



Ente per le Nuove tecnologie,  
l'Energia e l'Ambiente



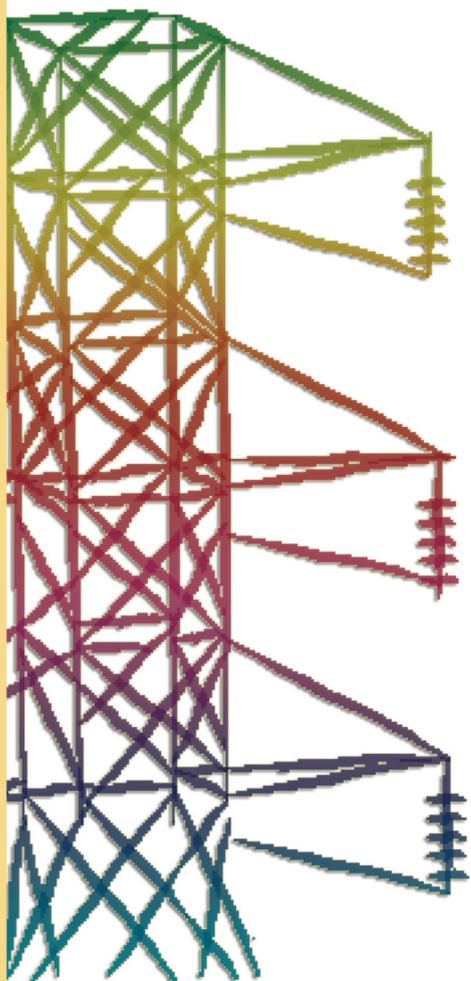
*Ministero dello Sviluppo Economico*

## **RICERCA SISTEMA ELETTRICO**

---

### **Sviluppo di metodi di controllo per il sistema edificio-impianto**

**Matteo De Felice, Stefano Panzieri**





Ente per le Nuove tecnologie,  
l'Energia e l'Ambiente



*Ministero dello Sviluppo Economico*

RICERCA SISTEMA ELETTRICO

Sviluppo di metodi di controllo per il sistema edificio-impianto

*Matteo De Felice, Stefano Panzieri*



## SVILUPPO DI METODI DI CONTROLLO PER IL SISTEMA EDIFICIO-IMPIANTO

Matteo De Felice (ENEA)

Stefano Panzieri (Dipartimento di Informatica e Automazione dell'Università Roma Tre)

Aprile 2009

Report Ricerca Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Area: Usi finali

Tema: Studio e dimostrazione di forme di finanza innovativa e di strumenti di programmazione e di pianificazione per la promozione di tecnologie efficienti per la razionalizzazione dei consumi elettrici su scala territoriale e urbana

Responsabile Tema: Ilaria Bertini, ENEA

Note sugli autori: <http://panzieri.dia.uniroma3.it/>

# “Sviluppo di metodi di controllo per il sistema Edificio-Impianto”

---

## Sistemi di controllo e gestione

La diffusione dei sistemi informatici come insieme di computer e dispositivi in grado di gestire ed elaborare informazioni permette di giorno in giorno un controllo sempre più capillare ed efficace di impianti che prima dell'era informatica mancavano di una gestione centralizzata di tutte le informazioni inerenti all'impianto: dati dei sensori, stati di funzionamento, storico dei segnali, allarmi e notifiche. Tali sistemi informatici si devono occupare della gestione dei dati, della loro elaborazione e disponibilità dell'informazione.

Per gestione dati si intende il complesso di funzioni destinate allo scambio dati con le apparecchiature, al trattamento dei dati per la generazione di un insieme di dati adatto alle funzioni di elaborazione e di rappresentazione, all'archiviazione dei dati grezzi e dell'informazione aggregata frutto delle elaborazioni. Per elaborazione si intende tutto quanto responsabile della corretta interpretazione dei dati visti come insieme rappresentativo dello stato di evoluzione del processo controllato e dell'attuazione delle azioni di controllo. Il terzo elemento è la presentazione dell'informazione finalizzata alla fruizione da parte di sistemi esterni e dalle interfacce tra operatori e sistema.

## Sistemi informatici e interfacce

Sistemi informatici distribuiti, quindi composti da computer interconnessi e collegati in rete, utilizzati per monitorare dei sistemi fisici prendono il nome di SCADA, acronimo per *Supervisory Control And Data Acquisition* (acquisizione dati, supervisione e controllo). Questi sistemi sono composti da sensori, microcontrollori, reti di comunicazione e computer con interfacce uomo-macchina (HMI, Human-Machine Interface) ed offrono la possibilità di acquisire e gestire i dati dell'impianto, acquisiti attraverso sensori, ed operare sullo stesso tramite microcontrollori ed attuatori. L'evoluzione del processo controllato viene osservato e controllato tramite i dati acquisiti così da poter gestire gli stati in cui questo si trova e gli eventuali transitori tramite le corrette attuazioni.

I dati gestiti sono classificati in base a:

1. Provenienza e direzione
2. Qualità: informazione analogica o digitale, impulsiva o complessa
3. Caratteristiche elettriche del segnale: voltaggio e corrente continua o alternata, misura di tensione/corrente/resistenza

Questi sistemi informatici sono gestiti da operatori che tramite un'interfaccia sono in grado di leggere i dati forniti dal computer e di interagire con esso, dando comandi e gestendo le informazioni. L'interfaccia è di tipo grafico, una GUI (*Graphic User Interface*) che oltre ad usare il testo per comunicare con l'utente, offre anche un output grafico tramite indicatori, grafici, segnali luminosi. L'interfaccia grafica permette – oltre ad un'esperienza più gradevole per l'operatore – una maggiore efficacia nel saper interpretare i dati che vengono dai sensori e dagli elaboratori, quantità di dati spesso enormi e che fluiscono in real-time, che senza l'ausilio di grafici e indicatori aggregati, sarebbero ingestibili per un singolo operatore.

### **Gli ambienti di simulazione**

L'imponente informatizzazione odierna ha permesso anche lo sviluppo di software via via più raffinato che – in congiunzione con una potenza di calcolo in crescita quasi esponenziale – offre possibilità fino a pochi anni fa impensabili, tra cui quella di essere in grado di simulare sistemi fisici complessi e di grandi dimensioni. Tale simulazione avviene modellando i fenomeni reali con formule matematiche tramite programmi software. I fenomeni simulati che spaziano dalle condizioni climatiche alle reazioni chimiche, sono un'approssimazione della realtà in quanto non si possono considerare tutte le variabili in gioco nei sistemi reali, perché sono un praticamente infinite e non tutte conosciute.

I software di simulazione sono diventati lo strumento principe in molte aree ingegneristiche, perché permettono di simulare situazioni che nella realtà sarebbe impossibile o molto costoso poter provare. La loro diffusione ha creato una vera e propria disciplina e una serie di competenze fondamentali per tutti quelli che – provenendo da diversi ambiti – utilizzano l'informatica per la progettazione e l'implementazione delle soluzioni in problemi molto complessi.

L'informatica da sempre si è basata sul concetto di modularità e *layer*, ovvero la possibilità di definire in fase di progettazione in che modo le singole unità comunichino e interagiscano (i protocolli o prototipi) in modo da far sì che sia possibile utilizzarle (e riutilizzarle) con la garanzia di funzionamento se si rispettano i protocolli o parametri già definiti. Tale concetto risulta fondamentale quando si parla di simulazione perché nei sistemi complessi, come gli impianti industriali di medie-grandi dimensioni o i sistemi ad alta tecnologia come i satelliti artificiali ad esempio, le singole unità che si vanno a simulare (il singolo impianto, il sensore, l'utenza nel caso degli impianti oppure gli attuatori, il sistema operativo, i sensori nel caso del satellite) possono essere sostituite dall'elemento "reale" facendo sì che l'aderenza della simulazione con la realtà aumenti via via che si aggiungono elementi reali al contesto "artificiale" della simulazione.

Uno dei pacchetti software più utilizzato nell'ambito delle simulazioni e del calcolo scientifico è MATLAB, sviluppato dalla Mathworks, un software divenuto quasi uno standard de facto in ambito ingegneristico-scientifico. Questo software, che abbiamo utilizzato nell'ambito di questo progetto, oltre a fornire un insieme pressoché smisurato di strumenti

di calcolo molto efficienti, permette di interfacciarsi tramite l'uso di codice C con software e hardware esterno, che si integra all'interno dell'ambiente di simulazione Simulink insieme al resto dei componenti simulati.

## Controllo e retroazione

Il comportamento e il funzionamento di uno o più dispositivi viene comandato e regolato con un sistema di controllo, una logica – che viene applicata tramite un operatore umano o tramite hardware-software in maniera automatica – che porta al raggiungimento di una serie di obiettivi o specifiche: l'apertura o chiusura a richiesta di una valvola, il mantenimento di un valore (es. temperatura) all'interno di una certa soglia ecc ecc. Esistono diversi tipi di sistemi di controllo ma è fondamentale per tutti il concetto di feedback (retroazione), la capacità di un sistema di usare le proprie uscite per modificare gli ingressi.

Si parla di anello di controllo (control loop) quando sensori, algoritmi di controllo e attuatori funzionano in modo di mantenere il valore di una variabile il più possibile uguale ad un valore di riferimento. Ad esempio se si vuole mantenere la temperatura all'interno di una stanza (misurata da sensori) stabile su un certo valore (il setpoint) l'algoritmo di controllo fa sì che il riscaldamento o il raffreddamento (gli attuatori) entri in funzione fino a che non si arrivi all'obiettivo. In Figura 1 è visibile lo schema generico di un control loop.

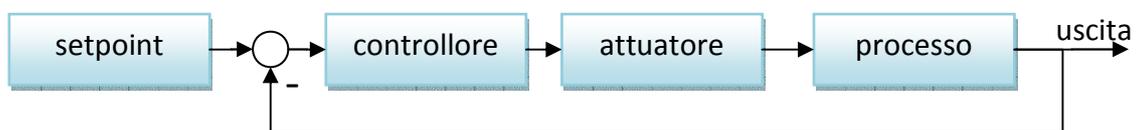


Figura 1: schema a blocchi di un sistema di retroazione

## Controllori PID

Il controllo PID è il modo in cui normalmente si definisce un controllo composto da tre azioni, il nome PID è composto infatti proprio dalle iniziali dei tre tipi di azione standard: P per proporzionale, I per integrale e D per derivativo. I controllori PID sono i più usati a livello industriale, anche i sistemi più complessi vedono spesso la presenza di questo tipo di controllori. Questo tipo di controllore oltretutto ha alla base una rappresentazione matematica molto semplice e potente, basata sulla trasformata di Laplace, che ha contribuito a renderlo facilmente applicabile ed utilizzabile. In Figura 2 è visibile uno schema a blocchi di un controllore PID generico, i parametri  $K_p$ ,  $K_i$  e  $K_d$  sono associati all'azione proporzionale, integrale e derivativa. In ingresso c'è l'errore, lo scostamento tra il valore da controllare e quello di riferimento (setpoint).

La progettazione di un controllore PID è composta dalla scelta della sua struttura e dei valori dei parametri.

La scelta della struttura consiste in quale azioni usare, ognuna delle quali introduce delle

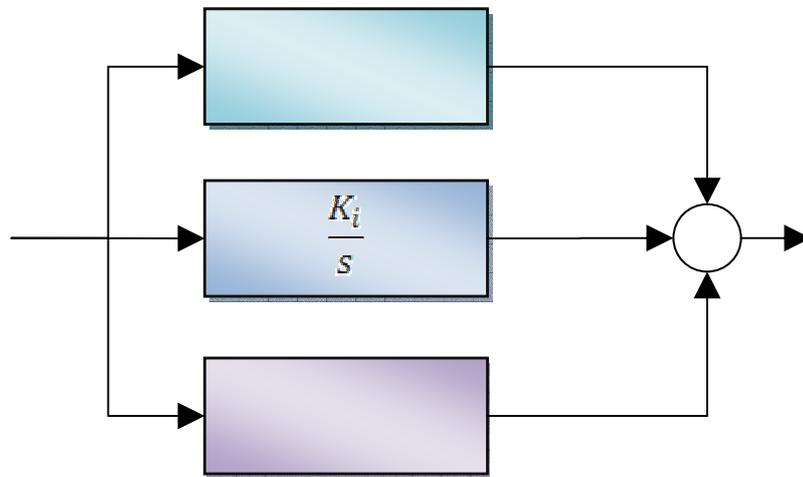


Figura 2: schema di un controllore PID

dinamiche al regime e durante al transitorio ben definite.

- a) Azione proporzionale: velocizza la risposta nel transitorio ma non elimina i disturbi a regime
- b) Azione integrale: elimina i disturbi a regime ma diminuisce la stabilità del controllore
- c) Azione derivativa: minimizza il transitorio eseguendo una “predizione” dell’errore, ma può sollecitare gli attuatori ed enfatizzare le alte frequenze

La scelta dei parametri (tuning) è basata sull’esperienza e per i sistemi più semplici esistono delle regole semplici per calcolare il valore dei parametri, introdotte da Ziegler e Nichols negli anni ‘40, successivamente migliorate e modificate anche dalle stesse aziende produttrici degli impianti da controllare.

### Algoritmi evolutivi, euristiche ed ottimizzazione

Gli algoritmi evolutivi sono una particolare classe degli algoritmi euristici ed in particolare sono inquadrati nella sfera delle meta-euristiche. Gli algoritmi euristici sono caratterizzati dalle seguenti proprietà:

- Approssimazione stocastica di alta qualità della soluzione di un determinato problema che può avvenire in diversi modi
- Robustezza al variare delle caratteristiche del problema da risolvere. Se un algoritmo euristico infatti, risolve il problema posto in un determinato caso, lo deve risolvere anche al cambiare delle condizioni al contorno
- Robustezza al variare dei parametri dell’algoritmo e ai vincoli delle soluzioni
- Facilità di implementazione

-Assenza di soggettività

-Forte adattabilità ai cambiamenti nei vincoli specifici contenuti nei problemi

Gli algoritmi genetici (GA, Genetic Algorithms) sono algoritmi euristici di ricerca ispirati al mondo biologico e in particolare al concetto di selezione naturale. Sono applicati specialmente nei contesti più complessi: funzione obiettivo discontinue, vincolate, non derivabili, non-lineari. L'algoritmo, di natura stocastica, partendo da un insieme di soluzioni di partenza si muove verso soluzioni migliori applicando operatori di ricerca stocastica e ricombinazione. Questi algoritmi sono molto diffusi ed utilizzati dagli anni '50, sono e sono disponibili diverse implementazioni anche già presente in pacchetti software commerciali di larga diffusione. Il termine algoritmo evolutivo va a comprendere tutti gli algoritmi che sono basati sulla metafora dell'evoluzione naturale ed emergenza della soluzione di un problema, tra cui quindi anche gli algoritmi genetici. Queste classi di algoritmi sono indicati per risolvere problemi nei quali si hanno scarse conoscenze a priori sulla soluzione e con uno spazio delle soluzioni di dimensioni tali da rendere impraticabile un approccio esaustivo.

Lo schema generale di funzionamento di questo tipo di algoritmi si riassume nei seguenti passaggi fondamentali:

1. Creazione di un insieme di soluzioni di partenza
2. Valutazione delle singoli soluzioni usando una funzione di *performance*
3. Selezione delle soluzioni migliori
4. Creazione di nuove soluzioni tramite operatori di mutazione e ricombinazione
5. Continua dal punto 2. finché non si realizza un criterio di step predeterminato

L'ottimizzazione classica è la disciplina che si occupa di trovare i punti di minimo (o massimo) di un modello matematico, composto da una serie di funzioni che possono essere sia reali che intere. La funzione da minimizzare (o ottimizzare) è chiamata funzione obiettivo, ed è composta da una serie di variabili chiamate variabili di decisione. Quando non si ha una sola funzione obiettivo ma diverse, si parla di ottimizzazione multi-obiettivo che ha fondamento sulla teoria di Pareto. Questo tipo di ottimizzazione non porta all'ottenimento di un punto ottimale ma di un insieme di punti definiti ottimi di Pareto, tra i quali sarà poi l'operatore o l'utente a scegliere.

## Logica Fuzzy

La logica fuzzy è una logica meno rigida di quella classica binario, dove gli assunti vero o falso sono gli unici stati di verità possibili. La realtà è sempre però ambigua, vaga, mutevole e la logica binaria offre una classificazione troppo rigida. Nella logica fuzzy l'appartenenza di un elemento ad un insieme è definito con un valore continuo  $[0,1]$  e non con uno stato binario (appartiene o non appartiene) e così nei sistemi fuzzy le proposizioni hanno un grado di verità continuo e perciò, nel ragionamento fuzzy, la conseguenza dovuta ad una causa è in proporzione al grado di verità della causa.

## Ottimizzazione e gestione ottimale di un sistema edificio-impianto

Partiamo dal definire il concetto di distretto energetico.

Il distretto energetico è un insediamento civile o industriale territorialmente localizzato che richiede un servizio di energia sia in forma termica che elettrica. Il maggiore guadagno in termini di efficienza energetica consiste nel mettere a fattor comune, in parte o in toto, questi servizi progettandoli e gestendoli con criteri di ottimizzazione multi-obiettivo.

Questa visione integrata permette di operare sull'ottimizzazione dei consumi delle utenze, della produzione dell'energia (anche in chiave economica) e della logistica dei trasporti.

Quest'ottimizzazione avviene in due fasi:

1. Progettazione ottimale: la progettazione viene svolta ponendo attenzione indicatori che rappresentano la capacità del progetto di soddisfare i requisiti delle utenze, sui costi, consumi e risparmio d'energia il tutto legato all'aspetto normativo vigente. Le varie scelte del progettista vengono simulate tramite modelli software in grado di prevedere la dinamica dell'intero sistema edificio-impianto.
2. Gestione ottimale: questa avviene tramite l'aiuto di sistemi intelligenti di controllo, diagnostica e gestione delle emergenze

Il ruolo di un simulatore software è di fondamentale importanza, nel primo caso è alla base della logica di CAD (Computer Aided Design), permette di verificare e sperimentare gli effetti delle scelte effettuate dal progettista che variano dal numero e dalla tipologia di componenti utilizzati alla loro dimensione. Nella gestione permette di simulare l'effetto delle scelte di controllo e di studiare lo scostamento del sistema reale da quello simulato, mantenendo il modello simulato sempre aggiornato.

Gli impianti considerati sono inseriti all'interno di un sistema dinamico che modella l'edificio e le sue proprietà. I singoli impianti quindi rispondono a delle specifiche di funzionamento ben definite, vincoli ed obiettivi da raggiungere sono alla base delle loro logiche di funzionamento. Essendo un sistema dinamico soggetto a diverse perturbazioni, più o meno prevedibili, gli algoritmi di controllo devono essere in grado di reagire alle variazioni e alle richieste che possono sopraggiungere durante il funzionamento.

Immaginando uno scenario futuro dove in un distretto energetico sono presenti diverse fonti di energia, i singoli impianti devono regolarsi in base ai flussi energetici di tutto il distretto e quindi una logica di controllo deve supervisionare le singole accensioni e spegnimenti. Inoltre all'interno dei singoli edifici laddove sono presenti diverse fonti energetiche bisogna decidere quali impianti utilizzare o se utilizzare l'energia proveniente dalla rete.

## ODESSE

La collaborazione tra ENEA e l'Università di Roma "Roma Tre" è stata finalizzata allo sviluppo di metodi di controllo per il sistema edificio-impianto. Tale lavoro si è svolto negli ambiti della piattaforma software di simulazione ODESSE (Optimal Design for Smart Energy), uno strumento che permette la simulazione dinamica di un edificio e dei suoi impianti e che offre al progettista un ausilio per ottimizzare la scelta e la tipologia degli impianti utilizzati. ODESSE è in grado di simulare dinamicamente l'esercizio di fonti energetiche funzionanti in generazione distribuita al servizio d'utenze reali (edifici) e con condizioni tariffarie, fiscali e normative reali. La costruzione dei modelli matematici alla base di ODESSE è fatta con un insieme di equazioni differenziali nel dominio del tempo a parametri variabili ed è basata sulla conoscenza fisica che governa i processi di funzionamento delle sorgenti distribuite (incluse le rinnovabili) delle reti e della dinamica di tutte le variabili che influenzano il confort termico degli edifici presenti all'interno del distretto.

La piattaforma è basata su codice scritto in C e su MATLAB/Simulink ed è completamente modulare, via via nuovi impianti vengono forniti dalle aziende e dalle università, così come vengono aggiunti nuovi materiali e nuove opzioni da offrire al progettista. I vari moduli comprendono: un'interfaccia grafica (vedi Figura 3), database di materiali, pre e post-processore dei dati di simulazione, simulatore dinamico, sistema di ottimizzazione parametrica basato su algoritmi evolutivi.

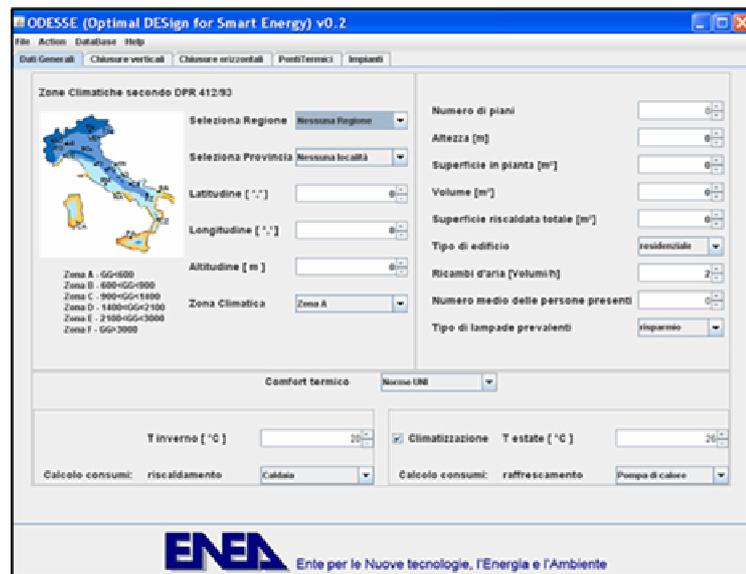


Figura 4: GUI di ODESSE

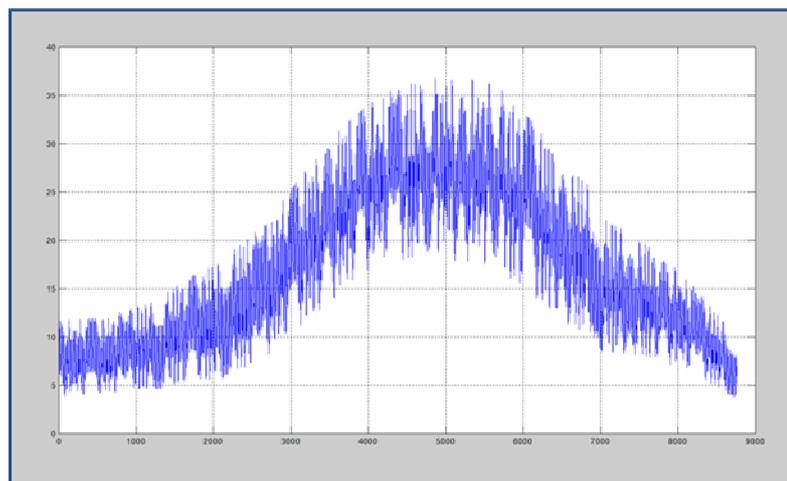


Figura 3: temperatura calcolata dal Neural Weather Generator

La piattaforma ODESSE simula giorno dopo giorno il comportamento degli impianti, le proprietà dei materiali e dell'edificio, rispettando il comportamento fisico sia a transitorio che a regime degli impianti e fornendo logiche di controllo degli stessi. Sono proprio queste logiche di controllo l'oggetto della prima parte di questo obiettivo intermedio.

Infatti le logiche implementate devono essere adattive, ovvero in grado di rimanere valide e funzionanti al variare dei parametri degli impianti e delle condizioni del contorno. Tra queste condizioni che variano ci sono ovviamente quelle climatiche, specialmente la temperatura che ha una grande influenza sul funzionamento degli edifici, essendo un valore fondamentale legato al comfort degli ambienti, questo incide sull'accensione-spegnimento degli impianti di riscaldamento e raffreddamento. La temperatura in ODESSE è simulata da un modulo chiamato NWG (Neural Weather Generator) che, date le coordinate geografiche, l'altitudine e il periodo dell'anno, è in grado di calcolare la temperatura media in gradi centigradi, con elevata precisione. In Figura 4 è visibile un esempio dell'uscita calcolata dal modulo NWG di una località in provincia di Lecce, Puglia.

## Gli impianti considerati

Nell'attività sono stati considerati i seguenti impianti:

- A. Microturbina a gas
- B. Caldaia a condensazione

L'impianto di solar cooling non è stato oggetto di questo lavoro per via dell'assenza del modello all'interno della piattaforma di simulazione.

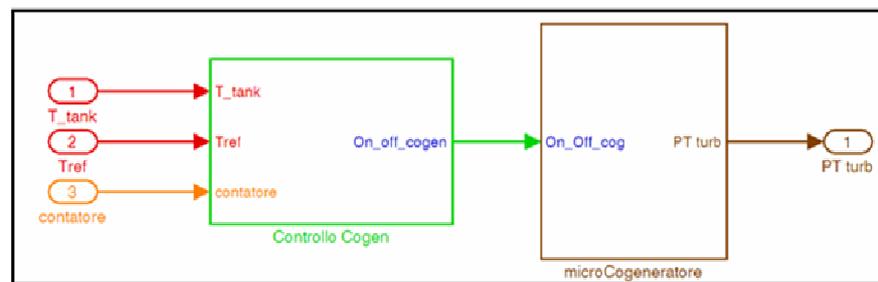


Figura 5: schema della microturbina con controllore

## Microturbina a gas

La microturbina a gas (MTG) è un impianto particolarmente adatto per applicazioni cogenerative e per la produzione distribuita di energia. Il grande vantaggio nell'uso della microturbina è la sua capacità di produrre sia energia elettrica che termica, evitando quindi dove possibile l'esigenza di una caldaia a gas, che comunque nelle configurazioni standard è sempre presente di *backup*, per compensare le mancanze

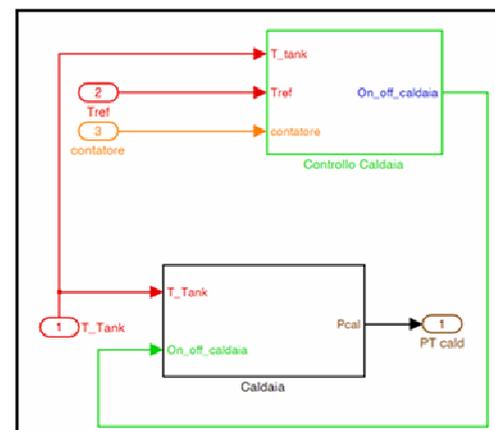


Figura 6: schema della caldaia con controllore

della turbina. Lo schema generale della turbina con controllore è presente in Figura 5.

Generalmente si distingue tra ultra-micro, micro e mini turbina a gas in base alla potenza elettrica, nel nostro caso parliamo di microturbina a gas in quanto il modello da noi considerato, la Turbec T100, presenta una potenza elettrica nominale di 100 kW. Le applicazioni di questa classe di microturbine sono spesso per la generazione distribuita grazie alla loro possibilità di essere messe in parallelo, la loro stabilità e le basse emissioni: sono quindi presenti sia nel settore industriale che nel terziario. Ovviamente la microturbina è in grado di erogare la potenza richiesta solo se l'utenza è in grado di smaltire il carico termico, quindi se la temperatura rilevata nella caldaia a recupero supera una soglia impostata la microturbina esegue una parzializzazione del carico fino a quanto la temperatura non ritorna al valore prefissato. E' proprio questa parzializzazione uno dei segnali di usati per il controllo di quest'impianto.

### Caldaia a condensazione

Le caldaie a condensazione sono dispositivi moderni in grado di ottenere rendimenti molto elevati recuperando il calore latente di condensazione del vapore acqueo nei fumi. Il modello da noi considerato è quello di una caldaia a condensazione a metano usata come *backup* del microgeneratore a turbina, anche se si può anche predisporre un utilizzo da *master* in assenza del microgeneratore. Il modello della caldaia con controllore è visibile in Figura 6.

### Modelli degli impianti

Come già accennato nell'ambito di quest'attività come piattaforma di simulazione si è usato MATLAB/Simulink. Tale ambiente è basato su una logica di modellazione dove i singoli blocchi sono sistemi o funzioni collegati fra di loro da archi che trasportano informazioni (segnali). Quindi al termine di questa prima attività si è arrivati alla definizione di una libreria base di modelli-blocchi dei singoli impianti, rendendo quindi possibile una riutilizzazione futura degli stessi.

Il modello della microturbina è visibile in Figura 8 ed è strutturato con degli input o costanti che vanno in ingresso ad un blocco scritto in linguaggio C che racchiude il modello vero e proprio del microgeneratore.

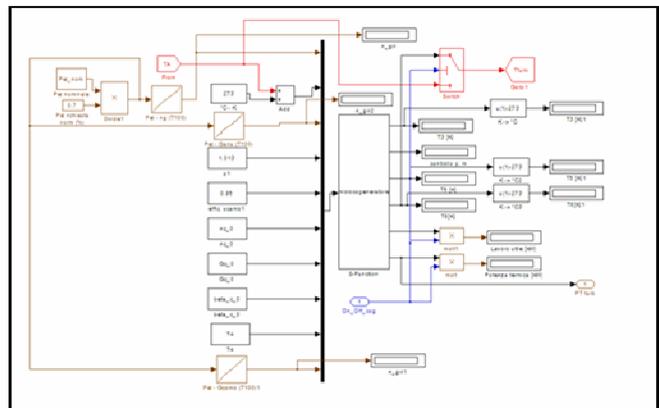


Figura 7: Modello MATLAB/Simulink della microturbina Turbec T100

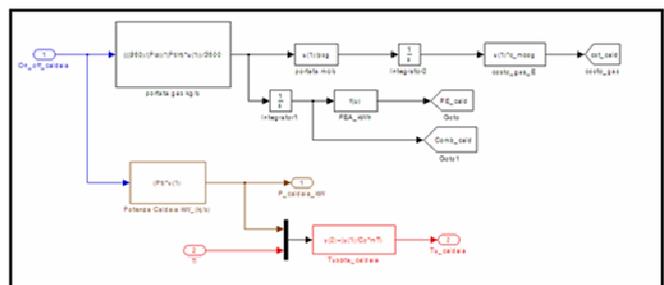


Figura 8: modello MATLAB/Simulink della caldaia

Il blocco scritto in linguaggio C prende il nome, in ambiente MATLAB, di S-Function. Lo scopo di questo tipo di blocchi Simulink è quello di permettere un'implementazione computazionalmente efficiente di un modello, cosa che è difficile ottenere usando il linguaggio di MATLAB (linguaggio M), che è interpretato e perciò meno efficace perché non direttamente eseguito dalla CPU. Questi blocchi sono strutturati in modo da interfacciarsi con il motore di integrazione numerica di Simulink in modo da calcolare i valori dell'equazioni differenziali (ODE) che descrivono il sistema che il blocco va a modellare.

In Figura 7 è presentato il modello della caldaia a condensazione.

## Descrizione dei segnali controllati

Per quanto riguarda la microturbina sono stati considerati i seguenti segnali:

1. Temperatura dell'acqua in uscita
2. Potenza termica generata
3. Potenza elettrica generata
4. Efficienza termica
5. Efficienza elettrica

Per quanto riguarda la caldaia:

1. Temperatura dell'acqua in uscita
2. Efficienza complessiva

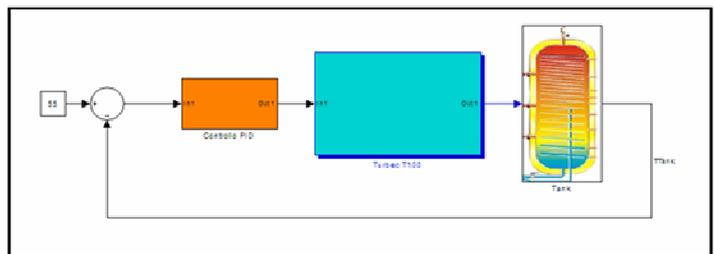


Figura 9: schema di controllo della microturbina con accumulo termico

## Controllo implementato

Il controllo degli impianti è stato soggetto a vincoli sia fisici (le caratteristiche delle macchine) che operativi, ovvero tenendo conto degli obiettivi da raggiungere e sulle norme da rispettare nell'operare sugli impianti.

Per la microturbina sono state sviluppate diverse metodologie di controllo. La prima (vedi Figura 9) si è basata su un controllore PID in grado di mantenere costante la temperatura di un tank di accumulo, al variare della temperatura esterna. Il controllore PID presenta in ingresso alla microturbina una serie di segnali un cui esempio è visibile in Figura 11, questi non sono altro che le regolazioni da dare alla microturbina in parzializzazione. Come si può vedere per via della natura del controllo PID ci sono numerose variazioni alla regolazione, tale condizione potrebbe generare uno stress eccessivo all'impianto che si vede di timestep in timestep cambiare l'ingresso di controllo.

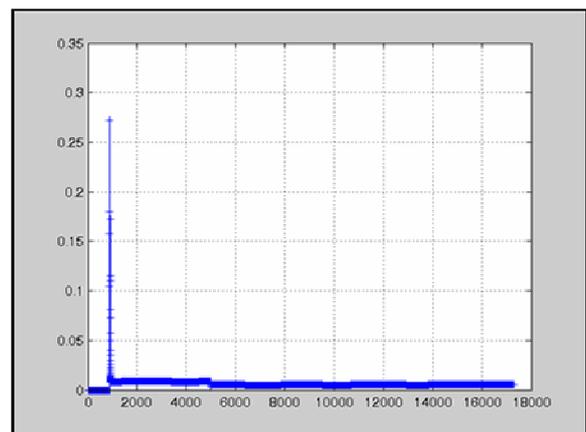


Figura 10: controllo PID della caldaia

Per questo motivo si è implementato un altro tipo di controllo, basato su un controllore discreto che offre uno sforzo di controllo minore, permettendo la definizione di una soglia di errore  $\epsilon$  entro la quale considerare l'errore nullo. In Figura 12 si può vedere un esempio di sforzo di controllo di questo controllore applicato alla microturbina. Questo controllore è programmato con una serie di regole che in base al range dell'errore offrono una regolazione più o meno energica, la scelta delle regole nonché del parametro  $\epsilon$  è stata effettuata con una serie di test e studio dei segnali considerati.

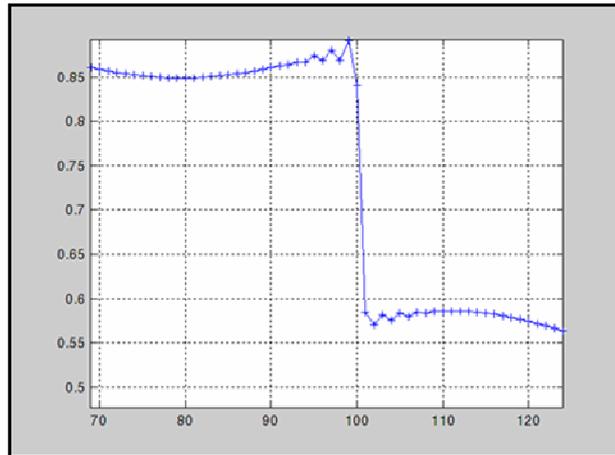


Figura 11: controllo per la microturbina dal regolatore PID

Per la caldaia a condensazione si è progettato un controllore PID in grado di mantenere la temperatura dell'accumulo termico al valore richiesto. In figura N è possibile vedere il controllo in ingresso alla caldaia per mantenere costante la temperatura dell'accumulo termico.

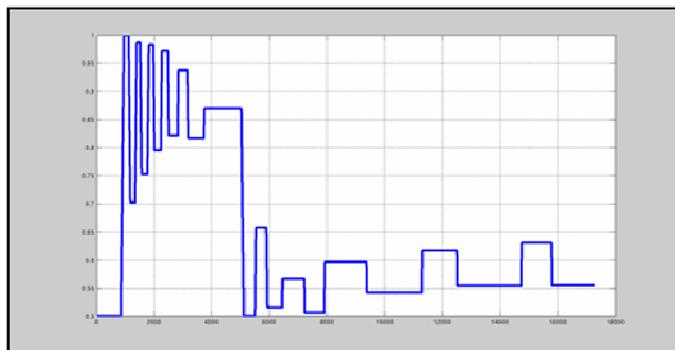


Figura 12: controllo del PID per la microturbina

Nella tabella seguente abbiamo riassunto le prestazioni dei controllori con un segnale costante (gradino di 55° C) e con una rampa (da 45°C a 80°C), il tutto è stato simulato nell'arco di una giornata, considerando disturbi e variazioni di temperatura. Come indici di confronto abbiamo considerato l'errore a regime e lo sforzo di controllo, definito come:  $\sum_{i=1}^M u_i^2$  con  $u_i$  l'i-esimo valore del controllo applicato all'impianto controllato.

In questa tabella è stato considerato un accumulo termico da controllare di 50000 m<sup>3</sup>, una microturbina da 100 kW e una caldaia da 350 kW.

Tipo di controllo	Errore a regime	Sforzo di controllo
Microturbina – PID – gradino	0.0293	7934
Microturbina – PID – rampa	0.0474	10912
Microturbina – contr.discr. – gradino	0.3127	6597
Microturbina – contr.discr. – rampa	0.4018	9817

Caldaia – PID – gradino	0.0052	0.86
Caldaia – PID – rampa	0.0093	1.2743

Abbiamo anche testato gli stessi controllori con degli impianti di taglia differente, considerando quindi una microturbina di 60 kW e una caldaia di 200 kW.

Tipo di controllo	Errore a regime	Sforzo di controllo
Microturbina – PID – gradino	0.0452	14422
Microturbina – PID – rampa	0.3232	14900
Microturbina – contr.discr. – gradino	0.2610	12874
Microturbina – contr.discr. – rampa	0.4283	13312
Caldaia – PID – gradino	0	1.9544
Caldaia – PID – rampa	0	3.84

Ora le stesse taglie e controllori per un accumulo termico di 10000 m<sup>3</sup>.

Tipo di controllo	Errore a regime	Sforzo di controllo
Microturbina – PID – gradino	0.0356	7145
Microturbina – PID – rampa	0.0384	9900
Microturbina – contr.discr. – gradino	0.4843	6872
Microturbina – contr.discr. – rampa	0.2805	6511
Caldaia – PID – gradino	0.0599	0.7381
Caldaia – PID – rampa	0.0965	1.0276

Tipo di controllo	Errore a regime	Sforzo di controllo
Microturbina – PID – gradino	0.0342	9905
Microturbina – PID – rampa	0.1134	14000
Microturbina – contr.discr. – gradino	0.4359	9000
Microturbina – contr.discr. – rampa	0.4573	13240
Caldaia – PID – gradino	0.0975	1.2958
Caldaia – PID – rampa	0.1685	2.3262

Dalla serie di sperimentazioni con le varie combinazioni di taglie e accumuli termici i controllori implementati sono risultati, come visibile dai valori presentati, efficaci anche al variare delle tipologie degli impianti e dei processi controllati.

## Gestione ottimale elettrica e termica

Come già detto il problema di ottimizzare un sistema edificio-impianti è decisamente complesso: sistemi non-lineari, diversi vincoli da rispettare e obiettivi da raggiungere, contesto dinamico con variazioni più o meno imprevedute (guasti, variazioni di richieste dall'utenza ecc ecc). Vista la complessità del problema si è deciso di avere un approccio black-box al problema, sviluppando un sistema di ottimizzazione e controllo basato su algoritmi evolutivi e logica fuzzy.

L'approccio black-box consiste nel trattare il problema considerando solo l'aspetto funzionale del sistema, osservando perciò gli input e gli output e le loro relazioni, evitando un approccio matematico-formale che, se non impossibile, è comunque molto complesso e oneroso. Gli algoritmi evolutivi sono molto adatti a questo tipo di approccio, perché le soluzioni che manipolano sono solamente una rappresentazione del problema di partenza e la loro ricerca avviene, al contrario delle tecniche classiche come la discesa del gradiente, seguendo delle funzioni di performance specificate che valutano l'efficacia di ogni soluzione.

Nel nostro caso abbiamo implementato un algoritmo evolutivo usando il linguaggio di programmazione C++, l'algoritmo esegue un'ottimizzazione multi-obiettivo considerando diverse funzioni obiettivo. La metodologia è un algoritmo evolutivo multi-obiettivo con elitismo ad ottimizzazione vincolata, utilizzando dei sistemi di *niching* e *crowding* (di gestione della diversità delle soluzioni) per far sì che il fronte di Pareto ottenuto sia il più ampio possibile. L'algoritmo è stato sviluppato come modulo all'interno di una piattaforma scientifica di ottimizzazione, così da permetterne lo studio e l'analisi sottoponendolo a

problemi di benchmark e facendo confronti con lo stato dell'arte sull'ottimizzazione multi-obiettivo (algoritmi NSGA-II, SPEA e PAES).

Uno schema dell'algoritmo è visibile in Figura 13.

L'algoritmo è stato poi esteso per considerare anche l'ottimizzazione dinamica ed è stata posta particolare attenzione alla robustezza e alla diversità delle soluzioni.

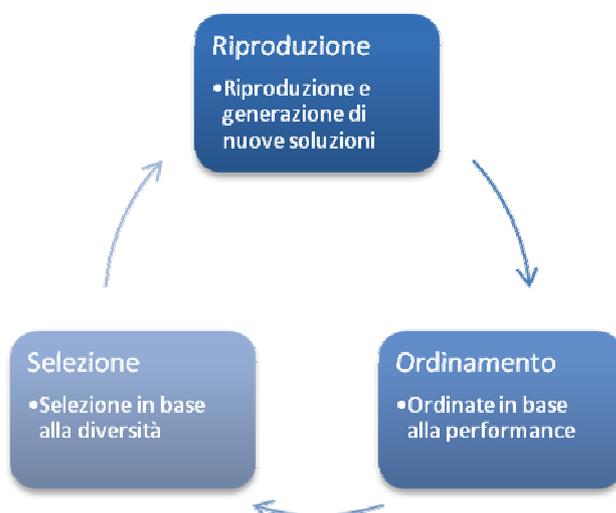


Figura 13: schema dell'algoritmo evolutivo utilizzato

Le soluzioni sono state rappresentate da stringhe di valori interi e reali che descrivono per ogni impianto presente nel sistema edificio-impianto le seguenti caratteristiche:

- Utilizzo (accensione) dell'impianto
- Soglie e parametri delle leggi di controllo del singolo impianto

Le soluzioni dell'algoritmo sono soggette ovviamente ai vincoli strutturali e fisici dell'intero sistema, per questo motivo durante la ricerca e l'esecuzione dell'algoritmo lo spazio delle soluzioni è stato suddiviso in regioni *feasible* e *non-feasible*, applicando una strategia che elimina le soluzioni non accettabili ma le considera comunque come punto di partenza per l'esplorazione dello spazio delle soluzioni.

In altri termini per ogni impianto vengono ottimizzate le sue leggi di utilizzo, le regole che lo governano. Nei sistemi più complessi queste regole possono essere di natura fuzzy. Le funzioni obiettivo da ottimizzare sono state le seguenti:

-Comfort termico all'interno degli edifici: media della somma degli scostamenti della temperatura effettiva da quella definita di comfort (temperatura di riferimento). La formula è la seguente:

$$\Delta T = \frac{\sum_{i=1}^N |T_{rif} - T_{room}|}{N}$$

con N il numero di campioni considerati per il calcolo.

-Costi di gestione

-Efficienza termica dei singoli impianti: la quantità di energia che viene trasformata in lavoro utile  $\eta_{th} = \frac{W_{out}}{Q_{in}}$

-Efficienza globale: il rapporto tra la quantità di energia netta consumata dall'edificio e la quantità di energia primaria usata dagli impianti per produrla

-Indice di impatto ambientale: quantità di CO<sub>2</sub> emessa dal sistema

L'algoritmo restituisce un fronte di Pareto, un insieme di soluzioni pareto-ottimali. Da questo set di soluzioni è poi possibile in base alla proprie priorità e necessità, dare più o meno importanza ad una singola funzione obiettivo in modo da scegliere il set di variabili più congeniale.

L'algoritmo funziona sia in modalità offline che in modalità online.

In modalità offline, a partire da uno stato viene fornito l'insieme di soluzioni pareto-ottimali senza considerare l'aggiornamento e l'evoluzione del sistema. Si parte quindi da una condizione iniziale e si trova il set di parametri più efficiente per quello stato iniziale: questo è ciò che generalmente si fa in ottimizzazione, chiamata in questo caso ottimizzazione statica.

In modalità online, l'insieme di soluzioni pareto-ottimali viene aggiornato insieme al sistema e alle sue variazioni. Ad ogni evoluzione del sistema, l'insieme delle soluzioni viene aggiornato rivalutando la loro efficacia (performance) che nel frattempo può essere mutata

ed esplorando nuove parti dello spazio delle soluzioni inseguendo la soluzione ottimale. Questo tipo di approccio si chiama ottimizzazione dinamica e considera delle proprietà delle soluzioni che nell'approccio statico non vengono considerate, come la loro robustezza e la diversità.

Una soluzione si può considerare robusta quando la sua performance non viene alterata (o viene alterata in maniera minima) quando cambiano i parametri del sistema. La diversità invece considera quanto le soluzioni siano simili tra di loro, calcolando una varianza o deviazione standard se si considerano soluzioni reali oppure considerano operatori di distanza come la distanza di Hamming.

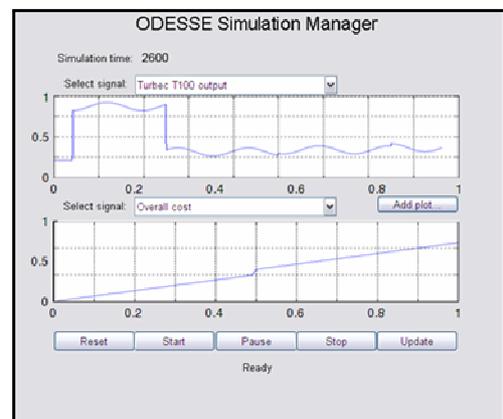


Figura 14: GUI del pannello di controllo della simulazione

## Pannello di gestione in Java/MATLAB.

La piattaforma MATLAB permette di realizzare GUI con uno strumento chiamato GUIDE (GUI Developer). Abbiamo sviluppato una GUI (vedi Figura 14) in grado di interfacciarsi con la simulazione Simulink ed in grado di gestirla e controllarla. E' possibile visualizzare gli output della simulazione, i vari indicatori e parametri, nonché fermarla ed avviarla. Tale applicazione, fondamentalmente composta da classi Java, offre anche la possibilità di gestire non una simulazione ma un reale impianto, infatti tra la GUI e la simulazione Simulink c'è uno strato intermedio di comunicazione. Questo strato non è altro che un'interfaccia software tra la GUI e il livello dell'impianto, che può essere o reale o simulato.

Lo strato intermedio fornisce le chiamate per la gestione dell'impianto alla GUI e nell'altro senso gestisce le callback della GUI ad ogni update del sistema. Quindi è questo strato a porsi da tramite tra il mondo software e l'eventuale mondo hardware, rendendo trasparente alla GUI (e quindi all'operatore) le operazioni di interfacciamento con gli impianti oggetto della simulazione, siano essi software o hardware.

Qualora sia necessario lo strato di comunicazione può girare su un sistema operativo real-time (RTOS), che garantisce un ritardo minimo sulla gestione degli interrupt hardware, quindi sull'acquisizione dei segnali provenienti dai sensori.

## **Caso reale e rapporto hardware**

Nel caso di un'applicazione del sistema di gestione e controllo progettato e implementato ad un sistema reale, è stato svolto uno studio sull'hardware da utilizzare, considerando:

- Schede di interfaccia
- Sensoristica
- Attuatori

Lo studio è stato effettuato al termine di esperienze ENEA ed è stata redatto un elenco con specifiche e figure dei vari componenti selezionati, ponendo particolare attenzione sulle loro accuratezze e classi di protezione secondo la normativa EN60529.