



Presentazione del progetto di:

Identificazione termodinamica di un edificio con tecniche PCA e PLS

Bonollo Federico, Cattai Daniele, Verga Marco

Nel precedente progetto

- Rilevazioni fisiche delle grandezze.
- Identificazione del modello dell'edificio.
- Selezione dei sensori con algoritmi greedy, ricerca locale e genetici.
- Metriche di validazione: fit del modello e traccia varianza dell'errore.
- Selezione delle uscite del modello per la predizione tempo variante di tutti gli andamenti.
- Criterio di scelta dei sensori a “posteriori”.

I nuovi obiettivi del progetto

- Analizzare l'utilizzo di software di simulazione all'interno del contesto di identificazione del modello dell'edificio.
- Determinare nuove tecniche per aumentare la bontà del modello dinamico lineare atto alla descrizione opportuna della distribuzione di temperatura interna nel tempo e nello spazio.
- Ulteriori criteri di validazione: MSE, test di bianchezza, test di indipendenza, stima funzione di covarianza dell'errore.
- Le tecniche PCR e PLS ricercano la combinazione ottima di ingressi e uscite?

L'identificazione a sottospazi

Classe di algoritmi numerici per l'identificazione di sistemi dinamici lineari a tempo discreto, di dimensione finita, tempo invarianti con ingressi stocastici.

Vantaggi: identificazione di sistemi MIMO di grandi dimensioni, in luogo della soluzione PEM computazionalmente proibitiva.

$$\begin{cases} x(t+1) &= A x(t) + B u(t) + w(t) \\ y(t) &= C x(t) + D u(t) + v(t) \end{cases}$$

ipotesi

Controllabilità della coppia $A, [BQ^{1/2}]$

Osservabilità coppia A, C

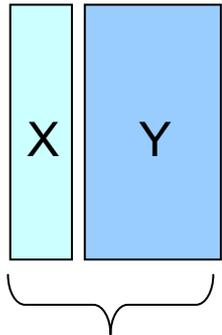


Tutti i modi del sistema eccitati da ingresso stocastico o deterministico



Tutti i modi osservabili dall'uscita y quindi identificabili

Fasi



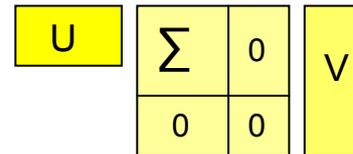
I/O organizzati come matrici di Hankel a blocchi



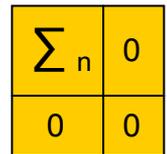
Costruzione di una base per lo spazio di stato attraverso una opportuna proiezione (ortogonale o oblique)



fattorizzazione SVD



Stima ordine n associata ai valori singolari dominanti



Ricostruzione stati e determinazione della matrice di osservabilità estesa



Risolvere ai minimi quadrati "least squares" il sistema:

$$\begin{cases} \hat{X}_{t+1} \simeq A\hat{X}_t + B\hat{U}_t + KE_t \\ Y_t \simeq C\hat{X}_t + E_t \end{cases}$$



Matrici A, B, C, D



Matrici Q, S, R dai residui

Metodi a sottospazi : N4SID Vs PBSID

N4SID

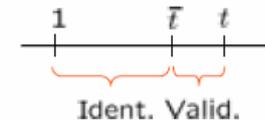
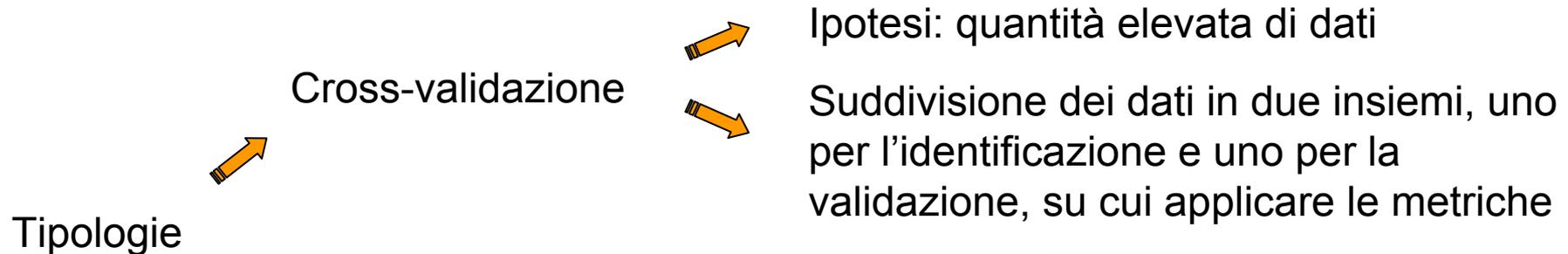
- possibilità di matrice A instabile.
- molto oneroso e lento.
- stima ordine alta.
- problemi gestione feedback.
- tecnica analizzata e non sfruttata in questo progetto.

PBSID e PBSIDopt

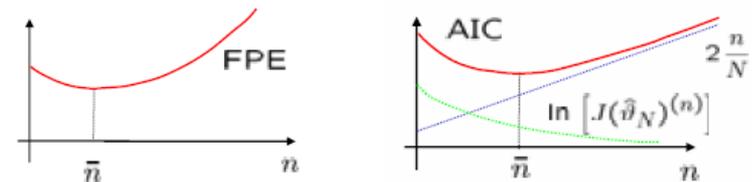
- computazionalmente più veloce.
- identifica modelli di ordine ridotto.
- sistemi asintoticamente stabili.
- consistente.
- no effetto bias.
- opera in presenza di feedback (come in questo progetto)
- soluzione di minimizzazione iterativa per la ricerca del predittore.

La validazione

La validazione è la fase in cui si verifica la bontà del modello identificato, valutandone il comportamento in catena aperta e la capacità predittiva.



Fpe e Aic: criteri in cui si stima l'ordine del modello rispetto alle diverse possibili realizzazioni dei dati o del modello stesso. Possibile metrica nel caso i dati siano pochi.



Metriche di validazione

Test di bianchezza: stima la funzione di correlazione del residuo, se questo è bianco allora il modello identificato è una buona descrizione della serie temporale.

Ip: $\epsilon(t)$ sia bianco

$$\hat{r}_\epsilon(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \epsilon(t+\tau)\epsilon(t) \xrightarrow{N \rightarrow \infty} \begin{cases} \hat{r}_\epsilon(\tau) \rightarrow 0 & \tau \neq 0 \\ \hat{r}_\epsilon(0) \rightarrow \lambda^2 = E\epsilon^2(t) \end{cases}$$

Test di indipendenza: analizza la stima della correlazione tra i residui e gli ingressi, fornendo esito positivo qualora:

$$r_{\epsilon u}(\tau) = E\epsilon(t+\tau)u(t) = 0$$

Entrambe le analisi vengono effettuate dalla funzione **resid** di Matlab, con le seguenti q.tà normalizzate:

$$x_\tau = \frac{\hat{r}_\epsilon(\tau)}{\hat{r}_\epsilon(0)} \quad x_\tau = \frac{\hat{r}_{\epsilon u}(\tau)}{[\hat{r}_\epsilon(0)\hat{r}_u(0)]^{1/2}}$$

Metriche di validazione

MSE: Errore quadratico medio “*mean square error*”

$$\begin{aligned}
 MSE &= \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \\
 &= \text{Var}\{\hat{y}\} + \text{mean}^2(\hat{y} - y)
 \end{aligned}$$

Fit: Percentuale di varianza spiegata

$$\text{fit} = 100 * \left[1 - \frac{\text{norm}(y_h - y)}{\text{norm}(y - \bar{y})} \right] \implies \begin{array}{l} y_h \text{ uscita di modello} \\ y \text{ uscita di validazione} \end{array}$$

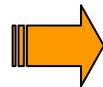
Il fit è calcolato e graficato dalla funzione **compare** di Matlab sia per la catena aperta che per la predizione.

DesignBuilder

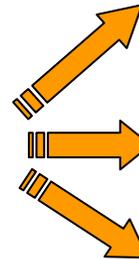
DesignBuilder è un simulatore per analisi energetiche di edifici semplici o complessi. Si basa sul motore di simulazione energetica *“EnergyPlus”* il quale fa riferimento alle normative CEN-Umbrella all’interno del direttivo EPBD *“Energy Performance of Buildings Directive”*.

Utilizzo del simulatore:

Generazione di diversi set di esperimenti



Identificazione



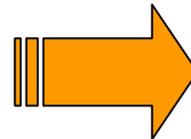
In quale periodo dell’anno si identifica un buon modello ?

Periodo e campionamento minimo per identificazione ?

Validità di un modello con altri set di esperimenti ?

Set di uscite: singolo andamento di temperatura per stanza.

Set di ingressi: Temperatura esterna, umidità, irraggiamento solare normale alla sup. e diffusa.

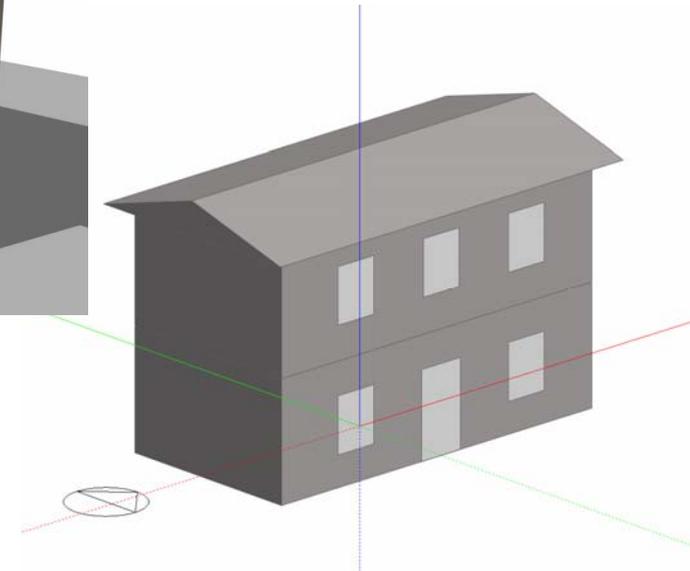
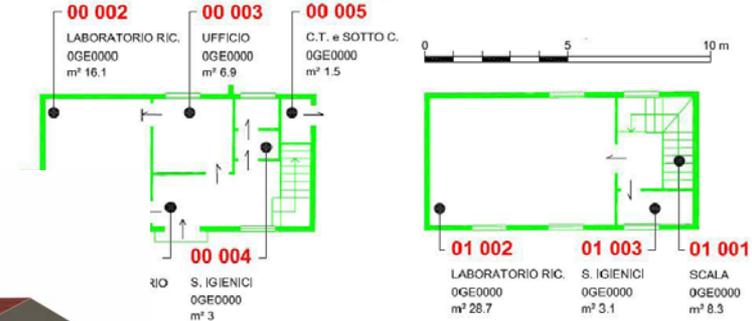


Limitazione nelle prove di selezione I/O manuali

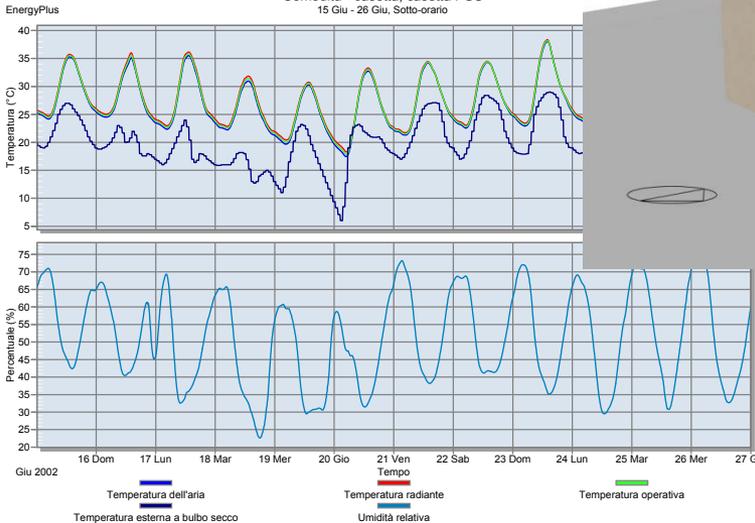
Dal reale al simulato ...



Complesso vecchio di Agraria, ex presidenza - vicolo Portello



Comodità - casetta, casetta PSC
15 Giu - 26 Giu, Sotto-orario

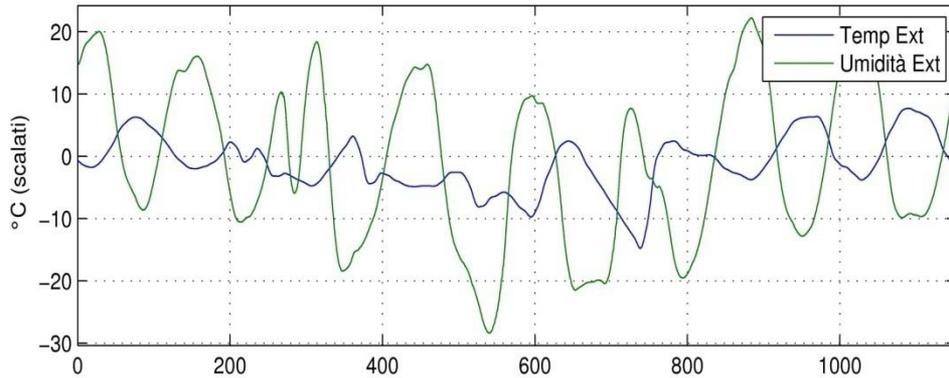


I modelli identificati

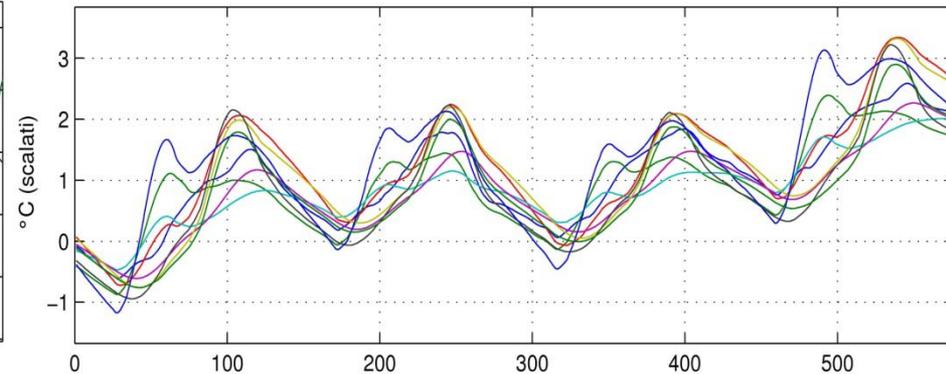
Modello estivo:

Ingressi: T_e + umidità

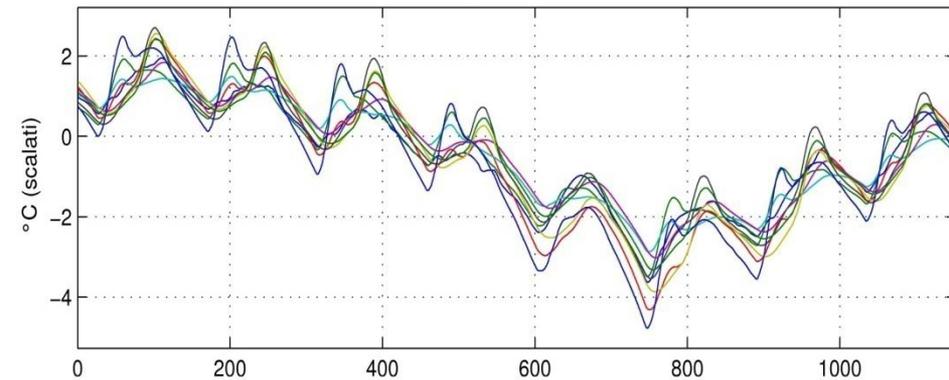
Ingressi per identificazione



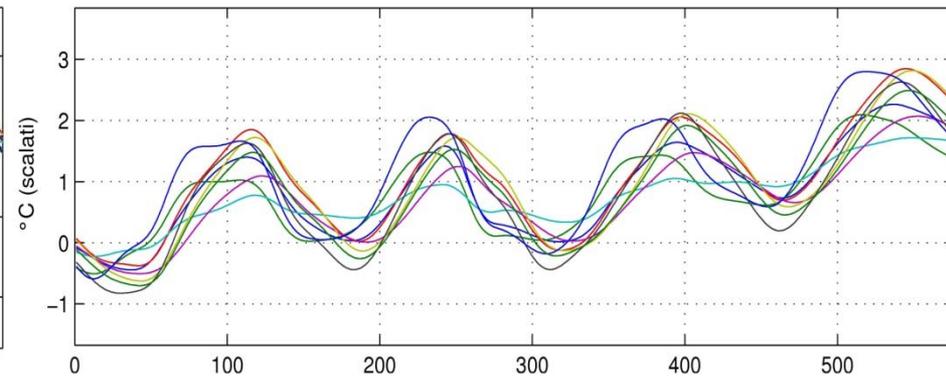
Uscite di validazione (da simulatore)



Uscite per identificazione



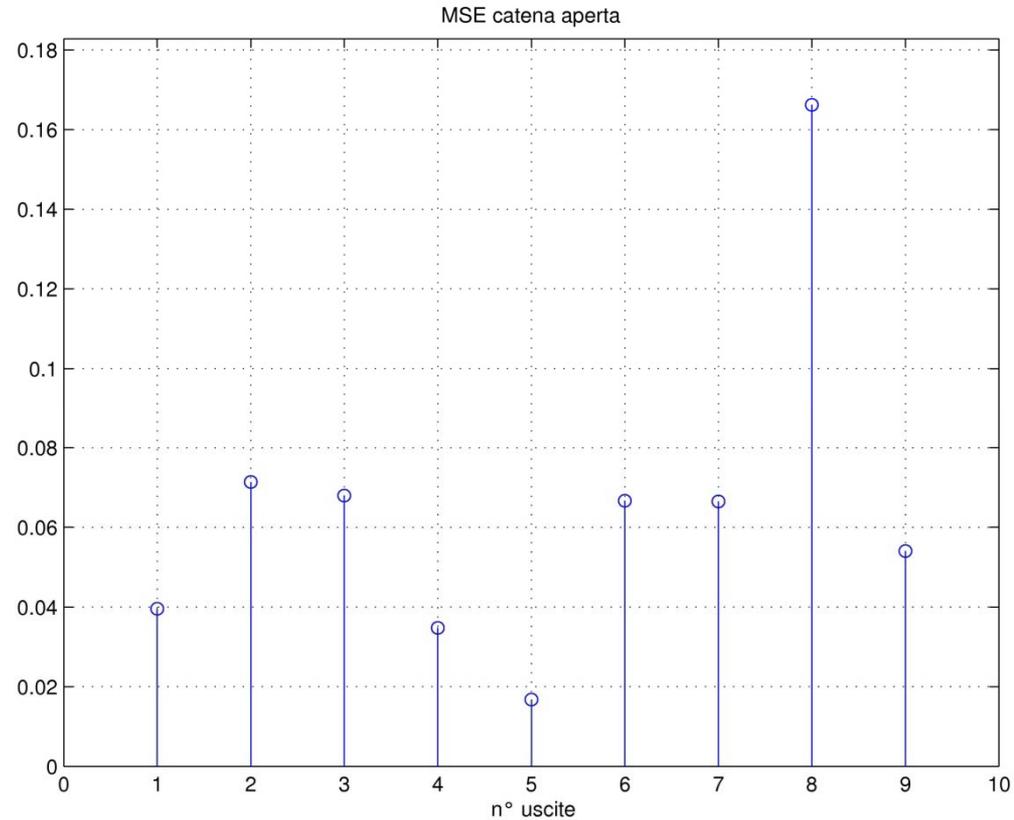
Uscite del modello identificato con gli ingressi di validazione



Modello estivo

Ingressi: T_e + umidità

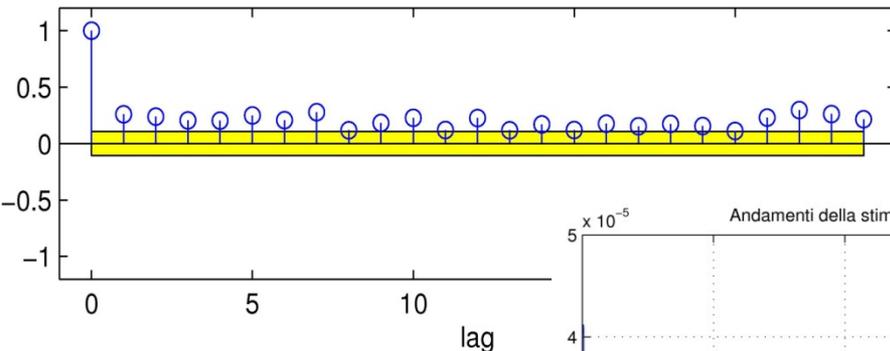
uscita n°	stanza	fit [%]	MSE [10^{-3}]
1	centrale termica	83,46	39,5
2	ingresso e atrio	72,98	71,4
3	laboratorio ricerca 2° piano	82,57	68,0
4	laboratorio ricerca 1° piano	77,13	34,7
5	pre-wc	88,66	16,7
6	scala	82,8	66,7
7	ufficio	77,59	66,5
8	bagno 2° piano	69,79	166,2
9	bagno 1° piano	77,96	54,1



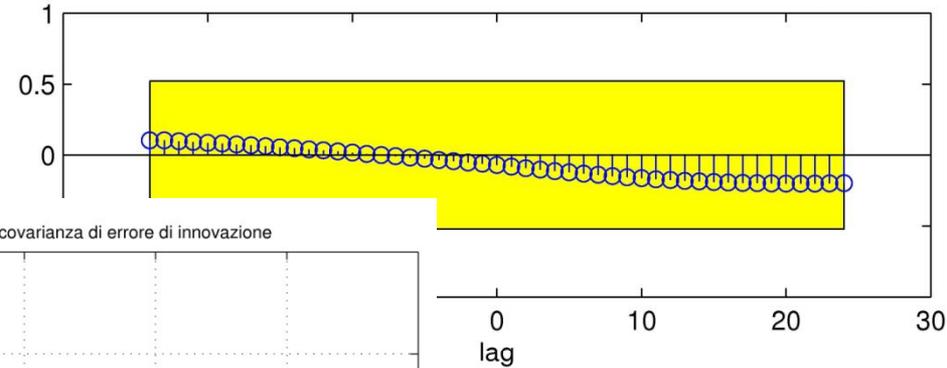
Modello estivo

Ingressi: T_e + $umidità$

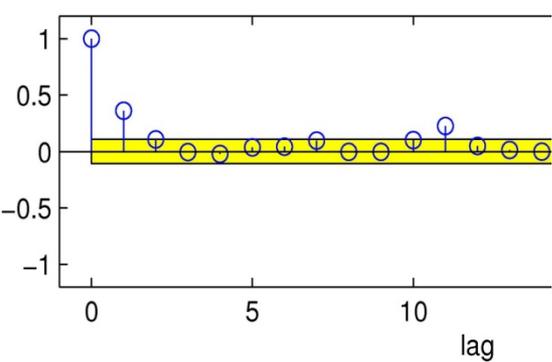
Correlation function of residuals. Output y1



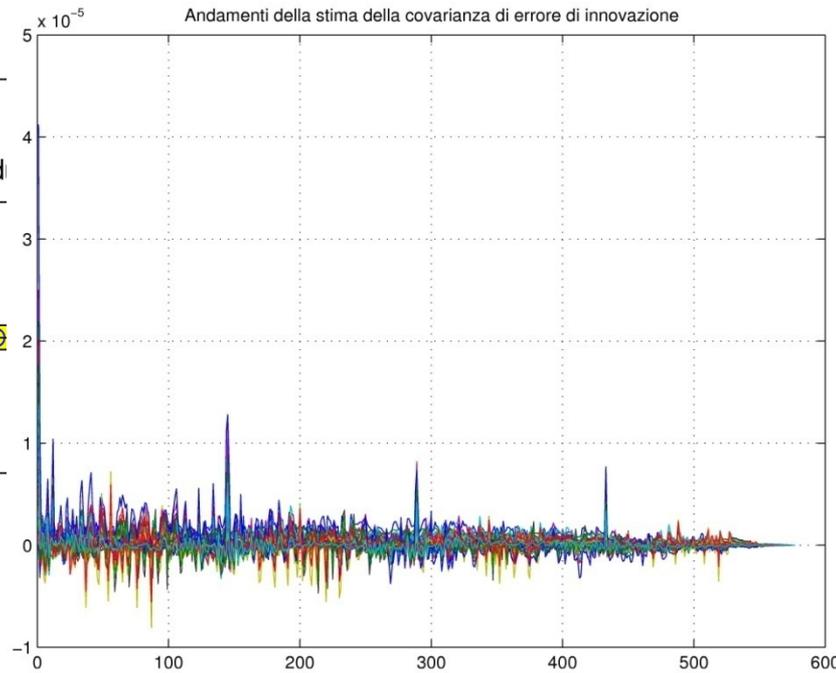
Cross corr. function between input u2 and residuals from output y5



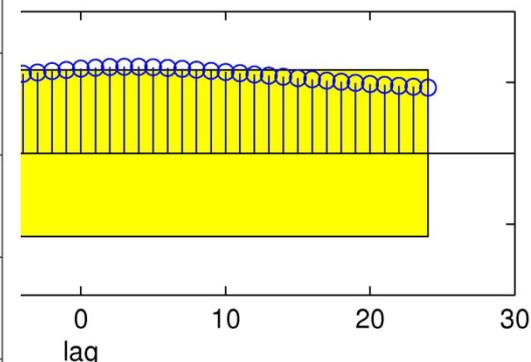
Correlation function of resid



Andamenti della stima della covarianza di errore di innovazione



input u1 and residuals from output y6



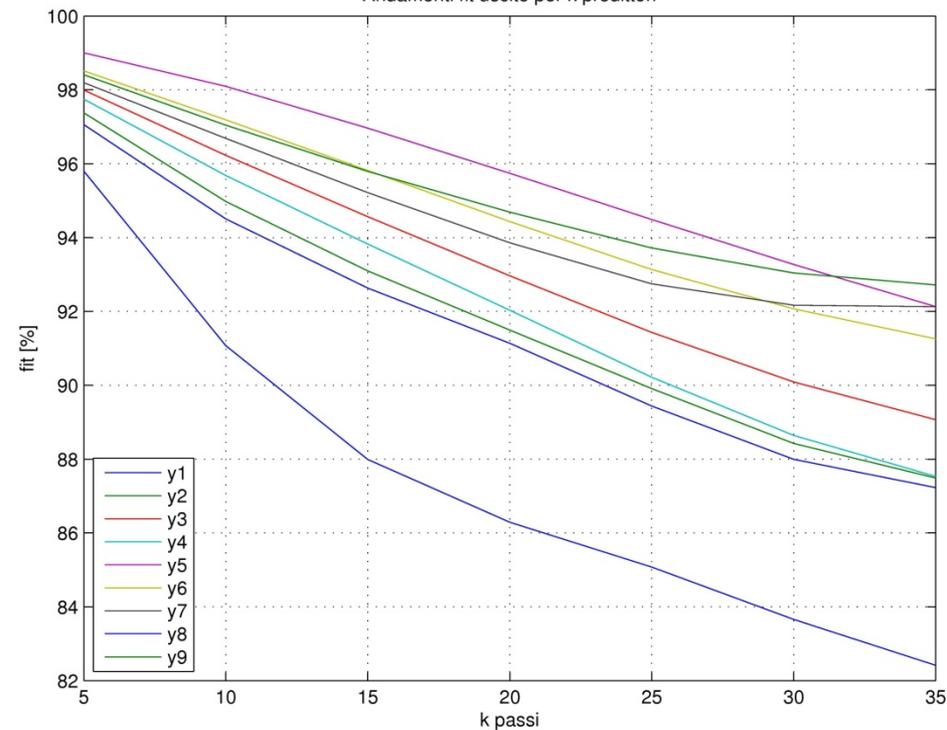
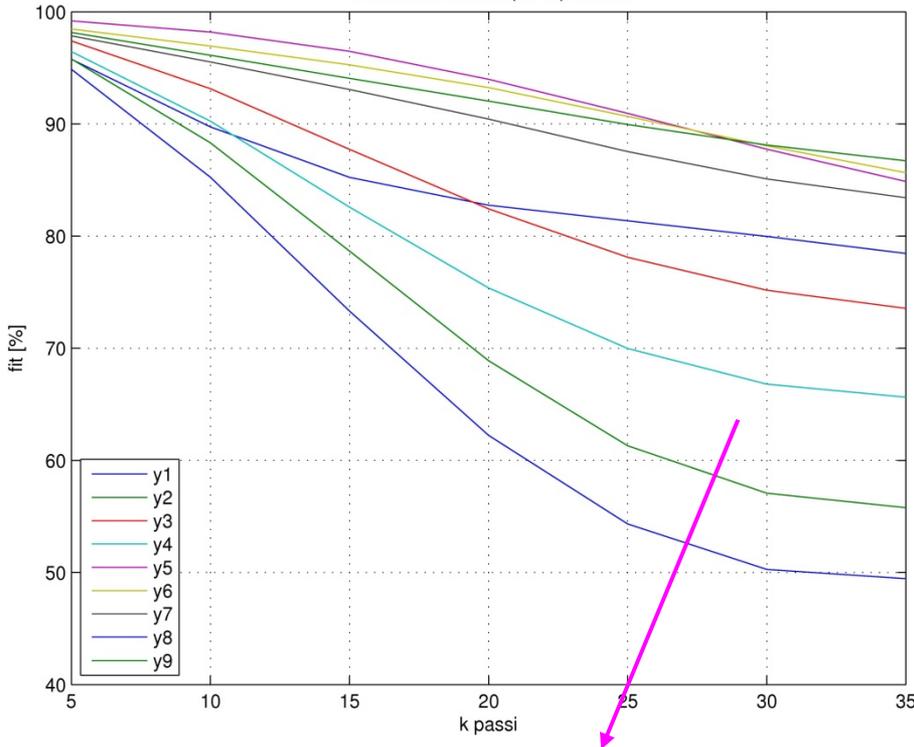
Modello estivo

Ingressi: T_e + umidità

Ingressi: T_e + umidità + irr. solare normale + irr. solare diffuso

Andamenti fit uscite per k predittori

Andamenti fit uscite per k predittori



Peggiori fit per le stanze affacciate a sud, sottoposte a maggiori oscillazioni termiche

Modello estivo

Parametri qualitativi:

Periodo per il campionamento < 10 min

Se > 10 min



Instabilità modello

Quantità minima di dati per una buona identificazione

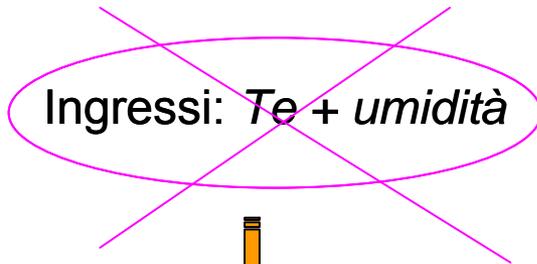


> 500 campioni

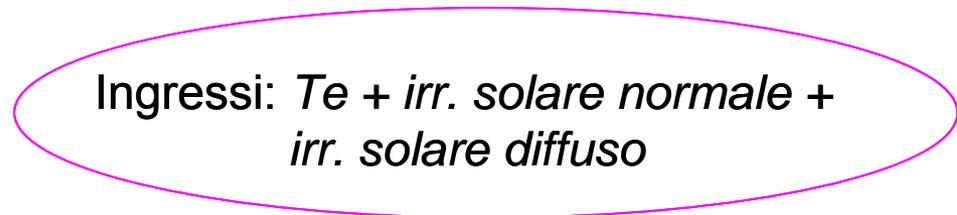


2 settimane sono sufficienti per identificare localmente il modello

Scelta ingressi:



Migliore in catena aperta



Migliore in predizione

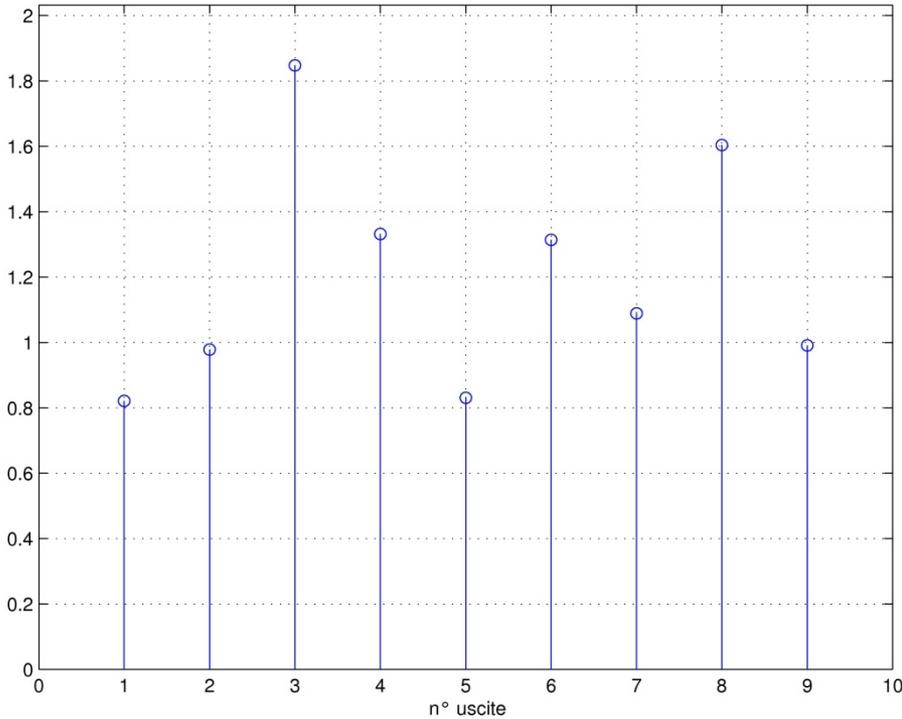
Modello invernale

Situazione anomala con ingressi $T_e + umidità$

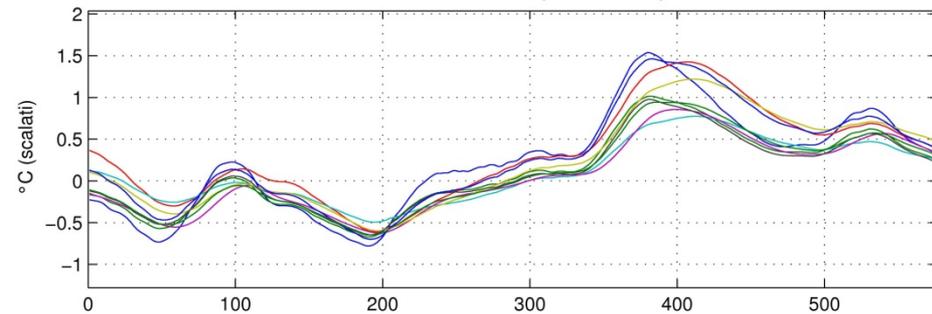


Necessario identificare con $T_e + irraggiamento$

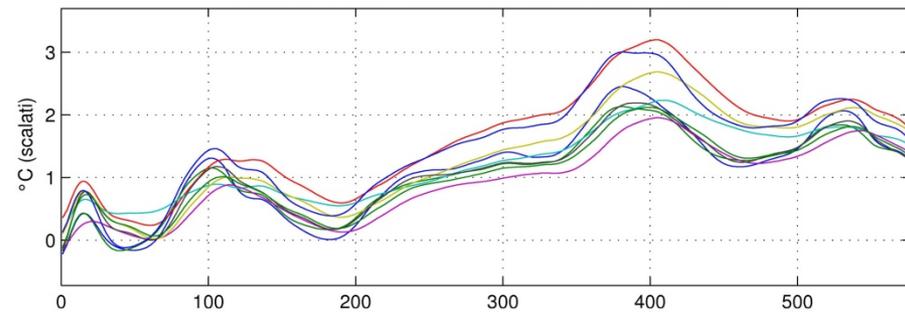
MSE catena aperta



Uscite di validazione (da simulatore)



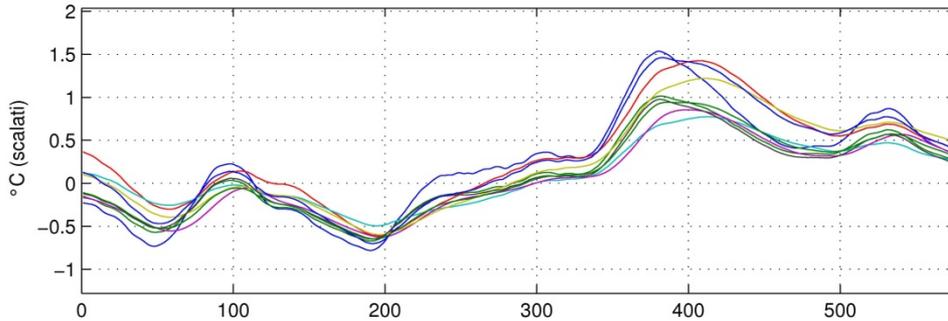
Uscite del modello identificato con gli ingressi di validazione



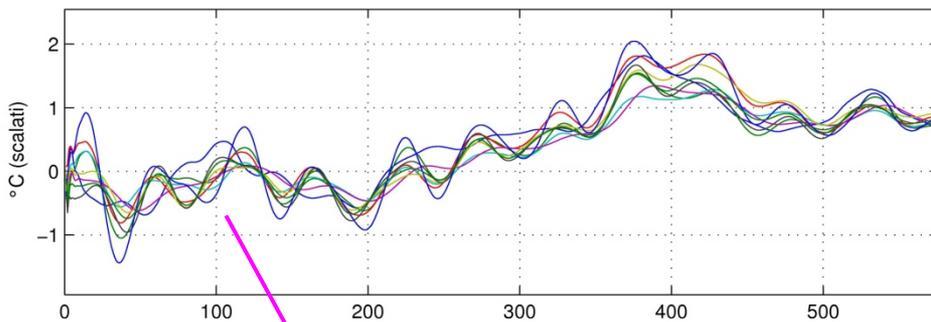
Modello estivo con set invernali

Ingressi: T_e + umidità + irr. solare normale + irr. solare diffuso

Uscite di validazione (da simulatore)

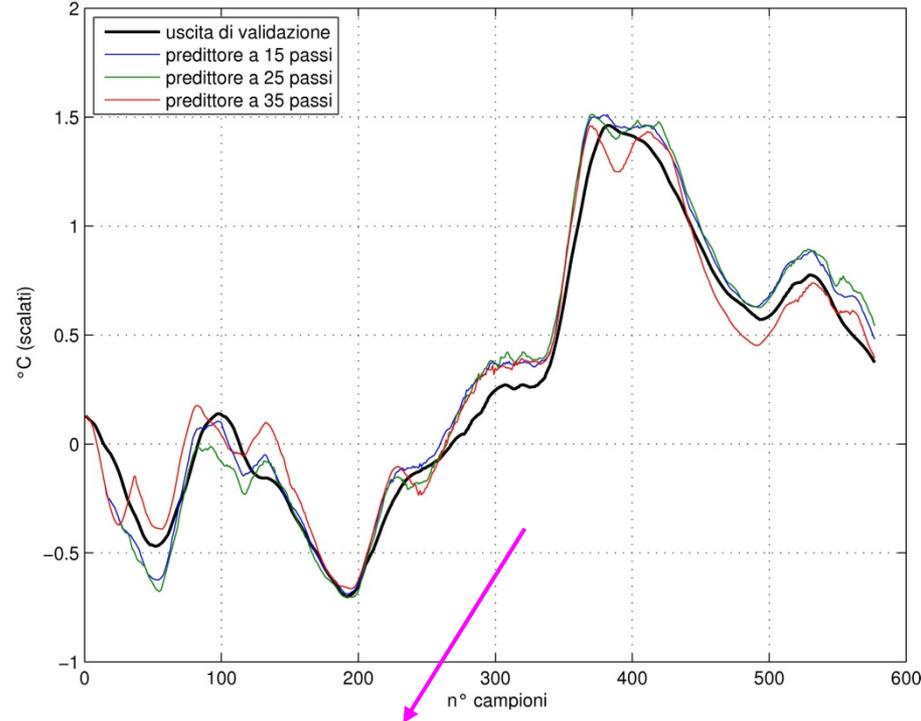


Uscite del modello identificato con gli ingressi di validazione



Fenomeni oscillatori
in catena aperta

Confronto validazione e predittori per la 8 uscita



Ottima predizione

Conclusioni simulatore:

Conclusioni:

- utile nel fornire diversi set di esperimenti
- validità “locale” dei modelli identificati
- non è sfruttabile per un’analisi della selezione delle uscite, essendo limitate a una per stanza.
- sfruttabile per un’analisi qualitativa e preventiva alla fase di raccolta dati di un edificio.

Sviluppi futuri:

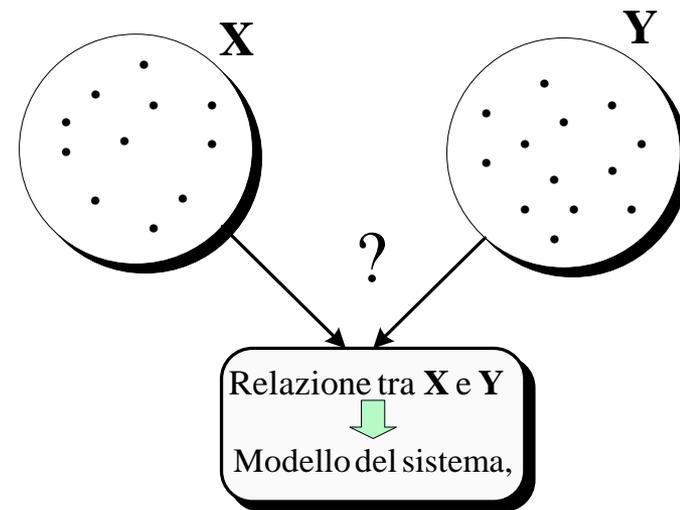
- come interfacciare un controllo con il simulatore.
- auspicabile confronto con altri simulatori, tali da fornire maggiori informazioni e dettagli di costruzione.

La Regressione Lineare

Ci poniamo nella condizione di avere due *collezioni* di dati \mathbf{X} e \mathbf{Y}

\mathbf{X} : sensori di temperatura esterni, luminosità
Total solar Radiation

\mathbf{Y} : sensori di temperatura interni

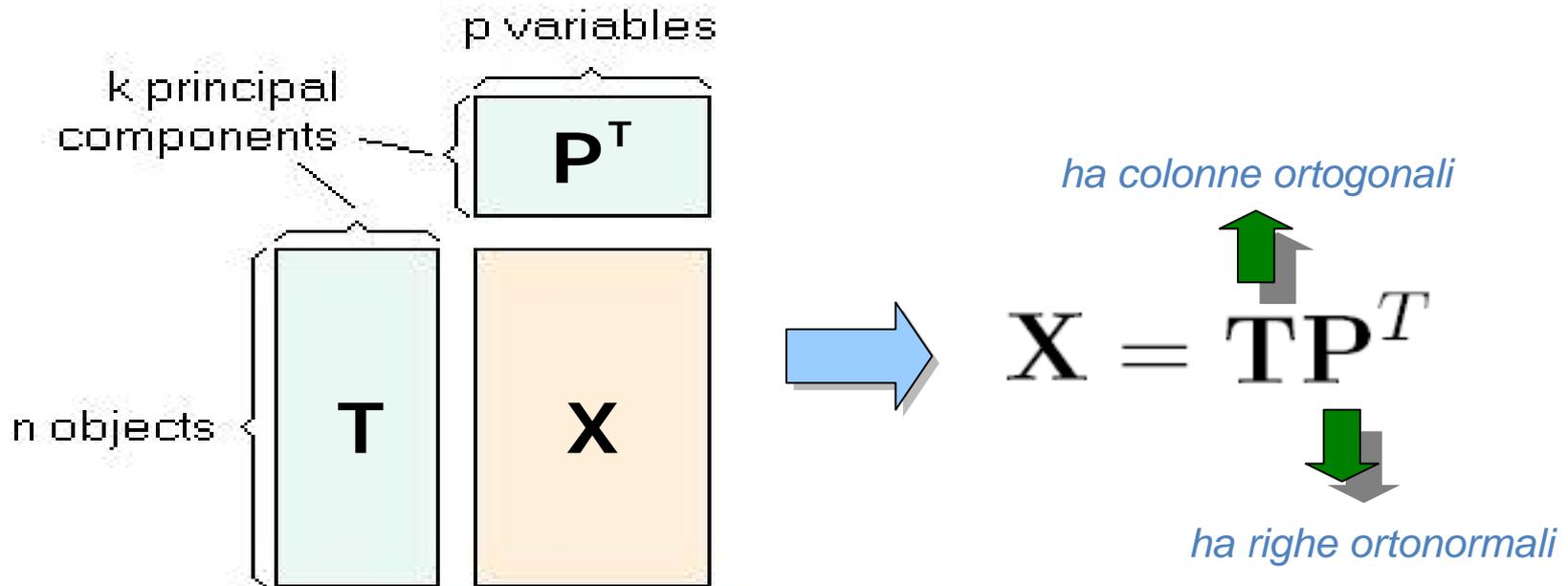


minimi quadrati  collinearità dei dati di ingresso



*Analisi delle
Componenti Principali*

PCA



$$\mathbf{P} = [1, \dots, p_a, \dots, p_A]$$

Loading : *proiettori dei dati lungo le componenti principali*

$$\mathbf{T} = [t_1, \dots, t_a, \dots, t_A]$$

score : *proiezioni dei dati lungo le direzioni di maggior rilevanza*

Single Value Decomposition

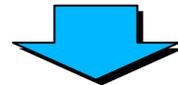
ortogonali : $M^T \cdot M = I$

$\hookrightarrow \underline{M^{-1} = M^T}$

$$X = U D P^T$$

diagonale : $d_1 \geq \dots \geq d_t \geq 0$

esist e tale che $d_t = 0$



$$X = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \cdot \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix} \simeq \underbrace{U_1 \cdot D_1}_{T_1} \cdot P_1^T$$

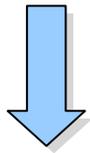
EIG decomposition

$$\begin{array}{c}
 \text{ortogonalità} \\
 \downarrow \\
 \mathbf{X}^T \mathbf{X} = \mathbf{P} \mathbf{D} \mathbf{U}^T \mathbf{U} \mathbf{D} \mathbf{P} = \mathbf{P} \mathbf{D}^2 \mathbf{P} \\
 \uparrow \quad \downarrow \updownarrow \\
 \text{SVD} \quad \quad \quad = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^T
 \end{array}$$

si è trovato un legame tra la decomposizione SVD e la decomposizione autovalori-autovettori, che per i nostri scopi hanno quindi una funzionalità equivalente

Scelta del numero di PC

$$peso(\lambda_i) = \frac{\lambda_i}{\lambda_1 + \dots + \lambda_r} \times 100$$



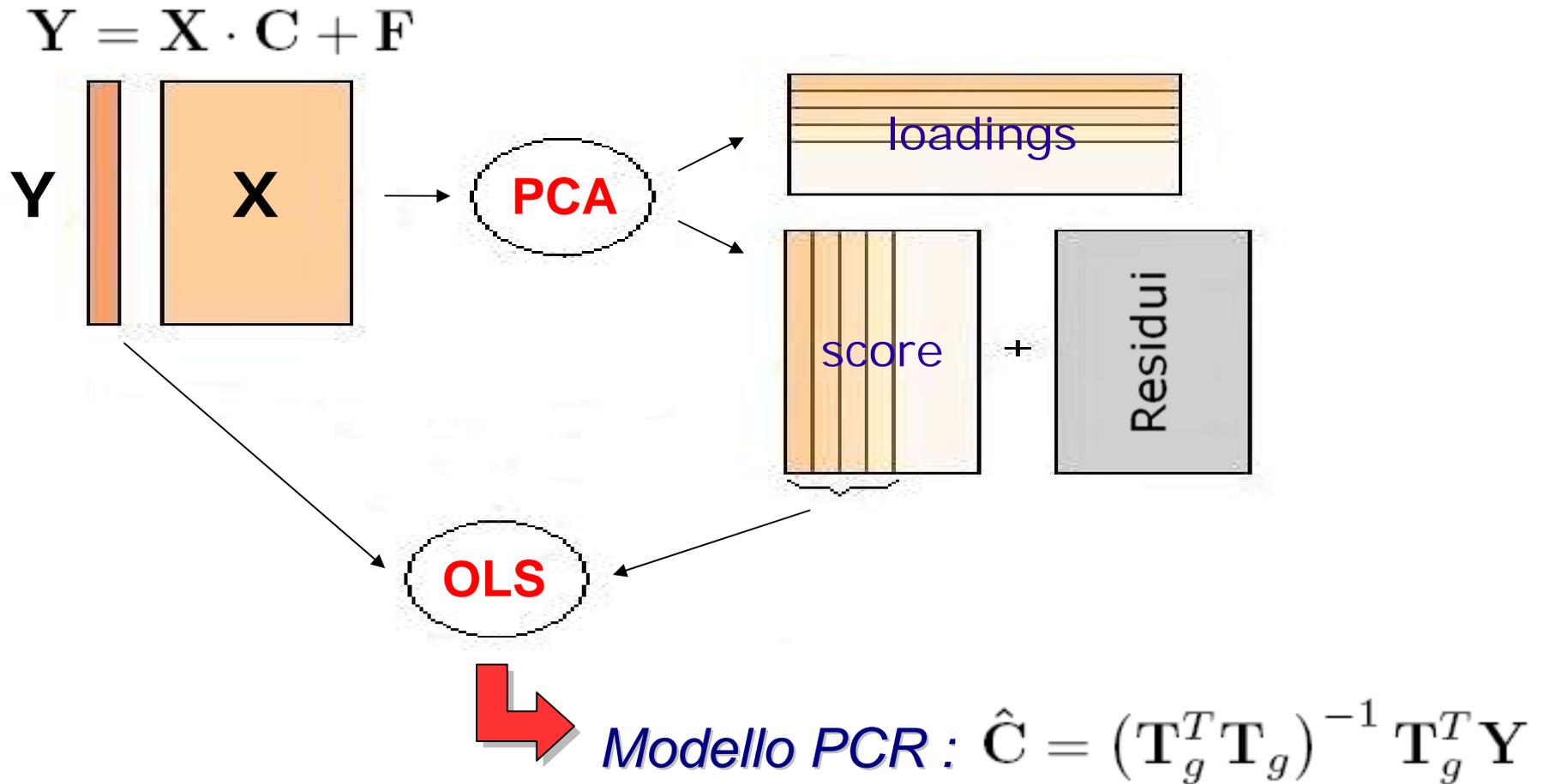
giustificazione

$$s^2(\mathbf{t}_j) = \frac{1}{n-1} \|\mathbf{t}_j\|^2 = \frac{1}{n-1} \mathbf{t}_j^T \cdot \mathbf{t}_j = \frac{1}{n-1} \lambda_j$$

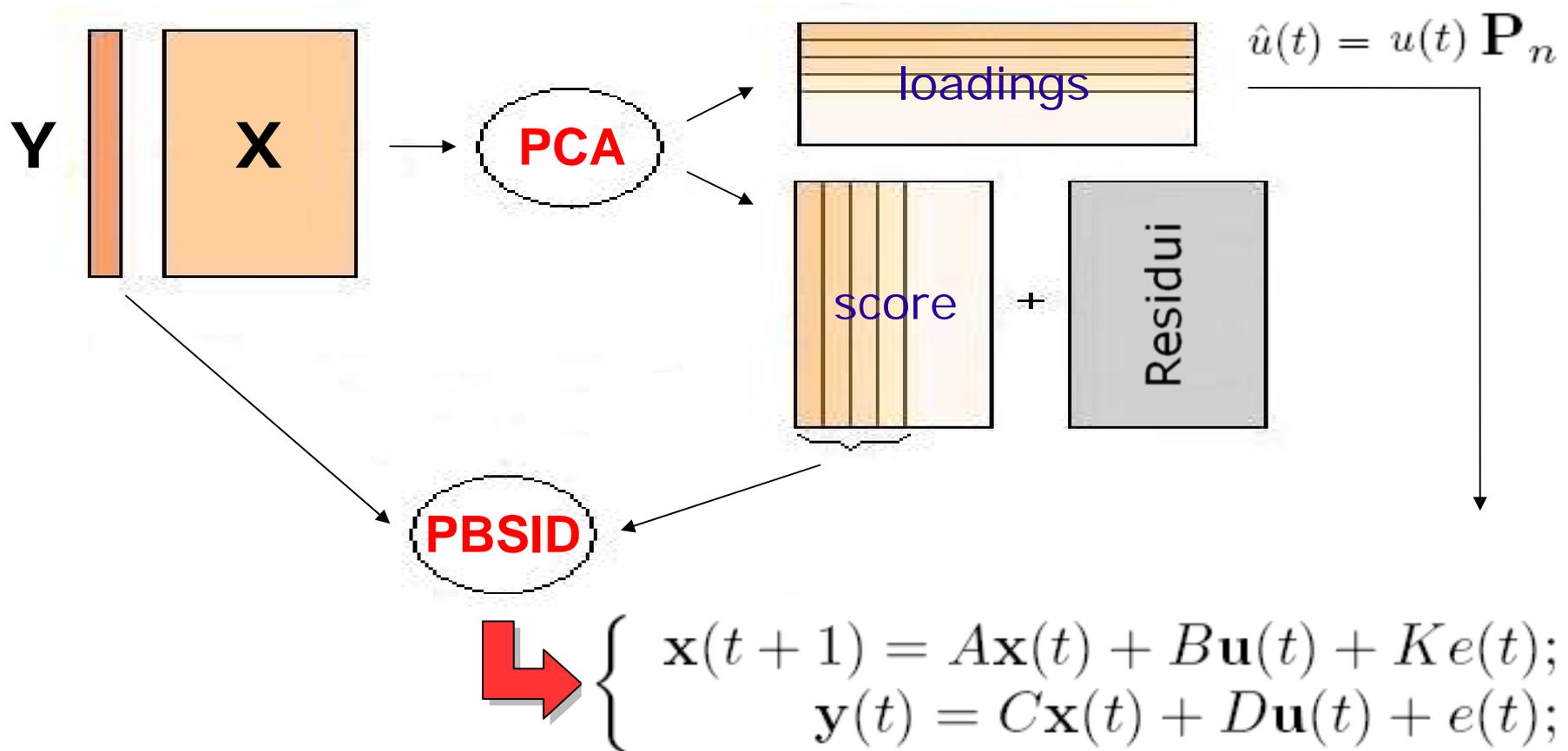


*Il peso così definito rappresenta la rilevanza
del j-esimo score per il sistema*

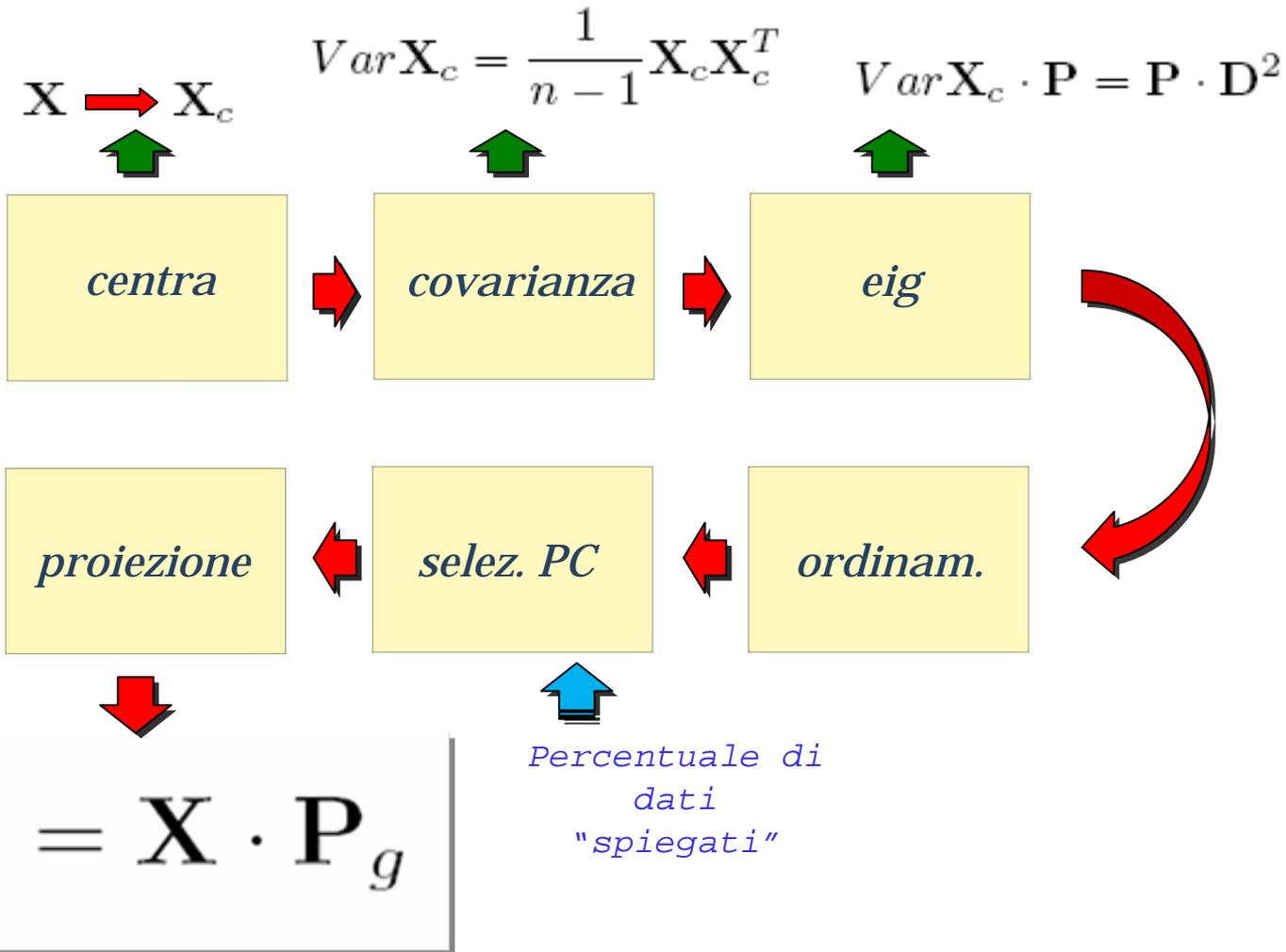
PCR



Rivisitazione di PCR

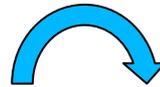


Implementazione di PCA

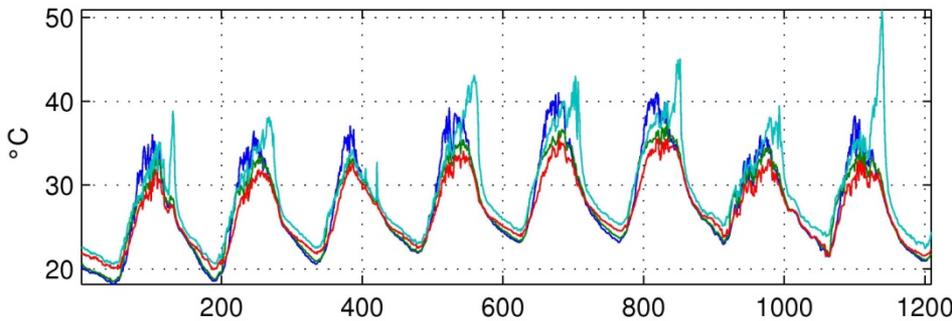


PC sugli ingressi

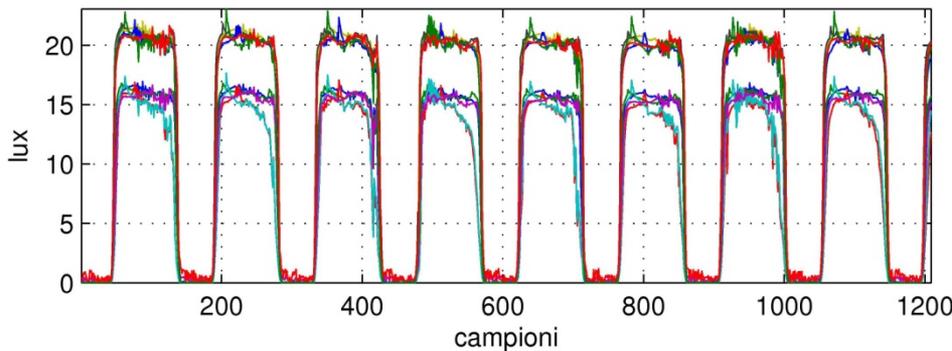
PCA



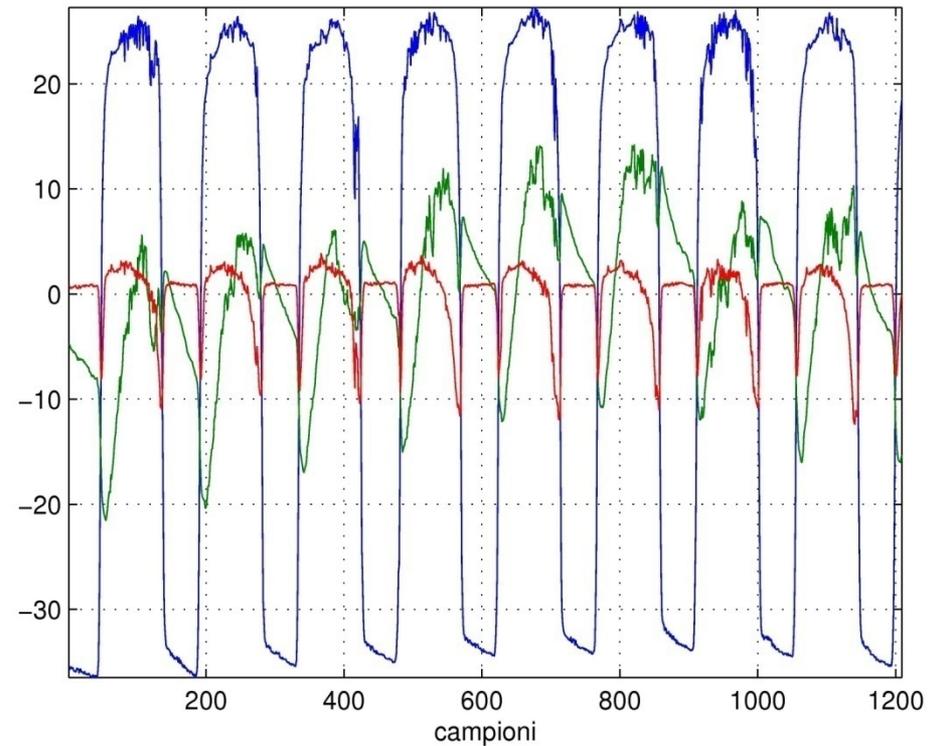
Temperature esterne



Luminosità attive e solari totali

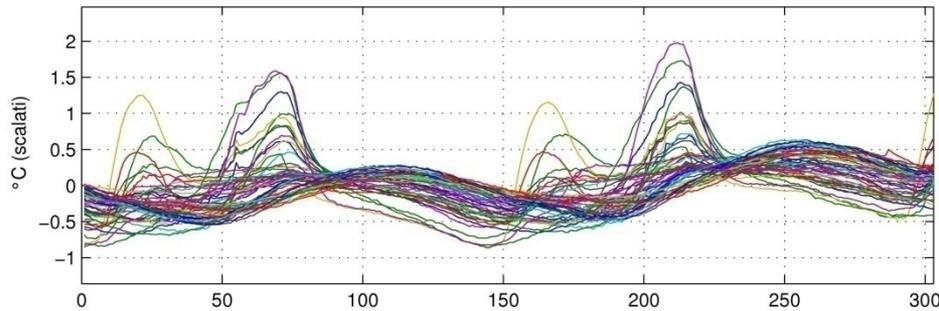


PC degli ingressi con 99.269% di informazione spiegata

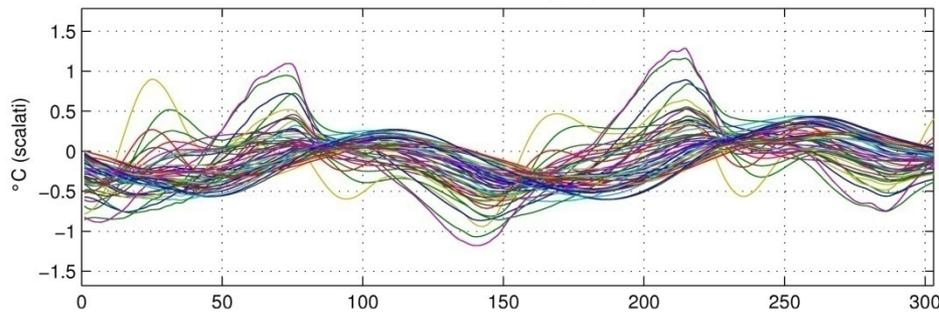


Risultati di PCR

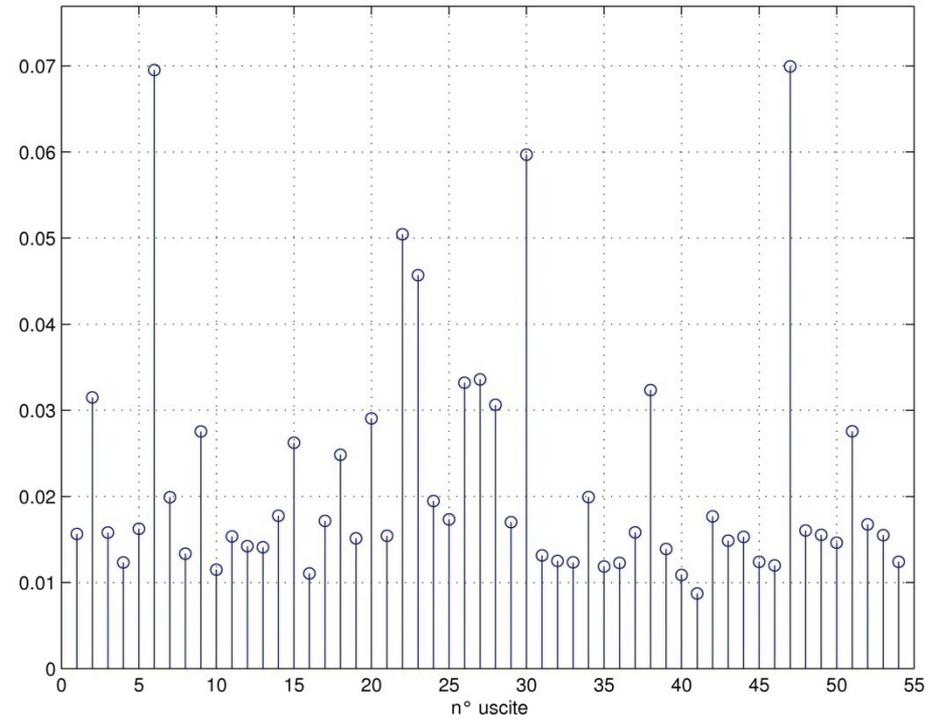
Uscite di validazione (da simulatore)



Uscite del modello identificato con gli ingressi di validazione

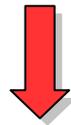


MSE catena aperta



Idea

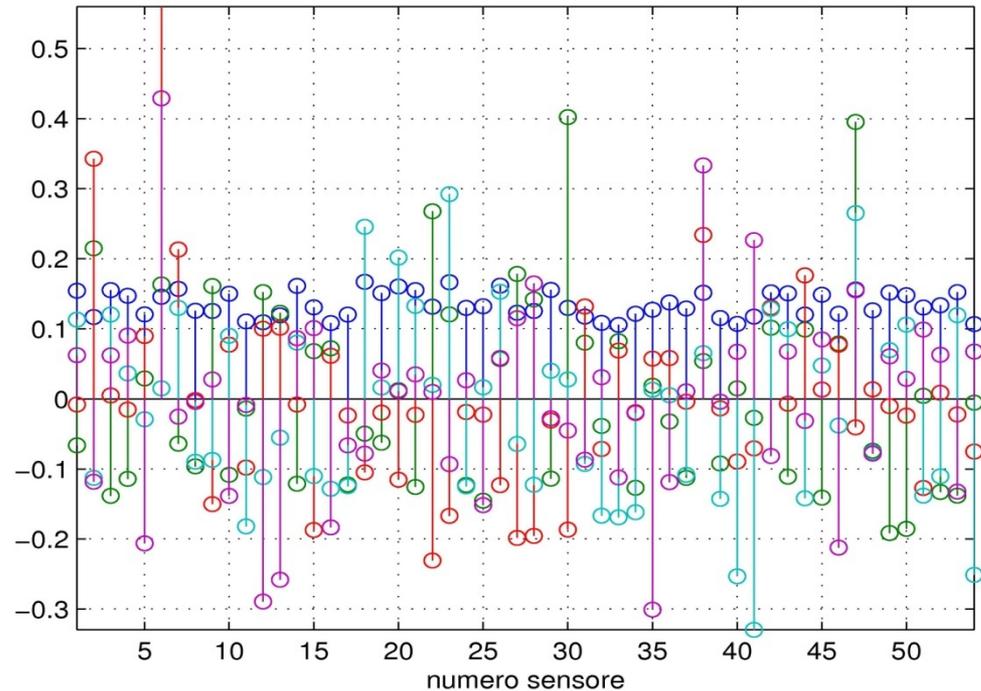
$$\mathbf{P} = [1, \dots, p_a, \dots, p_A]$$



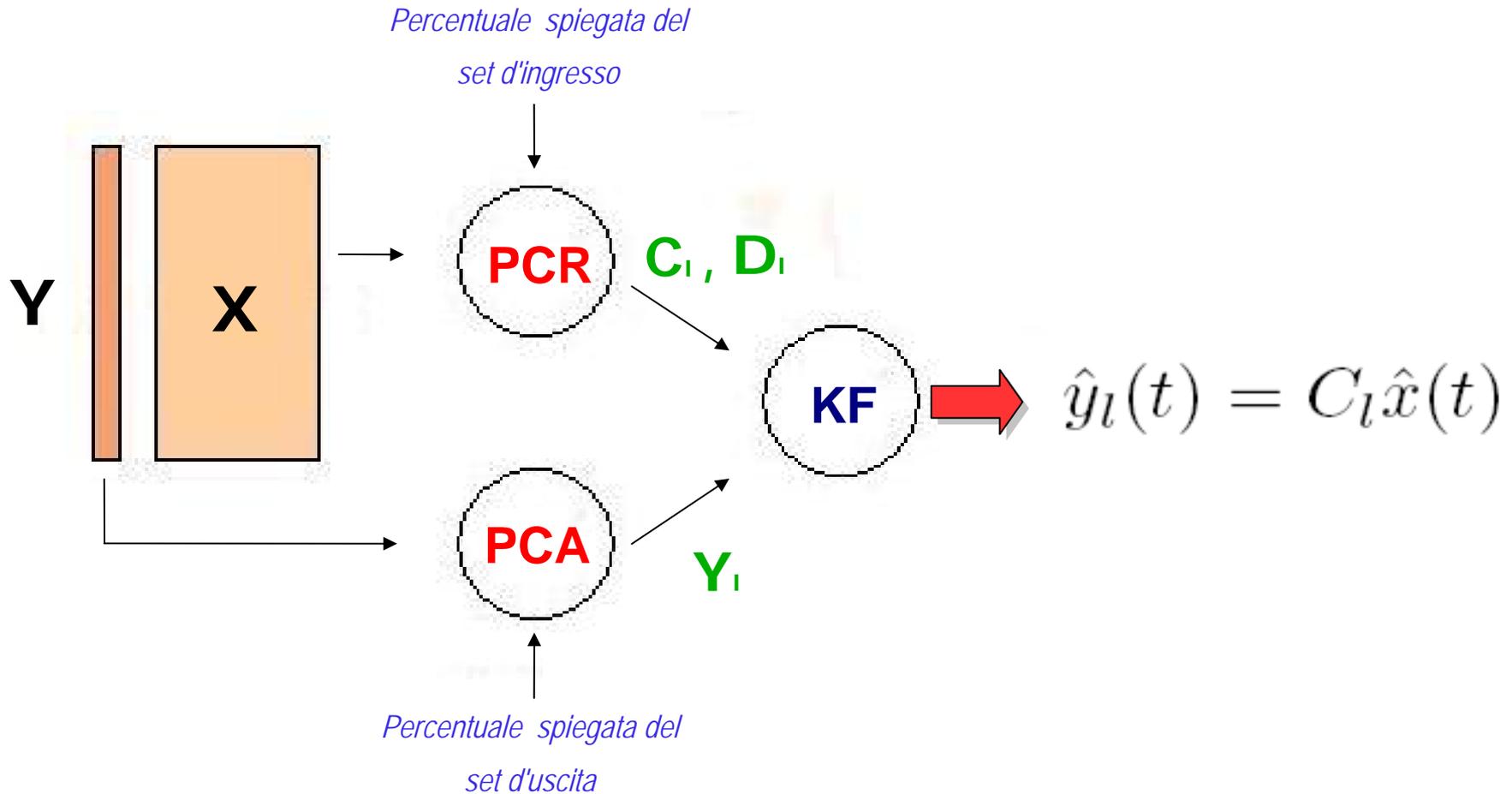
$$t_a = X p_a$$

il coefficiente i-esimo del j-esimo loading rappresenta il peso che l' i-esimo sensore ha per la j-esima PC

Loadings delle prime 5 componenti principali



Procedimento

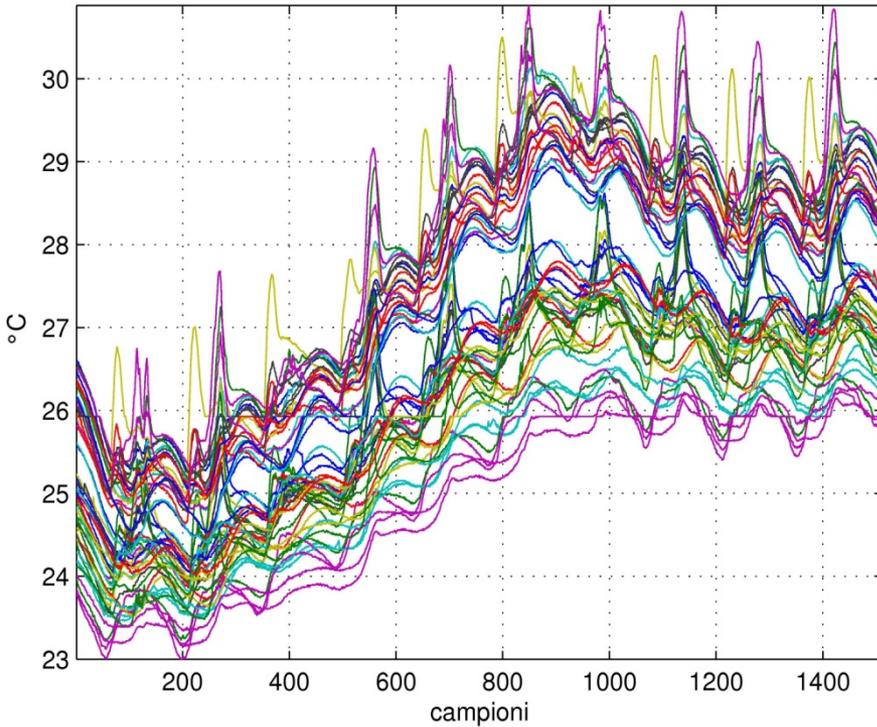


Componenti Principali delle uscite

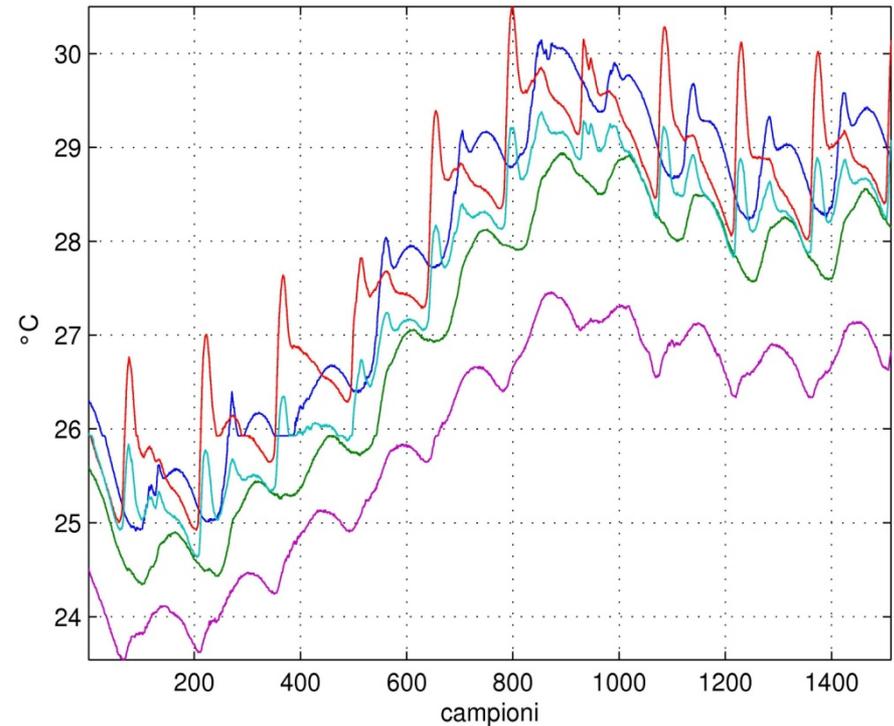
PCA



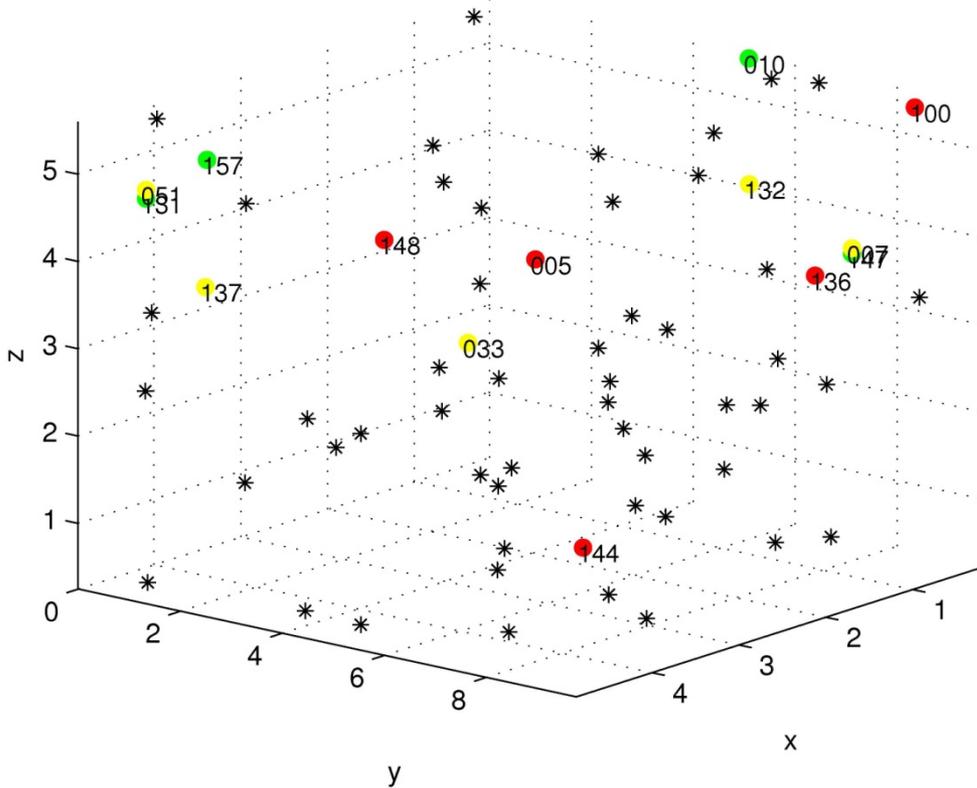
Temperature interne (uscite)



Uscite selezionate date le 5 componenti principali



Selezione e filtraggio



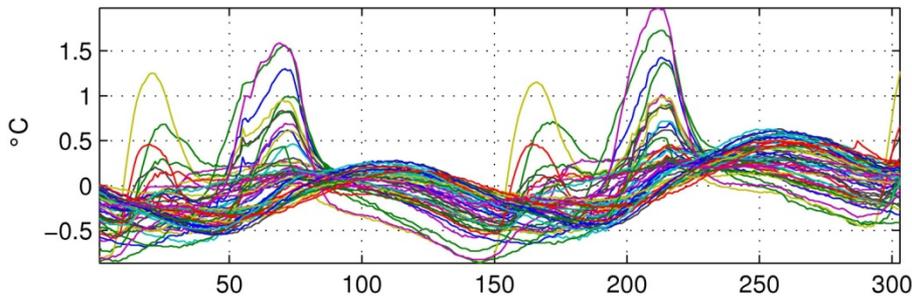
+ *modello* PCR



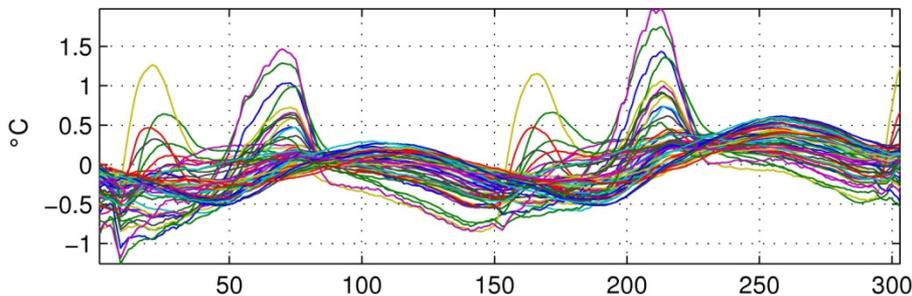
Filtro di Kalman

Risultati del sistema con 5 sensori e filtro di Kalman

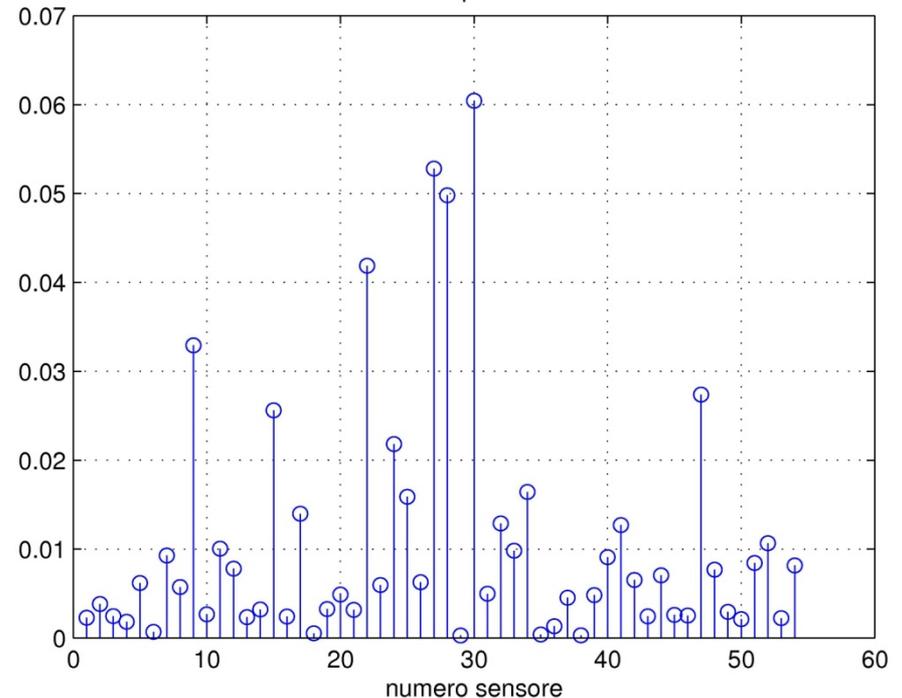
Uscite di validazione



Uscite con KF



MSE dei sensori rispetto alle uscite di KF



Conclusioni

- Le tecniche PCA sono utili per evitare il problema della collinearità
- Si analizza un solo set di dati alla volta, e non si prendono in considerazione i legami ingressi-uscite durante la selezione delle PC

 **PLS**

Sviluppi futuri :

SMI-PCA (J. Wang, S.J. Qin; 2002) che compie PCA considerando i set ingressi-uscite contemporaneamente

PLS: Proiezione su Strutture latenti

Si costruisce un'equazione di regressione tra gli scores di **X** e di **Y**



Le due matrici **X** e **Y** vengono decomposte con due relazioni esterne:

$\mathbf{X} = \mathbf{TP}' + \mathbf{E}$  Si utilizzano gli scores di **X** come nuove variabili **predittive**

$\mathbf{Y} = \mathbf{UQ}' + \mathbf{F}$  Si utilizzano gli scores di **Y** come nuove variabili **dipendenti**

Equazione di regressione interna tra gli scores

$$\mathbf{U} = \mathbf{Tb}_{\text{PLS}} + \mathbf{F}^*$$

Il modello ottenuto non è il migliore possibile... gli scores di **X** e **Y** possono non essere correlati

L'equazione di regressione rappresenta solo una minima parte di **Y**



? IDEA ?

PLS: IDEA... *WEIGHTS*

Si cerca di decomporre simultaneamente le matrici \mathbf{X} e \mathbf{Y} ruotando gli scores \mathbf{t}_a in modo da renderli più correlati agli scores \mathbf{u}_a



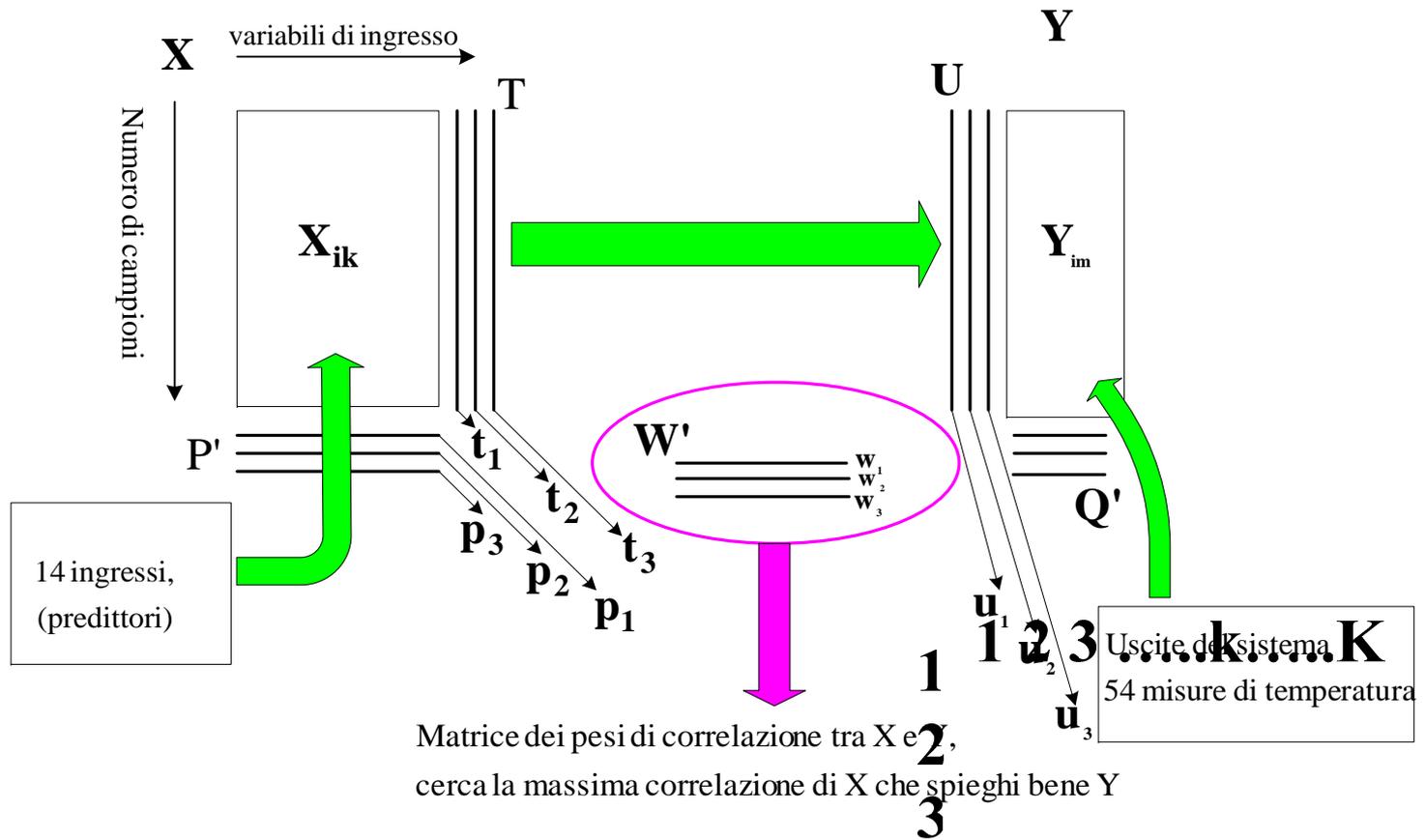
Introduciamo la matrice dei *weights* \mathbf{W}

Per calcolare i *loadings* e gli scores si utilizza l'algoritmo NIPALS (Non-linear Iterative Partial Least Squares)

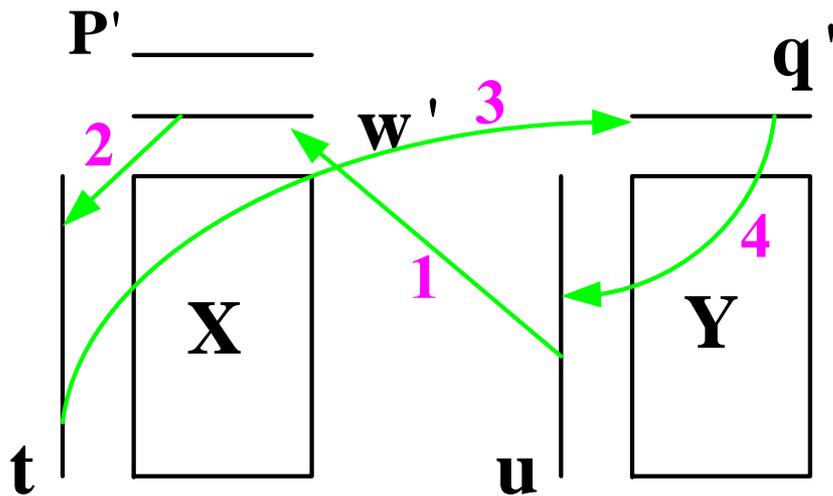
Si creano due relazioni: Una esterna costruita con la matrice dei *weights*
Una interna che calcola i residui per la prossima iterazione esterna

PLS: schema generale

PSL : Proiezione su strutture latenti (Partial Least Squares)



Algoritmo NIPALS per PLS



Si calcola in questo modo la relazione esterna. Si confronta il valore di t_a tra un'iterazione e l'altra finchè la differenza relativa è circa nulla. (scarto $10^{-8} \div 10^{-6}$)

Con la relazione interna si calcolano i coefficienti di regressione b_{PLS} e il vettore dei *loadings* P

La differenza tra i *weights* e *loadings* della matrice X è che:

→ I primi forniscono i pesi di ogni variabile predittiva a formare la variabile latente,

→ I secondi trovano la direzione della variabile latente e spiegano la varianza delle singole matrici

Questa è la differenza rispetto a PCA

PLS dinamico, modello ARX

La funzione *arx* di *matlab* non costruisce un modello ARX di 14 ingressi e 54 uscite

- Obiettivo: → costruire un modello ARX
→ utilizzare un numero opportuno di variabili latenti

$$\text{Modello ARX: } \mathbf{y}(\mathbf{k}) = \sum_{i=1}^{n_y} \mathbf{A}_i \mathbf{y}(\mathbf{k} - \mathbf{i}) + \sum_{j=1}^{n_u} \mathbf{B}_j \mathbf{u}(\mathbf{k} - \mathbf{j}) + \mathbf{v}(\mathbf{k})$$

Utilizzeremo le potenzialità dell'algoritmo PLS per stimare le matrici **A** e **B** del modello.

Costruzione dell'algoritmo di PLS dinamico

Per utilizzare l'algoritmo PLS in modo DINAMICO costruiamo la matrice $\mathbf{x}(\cdot)$

$$\mathbf{x}^T(\mathbf{k}) = \left[\mathbf{y}^T(\mathbf{k}-1), \mathbf{y}^T(\mathbf{k}-2) \dots \mathbf{y}^T(\mathbf{k}-n_y), \mathbf{u}^T(\mathbf{k}-1), \mathbf{u}^T(\mathbf{k}-2) \dots \mathbf{u}^T(\mathbf{k}-n_u) \right]$$

Di dimensione m , cioè somma di tutti le uscite e gli ingressi con il rispettivo ritardo n_u e n_y

$$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)]^T \in \mathcal{R}^{n \times m}$$

n campioni

$$\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)]^T \in \mathcal{R}^{n \times p}$$

p uscite



Matrici di ingresso e di uscita dell'algoritmo PLS

PLS: stima della matrice C

L'algoritmo PLS restituisce le matrici:

$$\mathbf{T}_n = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_m] \quad \rightarrow \quad \text{Scores di } \mathbf{X}, \text{ ortogonali}$$

$$\mathbf{P}_n = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m] \quad \rightarrow \quad \text{Loadings di } \mathbf{X}$$

$$\mathbf{B}_n = \text{diag}(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m) \quad \rightarrow \quad \text{Coefficienti di regressione del Modello } \textit{interno}$$

$$\mathbf{Q}_n = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m] \quad \rightarrow \quad \text{Loadings di } \mathbf{Y}$$



$$\mathbf{C}_n^{\text{PLS}} = (\mathbf{P}_n \mathbf{B}_n^T)^+ \mathbf{P}_n \mathbf{B}_n \mathbf{Q}_n^T$$

PLS risultati

La matrice $\mathbf{C}_n^{\text{PLS}}$ ha la forma :

$$\mathbf{C}_n^{\text{PLS}} = \left[\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{n_y}, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_{n_u} \right]^T \in \mathcal{R}^{m \times p}$$

Due problemi:



Scelta del numero di variabili latenti da utilizzare



Scelta dell'ordine del modello ARX, nel nostro caso

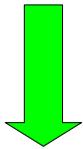
$$\mathbf{n}_y = \mathbf{n}_u = \mathbf{n}_{uy}$$

Scelta del numero di variabili latenti

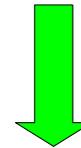
In letteratura esistono diversi criteri che danno risultati diversi in relazione a come vengono organizzati i dati.

Abbiamo utilizzato due indici: → RMSECV (Root Mean Squared Error of Cross-validation)
→ RMSEP (Root Mean Squared Error of Prediction)

Consideriamo due casi

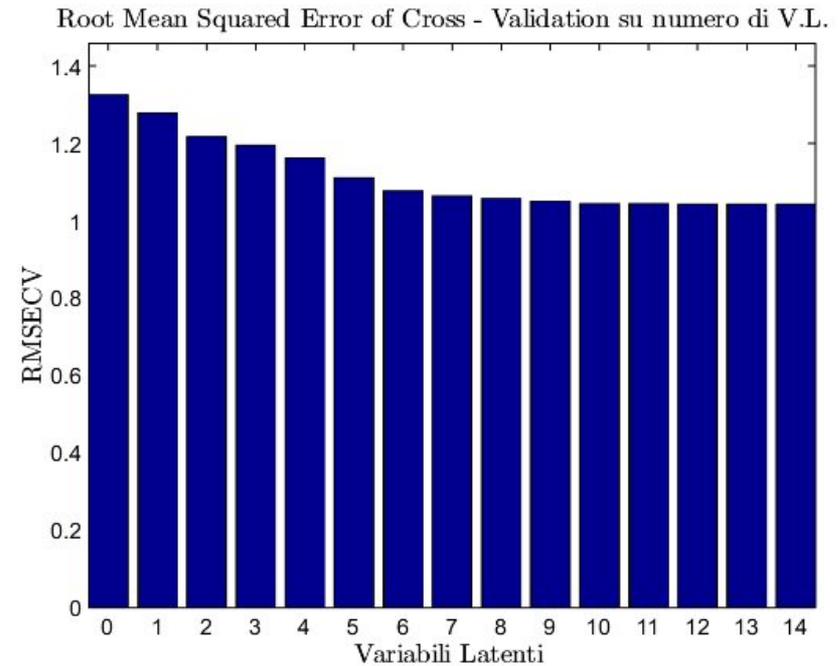
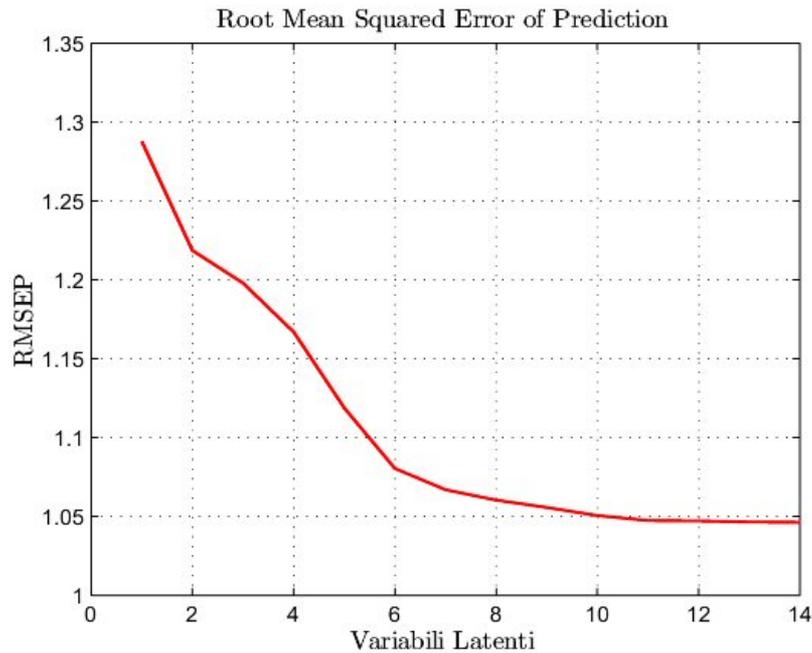


Caso Statico



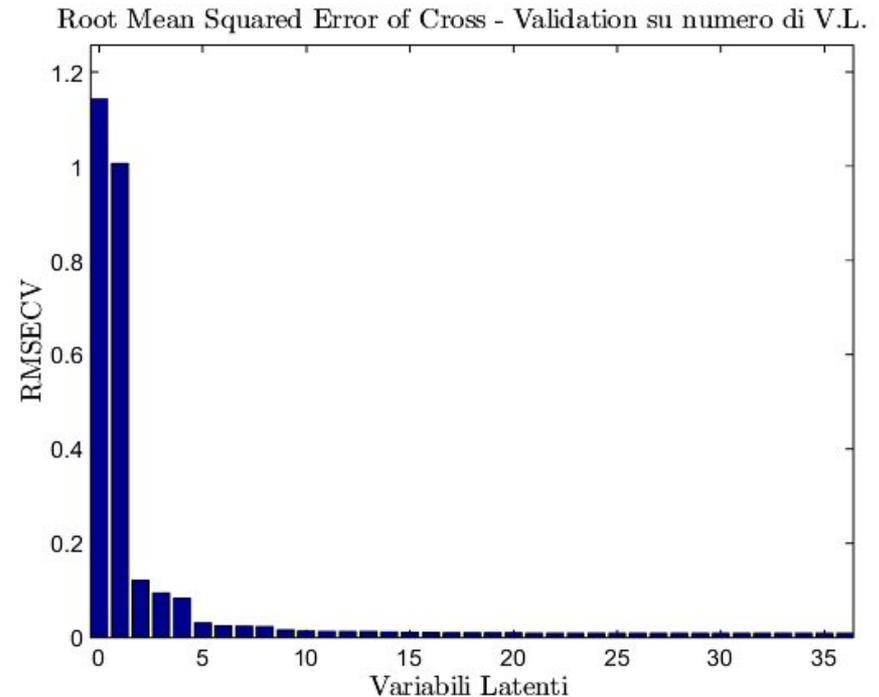
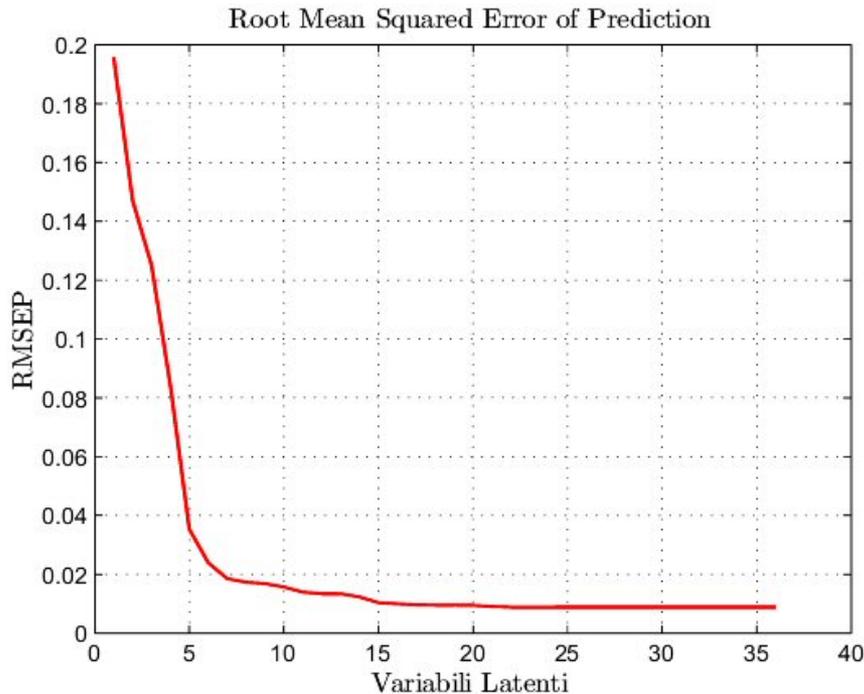
Caso Dinamico

Caso statico



Con 6 variabili latenti spieghiamo bene il sistema ma lo scarto tra la prima Variabile e la sesta è di circa **0.3** sia per **RMSECV** che per **RMSEP**

Caso dinamico



Nel caso di utilizzo del PLS dinamico le prime 6-10 variabili latenti ma lo scarto tra la prima variabile e la decima è di circa **0.2** per RMSEP e di circa **1** per RMSECV

Identificheremo quindi il modello utilizzando **5** e **10** variabili latenti

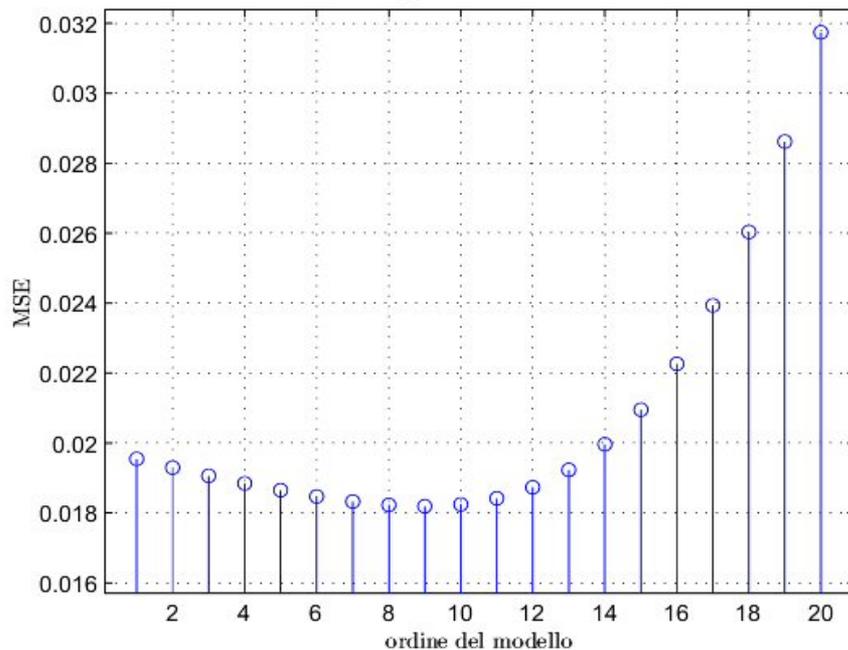
Scelta dell'ordine del modello

Due metodologie per la scelta dell'ordine $\mathbf{n}_y = \mathbf{n}_u = \mathbf{n}_{uy} = 9$

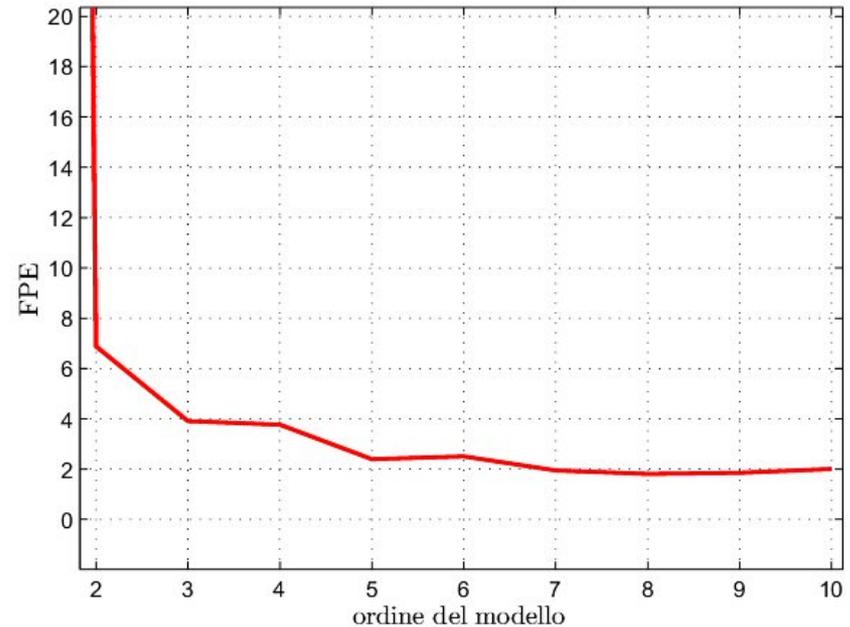
→ Attraverso il MSE

→ Attraverso l'FPE  La funzione *matlab* che calcola l'*Fpe* richiede di disporre di modelli di ordine diverso con diversi *Fpe*

MSE del sensore3 per 20 ordini di modello ARX



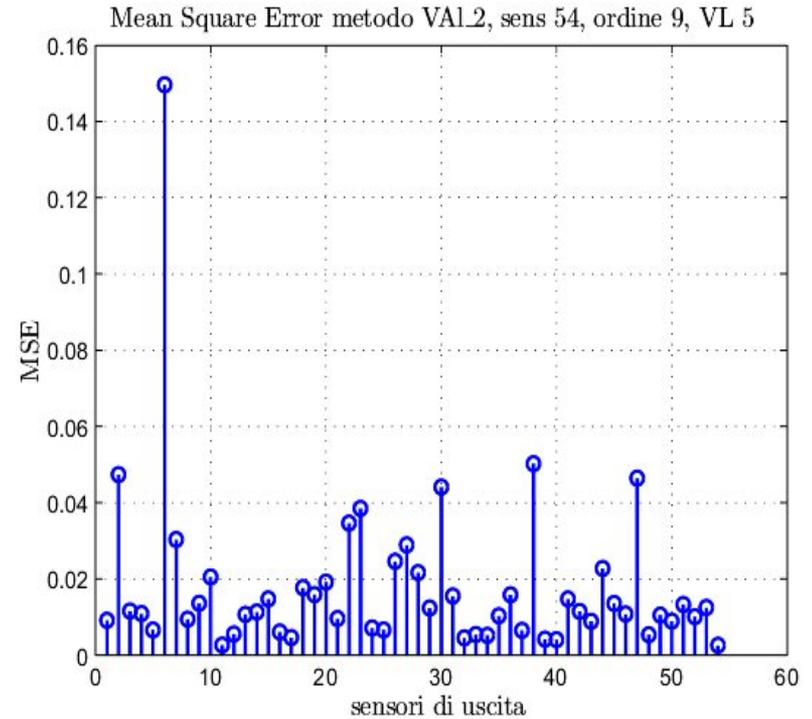
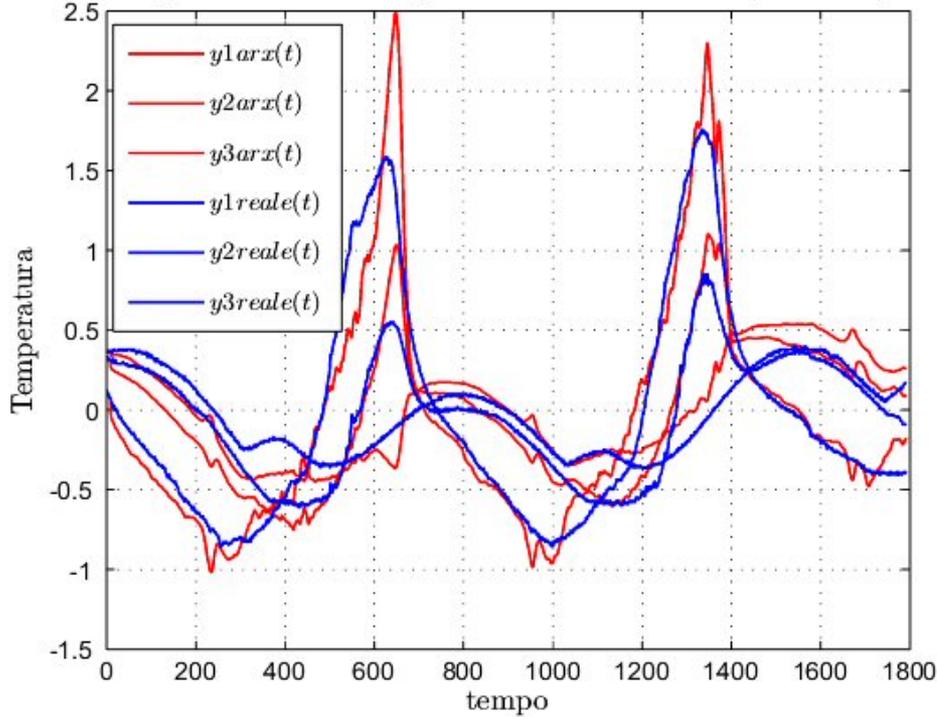
$\times 10^{-42}$ FPE di 10 modelli ARX con 14 ingressi e 10 uscite



PLS: Risultati sperimentali

5 variabili latenti, ordine 9

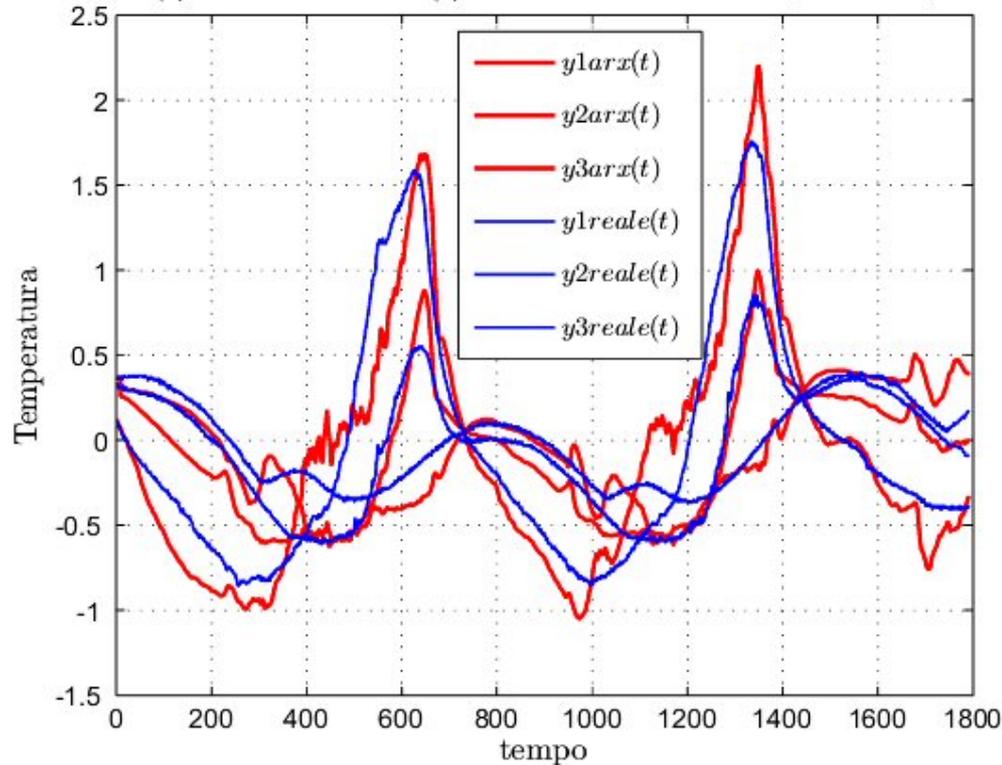
$Y_ARX(t)$ e di $Y_REALE(t)$ metodo VAL.2 uscite 3, ordine 9, V.L. 5



PLS: Risultati sperimentali

10 variabili latenti, ordine 9

$Y_ARX(t)$ e di $Y_REALE(t)$ metodo VAL.2 uscite 3, ordine 9, V.L. 10

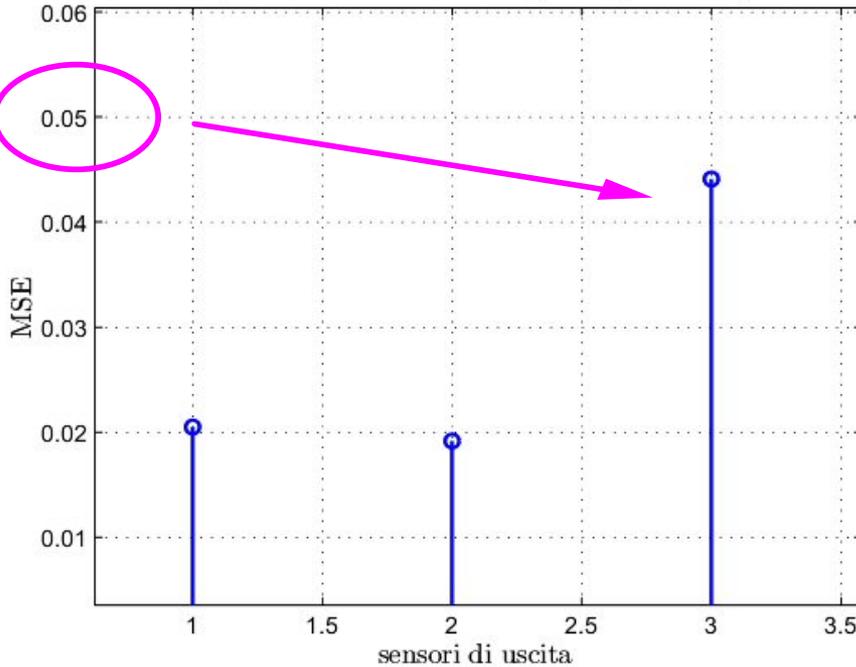


MSE 5 e 10 V.L.

MSE dei tre sensori con 5 V.L.



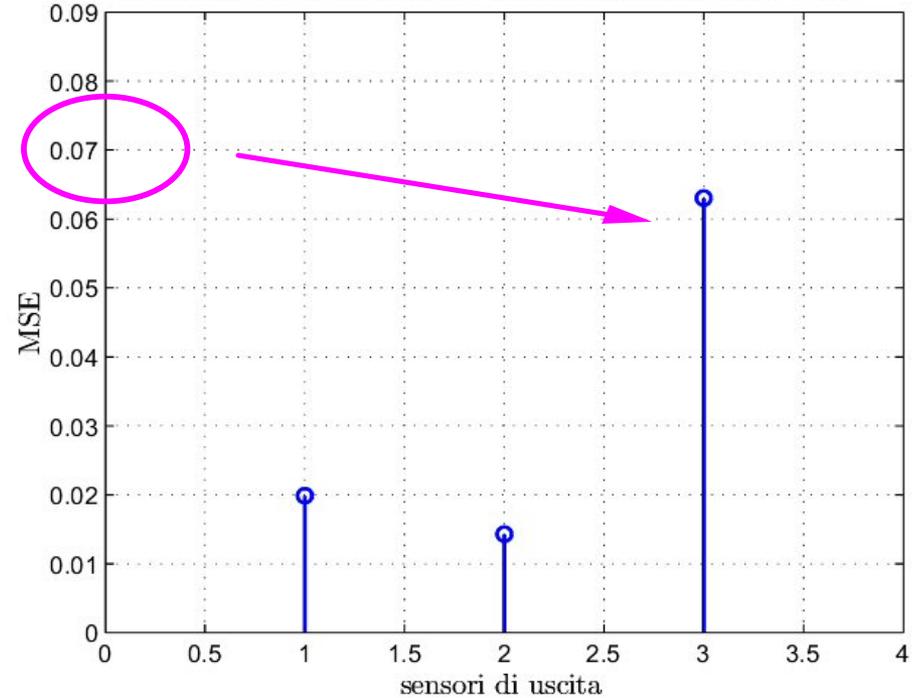
Mean Square Error metodo VAL2, sens 3, ordine 9, VL 5



MSE dei tre sensori con 10 V.L.

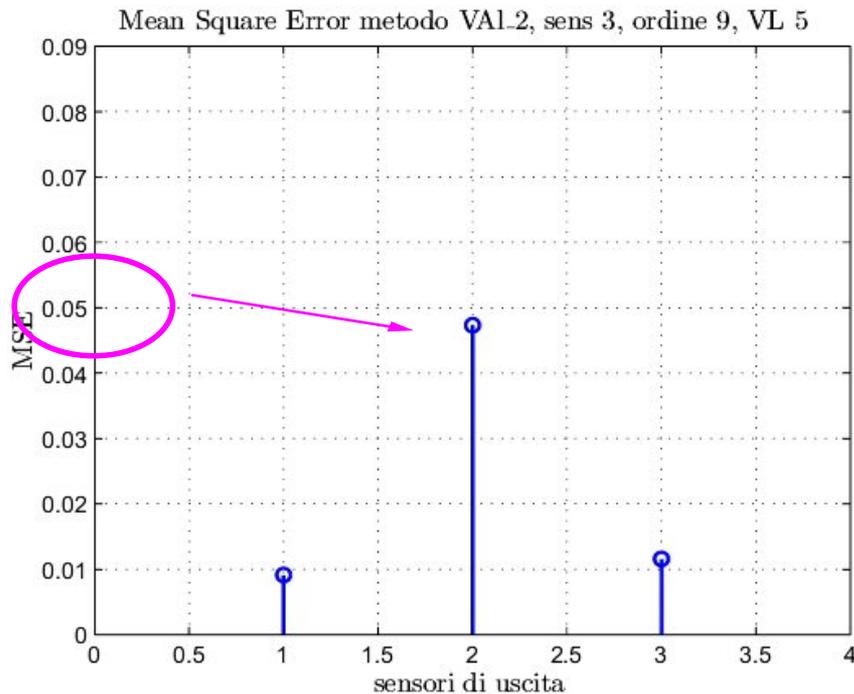


Mean Square Error metodo VAL2, sens 3, ordine 9, VL 10

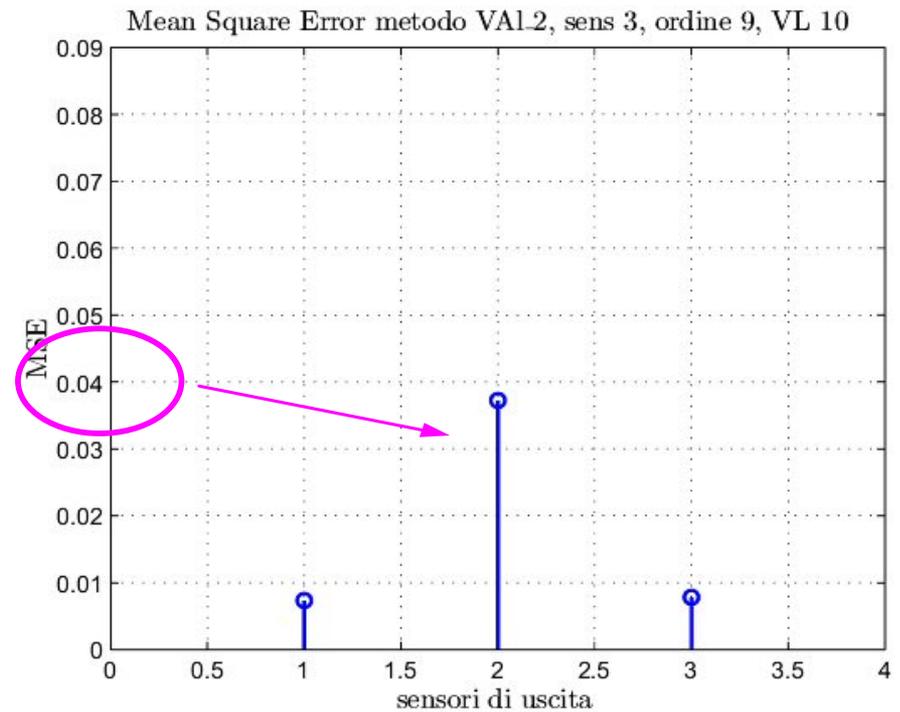


MSE 5 e 10 V.L.

MSE dei tre sensori con 5 V.L.



MSE dei tre sensori con 10 V.L.

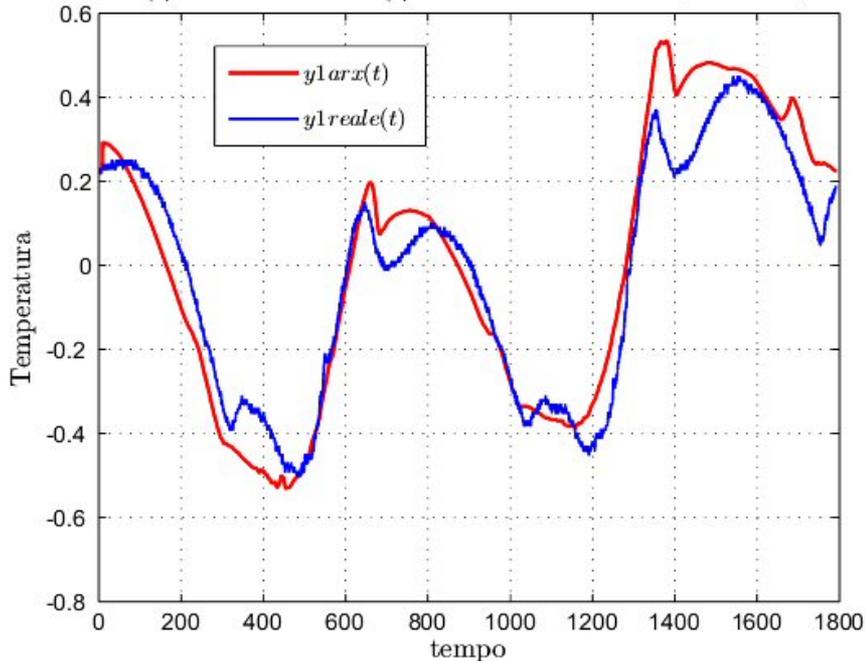


1 sensore a confronto 5 V.L. 10 V.L.

1 sensore 5 V.L.



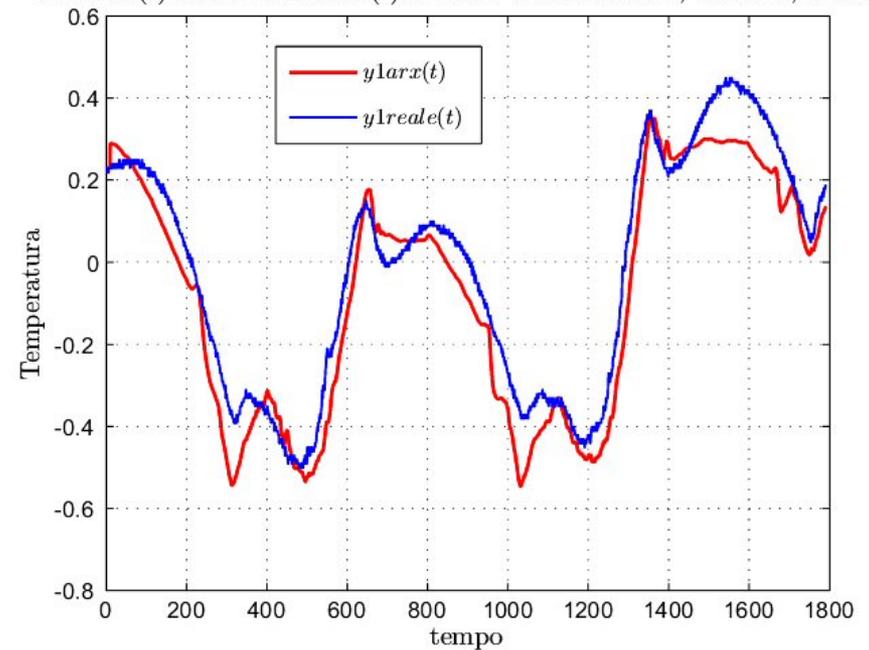
$Y_ARX(t)$ e di $Y_REALE(t)$ metodo VAL.2 uscite 1, ordine 9, V.L. 5



1 sensore 10 V.L.



$Y_ARX(t)$ e di $Y_REALE(t)$ metodo VAL.2 uscite 1, ordine 9, V.L. 10



Conclusioni e sviluppi futuri

- Soddisfatti del lavoro, si è esaminato lo stato dell'arte
- Non si riduce il numero dei sensori  Costruire una relazione dinamica
Che permetta di selezionare sensori
- PLS + SIM  A Framework for PLS-SIM integration, R.Muradore F.Bezzo
- Introdurre le applicazioni di controllo in base al quale valutare la bontà del modello