && tTes >= 200 % P_LFR < P_OK, bassa; T_TES alta
26 -        om = 6;
27 -    elseif tTes > 200 % P_LFR nulla, T_TES alta
28 -        om = 5;
29 -    end
30 -end

```

Figura 78: Codice implementato per la funzione *maximizeProduction*.

La prima parte della funzione ha il compito di modificare la soglia P_σ in accordo con quanto riportato nelle sottosezioni 4.3 e 5.3. In particolare, le righe di codice dalla 5 alla 9 fanno riferimento al caso in cui δw_i sia non negativa e quindi la soglia P_σ viene di conseguenza diminuita, mentre le righe dalla 10 alla 13 il caso in cui la derivata sia negativa e quindi viene aumentata la soglia per questioni legate al rendimento dell'*organic Rankine cycle*. Facendo riferimento a quanto riportato in Tabella 14 (sottosezione 5.3), infine, le righe di codice dalla 17 alla 18 implementano la regola 1, quella dalla 19 alla 21 la regola numero 2, dalla 22 alla 24 la 3, la 25 e 26 la 4 ed infine dalla 27 alla 29 la regola numero 5.

6 Risultati ottenuti

In questa sezione verranno discussi i risultati ottenuti con l'utilizzo dei nuovi controllori implementati. I risultati saranno valutati in termini di produzione aggiuntiva di energia termica, fascia orario di produzione della stessa ed in aggiunta verranno fatte valutazioni di tipo economico sui risparmi che si avrebbero adottando una soluzione di questo tipo per produrre energia termica ed elettrica invece che acquistarle dalla rete del fornitore. Occorre comunque osservare che la dimensione dell'impianto di cogenerazione è fatta in termini di energia termica e non elettrica, per cui ci si concentrerà più su questo aspetto nel discutere i risultati ottenuti.

Mentre in 6.1 e 6.2 verranno meglio descritti i risultati ottenuti con i controllori sviluppati, in Figura 79 viene riportato il miglioramento di prestazioni ottenuto rispetto il controllore base.

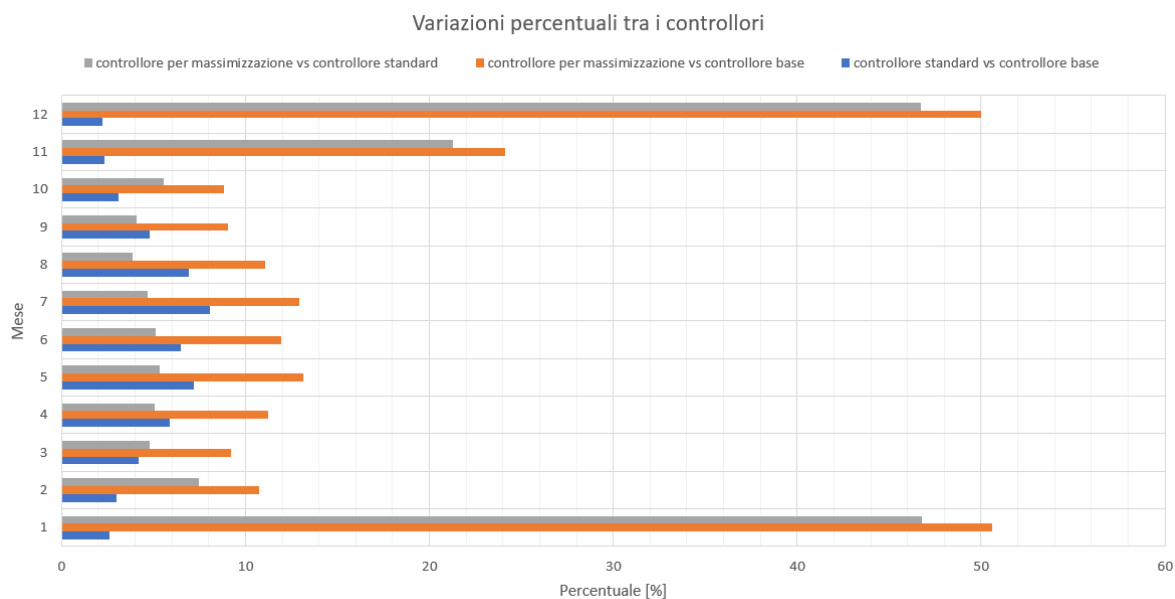


Figura 79: Variazione percentuale su scala mensile della produzione confrontando i controllori sviluppati in questo lavoro.

6.1 Controllore per l'inseguimento termico (avanzato)

Questo controllore assicura le prestazioni riportate nella Tabella 15. Si può osservare come il controllore avanzato assicuri un incremento di energia prodotta rispetto quello base dell'ordine del 5% circa annuo (la variazione percentuale, sempre in riferimento alla Tabella 15, varia a seconda del mese e quindi dall'irraggiamento solare diretto che consente al *linear Fresnel reflector* di raccogliere energia). Questo incremento delle prestazioni assicura, con una prima grossolana stima, un risparmio di circa 340€ (inteso come energia termica prodotta in più e quindi non più comprata dalla linea del fornitore) che si traduce (essendo i consumi e la produzione relativi a 4 residenze) a circa 85€ per appartamento. Una valutazione in termini economici delle prestazioni del controllore avanzato utilizzando il profilo dei consumi per abitazioni private è riportata in Tabella 16.

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Controllore base	1.115E3	3.050E3	4.957E3	5.978E3	7.080E3	7.850E3
Controllore avanzato	1.144E3	3.144E3	5.165E3	6.330E3	7.590E3	8.360E3
Variazione assoluta	0.029E3	0.120E3	0.208E3	0.343E3	0.510E3	0.510E3
Variazione percentuale	2.60%	2.98%	4.20%	5.89%	7.20%	6.50%
Mese	Lug	Ago	Set	Ott	Nov	Dic
Controllore base	8.530E3	7.510E3	6.290E3	3.860E3	1.700E3	0.900E3
Controllore avanzato	9.220E3	8.030E3	6.590E3	3.980E3	1.740E3	0.920E3
Variazione assoluta	0.690E3	0.520E3	0.300E3	0.120E3	0.050E3	0.020E3
Variazione percentuale	8.09%	6.92%	4.77%	3.11%	2.35%	2.22%

Tabella 15: Risultati ottenuti in termini di energia termica prodotta (in *kWh*) con il nuovo controllore in una simulazione relativa ad un anno in relazione a quelli ottenuti utilizzando il controllore base precedentemente sviluppato.

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Differenza di produzione	0.029E3	0.120E3	0.208E3	0.343E3	0.510E3	0.510E3
Valutazione economica	3€	12€	21€	34€	51€	51€
Mese	Lug	Ago	Set	Ott	Nov	Dic
Differenza di produzione	0.690E3	0.520E3	0.300E3	0.120E3	0.050E3	0.020E3
Valutazione economica	69€	52€	30€	12€	5€	2€

Tabella 16: Valutazione in euro dell'energia termica in *kWh* prodotta utilizzando il controllore base e quello avanzato (con profilo per l'inseguimento termico).

Nella Tabella sono presenti la differenza di produzione tra il controllore base e quello avanzato (ovvero $\Delta P = P_{base} - P_{standard}$, espressa in *kWh*) ed una prima stima economica della potenza aggiuntiva prodotta. Alla base della stima si è considerata la fascia *F0*, ovvero con la tariffa mono-oraria per cui il

prezzo dell'energia acquistata dalla rete rimane costante non variando quindi con l'orario della giornata. Per quanto detto, quindi la stima economica fatta va intesa come il costo per l'acquisto del surplus di energia prodotto nel caso si utilizzasse il controllore base e quindi il risparmio ottenuto utilizzando il controllore implementato in questo lavoro.

Da quanto riportato, si può osservare come il controllore avanzato consenta sempre di avere una produzione energetica maggiore rispetto quello base che si traduce in un risparmio annuo di 342€ che, ripartiti nei 4 ipotetici appartamenti collegati all'impianto, equivalgono a circa 86€ per abitazione.

Per quanto riguarda il periodo di produzione di energia termica giornaliera, invece, si faccia riferimento a quanto riportato in Tabella 17, in cui viene mostrato l'orario di inizio e fine produzione medio per ogni mese. In Figura 80 (per il controllore base) e 81 (per quello avanzato), invece, viene riportato in forma di grafico il contenuto della Tabella 17.

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Ora inizio controllore base	10.72	10.04	8.76	8.37	7.53	7.13
Ora inizio controllore avanzato	10.61	9.84	9.01	8.22	7.87	7.31
Ora fine controllore base	13.73	15.76	15.22	9.65	7.36	5.22
Ora fine controllore avanzato	14.58	17.81	14.49	8.34	7.09	5.81
Mese	Lug	Ago	Set	Ott	Nov	Dic
Ora inizio controllore base	7.37	8.20	8.37	8.11	10.18	11.35
Ora inizio controllore avanzato	7.37	8.20	8.37	8.11	10.18	11.35
Ora fine controllore base	6.29	9.08	10.18	17.69	12.94	11.37
Ora fine controllore avanzato	6.13	8.43	8.90	9.09	14.02	13.38

Tabella 17: Fascia oraria giornaliera di produzione di energia (con l'ora che viene rappresentata in base 10 e non sessagesimale).

Da quanto riportato nelle Figure 80 e 81, quindi, si può osservare come il controllore avanzato assicuri una produzione di energia in un intervallo di tempo più lungo durante l'arco di una giornata. Infatti, i grafici riportati, vanno interpretati come segue:

- Nei **mesi invernali** (gennaio, febbraio, novembre, dicembre in particolare), un orario di inizio produzione minore e di fine maggiore equivalgono ad un intervallo temporale di produzione di energia più lungo;
- Nei **mesi primaverili ed estivi**, la produzione di energia può iniziare in un giorno e finire nelle prime ore del successivo, facendo così abbassare il relativo valor medio. Questo implica che, in generale, più basso è tale grandezza, più ampio sarà l'intervallo temporale di produzione di energia. Questo vale in particolare per i mesi che vanno da aprile a settembre ed in parte per marzo ed ottobre, in cui si ha un irraggiamento solare maggiore rispetto ai mesi invernali, seppur non così alto come nei mesi primaverili ed estivi.

Per il calcolo in euro della stima si è fatto riferimento al prezzo di 1kWh termico pari a 0.10€ A tal proposito, si faccia riferimento a GDD Energy - Calcolo del prezzo di un kwh termico domestico.

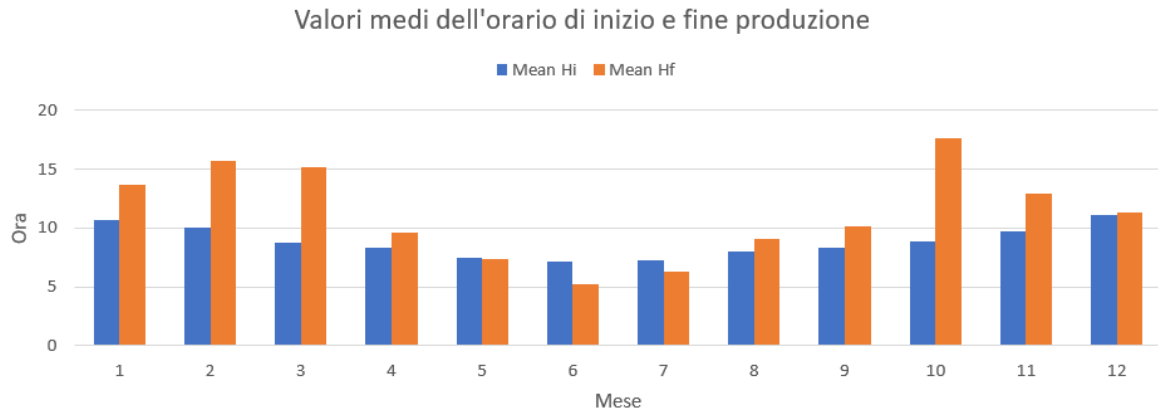


Figura 80: Ora media di inizio e fine produzione utilizzando il controllore base.

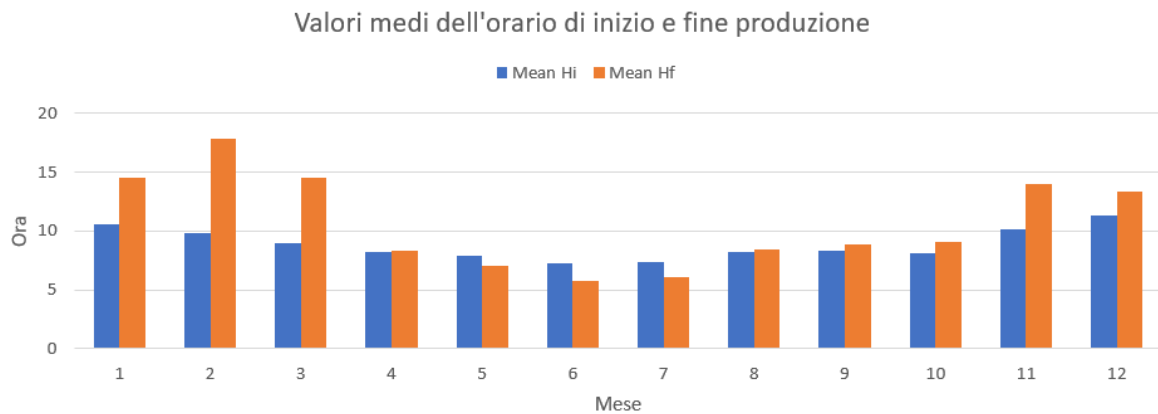


Figura 81: Ora media di inizio e fine produzione utilizzando il controllore avanzato.

Occorre comunque considerare che si tratta di un valor medio e, seppur dando delle informazioni, è opportuno tenere presente che il nuovo controllore garantisce produzioni di energia in giorni in cui il controllore base non assicurerebbe altrettanto se utilizzato. Questo può portare ad intervalli di produzione apparentemente più brevi per il controllore avanzato. Un esempio di quanto appena detto è riportato in Figura 82.

Production recap - old controller						Production recap - new controller					
Month	Day	Initial hour	Final hour	Value [kWh]		Month	Day	Initial hour	Final hour	Value [kWh]	
5	2	10	11	9,1937900000...		5	2	10	11	9,7878899999...	
5	3	7,598888889	16	66,045619999...		5	3	7,465555556	16	65,297839999...	
5	5	7,532222222	1	307,018310000...		5	4	7,465555556	2	230,2075	
5	6	7,465555556	1	307,558059999...		5	5	7,465555556	2	336,057910000...	
5	7	7,465555556	19,93222222	454,35254		5	6	7,398888889	2	336,407899999...	
5	8	8	17,99888889	95,823000000...		5	7	8	20	255,295840000...	
5	10	8	18,13222222	161,07152		5	8	8	17,73222222	100,936999999...	
5	12	7	1	301,583470000...		5	10	8	19	168,64719	
5	13	7	1	313,195329999...		5	11	8	2	219,931349999...	
5	14	7	1	300,853749999...		5	12	7	2	313,32646	
5	15	7	1	314,354850000...		5	13	8	2	324,787940000...	
5	16	7,198888889	1	303,458659999...		5	14	7	2	345,441790000...	
5	17	7,198888889	1	307,416930000...		5	15	7,865555556	2	326,754959999...	
5	18	7,555555556	0,598888889	202,844788889		5	16	7,132222222	2	226,848580000...	

Figura 82: Esempio di differenza di produzione con l'uso del nuovo controllore (a destra), il quale assicura la generazione di energia in giorni in cui il sistema non ne avrebbe prodotta con l'uso del controllore base (sinistra).

Per quanto riguarda il controllore avanzato applicato al profilo dei consumi di un edificio scolastico, invece, non si cerca di aumentare la produzione energetica in valore assoluto, bensì si vuole fare in modo che si abbia produzione nella fascia oraria di operatività di una scuola. Un esempio di questo è mostrato in Figura 83, in cui si può vedere come non venga utilizzato il TES al di fuori dell'orario scolastico, in modo da non scaricarlo inutilmente (ovvero non si manda il sistema in OM_5 o in OM_6).

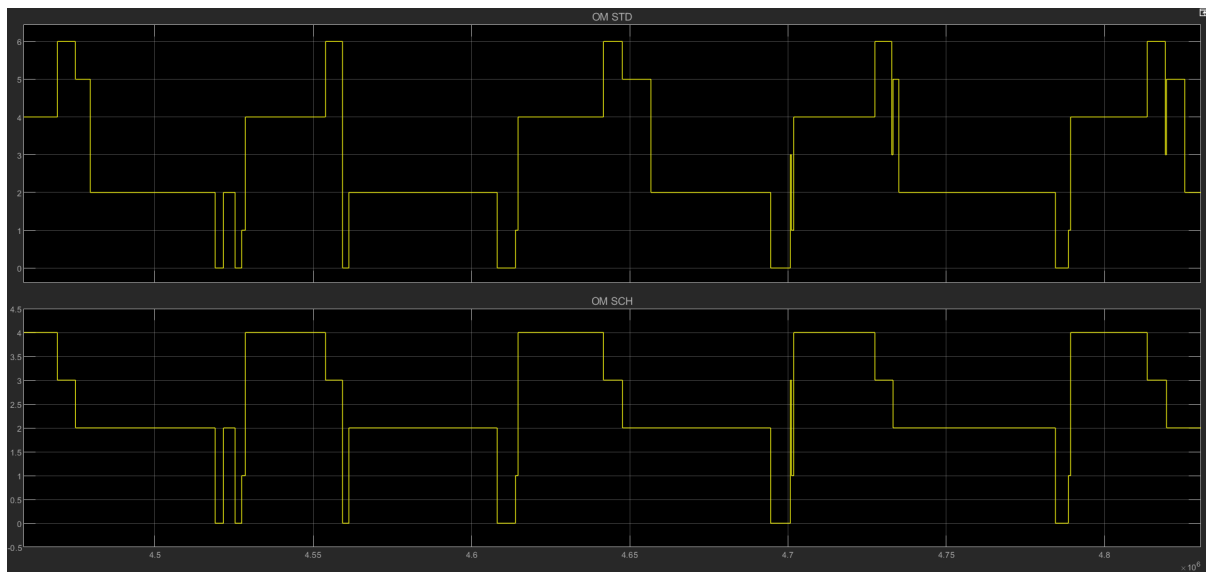


Figura 83: Differenza di *operative mode* calcolate con il controllore avanzato per consumi relativi a residenza private (grafico in alto) e per l'edificio scolastico (in basso).

6.2 Controllore per la massimizzazione della produzione

Complessivamente, i risultati ottenuti utilizzando questo controllore sono riportati in Tabella 18, in cui sono presenti i valori di energia termica prodotta in relazione con quelli ottenuti con il controllore base e quello avanzato per l'inseguimento termico.

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Controllore base	1.115E3	3.050E3	4.957E3	5.978E3	7.080E3	7.850E3
Controllore avanzato	1.679E3	3.378E3	5.413E3	6.650E3	8.010E3	8.790E3
Variazione assoluta	0.564E3	0.328E3	0.456E3	0.672E3	0.930E3	0.940E3
Variazione percentuale	50.58%	10.75%	9.20%	11.24%	13.14%	11.97%
Mese	Lug	Ago	Set	Ott	Nov	Dic
Controllore base	8.530E3	7.510E3	6.290E3	3.860E3	1.700E3	0.900E3
Controllore avanzato	9.6390E3	8.340E3	6.860E3	4.200E3	2.110E3	1.350E3
Variazione assoluta	1.100E3	0.830E3	0.570E3	0.340E3	0.410E3	0.450E3
Variazione percentuale	12.90%	11.05%	9.06%	8.81%	24.12%	50.00%

Tabella 18: Energia termica prodotta (in kWh) con il nuovo controllore in una simulazione di un anno confrontata con quella ottenuta con il controllore base.

In Tabella 19 viene riportata una valutazione in termini economici dei risparmi che si avrebbero adottando il controllore per la massimizzazione dell'energia prodotta per pilotare l'*operative mode* del sistema rispetto ciò che si avrebbe utilizzando il controllore base.

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Differenza di produzione	0.564E3	0.328E3	0.456E3	0.672E3	0.930E3	0.940E3
Valutazione economica	56€	33€	46€	67€	93€	94€
Mese	Lug	Ago	Set	Ott	Nov	Dic
Differenza di produzione	1.100E3	0.830E3	0.570E3	0.340E3	0.410E3	0.450E3
Valutazione economica	110€	83€	57€	34€	41€	45€

Tabella 19: Valutazione in euro dell'energia termica prodotta utilizzando il controllore base e quello per la massimizzazione dell'energia prodotta.

In Tabella 20, invece, viene riportato un confronto analogo, ma riferito alla produzione ottenuta con il controllore per la massimizzazione dell'energia prodotta e quello avanzato.

Dalle Tabelle sopra riportate, si può vedere come il controllore per la massimizzazione della produzione assicuri una maggiore energia termica prodotta rispetto sia il controllore base che quello avanzato (sebbene in termini più contenuti, soprattutto nei mesi estivi).

Per il calcolo in euro della stima si è fatto riferimento al prezzo di 1kWh termico pari a 0.10€. A tal proposito, si faccia riferimento a GDD Energy - Calcolo del prezzo di un kWh termico domestico.

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Differenza di produzione	0.535E3	0.234E3	0.248E3	0.320E3	0.420E3	0.430E3
Valutazione economica	54€	23€	25€	32€	42€	43€
Mese	Lug	Ago	Set	Ott	Nov	Dic
Differenza di produzione	0.430E3	0.310E3	0.270E3	0.220E3	0.370E3	0.430E3
Valutazione economica	43€	31€	27€	22€	37€	43€

Tabella 20: Valutazione in euro dell'energia termica prodotta utilizzando il controllore avanzato e quello per la massimizzazione dell'energia prodotta.

Per quanto riguarda i mesi estivi, invece, si rimanda a quanto riportato in Tabella 21, la quale contiene i dati relativi all'ora di inizio e fine produzione media mensile. Per una rappresentazione grafica del contenuto della Tabella, invece, si rimanda a quanto riportato in Figura 84 e 85 (relativamente al controllore base, si veda anche Figura 80 nella sottosezione 6.1 e quella per la massimizzazione della produzione, rispettivamente).

Mese	Gen	Feb	Mar	Apr	Mag	Giu
Ora inizio ctrl base	10.72	10.04	8.76	8.37	7.53	7.13
Ora inizio ctrl max prod.	10.09	9.59	8.51	7.72	7.12	6.74
Ora fine ctrl base	13.73	15.76	15.22	9.65	7.36	5.22
Ora fine ctrl max prod.	15.62	17.50	15.91	8.94	8.02	6.17
Mese	Lug	Ago	Set	Ott	Nov	Dic
Ora inizio ctrl base	7.37	8.20	8.37	8.11	10.18	11.35
Ora inizio ctrl max prod.	6.76	7.59	8.14	8.57	9.18	10.49
Ora fine ctrl base	6.29	9.08	10.18	17.69	12.94	11.37
Ora fine ctrl max prod.	7.05	7.88	9.10	16.67	15.37	14.98

Tabella 21: Fascia oraria giornaliera di produzione di energia (con l'ora che viene rappresentata in base 10 e non sessagesimale).

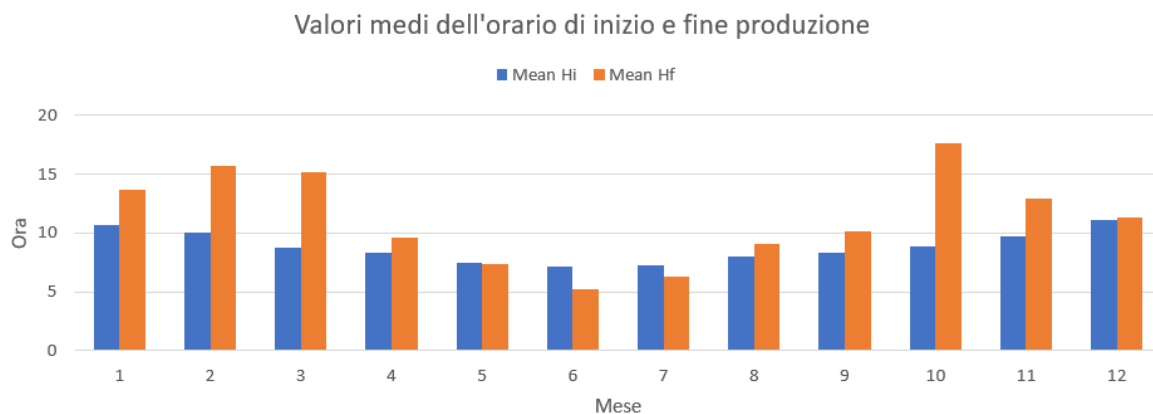


Figura 84: Ora media di inizio e fine produzione utilizzando il controllore base (si faccia riferimento alla Figura 80 nella sottosezione 6.1).

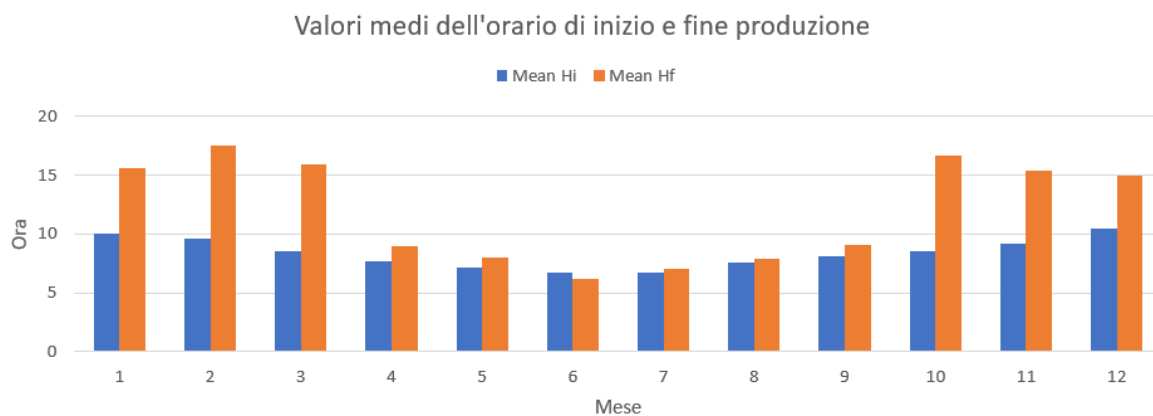


Figura 85: Ora media di inizio e fine produzione utilizzando il controllore per la massimizzazione della produzione di energia.

Per un'analisi dei dati ora riportati, occorre fare le stesse considerazioni riportate nella sottosezione 6.1. Quindi è necessario tener conto del fatto che i grafici di Figura 84 e 85 fanno riferimento a valori medi e che il controllore per la massimizzazione dell'energia prodotta offre range di produzione diversi in giorni diversi rispetto quello base. Questo porta ad inficiare il valor medio in questione.

7 Applicazione per l'analisi della produzione giornaliera di energia

L'applicazione che verrà qui descritta è stata implementata con lo scopo di semplificare la visualizzazione e quindi l'analisi dei dati ottenuti a seguito di una simulazione in Simulink.

I risultati ottenuti da una simulazione sono salvati in un file *csv* che conterrà i valori (in *kWh*) della produzione di energia termica in una giornata, in aggiunta con l'inizio e la fine della produzione stessa. Il file così creato in MATLAB verrà quindi caricato dall'applicazione che, a seguito di alcune elaborazioni dello stesso (ad esempio per rimuovere eventuali dati multipli relativi alla stessa produzione) procederà ad una visualizzazione a seguito della scelta operata dall'utente.

Nello specifico, le possibili operazioni implementate e messe a disposizione dell'utente sono:

- **Caricamento dei dati** relativi alla produzione del controllore avanzato, per la massimizzazione della produzione e base;
- **Visualizzazione dei dati** precedentemente caricati in forma tabellare;
- **Visualizzazione delle differenze di produzione** tra il controllore avanzato o quello per la massimizzazione della produzione rispetto quello base.

Per quanto riguarda l'interfaccia utente realizzata, si rimanda a quanto riportato in Figura 86, 87 e 88, in cui vengono mostrati i *mockup* delle varie finestre grafiche, le quali vengono ricalcate da quello effettivamente implementate. In particolare, in Figura 86 viene riportato il *mockup* della finestra grafica principale, ovvero quella che consente all'utente di caricare i dati di cui ha interesse. Caricati i file, è possibile selezionare quali di questi visualizzare sotto forma di grafico (eventualmente anche delle differenze di produzione tra controllori) nella finestra grafica il cui *mockup* è riportato in Figura 87, oppure in forma tabellare, come si può vedere dal *mockup* della finestra grafica in Figura 88.

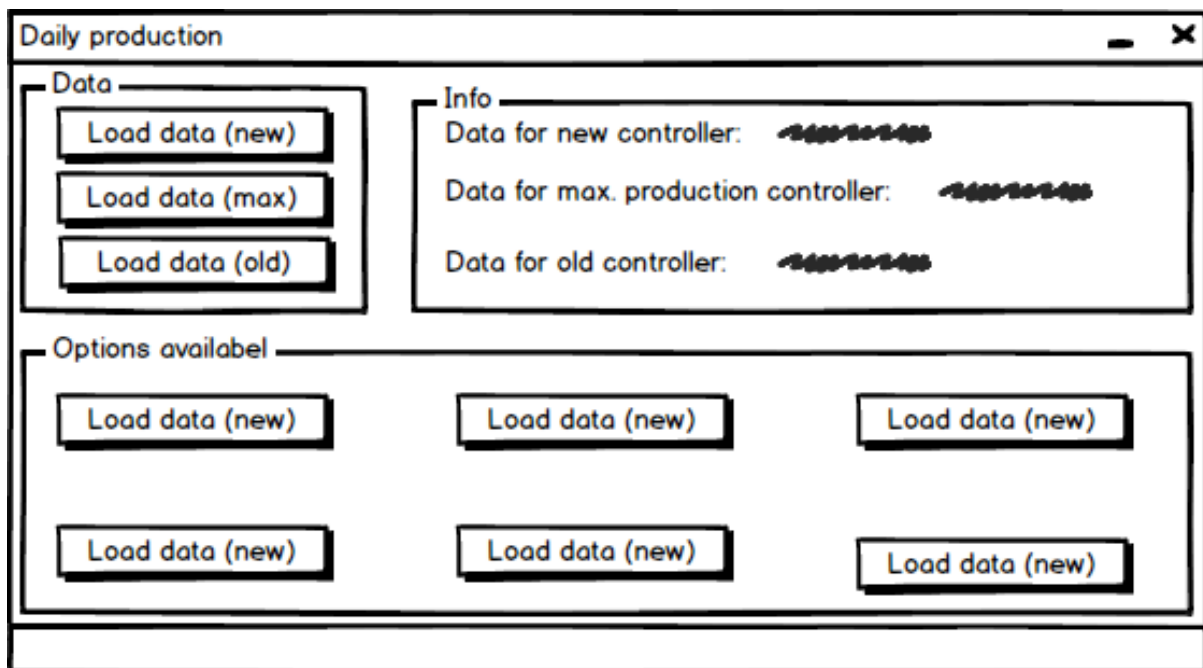


Figura 86: *Mockup* della finestra grafica principale.

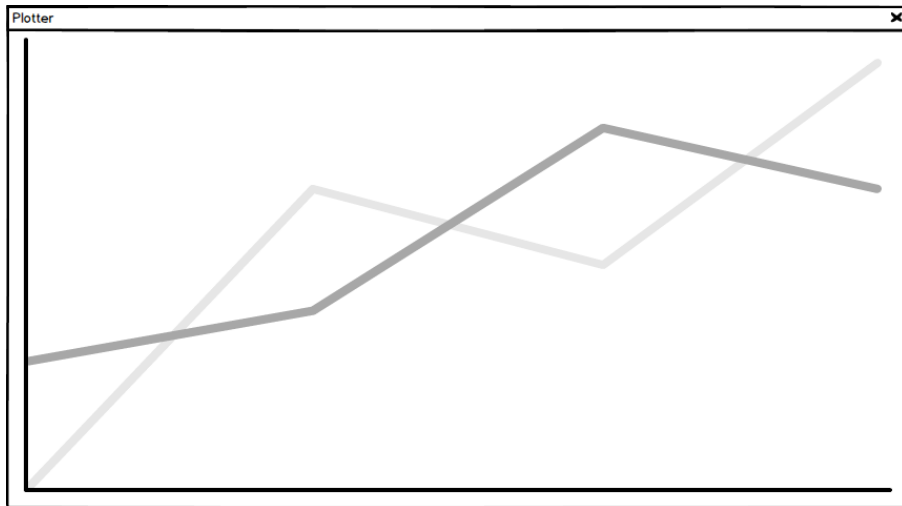


Figura 87: *Mockup* della finestra grafica per la visualizzazione dei grafici.

The image shows a window titled "Production recap" with a close button in the top right corner. It contains a table with five columns and multiple rows. The first three rows are filled with data, while the rest are empty. A vertical scrollbar is on the right side of the table.

Month	Day	Intial hour	Final hour	Value kWh
██████	███	██████	██████	██████
██████	███	██████	██████	██████
██████	███	██████	██████	██████

Figura 88: *Mockup* della finestra grafica per la visualizzazione tabellare dei dati.

7.1 Descrizione dell'architettura

In questa sottosezione verrà descritta l'architettura realizzata per l'implementazione dell'applicazione. Per fare questo si farà ricorso al diagramma delle classi riportato in Figura 89.

Nello specifico, si descriverà come i moduli software, ovvero le classi, sono state organizzate e come vengono utilizzate all'interno del programma. Per una descrizione più dettagliata delle funzionalità di ogni singola classe si rimanda invece alla sottosezione 7.2.

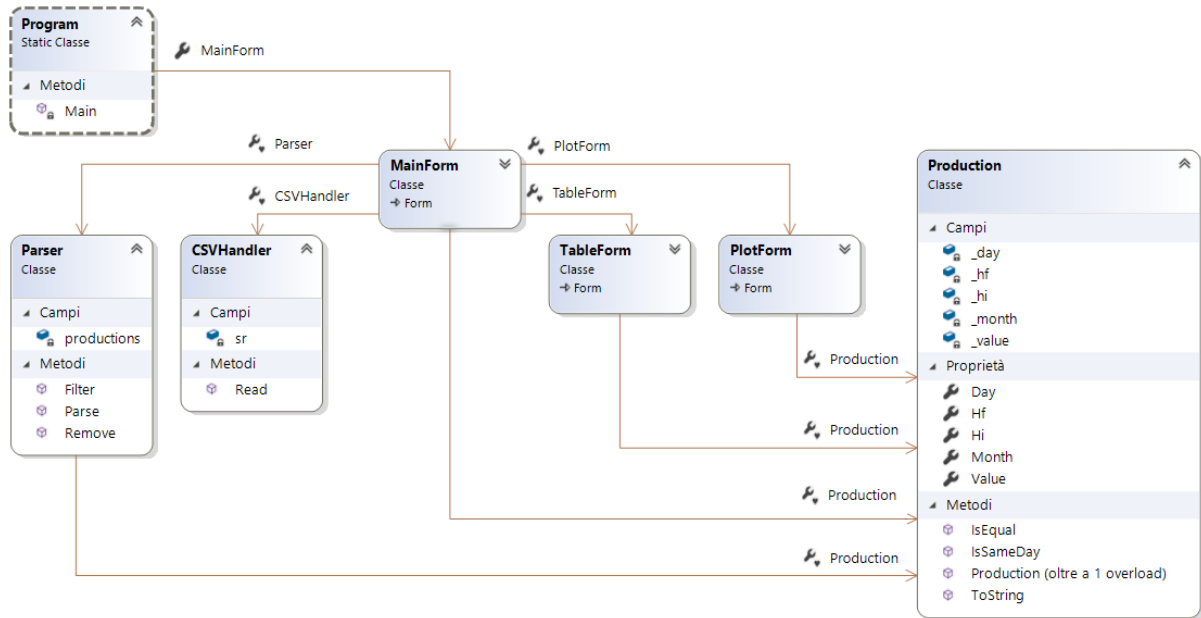


Figura 89: Diagramma UML delle classi dell'applicazione.

Classe *Production* La classe alla base dell'applicazione è *Production*. Questa classe infatti modella la produzione giornaliera in modo conforme alle necessità di visualizzazione ed analisi dei dati. Tra i suoi attributi ci saranno quelli relativi al tempo in cui si ha avuto la produzione, oltre che il valore di energia termica prodotta (in *kWh*).

Classi *CSVHandler* e *Parser* La classe *CSVHandler* ha il compito di leggere e caricare in memoria i dati della produzione di energia termica ottenuti al termine di una simulazione e salvati in un file separato da virgole. La classe *Parser*, invece, ha il compito di analizzare i dati caricati dal file *csv* ed estrarne le *feature*, ovvero ottenere da una riga del file i corrispettivi valori in termini di mese, giorno, ora di inizio e fine della produzione e valore della produzione stessa, creando quindi una lista di oggetti di tipo *Production*.

Classe *MainForm* Implementa la finestra grafica il cui *mockup* è riportato in Figura 86. Questa finestra grafica consente all'utente di scegliere ed eseguire una tra le opzioni disponibili.

Classi *PlotForm* e *TableForm* Implementano, rispettivamente, le finestre grafiche per la visualizzazione dei dati i cui *mockup* sono riportati in Figura 87 e 88.

Il diagramma UML delle sequenze in Figura 90 mostra l'interazione di massima tra i vari moduli software e l'utente. Si può vedere come l'utente scelga una delle possibili operazioni tramite l'oggetto *MainForm* e questo, a seconda della scelta operata richiama i metodi e quindi gli oggetti adeguati per eseguire la richiesta dell'utente.

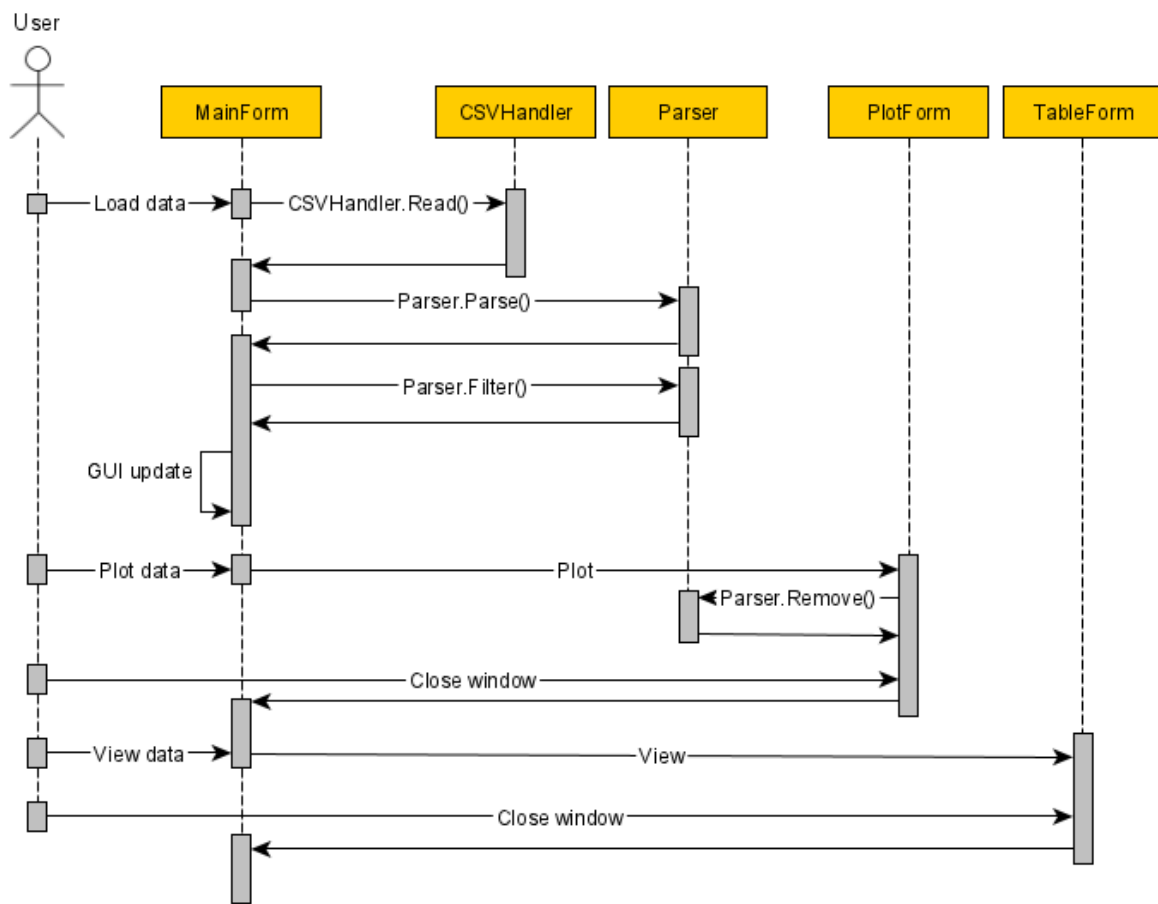


Figura 90: Diagramma UML delle sequenza.

7.2 Descrizione dei componenti software

In questa sottosezione verranno meglio descritte le varie classi che compongono l'applicazione, come queste sono strutturate (descrivendone quindi gli attributi) e quali compiti e funzionalità queste implementano grazie ai loro metodi.

Production Questa classe modella il dato relativo alla produzione dell'impianto di cogenerazione con l'aggiunta di informazioni temporali quali il mese, il giorno, l'ora di inizio e quella di fine della produzione di energia termica. Per fare questo, ognuna di queste grandezze sarà rappresentata mediante un appropriato attributo della classe, accessibile dall'esterno dai relativi metodi *getter* e *setter*. Il metodo costruttore presenta 2 *overload*, ovvero due metodi con lo stesso nome, ma con firma differente, ovvero due metodi con un diverso insieme di parametri. Il primo costruttore (senza parametri) avrà il compito, oltre che di creare l'oggetto in sé, di inizializzare gli attributi a valori di default. Il secondo costruttore (con parametri), dopo la creazione dell'oggetto, ne inizierà gli attributi con valori specificati. Si avrà poi il metodo *IsEqual* che stabilisce se 2 oggetti di tipo *Production* sono uguali ed il metodo *IsSameDay*, che determina se 2 oggetti fanno riferimento ad una produzione di energia avvenuta nello stesso giorno. Infine si avrà il metodo *ToString* che fornisce una descrizione testuale dell'oggetto.

CSVHandler Questa classe (statica) ha il compito di leggere il file *csv* contenente i dati ottenuti a seguito di una simulazione in Simulink e quindi di caricarli in memoria per successive elaborazioni. Questa funzionalità è implementata dall'unico metodo della classe, ovvero *Read*, che fornirà in uscita una lista di stringhe, rappresentanti ognuna una riga del file *csv* letto.

Parser Una volta letto il file grazie alla classe *CSVHandler*, questo verrà analizzato dalla classe (statica) *Parser* in modo da estrarne le *feature*, ovvero, presa una riga del file *csv*, grazie al metodo *Parse*, se ne estrarranno i relativi valori che verranno quindi salvati in una lista di oggetti di tipo *Production*. Il metodo *Filter*, invece, ha il compito di rimuovere o modificare eventuali dati letti dal file *csv* che sono errati o per renderli consistenti con il loro significato (ad esempio, per come è implementato il simulatore dell'impianto, si potrebbe avere una produzione di energia nel giorno 0 di un certo mese, quando in realtà questo dovrebbe essere il giorno 1). Infine, il metodo *Remove* ha il compito di accorpare eventuali produzioni di energia che avvengono nello stesso giorno in orari consecutivi tra loro.

PlotForm Questa classe ha il compito di visualizzare il grafico relativo alle produzioni giornaliere del singolo controllore (avanzato, massimizzazione della produzione e base) o la differenza di produzione tra questi (avanzato e base o massimizzazione della produzione e base). Il grafico mostrato sarà in funzione dell'opzione scelta dall'utente in *MainForm*.

TableForm Questa classe ha il compito di visualizzare, sotto forma di tabella, il riepilogo della produzione giornaliera dei vari controllori. I dati visualizzati dipenderanno dalla scelta dell'utente e da quale opzione questo abbia selezionato in *MainForm*.

MainForm Questa classe rappresenta l'interfaccia utente principale e consente di scegliere quale operazione eseguire. In particolare, a seconda della scelta fatta dall'utente, richiamerà uno dei metodi delle classi prima descritte, effettuerà controlli su quali dati siano stati caricati in memoria ed aggiornerà l'interfaccia grafica in accorso con le operazioni fatte dall'utente.

8 Conclusioni

In questa tesi sono stati implementati algoritmi di supervisione e controllo per un impianto di cogenerazione micro solare a seguito di uno studio fatto sul simulatore del sistema reale con l'obiettivo di migliorare la produzione. A tal fine, è stato prima sostituito e migliorato il controllore disponibile (base) con uno in logica fuzzy (avanzato) che permettesse di avere un incremento delle prestazioni, pur mantenendo l'obiettivo di inseguire il profilo di consumo termico.

In particolare, si sono tradotte le regole del controllore, impostate sostanzialmente sottoforma di *look-up table*, e quindi con una serie di "if then ... else ..." statici ed impostati a priori con la logica fuzzy, che ne rappresenta un'evoluzione più elegante, espandibile e manutenibile.

Inoltre, con l'aggiunta di nuove informazioni in ingresso al controllore (in particolare la grandezza $System_{info}$, oltre che T_{LFR}), si sono raggiunte prestazioni in termini di energia prodotta maggiori.

È stata inoltre implementata la possibilità di avere diverse modalità di controllo all'interno dello stesso controllore (avanzato), ovvero in funzione del carico collegato all'impianto modifica il controllore in modo da cercare di ottenere una produzione di energia il più compatibile possibile con il tipo di consumo. Nella tesi, come già esposto, si è scelto di implementare un controllore supponendo di avere come utenza un edificio scolastico o un'insieme di 4 residenze private (profilo per cui è stato effettivamente dimensionato il sistema in termini di capacità di produzione).

Successivamente è stato sviluppato un secondo controllore, con l'obiettivo di massimizzare la produzione di energia.

Infine è stato implementata un'applicazione per l'analisi della produzione giornaliera, da integrare con il simulatore dell'impianto di cogenerazione.

Il lavoro di tesi, quindi, si inserisce ed integra quanto precedentemente sviluppato all'interno del progetto *Innova MicroSolar* per quanto riguarda la parte di controllo e supervisione.

Tuttavia, è opportuno tenere presente che il progetto è ancora in fase di simulazione, per cui quanto implementato ed i risultati ottenuti dovranno essere rivisti in futuro per una possibile applicazione pratica.

Ciò non di meno, con le due varianti principali del controllore frutto del lavoro descritto in questo elaborato, si sono raggiunti gli obiettivi iniziali, portando ad un aumento delle prestazioni riassunto nel grafico riportato in Figura 79.

Infatti, con entrambi i nuovi controllori si riesce ad avere un aumento di energia prodotta rispetto il controllore base che, per il controllore avanzato corrisponde ad un aumento annuo medio nell'ordine del 4.75% circa. L'aumento raggiunge il 18.60% circa se si confrontano le prestazioni del controllore per la massimizzazione della produzione e quello base (lo stesso controllore per la massimizzazione dell'energia prodotta offre un aumento di circa il 13.40% rispetto quello avanzato).

Con il controllore per il profilo dei consumi scolastici, invece, si è cercato di rendere disponibile l'energia prodotta all'interno della fascia oraria di operatività, quando possibile.

Occorre comunque osservare che, per il modo in cui è stato progettato il controllore per la massimizzazione della produzione di energia, si ha che, soprattutto nei mesi invernali in cui l'irraggiamento solare è contenuto, il sistema (ed in particolare l'*organic Rankine cycle*) lavora con un rendimento inferiore a quello nominale. L'ORC, infatti, lavora ad energie ridotte (la potenza di attacco, infatti, passa da $13kWh$ a P_{σ} , che varia nell'intervallo compreso tra 5 e $13kWh$). Mentre questa condizione negativa viene messa in secondo piano dalla produzione maggiore che si ottiene nei mesi invernali (con picchi di oltre il 50% di produzione aggiuntiva) in cui la richiesta di energia termica è maggiore (per il riscaldamento domestico e per l'acqua sanitaria), in primavera ed estate ciò non è più valido. Tali richieste, infatti, sono minori in questi mesi ed i risparmi che si avrebbero in questo periodo sarebbero anche difficili da monetizzare.

Mentre una prima soluzione potrebbe essere quella di cambiare controllore, passando quindi a quello avanzato, una scelta ulteriore potrebbe essere quella di inserire nel sistema un sistema di assorbimento, ovvero un sistema che prende calore in ingresso, generando in uscita freddo, soddisfacendo così anche la domanda frigorifera estiva per monetizzare così anche il *surplus* di energia (che, per i sistemi cogenerativi come quello preso in esame in questo elaborato, sono elevati, così come è elevata la domanda frigorifera).

Una raccomandazione per ulteriori ricerche future potrebbe essere quella di integrare nuovi profili di consumo prendendo come esempio quello sviluppato per un edificio scolastico (ad esempio si potrebbe avere un profilo per un insieme di uffici o locali commerciali, che presentano fasce orarie differenti da quelle al momento considerate).

Una considerazione aggiuntiva, in relazione a quanto precedentemente accennato, è quella di sfruttare la tecnica con cui è stato implementato il controllore per la massimizzazione della produzione, andando a considerare anche il valore assoluto della derivata dei pesi e quindi implementare, ad esempio, un regolatore *PID* per pilotare il cambiamento della soglia P_o e cercare nel contempo di non degradare troppo le prestazioni del sistema e quindi non abbassarne troppo il rendimento con una soglia di attacco dell'ORC bassa, mantenendo comunque prestazioni che, seppur eventualmente inferiori a quelle ottenute in questo lavoro, siano dello stesso ordine di grandezza.

Inoltre, l'aspetto probabilmente più cruciale da sviluppare in futuri lavori è quello di introdurre l'informazione relativa alle previsioni meteo, ad esempio (nel modo più semplice ed intuitivo) aggiungendo un secondo controllore in cascata a quelli implementati che prenda in ingresso l'informazione predittiva e l'*operative mode* calcolata e la modifichi di conseguenza. Questo perché, ad esempio, se si è a conoscenza che il *linear Fresnel reflector* non sarà in grado di generare energia nei giorni a venire a causa di un basso irraggiamento solare, si potrebbe sfruttare il *thermal energy storage* in modo più intensivo, partendo dalla considerazione che non avrebbe senso avere un TES ad una temperatura più o meno elevata e destinata a scendere nei giorni successivi e non utilizzare l'energia calorifera associata (considerando che non sarà comunque possibile aumentare la temperatura in questione).

Infine, anche in relazione all'introduzione di nuovi possibili profili di consumo, potrebbe essere utile rivedere anche la rete neurale, andando a considerare *feature* differenti (in particolare togliendo la varianza tra queste, cosa già presa in considerazione in tesi, ma non effettivamente approfondita a livello di risultati ottenibili).

Con questo lavoro, dunque, è stato possibile implementare algoritmi atti ad ottimizzare la produzione di energia di un impianto di cogenerazione e la conseguente sostituzione del vecchio controllore con il nuovo, che consente anche di avere diversi profili di controllo in funzione dei consumi e dell'obiettivo di controllo.

9 Appendici

Nell'appendice A viene riportato il manuale utente del simulatore ed in particolare del sottosistema composto dai moduli software sviluppati nell'ambito della tesi, sezione 5e.

La sintassi del manuale sarà del tipo:

- Immagine del codice grafico di Simulink e descrizione delle funzionalità del blocco in questione, nel caso sia utilizzato solamente Simulink per sviluppare il codice;
- Numero di riga o blocco di codice e relativa descrizione, nel caso si sia utilizzata una funzione di MATLAB per lo sviluppo della funzionalità.

In Appendice B, invece, è riportato il manuale utente dell'applicazione sviluppata per avere un resoconto della produzione giornaliera (intesa come mese, giorno, ora di inizio e fine produzione di energia ed energia prodotta), sezione 7.

Il manuale, in questo caso, servirà a dare indicazioni su come utilizzare il software in questione, non andando nel dettaglio delle funzionalità delle singole righe o blocchi di codice.

A Manuale di uso e manutenzione del simulatore

La parte aggiuntiva del simulatore sviluppata in Simulink segue il seguente schema di colori per quanto riguarda i blocchi:

- **Rosso** per tutti i blocchi che contengono al loro interno un sottosistema (quindi, ad esempio, si avranno in rosso i blocchi contenenti il controllore, quello per la creazione del segnale per la stima delle performance del sistema, ecc.);
- **Verde** per tutti i blocchi che hanno il compito di eseguire un'operazione matematica (come ad esempio quelli per il calcolo della media o del valore quadratico medio online);
- **Blu** per tutti i blocchi per l'elaborazione dei segnali (ad esempio in blu si avrà il blocco per la modifica dell'*operative mode* in funzione della temperatura del TES);
- **Arancione** per tutti i blocchi per la visualizzazione di segnali o informazioni in generale (come quello per l'uscita della rete neurale per l'identificazione del segnale o per il conteggio delle commutazioni di fase).

A livello generale, la parte di simulatore implementata è riportata in Figura 91. Invece, in Figura 92 è mostrata la parte del *Model Browser* del Simulatore così come si presenta in Simulink con evidenziati i nuovi moduli implementati.

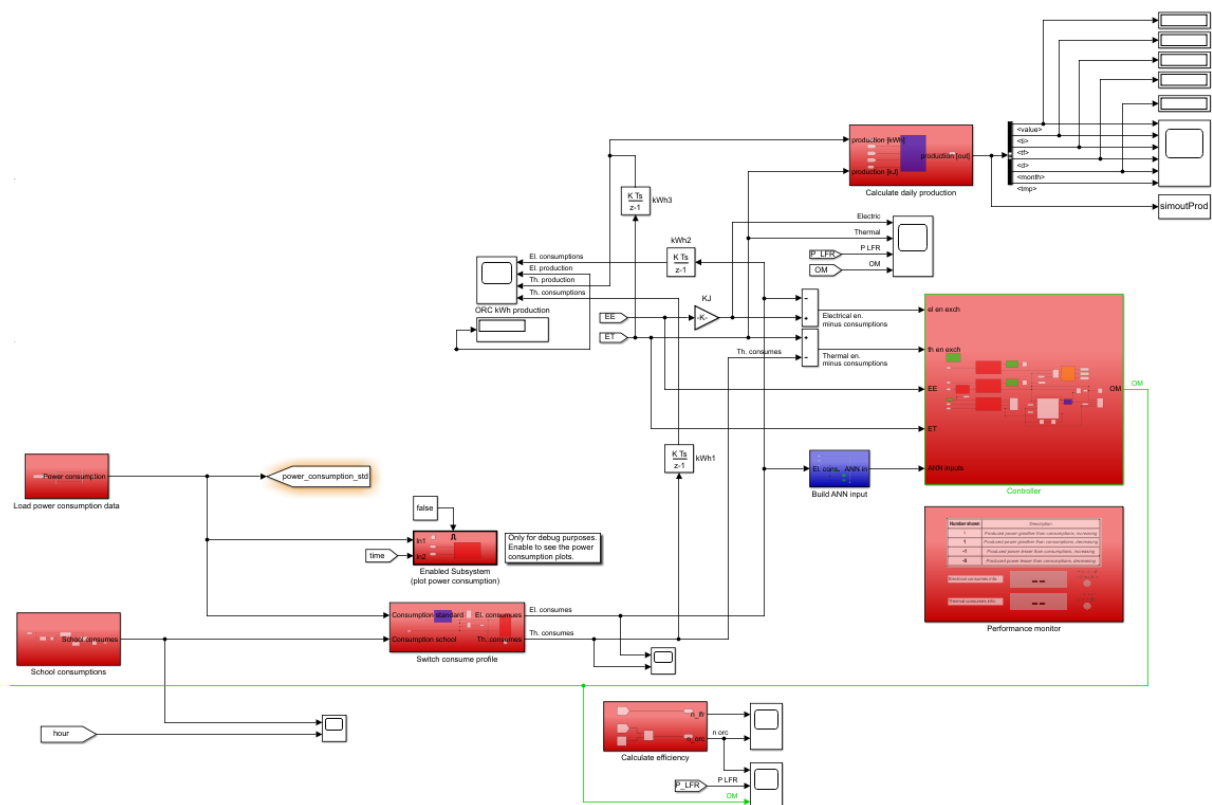


Figura 91: Blocchi di codice aggiuntivi implementati in questo lavoro.

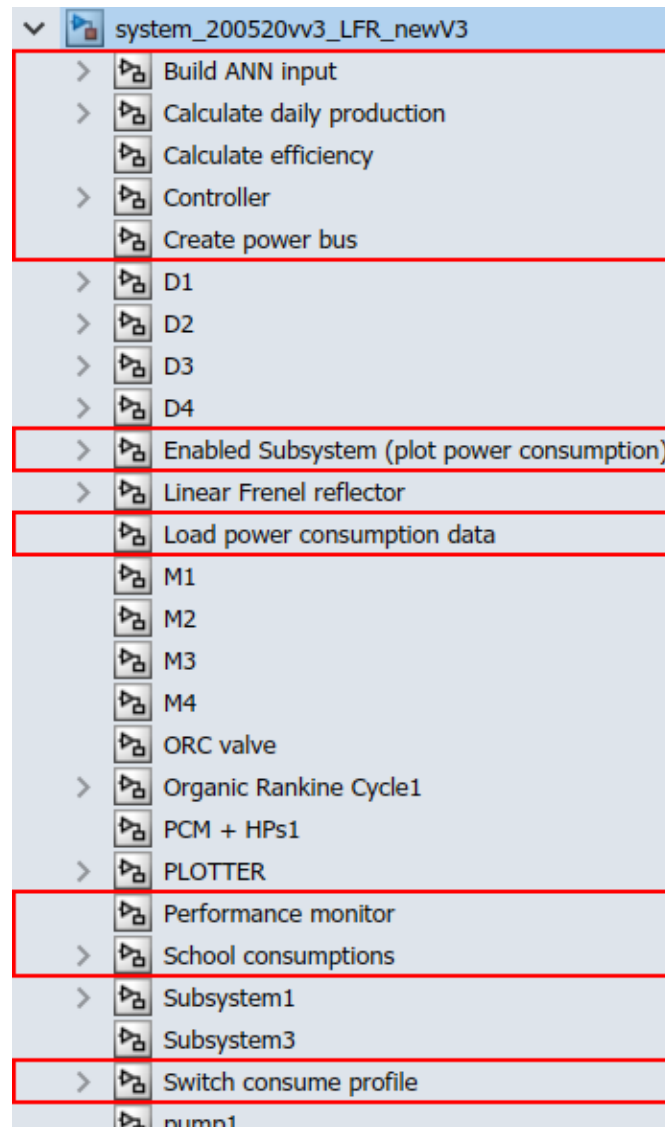


Figura 92: Porzione del *Model Browser* del simulatore all'interno di Simulink con i blocchi implementati evidenziati in rosso.

Di seguito viene riportato un elenco con tutti i blocchi evidenziati in Figura 92 e relativi sottoblocchi. In particolare, verrà data una descrizione della funzionalità che ognuno dei blocchi implementa seguendo l'ordine riportato sempre nella Figura 92 (partendo quindi dal blocco *Build ANN input* fino ad arrivare a *Switch consume profile*). Occorre infine osservare che eventuali blocchi e/o sottoblocchi utilizzati in più contesti (ad esempio quello per il calcolo della media online) verranno riportati una sola volta.

- **Build ANN input:** ha il compito di costruire l'input per la rete neurale in modo che questa sia poi in grado di riconoscere il profilo dei consumi in uso durante una simulazione. Complessivamente, si avrà in output un *bus* contenente i singoli segnali utilizzati come *feature* per il riconoscimento.
 - **Online mean.** Calcola la media online della grandezza in ingresso. Se $\bar{x}_i = \frac{1}{N} \sum_{i=1}^N x_i \rightarrow \bar{x}_{i+1} = \frac{1}{N+1}(N\bar{x}_i + x_{i+1})$;
 - **Online RMS.** Calcola il valore quadratico medio online della grandezza in ingresso. Se $RMS_i = \frac{1}{N} \sum_{i=1}^N x_i^2 \rightarrow RMS_{i+1} = \frac{1}{N+1}(NRMS_i + x_i^2)$;
 - **Online variance.** Calcola la varianza online della grandezza in ingresso. Se $\sigma_i^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}_i)^2 \rightarrow \sigma_{i+1}^2 = \frac{1}{N+1}[N\sigma_i^2 + (x_{i+1} - \bar{x}_{i+1})^2]$.

- **Calculate daily production:** restituisce in output la *struct* contenente le informazioni relativi alla produzione giornaliera. L'output del blocco viene poi salvato all'interno della variabile *simoutProd* del *workspace*. Tali dati saranno grezzi e necessiteranno di una eventuale ulteriore analisi (sezione 7).
 - **Analyze production.** Contiene l'implementazione della funzionalità.
 - * **Daily production analyzer.** Implementazione della funzionalità del blocco *Calculate daily production*.
 - **Edge detector.** Consente di determinare quando il segnale di ingresso presenta un fronte di salita o discesa. In questi casi il segnale in uscita da questo blocco sarà un impulso di ampiezza unitaria e 0 altrimenti.
 - **createProductionBus.** Funzione con il compito di creare, all'inizio della simulazione, un *bus* per rappresentare in modo coerente la *struct* per contenente le informazione di interesse. Per la modifica di questa parte, occorre andare nelle proprietà del blocco, quindi in *Callbacks* e infine in *InitFcn*.
 - **analyzeProduction.** Funzione che, prese le informazioni in input, fornisce una *struct* rappresentante la produzione di energia termica giornaliera del sistema.
- **Calculate efficiency.** Calcola il rendimento del *linear Fresnel reflector* e dell'*organic Rankine cycle* così come implementati nella versione del simulatore sviluppata per la *Simulink Student Challenge* del 2018. Sono stati implementati di nuovo per semplicità di utilizzo e visualizzazione.
- **Controller:** contiene l'implementazione del controllore per la gestione delle commutazioni delle varie *operative mode* oltre che quella della rete neurale.
 - **Neural network.** Contiene l'implementazione della rete neurale per il riconoscimento dei segnali implementata utilizzando lo strumento messo a disposizione da MATLAB per la costruzione di questo tipo di tecnologia.
 - **Thermal controller.** Contiene l'implementazione del controllore vero e proprio.
 - * **Thermal energy exchanged info.** Sottosistema per la generazione del segnale relativo alle performance del sistema.
 - * **Adaptive threshold.** Sistema dinamico per la generazione del segnale così come descritto nella sottosezione 5.2.3.
 - **Electrical subsystem.** Contiene un'implementazione analoga a quello per il controllore prima descritto, ma che prende in considerazione l'energia elettrica prodotto ed i consumi elettrici dell'utenza. Attualmente non utilizzato.
 - **Calculate weights.** Sottosistema che determina i pesi che si utilizzeranno nel caso di scelta del profilo per la massimizzazione dell'energia termica.
 - **maximizeProduction.** Funzione utilizzata per massimizzare la produzione di energia termica andando a modificare l'*operative mode* calcolata dal controllore avanzato.
 - **Check TES temperature.** Blocco con il compito di monitorare la temperatura del TES affinché questa non superi il valore massimo consentito (ovvero 280C).
 - * **removeSwitches.** Funzione utilizzata per eliminare eventuali commutazioni errate dell'*operative mode* durante la produzione di energia.
 - * **checkTemperature.** Funzione che imposta l' OM_{-1} nel caso in cui $T_{TES} \geq 280C$.
 - **Switch counter.** Blocco con il compito di visualizzare il numero di commutazione ottenute con un controllore durante una simulazione.
- **Create power bus.** Combina in un unico segnale varie potenze di rilevanza per il sistema ed informazioni aggiuntive (quali il rendimento dell'ORC, η_{ORC}).
- **Enabled Subsystem (plot power consumption).** Blocco utilizzato in fase di debug e normalmente non abilitato. Visualizza con un grafico il profilo dei consumi.
- **Performance monitor.** Interfaccia utente per la visualizzazione delle performance del sistema.

- **Load power consumption data.** Carica in memoria i dati (contenuti in un file Excel) dei consumi di 4 appartamenti sottoforma di vettore.
- **School consumptions.** Genera il profilo di consumo per un edificio scolastico. L'ampiezza del segnale dipende dall'orario all'interno della simulazione.
- **Switch consume profile.** In base alla selezione, fornisce in output il segnale relativo ai consumi termici ed elettrici dell'utenza collegata all'impianto.
 - **Array to element.** Prende un singolo elemento del vettore contenente i dati dei consumi di 4 residenze private e lo rende disponibile in output in base al tempo della simulazione (infatti, il periodo di campionamento dei dati relativi ai consumi è di 10 minuti).
 - **ANN train parameters.** Calcola e salva le *feature* utilizzate per l'addestramento della rete neurale in variabili nominate secondo la seguente regola: *simout<nome feature>* (ad esempio, quindi, per la variabile relativa alla *feature* "ora" si avrà *simoutHour*).

Infine, la scelta tra i vari obiettivi di controllo voluti e profilo dei consumi da tenere in considerazione può essere fatta come segue. Per definire l'obiettivo di controllo e quindi quale controllore tra quello per l'inseguimento termico e la massimizzazione della produzione è necessario utilizzare il *Manual Switch* contenuto nel blocco *Controller*. Se questo si trova nella posizione "1", si avrà l'OM ottenuta dal controllore per l'inseguimento termico, mentre se si troverà in posizione "2" si utilizzerà il controllore per la massimizzazione della posizione. Invece, per la scelta di quale profilo dei consumi prendere in considerazione, occorrerà eseguire una procedura analoga con il *Manual Switch* presente nel blocco *Switch consume profile*. Se questo si trova in posizione "1" si avranno i consumi standard per 4 abitazioni provate; se, invece, lo *switch* si trova in posizione "2", si avranno i consumi per un edificio scolastico.

B Manuale utente dell'applicazione per l'analisi della produzione giornaliera

Viene qui descritto come utilizzare il programma sviluppato in modo da poter permettere una visualizzazione e successivamente un'analisi più agevole dei dati ricavati da una simulazione. Per una descrizione accurata dell'architettura del software e dei moduli che lo compongono si rimanda invece alla sezione 7 e relativa sottosezione.

La finestra grafica che si apre all'avvio del programma (ovvero la finestra grafica principale che permette all'utente di scegliere di eseguire una delle opzioni messe a disposizione dall'applicazione) è mostrata in Figura 93. Da qui, per prima cosa, occorre caricare i dati che si vogliono visualizzare. Per fare questo è necessario premere il bottone "Load data (<tipo>)", in cui "new" caricherà i dati del controllore avanzato, "max" quelli del controllore per la massimizzazione della produzione ed "old" quelli del controllore base ("Data"). In funzione di quali dati sono stati caricati in memoria si aggiornerà di conseguenza anche la descrizione presente nel riquadro "Info".

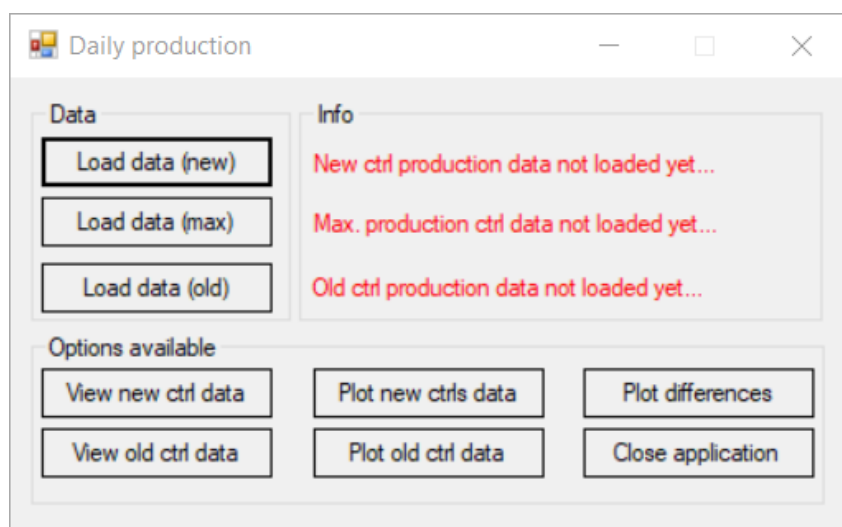


Figura 93: Finestra grafica principale del programma.

Una volta caricato il file, il relativo bottone viene disabilitato. Finita questa operazione, si avrà una schermata grafica del tipo di quella mostrata in Figura 94.

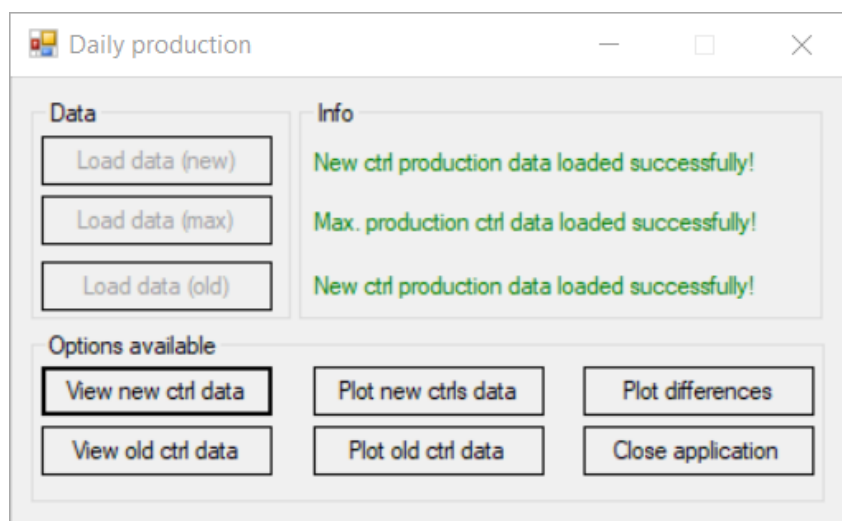


Figura 94: Finestra grafica principale del programma dopo il caricamento dei dati.

A questo punto è possibile scegliere una delle opzioni disponibili, ovvero la visualizzazione sotto forma di tabella oppure di grafico dei risultati contenuti nei file precedentemente caricati del controllore di cui si ha interesse. I bottoni con *label* del tipo "*<Comando> new ctrl data*" fanno riferimento al controllore avanzato e quello per la massimizzazione della produzione energetica (in questo caso, quindi, si apriranno 2 nuove finestre grafiche a seconda dell'opzione scelta), mentre quelli del tipo "*<Comando> old ctrl data*" fanno riferimento ai dati relativi al controllore base. In funzione di quanto selezionato dall'utente, si avrà una schermata del tipo di quella mostrata in Figura 95 per la visualizzazione tabellare dei dati, mentre per quella graficata si rimanda a quanto riportato in Figura 96.

Production recap - new controller

	Month	Day	Initial hour	Final hour	Value [kWh]
▶	1	1	14	15,19888889	12,83489341
	1	3	11	15	18,98910984
	1	4	10,06555556	15	47,49920057
	1	6	10,06555556	15,59888889	76,51928798
	1	7	10,19888889	15	62,7485249
	1	8	10,59888889	15,39888889	75,4298139
	1	11	11	14	28,4099972
	1	13	10	16	68,5184697
	1	14	10	13,99888889	50,8226687
	1	15	10	16	83,3265964
	1	16	10	15	71,4224216
	1	17	10	16	86,382215
	1	18	10	15	45,3461476
	1	21	11,29888889	0	17,262867

Figura 95: Finestra grafica per la visualizzazione tabellare dei dati.

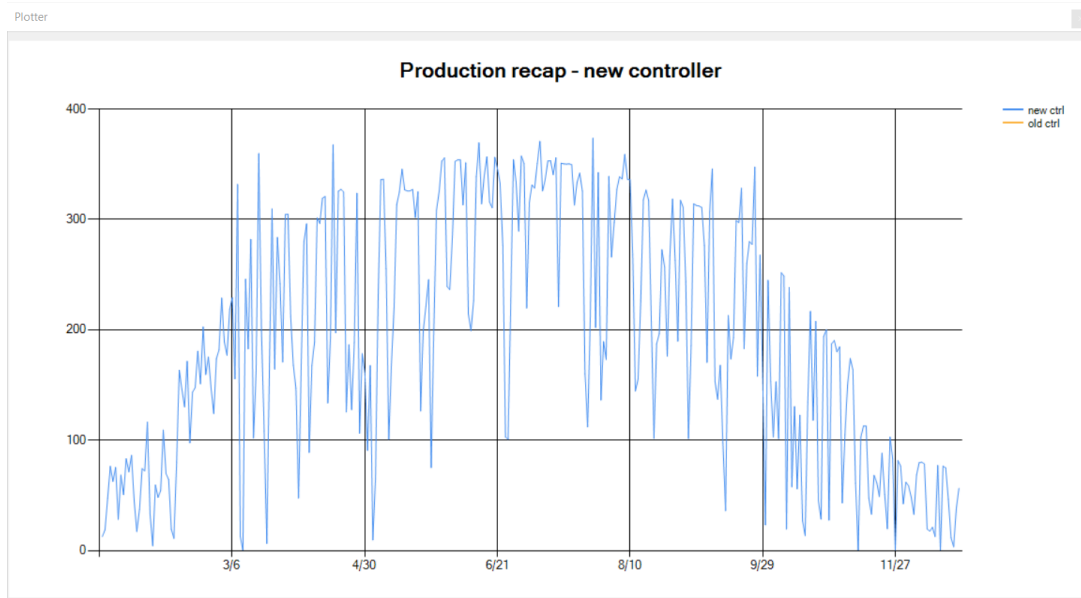


Figura 96: Finestra grafica per la visualizzazione sotto forma di grafico dei dati.

In aggiunta, in Figura 97 viene mostrata la finestra grafica ottenuta selezionando l'opzione "Plot differences", ovvero il grafico in cui vengono mostrate le differenze di produzione tra il controllore base ed il controllore per la massimizzazione della produzione di energia rispetto quello base (si avranno 2 finestre grafiche, una per ognuno dei controllori implementati).

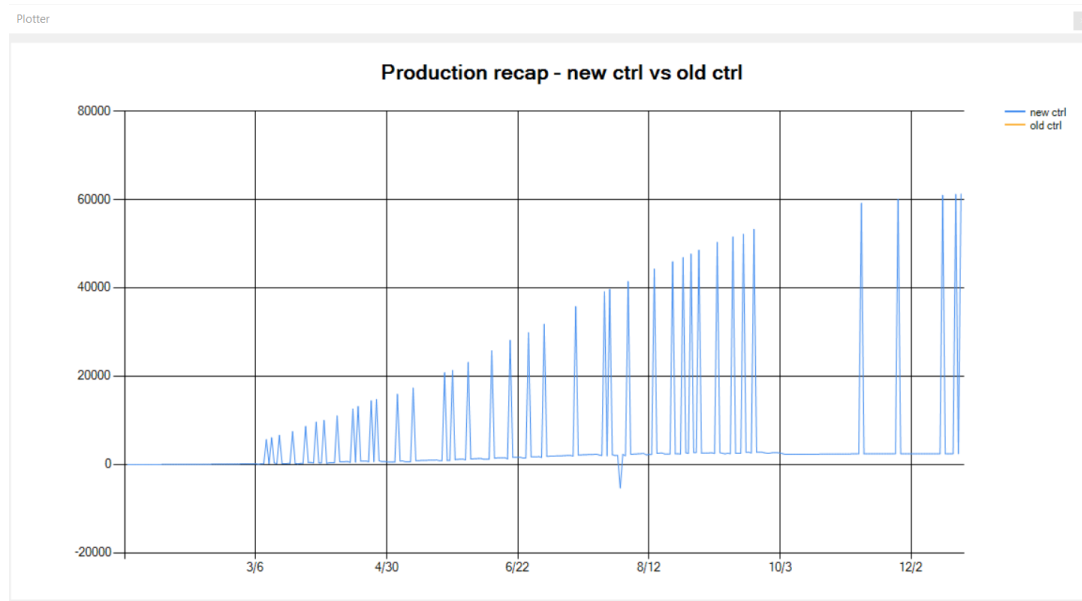


Figura 97: Finestra grafica per la visualizzazione sotto forma di grafico della differenza di produzione tra i controllori.

Il bottone "Close application" serve infine per chiudere l'applicazione e liberare quindi le relative risorse.

A Bibliografia e sitografia

Riferimenti bibliografici

- [1] A. Arteconi, L. Del Zotto, R. Tascioni, L. Cioccolanti, "Modelling system integration of a micro solar Organic Rankine Cycle plant into a residential building", *Applied Energy* (vol. 251), Elsevier, 2019
- [2] G. Bonvicini, "Applicazione di cicli ORC per il recupero di potenza in impianti turbogas di piccola taglia", *La Termotecnica*, 2014
- [3] A. Sevault, H. Kauko, M. Bugge, K. Banasiak, "Phase change materials for thermal energy storage in low- and high-temperature applications: a state-of-the-art", *SINTEF Energy Research*, 2017
- [4] L. Jetto, *Materiale del corso di Metodi e Tecniche per l'Automazione*, Università Politecnica delle Marche
- [5] L. A. Zadeh, "Fuzzy Sets", in *Information and Control* (vol. 83, n. 3), 1965
- [6] D. Dubois, H. Prade, "Possibility Theory An Approach to Computerized Processing of Uncertainty", Springer US, 1988
- [7] H.J. Zimmermann, "Fuzzy Sets, Decision Making and Expert Systems", Springer Netherlands, 1987
- [8] B. Arfi, "Linguistic Fuzzy Logic Methods in Social Sciences", Springer, 2010
- [9] E.H. Mamdani, S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *International journal of non-machine studies*, 1975
- [10] P. Sernani, *Materiale del corso di "Intelligenza Artificiale"*, Università Politecnica delle Marche
- [11] A. Freddi, *Materiale del corso di "Misure e Strumentazione per l'Automazione"*, Università Politecnica delle Marche
- [12] C. Gallo, Centro di Ricerca Interpartimentale Bioagromed, "Reti Neurali Artificiali: Teoria ed Applicazioni", Università degli Studi di Foggia, 2007
- [13] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review* (vol. 65, n. 6), 1958
- [14] W.S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics* 5, 1943
- [15] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors", *Nature* 323, 1986
- [16] J. P. Hespanha, D. Liberzon, A. S. Morse, "Overcoming the limitations of adaptive control by means of logic-based switching", *System & Control Letters* (vol. 49, n. 1), 2003

Riferimenti sitografici

- [S1] GSE - Guida alla cogenerazione ad alto rendimento.
https://www.gse.it/documenti_site/Documenti%20GSE/Servizi%20per%20te/COGENERAZIONE%20AD%20ALTO%20RENDIMENTO/Guide/GUIDA_CAR_AGGIORNAMENTO.pdf
- [S2] Gazzetta Ufficiale - Decreto legislativo 8 febbraio 2007, n. 20.
<https://www.gazzettaufficiale.it/eli/id/2007/03/06/007G0032/sg>
- [S3] Wikipedia - Cogenerazione.
[https://it.wikipedia.org/wiki/Cogenerazione#Piccola_cogenerazione_\(e_microcogenerazione\)](https://it.wikipedia.org/wiki/Cogenerazione#Piccola_cogenerazione_(e_microcogenerazione))
- [S4] C. Moler - The Origins of MATLAB.
<https://it.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html>
- [S5] Innova-MicroSolar - Innovative Micro Solar Heat and Power System for Domestic and Small Business Residential Buildings.
<http://innova-microsolar.eu/>
- [S6] European Commission - Horizon 2020.
<https://ec.europa.eu/programmes/horizon2020/en>
- [S7] Wikipedia - Parabolic trough.
https://en.wikipedia.org/wiki/Parabolic_trough
- [S8] Smithsonian Magazine - Science Makes a Better Lighthouse Lens.
<https://www.smithsonianmag.com/science-nature/science-makes-a-better-lighthouse-lens-170677431/>
- [S9] Enciclopædia Britannica - Fresnel lens.
<https://www.britannica.com/technology/Fresnel-lens>
- [S10] Wikipedia - Organic Rankine cycle.
https://en.wikipedia.org/wiki/Organic_Rankine_cycle
- [S11] Università di Pisa - Energia Applicata, Cicli ORC.
https://elearn.ing.unipi.it/pluginfile.php/157412/mod_resource/content/1/ORC.pdf
- [S12] Mathworks - Fuzzy inference process.
<https://it.mathworks.com/help/fuzzy/fuzzy-inference-process.html>
- [S13] Mathworks - Mamdani and Sugeno Fuzzy Inference Systems.
<https://it.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html>
- [S14] tutorialspoint - Fuzzy Logic.
https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_inference_system.htm
- [S15] Mathworks - Fuzzy Inference Process.
<https://it.mathworks.com/help/fuzzy/fuzzy-inference-process.html>
- [S16] Mathworks - Reti neurali.
<https://it.mathworks.com/discovery/neural-network.html>
- [S17] Il Sole 24 Ore - Machine learning, deep learning e reti neurali. Ecco di cosa parliamo.
<https://www.ilsole24ore.com/art/machine-learning-deep-learning-e-reti-neurali-ecco-cosa-parliamo--AEaToEBH>
- [S18] A.P. Bishop - The Role of Computer Networks in Aerospace Engineering.
https://www.ideals.illinois.edu/bitstream/handle/2142/7926/librarytrendsv42i4j_opt.pdf?sequence=4
- [S19] Università Roma Tre - Intelligenza artificiale.
http://www.dia.uniroma3.it/~ia/docs/old/nn_2.pdf

- [S20] Politecnico di Torino - Apprendimento non supervisionato.
https://areeweb.polito.it/didattica/gcia/tesine/Tesina_Kohonen/APPRENDIMENTO_NON_SUPERVISIONATO.pdf
- [S21] M. Nielsen - Neural Networks and Deep Learning.
<http://neuralnetworksanddeeplearning.com/chap5.html>
- [S22] Wikipedia - *Overfitting*.
<https://en.wikipedia.org/wiki/Overfitting>
- [S23] Wikipedia - Fattore di capacità.
https://it.wikipedia.org/wiki/Fattore_di_capacit%C3%A0
- [S24] J. P. Hespanha - Tutorial on Supervisory Control.
<https://web.ece.ucsb.edu/~hespanha/published/tr-tutorial-cdc01.pdf>
- [S25] J. P. Hespanha - Hybrid Control and Switched Systems, Lecture #15 Supervisory Control (University of California at Santa Barbara).
<https://web.ece.ucsb.edu/~hespanha/ece229/Lectures/Supervisory1.pdf>

Riferimenti per le Figure

- [F1] Innova MicroSolar - Innovative Micro Solar Heat and Power System for Domestic and Small Business Residential Buildings.
<http://innova-microsolar.eu>
- [F2] L. Cioccolanti, R. Tascioni, A. Arteconi, "Mathematical modelling of operation modes and performance evaluation of an innovative small-scale concentrated solar organic Rankine cycle plant", Applied Energy (vol. 221), Elsevier, 2018
- [F3] Enciclopædia Britannica - Fresnel lens.
<https://www.britannica.com/technology/Fresnel-lens>
- [F4] Wikipedia - Parabolic trough at Harper Lake in California.
https://upload.wikimedia.org/wikipedia/commons/a/a8/Parabolic_trough_at_Harper_Lake_in_California.jpg
- [F5] Wikipedia - ORC power plant.
https://en.wikipedia.org/wiki/File:ORC_power_plant.JPG
- [F6] Wikipedia - Organic Rankine Cycle with Regenerator.
https://en.wikipedia.org/wiki/File:Organic_Rankine_Cycle_with_Regenerator.png
- [F7] Wikipedia - Ideal and real organic rankine cycle.
https://en.wikipedia.org/wiki/File:Ideal_and_real_organic_rankine_cycle.jpg
- [F8] A. Sevault, H. Kauko, M. Bugge, K. Banasiak, "Phase change materials for thermal energy storage in low- and high-temperature applications: a state-of-the-art", SINTEF Projects, 2017
- [F10] Mathworks - Mamdani and Sugeno Fuzzy Inference Systems.
<https://it.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html>
- [F11] Wikipedia - Simple XOR-function solved with a 3-Layer-MLP.
https://upload.wikimedia.org/wikipedia/commons/b/b2/Perceptron_XOR.jpg
- [F12] M. Nielsen - Neural Networks and Deep Learning, chapter 5.
<http://neuralnetworksanddeeplearning.com/chap5.html>
- [F13] W. Cui, H. Wang, "A New Anomaly Detection System for School Electricity Consumption Data", Information (vol. 8), 2017