

The abstract goes here.  
IEEEtran, journal, L<sup>A</sup>T<sub>E</sub>X, paper, template.

#### Raccolta dati

**Impostazione** Per eseguire le campagne di racca

**I dispositivi utilizzati** Per eseguire le misurazioni delle grandezze di interesse sono stati utilizzati i dispositivi equipaggiati

**Viene riportata qui di seguito una breve descrizione dei sensori utilizzati dai *Tmote*:**

**Umidità e temperatura:** Il sensore di temperatura e umidità utilizzato sul mote è il sensore Sensiron SHT11. Esso consta

con costanti  $b_1 = -39.6$ ,  $b_2 = 0.01$ ,  $b_3 = -2 \cdot 10^{-8}$  e  $b_4 = 7000$

Il sensore per rilevare l'umidità è basato sulla variazione di capacità di un polimero in funzione dell'umidità presente nel

con costanti  $c_1 = 0.01$ ,  $c_2 = 8 \cdot 10^{-6}$ ,  $c_3 = -4$ ,  $c_4 = 0.0405$  e  $c_5 = -2.8 \cdot 10^{-6}$ . Per maggiori informazioni si rimanda al c

**Luminosità** I sensori presenti nel mote per misurare la quantità di luce sono due fotodiodi prodotti dalla Hamamatsu. I

con  $AR_{raw}$  il valore in bit fornito dal sensore S1087. Analogamente a partire dal dato in bit  $TR_{raw}$  del sensore S1087-0

Per la fase di identificazione la misura che viene utilizzata è quella fornita dal secondo sensore, ovvero quella comprensiva

**Memoria flash** Il mote è equipaggiato con una unità di memoria flash esterna di tipo seriale di capienza 1024kb. Essend

**Codice utilizzato** Si fornisce qui di seguito una breve descrizione del codice che viene utilizzato per la gestione dei mote.

Il firmware caricato su ogni mote divide la sessione di acquisizione in tre fasi. La prima parte è dedicata al setup che vien

L'interfaccia con il PC è sviluppata in Java su piattaforma Linux ed utilizza le librerie di comunicazione seriale e le class

[H]

Metodi di identificazione ai sottospazi I metodi di identificazione a sottospazi sono particolarmente adatti per l'identifica

Il problema è dunque, date le osservazioni  $\{y_1, \dots, y_T, u_1, \dots, u_T\}$ ,  $y_t \in \mathbb{R}^p$ ,  $u_t \in \mathbb{R}^m$  costruire il modello:

che descriva i dati (con  $e(t)$  rumore bianco).

A partire dal processo stocastico  $\mathbf{y}(t)$  si scrivono i dati nella forma  $\{y_t, \dots, y_{t+N-1}\} = \mathbf{Y}_t$  per creare una corrispondenza b

di dimensione  $Y_{[t,s]} \in \mathbb{R}^{p(s-t)xN}$  e da questa si definiscono le matrici delle uscite passate e future

e si svolgono delle operazioni analoghe con i dati d'ingresso  $\mathbf{u}(t)$ . A questo punto si può costruire una base per lo spazio

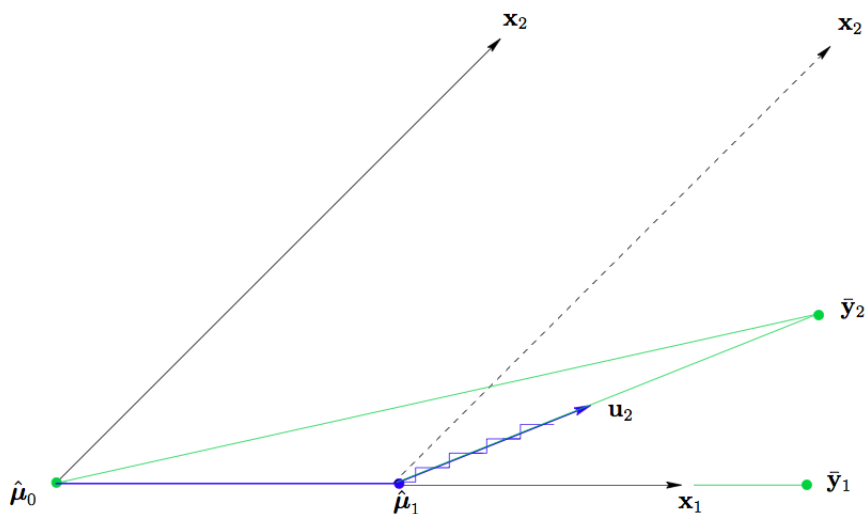
dove  $\hat{E}_t := Y_t - \hat{C}X_t$ .

Si definisce poi la matrice di osservabilità del sistema come:

e matrici triangolari inferiori a blocchi di Toeplitz con i parametri di Markov del sistema:

Introduzione agli algoritmi L'identificazione black-box è ampiamente utilizzata per ricostruire la dinamica di modelli da dati sperimentali, ma è spesso computazionalmente problematica poiché la teoria sottostante ai metodi di stima dell'ordine del modello (ad esempio AI) è complessa. Oggi sono stati introdotti nuovi approcci parametrici (come LAR e LASSO) che inducono sparsità nei risultati. Gli algoritmi che verranno testati maggiormente in questo lavoro adottano un punto di vista Bayesiano e prendono come

LARS L'algoritmo Il LARS ( Least Angle Regression ) è una versione semplificata della procedura Stagewise la quale us  
 Come nel più classico algoritmo Forward Selection si parte con tutti i coefficienti della stima  $\beta$  uguali a zero e si trova il  
 Si dà ora una veloce descrizione geometrica di questo processo. Con il LARS si cerca di dare una stima di  $\hat{\mu} = X\hat{\beta}$  in un



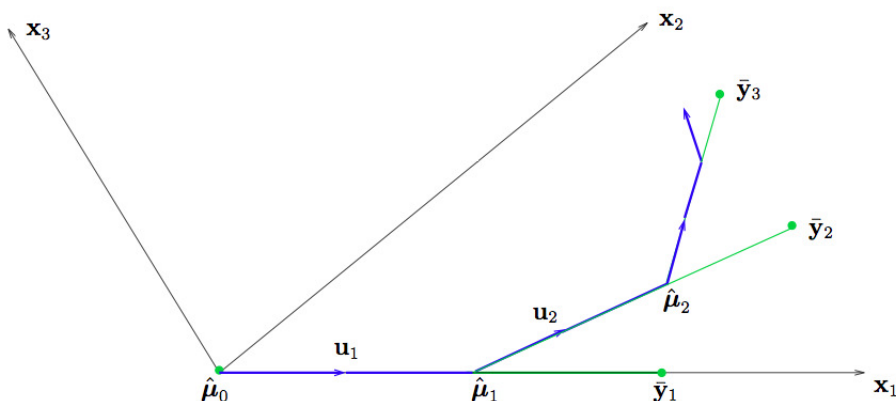
[htbp]

In figura viene illustrato il caso con due covarianze  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2)$  cioè con  $m=2$ . In questo caso la correlazione corrente

dimostrando una dipendenza solo da  $\bar{y}_2$  nello spazio lineare  $\mathcal{L}(X)$  generato da  $\mathbf{x}_1$  e  $\mathbf{x}_2$ . Come precedentemente spiegato

A questo punto l'algoritmo Stagewise sceglierebbe  $\hat{\gamma}_1$  uguale ad un qualche valore  $\epsilon$  (molto piccolo) per poi ripetere il pr  
 Sia  $\mathbf{u}_2$  il versore della bisettrice, il nuovo passo del LARS sarà:

con  $\hat{\gamma}_2$  scelta per rendere  $\hat{\mu}_2 = \bar{y}_2$  nel caso di  $m=2$ . Si noti che con più di 2 covarianze  $\hat{\gamma}_2$  sarebbe più piccola in conform



[htbp]

Da questo processo si evince che il calcolo teorico di  $\hat{\gamma}_1, \hat{\gamma}_2, \dots, \hat{\gamma}_m$  sia molto più veloce per il LARS rispetto al calcolo  
 Si presume ora di assumere i vettori covarianze  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$  linearmente indipendenti. Per  $\mathcal{A}$ , un sottoinsieme degli in

dove la variabile segno  $s_j$  può assumere i valori  $\pm 1$ . Si prende

dove  $\mathbf{1}_{\mathcal{A}}$  è un vettore con elementi unitari con lunghezza pari  $|\mathcal{A}|$ , la dimensione di  $\mathcal{A}$ .  
 Sia

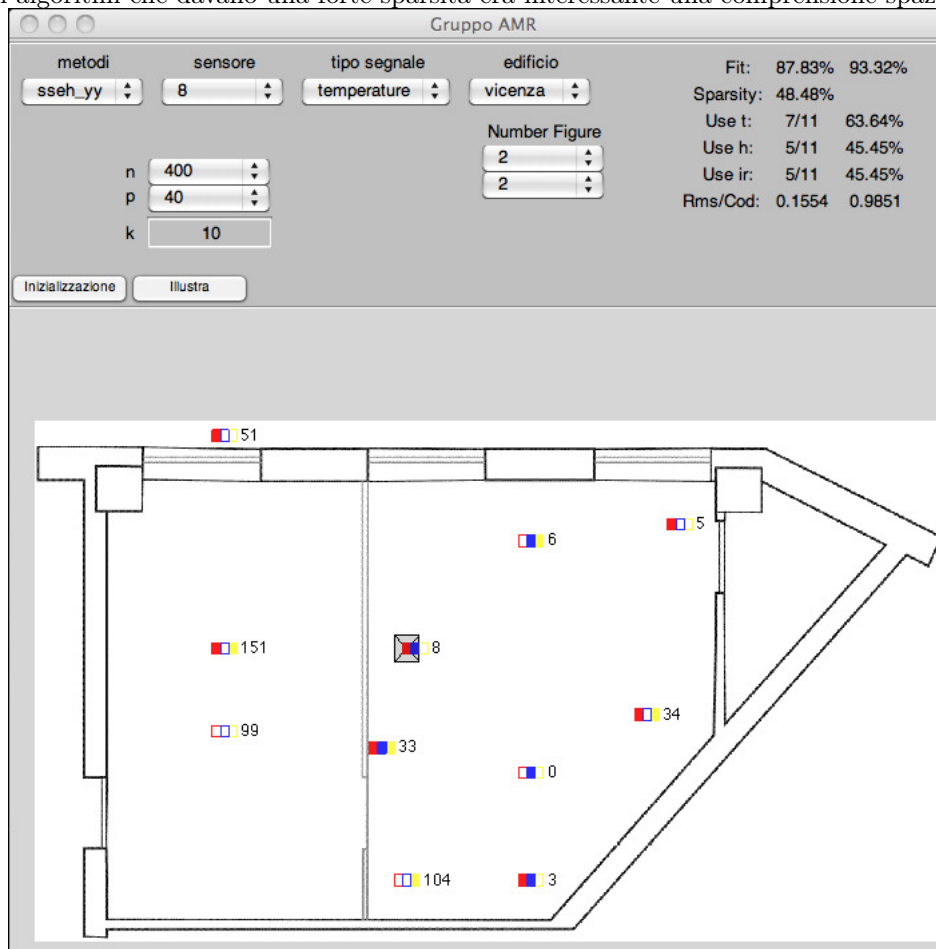
il vettore equiangolare ovvero il vettore unitario che rende gli angoli (minori di  $90^\circ$ ) con le colonne della matrice  $\mathcal{A}$  tutti

Si descrivono ora i passi seguiti dall'algoritmo LARS. Come in Stagewise si parte con  $\hat{\mu}_0 = \mathbf{0}$  e si costruisce  $\hat{\mu}$  iterativam

il vettore delle correlazioni correnti come definito dalla generica  $\hat{\mathbf{c}} = \mathbf{c}(\hat{\mu}) = X'(\mathbf{v} - \hat{\mu})$ . L'insieme attivo  $\mathcal{A}$  è composto

L'interfaccia grafica

Come visto nei paragrafi precedenti ci si è trovati di fronte ad un enorme quantità di identificazioni legate ad un serie di Per gli algoritmi che davano una forte sparsità era interessante una comprensione spaziale di quali ingressi vengono usati



[htbp]

L'interfaccia grafica realizzata non è un simulatore, cioè non richiama direttamente le funzioni di identificazione; cosa ch

La GUI è composta da due zone principali: la prima la zona di selezione dei parametri , mentre la seconda è una zona g

Come si vede i campi di scelta sono: "Metodi", "Sensore", "Tipo Segnale" ed "Edificio". Con il primo si seleziona il tipo d

'sseh\_yy'

'sseh\_ny'

'n4sid'

'ssglar'

'glar'

'pem'

Per una descrizione leggere i paragrafo competente in cui vengono spiegati.

Con il secondo menù a tendina si seleziona quale sensore va considerato come uscita , l'elenco conterrà gli id dei vari ser

A destra dei quattro menù di selezione vi è una zona in cui vengono illustrati ogni volta i risultati dei fit dell'identificazi

Invece sotto i quattro menù a tendina si trovano due colonne , la prima riguardante i parametri dell'identificazi

Sempre nella parte superiore dell'interfaccia, in basso a sinistra si trovano due pulsanti: "inizializza" ed "illustra". Il prim

La parte in cui viene visualizzata la mappa dell'edificio ogni sensore è raffigurato da un quadrato di colore rosso per la t

## La struttura Dati

Tutti i dati e tutte le simulazioni sono state salvate in un'unica struttura di Matlab per avere in un unico luogo tutte le informazioni che riguardano la temperatura.

t: [1x1 struct] : che ha sua volta è una struttura contenente tutte le informazioni che riguardano la temperatura.

h: [1x1 struct] : che ha sua volta è una struttura contenente tutte le informazioni che riguardano l'umidità.

name\_of\_y , una stringa contenente il nome esteso del sensore.

id , una stringa contenente il numero del sensore, come id nella rete di sensori.

Le strutture .h e .t contengono a loro volta gli stessi campi, per cui la descrizione che avverrà in seguito sarà fatta per entrambi.

.y\_org: [2494x1 double] : vettore contenente il segnale che si vuole identificare. Quindi sarà n-esima temperatura o umidità.

.y: [2494x1 double] : vettore contenente il segnale che si vuole identificare in media nulla.

.y\_norm: [2494x1 double] : vettore contenente il segnale che si vuole identificare con media nulla e varianza unitaria.

.Udati\_org: [2494x32 double] : Una matrice che contiene tutti i segnali a meno del segnale che si vuole identificare. La matrice Udati\_norm è la stessa matrice Udati\_org con normalizzazione in media a 0 ma non in varianza unitaria.

.Udati\_norm: [2494x32 double] : La stessa matrice Udati\_org con normalizzazione in media a 0 e con varianza unitaria.

.Colonne\_U: [1x32 double] : Un vettore che contiene gli indici dei segnali contenuti in Udati\_org per ricostruire poi la matrice Udati\_norm.

.sseh\_yy: [1x2 struct] : Struttura in cui è rinchiusa tutta l'identificazione eseguita con l'algoritmo sseh usando il parametro sseh\_norm.

.sseh\_ny: [1x2 struct] : Struttura in cui è rinchiusa tutta l'identificazione eseguita con l'algoritmo sseh usando il parametro sseh\_norm.

.n4sid: [1x1 struct] : Struttura in cui è rinchiusa tutta l'identificazione eseguita con l'algoritmo n4sid.

.ssglar: [1x1 struct] : Struttura in cui è rinchiusa tutta l'identificazione eseguita con l'algoritmo ssglar.

.pem: [1x1 struct] : Struttura in cui è rinchiusa tutta l'identificazione eseguita con l'algoritmo pem.

.glar: [1x1 struct] : Struttura in cui è rinchiusa tutta l'identificazione eseguita con l'algoritmo glar.

.sseh\_yy\_norm: [1x2 struct] : stessa descrizione della sseh\_yy solo con i dati normalizzati con varianza unitaria.

.sseh\_ny\_norm: [1x2 struct] : stessa descrizione della sseh\_ny solo con i dati normalizzati con varianza unitaria.

.n4sid\_norm: [1x1 struct] : stessa descrizione della n4sid solo con i dati normalizzati con varianza unitaria.

.ssglar\_norm: [1x1 struct] : stessa descrizione della ssglar solo con i dati normalizzati con varianza unitaria.

.pem\_norm: [1x1 struct] : stessa descrizione della pem solo con i dati normalizzati con varianza unitaria.

.glar\_norm: [1x1 struct] : stessa descrizione della glar solo con i dati normalizzati con varianza unitaria.

La struttura Dati(index).h.sseh\_ny e Dati(index).h.sseh\_yy hanno la stessa forma <sup>3</sup>. Essendoci due variabili che potevano essere trattate separatamente.

Mnpf : Struttura idpoly di Matlab contenente il modello identificato

n\_ : Valore del n usato nella simulazione.

p\_ : Valore del p usato nella simulazione.

dati\_val : Struttura iddata di Matlab contenente i dati.

La struttura Dati(index).t.n4sid contiene due strutture:

struc\_n4sid: [1x3 struct] insieme dell'identificazioni fatte

best\_mod: [1x1 struct] miglior identificazione fatta, contiene tutta la struttura.

Ovviamente struc\_n4sid e best\_mod hanno la stessa struttura:

dati\_id: [601x1x32 iddata] iddata con i campioni per l'identificazione

dati\_val: [800x1x32 iddata] iddata con i campioni per la validazione

iden\_n4sid: [4-D idss] la struttura idss identificata

n: Campioni usati nell'identificazione

fit: Fit dell'identificazione

I dati della ssglar sono tutti racchiusi nella struttura mod ( Dati(index).h.ssglar(1).mod ) la cui forma è la seguente:

Mnpg: [1x32x186 idpoly] Struttura idpoly di Matlab con la struttura identifica.

Mnpc: [1x32x186 idpoly] Struttura idpoly di Matlab usando la soglia del 5% per avere una sparsità maggiore.

serie di costanti

beta

sigma

NT

beta2

sigma2

NT2

n

pRKHS

Mgl: [1x32x240 idpoly] Originale

Mgls: [1x32x120 idpoly]

deg: 30

n: 500

Come ultimo campo si trova l'identificazione pem contenuta in Dati(index).t.pem.mod con la seguente struttura:

Mpem\_BIC: [1x32x204 idpoly] Identificazione usando il metodo BIC per la stima dell'ordine del modello.

Mpem\_AICC: [1x32x340 idpoly] Identificazione usando il metodo AICC per la stima dell'ordine del modello.

n : numero di campioni usati.

Pem: Prediction Error Methods

In questo capitolo viene spiegato brevemente il principio di funzionamento dei metodi a minimizzazione dell'errore di predizione. Dato un modello  $M(\theta)$  appartenente a una certa classe parametrica  $\mathcal{M} \equiv \{M(\theta); \theta \in \Theta\}$  e dato una sequenza di  $N$  misure

si procede nel modo seguente:

Per un qualche valore di  $\theta$  fissato si costruisce il miglior (secondo qualche criterio) predittore all'istante  $t - 1$  dell'uscita

La predizione  $\hat{y}_\theta(t|t-1)$  si può pensare come funzione dei dati passati e quindi come una funzione aleatoria che viene influenzata dai dati passati. Si formano gli errori di predizione

Analogamente a quanto detto nel caso del predittore anche gli errori di predizione possono essere visti come quantità aleatorie. Si minimizza rispetto a  $\theta$  una cifra di merito che descrive quanto bene il modello predice il dato successivo. Ad esempio

o più in generale una media degli errori quadratici di predizione pesati con una cifra di merito non negativa  $\beta(N, t) > 0$ .

che per  $N$  piccoli dà peso minore agli errori di predizione compiuti nella fase iniziale dell'algoritmo quando l'influenza dei dati passati è maggiore. In ogni caso si ricava la stima di  $\theta$  dalla minimizzazione della cifra di merito

che è la stima PEM del parametro del modello. Lo stimatore  $\hat{\theta}_N$  viene chiamato stimatore PEM del parametro  $\theta$ . Infine si prende come stima della varianza dell'innovazione  $\lambda^2 = \text{var}\{\mathbf{e}(t)\}$  l'errore quadratico residuo

Per quanto questa procedura sembra intuitivamente sensata, l'unica giustificazione per la sua adozione nei procedimenti di stima è

Finora si è visto come ricavare lo stimatore PEM nel caso si ha un parametro  $\theta$  fissato. Poiché in generale però non si ha

dove  $rss$  (Residual Sum of Squares) è definito da  $rss = \sum_{t=1}^N \varepsilon_\theta(t)^2$ , mentre  $N$  è il numero di misure e  $p$  è l'ordine del polinomio. Nel progetto in esame è stata implementata una routine nella quale viene considerato la classe dei modelli ARMAX. In

#### Validazione

Una parte molto importante dell'identificazione è la fase di validazione del modello ottenuto. Si cerca di capire la bontà dei dati per l'identificazione, dati per la validazione.

La validazione consiste nel confrontare l'uscita reale con l'uscita del modello identificato avendo come ingressi i dati reali. Di seguito vengono spiegati i diversi indici significativi alla fine di fare validazione.

**Covarianza e correlazione:** La covarianza (campionaria) e la correlazione (campionaria) si possono facilmente calcolare in Matlab. La covarianza indica se esiste una dipendenza tra le due variabili. Valori elevati indicano una forte legame tra le due variabili.

**Test di bianchezza:** I due indici appena visti spiegano l'importanza del test di bianchezza. Serve per verificare se l'errore di predizione è bianco. Per effettuare il test si bianchezza è utile il comando `resid` di Matlab che plotta un grafico per la auto-correlazione del residuo. Assumendo che l'errore di predizione  $\varepsilon(t)$  sia bianco si ha che la covarianza  $r_\varepsilon(\tau) = 0$  per ogni  $\tau \neq 0$  e quindi per la covarianza campionaria si ha per  $N \rightarrow \infty$

Per il test di bianchezza infine viene considerato la quantità normalizzata

**Test di indipendenza:** Il test di indipendenza è relativo alla cross-correlazione tra l'errore di predizione e gli ingressi reali. Il comando `resid` assume che il residuo  $\varepsilon(t)$  sia indipendente dagli ingressi passati e sia a media nulla e che si ha quindi  $r_\varepsilon(\tau) = 0$  per ogni  $\tau \neq 0$ . Per  $\tau$  si considerano sia valori positivi che negativi. Mentre se il modello non è adeguato per descrivere il sistema per valori negativi di  $\tau$ . Calcolando la correlazione campionaria come

per il test di indipendenza si considera la quantità normalizzata

**Mapa zeri-poli:** Tracciando la mappa zeri-poli di ogni funzione di trasferimento del modello identificato si può dare una buona idea del fit: Il fit da la percentuale della varianza spiegata ed è calcolato mediante la seguente formula:

dove  $y_h$  sono le uscite dal modello identificato, mentre  $y$  è l'uscita reale. Dalla formula si vede che più l'uscita del modello si avvicina all'uscita reale, più il fit è buono. Errore quadratico medio: Un indice che valuta la prestazione del modello identificato che viene spesso usato è l'MSE (mean square error)

Questo indice da quindi informazioni sulla bontà dello stimatore in termini di varianza e dispersione della stima. Spesso si usa anche l'indice di bontà del fit

**Sparsità** La sparsità da la percentuale di funzioni di trasferimento nulle nel modello identificato. Avere una funzione di trasferimento nulla indica che il modello non è in grado di spiegare la dinamica del sistema.



Subsection Heading Here Subsection text here.

Subsubsection Heading Here Subsubsection text here.

Conclusion The conclusion goes here.

Proof of the First Zonklar Equation Appendix one text goes here.

Appendix two text goes here.

\*Acknowledgment

The authors would like to thank...

H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.