

Identificazione termodinamica di un edificio

Elena Toffoli, Giancarlo Baldan e Guido Albertin

Abstract—La relazione si propone di illustrare il lavoro svolto per identificare il modello termodinamico di un edificio. Le rilevazioni di temperatura all'esterno e all'interno della struttura sono state utilizzate per ottenere un modello dinamico lineare capace di descrivere opportunamente la distribuzione della temperatura nel tempo e nello spazio negli ambienti interni. Come ulteriore variabile indipendente si è impiegata una misura della radiazione solare ricevuta. Le dimensioni e la disposizione delle stanze permettono di identificare l'edificio come un'unità abitativa residenziale di circa cento metri quadri disposti su due piani. Alla fase di misurazione e successiva identificazione di un modello lineare tramite metodi di tipo subspace con selezione degli ingressi ottimali per la predizione in catena aperta è stata affiancata una fase di modellizzazione e simulazione della dinamica della temperatura nell'edificio. Un ulteriore passo è stata la successiva selezione delle uscite più significative del modello lineare allo scopo di ottenere un filtro numerico tempo variante in grado di predire con sufficiente accuratezza la temperatura all'interno dell'edificio riducendo comunque il numero di sensori iniziali di almeno un ordine di grandezza.

I. INTRODUZIONE

Il progetto prende in esame sistemi dinamici distribuiti in cui si hanno a disposizione molteplici misure (dirette o indirette) delle quantità di interesse. Il primo obiettivo è identificare il modello dinamico che governa l'intero processo tramite tecniche di identificazione per sistemi lineari. Successivamente il modello ottenuto sarà impiegato per ridurre il numero di sensori impiegati al fine di ridurre i costi per il monitoraggio e semplificare l'eventuale implementazione di algoritmi di controllo. Pertanto si tratterà di individuare quali sensori tra quelli inizialmente collocati forniscano maggiore quantità di informazione sul sistema. In altri termini la qualità della stima non deve peggiorare significativamente eliminando un considerevole numero di sensori opportunamente selezionati. Non si tratterà quindi di posizionamento ottimo a priori ma di un criterio di scelta a posteriori su un insieme di posizionamenti predeterminati.

Nel dettaglio il sistema preso in esame è il comportamento termico di un edificio isolato su due piani e di circa 300 metri cubi le cui piante vengono riportate in figura 1.

Si vuole quindi ricavare il modello dinamico che descriva l'andamento della temperatura all'interno degli ambienti. Per tale scopo si hanno a disposizione una sessantina di dispositivi *MoteIV Tmote Sky* muniti di sensori di temperatura, umidità e luminosità (sia nella banda del visibile che nell'infrarosso).

Per l'identificazione del modello si è ricorsi a delle tecniche di identificazione basate su sottospazi quali N4Sid e PBSID. Una prima fase di selezione dei sensori è legata alla scelta degli ingressi esogeni da utilizzare per l'identificazione; si è cercato di arrivare alla selezione di due o tre ingressi partendo dai nove che si hanno a disposizione. Non essendo attuabile una ricerca esaustiva della combinazione ottima di ingressi

si ricorre ad un algoritmo di tipo greedy rinunciando così alla soluzione ottima ovvero il miglior set di ingressi possibile secondo un criterio di bontà della stima. L'implementazione di tecniche volte alla riduzione degli ingressi a partire dal sistema identificato con tutte le misure disponibili si è rivelata non vantaggiosa, in quanto il comportamento del sistema identificato in evoluzione libera risultava peggiore rispetto a quello derivante da un'identificazione con un sottoinsieme degli ingressi. Una volta ricavato il modello viene introdotto un filtro di Kalman tempo variante allo scopo di calcolare un stima delle uscite basandosi sulle misure fornite da un ridotto numero di sensori interni. In questo caso oltre ad un algoritmo greedy si utilizzano altre procedure euristiche per avvicinarsi il più possibile ad una soluzione ottima. Nello specifico si ricorre ad algoritmi di ricerca locale e genetici. Ci si è proposti di ridurre il numero complessivo di sensori di un ordine di grandezza e quindi di utilizzarne un numero che va da sei a dieci dei 65 a disposizione.

Alla fase sperimentale di raccolta di dati si affianca una fase di modellizzazione e simulazione della dinamica della temperatura nell'edificio. Questa fase ha richiesto uno studio preliminare sulle leggi della diffusione del calore universalmente accettate e, successivamente, lo studio di metodi di calcolo numerico dedicati alle equazioni alle derivate parziali. Si sono dovuti affrontare in questa parte problemi numerici che difficilmente si possono incontrare come l'inversione di matrici da 2GB. Il vantaggio di disporre di un simulatore sta nel fatto che è possibile generare dati anche quando non è possibile farlo sperimentalmente ed è inoltre il primo passo per la simulazione di un sistema di controllo.

II. PROPAGAZIONE DEL CALORE

Il calore è una particolare forma di energia che si manifesta solo tra corpi che non sono in equilibrio termico. Lo studio di come questa grandezza si propaghi nello spazio ha portato alla formulazione di alcune semplici leggi teoriche sulla sua diffusione.

In questa sezione verranno presentate brevemente tali leggi e saranno messi in evidenza gli aspetti d'interesse per i nostri scopi.

A. Trasmissione per conduzione

La trasmissione del calore per conduzione avviene in qualsiasi regione dello spazio non vuota. Questo tipo di trasmissione è governata dalla legge di Fourier che mette in relazione il flusso di calore attraverso una sezione ortogonale ad una data direzione con la temperatura. Se, ad esempio, la direzione in esame è quella individuata dall'asse x , tale legge assume la forma:

Complesso vecchio di Agraria, ex presidenza - vicolo Portello

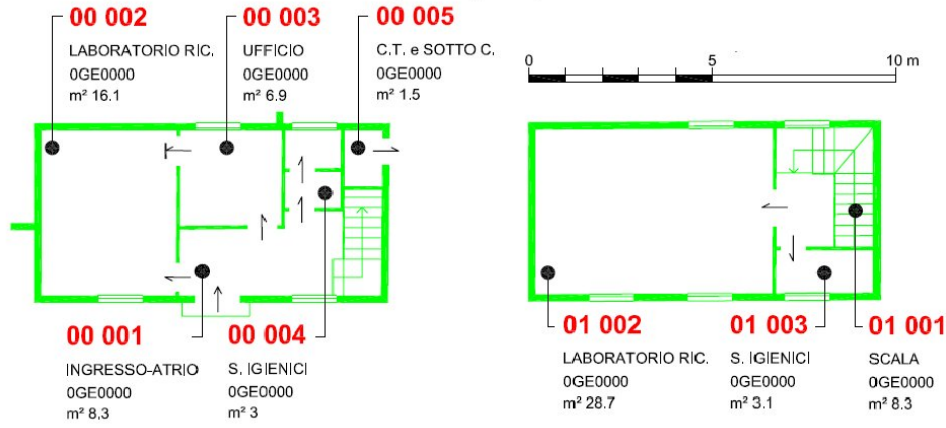


Fig. 1. Piante dell'edificio

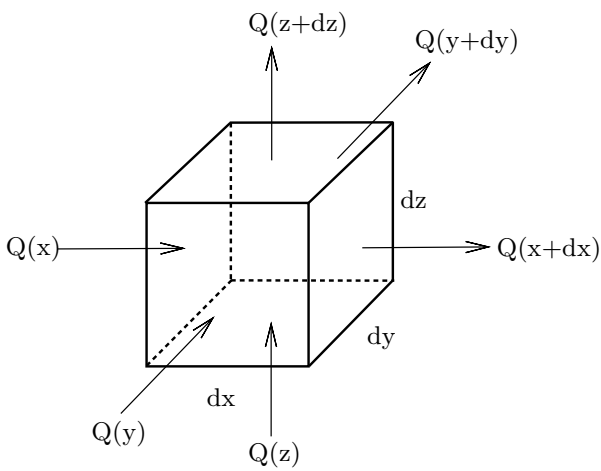


Fig. 2. Volume infinitesimo di materiale e rappresentazione del calore scambiato per unità di tempo in tutte le direzioni

$$q = -k_x \frac{\partial T}{\partial x} \quad (1)$$

in cui q rappresenta il flusso di calore (W/m^2) e k_x , nota come conducibilità termica lungo l'asse x , è una proprietà fisica del mezzo in cui avviene la propagazione ed ha le dimensioni di W/mK .

A partire da questa relazione è possibile ottenere la legge completa per la conduzione del calore facendo un semplice bilancio energetico. A tal fine si prenda in esame la figura 2 in cui il calore assorbito dal volume di materiale è dato da:

$$Q_{ass} = Q(x) + Q(y) + Q(z) = q(x)dydz + q(y)dx dz + q(z)dxdy$$

mentre il calore ceduto risulta:

$$Q_{ced} = Q(x + dx) + Q(y + dy) + Q(z + dz) = q(x+dx)dydz + q(y+dy)dx dz + q(z + dz)dxdy.$$

Supponendo, per ragioni di completezza, che all'interno dell'elemento si trovi una sorgente di calore che sviluppa G

Watt di potenza per unità di volume, la variazione di energia interna che subisce l'elemento in esame in un intervallo temporale infinitesimo dt si può così calcolare:

$$dE = [G dxdydz + Q_{ass} - Q_{ced}] dt = [G dxdydz + (q(x) - q(x + dx))dydz + (q(y) - q(y + dy))dxdz + (q(z) - q(z + dz))dxdy] dt.$$

Effettuando ora uno sviluppo in serie di Taylor arrestato al secondo termine, l'aumento di energia interna assume poi la forma:

$$dE = \left[G - \frac{\partial q}{\partial x} - \frac{\partial q}{\partial y} - \frac{\partial q}{\partial z} \right] dxdydz dt. \quad (2)$$

Supponiamo infine che non vi siano cambiamenti di fase e che tutta l'energia interna venga convertita in un aumento di temperatura secondo la nota legge:

$$dE = c dm dT = \rho c dxdydz dT \quad (3)$$

in cui ρ e c rappresentano rispettivamente la densità e la capacità termica del mezzo. In tale ipotesi, uguagliando i secondi membri delle relazioni 2 e 3 ed utilizzando la 1 si perviene alla seguente equazione:

$$\rho c \frac{\partial T}{\partial t} = G + \frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_z \frac{\partial T}{\partial z} \right)$$

che può essere resa più compatta con la notazione vettoriale:

$$\rho c \frac{\partial T}{\partial t} = G + \nabla \cdot \left(k_x \frac{\partial T}{\partial x}, k_y \frac{\partial T}{\partial y}, k_z \frac{\partial T}{\partial z} \right) \quad (4)$$

e che governa la conduzione del calore nello spazio e nel tempo.

L'equazione trovata è una equazione differenziale alle derivate parziali, lineare e nell'unica incognita T . In essa compaiono cinque parametri¹ che possono essere funzione dello spazio se il mezzo non è omogeneo. Inoltre nel caso il mezzo sia isotropo le tre conducibilità relative alle tre direzioni

¹Solo quattro indipendenti

| | Densità (Kg/m^3) | Capacità ($J/kg K$) | Conducibilità (W/mK) |
|----------|-------------------------|--------------------------|-----------------------------|
| Aria | 1.29 | 717.9 | 0.0252 |
| Muro | 1700 | 837 | 0.658 |
| Tramezza | 500 | 837 | 0.128 |
| Legno | 421 | 2720 | 0.15 |
| Vetro | 2800 | 800 | 0.81 |
| Cotto | 1000 | 837 | 0.658 |

TABLE I

PROPRIETÀ FISICHE DI ALCUNI MATERIALI PRESENTI GENERALMENTE IN UN EDIFICIO

sono uguali tra loro. Pertanto nel più comune dei casi in cui il mezzo è omogeneo ed isotropo la conducibilità può essere portata fuori dal simbolo di gradiente e l'equazione assume la sua forma più comune:

$$\rho c \frac{\partial T}{\partial t} = G + k \nabla^2 T. \quad (5)$$

I valori di densità, capacità e conducibilità termica di alcune sostanze che saranno utilizzate in seguito sono riportate in tabella I

Si osservi come l'aria presenti una conducibilità termica molto bassa il che la rende un ottimo isolante dal punto di vista del calore scambiato per conduzione.

B. Trasmissione per irraggiamento

Qualsiasi corpo ad una temperatura superiore allo zero assoluto emette radiazione elettromagnetica. Questa radiazione, che aumenta con l'aumentare della temperatura, trasporta energia che può considerarsi una forma di calore. Quando un altro corpo viene investito da questa radiazione infatti, ne assorbe una percentuale che tende ad innalzare la sua temperatura fino al raggiungimento dell'equilibrio termico in cui la radiazione emessa è pari a quella assorbita.

I fenomeni di emissione ed assorbimento hanno andamenti diversi non solo in funzione della temperatura ma anche della lunghezza d'onda della radiazione. Ad esempio il sole, che ha un comportamento molto simile a quello di un corpo nero, ha una distribuzione dell'irraggiamento² in funzione della lunghezza d'onda che è ben descritta dalla legge di Planck:

$$I(\lambda) = \frac{2h\pi c^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1}$$

il cui andamento è riportato in figura 3.

Analogamente anche l'assorbimento³ dipende, anche se in modo meno marcato, dalla lunghezza d'onda oltre che dal tipo di materiale. Si trova infatti che la percentuale di energia assorbita relativamente alle lunghezze d'onda inferiori a $\lambda_c = 4 \cdot 10^{-6}$ è in generale molto inferiore a quella relativa a lunghezze d'onda maggiori. Questa disparità è evidente in tabella II.

²Potenza prodotta per unità di superficie

³Consideriamo in questo caso un'incidenza normale alla superficie da parte della radiazione

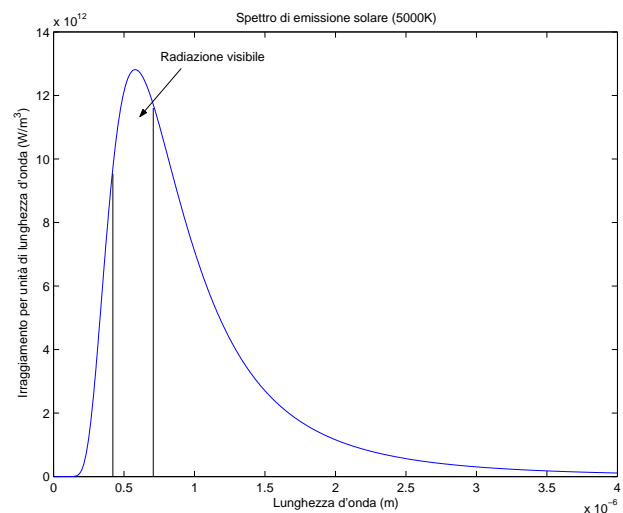


Fig. 3. Spettro di emissione da parte di un corpo nero alla temperatura di 5000K

| | Percentuale assorbita ($\lambda < \lambda_c$) | Percentuale assorbita ($\lambda > \lambda_c$) |
|-------|--|--|
| Muro | 0.12 | 0.91 |
| Legno | 0 | 0.9 |
| Cotto | 0.55 | 0.92 |

TABLE II

COEFFICIENTI DI ASSORBIMENTO DI ALCUNI MATERIALI PRESENTI ALL'ESTERNO DI UN EDIFICIO

C. Trasmissione per convezione

Con trasferimento di calore per convezione si intendono due tipi di trasferimento assai diversi. Il primo fa riferimento alla propagazione del calore all'interno di un fluido il secondo invece al trasferimento di calore tra un solido ed un fluido.

La trasmissione per convezione all'interno di un fluido prevede il trasferimento di calore per mezzo di uno spostamento di materia. I fluidi infatti tendono a modificare con la temperatura altre loro proprietà fisiche quali la densità. Ciò comporta che il fluido più freddo e quindi più denso stratifichi in basso mentre il fluido più caldo tende ad andare verso l'alto. Quindi, anche se nessuna particella elementare ha scambiato energia, di fatto nei fluidi avviene una redistribuzione della temperatura e quindi un flusso di calore.

Un modello matematico che descrive bene questo tipo di fenomeni è stato presentato da Navier e Stokes e prevede di imporre non solo un bilancio dell'energia scambiata ma anche due condizioni di conservazione della massa totale e della quantità di moto. Ne risulta un vero e proprio sistema di equazioni termodinamiche non lineari nelle tre incognite temperatura, densità e velocità. Non riportiamo in questa sede la formulazione di tali equazioni perché esulano dagli scopi di questo lavoro. Maggiori approfondimenti al riguardo si possono trovare in [16]

La trasmissione di calore per convezione tra un fluido ed un solido avviene secondo una legge molto semplice detta legge di Newton in base alla quale il flusso di calore tra un solido ed un fluido a contatto è proporzionale alla differenza delle

loro temperature:

$$q = h(T_1 - T_2).$$

La costante h , che prende il nome di coefficiente di convezione, ha dimensioni fisiche di $W/m^2 K$ e cambia in base alle caratteristiche del fluido e del solido. Se il fluido è stagnante h è solo una funzione della normale alla superficie di contatto, nel senso che vale circa $4W/m^2 K$ sulle superfici orizzontali a flusso ascendente mentre scende a $1.5W/m^2 K$ per le superfici orizzontali a flusso discendente. Se invece il fluido è forzato⁴ h dipende in modo preponderante dalla velocità del fluido ed in minor misura dal tipo di solido. In condizioni climatiche normali, per la maggior parte dei materiali che rivestono un edificio, h si può considerare prossimo a $15W/m^2 K$

III. SIMULAZIONE NUMERICA

A partire dai modelli teorici sulla propagazione del calore testé introdotti è possibile costruire un software di simulazione numerica che permetta di calcolare l'evoluzione della temperatura all'interno di un certo sistema. Lo scopo di questo simulatore è duplice, da una parte è in grado di fornire una quantità molto vasta di dati che possono rivelarsi utili per testare le prestazioni degli algoritmi di identificazione e selezione dei sensori, dall'altra permette di verificare che anche per un sistema complesso retto da una equazione alle derivate parziali è possibile trovare un modello lineare che approssima la sua dinamica. Questo fatto avvalorava le successive scelte riguardo le tecniche di identificazione impiegate che sono tutte basate su modelli lineari.

Questa sezione illustra uno dei più classici procedimenti per la realizzazione di un integratore di equazioni alle derivate parziali, il metodo agli elementi finiti. Questa procedura si può suddividere in tre fasi distinte: la generazione della mesh, la formulazione delle equazioni e l'integrazione delle stesse. Ciascuna fase verrà ora analizzata con maggior dettaglio.

A. La costruzione della mesh

Così come qualsiasi integratore di equazioni differenziali ordinarie prevede una discretizzazione temporale, il metodo agli elementi finiti o *FEM* necessita che lo spazio in cui esiste il sistema da simulare venga discretizzato. Discretizzare lo spazio significa di fatto creare una partizione del volume costituita da poliedri in modo tale che ogni faccia di ciascun poliedro sia in comune con quella di un altro, questa particolare partizione prende comunemente il nome di mesh ed i poliedri che la compongono vengono detti elementi. Il tipo di poliedro utilizzato in questa sede è il tetraedro; quest'ultimo offre numerosi vantaggi tra cui la semplicità nella formulazione successiva delle equazioni e la facilità nel reperire algoritmi che generano discretizzazioni in modo automatico.

Il sistema che andremo a simulare, e quindi il volume che dovremo discretizzare, è rappresentato dallo stesso edificio su cui sono state svolte le indagini sperimentali e la realizzazione della mesh è avvenuta in due fasi successive. In una prima fase

⁴Come avviene ad esempio all'aria aperta dove il vento fa da forzante

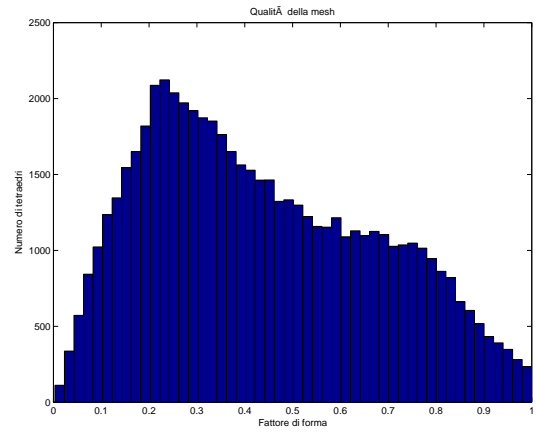


Fig. 4. Distribuzione della qualità dei tetraedri appartenenti alla mesh

è stata realizzata la mesh delle pareti dell'edificio utilizzando l'algoritmo distmesh descritto in [18] al quale in input è stata fornita una dettagliata descrizione geometrica del sistema, nella seconda fase, poi, è stata realizzata la mesh dell'aria contenuta nelle stanze mediante un algoritmo advancing front da noi sviluppato seguendo, in parte, le indicazioni riportate in [17]. Questa tecnica basata su due mesh successive è ideale per generare mesh in sistemi in cui vi siano due sottosistemi che interagiscono tra loro, nel nostro caso aria e pareti, ed in cui la superficie di separazione tra i due sottosistemi deve avere una geometria prefissata. Le mesh che si ottengono in questo modo sono dette a sotto domini.

Gli algoritmi advancing front partono dalla mesh di una superficie costituita da triangoli e, ad ogni iterazione, eliminano una faccia dalla superficie inserendo un nuovo tetraedro. La difficoltà nell'implementazione di questi algoritmi sta nel fatto che essi si possono classificare come algoritmi greedy e, come tali, sono "miopi" e le scelte effettuate ai primi passi possono compromettere la creazione dei tetraedri successivi. Per evitare che questo potesse avvenire sono stati realizzati numerosi controlli di intersezione geometrica e si è resa necessaria una fase intensiva di ingegnerizzazione del codice.

La mesh ottenuta è riportata in figura 5 e, come si può facilmente osservare, non è di eccezionale qualità in termini di descrizione geometrica del sistema⁵. Questo aspetto qualitativo non è l'unico che viene impiegato per valutare la bontà di una mesh, un secondo metodo consiste nel calcolare un indice di bontà di ciascun tetraedro e visualizzare un istogramma in cui vengono conteggiati i tetraedri che appartengono a determinate fasce di qualità. L'indice con cui sono stati valutati i tetraedri nel nostro caso è così ottenuto: di ogni tetraedro si calcola il raggio circoscritto ρ e volume V . Quindi si calcola il volume V_{opt} di un tetraedro equilatero avente lo stesso raggio mediante la formula $V_{opt} = \frac{8\rho^3}{9\sqrt{3}}$ ed infine si considera il seguente parametro adimensionale: $\alpha = \frac{V_{opt}-V}{V_{opt}}$. Quando il tetraedro è simile ad un tetraedro equilatero si ha $\alpha \simeq 0$ mentre se è

⁵Ad esempio gli infissi hanno profili molto irregolari e spessore pari a quello dei muri

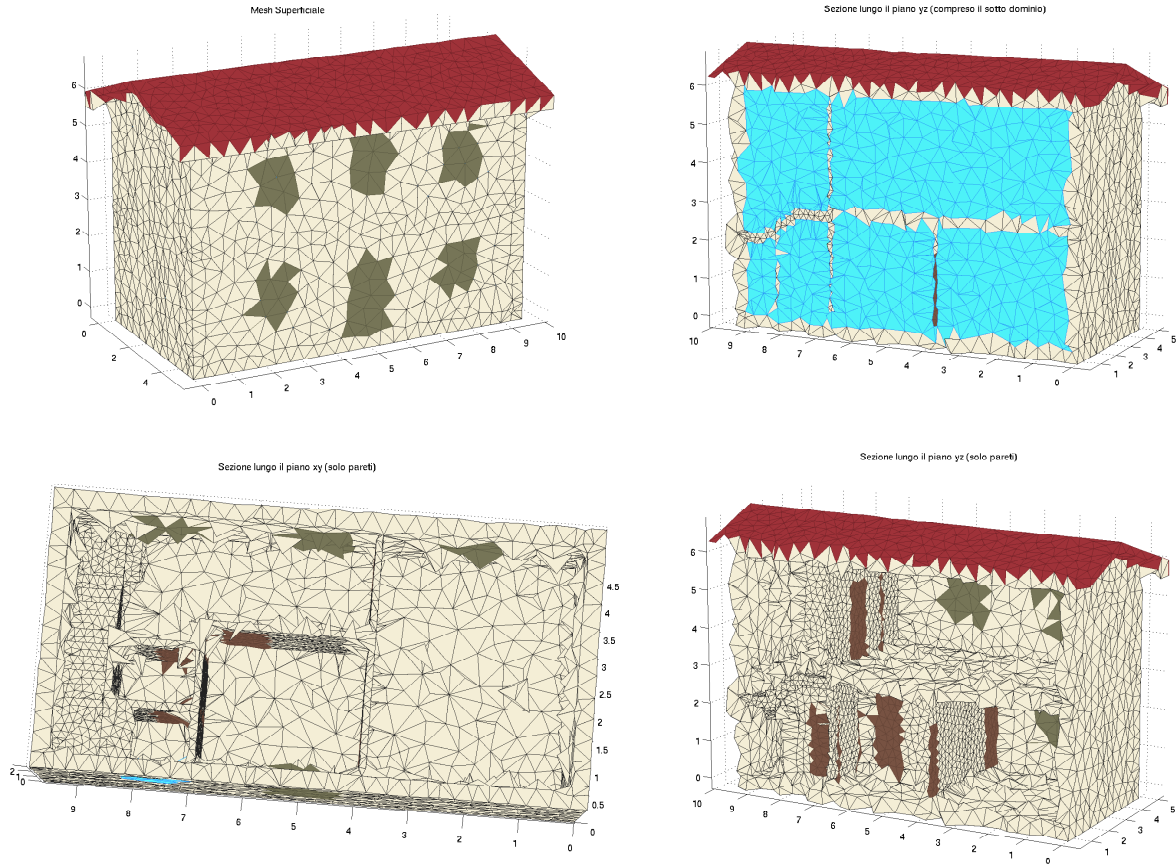


Fig. 5. Mesh a sotto domini realizzata per il simulatore. Numero di nodi 10322, Numero di tetraedri 58.368, numero di nodi quando i nodi di superficie tra aria e pareti vengono raddoppiati per gestire lo scambio per convezione: 16288

vicino ad essere degenerare, il valore di α sarà prossimo ad 1. I risultati riguardo al calcolo dell'indice di qualità della mesh sono riportati in figura 4, come si può notare la mesh contiene anche tetraedri con indice di qualità piuttosto basso.

B. La formulazione delle equazioni

Prima di capire come vengono costruite le equazioni che governano il sistema simulato, è opportuno capire come sono rappresentate le possibili soluzioni delle equazioni differenziali. Questa è una scelta che viene fatta a priori e che può influenzare le prestazioni dell'algoritmo sia in termini di complessità di calcolo che di qualità della soluzione. Nel nostro caso si è scelto di supporre le soluzioni lineari all'interno di ciascun elemento⁶ il che significa che la temperatura all'interno di un elemento è data da

$$T = \alpha + \beta x + \gamma y + \delta z \quad (6)$$

⁶Per lo studio di certi problemi è opportuno scegliere un'espressione più flessibile per la grandezza in esame. Ad esempio una possibile scelta è quella di utilizzare un polinomio di secondo grado che ha un numero maggiore di gradi di libertà. Lo svantaggio sta nel fatto che è necessario aggiungere altri nodi a metà dei lati per poter dire univocamente che valore assume la temperatura all'interno dell'elemento e questo comporta una maggiore complessità di calcolo

Se indichiamo con T_a T_b T_c e T_d le temperature dei vertici del generico elemento e con (x_a, y_a, z_a) , (x_b, y_b, z_b) , (x_c, y_c, z_c) , (x_d, y_d, z_d) le loro coordinate spaziali, è possibile imporre che valga anche per essi la relazione 6 ottenendo il seguente sistema di equazioni:

$$\begin{bmatrix} T_a \\ T_b \\ T_c \\ T_d \end{bmatrix} = \begin{bmatrix} 1 & x_a & y_a & z_a \\ 1 & x_b & y_b & z_b \\ 1 & x_c & y_c & z_c \\ 1 & x_d & y_d & z_d \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} \quad (7)$$

il quale ammette un'unica soluzione se i vertici del tetraedro non sono complanari. Pertanto i coefficienti α β γ e δ sono combinazioni lineari delle temperature dei quattro vertici. Una volta sostituiti i valori di questi parametri in funzione delle temperature sui nodi nella relazione 6 ed aver riordinato i termini, si perviene ad un risultato piuttosto ovvio: la temperatura in ogni punto interno all'elemento è una combinazione lineare convessa delle temperature dei quattro nodi e può quindi scriversi come

$$T = N_a(x, y, z)T_a + N_b(x, y, z)T_b + N_c(x, y, z)T_c + N_d(x, y, z)T_d$$

in cui le funzioni N_a , N_b , N_c ed N_d , che vengono chiamate

funzioni di forma, sono ancora lineari nelle tre coordinate spaziali ed i loro coefficienti possono essere agevolmente calcolati a partire dalle coordinate dei vertici dell'elemento.

Una volta capito come sono fatte le soluzioni ammissibili per questa tecnica di simulazione esaminiamo la strategia adottata per determinare quella che meglio approssima la soluzione dell'equazione differenziale. La tecnica qui utilizzata prende il nome di metodo di Galerkin e prevede di imporre che il residuo dell'equazione differenziale calcolata sulla nostra funzione approssimante abbia integrale esteso a tutto il volume nullo. In formule se si indica con \bar{T} la soluzione approssimata e con \mathbf{L} l'operatore che definisce l'equazione differenziale $\mathbf{L}(T) = 0$, allora la condizione da imporre è:

$$\int_{\Omega} \mathbf{L}(\bar{T}) = 0.$$

Dovendo poi determinare n diversi parametri (le temperature agli n nodi) il metodo di Galerkin prevede di imporre l'annullamento di questo integrale quando il residuo viene moltiplicato per le n funzioni di forma relative a ciascun nodo ossia le n temperature incognite si trovano risolvendo il sistema di equazioni:

$$\int_{\Omega} N_k \mathbf{L}(\bar{T}) d\Omega = 0. \quad \forall k = 1, \dots, n \quad (8)$$

L'espressione per \mathbf{L} utilizzata è stata quella determinata in 4 in cui si è posto $G = 0$ in accordo col fatto che in fase sperimentale l'edificio in esame è stato sollecitato solo dall'esterno per convezione ed irraggiamento. Pertanto si è trascurato del tutto il trasferimento di calore per convezione dell'aria interna all'edificio.

Sostituendo ora in 8 la 4 si trova⁷:

$$\int_{\Omega} N_k \left[\nabla \cdot \left(k_x \frac{\partial T}{\partial x}, k_y \frac{\partial T}{\partial y}, k_z \frac{\partial T}{\partial z} \right) - \rho c \frac{\partial T}{\partial t} \right] d\Omega = 0. \quad \forall k \quad (9)$$

Ricordando ora l'identità vettoriale:

$$f \nabla \cdot A = \nabla \cdot (fA) - A \cdot (\nabla f),$$

la 9 diviene:

$$\begin{aligned} & \int_{\Omega} \nabla \cdot \left(N_k k_x \frac{\partial T}{\partial x}, N_k k_y \frac{\partial T}{\partial y}, N_k k_z \frac{\partial T}{\partial z} \right) d\Omega + \\ & - \int_{\Omega} \left[\frac{\partial N_k}{\partial x} k_x \frac{\partial T}{\partial x} + \frac{\partial N_k}{\partial y} k_y \frac{\partial T}{\partial y} + \frac{\partial N_k}{\partial z} k_z \frac{\partial T}{\partial z} \right] d\Omega + \\ & - \int_{\Omega} \rho c \frac{\partial T}{\partial t} d\Omega = 0. \quad \forall k \end{aligned}$$

ed applicando infine il teorema di Gauss si trova

$$\begin{aligned} & \int_{\partial\Omega} n \cdot \left(N_k k_x \frac{\partial T}{\partial x}, N_k k_y \frac{\partial T}{\partial y}, N_k k_z \frac{\partial T}{\partial z} \right) dS + \\ & - \int_{\Omega} \left[\frac{\partial N_k}{\partial x} k_x \frac{\partial T}{\partial x} + \frac{\partial N_k}{\partial y} k_y \frac{\partial T}{\partial y} + \frac{\partial N_k}{\partial z} k_z \frac{\partial T}{\partial z} \right] d\Omega + \\ & - \int_{\Omega} \rho c \frac{\partial T}{\partial t} d\Omega = 0. \quad \forall k \end{aligned}$$

⁷Per non appesantire troppo la notazione, d'ora in poi indicheremo con T la soluzione approssimante

L'integrale appena scritto può essere calcolato su ogni elemento sommando poi tutti i contributi (operazione di assemblamento) il vantaggio è che, all'interno di ogni elemento, la temperatura si può esprimere come combinazione delle incognite cercate che possono essere portate fuori dal simbolo di integrale. Inoltre il primo termine rappresenta lo scambio di calore che ogni elemento ha con gli adiacenti. Ricordando che, all'interno di un elemento generico, si ha $T = \sum_{j=1}^4 N_j T_j$ i termini della relazione precedente divengono:

$$\begin{aligned} & - \sum_{j=1}^4 \int_{\partial\Omega_i} h N_k (N_j T_j - T_e) dS_i - \sum_{j=1}^4 \int_{\partial\Omega_i} q N_k dS_i \\ & - \sum_{j=1}^4 \int_{\Omega_i} \left[\frac{\partial N_k}{\partial x} k_x \frac{\partial N_j}{\partial x} + \frac{\partial N_k}{\partial y} k_y \frac{\partial N_j}{\partial y} + \frac{\partial N_k}{\partial z} k_z \frac{\partial N_j}{\partial z} \right] T_j d\Omega_i \\ & - \sum_{j=1}^4 \int_{\Omega_i} \rho c N_j \frac{\partial T_j}{\partial t} d\Omega_i \end{aligned}$$

Ovviamente per i tetraedri interni i primi due termini sono sempre nulli perché gli elementi adiacenti hanno le facce alla stessa temperatura ed inoltre si è ipotizzato che non possano esserci scambi di calore per irraggiamento all'interno dell'edificio. I primi due termini saranno quindi non nulli solo sui tetraedri superficiali dove T_e e q rappresentano le condizioni al contorno dell'intero edificio ossia gli ingressi del sistema.

C. L'integrazione delle eq differenziali

La procedura di assemblamento descritta conduce alla formulazione del seguente sistema di equazioni differenziali lineari a tempo continuo nelle incognite T_i :

$$C\dot{T} = KT + Fu$$

L'integrazione numerica di tale sistema non è affatto banale. La maggior difficoltà incontrata è dovuta alla necessità di invertire la matrice C di numeri reali di dimensione 16288×16288 (in precisione double). Se dal punto di vista computazionale il calcolo su un moderno pc non richiede tempi superiori al minuto un problema notevole si è avuto in termini di quantità di memoria necessaria. Difatti sulla macchina di simulazione, dotata di 2gb di memoria ram, non era possibile eseguire il calcolo senza dover ricorrere allo swap su disco fisso (la dimensione stessa della matrice supera la memoria disponibile) il che innalzava i tempi in maniera notevole ed imprevedibile. Si è pertanto ricorsi ad un famoso lemma di inversione di matrici quadrate basato sul complemento di Schur di una matrice (vedi Appendice A su [6]).

Nel lemma di inversione a blocchi sia ha che, se A e D sono matrici quadrate ed A è invertibile, vale

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -A^{-1}B \\ I \end{bmatrix} [D - CA^{-1}B]^{-1} [-CA^{-1}I]$$

posto che esista $(D - CA^{-1}B)^{-1}$ (complemento di Schur di A). Si nota facilmente che in tale maniera ci si riduce a dover invertire matrici di dimensione inferiore rispetto a quella originale, al costo di dover effettuare più inversioni in luogo di una. Scegliendo in tale modo un'opportuna partizione della matrice di sistema da invertire ed applicando il lemma si è riusciti ad effettuare il calcolo originale con tempi di computazione eccellenti nonostante i limiti del computer in uso. Lo stesso trucco è stato adottato anche per discretizzare il sistema utilizzando la formula di Tustin (impensabile effettuare discretizzazione esatta).

Inoltre un'altra parte che si è dovuto sviluppare per poter eseguire la simulazione è la generazione degli ingressi. Il modello richiede infatti per ogni tetraedro appartenente alla superficie esterna dell'edificio la definizione di due ingressi, temperatura ed irraggiamento. Questi ingressi dipendono fortemente dal periodo dell'anno in cui si vuole simulare il sistema fisico e per tale ragione si è sviluppata una procedura sofisticata per rendere gli ingressi quanto più possibile rispondenti alle condizioni reali. La simulazione è infatti stata portata a termine su un tempo di dieci giorni esattamente corrispondente al periodo di raccolta dati (solo sfasato di un anno per i motivi che vedremo).

Per quanto riguarda la temperatura si è scelto di sfruttare alcuni database pubblicamente disponibili e liberamente accessibili con una semplice interfaccia http. Si è scelto un punto di raccolta dati nelle vicinanze di Padova e si è automatizzata la procedura di lettura dal database, integrando tramite interpolazione spline eventuali campioni mancanti dalla lista di temperature. Per avere a disposizione un periodo dell'anno perfettamente confrontabile si è scelto di recuperare i dati relativi all'anno precedente rispetto al lavoro (2006). Si è a posteriori verificato come gli andamenti ottenuti in fase di acquisizione siano sostanzialmente simili a quelli impiegati per la simulazione.

Più complesso è stato il caso delle misure di irraggiamento: a partire dall'informazione sulla posizione del sole (azimuth e declinazione) ottenute in funzione di latitudine e longitudine del luogo e dal periodo dell'anno tramite una funzione Matlab per il calcolo delle effemeridi basata su [19] ed espressamente pensata per applicazioni di simulazione dipendenti dall'incidenza della radiazione solare si è ottenuto, per ogni istante di simulazione, il vettore che punta verso il sole. Il fattore di irraggiamento per ogni singolo tetraedro (misura che dipende dall'incidenza di un flusso su di una superficie) è poi stato posto uguale al coseno tra tale vettore e la normale al tetraedro. Si è ovviamente tenuto conto di albe e tramonti (di notte non c'è irraggiamento) ma anche della possibile ombra presente sul tetraedro esterno, includendo anche le ombre generate da alcuni alberi posti nelle vicinanze. L'algoritmo per tale calcolo è stato sviluppato in proprio e si basa su semplici operazioni geometriche sui baricentri della mesh esterna ed il vettore che punta al sole. I risultati sono molto soddisfacenti ed il tempo di calcolo minimo (rispetto alla simulazione vera e propria). L'informazione sull'angolo di incidenza è stata poi impiegata moltiplicandola per le opportune costanti di irraggiamento e per l'area della faccia del tetraedro interessata. Si è infine creato un processo stocastico

che a partire da rumore bianco genera un andamento plausibile per la copertura nuvolosa del cielo: a posteriori si è verificato come tale informazione rispecchi abbastanza fedelmente i dati di luminosità acquisiti.

Alcuni risultati della simulazione numerica sono presentati in sezione VIII e nelle figure 25, 26 e 27.

IV. METODI DI IDENTIFICAZIONE SUBSPACE

I metodi di identificazione di tipo a sottospazio sono una classe di algoritmi numerici per l'identificazione di sistemi dinamici lineari a tempo discreto di dimensione finita e tempo invarianti con ingressi stocastici e deterministici

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= Cx(t) + Du(t) + v(t) \end{cases} \quad (10)$$

con $w(t)$ e $v(t)$ rumori bianchi stazionari e a media nulla chiamati rispettivamente di modello e di misura e tali che

$$E \left[\begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \begin{bmatrix} w(t)^T & v(t)^T \end{bmatrix} \right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{ts} \geq 0. \quad (11)$$

I vettori sono tali che $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ e $y \in \mathbb{R}^l$. Le matrici di covarianza sono semidefinite positive mentre le proprietà strutturali del sistema dinamico $\Sigma = (A, [BQ^{1/2}], C, D)$ sono la controllabilità della coppia $A, [BQ^{1/2}]$ (tutti i modi del sistema sono eccitati da un ingresso stocastico o deterministico) e l'osservabilità della coppia A, C (tutti i modi sono osservabili in y e quindi possono essere identificati). Il campo di utilizzo principale di metodi subspace risulta essere l'identificazione di sistemi MIMO di grandi dimensioni, dove metodi del tipo a minimizzazione di errore sarebbero impraticabili dal punto di vista pratico a causa della quantità di calcoli in gioco.

Il problema matematico che si propone di risolvere l'identificazione è di trovare un ordine n e le matrici A, B, C, D, Q, R, S dato un insieme di s ingressi u_0, \dots, u_{s-1} ed uscite y_0, \dots, y_{s-1} . I metodi di subspace identification permettono di ricavare gli stati del sistema a partire dai dati di ingresso ed uscita attraverso opportune proiezioni ortogonali o oblique degli sottospazi generati dalle righe di particolari matrici nei sottospazi di altre matrici analoghe. Nello specifico si tratta di matrici di Hankel a blocchi in cui sono organizzati in particolare maniera i dati di ingresso ed uscita del sistema Σ . A partire dalla proiezione attraverso una decomposizione ai valori singolari si ottiene una stima dell'ordine del sistema, la matrice di osservabilità (estesa) e opzionalmente la sequenza degli stati del sistema. Infine le matrici del sistema A, B, C, D sono ottenute tramite la risoluzione esatta oppure ai minimi quadrati (se sono presenti ingressi stocastici) di un opportuno sistema lineare; nel secondo caso dai residui si ricavano a loro volta le matrici di covarianza Q, S, R e di conseguenza gli ingressi stocastici del sistema.

Tutti i passaggi coinvolti possono essere eseguiti al calcolatore impiegando gli strumenti dell'algebra lineare (decomposizione QR e SVD) dei quali esistono versioni computazionalmente molto efficienti. Inoltre non è richiesto l'impiego di algoritmi di minimizzazione iterativi ed in tale modo non si presentano problemi di convergenza come può accadere con i metodi PEM; inoltre sfruttando la particolare struttura delle matrici

coinvolte gli algoritmi numerici sono ottimizzabili ed eseguibili su un normale calcolatore in tempi ragionevoli anche per sistemi di dimensione elevata (n, m, l) ed in presenza di numerosi campioni $(s \gg 1)$. Esistono numerose varianti nell'implementazione di questi metodi e molte di esse, come accennato, non richiedono ad esempio il calcolo esplicito degli stati del sistema oppure la risoluzione completa di alcune decomposizioni.

Nel seguito si è scelto di presentare seppur in maniera introduttiva sia una versione dell'identificazione a sottospazi del problema deterministico sia una versione dell'identificazione mista deterministica/stocastica. Nel primo caso si è tentato di affrontare il problema esponendo alcuni risultati teorici e delineando alcune operazioni comuni nell'identificazione di sistemi deterministici, stocastici e misti. Nel secondo caso si è prediletta un'impostazione pratica, tralasciando i risultati teorici necessari per focalizzarsi sulla parte algoritmica. Il secondo caso include alcuni tra gli algoritmi più noti (N4SID, MOESP e CVA) e che sono stati impiegati per l'identificazione dei dati. Per entrambi i casi si rimanda a [1] e a [2] per un'esposizione rigorosa dei risultati.

È stata utilizzato anche l'algoritmo di identificazione subspace PBSID_{opt} di più recente introduzione (vedi [3]) di cui si è sfruttata un'implementazione software di tipo ricorsiva gentilmente concessa dal prof. Alessandro Chiuso. Con tale algoritmo si sono ottenuti i risultati più soddisfacenti e si è rivelato anche il migliore dal profilo della quantità di calcoli richiesti.

A. Alcune notazioni impiegate

Le notazioni da introdurre riguardano la creazione delle matrici di Hankel a blocchi e delle matrici derivanti dal modello nello spazio degli stati.

1) *Matrici di Hankel a blocchi*: Supponendo di disporre di s misure di ingressi ed uscite, a partire dal vettore degli ingressi $\mathbf{u} = u_0, \dots, u_{s-1}$ con il generico $u_t \in \mathbb{R}^{m \times 1}$ si pone

$$U_{i|k} = \begin{bmatrix} u_i & u_{i+1} & u_{i+2} & \dots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \dots & u_{i+j} \\ u_{i+2} & u_{i+3} & u_{i+4} & \dots & u_{i+j+1} \\ u_{i+2} & u_{i+3} & u_{i+4} & \dots & u_{i+j+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_k & u_{k+1} & u_{k+2} & \dots & u_{k+j-1} \end{bmatrix} \quad (12)$$

Si tratta di una matrice di Hankel con diagonali a pendenza positiva costanti a blocchi; si ha che $U_{i|k} \in \mathbb{R}^{m(k-i) \times j}$. Da questa si definiscono

$$U_{0|2i-1} = \begin{bmatrix} U_p \\ U_f \end{bmatrix} = \begin{bmatrix} U_{0|i-1} \\ U_{i|2i-1} \end{bmatrix} \quad (13)$$

$$= \begin{bmatrix} U_p^+ \\ U_f^- \end{bmatrix} = \begin{bmatrix} U_{0|i} \\ U_{i+1|2i-1} \end{bmatrix} \quad (14)$$

Le dimensioni delle matrici coinvolte dipendono, oltre che da m , da i (parametro impostabile e solitamente tale per cui $i \geq n$) e da j (solitamente dipendente dalla quantità di campioni disponibili e quindi posto alla lunghezza massima possibile, $j = s - 2i + q$). Si definiscono con ingressi passati le matrici U_p e U_p^+ e con ingressi futuri le matrici U_f e U_f^- ; peraltro si tratta di una nomenclatura non precisa (ad esempio U_p e U_f

condividono diversi elementi).

In maniera analoga si definiscono le matrici $Y_{0|2i-1}, Y_p, Y_f, Y_p^+$ e Y_f^- e si introducono le matrici

$$W_{0|i-1} = W_p = \begin{bmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{bmatrix} = \begin{bmatrix} U_p \\ Y_p \end{bmatrix} \quad (15)$$

$$W_p^+ = \begin{bmatrix} U_p^+ \\ Y_p^+ \end{bmatrix}. \quad (16)$$

ed in maniera similare W_f e W_f^- . Infine la sequenza degli stati del sistema viene indicata con

$$X_i = [x_i \ x_{i+1} \ \dots \ x_{i+j-2} \ x_{i+j-1}] \in \mathbb{R}^{n \times j} \quad (17)$$

e si introducono i due simboli $X_p = X_0$ e $X_f = X_i$.

2) *Matrici dipendenti dal sistema* Σ : La matrice di osservabilità estesa è definita da

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix} \quad (18)$$

con $i > n$; essa ha rango (colonna) n pieno essendo Σ osservabile. La matrice invertita di controllabilità estesa è data da

$$\Delta_i^r = [A^{i-1}B \ \dots \ A^2B \ AB \ B]. \quad (19)$$

la quale ha rango (riga) n pieno essendo Σ controllabile. Si pone inoltre

$$H_i = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{i-2}B & CA^{i-3}B & D & \dots & D \end{bmatrix} \quad (20)$$

con $H_i \in \mathbb{R}^{il \times im}$ matrice di triangolare inferiore a blocchi di Toeplitz con i parametri di Markov del sistema.

B. Introduzione ad alcune operazioni geometriche impiegate

Nel seguito si vogliono introdurre alcuni strumenti propri dell'algebra lineare illustrando alcuni metodi numerici utili a risolvere efficacemente i problemi posti. Non verranno proposte dimostrazioni ma solamente e definizioni principali e il necessario per comprendere i risvolti delle operazioni compiute.

Si assume di avere tre generiche matrici $A \in \mathbb{R}^{p \times j}$, $B \in \mathbb{R}^{q \times j}$ e $C \in \mathbb{R}^{r \times j}$ con $j \geq \max(p, q, r)$; le operazioni riguardano gli spazi generati dalle righe delle suddette matrici.

1) *Proiezioni ortogonali*: La proiezione ortogonale dello spazio generato dalle righe di A nello spazio generato dalle righe di B è definito come A/B ; la rappresentazione matriciale è ⁸

$$A/B = AB^T(BB^T)^\dagger B = A\Pi_B \quad (21)$$

⁸La pseudoinversa di Moore-Penrose è indicata con [†]

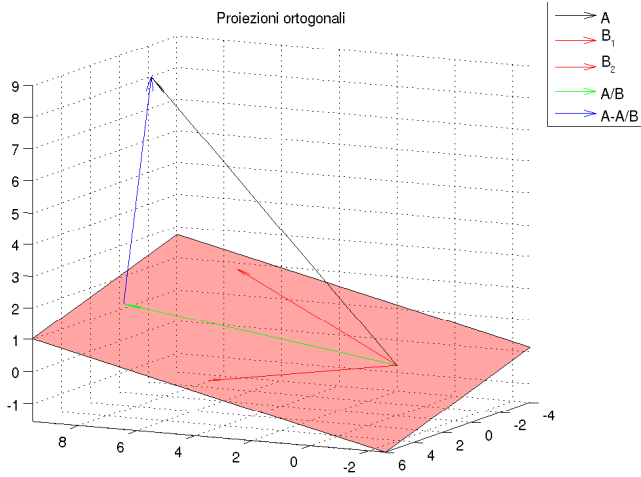


Fig. 6. Proiezione ortogonale del sottospazio generato dalle righe di A nel sottospazio generato dalle righe di B . In questo caso $A \in \mathbb{R}^{1 \times 3}$ (il sottospazio coincide con la retta individuata dal vettore A , di colore nero) e $B \in \mathbb{R}^{2 \times 3}$ (il sottospazio coincide con il piano di colore rosso trasparente individuato dai due vettori di B indicati in rosso). Lo spazio A/B è dato dalla retta individuata dal vettore appartenente al piano e proiezione di A (di colore verde) mentre lo spazio ad esso ortogonale e proiezione di A è individuato dal vettore in blu. In questo caso L_{21} è un vettore che combina linearmente le colonne di Q_1 che a loro volta formano lo spazio riga di B mentre L_{22} è uno scalare.

e vale $A/B \in \langle B \rangle$ mentre la proiezione del sottospazio generato dalle righe di A nel sottospazio ortogonale dello spazio righe di B è dato da

$$A/B^\perp = A - A\Pi_B = A(I - B^T(BB^T)^\dagger B) = A\Pi_{B^\perp}. \quad (22)$$

Le operazioni permettono quindi di scrivere

$$A = A\Pi_B + A\Pi_{B^\perp} \quad (23)$$

e quindi di decomporre il sottospazio righe di A in due matrici i cui sottospazi riga risultano ortogonali tra loro (vale $\Pi_B \Pi_{B^\perp} = I$).

Utilizzando la decomposizione LQ⁹ è possibile scrivere

$$\begin{bmatrix} B \\ A \end{bmatrix} = LQ^T = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \quad (24)$$

con

$$\begin{aligned} L_{11} &\in \mathbb{R}^{q \times q} & L_{21} &\in \mathbb{R}^{p \times q} & L_{22} &\in \mathbb{R}^{p \times p} \\ Q_1 &\in \mathbb{R}^{j \times q} & Q_2 &\in \mathbb{R}^{j \times p} \end{aligned}$$

ed in tal modo è possibile ottenere

$$A/B = L_{21}Q_1^T \quad (25)$$

$$A/B^\perp = L_{22}Q_2^T. \quad (26)$$

⁹Le proprietà fondamentali sono, posto $M = LQ^T$ con $M \in \mathbb{R}^{n \times m}$ matrice generica non necessariamente quadrata, che $L \in \mathbb{R}^{n \times m}$ è triangolare inferiore e che $Q \in \mathbb{R}^{m \times m}$ è tale per cui $QQ^T = I_m$. A causa del simbolo comune si è preferito mantenere Q sia per la decomposizione che per la varianza di modello per cui anche nel seguito verranno impiegati entrambi, comunque in contesti diversi facilmente riconoscibili.

Un esempio nello spazio tridimensionale ($j = 3$) calcolato tramite ambiente Matlab impiegando tale algoritmo¹⁰ è riportato in figura 6.

2) *Proiezioni oblique*: Il sottospazio righe di A può essere scomposto come illustrato prima in due sottospazi tra loro ortogonali ma anche in due sottospazi tra loro non ortogonali e nel residuo appartenente al complemento ortogonale dello spazio formato congiuntamente dalle righe di B e C . Simbolicamente si scompone

$$A = L_B B + L_C C + L_{[BC]^\perp} \begin{bmatrix} B \\ C \end{bmatrix}^\perp \quad (27)$$

e si definisce $L_B B$ come la proiezione obliqua dello spazio righe di A lungo lo spazio righe di C nello spazio righe di B

$$A/CB = L_B B \quad (28)$$

e di conseguenza $A/CB \in \langle B \rangle$.

L'interpretazione geometrica è proiettare lo spazio righe di A nello spazio righe congiunto di B e C per poi decomporre il risultato lungo lo spazio riga di B (A/CB) e lungo lo spazio riga di C (A/BC). Da questo si può ottenere un metodo per la proiezioni oblique utilizzando ancora la scomposizione LQ; si usa

$$\begin{bmatrix} C \\ B \\ A \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{bmatrix} \quad (29)$$

con

$$\begin{aligned} L_{11} &\in \mathbb{R}^{r \times r} & L_{21} &\in \mathbb{R}^{r \times q} & L_{22} &\in \mathbb{R}^{q \times q} \\ L_{31} &\in \mathbb{R}^{r \times p} & L_{32} &\in \mathbb{R}^{q \times p} & L_{33} &\in \mathbb{R}^{p \times p} \\ Q_1 &\in \mathbb{R}^{j \times r} & Q_2 &\in \mathbb{R}^{j \times q} & Q_3 &\in \mathbb{R}^{j \times p}. \end{aligned}$$

La proiezione ortogonale di A sullo spazio congiunto di B e C è

$$A/ \begin{bmatrix} C \\ B \end{bmatrix} = [L_{31} \quad L_{32}] \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$$

ma anche come combinazione lineare di B e C

$$A/ \begin{bmatrix} C \\ B \end{bmatrix} = [L_C \quad L_B] \begin{bmatrix} C \\ B \end{bmatrix} = [L_C \quad L_B] \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$$

da cui

$$[L_C \quad L_B] = [L_{31} \quad L_{32}] \begin{bmatrix} L_{11}^{-1} & 0 \\ * & L_{22}^{-1} \end{bmatrix}$$

ed infine

$$A/CB = L_B B = L_{32} L_{22}^{-1} B \quad (30)$$

notando come non sia necessario il calcolo esplicito della matrice Q e come l'inversa di L_{22} sia agilmente calcolabile in quanto è una matrice diagonale inferiore.

Due proprietà delle proiezioni oblique sono

$$A/AB = 0$$

$$A/BA = A$$

¹⁰È disponibile una routine per la decomposizione QR: per ottenere una decomposizione di tipo LQ della matrice M sono necessari i comandi $[Q, R] = \text{qr}(M')$, $L = R'$ che portano all'identità $M = L * Q'$.

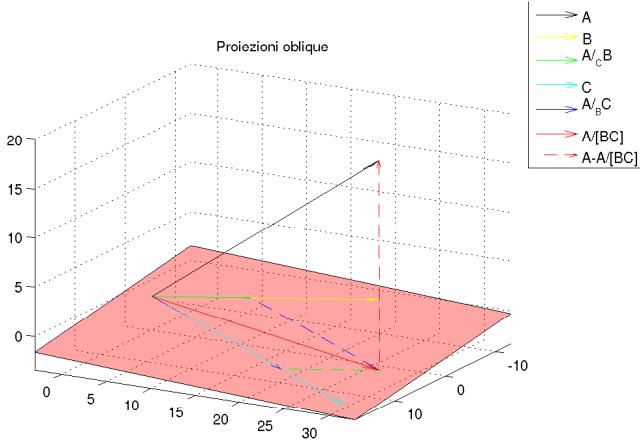


Fig. 7. La proiezione obliqua dello spazio vettoriale formato dalle righe di A (è la retta individuata dal vettore nero) lungo lo spazio riga di C sullo spazio vettoriale formato dalle righe di B (la retta individuata dal vettore giallo) è il vettore A/CB (in verde). In questo caso L_B è uno scalare. Analogamente la proiezione di A lungo B nello spazio delle righe di C (A/B_C) è indicata in blu (in azzurro il vettore C); L_C è uno scalare. Dalla figura si evince come la proiezione di A sullo spazio formato congiuntamente da B e C ($A/[BC]$, in rosso) è data dalla combinazione lineare $L_B B + L_C C$.

e da esse è possibile definire l'operazione stessa di proiezione obliqua: in tal modo si ricava una definizione rigorosa e consistente di proiezione obliqua tra matrici che è

$$A/CB = [A/C^\perp][B/C^\perp]^\dagger B. \quad (31)$$

C. Identificazione di un sistema deterministico

Se il sistema non è soggetto a disturbi di tipo stocastico (cioè si ha $w_t = 0$ e $v_t = 0$, $t \in [0, s-1]$) un possibile approccio consiste nell'effettuare dapprima una proiezione obliqua delle matrici di Hankel introdotte precedentemente e poi una decomposizione ai valori singolari del risultato per ottenere gli stati del sistema (senza conoscere ovviamente le matrici che compongono Σ). Dagli stati, dagli ingressi e dalle uscite è possibile poi impostare un sistema lineare per ricavare le matrici di Σ ed identificare il sistema.

1) *Recupero degli stati e della matrice di osservabilità estesa*: Si dimostra (è sufficiente sostituire con le equazioni di stato 10) che, utilizzando la notazione introdotta in IV-A, valgono le identità

$$Y_p = \Gamma_i X_p + H_i U_p \quad (32)$$

$$Y_f = \Gamma_i X_f + H_i U_f \quad (33)$$

$$X_f = A^i X_p + \Delta_i^r U_p \quad (34)$$

Un teorema permette di affermare che è possibile ricostruire la sequenza degli stati X_f e la matrice estesa di osservabilità Γ_i direttamente dai dati u_t e y_t . Sotto le ipotesi di avere

- l'ingresso u_t persistentemente eccitante di ordine $2i$ e quindi di avere $\text{rank } R^{uu} \geq 2mi$ (vedi [5] per la definizione di R^{uu}) con i pari almeno all'ordine del

sistema da identificare (la matrice a blocchi di Hankel $U_{0|2i-1}$ è di rango riga pieno)

- l'intersezione dello spazio righe di U_f con lo spazio righe di X_p è vuota
 - esistono due matrici di pesi arbitrarie, $W_1 \in \mathbb{R}^{li \times li}$ di rango pieno e $W_2 \in \mathbb{R}^{j \times j}$ tale per cui $\text{rank } W_p = \text{rank } W_p W_2$ che determinano la base per lo spazio degli stati
 - $j, s \rightarrow \infty$
- e definendo

$$\mathcal{O}_i = Y_f / U_f W_p \quad (35)$$

$$W_1 \mathcal{O}_i W_2 = [U_1 \ U_2] \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S_1 V_1^T \quad (36)$$

valgono le seguenti proposizioni

- vale la relazione $\mathcal{O}_i = \Gamma_i X_f$ cioè che la matrice ottenuta in 35 è il prodotto della matrice di osservabilità estesa per la matrice degli ingressi futuri (ed è possibile ottenere il vettore degli ingressi $X_f = \Gamma_i^\dagger \mathcal{O}_i$)
- l'ordine di Σ è pari alla dimensione di S_1
- se $T \in \mathbb{R}^{n \times n}$ e $TT^{-1} = I_n$ (generica matrice di cambio di base) la matrice di osservabilità estesa è pari a

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T \quad (37)$$

e la parte della sequenza di stato che appartiene allo spazio colonna di W_2 si ricava da

$$X_f W_2 = T^{-1} S_1^{1/2} V_1^T. \quad (38)$$

Un accenno della prova del teorema si ha notando che da 32 si ottiene

$$X_p = \Gamma_i^\dagger Y_p - \Gamma_i^\dagger H_i U_p$$

che sostituita in 34 porge

$$X_f = [\Delta_i^r - \Gamma_i^\dagger H_i] U_p + A^i \Gamma_i^\dagger Y_p = L_p W_p$$

con $L_p = [\Delta_i^r - \Gamma_i^\dagger H_i \quad A^i \Gamma_i^\dagger]$ da cui 33 diventa

$$Y_f = \Gamma_i L_p W_p + H_i U_f \quad (39)$$

che mostra come gli ingressi futuri siano combinazione di due spazi righe (W_p e U_f) non ortogonali e la cui interpretazione geometrica è riportata figura 8. Effettuando la proiezione sul complemento ortogonale di U_f si ha

$$Y_f / U_f^\perp = \Gamma_i L_p W_p / U_f^\perp$$

e post-moltiplicando per $[W_p / U_f^\perp]^\dagger W_p$ e ricordando la definizione di proiezione obliqua 31 si ottiene

$$\mathcal{O}_i = Y_f / U_f W_p = [Y_f / U_f^\perp] [W_p / U_f^\perp]^\dagger W_p = \Gamma_i L_p W_p$$

Si è sfruttato il fatto che $(W_p / U_f^\perp) (W_p / U_f^\perp)^\dagger W_p = W_p$, dimostrabile grazie alle condizioni sul rango di R^{uu} e degli spazi righe congiunti di U_f e Y_p (vedi [2]) e ciò porta a concludere che $\mathcal{O}_i = \Gamma_i X_f \in \mathbb{R}^{li \times j}$.

Ora è possibile spezzare il prodotto $W_1 \mathcal{O}_i W_2 = W_1 \Gamma_i X_f W_2$ in due parti entrambe di rango pieno n grazie alle ipotesi del teorema: W_1 è di rango pieno ed il rango di Γ_i è n , mentre il prodotto $W_2 X_f$ lo è per ipotesi. In tale modo è possibile recuperare le due matrici a meno di una trasformazione di

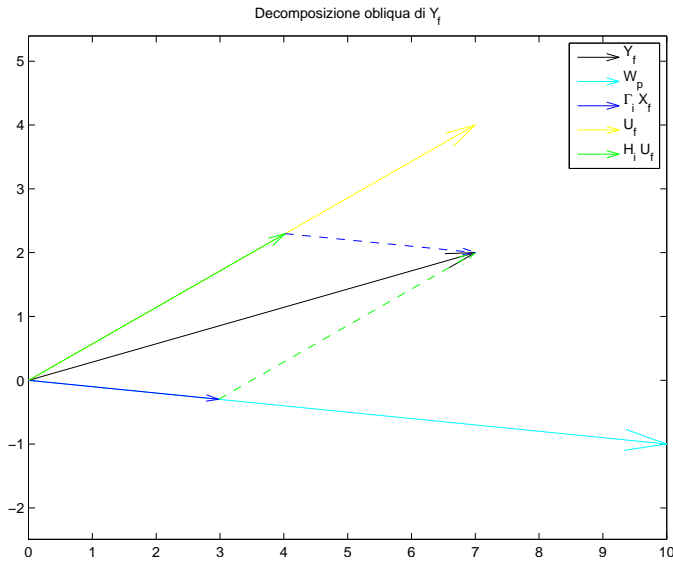


Fig. 8. Illustrazione della formula 39: lo spazio righe di Y_f (vettore nero) viene decomposto tramite una proiezione obliqua lungo lo spazio degli ingressi futuri U_f (in giallo) sullo spazio degli ingressi e delle uscite passate W_p (in azzurro). La matrice O_i è riportata in blu.

similarità T impiegando la decomposizione SVD¹¹. Si nota come essendo il sistema Σ definito a meno di una trasformazione di similarità T : sia $\tilde{\Sigma} = (T^{-1}AT, T^{-1}B, CT, D)$ sistema equivalente a Σ , si ha che la matrice di osservabilità è legata secondo $\tilde{\Gamma}_i = \Gamma_i T$ e gli stati secondo $\tilde{X}_i = T^{-1}X_i$. Vale inoltre il fatto che lo spazio righe di O_i coincide con lo spazio righe di X_i e che lo spazio colonne di O_i coincide con lo spazio colonne di Γ_i .

2) *Recupero delle matrici del sistema*: Per ottenere le matrici di Σ è possibile a partire dalle relazioni 10 e dalle definizioni in IV-A impostare il sistema lineare

$$\begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} \quad (40)$$

il quale è risolvibile in (A, B, C, D) in maniera esatta (il sistema è deterministico). Nella sezione precedente si è ricavato X_i mentre per ricavare X_{i+1} si nota come con un ragionamento analogo valga

$$O_{i+1} = Y_f^- / U_f^- W_p^+ = \Gamma_{i-1} X_{i+1}. \quad (41)$$

A questo punto, sapendo che Γ_{i-1} è una sottomatrice di Γ_i , si ottiene $X_{i+1} = \Gamma_{i-1}^\dagger O_{i+1}$.

¹¹La matrice $M \in \mathbb{R}^{l \times m}$ può essere fattorizzata con SVD in $M = U\Sigma V^T$ con U matrice $l \times l$ e V matrice $m \times m$ tali che $UU^T = I_l$ e $VV^T = I_m$ (sono ortonormali). Lo spazio colonna di M (uscite) ha U come base ortonormale, mentre lo spazio riga di M ha V come base ortonormale. I valori singolari $\sigma_i, i \in \{1, \dots, k = \min(l, m)\}$ appartengono alla diagonale della matrice $\Sigma \in \mathbb{R}^{l \times m}$ e sono tali per cui $\sigma_i \geq \sigma_{i+1}$. Sia $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_k)$ allora si ha $\Sigma = [\Sigma_1 0]^T$ se $l \geq m$ oppure $\Sigma = [\Sigma_1 0]$ se $m \geq l$. Si ha che $\sigma_i = \sqrt{\text{eig}(MM^T)} = \sqrt{\text{eig}(M^T M)}$ ed inoltre le colonne di U e di V sono, prese nell'ordine da sinistra a destra, rispettivamente i vettori singolari sinistri e destri chiamati *direzioni più importanti*.

D. Identificazione deterministica e stocastica

In letteratura sono presenti diversi algoritmi di identificazione a sottospazio per ricavare le matrici di un generico sistema Σ presentato nelle formule 10. Lo scopo è di identificare le matrici A, B, C, D, Q, R ed S e la procedura si compone come in IV-C di due passi principali: prima è necessario ottenere gli stati del sistema da ingressi ed uscite per poi impostare a partire dal risultato ottenuto un problema ai minimi quadrati dal quale ricavare le matrici sopra menzionate. La differenza fondamentale è che la sequenza degli stati ottenuta è ricavata da un banco di filtri di Kalman non stazionari con particolari varianze P_0 e condizioni iniziali x_0 . Non verranno illustrati i teoremi rilevanti ma solamente introdotte le nozioni e gli spunti più importanti.

1) *Calcolo della sequenza degli stati*: La sequenza degli stati di un sistema Σ è ottenibile a partire dalle sequenze di ingressi ed uscite seguendo un meccanismo analogo a quello presentato in IV-C1, dapprima eseguendo una proiezione ortogonale dello spazio righe degli ingressi futuri sullo spazio righe congiunto di ingressi ed uscite passate per poi effettuare una decomposizione ai valori singolari che determina l'ordine del sistema, la matrice di osservabilità e la sequenza di stati che si rivela essere ottenuta da un particolare filtro di Kalman tempo variante. Al contrario del caso deterministico in cui la matrice Y_f risiede completamente nello spazio righe dato dall'unione di ingressi ed uscite passate, ora la proiezione permette di eliminare la parte di informazione dovuta all'ingresso stocastico.

Sotto le ipotesi di avere

- rumori di modello w_t e di misura v_t scorrelati dall'ingresso u_t e non identicamente nulli
- l'ingresso u_t è persistentemente eccitante di ordine $2i$
- un numero di campioni elevato ($j \rightarrow \infty$)

si dimostra che vale per la proiezione obliqua

$$O_i = Y_f / U_f W_p = \Gamma_i \hat{X}_i \quad (42)$$

con Γ_i matrice di osservabilità estesa e \hat{X}_i una sequenza di stati di un filtro di Kalman a banche non stazionario. Se fossero note le matrici del sistema Σ si potrebbe istanziare il filtro e farlo correre lungo gli ingressi e le uscite, schematizzando per la colonna k -esima di \hat{X}_i con

$$[\hat{x}_k^0, P_0] \rightarrow [u_k, \dots, u_{k+i-1}, y_k, \dots, y_{k+i-1}] \rightarrow \hat{x}_{k+i}$$

notando come non si impieghino tutti i dati disponibili (solo i uscite ed ingressi), come esista una dipendenza esplicita dalle condizioni iniziali \hat{x}_k^0 e come sia possibile ripetere la procedura per ogni colonna, ottenendo $\hat{X}_i = \hat{X}_i(\hat{X}_i^0, P_0) = [\hat{x}_i \dots \hat{x}_{k+i} \dots \hat{x}_{i+j-1}]$ rendendo esplicita la dipendenza dalle condizioni iniziali. Un contributo importante che si incontra nella dimostrazione è che tale filtro di Kalman non stazionario può essere calcolato in maniera non ricorsiva con formule esplicite dipendenti dalle matrici di Σ . Esistono formule esplicite per ottenere \hat{X}_i^0 e P_0 . In maniera del tutto simile si ha che

$$O_{i-1} = Y_f^- / U_f^- W_p^+ = \Gamma_{i-1} \hat{X}_{i+1}. \quad (43)$$

Dal punto di vista algoritmico si può partizionare la decomposizione LQ della matrice

$$\begin{bmatrix} U_{0|i-1} \\ U_{i|i} \\ U_{i+1|2i-1} \\ Y_{0|i-1} \\ Y_{i|i} \\ Y_{i+1|2i-1} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} & 0 & 0 \\ L_{51} & L_{52} & L_{53} & L_{54} & L_{55} & 0 \\ L_{61} & L_{62} & L_{63} & L_{64} & L_{65} & L_{66} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \\ Q_5^T \\ Q_6^T \end{bmatrix}$$

Usando la scrittura

$$Y_f = Y_f / U_f W_p + L_{U_f} U_f = L_{U_p} U_p + L_{Y_p} Y_p + L_{U_f} U_f$$

e ricordando le definizioni di IV-A e il procedimento introdotto in IV-B2 si possono ottenere i combinatori come soluzione di

$$\begin{bmatrix} L_{51} & L_{61} \\ L_{52} & L_{62} \\ L_{53} & L_{63} \\ L_{54} & L_{64} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix}^T \begin{bmatrix} L_{U_p}^T \\ L_{Y_p}^T \\ L_{U_f}^T \\ L_{Y_p}^T \end{bmatrix}$$

Usando invece la scrittura

$$Y_f^- = Y_f^- / U_f^- W_p^+ + L_{U_f^-} U_f^- = L_{U_p^+} U_p^+ + L_{Y_p^+} U_p^+ + L_{U_f^-} U_f^-$$

si possono ottenere i combinatori come soluzione di

$$\begin{bmatrix} L_{61} \\ L_{62} \\ L_{63} \\ L_{64} \\ L_{65} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} & 0 \\ L_{51} & L_{52} & L_{53} & L_{54} & L_{55} \end{bmatrix}^T \begin{bmatrix} L_{U_p^+}^T \\ L_{Y_p^+}^T \\ L_{U_f^-}^T \\ L_{Y_p^+}^T \end{bmatrix}$$

Per quanto riguarda il calcolo delle sequenze tramite decomposizione ai valori singolari si ha che

$$\mathcal{O}_i = \begin{bmatrix} L_{U_p^+} & L_{Y_p^+} \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \end{bmatrix}. \quad (44)$$

Sia ora

$$\mathcal{O}_i = [U_1 \ U_2] \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S_1 V_1^T \quad (45)$$

Per le proprietà della SVD si ha che Γ_i appartiene allo spazio delle prime n colonne di U_1 che coincide con lo spazio generato da $\mathcal{O}_i \mathcal{O}_i^T$. Per le proprietà della scomposizione LQ, Q è ortogonale perciò si ha che

$$\mathcal{O}_i \mathcal{O}_i^T = \begin{bmatrix} L_{U_p} & L_{Y_p} \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} L_{U_p^+} & L_{Y_p^+} \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix}^T$$

di conseguenza si può recuperare Γ_i dalla scomposizione singolare di

$$\begin{bmatrix} L_{U_p} & L_{Y_p} \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} = U_1 S_1 \tilde{V}_1^T \quad (46)$$

dove si nota che cambia rispetto a 45 V_1 con \tilde{V}_1 (i valori singolari rimangono identici, essendo invariati gli autovalori

dato che Q è ortonormale). In definitiva si ha che la matrice di osservabilità estesa è

$$\Gamma_i = U_1 S_1^{1/2} \quad (47)$$

e le sequenze degli stati sono

$$\hat{X}_i = \Gamma_i^\dagger \mathcal{O}_i = S_1^{1/2} \tilde{V}_1 [Q_1 \ Q_2 \ Q_3 \ Q_4]^T. \quad (48)$$

$$\hat{X}_{i+1} = \Gamma_{i-1}^\dagger \mathcal{O}_{i-1} \quad (49)$$

Le colonne corrispondenti di \hat{X}_i e \hat{X}_{i+1} sono stime degli stati secondo il filtro di Kalman ottenuti in due istanti successivi ma non risultano uguali in quanto cambiano le condizioni iniziali; se gli ingressi fossero puramente stocastici, le colonne invece corrisponderebbero (si partirebbe sempre da stato iniziale nullo).

2) *Calcolo delle matrici del sistema:* Da 46 si possono ottenere le informazioni necessarie per il calcolo delle matrici del sistema: l'ordine viene dalla dimensione di S_1 (pari al numero di valori singolari diverso da zero), mentre la matrice di osservabilità estesa Γ_i e la sua sottomatrice Γ_{i-1} derivano da 47 e la sequenza degli stati del filtro di Kalman da 48 e 49. Si può quindi impostare un sistema lineare sovradeterminato

$$\begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \quad (50)$$

da risolvere ai minimi quadrati secondo la norma di Frobenius¹² tramite la relazione

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \min_{A,B,C,D} \left\| \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ U_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \right\|_F^2.$$

La soluzione è esprimibile in forma chiusa (non riportata) e nello sfruttando il fatto che la matrice Q è ortonormale e si elide nelle soluzioni è possibile evitarne completamente il calcolo (a meno di volere esplicitamente i vettori degli stati) con un notevole impatto sul fronte computazionale. Le covarianze di rumore a loro volta vengono calcolate in maniera campionaria a partire dai residui della minimizzazione, e quindi come

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \frac{1}{j} \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w^T & \rho_v^T \end{bmatrix}. \quad (51)$$

Per via del fatto che gli stati del filtro di Kalman a banchi \hat{X}_i e \hat{X}_{i+1} sono originati a partire da condizioni iniziali differenti le stime $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ sono affette in questo procedimento da un errore di bias a meno che non si verifichi almeno una tra: utilizzare un indice $i \rightarrow \infty$, avere un sistema puramente deterministico (caso illustrato in IV-C) oppure eccitare il sistema con ingressi deterministici di rumore bianco. Algoritmi che non presentano tale errore di media si trovano in letteratura. A loro volta anche le stime di $\hat{Q}, \hat{S}, \hat{R}$ presentano un errore di bias.

¹²La norma di Frobenius di una matrice $A \in \mathbb{R}^{n \times m}$ è definita da $\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2}$.

E. La variante $PBSID_{opt}$

L'algoritmo di identificazione subspace presentato precedentemente è utilizzabile per ricavare sistemi dinamici dove non sia presente feedback tra l'ingresso e le uscite (come nel caso di un modello lineare della termodinamica di un edificio in cui gli ingressi siano esogeni al sistema e comunque non influenzati dallo stesso). Un algoritmo di più recente introduzione è il PBSID (predictor based subspace identification) in cui, sfruttando differenti matrici di Hankel a blocchi ma comunque utilizzando le operazioni in due fasi introdotte precedente e cioè proiezioni oblique per ottenere una base per lo stato e la matrice di osservabilità estesa e successiva SVD per separarle seguite da un problema ai minimi quadrati per ricavare le matrici del sistema, si riesce ad operare su sistemi in cui sia presente feedback. Il problema che si presenta per gli algoritmi non espressamente per gestire ingressi correlati alle uscite è rintracciabile nel fatto che la proiezione obliqua 42 non permette di eliminare il rumore additivo stocastico in quanto esso è dipendente dagli ingressi passati (su cui viene effettuata la proiezione).

Una versione ottimizzata dell'algoritmo PBSID, $PBSID_{opt}$, è stata sviluppata in [3] ed utilizzata con successo per identificare sistemi MIMO dotati di sessanta ingressi e sessanta uscite (vedi [4]). In linea di massima rispetto alla versione non ottimizzata si ha una migliore prestazione nei sistemi con contenuto alle basse frequenze: ammettendo che il comportamento termico dell'edificio in questione sia caratterizzato da una dinamica lenta ci si aspettano modelli abbastanza performanti. Dal punto di vista teorico inoltre le proprietà asintotiche risultano migliori mentre computazionalmente è molto vantaggioso per il fatto che neppure le proiezioni oblique in questo caso devono essere calcolate esplicitamente in quanto in luogo di esse viene effettuato un passaggio di minimizzazione iterativa.

F. Commenti

L'algoritmo presentato in sommi capi per l'identificazione combinata deterministica stocastica può essere modificato aggiungendo alla decomposizione SVD 46 una pesatura con matrici W_1 e W_2 analoghe a quelle 35. Una scelta opportuna permette di ricondursi agli algoritmi più noti per l'identificazione subspace, rispettivamente N4SID, CVA e MOESP: i primi due condividono il metodo di calcolo delle matrici di Σ ed è analogo a quello presentato, mentre il terzo impiega la matrice estesa di osservabilità. Il concetto chiave è il fatto che la pesatura introduce un cambiamento di base nel sistema identificato il quale a sua volta si ripercuote in particolari proprietà spettrali della soluzione del problema di minimizzazione.

La riduzione di ordine di un modello identificato è un'operazione estremamente semplice in quanto, per la particolare struttura della decomposizione SVD da cui si originano i vettori che vengono coinvolti nella minimizzazione che porge le matrici del sistema, è sufficiente scartare le righe e le colonne di S_1 in eccesso e regolare le matrici U_1 e V_1 di conseguenza. Ovviamente in via del tutto generale la scelta degli autovalori di S_1 è dipendente dalla precisione di macchina e dagli errori

di arrotondamento; includere od escludere un valore singolare è una scelta che può compiere l'utente finale a seconda delle prestazioni desiderate da parte del modello identificato.

G. Implementazione software impiegata

Come accennato l'implementazione software degli algoritmi di identificazione a sottospazi è relativamente agevole in quanto non essendoci una funzione costo da minimizzare iterativamente non si presentano problemi di convergenza. Resta il problema della quantità di dati da processare: l'impiego delle funzioni di libreria per l'algebra lineare numerica è proibito dalla dimensione delle matrici in gioco (i risultati teorici sono garantiti per una quantità di campioni tendente a infinito che dal punto di vista pratico può tradursi in una necessità anche di una decina di migliaia di campioni). Inoltre la particolare struttura delle matrici a blocchi permette di sfruttare numerose proprietà delle scomposizioni usate permettendo di ottenere miglioramenti sostanziali dei tempi di calcolo. Per ultimo, la progettazione di un algoritmo richiede numerosi test di correttezza e quindi ci si è necessariamente dovuti rivolgere ad impiegare software scritto da terze parti. Esistono numerose librerie ed implementazioni commerciali oppure disponibili liberamente ed utilizzabili non commercialmente; si è scelto inizialmente di utilizzare l'algoritmo di identificazione subspace $n4sid$ implementato nell'Identification Toolbox di Matlab in quanto comunemente disponibile sulle piattaforme a disposizione. Si è in seguito introdotta la versione di $PBSID_{opt}$ implementata per l'ambiente Matlab e fornita dal prof. Chiuso.

V. MODELLO DINAMICO IDENTIFICATO

In questa sezione ci si propone di introdurre il modello del sistema identificato impiegando gli algoritmi N4SID e $PBSID_{opt}$ assieme ad alcune note implementative. Successivamente si riporteranno le equazioni di un filtro di Kalman tempo variante utili a ricostruire gli stati del modello e di conseguenza le misure delle uscite. Lo scopo è quello di impiegare un numero minore di sensori (quindi utilizzare solo alcune righe della matrice C identificata) per ricostruire l'intero set di misure di temperatura all'interno dell'edificio. La forma del sistema identificato è del tipo *forward innovation*

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{cases} \quad (52)$$

perfettamente equivalente a 10 e dove $u(t) \in \mathbb{R}^m$ sono ingressi deterministici noti (misure di temperatura e luminosità valide come ingressi esogeni), $y(t) \in \mathbb{R}^l$ sensori di temperatura interni ed $e(t)$ è rumore bianco tale che $E[e(t)e(s)^T] = Z\delta_{st}$, $Z \in \mathbb{R}^{l \times l}$.

Si nota che la procedura impiegata è a carattere centralizzato e prevede perciò di poter disporre istante per istante di tutte le misure richieste.

A. Modello identificato da N4SID

L'identificazione tramite N4SID e sue varianti ha richiesto di specificare esplicitamente di voler generare un modello asintoticamente stabile (autovalori di A dentro il cerchio unitario).

A livello intuitivo il modello di un edificio, in evoluzione libera e quindi con apporti esterni nulli di calore, dovrebbe gradatamente raffreddarsi. Reputando questa una condizione fondamentale si è forzato l'algoritmo a scegliere una matrice A stabile tramite l'apposita opzione.

Si è verificato sperimentalmente che nel particolare sistema fisico in esame per ottenere una stima non affetta da bias è necessario impiegare un sistema di ordine almeno pari al numero di sensori di misura delle uscite, cioè $x(t) \in \mathbb{R}^n$, $n \geq l$. Tenendo conto che nella decomposizione SVD l'ordine n viene determinato in base ai valori singolari diversi da zero (entro la precisione desiderata) tale termine può comunque essere forzato ad essere pari ad i (parametro di N4SID, determinato automaticamente maggiore od uguale a n). È naturalmente possibile identificare un modello di ordine $\bar{n} > l$ per poi eventualmente ridursi ad un modello di ordine minore, pur mantenendo il vincolo empirico di avere almeno l stati. Sarebbero necessarie ulteriori analisi ed indagini per proporre una spiegazione del fenomeno ma allo stato attuale non si è in grado di congetturare alcuna causa.

B. Modello identificato da $PBSID_{opt}$

L'identificazione tramite algoritmo $PBSID_{opt}$ nella sua implementazione ricorsiva si è rivelata migliore rispetto a quella precedentemente menzionata in quanto, oltre ad essere particolarmente veloce, ha prodotto sistemi stabili e con modello non affetto da bias. La selezione dell'ordine del modello in questo caso è stata lasciata al programma che sceglie automaticamente di troncatura la matrice S_1 (ottenendo in generale sistemi di ordine 30-40, vedi figura 24) secondo un criterio di rapporto tra valori singolari adiacenti e tra il valore massimo e minimo. La versione non ricorsiva dell'algoritmo ha invece presentato dei problemi sullo scaling delle matrici su cui effettuare le operazioni di decomposizione; nei sistemi identificati si sono rilevati marcati errori di bias ed in definitiva non si è rivelata idonea all'utilizzo al contrario della stessa implementazione ricorsiva. Tra i parametri impostabili dall'utente vi sono, oltre alla forzatura sull'ordine del modello, i parametri ks_p e ks_f : essi concernono la dimensione delle matrici di Hankel a blocchi ed influenzano i risultati dell'identificazione.

C. Filtro di Kalman tempo variante

Il modello dinamico 52 può essere usato per simulare in catena aperta il sistema: impiegando una parte di dati raccolti per l'identificazione e la rimanente per la validazione. Si è visto come le prestazioni degradino significativamente quando si tentano di seguire dati di temperatura non *spiegati* dal modello. Le cause sono molteplici e si possono annoverare tra esse sia il fatto che i dati raccolti presentano dapprima un trend di crescita per poi subire un assestamento (per includere tale tipo di informazione si è scelto di avere tra gli ingressi esogeni $u(t)$ almeno un sensore di temperatura interno) che la possibilità di trovarsi di fronte ad un sistema fisico altamente non lineare di cui non è possibile avere un modello lineare nell'intorno del range di temperature interessate. Altra concausa è sicuramente il fatto di non avere avuto nei giorni di rilevazione condizioni meteo ideali: risultati migliori sarebbero potuti emergere se

le temperature e l'irraggiamento fossero stati molto più vari rispetto alle giornate serene e calde avute.

L'idea è quindi quella di utilizzare un feedback dalle misure di uscita per correggere il modello: avendo a disposizione il modello di stato 52 si è scelto di implementare un filtro di Kalman tempo variante per ricavare gli stati e quindi le stime delle uscite. Il concetto chiave è di utilizzare solamente alcune delle possibili uscite (scegliendole opportunamente) per poi eseguire il filtro di Kalman istanziato con una matrice \hat{C} le cui righe corrispondono al sottoinsieme di uscite misurate scelte. Dallo stato ricostruito è quindi immediato, utilizzando la matrice C originale, simulare le uscite non misurate. Riguardo ai metodi di scelta si rimanda ai capitoli successivi, mentre si riportano le equazioni ricorsive impiegate (da [6]).

Sia M una generica matrice $m \times n$ e siano $r \in \{1, \dots, m\}$ e $c \in \{1, \dots, n\}$ due generici sottoinsiemi di indici di righe e colonne rispettivamente: la sottomatrice di M ottenuta scegliendo le righe r e le colonne c sarà indicata con $M_{[r,c]}$; se $c = 1$ scriveremo $M_{[r]}$ e se si indicano tutte le colonne (rispettivamente le righe) si indica $M_{[r, \cdot]}$ (risp. $M_{[\cdot, c]}$). Indichiamo con $\hat{l} \in \{1, \dots, l\}$ la scelta dei sensori di ingresso: il sistema 52 diventa

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t) + Ke(t) \\ y_{[\hat{l}]}(t) &= C_{[\hat{l}, \cdot]}x(t) + Du(t) + e_{[\hat{l}]}(t) \end{cases} \quad (53)$$

Ponendo il rumore di modello $w(t) = Ke(t)$ ed il rumore di misura $v(t) = e_{[\hat{l}]}(t)$ la matrice di covarianze incrociata dei rumori diventa

$$\begin{bmatrix} Q & S^T \\ S & R \end{bmatrix} = \begin{bmatrix} KZK^T & KZ_{[\hat{l}, \cdot]} \\ K^T Z_{[\cdot, \hat{l}]} & Z_{[\hat{l}, \hat{l}]} \end{bmatrix}.$$

Questo modello può essere riscritto con il rumore di modello e di misura scorrelati con $\tilde{w}(t) = w(t) - SR^{-1}v(t)$ di varianza $\tilde{Q} = Q - SR^{-1}S$, ottenendo

$$\begin{cases} x(t+1) &= (A - SR^{-1}C)x(t) + Bu(t) + \tilde{w}(t) \\ y_{[\hat{l}]}(t) &= C_{[\hat{l}, \cdot]}x(t) + Du(t) + e_{[\hat{l}]}(t) \end{cases}$$

Le equazioni del filtro di Kalman con ingressi in questo caso diventano quelle consuete, ponendo $\Lambda(t) = C_{[\hat{l}, \cdot]}P(t|t-1)C_{[\hat{l}, \cdot]}^T + R$ e $L(t) = P(t|t-1)C_{[\hat{l}, \cdot]}^T\Lambda(t)^{-1}$

- stime a priori

$$\begin{aligned} \hat{x}(t+1|t) &= F\hat{x}(t|t) + SR^{-1}y(t) + Bu(t) \\ P(t+1|t) &= FP(t|t)F^T + \tilde{Q} \end{aligned}$$

- stime a posteriori

$$\begin{aligned} \hat{x}(t|t) &= \hat{x}(t|t-1) + L(t) \left[y(t) - C_{[\hat{l}, \cdot]}\hat{x}(t|t-1) \right] \\ P(t|t) &= P(t|t-1) - L(t)C_{[\hat{l}, \cdot]}P(t|t-1) \end{aligned}$$

- condizioni iniziali

$$\hat{x}(0|-1) = \mu_0 \quad P(0|-1) = P_0$$

e si ottengono le stime dei sensori non misurati $\tilde{l} = \{1, \dots, l\} \setminus \hat{l}$ con la relazione $\hat{y}_{[\tilde{l}]}(t) = C_{[\tilde{l}, \cdot]}\hat{x}(t|t)$.

Per la determinazione degli stati iniziali del filtro di Kalman migliori al fine di minimizzare il transitorio ci si è posti in un contesto applicativo in cui una volta installati il numero

ridotto di sensori si debba far partire il monitoraggio: al primo istante si suppone di avere a disposizione una o più letture di temperatura dai sensori interni. Da esse si ricava una temperatura media della casa all'istante iniziale e si tenta di approssimare lo stato iniziale ai minimi quadrati, impiegando la pseudoinversa della matrice C . Matematicamente a partire dalle misure iniziali $y_{\hat{l}}(0)$ e avendo $N = \dim \hat{l}$ si ha

$$\begin{aligned}\bar{y}_0 &= \frac{1}{N} \sum_{i \in \hat{l}} y_i(0) \\ \mu_0 &= C^\dagger [\bar{y}_0 \quad \dots \quad \bar{y}_0]^T\end{aligned}$$

L'idea si è rivelata portare risultati migliori nel caso di un filtro di Kalman applicato al sistema identificato con PBSID_{opt} ma non si è riusciti a formulare nessuna ipotesi al riguardo. La varianza iniziale P_0 del filtro è invece stata tarata in maniera empirica sino a raggiungere risultati soddisfacenti; è da notare che una matrice varianza iniziale che dia buoni risultati su un modello identificato con N4SID non è detto fornisca prestazioni simili su un modello derivante da PBSID.

VI. SELEZIONE DEGLI INGRESSI

Il lavoro presentato in questa relazione si propone di effettuare la stima della temperatura negli ambienti interni di un edificio al fine di sviluppare in progetti futuri metodologie per il controllo delle stesse. La strategia impiegata è stata di rilevare con un numero elevato di sensori la temperatura in varie posizioni delle stanze (56 sensori per circa 300 m³) assieme ad alcune misure di luminosità e di temperatura ottenute all'esterno dell'edificio (almeno una temperatura ed una luminosità per facciata per un totale di nove misure). Se si considera infatti ad un edificio ed al monitoraggio di temperatura, in linea di principio, si vorrebbero ottenere delle stime di temperatura con un'accuratezza elevata tramite l'utilizzo di un modello dinamico che necessiti di poche misure per essere ricostruito. In questo modo la gestione di una rete di sensori, alla scopo di regolare il confort all'interno dell'edificio, risulterebbe molto più semplice ed economica. Quello in esame è un caso molto semplice nel quale si considerano come ingressi le misure di temperatura ed irraggiamento ottenute all'esterno dell'edificio mentre in applicazioni pratiche andrebbe tenuto conto anche di scambiatori di calore presenti all'interno quali radiatori e condizionatori. L'obiettivo è di riuscire a stimare le temperature interne utilizzando un modello dinamico con un numero di sensori limitato (un ordine di grandezza inferiore). La strada scelta è di identificare un modello lineare a partire dagli ingressi esogeni di temperatura e luminosità e verificare le capacità predittive del modello confrontandole con dati reali (di validazione, non impiegati per la stima del modello). A partire da tale modello per migliorarne ulteriormente le prestazioni si è scelto di implementare un filtro di Kalman non stazionario a partire da un sottoinsieme delle misure al fine di contenere l'errore di stima dello stato del sistema entro margini ragionevoli. Grazie alla relazione individuata precedentemente tra gli stati interni e l'insieme di tutte le uscite si riesce quindi a stimare le misure di temperatura

interne mancanti. Attraverso molteplici prove sperimentali si è scelto di procedere alla scelta dei sensori che entreranno nel modello finale scegliendo un numero limitato di ingressi esogeni per il modello lineare. Si è rilevato che utilizzando solamente ingressi esterni all'edificio si ha una stima del modello affetta da bias e quindi è sensato usare almeno una misura di temperatura interna come ingresso indipendente.

Per scegliere i sensori necessari ad ottenere il modello lineare le strategie potevano essere essenzialmente due: identificare con tutti gli ingressi e successivamente individuare in tale modello quelli più rilevanti oppure scegliere a priori alcuni sensori e assegnare un indice tramite un'opportuna metrica e confrontare le varie scelte. Per le tecniche di selezione degli ingressi di un modello complesso le opzioni erano molteplici (vedi [7]) e basate su impostazioni teoriche complete e perciò molto appetibili. Si è però visto come la riduzione del modello ottenuto con tutti gli ingressi (o comunque con molti ingressi) non sia una strada non percorribile: spesso il modello ottenuto identificando con molti ingressi risulta peggiore rispetto ad un modello ottenuto con pochi ingressi. Una motivazione che possiamo formulare come ipotesi è che gli ingressi esogeni dall'esterno (temperatura e luminosità) risultino molto simili tra loro e portando poca informazione causino dei problemi agli algoritmi di identificazione rendendo il sistema identificato poco robusto. Spesso addirittura l'algoritmo PBSID_{opt} ha portato a identificare modelli instabili la cui simulazione divergeva rapidamente. Ci si è perciò focalizzati nell'impiegare un numero di ingressi ridotto e essenzialmente di selezionare pochi ingressi per ogni tipologia possibile ed eleggere il sottoinsieme migliore secondo una metrica dipendente dal particolare sistema identificato con lo stesso sottoinsieme.

Lo scopo diventa perciò di capire quali sensori possono essere eliminati senza che la stima della temperatura peggiori in maniera sensibile. Solitamente in letteratura si utilizzano come ingressi esogeni le misure di temperatura e luminosità ottenute sulle pareti esposte a sud e quindi in linea teorica maggiormente irraggiate ma grazie alla disponibilità di dati raccolti è stato possibile indagare maggiormente ed operare una scelta sub-ottimale. La scansione esaustiva di tutte le combinazioni è computazionalmente proibitiva e si è quindi scelta la strada di un algoritmo di tipo greedy (vedi oltre per ulteriori informazioni sugli algoritmi greedy).

La metrica scelta è la somma della varianza degli errori di stima delle uscite (ricordando che il modello è in forma di stato stocastica di tipo *forward innovation*): da ogni tipologia di sensori (quattro nel caso specifico, temperatura interna ed esterna, luminosità di tipo active radiation e di tipo total solar radiation) viene scelta una particolare combinazione e con quella si procede all'identificazione. Viene poi simulato il modello sui dati di identificazione e in base all'errore di stima delle uscite si calcola la varianza di esso e si sceglie la traccia della stessa come indice di bontà.

Per poter scegliere gli ingressi più adeguati si è implementata una procedura in Matlab alla quale viene passata la totalità dei dati raccolti ed un vettore indicante la numerosità di ingressi desiderata ovvero quanti sensori interni e quanti di temperatura e luminosità esterni si intendono utilizzare per

l'identificazione. È ovviamente possibile impostare la scelta in maniera da non aver nessun ingresso di una particolare tipologia. Schematicamente l'algoritmo funziona nel modo seguente e comunque ricalca una strategia successivamente illustrata nella selezione delle misure per il filtro di Kalman. A partire da una soluzione vuota e si analizzano sequenzialmente i differenti tipi di ingressi; l'ordine dell'analisi va scelto in maniera arbitraria (ad esempio si può partire dai sensori di temperatura interni sino a giungere a quelli di luminosità). Si identifica il modello, utilizzando a scelta uno dei due algoritmi di identificazione basata su sottospazi, servendosi uno alla volta di tutti i possibili ingressi. Tra i modelli essi si sceglie quello che dà i migliori risultati sulla base della metrica considerata e lo si aggiunge alla soluzione di vuota di partenza. Partendo dalla nuova soluzione si procede iterativamente sino al raggiungimento numerosità desiderata operando di volta in volta la scelta più conveniente. Bisogna ricordare che questo tipo di algoritmi qualunque scelta, una volta effettuata, non viene mai rimessa in discussione: non c'è alcuna garanzia di ottimalità.

VII. SELEZIONE DEI SENSORI

Il passo successivo all'identificazione è la selezione delle uscite più significative del modello lineare allo scopo di ottenere un filtro numerico in grado di predire con sufficiente accuratezza la temperatura all'interno dell'edificio riducendo il numero di sensori iniziali di almeno un ordine di grandezza. Si tratta in sostanza di ridursi all'utilizzo complessivo di 6-8 sensori di cui necessariamente tre indipendenti utilizzati come ingressi, già selezionati in fase d'identificazione, ed i restanti da utilizzare per la predizione della temperatura all'interno dell'edificio.

La determinazione della soluzione ottima di questo problema risulta troppo onerosa in termini di tempo di calcolo. Se si volessero infatti comparare tutte le possibili combinazioni di n sensori presi in gruppi di K in K la complessità risulterebbe proporzionale a $C_{n,K} = \frac{n!}{K!(n-K)!}$. Nello specifico siamo interessati alle combinazioni di 53 sensori presi di 5 in 5 oppure di 3 in 3 ed in questo caso il valore di $C_{n,K}$ assume valori superiori a $2.8 \cdot 10^6$.

Da qui, la necessità di risolvere il problema in tempi di calcolo ragionevoli impedisce l'utilizzo di approcci di tipo esatto; nasce in questo caso l'esigenza di sviluppare delle tecniche in grado comunque di fornire delle buone soluzioni ammissibili in tempi di calcolo non troppo elevati, sacrificando l'ottimalità della soluzione in favore dell'efficienza con cui questa soluzione viene calcolata.

A. Criteri di scelta

Prima di passare alla descrizione degli algoritmi euristici utilizzati per la selezione dei sensori è bene introdurre quelli che sono i parametri utilizzati per la valutare la bontà delle singole soluzioni e per stabilire quali sensori costituiscano effettivamente il set migliore al fine di predire l'andamento della temperatura all'interno dell'edificio. Si sono introdotte a questo scopo due differenti metriche la prima basata sulla

traccia della varianza del modello ottenuto riducendo le uscite, la seconda sul fit rispetto all'andamento reale.

1) *Varianza da DARE*: Dato che la stima delle uscite non misurate avviene tramite un filtro di Kalman tempo variante si è utilizzata come varianza a priori sugli stati il valore a cui essa converge ovvero la soluzione dell'equazione algebrica di Riccati (DARE) sul sistema con l'insieme delle uscite ridotto. Partendo dal modello Σ identificato si calcola la traccia della varianza delle uscite passando per le matrici di varianza di rumore di misura e di modello che tengono conto delle sole componenti relative ai sensori selezionati. L'equazione di Riccati asintotica a tempo discreto

$$\bar{P} = F \left[\bar{P} - \bar{P} C_{[\hat{l},]}^T (C_{[\hat{l},]} \bar{P} C_{[\hat{l},]}^T + R)^{-1} C_{[\hat{l},]} \bar{P} \right] F^T + \tilde{Q}$$

restituisce la varianza d'errore a regime sugli stati a partire da un sottoinsieme \hat{l} delle uscite (vedi V-C per la notazione rilevante).

Per la risoluzione di tale equazione nell'ambiente Matlab si utilizza l'algoritmo a iterazione esponenziale (*doubling algorithm*, vedi [6]) basato sull'idea si calcolare la soluzione \bar{P} iterando, anziché l'equazione alle differenze, uno schema lineare. L'ammontare dei calcoli per iterazione richiesto dall'algoritmo è dello stesso ordine di grandezza di quello richiesto dall'equazione di Riccati, ma la velocità di convergenza è notevolmente superiore. A partire dalla soluzione della DARE \bar{P} si calcola, passando per la matrice di varianza a posteriori sugli stati $P(\hat{l})$, la varianza relativa alle uscite ($\Phi(\hat{l})$) la cui traccia viene usata come parametro per il confronto tra soluzioni differenti:

$$\begin{aligned} P(\hat{l}) &= \bar{P} - \bar{P} C_{[\hat{l},]}^T (C_{[\hat{l},]} \bar{P} C_{[\hat{l},]}^T + R)^{-1} C_{[\hat{l},]} \bar{P} \\ \Phi(\hat{l}) &= C P(\hat{l}) C' \end{aligned}$$

La funzione costo che ci si propone di minimizzare è infatti

$$J_d(\hat{l}) = \text{tr}(\Phi(\hat{l})).$$

Possiamo affermare con certezza che la complessità aumenta con la dimensione delle matrici in gioco ma non siamo in grado di stabilire quale sia la reale velocità di convergenza. Per questo nel seguito considereremo ciascuna chiamata alla procedura di calcolo della varianza di costo fisso.

2) *Fit del modello*: Il calcolo del fit avviene tramite l'utilizzo di uno stimatore dinamico, ovvero filtro di Kalman tempo variante, che permette di prevedere l'andamento delle temperature nell'intero edificio sulla base delle sole misure ottenute coi sensori selezionati. Tali stime vengono poi confrontate coi valori reali misurati e sulla base dell'entità dell'errore commesso si valuta la qualità della scelta corrente. La funzione che si intende minimizzare è rappresentata in questo caso dalla somma degli errori quadratici commessi su ciascuna uscita (i) per ciascun istante (t)

$$J_f(\bar{l}) = \sum_t \sum_i (y_{kalman} - y_{reale})^2$$

dipendente dal particolare set \bar{l} di sensori utilizzato (vedi V-C). Tale quantità è in relazione di proporzionalità con la varianza campionaria. La complessità risultante è lineare nella quantità di dati utilizzata per i calcoli.

B. Algoritmi Euristicici per la selezione dei sensori

Nelle applicazioni reali, il ricorso ad algoritmi approssimati o euristici è piuttosto diffuso in quanto spesso si è interessati ad avere una possibile soluzione per il problema in esame al fine di valutare velocemente degli scenari di lavoro. Inoltre molti dei parametri in gioco nelle applicazioni reali sono delle stime che possono essere soggette ad errore, per cui non vale la pena di aspettare troppo tempo per avere una soluzione il cui valore (o la cui ammissibilità) è di valutazione incerta.

Un algoritmo euristico è un qualunque metodo che fornisca una soluzione per un problema. Generalmente si vuole che un algoritmo euristico richieda un tempo di calcolo polinomiale nella dimensione dell'istanza che deve risolvere; anche se per problemi complessi si rilassa questo vincolo, richiedendo solo che l'algoritmo sia sufficientemente "veloce". Ovviamente l'algoritmo euristico ideale dovrebbe essere in grado di determinare sempre la soluzione ottima di un problema. Gli algoritmi euristici possono essere classificati nel seguente modo:

- 1) Algoritmi Costruttivi che partono da una soluzione vuota per poi determinare in modo iterativo i nuovi elementi da aggiungere in soluzione fino ad arrivare ad una soluzione completa. Sono generalmente di tipo greedy, basati su tecniche di ottimizzazione oppure basati su enumerazione implicita;
- 2) Algoritmi di Ricerca Locale che partono da una soluzione iniziale (generalmente ammissibile) e cercano iterativamente di migliorarla, effettuando delle (semplici) modifiche alla soluzione corrente. Essi terminano quando non è più possibile migliorare la soluzione corrente con le modifiche del tipo prescelto (ottimo locale);
- 3) Algoritmi Meta-euristici (evoluzioni degli algoritmi di ricerca locale) che utilizzano speciali tecniche per evitare di fermarsi in un ottimo locale;

Il tempo di calcolo, e la qualità delle soluzioni prodotte, cresce man mano che si passa dagli algoritmi costruttivi agli algoritmi di ricerca locale e ai meta-euristici.

1) *Algoritmo Greedy*: L'idea di base di questi algoritmi è di partire da una soluzione vuota e di costruire iterativamente la soluzione seguendo un semplice criterio di espansione, che consiste nell'effettuare, ad ogni iterazione, la scelta localmente ottima compatibilmente con i vincoli del problema. Qualunque scelta, una volta effettuata, non viene mai rimessa in discussione anche se alla lunga le scelte fatte possono non essere ottimali.

La soluzione di partenza non contiene alcun sensore. Alla prima iterazione vengono analizzati tutti i sensori e viene aggiunta alla soluzione quello che introduce il miglioramento più consistente in riferimento alla metrica considerata. Il procedimento si ripete, aggiungendo senza un sensore per ciascuna iterazione, sin a quando non si raggiunge la cardinalità della soluzione desiderata.

```
S=[];
for i=1:K
    OttimoLocale=inf;
    SensoreMigliore=-1;
```

```
for j=1:(n+1-K)
    S=S U {j}
    tr=metrica(S)
    if tr < OttimoLocale
        SensoreMigliore=j;
    end
end
S=S U {SensoreMigliore};
end
return S
```

La complessità di tale algoritmo è $O(K * n * c_{\text{dare}})$ dove n rappresenta il numero complessivo di sensori, K la cardinalità della soluzione e c_{dare} il costo della chiamata alla DATE.

2) *Algoritmo di ricerca locale*: L'idea di base degli algoritmi di ricerca locale è quella di definire una soluzione iniziale (soluzione corrente) e cercare di migliorarla esplorando un intorno (opportunamente definito) della stessa. Se l'ottimizzazione sull'intorno della soluzione corrente produce una soluzione migliorante il procedimento viene ripetuto partendo, come soluzione corrente, dalla soluzione appena determinata.

Per quel che riguarda la definizione della soluzione iniziale, si potrebbe pensare che sia meglio partire da una soluzione "buona" piuttosto che da una soluzione "scadente"; non esistendo nessun risultato teorico che confermi questa sensazione si son fatte diverse prove, sia partendo dalla soluzione ottenuta tramite l'algoritmo greedy, sia utilizzando set generati in modo casuale, senza riscontrare differenze sostanziali.

La definizione dell'intorno dipende dal tipo di mosse che si vogliono effettuare. È ovvio che se, per qualunque soluzione ammissibile, l'intorno coincide con l'intera regione ammissibile, la procedura restituisce la soluzione ottima del problema; questo però avviene per enumerazione di tutte le soluzioni della regione ammissibile (cosa ovviamente impossibile o quantomeno poco pratica). In generale, maggiore è la dimensione dell'intorno e più alta è la probabilità di sfuggire dagli ottimi locali; per questo si vorrebbe, per quanto possibile, avere intorno che contengano molte soluzioni. Altra caratteristica che si vorrebbe per l'intorno è il fatto di essere connesso; questo vuol dire che qualunque sia la soluzione di partenza, è sempre possibile raggiungere qualunque soluzione ammissibile (compresa quella ottima) tramite una opportuna sequenza di mosse; quindi se l'intorno è connesso, un algoritmo che non si blocchi negli ottimi locali troverebbe sempre la soluzione ottima del problema.

Nel caso in esame tutte le possibili combinazioni di cinque sensori rappresentano soluzioni ammissibili. Definiamo per semplicità S la soluzione di partenza ed I l'insieme formato da tutti i sensori utilizzati per misurare la temperatura all'interno dell'edificio esclusi quelli appartenenti ad S :

```
S = [2 24 30 37 49]
I = [1 3 4 .. 48 50 51 52 53].
```

L'intorno locale da noi considerato è costituito da tutte le combinazioni ottenibili sostituendo uno alla volta gli elementi che compongono la soluzione di partenza con tutti quelli contenuti in I . In ciascun ciclo si confrontano tutte le possibili combinazioni del tipo $S \setminus \{s\} \cup \{i\}$ con $s \in S$ ed $i \in I$.

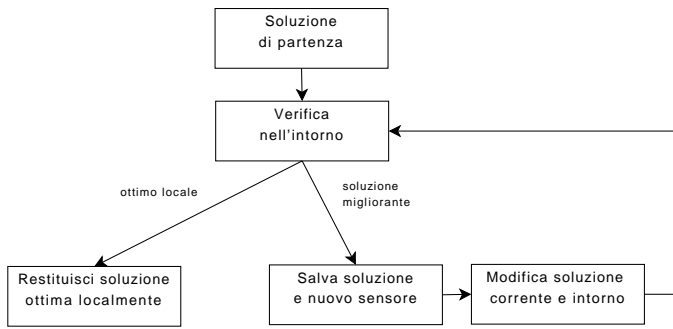


Fig. 9. Algoritmo di ricerca locale.

Se si trova un set migliorante rispetto a quello di partenza lo si utilizza come soluzione iniziale per un'ulteriore iterazione dell'algoritmo. In tal caso, oltre a memorizzare la nuova soluzione corrente S' , si tiene memoria anche del sensore che ha portato il miglioramento \bar{i} in modo da impedire la sua sostituzione durante l'iterazione successiva così da non ricadere in combinazioni già testate e risultate peggiori rispetto a S' . Si osservi che l'insieme I sarà necessariamente diverso da quello di partenza ovvero $I' = I \setminus \bar{s}$ data la necessità di non avere elementi ripetuti all'interno della soluzione; al contrario l'elemento \bar{s} sostituito in S per passare ad S' non verrà aggiunto ad I' in quanto porterebbe solamente alla ripetizione di casistiche già considerate.

In generale l'algoritmo termina quando non è più possibile trovare delle soluzioni miglioranti nell'intorno della soluzione corrente, oppure quando è stata determinata la soluzione ottima (valore della soluzione uguale a qualche bound); in alternativa si può far terminare l'algoritmo dopo un prefissato tempo di calcolo o numero di iterazioni. Nella nostra implementazione termina quando non è più possibile determinare una soluzione migliorante od al più dopo un numero prefissato di iterazioni. La complessità di quest'algoritmo è superiore a quella dell'algoritmo greedy considerato in precedenza. Si dovrà tener conto infatti anche del numero di iterazioni N_{max} che l'algoritmo compie, nel caso peggiore, prima di terminare. La complessità è dunque $O(K * (n - K) * costo_{dare} * N_{max})$.

```

S;
I = n \ S;
sensoreMigliore = [];
for k = 1:Nmax
    for i = 1:|S \ sensoreMigliore|
        OttimoLocale = metrica(S);
        for j = 1:|I|
            S' = S \ {i} U {j}
            tr = metrica(S')
            if tr < OttimoLocale
                SensoreMigliore = j;
                SensoreDaSostituire = i;
                OttimoLocale = tr;
            end
        end
    end
    if OttimoLocale == metrica(S)
        break
    else

```

```

            S = S \ {SensoreDaSostituire}
            ... U {SensoreMigliore};
        end
    end
end
return S

```

3) *Algoritmo genetico*: La soluzione cui converge un algoritmo di ricerca locale è un ottimo locale relativamente all'intorno scelto. In generale la soluzione trovata non è un ottimo globale. Per migliorare la soluzione trovata con tecniche di ricerca locale è possibile eseguire l'algoritmo più volte, partendo da soluzioni iniziali differenti e/o definendo l'intorno (o la strategia di esplorazione dell'intorno) in modo diverso in ciascuna iterazione. Un altro modo per cercare di migliorare le soluzioni ottenute consiste nell'utilizzo di algoritmi meta-euristici; questi sono essenzialmente delle estensioni degli algoritmi di ricerca locale, nei quali sono introdotte opportune tecniche mirate ad evitare di terminare in un ottimo locale. L'idea di base di un algoritmo meta-euristico è la possibilità di effettuare anche mosse peggioranti, al fine di non fermarsi in ottimi locali. Spesso questi algoritmi non sono estremamente competitivi rispetto a quelli sviluppati ad hoc per il problema, nel senso che le soluzioni prodotte sono paragonabili in termini di valore della soluzione ma i tempi di calcolo richiesti possono essere molto più elevati. In compenso, i tempi di sviluppo e di implementazione di un algoritmo meta-euristico di base sono molto inferiori rispetto ad una tecnica ad hoc. La fase critica nello sviluppo di questi algoritmi finisce spesso per essere la taratura di una serie di parametri che compaiono nell'algoritmo stesso.

Gli algoritmi genetici sono basati sull'osservazione dell'evoluzione di una popolazione di individui: gli individui si combinano per generarne di nuovi in generazioni successive ed il processo evolutivo tende a selezionare (far sopravvivere) solo gli individui più idonei. Facendo corrispondere a ciascun individuo una soluzione ed associando a ciascuna soluzione un valore che ne descriva la bontà, questi algoritmi cercano di simulare il processo evolutivo, provando a far sopravvivere solo le soluzioni di qualità elevata. Ciascun individuo nel nostro caso non è altro che un set di cinque sensori distinti ed i singoli sensori vengono detti geni. Le caratteristiche fondamentali di un algoritmo genetico sono rappresentate dal fatto che l'evoluzione dell'algoritmo non avviene da un individuo ad un altro individuo, ma da una popolazione ad un'altra popolazione e che le regole di transizione utilizzate non sono deterministiche ma probabilistiche. L'algoritmo parte con una popolazione iniziale di individui e cerca di ottimizzare il fit della popolazione mediante ricombinazione (cross-over) e mutazione dei geni nel corso di diverse generazioni. Il processo viene ripetuto fino a che non è stato generato un numero prefissato di popolazioni oppure non è stata dimostrata l'ottimalità della miglior soluzione trovata. Essendo l'obiettivo dell'algoritmo quello di cercare di migliorare la soluzione ottenuta con quello di ricerca locale si è deciso di sfruttare come popolazione di partenza l'insieme di ottimi locali ottenuti in ciascuna iterazione dell'algoritmo precedentemente descritto. Uno dei parametri da regolare sarà la cardinalità della popolazione di partenza. Ogni volta che viene chiamato l'algoritmo

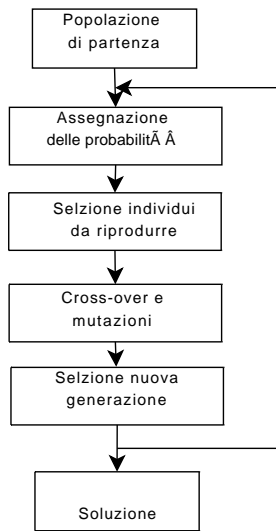


Fig. 10. Algoritmo genetico.

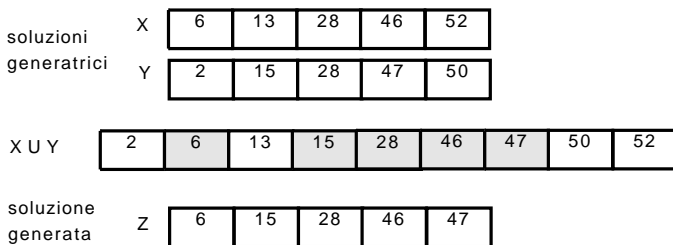


Fig. 11. Procedura di cross-over.

si verifica che la che la numerosità della popolazione sia pari a quella desiderata ed in caso contrario vengono introdotti nuovi individui tramite una procedura di generazione casuale. Una volta ottenuta la popolazione iniziale a ciascun individuo viene attribuita una probabilità, proporzionale alla qualità della soluzione che esso rappresenta, è che individua la probabilità che ha di riprodursi. In tal modo gli individui migliori si riproducono con maggiore frequenza rispetto agli altri. Infatti, proprio è tenendo conto di tali probabilità, viene selezionato un numero di prestabilito di individui che accoppiati in modo casuale dà origine a nuovi individui.

Per poter implementare un algoritmo genetico oltre alla popolazione di partenza è necessario definire e le procedure di cross-over e di mutazione che specificano il modo in cui si passa da una generazione all'altra. Le prime servono per ottenere una nuova soluzione z combinando due soluzioni generatrici x e y . Nella nostra implementazione questa procedura avviene unendo i vettori che rappresentano i due individui di partenza; nel caso in cui le soluzioni generatrici presentino geni in comune essi verranno considerati una sola volta. A partire da tale vettore vengono poi selezionati in maniera casuale i cinque elementi che rappresentano l'individuo della generazione successiva. Le procedure di mutazione sono mirate ad evitare la perdita irrimediabile di informazioni genetiche, mantenendo una diversificazione delle soluzioni trovate; questo serve ad evitare di rimanere confinati in una certa parte della regione ammissibile e di terminare con un ottimo locale.

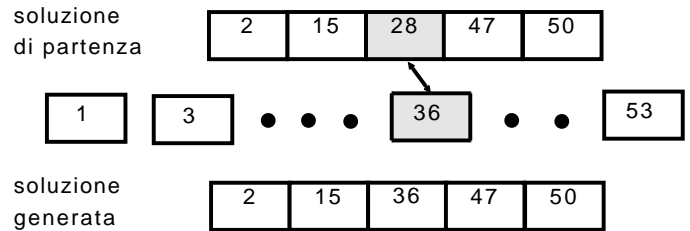


Fig. 12. Procedura di mutazione.

Questa procedura viene applicata ai singoli individui e consiste nella sostituzione di un gene con un altro scelto casualmente tra quelli che non appartengono all'individuo. I nuovi individui vanno poi a sostituire in pari numero i peggiori dal punto di vista della metrica considerata della generazione precedente. Tale sostituzione non porta in generale ad un miglioramento (i nuovi individui possono avere caratteristiche inferiori rispetto a quelli che vanno a sostituire) ma proprio questo tipo di procedura permette di sfuggire con maggiore probabilità dagli ottimi locali. È bene evitare anche che i nuovi nati risultino identici ad individui già presenti perchè, soprattutto in presenza di popolazioni poche numerose, ci si trova dopo poche iterazioni ad avere individui tutti uguali e quindi nell'impossibilità di ottenere ulteriori miglioramenti. L'algoritmo termina dopo un numero prestabilito di iterazioni. In altri termini i passi dell'algoritmo sono i seguenti:

1. poni $Niter := 0$ e determina una popolazione iniziale di n soluzioni
2. assegna a ciascuna soluzione x_i una probabilità $p(x_i)$ proporzionale alla metrica $f(x_i)$
3. seleziona $2m$ soluzioni casualmente, con probabilità $p(x_i)$, dalla popolazione corrente $P(k)$
4. accoppia le $2m$ soluzioni formando $2m$ nuove soluzioni con procedure di cross-over e di mutazione
5. determina una nuova popolazione $P(k+1)$;
6. if $(Niter < Nmax)$ then $Niter := Niter + 1$, vai al punto 2; altrimenti restituisci la soluzione;

Si sono riscontrati alcuni problemi nell'implementazione di quest'algoritmo legati alla taratura dei parametri (numerosità della popolazione, numero di generazioni ovvero di iterazioni, numero di nuovi individui generati in ciascuna iterazione). La complessità stessa dell'algoritmo varia notevolmente in funzione di tutti queste variabili.

VIII. RISULTATI SPERIMENTALI

In questa sezione sono presentati i risultati sperimentali ottenuti dall'identificazione e dal successivo modello con filtro numerico non stazionario.

A. Note sulle condizioni sperimentali

L'edificio in cui sono state effettuate le misurazioni e di cui si è tentato di modellizzare la dinamica termica è all'interno dell'area del Dipartimento di Elettronica ed Informatica dell'Università di Padova. Si tratta una struttura attualmente in disuso e perciò è stato possibile portare a



Fig. 13. Facciata a est dell'edificio.

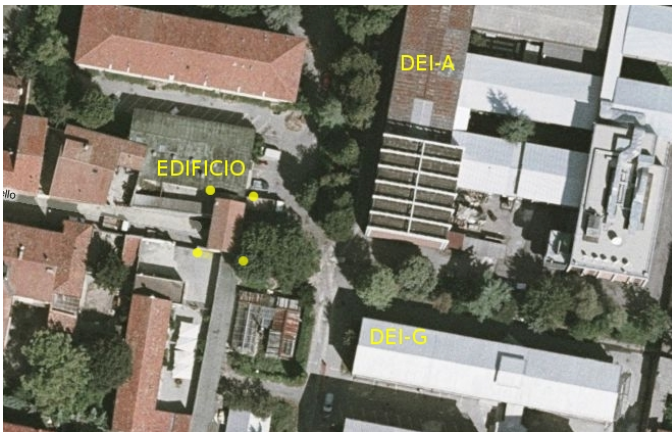


Fig. 14. Fotografia aerea dell'area: la casa è l'edificio contornato da punti gialli.

termine dieci giorni di misurazioni (dalle 22 circa del 15 giugno 2007 alle 10 circa del 26 giugno) ininterrotte con porte e finestre chiuse - come accennato nell'extended abstract uno dei principali problemi sull'identificazione termodinamica è lo spostamento di masse d'aria che causa forti non linearità e dinamiche non predicibili. Le dimensioni sono quelle di una piccola abitazione monofamiliare isolata e tale è anche la struttura interna in quanto a suddivisione in ambienti: la base è all'incirca dieci metri per cinque e l'edificio si sviluppa su due piani alti 2.6 m, trattandosi perciò di circa 100 m² e 300 m³. Una foto è riportata in figura 13 mentre dalla mappa aerea in figura 14 è possibile verificare l'orientazione dell'edificio (l'ingresso fotografato in 13 è il lato che guarda a est). Dalla stessa si deduce come esistano edifici di notevoli dimensioni dal lato est ma relativamente distanti, lato dove peraltro si trovano anche alcuni alberi, il che permette di affermare che l'insolazione delle facciate risulta pressoché totale durante tutto l'arco della giornata e leggermente inferiore nella mattinata.

Si avevano a disposizione un totale di 65 sensori e si è deciso di disporre

- 4 di temperatura esterna (uno per facciata) posti in

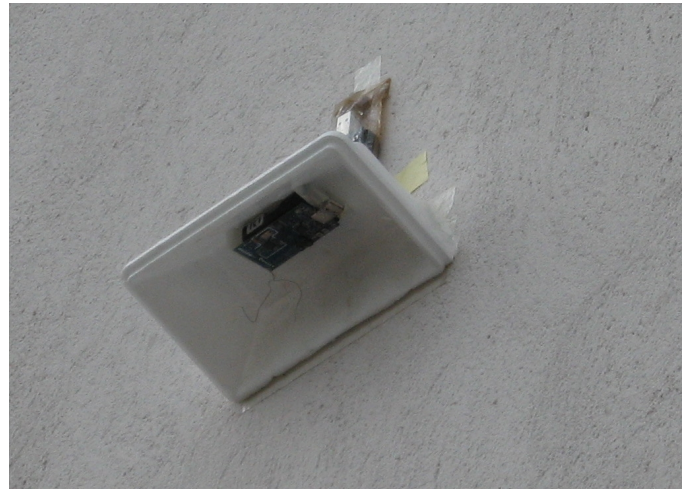


Fig. 15. Sotto la scatola di polistirolo è posizionato il mote per la misura di temperatura; sopra la scatola e protetto da un sacchetto di plastica trasparente è posizionato il mote per le misure di luminosità (la superficie sensibile dei fotodiodi è parallela alla parete).

posizione riparata dalla luce diretta del sole;

- 5 di luminosità (uno per facciata, due in quella dal lato est) che forniscono in tutto 10 possibili misure (active radiation e total solar radiation, vedi oltre nella sezione IX-A);
- 56 di temperatura all'interno collocati secondo un criterio di uniformità e in generale secondo le seguenti osservazioni
 - in ogni stanza 8 sensori (posizionati agli angoli e distanti 30 cm dalle pareti);
 - nelle stanze più grandi uno o più sensore al centro della stanza e a circa 1.5 m da terra;
 - in corrispondenza di porte e finestre, a 50 cm dalla stanza e a circa 1.5 m da terra.

Per rilevare le misure all'esterno si è stati costretti a distinguere tra sensori atti alla raccolta di dati di temperatura e di luminosità. Infatti ponendo i sensori di temperatura in pieno sole si sarebbero scaldati in maniera eccessiva alterando così la veridicità delle misurazioni fatte; al contrario per valutare l'irraggiamento non è stato possibile porre i sensori di luminosità in ombra. Come già accennato si sono mantenuti chiusi gli scuri (sigillando parzialmente eventuali spifferi dovuti alle non perfette condizioni della casa) ma tenendo aperte le finestre (le quali essendo dotate di doppi vetri avrebbero mantenuto un isolamento termico eccessivo). Allo stesso modo si sono chiuse tutte le porte di comunicazione interne. Per la posizione dei sensori esterni (ad un'altezza di circa 4.5 m da terra si veda la figura 15 per quelli disposti a sud ed a nord; i sensori esterni a est ed ad ovest sono stati posizionati sfruttando finestre ed infissi. Per la posizione dei sensori interni vedere la figura 16.

Dalle misure x, y, z delle posizioni dei mote e dalla tipologia della misura a cui erano destinati si sono ricavati i dati corrispondenti dalla simulazione ad elementi finiti dell'edificio. Grazie a tali dati ottenuti al computer si sono potuti effettuare esperimenti sull'identificazione e l'ottenimento di un modello numerico in grado di descrivere efficacemente del sistema.

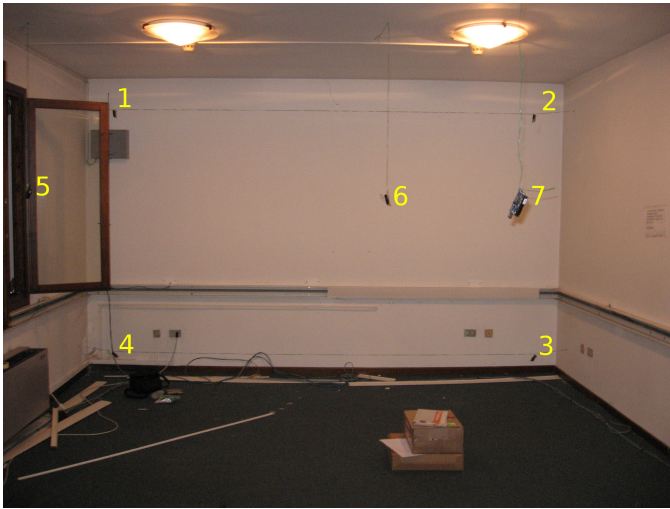


Fig. 16. Posizione dei sensori di temperatura interni (stanza piano superiore): 1,2,3,4 agli angoli; 5 davanti ad una finestra; 6,7 al centro della stanza.

È da notare come l'andamento simulato ricalchi in maniera fedele quanto si è poi misurato in fase di acquisizione dati, rendendo la simulazione uno strumento prezioso per testare gli algoritmi implementati.

B. Identificazione

Prima di procedere all'identificazione del modello lineare a tempo discreto i dati sono stati opportunamente vagliati al fine di rimuovere eventuali fenomeni anomali: si sono quindi scartate le misure di due sensori di temperatura in quanto ritenute affette da errori giudicati molto gravi (confronta con figura 35). Si è inoltre deciso di scalare gli ingressi di luminosità in un range prossimo a quello delle misure di temperatura usate come ingressi, al fine di migliorare il condizionamento numerico dei calcoli interessati.

La scelta finale sulle modalità di utilizzo degli algoritmi di tipo subspace impiegati (PBSID_{opt}, N4SID e varianti MOESP/CVA) è frutto di un procedimento sperimentale dettato sia da considerazioni teoriche sugli algoritmi coinvolti (vedi [15] per un compendio sulle tecniche di identificazione termodinamica per edifici) che da osservazioni basate su un approccio personale ed empirico alle problematiche. Grazie poi alla procedura iterativa greedy menzionata in VI si è scelto in maniera pseudo ottimale il sottoinsieme di ingressi esogeni che meglio potesse rispondere alle prerogative del progetto. Dal punto di vista metodologico si è come di consueto impiegata una parte dei dati per l'identificazione e il rimanente per la validazione dei risultati. La scelta dell'ordine del sistema è stata lasciata al valore ricavato di default per quanto riguarda PBSID_{opt} (vedi in figura 24 un tipico andamento dei valori singolari di S_1) mentre è stata forzata ad essere (almeno) pari al numero di misure per N4SID (come spiegato in V-A).

Sono state considerate svariate possibilità e si è scelto di riportare nelle figure seguenti il confronto tra i due metodi di identificazione subspace utilizzando due oppure tre ingressi esogeni scelti di caso in caso secondo l'algoritmo greedy sopra menzionato. In figura 17 è riportato un tipico andamento per

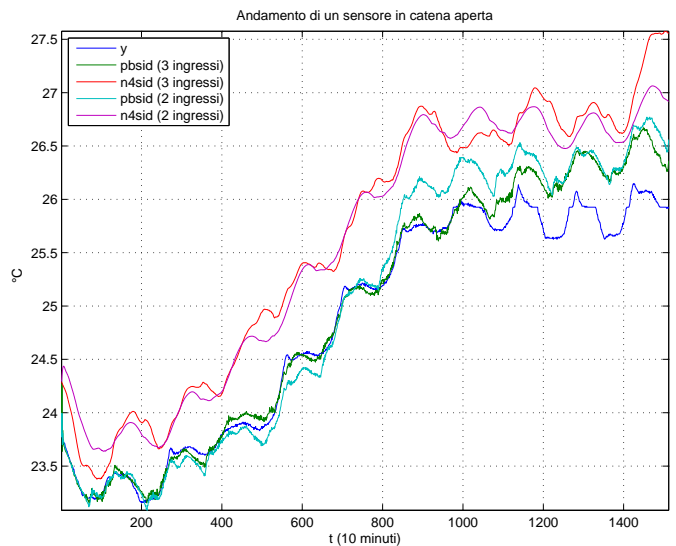


Fig. 17. In figura sono riportati gli andamenti della temperatura di un particolare sensore interno all'edificio. In blu la misura reale, in verde ed azzurro l'andamento del modello derivante da PBSID_{opt} con tre e due ingressi rispettivamente ed in rosso e viola l'andamento dell'algoritmo N4SID con tre e due ingressi.

i sistemi identificati e simulati. I primi due terzi dei dati sono stati usati per la validazione, il rimanente (dall'indice 1000 sino alla fine) la parte di validazione. Si nota come N4SID appaia affetto da bias già negli istanti iniziali (sui dati di identificazione e quindi dove le prestazioni dovrebbero essere migliori) e che diverga sensibilmente nella parte di validazione. Il PBSID_{opt} si comporta invece decisamente meglio ed addirittura l'andamento migliore si registra per soli due ingressi (uno di temperatura interna ed uno di luminosità). Si nota come tutti e quattro i modelli divergano sensibilmente in validazione: le caratteristiche dinamiche del sistema sono leggermente differenti dai primi due terzi e le equazioni dinamiche non possono spiegare le nuove caratteristiche dell'andamento di temperatura.

Avendo a disposizione un numero elevato di dati ottenuti con un campionamento molto fitto (trenta secondi) il primo passo è stato ridurre il numero di campioni: si è preferito non applicare alcun trattamento numerico ai dati ma solamente di scegliere un tempo di campionamento maggiore e multiplo del primo e sottocampionare in tal modo le sequenze. Sia in base a considerazioni empiriche che grazie alle simulazioni numeriche sviluppate si è scelto un tempo di campionamento di dieci minuti. Analizzando la risposta in catena aperta di alcuni sistemi identificati al variare del tempo di campionamento (vedi 18) si evince come tale scelta per la variabile progettuale sia la più interessante, peraltro senza escludere a priori la possibilità di modificarla verso l'alto od il basso a seconda delle esigenze come si può notare in figura 19.

Nelle figure 20 e 21 sono riportate alcune informazioni indicative del comportamento degli algoritmi: per ogni sensore di misura (le uscite) si è calcolata la media nel tempo ed il massimo dell'errore $e(t) = y(t) - y_{ca}(t)$ nella parte di validazione. Come si nota le prestazioni sono migliori per PBSID_{opt} sia per quanto riguarda l'errore medio che massimo.

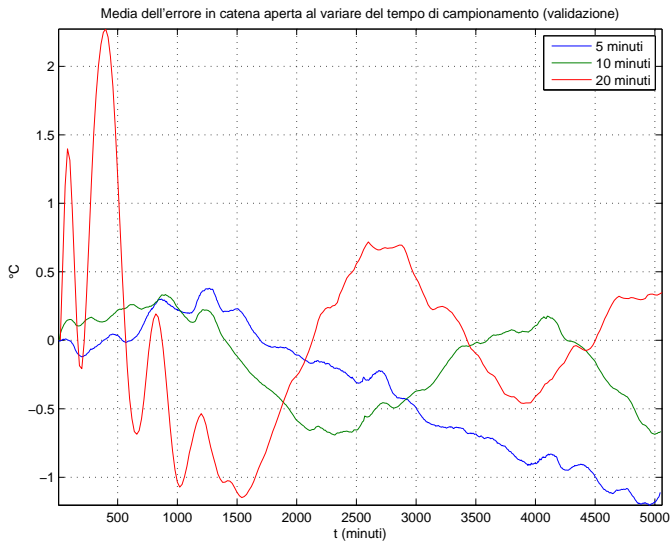


Fig. 18. Il grafico mostra l'errore medio al variare del tempo sui dati di validazione (mediato tra le uscite misurate) di un sistema identificato con N4SID troncato all'ordine pari al numero di uscite e con tre ingressi esogeni giudicati ottimali dall'algoritmo greedy al variare del tempo di campionamento. Si nota come non sia possibile definire un particolare tempo di campionamento come non adatto: in base alle applicazioni per cui è destinato il modello numerico è possibile scegliere tale variabile progettuale in un intervallo che può andare da un minuto ad alcune decine di minuti senza incorrere in penalizzazioni eccessive.

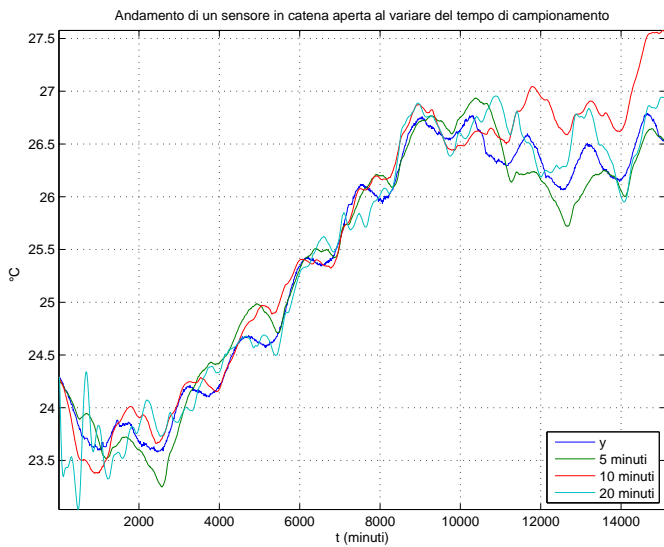


Fig. 19. Andamento della simulazione di un modello dinamico identificato con differenti tempi di campionamento: non si notano differenze sostanziali.

Nelle figure 22 e 23 si possono apprezzare altre caratteristiche degli algoritmi di identificazione, nel primo caso riportando l'errore massimo tra l'insieme degli ingressi al variare del tempo e nel secondo riportando l'errore medio. Ancora l'algoritmo N4SID si dimostra avere prestazioni peggiori. In generale sono perciò migliori i modelli dinamici identificati da PBSID_{opt} nella sua implementazione ricorsiva. La selezione degli ingressi è stata provata con successo sui dati ottenuti dalla simulazione agli elementi finiti.

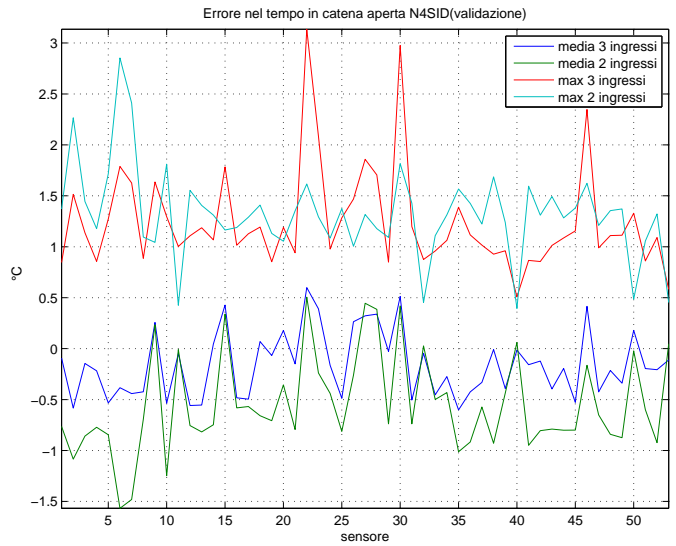


Fig. 20. Per ogni sensore è stata calcolata la media e l'errore massimo sui dati di validazione per i modelli di tipo N4SID. Diversi sensori sono affetti da bias ed alcuni manifestano errori massimi rilevanti.

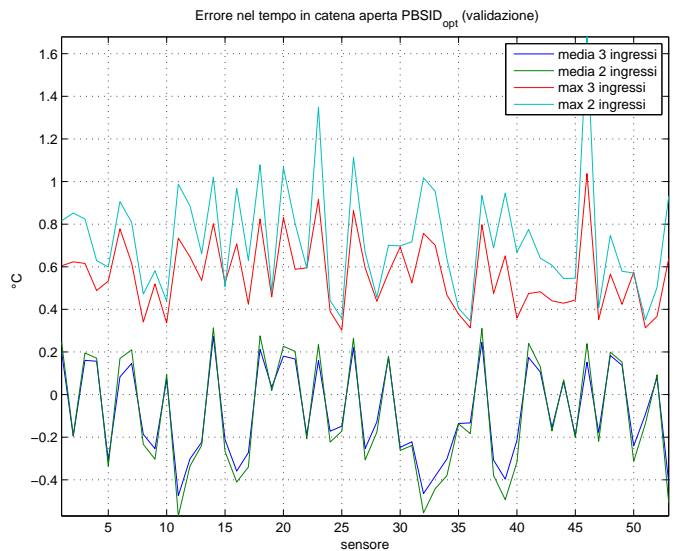


Fig. 21. Per ogni sensore è stata calcolata la media e l'errore massimo sui dati di validazione per i modelli di tipo PBSID_{opt}. I risultati sono migliori rispetto a quelli del grafico 20.

C. Selezione dei sensori

I modelli dinamici possono essere usati per simulare il sistema in catena aperta ma si è visto come le prestazioni possano non essere delle migliori di fronte a dinamiche inattese che il modello stesso non è in grado di spiegare. L'idea già presentata è quella di inserire un feedback dalle misure di uscita per correggere il modello. È stato implementato un filtro di Kalman tempo variante per ricavare gli stati e quindi le stime delle uscite utilizzando solamente alcune delle possibili combinazioni di misure selezionate sulla base di algoritmi euristici. Anche in questo caso è stato vantaggioso sperimentare gli algoritmi sul sistema simulato numericamente (vedi 25,26 e 27). I miglioramenti rispetto alla simulazione in catena aperta che si sono potuti ottenere procedendo in questa maniera

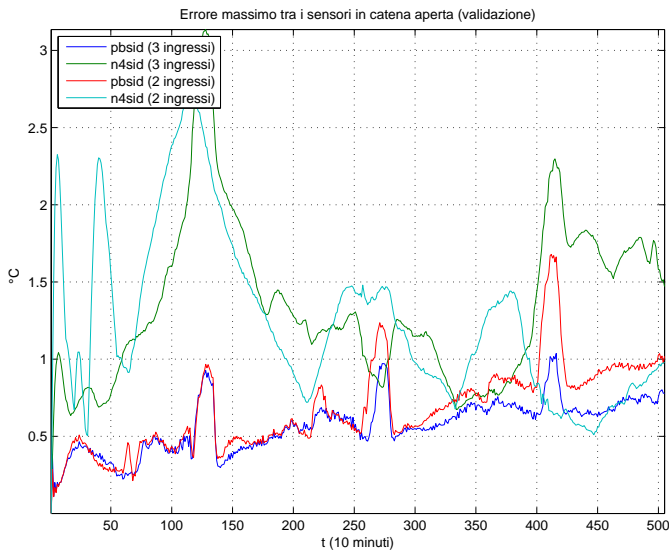


Fig. 22. Al variare del tempo il grafico riporta il massimo errore rilevato tra l'insieme delle uscite.

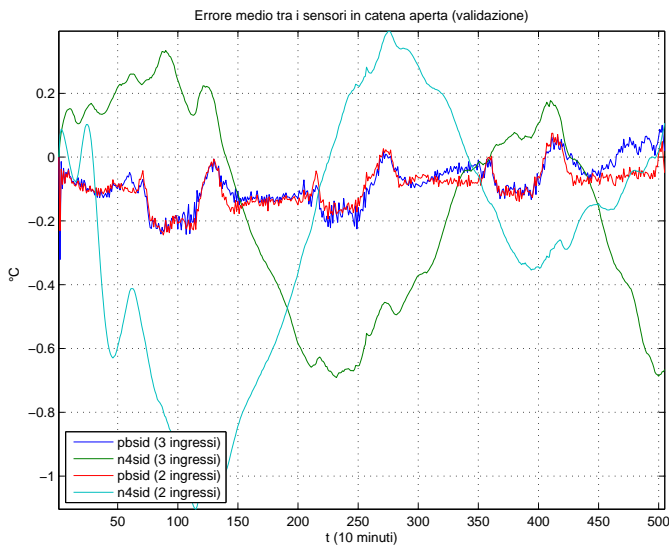


Fig. 23. Al variare del tempo il grafico riporta l'errore di stima mediato tra l'insieme delle uscite.

sono evidenti in figura 28. I modelli considerati in questa sezione sono quelli derivanti, per le due differenti procedure di identificazione, dall'utilizzo di tre ingressi esogeni scelti in modo sub-ottimale tramite l'algoritmo greedy descritto in precedenza. Si tratta in particolare di un sensore di temperatura interno, di uno esterno ed uno di luminosità (*active radiation*). A conferma dei migliori risultati ottenuti con l'algoritmo PBSID_{opt} si riporta in figura 29 l'andamento delle temperature di un sensore interno a confronto con le stime ottenute col filtro di Kalman per ciascun procedimento di identificazione. Anche in questo caso si osservano risultati migliori con il modello derivante dall'algoritmo PBSID_{opt}; per questo motivo i grafici che seguono non faranno più riferimento all'utilizzo di N4SID. Essendo lo scopo primario quello di ridurre il più possibile la numerosità di sensori impiegati si sono effettuate delle prove per valutare le differenze qualitative tra

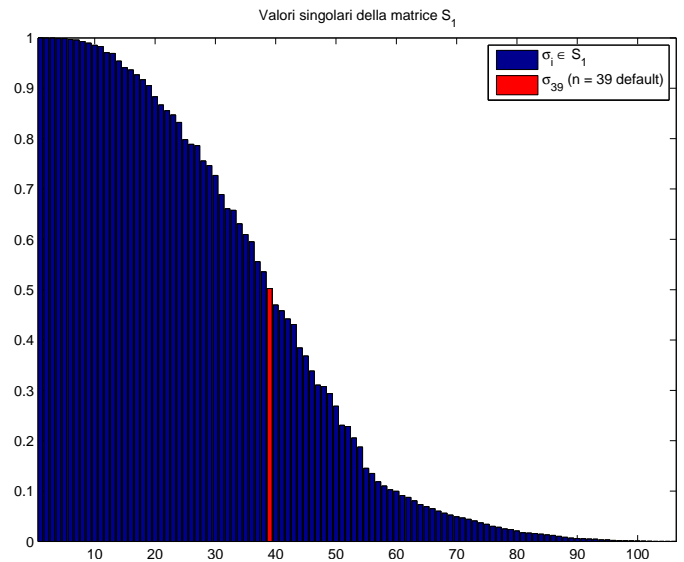


Fig. 24. Valori singolari (normalizzati all'unità) della matrice diagonale $S_1 \in \mathbb{R}^{106 \times 106}$ ottenuta dalla decomposizione SVD nell'algoritmo PBSID_{opt} con tre ingressi. Come si nota l'ordine del sistema che si avrebbe non ignorando alcuna colonna dei sottospazi riga e colonna della matrice di partenza avrebbe dimensione $n = 106$, mentre il valore proposto dal programma per il troncamento è $n_{def} = 39$. Si nota come sia presente uno scalino attorno al valore 53, al di sopra del quale l'ampiezza cala significativamente e che potrebbe essere una valida alternativa per il troncamento.

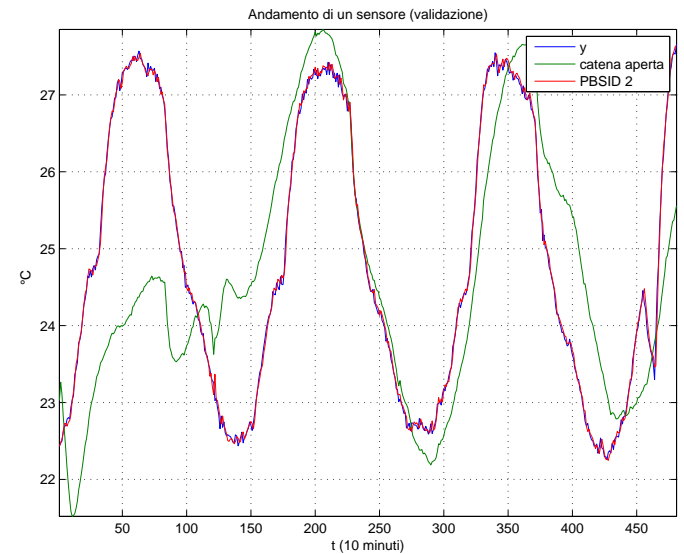


Fig. 25. Andamento di un sensore sui dati di validazione. Il modello è stato identificato con PBSID_{opt} sui dati di simulazione con tre ingressi (uno per tipologia, scelti in maniera greedy) e il suo andamento in catena aperta è riportato nel grafico assieme alle stime calcolate con un filtro di Kalman (per il quale sono state usate due uscite scelte tramite ricerca locale).

stime basate su insiemi più o meno grandi di *mote*. Nelle immagini 30 e 31 si osservano variazioni piuttosto elevate e di conseguenza stime molto imprecise quando ci si serve di uno o due uscite. Tali differenze sono maggiormente evidenti nelle fasi di transitorio. È bene paragonare a questo punto le soluzioni che si ottengono coi diversi algoritmi euristici per la selezione dei sensori utilizzati per stimare le uscite tramite il filtro di Kalman. All'algoritmo di ricerca locale si è

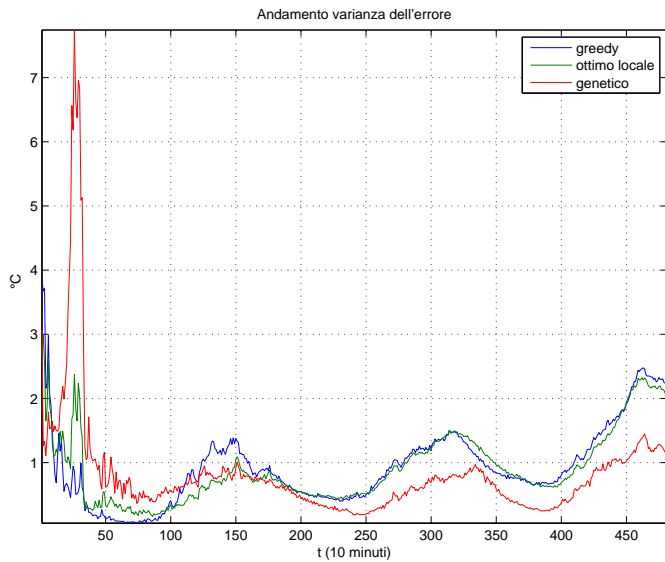


Fig. 26. Varianza dell'errore al crescere del tempo sull'insieme dei sensori per i dati di validazione della simulazione. Il modello è stato identificato con PBSID_{opt} come in figura 25 e sono riportate le scelte ottenute tramite gli algoritmi euristici di selezione delle uscite.

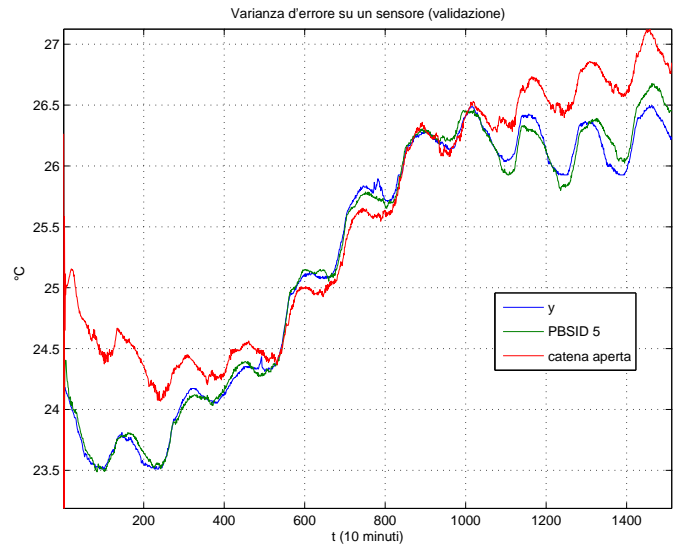


Fig. 28. In figura sono riportati gli andamenti della temperatura di un particolare sensore interno all'edificio. In blu la misura reale, in verde la stima ottenuta con un filtro di Kalman tempo variante sulla base del modello derivante da PBSID_{opt} ed in rosso l'andamento in catena aperta derivante dallo stesso modello.

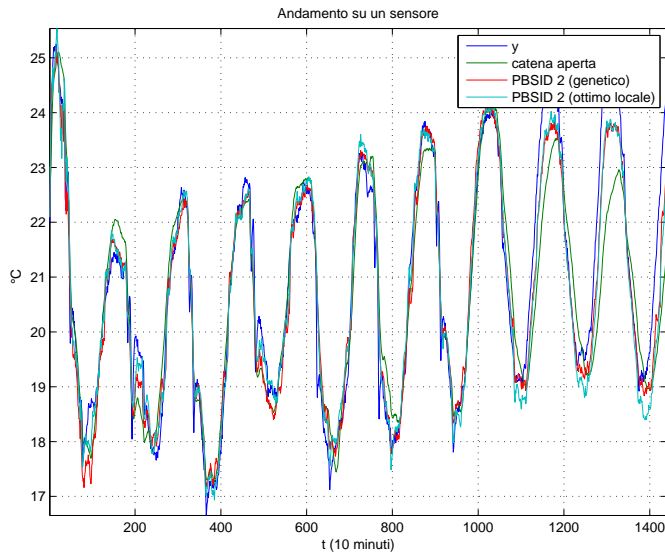


Fig. 27. Andamento della misura e delle stime su un sensore in simulazione (stesse condizioni di figura 25). Si nota come nella parte di validazione (ultimo terzo dei dati) la stima in catena aperta tenda a divergere.

passata la soluzione ottenuta con quello greedy e nella totalità dei casi analizzati ha portato dei miglioramenti in termini di varianza a posteriori sulle uscite. Qualche difficoltà, legata alle tarature dei parametri ed all'aleatorietà della procedura, si è riscontrata col l'algoritmo genetico. Tale algoritmo ha comunque portato ad alcuni miglioramenti nella soluzione. Nelle figure 32 e 33 si possono apprezzare le differenze in termini di varianza empirica sulle uscite e di qualità della stima della temperatura di un particolare sensore. Si è riscontrato che spesso si ha un buon accordo sull'impiego della metrica basata su DARE e su quella basata sull'errore calcolato simulando il filtro stesso. I risultati presentati in questa sezione sono tutti basati sulla metrica DARE, molto più veloce in termini

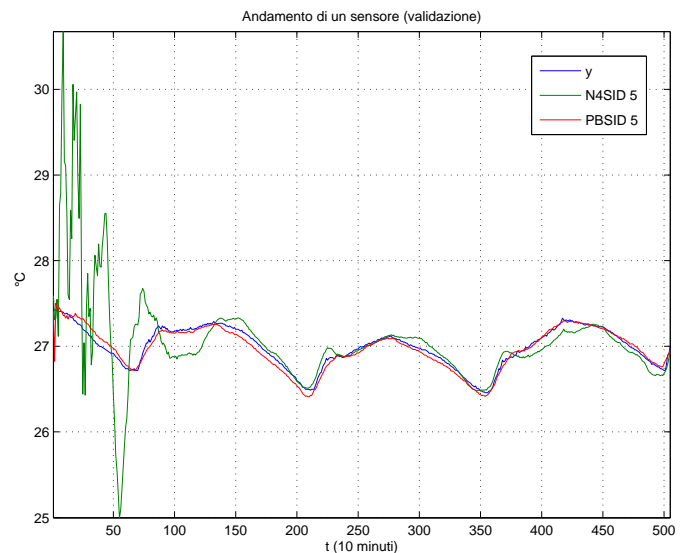


Fig. 29. In figura sono riportati gli andamenti della temperatura di un particolare sensore interno all'edificio. In blu la misura reale, in verde l'andamento della stima ottenuta con un filtro di Kalman tempo variante sulla base del modello derivante da PBSID_{opt} ed in rosso l'andamento della stima basata sul modello N4SID. In entrambe i casi la stima è fatta con un set di cinque sensori e sui soli dati di validazione.

computazionali. Si nota che l'altra metrica è da privilegiare qualora si intenda minimizzare anche la durata e l'entità del transitorio all'avvio del filtro (pur dipendendo ad esempio anche da altri parametri quali la varianza iniziale): a tal proposito si nota come le condizioni iniziali per gli stati stimati attraverso il metodo empirico proposto in V-C si comportino molto meglio con i sistemi identificati tramite PBSID_{opt} e come quest'ultimo metodo sia in definitiva da preferire.

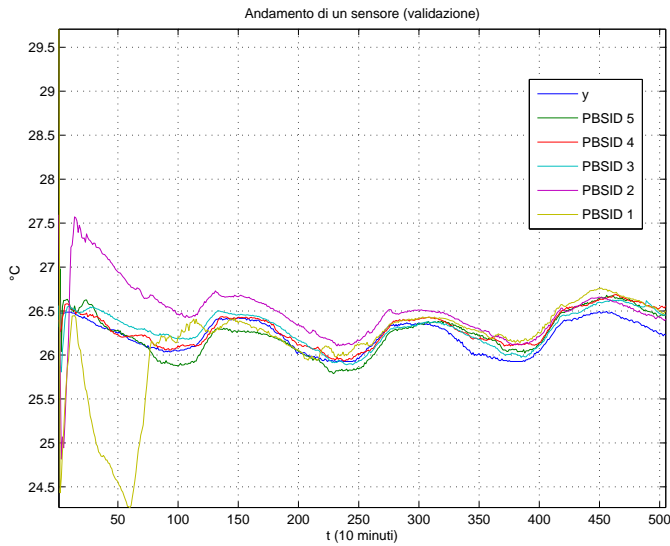


Fig. 30. In figura sono riportati gli andamenti della temperatura di un particolare sensore interno all'edificio. Si confrontano i risultati delle stime ottenute con le misure di set di uscite di differenti numerosità. Si osservi come, soprattutto durante il transitorio, siano meno affidabili i risultati ottenuti con pochi sensori.

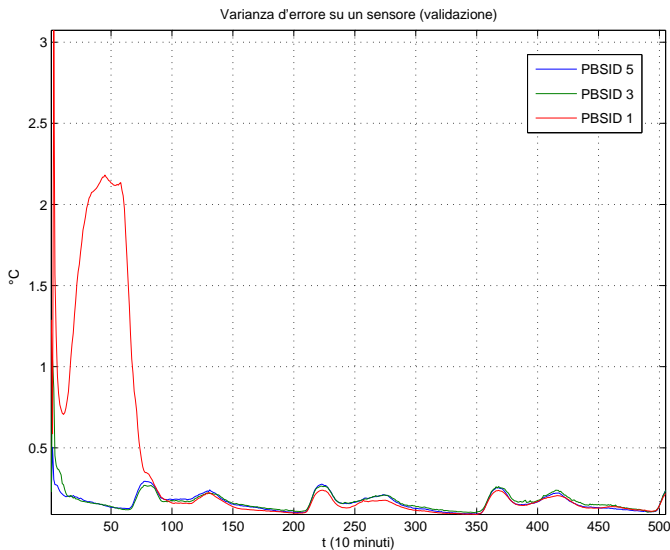


Fig. 31. Il confronto avviene tra le varianze d'errore sull'uscita di un particolare sensore all'interno dell'edificio. Si osserva come si ottengano risultati migliori quando la stima si basa sulle misure di un numero maggiore di uscite.

IX. ACQUISIZIONE DEI DATI

Per la misurazione dei dati dell'edificio sono stati impiegati i dispositivi hardware *MoteIV Tmote Sky*: pur essendo espressamente progettati per un utilizzo in reti wireless essi dispongono delle caratteristiche necessarie alla raccolta delle misure. Oltre alla disponibilità *on-board* dei sensori di luminosità e temperatura necessari e la possibilità di immagazzinare dati in un'apposita memoria di tipo flash, i vantaggi derivanti dall'utilizzo di questa soluzione sono stati l'impiego di un ambiente software di programmazione altamente integrato ed i particolari accorgimenti implementati sia a livello hardware che software per garantire una durata sufficiente della carica

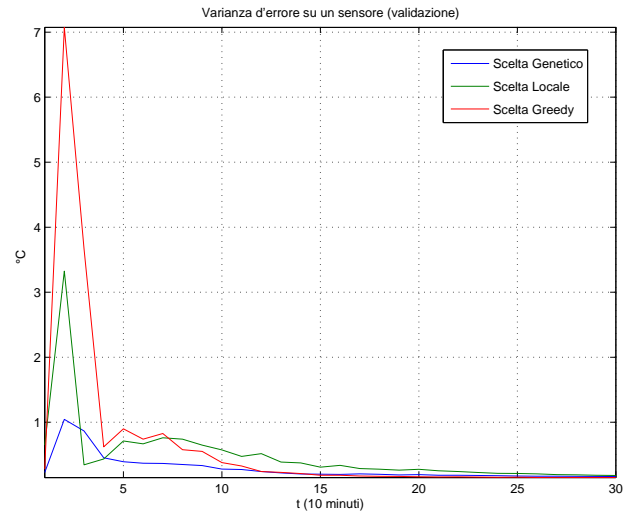


Fig. 32. La varianza d'errore delle stime ottenute relativamente all'andamento della temperatura di un singolo sensore con il filtro di Kalman che utilizza i sensori selezionati dai differenti algoritmi genetici. Si osserva come calando passando dall'utilizzo delle soluzioni fornite dell'algoritmo greedy sino a quello genetico.

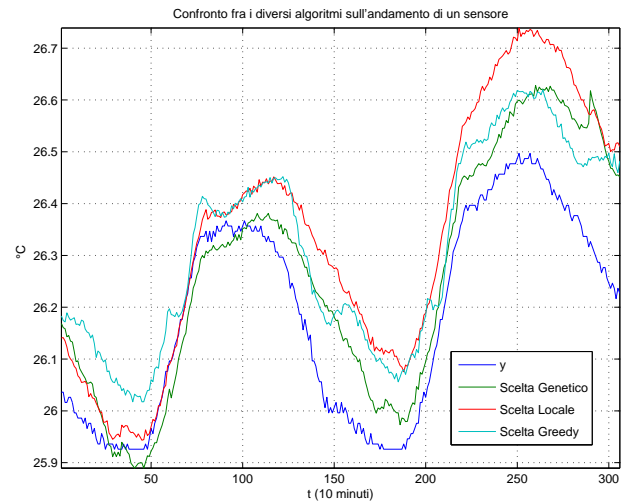


Fig. 33. Nel grafico è riportato l'andamento reale della temperatura di un particolare sensore interno all'edificio comparato con le stime ottenute coi sensori selezionati dai diversi algoritmi.

per il periodo di raccolta. L'obiettivo minimo era una settimana di analisi, con un tempo di campionamento minimo di cinque minuti (fattori dovuti a considerazioni sull'orizzonte di tempo necessario ad algoritmi di identificazione subspace).

A causa sia di problemi logistici che per limitazioni temporali, unitamente alla non ancora matura disponibilità di protocolli efficienti per lo scambio dati via wireless ed al fatto che una parte dei dispositivi non presentano un'antenna esterna adatta alla trasmissione a distanza del segnale radio, si è scelto di effettuare una raccolta dati in maniera *off-line*. I dati raccolti dai sensori ed alcune informazioni di controllo aggiuntive sono stati scritti sulla memoria flash per poi essere letti via port usb. Una particolare cura è stata posta all'aspetto del consumo energetico: se è vero che non si è impiegato intensivamente il modulo di trasmissione e ricezione radio e che la durata

dell'esperimento era limitata per vari fattori non era peraltro nemmeno possibile ripetere l'acquisizione o perlomeno avere un controllo della stessa se non a posteriori. Difatti qualsiasi modulo di comunicazione verso l'esterno (via radio o via usb) è stato bloccato alla partenza della fase di campionamento per risparmiare energia; allo stesso modo sono stati disattivati i led di stato. Nonostante i mote siano espressamente progettati per garantire una lunga durata delle batterie i componenti necessari all'esperimento (sensori e memoria flash) hanno in comune il problema di necessitare di una tensione di alimentazione costante di circa 3 V forniti da due batterie di tipo AA alcaline, nonostante il processore possa funzionare con tensioni fino a 1.9 V. Infatti l'acquisizione dei dati dai sensori è basata sull'impiego di convertitori ADC i quali a loro volta sono costruiti a partire da comparatori di tensione che necessitano di una tensione di riferimento quanto più costante possibile. Al diminuire della tensione applicata possono accadere due inconvenienti di notevole rilevanza: una misurazione a 14 bit di precisione potrebbe avere solamente i 12 bit upper significativi oppure potrebbe essere necessario adottare dei fattori differenti nella conversione in unità ingegneristiche. La memoria d'altra parte necessita di una tensione di alimentazione quanto più prossima a quella nominale per poter essere scritta in maniera sicura (minima tensione operativa 2.6 V). Gli esperimenti condotti hanno evidenziato come dopo 15 giorni la tensione nominale delle batterie fosse mediamente di 2.9 V, sufficiente al corretto funzionamento.

A. Descrizione sensori impiegati

1) *Temperatura e umidità*: Il sensore di temperatura ed umidità disponibile sul mote è il modello Sensirion SHT11. Esso è composto da un sensore basato sulla dipendenza dalla temperatura della banda di non conduzione di un semiconduttore e da un sensore basato sulla variazione di capacità di un polimero dovute all'umidità relativa nell'aria. Nel chip è incluso un convertitore analogico digitale a 14 bit e la logica per la comunicazione seriale. La calibrazione del sensore è gestita internamente in base a tabelle preimpostate. L'interfaccia al sensore è interamente gestita ad alto livello tramite routine disponibili con il sistema operativo TinyOS 2.x e conformi alla tipica sequenza di richiesta lettura seguita da un callback ad una funzione di lettura eseguita.

Il sensore di temperatura è definito come *bandgap proportional to absolute temperature* e la formula di conversione dalla lettura a 14 bit T_{raw} immagazzinata nella flash al valore in gradi centigradi è

$$T = b_1 + b_2 T_{raw} - b_3 (T_{raw} - b_4)^2 \quad [^{\circ}C] \quad (54)$$

con costanti $b_1 = -39.6$, $b_2 = 0.01$, $b_3 = -2 \cdot 10^{-8}$ e $b_4 = 7000$. Il sensore di umidità ha invece formula di conversione dal valore a 12 bit H_{raw}

$$RH_{true} = (T - 25) (c_1 + c_2 H_{raw}) + c_3 + c_4 H_{raw} + c_5 H_{raw}^2 \quad [\%]. \quad (55)$$

con costanti $c_1 = 0.01$, $c_2 = 8 \cdot 10^{-6}$, $c_3 = -4$, $c_4 = 0.0405$ e $c_5 = -2.8 \cdot 10^{-6}$ (vedi [8] e [9]).

Essendo il sensore (di massa negligibile) montato a contatto di un corpo di dimensioni rilevanti (mote e due pile AA) è

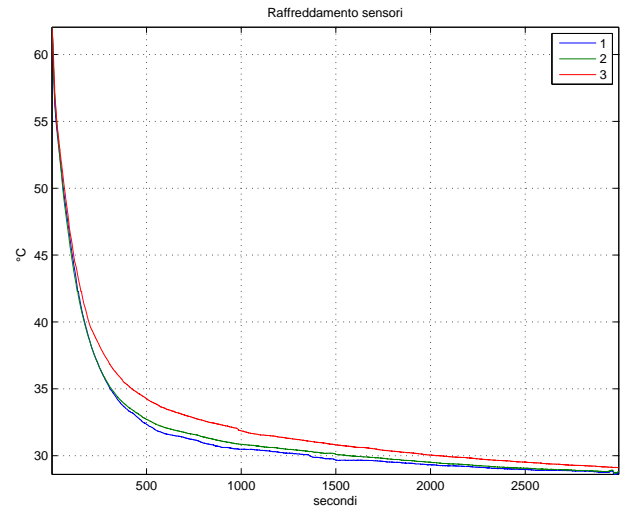


Fig. 34. Andamento curve di raffreddamento in un esperimento con tre sensori. Il tempo medio su tre esperimenti per l'assettamento all'1% del valore di regime è di 42 minuti e 30 secondi circa. Si conclude che il dispositivo comprensivo di batterie è dotato di una notevole inerzia termica.

possibile ricavare per via sperimentale una costante termica. Un problema riscontrato nell'analisi a posteriori dei dati è dovuto ad un probabile errore nel sensore di temperatura: in diversi mote la lettura rimane bloccata al valore di $25.93^{\circ}C$ per una quantità di tempo notevole (nel caso peggiore anche per oltre 15 ore continue). Il fatto che per diversi mote il valore errato sia il medesimo e che sia un problema circoscritto ad alcuni esemplari fa pensare ad un difetto di comunicazione tra sensore e microcontrollore; peraltro finita la fase critica il sensore riprende un funzionamento all'apparenza normale. La possibilità di un errore a livello di firmware programmato è da escludere, in quanto non sarebbe stata circoscritta al solo sensore di temperatura. Si è scelto di mantenere comunque tali dati, in quanto per ampie porzioni essi sono significativi: l'errore è pertanto stato assimilato ad un rumore di misura (purtroppo non bianco e scorrelato). Altre letture di temperatura sono invece state scartate in quanto evidentemente problematiche (vedi 35). Un problema metodologico che si è incontrato nella fase di programmazione e di testing dei sensori di temperatura si presenta quando il mote è interfacciato alla porta usb il regolatore di tensione dai 5 V della porta ai 3 V necessari al mote causa una notevole dissipazione di calore che porta all'inaffidabilità delle letture effettuate: qualsiasi test in tal senso è da svolgere con alimentazione a batteria.

2) *Luminosità*: Per la misura della quantità di luce incidente il mote ha a disposizione due sensori prodotti da Hamamatsu: il modello S1087 per la rilevazione di luce nello spettro della luce visibile (320 – 730nm) e il modello S1087-01 per la rilevazione dello spettro visibile ed infrarosso (320–1100nm). Per le finalità dell'esperimento l'informazione necessaria è data dalla quantità di luce solare irradiata sulle pareti dell'edificio: i sensori sono perciò stati disposti sulle pareti esterne in maniera da orientare la superficie sensibile parallelamente al muro. In via teorica i materiali esterni (intonaco e tegole) sono sensibili sia alla luce visibile (photosynthetically active radiation) che all'infrarosso e quindi

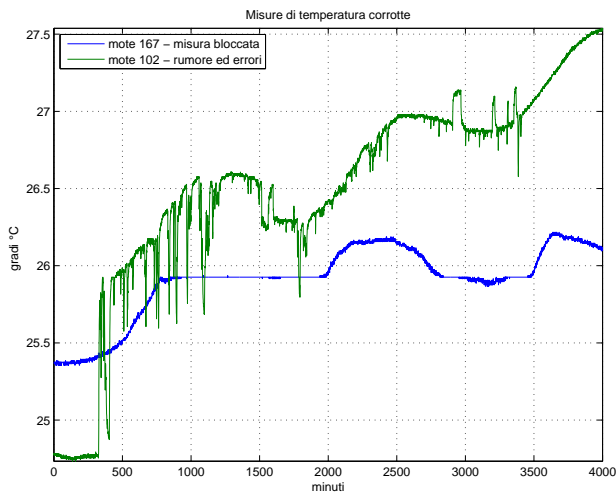


Fig. 35. Esempi di problemi di misurazioni riscontrati. In blu la misura di temperatura di un mote risulta bloccata, in verde una misura affetta da rumore molto marcato.

la lettura più significativa dovrebbe essere quella data dal sensore S1087-01 (total solar radiation), peraltro si è scelto di registrare entrambe le rilevazioni per adattare la scelta a eventuali esigenze incontrate nella fase di identificazione dei dati. La curva di sensibilità dei sensori è pressochè lineare al crescere della lunghezza d'onda (sono maggiormente sensibili a lunghezze d'onda maggiori e quindi con meno energia).

I sensori in esame forniscono un valore analogico che viene acquisito in via digitale dal microcontrollore; il sistema operativo TinyOS 2.x include tutte le routine di alto livello necessarie per la fase di lettura. I fattori di conversione in unità fisiche sono rispettivamente per il S1087

$$AR = 625 AR_{raw} \quad [\text{lux}] \quad (56)$$

e per il S1087-01

$$TR = 76.9 TR_{raw} \quad [\text{lux}] \quad (57)$$

a partire dai valori a 16 bit unsigned AR_{raw} e TR_{raw} ottenuti dal codice implementato in TinyOS. La misura lux indica un'unità di flusso luminoso (illuminamento) e cioè la potenza luminosa incidente su una superficie. Difatti un lux è un watt su metro quadrato e valori tipici spaziano dai 100000 lux in pieno sole ad un lux circa per l'illuminamento dato dalla luna piena. Il lux è peraltro una misura di fotogrammetria e non di radiometria e cioè, oltre a riguardare un campo di frequenze elettromagnetiche limitato e comprendente la luce visibile, ultravioletta ed infrarossa, include una pesatura sulla potenza di ogni diversa frequenza proporzionale alla curva di sensibilità tipica di un occhio umano. Non è pertanto propriamente corretto asserire che la potenza incidente su una facciata dell'edificio è proporzionale alla superficie ed all'illuminamento in quanto la curva di assorbimento del materiale costruttivo è diversa da quella usata dai sensori.

È interessante notare come avendo posizionato i dispositivi in pieno sole (protetti da appositi sacchetti in plastica trasparente) essi si sono trovati ad operare in condizioni di temperature abbastanza elevate: è plausibile supporre che i comparatori

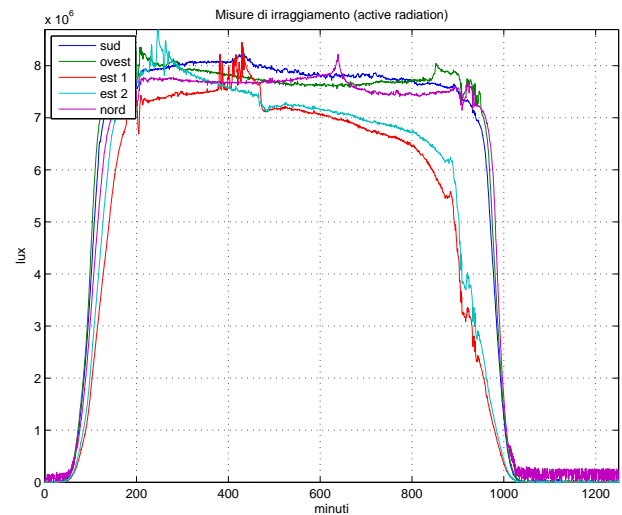


Fig. 36. Misura di luminosità per la durata di un giorno. Si nota che il lato ad est è meno irraggiato a causa della presenza di alberi ed edifici. Per gli altri sensori si nota una sostanziale identità di rilevazione il che porta a concludere come tali sensori non siano adatti a misurare in maniera precisa l'irraggiamento: il lato a nord (sempre in ombra) non riporta valori significativamente differenti rispetto a quelli a sud ed a ovest, probabilmente per la presenza di riflesso dovuto ai muri dirimpetto.

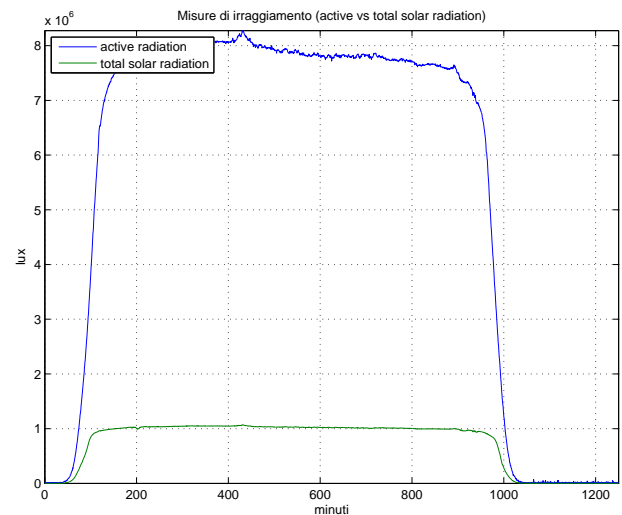


Fig. 37. Confronto tra photosynthetically active radiation e total solar radiation.

delle unità ADC siano stati affetti da un notevole rumore. Un altro problema evidenziato in fase di identificazione è che sensori posti su lati differenti abbiano un comportamento sostanzialmente analogo in quanto sensibili non solo alla luce diretta ma all'illuminazione ambientale complessiva. In tal modo non si è potuto avere un indicatore evidente di quando la facciata era in pieno sole o meno.

3) *Memoria flash*: Il mote dispone di un chip di memoria flash esterna di tipo seriale ST M25P80 di capienza 1024kb. Essa è configurata in lettura e scrittura solamente nei primi quindici settori da 65535 byte dei sedici che la compongono, essendo preimpostata in fabbrica nel sedicesimo settore una copia del bootloader e di sistema operativo minimi (golden image) per ripristinare il dispositivo in caso di errori. Di

conseguenza la memoria effettivamente disponibile per la rilevazione delle misure è di 983040 byte; la struttura dati scelta come unità minima di immagazzinamento è composta da 6 campi da 16 bit ciascuno il che porta ad una capacità teorica di 81920 campioni. Al fine di garantire un controllo di eventuali errori uno dei sei campi dati è un numero sequenziale aumentato ad ogni istante di campionamento al fine di evidenziare se eventuali campioni non fossero stati scritti sulla memoria; essendo tale campo a 16 bit viene posto un limite superiore alla quantità di dati salvati e pertanto la soluzione trovata permette di rilevare fino a 65535 campioni. Ipotizzando un tempo di campionamento di 30 secondi (ampiamente inferiore rispetto alle dinamiche termiche di un edificio - sarà infatti necessario sottocampionare in fase di identificazione) si ottiene una durata massima dell'esperimento di circa 22 giorni e 18 ore. Alla luce dell'effettiva disponibilità dell'edificio per le misurazioni e di altri fattori logistici si è riusciti a sfruttare meno della metà delle potenzialità di memoria.

B. Codice sviluppato

Il codice necessario per la raccolta dati è concettualmente molto semplice e non presenta difficoltà implementative dal punto di vista della potenza di calcolo necessaria. Per contro è stato necessaria un'attenta fase di testing per garantire l'affidabilità del processo di campionamento ed immagazzinamento dei dati, senza dimenticare l'apparato software sviluppato dal lato pc per la comunicazione e la gestione dei mote. L'idea di base è stata individuare tre distinte fasi di funzionamento del mote al fine di separare le porzioni di codice necessarie e di disattivare le funzionalità non più richieste. L'unità base di memorizzazione nella flash è una struttura dati del tipo

```
typedef nx_struct DataMsg{
    nx_uint16_t temp;
    nx_uint16_t hum;
    nx_uint16_t tsr;
    nx_uint16_t ar;
    nx_uint16_t volt;
    nx_uint16_t id;
} DataMsg_t;
```

la quale è utilizzata anche nella trasmissione via seriale (opportunamente incapsulata in un apposito pacchetto). I primi quattro campi si riferiscono alle misure ottenute dai sensori, il quinto riguarda il monitoraggio della tensione interna al processore e l'ultimo campo è un numero progressivo utile a rilevare errori in fase di trasmissione del contenuto della flash verso il pc.

Il codice complessivo risultante è di circa 24000 byte, la cui maggior parte è dovuta al codice per la gestione della funzione di logging su flash.

1) *Firmware*: Il programma caricato sui mote suddivide la sessione di acquisizione dati in tre fasi distinte: setup, sincronizzazione e acquisizione vera e propria. Il passaggio da una fase e l'altra è gestita da un timer (allo scadere del tempo prefissato si passa alla fase successiva) e da una semplice macchina a stati. In figura 38 è riportato uno schema di

principio delle funzioni del mote.

La prima fase è destinata alle operazioni di comunicazioni con l'utente via porta usb e pc: è stato implementato un protocollo minimale di scambio comandi ed informazioni gestito dal lato pc con un'applicazione in Java basata sulle librerie fornite nella distribuzione di TinyOS. Una volta che il bootloader ha caricato il firmware, la fase di setup può essere interrotta se dall'apposito programma dal lato pc viene mandato un comando di freeze. Da questo momento in poi il mote interrompe il normale flusso e resta in ascolto di eventuali richieste. Tra queste ultime si annoverano diverse funzioni di debug pensate per verificare lo stato del mote e della memoria flash ma soprattutto le due funzioni principali di cancellazione flash (in realtà più propriamente viene reinizializzato il log e impostato di lunghezza zero) e soprattutto di lettura flash.

La procedura di lettura flash consiste in una richiesta di lettura flash a cui viene risposto un ack al lato pc, per poi richiamare la funzione di libreria per portare a zero l'indice di lettura della flash. Una volta portata a termine tale operazione (tramite il meccanismo a callback di TinyOS) viene richiesta una lettura dalla memoria (in effetti la prima posizione del log, contraddistinta da `id = 0`). A seguito della corretta lettura il pacchetto di dati recuperato viene incapsulato in un pacchetto per la trasmissione seriale e trasmesso. L'ultimo callback, che decreta l'invio andato a buon fine, causa una nuova richiesta di lettura dalla flash dalla successiva posizione. L'invio dei dati si basa quindi su una catena di richieste a chiamate di sistema e successive conferme. Non è stato implementato un sistema basato sull'interscambio di richieste con il lato pc in quanto sono emersi numerosi problemi dal punto di vista della gestione dell'utilizzo fisico del canale usb. La debolezza del procedimento risiede principalmente nel fatto che non è possibile avere un feedback sull'effettiva ricezione dei dati da parte del programma Java; peraltro il supporto software della funzione di logging non permette di posizionare a piacimento la lettura dal log ma solo di gestire avanzamenti di una posizione alla volta. La strategia per risolvere eventuali errori di trasmissione (non rari se si impiegano cavi di prolunga usb) consiste nella ritrasmissione dell'intero contenuto del log in maniera tale da avere tutti i pacchetti. Una schematizzazione è riportata in figura 39.

Il passaggio alla seconda fase avviene allo scadere di un timer opportunamente impostato e causa la disabilitazione delle comunicazioni via usb; la durata della fase di sincronizzazione è a sua volta impostabile. Gli eventi che vengono rilevati dal mote sono la pressione del tasto user oppure la ricezione di un pacchetto radio (pacchetto di sincronizzazione). La pressione del tasto su un mote causa la partenza di un nuovo timer e l'invio del pacchetto di sincronizzazione; la ricezione di tale pacchetto da parte di altri mote causa a sua volta la partenza del medesimo timer. Una nuova pressione del bottone o una nuova ricezione del pacchetto causa l'azzeramento del timer (utile se qualche mote non ha ricevuto il pacchetto). Allo scadere di esso vengono effettuate le operazioni di azzeramento del log su flash e, tramite un ulteriore timer, viene programmata la partenza del campionamento.

La terza fase di lavoro è la fase di acquisizione dati: in tale fase il modulo radio viene disattivato e un timer periodico (a cui

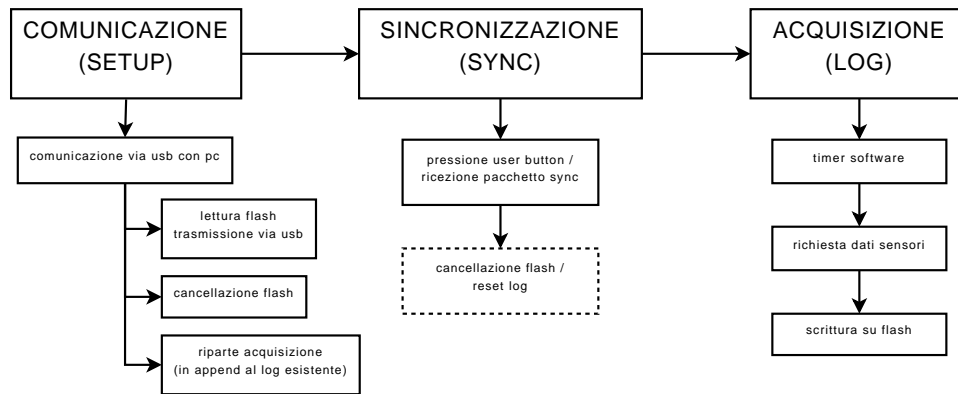


Fig. 38. Diagramma schematico delle funzionalità richiamate in ogni stato del mote.

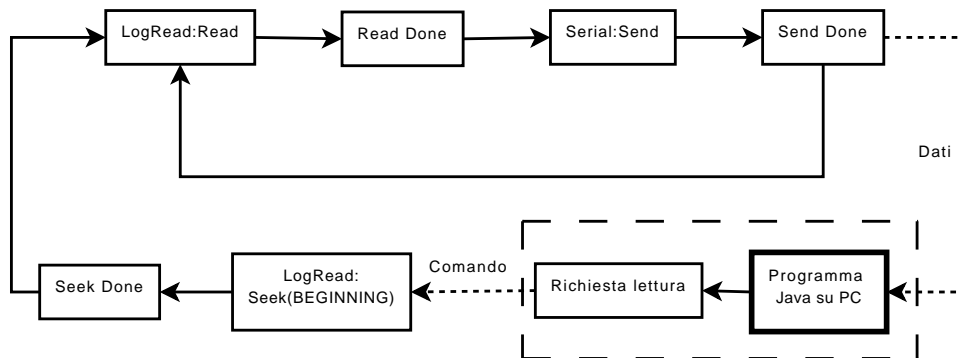


Fig. 39. Diagramma schematico della procedura di lettura del mote.

è assegnata la durata del tempo di campionamento) comanda la lettura dei valori registrati dai sensori, che vengono salvati tramite il consueto meccanismo a callback. Allo scadere del successivo istante di campionamento i dati precedentemente letti vengono scritti su flash (se la lettura era andata a buon fine e se la flash è disponibile ad accettare dati, in quanto il protocollo spi impiegato è condiviso con altri chip) e viene comandata una nuova lettura. In questa fase il processore è in idle per la maggior parte del tempo (con tempi di campionamento ragionevolmente lunghi) e lo scheduler dei task di TinyOS attiva automaticamente la modalità risparmio energetico più conveniente (esistono diverse modalità in relazione a che risorse il sistema operativo reputa ancora in uso). Tutte le variabili impostabili dall'utente sono riportate in un apposito file `.h` ed opportunamente commentate.

2) *Interfaccia PC*: La gestione del mote è affidata ad un apposito programma sviluppato in Java su piattaforma Linux impiegando le librerie di comunicazione seriale e le classi per la gestione dei pacchetti fornite con TinyOS 2.x. Il programma è stato sviluppato sia per la lettura finale dei dati che per il testing del sistema e quindi contiene molteplici comandi utili a verificare ogni singola funzione del firmware. Allo scopo sono state progettate alcune funzioni utili al debug (il mote può inviare al pc un pacchetto contenente dati quali interi o stringhe di caratteri) assieme ad un sistema rudimentale di invio comandi al mote. Quest'ultimo è completamente integrato nel firmware e nel programma java ed è quindi necessario riportare ogni nuova funzionalità su entrambi i sorgenti.

La presenza di un buffer addizionale assicura la ricezione e l'invio dei comandi. La comunicazione con il mote è possibile solamente durante la fase di setup del mote. Il file di output generato dalla lettura è un file di testo (importabile in ambiente Matlab) con i valori raw dei sensori (la trasformazione in unità ingegneristiche è affidata ad appositi script Matlab che permettono inoltre una comoda rappresentazione in grafici dei valori ottenuti). È presente un help integrato al programma per i comandi base.

Per la lettura e la programmazione di una grande quantità di dispositivi è stato inoltre scritto uno script in Python che tramite l'impiego di thread permette di gestire in parallelo più mote. La programmazione avviene avviando lo script con l'apposita opzione e inserendo i mote in una porta usb (viene automaticamente lanciato il comando apposito); all'uscita dallo script viene riportato il numero totale di mote programmati con successo. La lettura avviene lanciando lo script con l'opzione apposita: automaticamente viene letto il numero seriale del mote che verrà impiegato come nome del file di output del programma in java illustrato precedentemente. A causa di problemi intrinseci al software Java e soprattutto alla libreria fornita la gestione in real time dei dati in arrivo a volte può fallire ed è preferibile non effettuare la lettura contemporanea di più dispositivi. Grazie ad un controllo interno è comunque garantito che (a meno di diraggiungere un timeout impostabile) il trasferimento dei dati dal mote al pc venga ripetuto sintanto che tutti i campioni sono stati salvati su file.

A titolo di esempio i tempi riscontrati sono stati per la programmazione di 65 mote circa venti minuti e per la lettura all'incirca tre ore per dieci giorni di acquisizione con tempo di campionamento di trenta secondi (oltre trentamila campioni per mote).

X. CONCLUSIONI

L'approccio seguito per il problema di identificazione in esame ha portato diversi frutti: è stato possibile ricavare un modello dinamico lineare tempo invariante stabile in grado di raccogliere la maggior parte della dinamica rilevata sul sistema fisico reale. L'affinamento tramite un filtro di Kalman tempo variante ha permesso di ottenere un modello numerico in grado di stimare con precisione inferiore al decimo di grado la temperatura dei sensori impiegando un insieme di ingressi misurati di un ordine di grandezza inferiore, superando con successo l'obiettivo minimo di stimare i dati con una precisione inferiore al mezzo grado centigrado. Si è difatti dimostratosi possibile stimare 54 misure con un numero di ingressi inferiore alla decina: risultati soddisfacenti sono stati raggiunti già con soli sei ingressi.

La metodologia di lavoro ha sofferto inizialmente per un difetto nella definizione del problema di identificazione, in quanto solitamente tale fase è intrapresa con una chiara motivazione degli scopi da raggiungere (tipicamente il controllo del sistema). Una volta definiti gli obiettivi minimi si è proceduto muovendosi su più fronti: trattandosi di un sistema fisico si è potuto modellizzarlo in maniera precisa a partire dai principi base ed utilizzare poi i dati generati in tale modo per passare alla fase di identificazione, la quale a sua volta ha richiesto l'introduzione di strumenti teorici avanzati per gestire la quantità di dati in gioco. La scelta del sottoinsieme ottimale di sensori si è a sua volta rivelato un problema di ottimizzazione combinatoria e quindi attualmente non risolvibile in tempi polinomiali: grazie alla definizione di alcune metriche opportune si è giunti alla formulazione di alcune efficaci tecniche euristiche per ottenere il risultato desiderato.

Una direzione futura in cui sarebbe possibile muoversi è la visione del sistema dal punto di vista del controllo, cioè assumendo che alcuni ingressi siano controllabili e quindi di poter influenzare il comportamento termodinamico dell'edificio. Con lo stesso obiettivo si potrebbe inserire all'interno del simulatore la dinamica di alcuni termoregolatori ed utilizzarlo per testare eventuali algoritmi di controllo. D'altra parte sarebbe importante disporre di dati sufficienti a coprire un periodo dell'anno più vasto (e possibilmente con più variazione durante le giornate) al fine di identificare in maniera più affidabile il sistema lineare.

BIBLIOGRAFIA

- [1] Subspace identification methods, Katrien De Cock, Bart De Moor, K.U. Leuven Department of Electrical Engineering http://homes.esat.kuleuven.be/~decock/publications/subspace_methods.pdf
- [2] Subspace Identification for Linear Systems: Theory, Implementation, Applications, Peter Van Overschee and Bart De Moor, Kluwer Academic Publishers, 1996 <ftp://ftp.esat.kuleuven.be/pub/sista/nackaerts/other/alln.ps.gz>
- [3] The role of vector autoregressive modeling in predictor-based subspace identification, Alessandro Chiuso, Automatica (Journal of IFAC) Volume 43, Issue 6 (June 2007)
- [4] An overview of subspace identification, Alessandro Chiuso, http://www.dei.unipd.it/~chiuso/DOWNLOAD/Talk_KTH.pdf
- [5] System Identification, Torsten Söderström, Petre Stoica, Prentice Hall, 1989
- [6] Fitraggio Statistico (Wiener, Levinson, Kalman) e Applicazioni, Giorgio Picci, Libreria Progetto, 2006
- [7] A review of methods for input/output selection, Marc van de Wal, Bram de Jager Automatica (Journal of IFAC) Volume 37, Issue 4 (April 2001)
- [8] SHT1x / SHT7x Humidity & Temperature Sensor Datasheet, v3.00 March 2007, Sensirion AG http://www.sensirion.com/en/pdf/product_information/Data_Sheet_humidity_sensor_SHT1x_SHT7x_E.pdf
- [9] RH and Temperature Non-Linearity Compensation, rev. 1.4, Sensirion AG http://www.sensirion.com/pdf/product_information/Non-Linearity_Compensation_Humidity_Sensors_E.pdf
- [10] Getting data from Tmote Sky, MoteIV Corporation http://www.moteiv.com/community/Getting_Data_from_Tmote_Sky's_Sensors
- [11] S1087/S1087-01 Datasheet, Hamamatsu Corporation http://sales.hamamatsu.com/assets/pdf/parts_S/S1087_etc.pdf
- [12] Algoritmi euristici, Michele Monaci, dispense per il corso di ricerca operativa 2, recuperato il 21 luglio 2007 <http://www.dei.unipd.it/~monaci/ricop2.htm>
- [13] Algoritmi genetici (GA-Genetic Algorithm) etd.adm.unipi.it/theses/available/etd-09192005-163646/unrestricted/04_CAPITOLO2.pdf
- [14] Algoritmi euristici:III – Ricerca Locale, Daniele Vigo, D.E.I.S. - Università di Bologna, revisione 1.0 dicembre 2003 www.or.deis.unibo.it/didatt_pages/labos_ce/euristici-III-ricerca%20locale.pdf
- [15] System identification applied to building performance data, J. J. Bloem, Institute for system engineering and informatics, Joint research center european commission 1994
- [16] Foudamentals of the finite element method for heat and fluid flow, Roland W. Lewis, Perumal Nithiarasu and Kankanhalli N. Seetharamu, Wiley 2004
- [17] An automatic tetrahedral mesh generation scheme by the advancing front method, C. T. Chan and K. Anastasiou, communications in numerical methods in engineering Vol 13: 33-46 (1997)
- [18] A simple mesh generator in matlab, Per-olof Persson and Gilbert Strang
- [19] Solar position algorithm for solar radiation application, Reda, I., Andreas, A. National Renewable Energy Laboratory (NREL) Technical report NREL/TP-560-34302 (2003); un'implementazione in Matlab di Vincent Roy è disponibile presso Matlab Central File Exchange