

Stima e algoritmi di consensus distribuito: considerazioni su IKF e decomposizione del modello

Loris Antoniazzi

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Padova
E-mail: speorcio@libero.it

Marco Bortolomiol

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Padova
E-mail: marco.bortolomiol@tiscali.it

Sommario—Con questo lavoro s'intende esaminare due diverse strategie per la stima distribuita dello stato, nel particolare caso di sistemi di larga scala, che per loro natura si prestano ad essere decomposti in sottosistemi (simili tra loro). Gli approcci considerati sono la stima con filtri di Kalman distribuiti basati sulla tecnica del consensus, con diverse varianti implementative, e quella basata sulla scomposizione del modello globale in modelli decentralizzati di ordine ridotto. Per il primo si vuol mostrare come in caso di limitate comunicazioni per periodo di campionamento, sia da preferire la ricerca di una stima quanto più accurata possibile a livello locale, piuttosto che una convergenza alla stima centralizzata che, in certi casi, è ottenibile solo con un notevole numero di comunicazioni. Per il secondo, è interessante notare come un'appropriata decomposizione possa portare a una stima decentralizzata che, nonostante agisca su un modello ridotto, segue, con implementazione diversa, la stessa filosofia del metodo precedente, ottenendo prestazioni confrontabili. Tali soluzioni saranno confrontate sulla base di un semplice esempio, tramite simulazioni.

I. INTRODUZIONE

IL recente sviluppo tecnologico nel campo delle comunicazioni wireless e il contemporaneo diminuire dei costi e della dimensione dei componenti elettronici hanno dato origine ad un rapido sviluppo dei sistemi interconnessi basati su agenti con capacità di misura e di elaborazione integrate. Lo scopo fondamentale di una rete di sensori (*wireless sensor network*) è quello di produrre, su un periodo esteso di tempo, un'informazione globale significativa, ottenuta da una serie di dati locali provenienti dai singoli sensori. Le reti di sensori sono nate per scopi militari, ma il loro campo di applicazione è molto vasto. I nodi sensore che compongono la rete, infatti, possono essere disseminati in larghi spazi e funzionare in condizioni avverse, mettendosi in contatto tra di loro e riconfigurandosi nel caso che uno di essi smettesse di funzionare. Ad esempio, sparsi in una foresta, possono funzionare come rilevatori anti-inquinamento o per la prevenzione degli incendi. I principali campi di applicazione sono:

- civile e domestico: avvistamento d'incendi, monitoraggio della temperatura e della ventilazione in edifici, monitoraggio dell'ambiente, biosensoristica, di-

saster recovery, casa intelligente e domotica, servizi personalizzati e rilevazione delle intrusioni;

- sanitario: interfacce per disabili, monitoraggio integrato di pazienti, monitoraggio e controllo di anziani, controllo remoto di dati fisiologici, tracking di pazienti e medici all'interno di ospedali, somministrazione di farmaci;
- militare: sorveglianza, rilevazione di target, monitoraggio delle forze sul campo, rilevazione di attacchi chimici, biologici e nucleari;
- commerciale: realizzazione di tastiere virtuali, analisi di materiali sottoposti a stress meccanico o strutturale, guida e controllo di robot in industrie automatizzate, monitoraggio e controllo del traffico, controllo di inventari e *precision farming*.

Un problema che caratterizza i sistemi interconnessi è la raccolta di una vasta quantità di misure, provenienti dai vari nodi sensore, in un unico centro di elaborazione. Si crea uno strozzamento del flusso di dati (*bottleneck*) che può dare luogo a problemi di varia natura, come l'eccessivo ritardo di arrivo dei pacchetti o la limitata larghezza di banda della rete. Esiste, pertanto, un forte interesse nella ricerca di soluzioni che permettano di risolvere questo problema eseguendo la stima a livello distribuito. Data la varietà di sistemi fisici sui quali si possono applicare approcci di tipo distribuito, è evidente che non esiste una soluzione ottima che si adatti a tutti i problemi. Ogni caso va studiato tenendo conto delle proprietà dinamiche del sistema e della struttura della rete di sensori che lo monitorano. Esiste una letteratura molto vasta che prende in considerazione i vari aspetti della stima distribuita. Un approccio diffuso che è stato preso in considerazione è quello che si basa su filtri di Kalman distribuiti che utilizzano algoritmi di consensus, si veda in particolare [4]. In [5], si è visto che le tecniche che tentano di replicare, ad ogni nodo, il filtro centralizzato attraverso il consensus non forniscono necessariamente la scelta migliore nel caso che in ogni intervallo di campionamento ci siano poche comunicazioni tra i sensori e necessitino comunque di un elevato scambio di dati. In questa sede, ci si è concentrati su sistemi che per loro natura si prestano particolarmente

alla semplificazione del loro modello di stato in modelli di ordine ridotto, che condividono con i vicini solo un limitato numero di stati. Si tratta di sistemi, anche di vaste dimensioni, composti da sottosistemi simili tra loro, in cui non c'è la necessità che ogni nodo conosca l'intero stato del sistema per effettuare una stima locale sugli stati del modello ridotto di appartenenza. Accontentandosi di avere delle stime locali solo su modelli ridotti, nell'applicazione di algoritmi di consensus è sufficiente, per avere una stima di uguale prestazione di quella del caso centralizzato, un basso numero di comunicazioni per periodo di campionamento. Mentre, per avere ad ogni nodo una conoscenza dell'informazione sullo stato dell'intero sistema, si può pensare di utilizzare un trasferimento delle informazioni locali tra i nodi in maniera meno onerosa computazionalmente rispetto al caso del consensus. Si pensi, ad esempio, ad un monitoraggio ambientale in cui si è in presenza di un territorio decomponibile in micro-aree di caratteristiche simili e aventi interazioni reciproche limitate.

Riprendendo [6] e [7], si presenta una soluzione decentralizzata per la stima dello stato, nella quale le osservazioni non necessitano di essere raccolte presso un'unica unità centrale di elaborazione, perciò è richiesta una capacità computazionale di ordine inferiore. Per fare questo è necessario che il modello globale sia distribuito in modelli ridotti, riducendo i calcoli necessari a livello di ciascun sensore, ma mantenendo comunque l'accoppiamento tra sottosistemi creati. Partendo dal filtro di Kalman in forma di informazione per la stima dello stato, disponendo di un modello distribuito e sfruttando l'intrinseca struttura fisica del sistema, si mostrerà quindi come sia possibile implementare dei filtri di informazione locali per questi modelli ridotti. Tali sottosistemi, relativi a ciascun sensore, raccolgono solo gli stati pertinenti alle osservazioni del sensore stesso, mantenendo comunque l'accoppiamento con i modelli vicini attraverso lo scambio di informazioni tra i filtri locali di informazione.

Con riferimento a [6] e [7], vengono riportati il principio e i passaggi fondamentali per la decomposizione spaziale di un sistema a larga scala, a topologia fissata, il cui modello della dinamica risulta già fissato. La scelta dei sistemi locali può essere fatta in vari modi, qui si fornisce un approccio basato sul sensore; una volta effettuata la scelta, la topologia di accoppiamento tra i sistemi locali è fissata dalla dinamica del sistema globale. Per comodità, si analizzerà quindi questo approccio nel caso di un modello tempo-invariante, ricordando comunque che i risultati possono essere, con opportune modifiche, estesi anche al caso tempo-variante.

L'accoppiamento dei sistemi locali implica un accoppiamento anche tra i filtri di Kalman locali in forma di informazione. Si noti che a causa di tale accoppiamento la matrice di covarianza dell'errore riferita ad ogni sistema locale è una funzione della matrice di covarianza dell'errore globale. Di conseguenza, è necessaria una procedura per calcolare la matrice di covarianza dell'errore globale in modo distribuito. Per raggiungere questo obiettivo, mantenendo la

località dei calcoli, si assume che le matrici di informazione Z , ossia le inverse delle matrici di covarianza dell'errore P , siano L -banded. Tale approssimazione equivale a forzare l'errore di processo nel filtro di Kalman ad essere un processo di Gauss-Markov di ordine L [8], risultando così ottimo secondo Kullback-Leibler [9]. Queste approssimazioni, consentono di passare dalle variabili locali alle variabili globali solamente attraverso l'utilizzo di comunicazioni e variabili locali. L'implementazione del filtro centralizzato di informazione sotto queste ipotesi e approssimazioni risulta essere praticamente indistinguibile dalla forma esatta [10]. Sulla base di un semplice esempio fisico, la conduzione del calore su una barra unidimensionale circolare, si simuleranno i comportamenti dei filtri locali implementati sui modelli distribuiti. Si confronteranno poi le prestazioni ottenute con quelle dei filtri locali che sfruttano la tecnica del consensus.

Si riporta a seguire la struttura di questo lavoro. Nella Sezione II si presentano i filtri di Kalman riscritti in forma di informazione, adatti per la stima a livello distribuito. Lo stimatore ottimo centralizzato costituirà il riferimento in base al quale saranno eseguiti i confronti tra gli altri filtri presentati successivamente. I filtri di Kalman locali sono invece necessari per eseguire la stima dello stato sui modelli ridotti ricavati dalla decomposizione del modello globale. Nella Sezione III si presentano i filtri di Kalman locali in forma di informazione con algoritmi di consensus. Nella Sezione IV si riprende il metodo proposto in [6] per la distribuzione del modello globale. Nella Sezione V si discuterà il metodo proposto in [9] per il calcolo della matrice di covarianza dell'errore in modo distribuito. Si presenterà l'approssimazione di Gauss-Markov dell'errore di stima e l'algoritmo DIC1. Si stabiliranno inoltre le modalità con cui viene effettuata la fusione delle stime. Nella Sezione VI si presentano le condizioni sperimentali in cui ci si è posti nell'esecuzione delle simulazioni. Vengono esposte le caratteristiche del processo fisico e ciò che ne deriva a livello di proprietà dinamiche del sistema. Si presentano inoltre le ipotesi relative all'osservazione del sistema mediante sensori. Nella Sezione VII vengono presentati i risultati ottenuti con i filtri di Kalman locali in forma di informazione che utilizzano la tecnica del consensus in diverse varianti implementative. La Sezione VIII discute i risultati ottenuti dal filtro distribuito, confrontandolo con il precedente. Infine, la Sezione IX riassume i risultati ottenuti e fornisce alcuni spunti per ulteriori sviluppi futuri.

II. FILTRO DI KALMAN IN FORMA DI INFORMAZIONE

Consideriamo un insieme V degli N nodi sensore con cui viene monitorato un processo fisico. I sensori possono comunicare tra loro attraverso una rete tempo invariante che può essere modellizzata da un grafo orientato $\mathcal{G} = (V, E)$ dove l'arco $(i, j) \in E \Leftrightarrow$ il nodo j può trasmettere informazione al nodo i .

Si consideri il modello globale stocastico lineare del sistema da osservare, che evolve secondo l'equazione

$$\begin{cases} x_{k+1} = Ax_k + Bw_k \\ y_k = Cx_k + v_k \end{cases} \quad (1)$$

dove x è il vettore di stato n -dimensionale e y è il vettore delle osservazioni di dimensione p . Le matrici A , B e C rappresentano, rispettivamente, la matrice di stato, la matrice di accoppiamento tra stato e rumore di modello, e la matrice delle osservazioni. I termini, w_k e v_k , rappresentano il rumore di modello e il rumore di osservazione, sono delle variabili aleatorie gaussiane¹ a media nulla e sono incorrelate tra loro e con la variabile aleatoria x_0 che descrive lo stato iniziale. In seguito sono riportate le statistiche che caratterizzano le suddette variabili aleatorie:

$$\begin{cases} w_k \sim \mathcal{N}(0, BSB^T), & \mathbb{E}[w_k w_h] = BSB^T \delta(k-h) \\ v_k \sim \mathcal{N}(0, R), & \mathbb{E}[v_k v_h] = R\delta(k-h) \\ x_0 \sim \mathcal{N}(\bar{x}_0, P_0) \end{cases} \quad (2)$$

Inoltre il rumore di osservazione relativo a ciascun sensore è scorrelato da quello degli altri, pertanto la varianza di v_k può essere scritta come una matrice diagonale a blocchi.

$$R = \begin{bmatrix} R_1 & & 0 \\ & \ddots & \\ 0 & & R_N \end{bmatrix} \quad (3)$$

Il filtro di Kalman, nella sua formulazione classica, si definisce come:

$$\hat{x}_{k+1|k+1} = \mathbb{E}[x_{k+1}|y_0, \dots, y_{k+1}] = \mathbb{E}[x_{k+1}|y_{k+1}, Y^k] \quad (4)$$

dove $Y^k = (y_k, \dots, y_1, y_0)$. Nel caso, come quello in esame, di avere N misure scorrelate, risulta conveniente sfruttare la forma di informazione (o di covarianza), ricavabile applicando il Lemma di Inversione di Matrice. Per la dimostrazione del lemma si veda l'Appendice A. Ne risulta che l'espressione del filtro centralizzato è la seguente:

$$\begin{aligned} \hat{x}_{k+1|k+1}^c &= P_{k+1|k+1} \left(P_{k+1|k}^{-1} \hat{x}_{k+1|k}^c + \sum_{i=1}^N C^{iT} R^{i-1} y_{k+1}^i \right) \\ P_{k+1|k+1} &= \left(P_{k+1|k}^{-1} + \sum_{i=1}^N C^{iT} R^{i-1} C^i \right)^{-1} \end{aligned} \quad (5)$$

Tramite la forma di informazione si riesce a dare al filtro di Kalman una forma computazionalmente più semplice, ma anche più adatta al partizionamento per la stima decentralizzata tramite sistemi multi-sensore. Per distribuire la stima del vettore globale di stato x , si vanno a realizzare dei filtri locali basati sull'intero modello, ognuno facente

capo a ciascuno dei $i = 1, \dots, N$ nodi sensore tramite i quali è monitorato il sistema. Il processo di osservazione del sistema y si può quindi partizionare in N vettori y_1, \dots, y_N esprimibili come $y^i = C^i x + v^i$, ai quali vengono associate le relative stime locali $\hat{x}^i = \hat{E}[x|y_i]$. Ogni stimatore viene calcolato solamente in base all'informazione disponibile al sensore stesso e alle stime precedenti; in altre parole non avviene alcun tipo di comunicazione tra i sistemi locali, non ci si avvale della tecnica del consensus. Togliendo le medie, le stime locali, suddivise in predizione e aggiornamento, si possono esprimere con le seguenti equazioni:

- *Stime a priori* (Aggiornamento temporale)

$$\begin{aligned} \hat{x}_{k+1|k}^i &= A\hat{x}_{k|k}^i \\ P_{k+1|k} &= AP_{k|k}A^T + Q \end{aligned} \quad (6)$$

- *Stime a posteriori* (Aggiornamento rispetto alle misure)

$$\begin{aligned} \hat{x}_{k+1|k+1}^i &= P_{k+1|k+1} P_{k+1|k}^{-1} \hat{x}_{k+1|k}^i + \\ &+ P_{k+1|k+1} C^{iT} R^{i-1} y_{k+1}^i \\ P_{k+1|k+1} &= P_{k+1|k} - P_{k+1|k} \cdot \\ &\cdot C^{iT} \left(C^i P_{k+1|k} C^{iT} + R \right)^{-1} C^i P_{k+1|k} \end{aligned} \quad (7)$$

III. FILTRI DI KALMAN LOCALI IN FORMA DI INFORMAZIONE CON ALGORITMI DI CONSENSUS

In questa sezione si intende mostrare che la forma di informazione del filtro di Kalman per una rete di sensori può essere espressa equivalentemente tramite problema di consensus. Si consideri, innanzitutto, l'espressione del filtro centralizzato (5). E' possibile darne la seguente riscrittura, in termini di equazioni di predizione e di aggiornamento.

- *Filtro centralizzato* (Predizione e Aggiornamento)

$$\begin{aligned} \hat{x}_{k+1|k}^c &= A\hat{x}_{k|k}^c \\ \hat{x}_{k+1|k+1}^c &= P_{k+1|k+1} \left(P_{k+1|k}^{-1} \hat{x}_{k+1|k}^c + \sum_{i=1}^N C^{iT} R^{i-1} y_{k+1}^i \right) \\ &= L\hat{x}_{k+1|k}^c + NP_{\infty|\infty} \frac{1}{N} \sum_{i=1}^N C^{iT} R^{i-1} y_{k+1}^i \end{aligned} \quad (8)$$

Facendo riferimento all'equazione di aggiornamento, si noti che il primo termine è costante, dipende dalla stima al passo precedente $\hat{x}_{k+1|k}^c$, moltiplicata per una matrice costante $L = P_{\infty|\infty} P_{\infty+1|\infty}^{-1}$. Il secondo termine, invece, può essere espresso in funzione della media dei vettori di informazione. Tale media è ottenibile tramite l'applicazione della tecnica del consensus. Per ogni sensore, viene acquisito il segnale misurato e viene mediato assieme a tutte le altre misure. Analogamente, gli N filtri locali si

¹Si ricordi che il simbolo \mathcal{N} indica una v.a. gaussiana

possono scrivere come:

- *Filtri locali* (Predizione e Aggiornamento)

$$\begin{aligned}\hat{x}_{k+1|k}^i &= A\hat{x}_k^i \\ \hat{x}_{k+1|k+1}^i &= L\hat{x}_{k+1|k}^i + NP_{\infty|\infty}C^{iT}R^{i-1}y_{k+1}^i\end{aligned}\quad (9)$$

E' possibile poi realizzare un filtro che implementi l'algoritmo di consensus. Tra un istante di campionamento e il seguente, avviene un numero L di comunicazioni locali tra ogni nodo sensore e i suoi vicini. Le comunicazioni avvengono secondo il seguente schema.

- *Inizializzazione* (Aggiornamento e Predizione)

$$\begin{aligned}\hat{x}_{k+1|k+1,0}^i &= L\hat{x}_{k+1|k}^i + My_{k+1}^i \\ \hat{x}_{k+2|k+1,0}^i &= A\hat{x}_{k+1|k+1,L-1}^i\end{aligned}\quad (10)$$

Al primo istante di comunicazione, la predizione viene inizializzata all'ultimo valore del filtro, relativo al precedente istante di campionamento; l'aggiornamento viene fatto sulla base della nuova misura. Si tenga conto che $M = NP_{\infty|\infty}C^{iT}R^{i-1}$. L'evoluzione dell'algoritmo per le successive comunicazioni segue invece la relazione riportata in seguito. Quando si arriva all'ultimo istante di comunicazione, il valore dello stimatore tende alla media delle condizioni iniziali del filtro.

- *Evoluzione*

$$\begin{aligned}\hat{x}_{k+1|k+1,l+1}^i &= \sum_{j \in \mathcal{N}(i)} Q_{ij}\hat{x}_{k+1|k+1,l}^j \\ \hat{x}_{k+1|k+1,L-1}^i &\longrightarrow \frac{1}{N} \sum_{j=1}^N \hat{x}_{k+1|k+1,0}^j\end{aligned}\quad (11)$$

$$\sum_{j \in \mathcal{N}(i)} Q_{ij}C^{jT}R^{j-1}y_{k+1}^j \longrightarrow \frac{1}{N} \sum_{j=1}^N C^{jT}R^{j-1}y_{k+1}^j$$

Dove $Q_{ij} \in \mathcal{R}^{N \times N}$ è la matrice di consensus associata al grafo di comunicazione \mathcal{G} . Contiene i pesi associati agli archi $(i, j) \in E$ tali per cui il nodo j può trasmettere informazione al nodo i .

IV. DISTRIBUZIONE DEL MODELLO PER FILTRAGGIO DI KALMAN DISTRIBUITO

A titolo di esempio, si consideri la fig.1, dove è riportato il grafo orientato di un sistema su larga scala. I cerchi rappresentano le variabili di stato e le frecce mostrano l'accoppiamento dinamico di esse. Ovvero, l'arco $(i, j) \in E$, cioè $E_{j,i} = 1$ se $A_{j,i} \neq 0$, dove E è la matrice delle adiacenze del grafo $\mathcal{G} = (V, E)$ e A è la matrice di evoluzione dello stato. La fig.2 mostra una possibile scelta dei sistemi locali, racchiusi negli ovali, per il sistema in

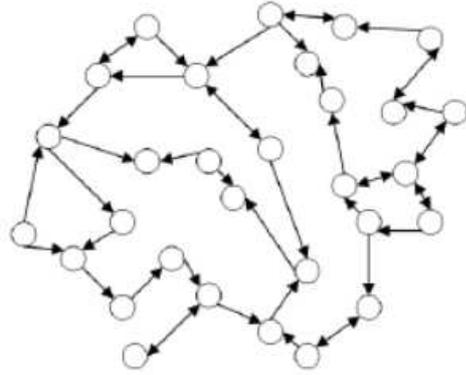


Fig. 1. Grafo di un generico sistema su larga scala

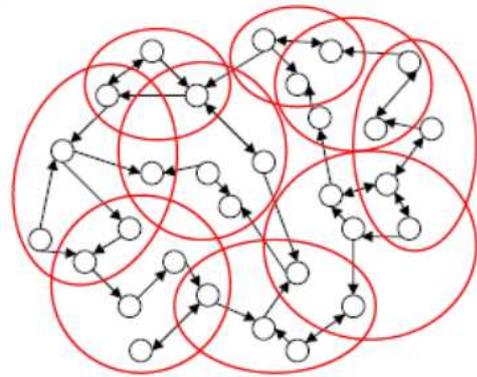


Fig. 2. Individuazione dei sistemi locali

fig.1; tale scelta viene compiuta associando, a ciascuno dei sistemi locali, un sensore. I sottosistemi comprendono quindi tutti gli stati che il sensore può osservare direttamente o indirettamente. Con indirettamente si intende che il sensore osserva una combinazione lineare di alcuni stati. Si suppone, senza perdita di generalità, che ogni stato sia osservato (direttamente o indirettamente) da almeno un sensore. Partendo dal modello generale (1), si mostra ora come distribuire il sistema globale per ricavare i sistemi locali di ordine ridotto. Su questi modelli, si è poi in grado di implementare i filtri di Kalman locali in forma di informazione associati a ciascun sensore, che risultano computazionalmente efficienti. La spiegazione della procedura per la distribuzione del modello, sarà, per maggior chiarezza, accompagnata da un'esempio, che considera il sistema di dimensione $n = 5$ con $N = 3$ sensori, descritto dal grafo orientato $\mathcal{G} = (V, E)$ in fig.3. I cerchi e i quadrati, associati ai vertici V rappresentano gli stati $\{x_i\}$ e gli ingressi $\{u_i\}$; i triangoli rappresentano i sensori; le frecce, definite dalla matrice delle adiacenze E in (14), corrispondono ai legami della dinamica. Ogni sensore l , può osservare gli n_l stati compresi nell'ellisse corrispondente, che delimita l'insieme di taglio $V^{(l)}$.

Tale sistema è descritto dal modello:

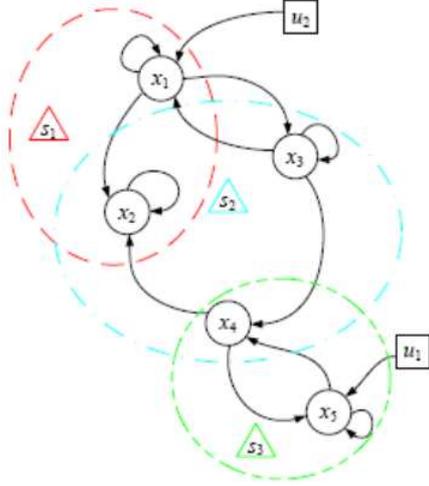


Fig. 3. Esempio di individuazione dei sistemi locali

$$\begin{aligned}
 x_{k+1} &= \begin{bmatrix} a_{11} & 0 & a_{13} & 0 & 0 \\ a_{21} & a_{22} & 0 & a_{24} & 0 \\ a_{31} & 0 & a_{33} & 0 & 0 \\ 0 & 0 & a_{43} & 0 & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix} x_k \\
 &+ \begin{bmatrix} 0 & b_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ b_{51} & 0 \end{bmatrix} u_k \\
 &= Ax_k + Bu_k \\
 \\
 y_k &= \begin{bmatrix} y_k^{(1)} \\ y_k^{(2)} \\ y_k^{(3)} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & 0 & 0 & 0 \\ 0 & c_{22} & c_{23} & c_{24} & 0 \\ 0 & 0 & 0 & c_{34} & c_{35} \end{bmatrix} x_k \\
 &+ \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \\ w_k^{(3)} \end{bmatrix} \\
 &= Cx_k + w_k
 \end{aligned} \tag{12}$$

dove $u_k \sim \mathcal{N}(0, Q)$ è un rumore bianco in ingresso e $w_k \sim \mathcal{N}(0, R)$ è il rumore di misura. Si tengano presente le ipotesi fatte nella Sezione II per le quali le sequenze $u_{k \geq 0}$, $w_{k \geq 0}$ e la condizione iniziale x_0 , sono tutte statisticamente indipendenti. Si è inoltre assunto che $R = \text{blockdiag}[R_1, R_2, \dots, R_N]$, ovvero che i rumori di misura dei vari sensori siano incorrelati. È ovviamente richiesta anche l'osservabilità della coppia (A, C) . La matrice delle adiacenze E del sistema (12) è la seguente:

$$E = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{14}$$

La scomposizione della dinamica, secondo gli insiemi di taglio $V^{(l)}$ basati sui sensori, assume una forma del tipo seguente, nella quale l individua il generico sensore:

$$\begin{aligned}
 x_{k+1}^{(l)} &= A^{(l)}x_k^{(l)} + D^{(l)}d_k^{(l)} + B^{(l)}u_k^{(l)} \\
 y_k^{(l)} &= C^{(l)}x_k^{(l)} + w_k^{(l)}
 \end{aligned} \tag{15}$$

dove $x_k^{(l)}$ è il vettore di stato del sistema ridotto e indica l'insieme degli stati osservabili dall' l -simo sensore; $A^{(l)}$ è la porzione di A che descrive l'evoluzione di stato di tali stati osservabili; $d_k^{(l)}$ raccoglie le componenti di x_k che entrano come ingressi nella dinamica di $x_k^{(l)}$, in quanto collegati tramite una freccia entrante nell'insieme di taglio $V^{(l)}$; $D^{(l)}$ è formata dagli elementi di A che descrivono il legame tra $x_k^{(l)}$ e $d_k^{(l)}$; $B^{(l)}$ raccoglie le componenti di B che legano $x_k^{(l)}$ a $u_k^{(l)}$; $C^{(l)}$ è la parte di C relativa a $x_k^{(l)}$; infine $w_k^{(l)}$ è il rumore di misura corrispondente al sensore l . Considerando, ad esempio, il sensore $l = 1$, per il quale $x^{(1)} = [x_1, x_2]$, $V^{(1)} = [x_1, x_2, u_2]$ la decomposizione del modello restituisce:

$$\begin{aligned}
 x_{k+1}^{(1)} &= \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} x_k^{(1)} \\
 &+ \begin{bmatrix} a_{13} & 0 \\ 0 & a_{24} \end{bmatrix} \begin{bmatrix} x_{3,k} \\ x_{4,k} \end{bmatrix} + \begin{bmatrix} b_{12} \\ 0 \end{bmatrix} u_{2,k} \\
 &= A^{(1)}x_k^{(1)} + D^{(1)}d_k^{(1)} + B^{(1)}u_k^{(1)} \\
 y_k^{(1)} &= C^{(1)}x_k^{(1)} + w_k^{(1)}
 \end{aligned} \tag{16}$$

Nel caso le componenti del vettore di stato $d_k^{(l)}$ non siano disponibili ai corrispondenti sensori di osservazione, si utilizzano le loro stime aggiornate $\hat{d}_{k|k}^{(l)}$, in questa maniera il modello ridotto associato al generico sensore l diventa:

$$\begin{aligned}
 x_{k+1}^{(l)} &= A^{(l)}x_k^{(l)} + D^{(l)}\hat{d}_{k|k}^{(l)} + B^{(l)}u_k^{(l)} \\
 y_k^{(l)} &= C^{(l)}x_k^{(l)} + w_k^{(l)}
 \end{aligned} \tag{17}$$

I filtri locali in forma di informazione (analoghi a quelli descritti in Sezione II) utilizzati per la stima dello stato, si applicano, in questo caso, ai modelli ridotti basati

sul sensore (15). Si osservi che questi modelli possono condividere degli stati tra diversi insiemi di taglio, come si nota anche in fig.3, sarà pertanto necessario effettuare una fusione delle diverse osservazioni che riguardano le medesime componenti dello stato, al fine di ottenere le prestazioni del filtro centralizzato (5). Per i filtri di informazione locali, la fusione delle osservazioni coinvolge solo un particolare sottoinsieme delle osservazioni di sensori vicini, quindi comporta solo comunicazione locale e computazioni che coinvolgono le variabili del modello ridotto.

V. CALCOLO DELLA MATRICE DI COVARIANZA DELL'ERRORE E FUSIONE DELLE STIME

Un aspetto fondamentale per l'implementazione è quello che riguarda il calcolo della matrice di covarianza dell'errore P in modo distribuito. Come già accennato, l'accoppiamento dei sistemi locali implica un accoppiamento anche tra i filtri di Kalman locali in forma d'informazione. A causa di ciò, la matrice di varianza dell'errore riferita ad ogni sistema locale è funzione della matrice di varianza dell'errore globale. Nasce quindi l'esigenza di avere una procedura in grado di calcolare la matrice di covarianza dell'errore globale in modo distribuito. Per raggiungere questo obiettivo, mantenendo nel contempo la località dei calcoli, si assume che le matrici di informazione Z , ossia le inverse delle matrici di covarianza dell'errore P , siano L -banded. Tale approssimazione equivale a forzare l'errore di processo nel filtro di Kalman ad essere un processo di Gauss-Markov di ordine L [8], risultando così ottimo secondo Kullback-Leibler [9]. Queste approssimazioni, consentono di passare dalle variabili locali alle variabili globali solamente attraverso l'utilizzo di comunicazioni e variabili locali.

A. Approssimazione di Gauss-Markov dell'errore di stima

Gli errori Gaussiani di stima e predizione del filtro di Kalman sono definiti come

$$\begin{aligned}\epsilon_{k|k} &= x_k - \hat{x}_{k|k} \\ \epsilon_{k+1|k} &= x_k - \hat{x}_{k+1|k}\end{aligned}\quad (18)$$

e sono completamente caratterizzati dalle loro matrici di covarianza, $P_{k|k}$ e $P_{k+1|k}$ di dimensione $n \times n$. Per ridurre l'onerosità computazionale richiesta dal manipolare tali matrici ad ogni sensore, si approssima il processo di errore gaussiano come un processo di Gauss-Markov di ordine L . Questo equivale ad approssimare $Z = P^{-1}$, inversa della matrice di covarianza dell'errore, con una matrice di tipo L -banded²[8], il che risulta ottimo secondo Kullback-Leibler o secondo il criterio di massima entropia [9]. Con

²Si dice che una matrice è L -Banded ($L \geq 0$), se gli elementi al di fuori della fascia delimitata dalla L -esima diagonale, superiore e inferiore, sono uguali a zero

questa approssimazione sul processo che descrive l'errore, le matrici di covarianza $P_{k|k}$ e $P_{k+1|k}$ sono completamente caratterizzate dalle loro L sovra e sottodiagonali. Infatti, qualsiasi elemento al di fuori di queste, in una matrice la cui inversa è L -banded, può essere scritto come una funzione degli elementi all'interno delle diagonali [10]. In questo modo, preservare la Gauss-Markovianità dell'errore di stima equivale a preservare, per le matrici di covarianza $P_{k|k}$ e $P_{k+1|k}$, la caratteristica di avere inverse L -banded. Facendo uso di questa approssimazione, si procede con la definizione dei filtri di Kalman locali in forma di informazione, del tutto analogamente a quanto visto in Sezione II.

Si può verificare che la matrice locale di covarianza dell'errore è data da:

$$P_{k+1|k}^{(l)} = A_l P_{k|k} A_l^T + B^{(l)} Q^{(l)} B^{(l)T} \quad (19)$$

dove A_l è una sottomatrice $n_l \times n$ di A , nella quale le n_l righe corrispondono agli n_l stati osservati dal sensore l . Inoltre, si nota la dipendenza di $P_{k+1|k}^{(l)}$ dalla covarianza d'errore globale $P_{k|k}$. Partendo da (15), è possibile scrivere un'espressione equivalente alla (19) per la matrice locale di covarianza $P_{k+1|k}^{(l)}$, utilizzando solo le sottomatrici locali $A^{(l)}$ e $D^{(l)}$. Questa espressione richiede le sottomatrici $P_{k|k}^{(\cdot)}$ di $P_{k|k}$, dai sensori vicini. $P_{k|k}^{x^{(l)}d^{(l)}}$ è la covarianza incrociata dell'errore di stima tra le componenti dello stato osservate dal sensore l e quelle di $\hat{d}_{k|k}^{(l)}$ osservate dai vicini. Mentre $P_{k|k}^{d^{(l)}d^{(l)}}$ è la covarianza incrociata del vettore degli ingressi $d^{(l)}$.

$$\begin{aligned}P_{k+1|k}^{(l)} &= A^{(l)} P_{k|k}^{(l)} A^{(l)T} + A^{(l)} P_{k|k}^{x^{(l)}d^{(l)}} D^{(l)T} \\ &+ \left(A^{(l)} P_{k|k}^{x^{(l)}d^{(l)}} D^{(l)T} \right)^T + D^{(l)} P_{k|k}^{d^{(l)}d^{(l)}} D^{(l)T} \\ &+ B^{(l)} Q^{(l)} B^{(l)T}\end{aligned}\quad (20)$$

B. Algoritmo DICI per il calcolo distribuito della matrice di covarianza dell'errore

Al fine di distribuire e ridurre la complessità computazionale a livello di ciascun sensore, si vuole un metodo per il calcolo distribuito delle varie sottomatrici $P_{k|k}^{(\cdot)}$ di $P_{k|k}$. Per far questo si utilizza l'algoritmo DICI (Distributed Iterate-Collapse Inversion) per il calcolo iterativo dell'inversione di matrice, allo scopo di ricavare la matrice locale di covarianza dell'errore $P^{(l)}$ da quella locale di informazione $Z^{(l)}$. Prima di tutto, per facilitare l'esposizione si introducono le notazioni:

$$P_{k+1|k+1} = Z_{k+1|k+1}^{-1} \quad (21)$$

$$P_{k+1|k} = Z_{k+1|k}^{-1} \quad (22)$$

$$\hat{z}_{k+1|k} = Z_{k+1|k} \hat{x}_{k+1|k} \quad (23)$$

$$\hat{z}_{k+1|k+1} = Z_{k+1|k+1} \hat{x}_{k+1|k+1} \quad (24)$$

Si definiscono le variabili di osservazione globale n -dimensionali:

$$i_k = C^T R^{-1} y_k \quad (25)$$

$$\mathcal{I} = C^T R^{-1} C \quad (26)$$

e le variabili di osservazione locale n -dimensionali al sensore l

$$i_{l,k} = C_l^T R_l^{-1} y_k^{(l)} \quad (27)$$

$$\mathcal{I}_l = C_l^T R_l^{-1} C_l \quad (28)$$

Si noti che $C_l \neq C^{(l)}$. Infatti, C_l ha dimensione pari a quella del vettore di stato x , dal quale seleziona le sole componenti osservate dall' l -simo sensore. Si noti inoltre come le osservazioni globali possano essere espresse come fusione delle osservazioni locali degli N sensori:

$$\begin{aligned} i_k &= C^T R^{-1} y_k \\ &= C_1^T R_1^{(-1)} y_k^{(1)} + \dots + C_N^T R_N^{(-1)} y_k^{(N)} \\ &= \sum_{l=1}^N i_{l,k} \end{aligned} \quad (29)$$

Analogamente:

$$\mathcal{I} = \sum_{l=1}^N \mathcal{I}_l \quad (30)$$

Con queste notazioni è possibile riscrivere il filtro di Kalman centralizzato in forma d'informazione (5). Il passo di aggiornamento risulta:

• Aggiornamento

$$\begin{aligned} Z_{k+1|k+1} &= Z_{k+1|k} + \sum_{l=1}^N \mathcal{I}_l \quad k \geq -1 \\ \hat{z}_{k+1|k+1} &= \hat{z}_{k+1|k} + \sum_{l=1}^N i_{l,k+1} \quad k \geq -1 \end{aligned} \quad (31)$$

Mentre quello di predizione è dato da:

• Predizione

$$\begin{aligned} Z_{k+1|k} &= P_{k+1|k}^{-1} = \left(AZ_{k|k}^{-1} A^T + BQB^T \right)^{-1} \\ k &\geq 0, \quad Z_{0|-1} = P_0^{-1} \end{aligned} \quad (32)$$

$$\begin{aligned} \hat{z}_{k+1|k} &= Z_{k+1|k} \left(AZ_{k|k}^{-1} \hat{z}_{k|k} \right) \\ k &\geq 0, \quad \hat{z}_{0|-1} = 0 \end{aligned}$$

In modo analogo a quanto visto in precedenza, si può giungere ad una formulazione dei filtri di Kalman locali in forma di informazione per i sottosistemi associati a ciascun sensore, secondo queste nuove variabili. Le matrici locali di covarianza dell'errore, $P_{k|k}^{(l)}$ e $P_{k+1|k}^{(l)}$, nel caso in cui una stessa componente dello stato x_k venga stimata da più di un sensore, risultano parzialmente sovrapposte. Analogamente anche le matrici di informazione locali $Z_{k|k}^{(l)}$ e $Z_{k+1|k}^{(l)}$ risulteranno sovrapposte, come si può notare nella (33), che si riferisce all'esempio in considerazione nel caso in cui Z sia L -banded con $L = 1$. Si noti inoltre che $P^{(l)} \neq (Z^{(l)})^{-1}$.

$$\begin{aligned} P &= \begin{bmatrix} \lceil p_{11} & p_{12} \rceil & p_{13} & p_{14} & p_{15} \\ \lfloor p_{21} & \lceil p_{22} \rceil & p_{23} & p_{24} \rceil & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & \lfloor p_{42} & p_{43} & \lceil p_{44} \rceil & p_{45} \rceil \\ p_{51} & p_{52} & p_{53} & \lfloor p_{54} & p_{55} \rfloor \end{bmatrix} \\ &= \begin{bmatrix} \lceil z_{11} & z_{12} \rceil & 0 & 0 & 0 \\ \lfloor z_{21} & \lceil z_{22} \rceil & z_{23} & 0 \rceil & 0 \\ 0 & z_{32} & z_{33} & z_{34} & 0 \\ 0 & \lfloor 0 & z_{43} & \lceil z_{44} \rceil & z_{45} \rceil \\ 0 & 0 & 0 & \lfloor z_{54} & z_{55} \rfloor \end{bmatrix}^{-1} = Z^{-1} \end{aligned} \quad (33)$$

Si può dimostrare, (si veda [7], sez. VII-A), che la matrice locale di informazione predetta $Z_{k+1|k}^{(l)}$ è funzione della covarianza dell'errore locale $P_{k|k}^{(l)}$, che si vuole a sua volta calcolare dalla matrice d'informazione locale $Z_{k|k}^{(l)}$ ottenuta con il passo di aggiornamento (31). In Appendice B si presentano i passaggi fondamentali per la comprensione dell'algoritmo per il calcolo distribuito della matrice di covarianza dell'errore.

C. Fusione delle osservazioni

Le relazioni (29) e (30) mostrano come, nel caso centralizzato, la fusione delle osservazioni equivale a sommare le variabili di informazione n -dimensionali $i_{l,k}$ e \mathcal{I}_l . Nel caso distribuito, invece, le variabili di osservazione locale, n_l -dimensionali³, date da:

³Si noti la differenza tra le notazioni con apice (l) , che indicano variabili di ordine ridotto al sensore l (vettori di dimensioni $n_l \times 1$ o matrici $n_l \times n_l$) e quelle con pedice l , che si riferiscono a variabili n -dimensionali.

$$i_k^{(l)} = (C^{(l)})^T R^{(l)-1} y_k^{(l)} \quad (34)$$

$$\mathcal{I}^{(l)} = (C^{(l)})^T R^{(l)-1} C^{(l)} \quad (35)$$

relative a diversi vettori di stato locali $x_k^{(l)}$, non possono essere sommate direttamente. Senza approfondire il metodo di fusione, basato sul *bipartite fusion graph* (si veda [11]), si riporta l'idea fondamentale sulla quale si basa. Il metodo prevede che ad ogni sensore, le componenti del vettore d'informazione vengano sommate a quelle relative alla medesima componente x_j dello stato osservato, provenienti dagli altri sensori che la osservano, e che riescono a comunicare direttamente. Nel caso del sistema considerato come esempio, questo comporta che i vettori d'informazione a ciascun sensore:

$$i_k^{(1)} = \begin{bmatrix} i_{k,x_1}^{(1)} \\ i_{k,x_2}^{(1)} \end{bmatrix}, \quad i_k^{(2)} = \begin{bmatrix} i_{k,x_2}^{(2)} \\ i_{k,x_3}^{(2)} \\ i_{k,x_4}^{(2)} \end{bmatrix}, \quad i_k^{(3)} = \begin{bmatrix} i_{k,x_4}^{(3)} \\ i_{k,x_5}^{(3)} \end{bmatrix} \quad (36)$$

vengano fusi a formare i vettori $I_{f,k}^{(l)}$:

$$i_{f,k}^{(1)} = \begin{bmatrix} i_{k,x_1}^{(1)} \\ i_{k,x_2}^{(1)} + i_{k,x_2}^{(2)} \end{bmatrix}, \quad i_{f,k}^{(2)} = \begin{bmatrix} i_{k,x_2}^{(2)} + i_{k,x_2}^{(1)} \\ i_{k,x_3}^{(2)} \\ i_{k,x_4}^{(2)} + i_{k,x_4}^{(3)} \end{bmatrix}, \quad i_{f,k}^{(3)} = \begin{bmatrix} i_{k,x_4}^{(3)} + i_{k,x_4}^{(2)} \\ i_{k,x_5}^{(3)} \end{bmatrix}. \quad (37)$$

Generalizzando, a ciascun sensore l si può scrivere il vettore $i_{f,k}^{(l)}$, risultato della fusione delle osservazioni, come raccolta delle sue componenti $i_{f,k,x_j}^{(l)}$ relative alla osservazione di x_j (componente dello stato).

$$i_{f,k,x_j}^{(l)} = \sum_{s \in \mathcal{N}_{l,j}} i_{k,x_j}^{(s)} \quad (38)$$

Dove $\mathcal{N}_{l,j}$ indica l'insieme dei sensori vicini a l , che come l osservano la componente j -esima dello stato x e possono comunicare direttamente con l . Nel caso in cui tutti i sensori che osservano la medesima componente non riescano a comunicare direttamente, per l'implementazione della fusione si utilizza un algoritmo iterativo di media pesata (consensus).

Una procedura simile può essere utilizzata anche per la fusione delle matrici d'informazione $\mathcal{I}^{(l)}$. Tuttavia, assumendo che il modello di osservazione del sistema sia stazionario (C ed R tempo-invarianti), le matrici $\mathcal{I}^{(l)}$, e conseguentemente $\mathcal{I}_f^{(l)}$, possono essere calcolate una sola volta attraverso un procedimento fuori linea.

D. Riassunto passaggi per l'implementazione

Si riportano i principali passaggi, seguiti per l'implementazione del metodo del filtro di Kalman in forma d'informazione, basato sul modello ridotto.

D.1 Condizioni iniziali

Dalle consuete condizioni iniziali $\hat{x}_{0|-1} = 0$, $P_{0|-1} = I_{n \times n}$, si ricavano facilmente quelle per il modello distribuito $\hat{x}_{0|-1}^{(l)} = 0$, $P_{0|-1}^{(l)} = I_{n_l \times n_l}$; da queste, attraverso il Lemma di Inversione di Matrice si trovano $Z_{0|-1}^{(l)}$ e $\hat{z}_{0|-1}^{(l)}$.

D.2 Passo di aggiornamento

Il passo di aggiornamento è implementato usando le variabili in forma di informazione, ed è dato da:

$$Z_{k|k}^{(l)} = Z_{k|k-1}^{(l)} + \mathcal{I}_f^{(l)} \quad (39)$$

$$\hat{z}_{k|k}^{(l)} = \hat{z}_{k|k-1}^{(l)} + i_{f,k}^{(l)}$$

dove $\mathcal{I}_f^{(l)}$, $i_{f,k}^{(l)}$, sono le variabili ricavate dalla fusione; si veda la Sezione V-C. Dalle matrici di informazione locale $Z_{k|k}^{(l)}$, attraverso l'algoritmo (*DICI-OR*), si ottengono le matrici locali di covarianza dell'errore $P_{k|k}^{(l)}$. Infine, utilizzando la (24), si convertono le stime $\hat{z}_{k|k}^{(l)}$ dal dominio dell'informazione a quello del filtro di Kalman $\hat{x}_{k|k}^{(l)}$.

D.3 Passo di predizione

Il passo di predizione si effettua per semplicità nella variabile di stato x , nella forma direttamente deducibile dalla (17), nel modo seguente:

$$\hat{x}_{k+1}^{(l)} = A^{(l)} \hat{x}_{k|k}^{(l)} + D^{(l)} \hat{d}_{k|k}^{(l)} \quad (40)$$

Per il calcolo della varianza dell'errore di predizione $P_{k+1|k}^{(l)}$, come già detto in Sezione V-A, si utilizza la (20). In tale relazione si ha che: $P_{k|k}^{(l)}$ è la matrice locale di covarianza dell'errore; $P_{k|k}^{d^{(l)}}$ è la matrice locale di covarianza dell'errore riferita a $d_k^{(l)}$, che si ricava dai sensori che hanno le componenti di $d_k^{(l)}$ nel loro modello ridotto; $P_{k|k}^{x^{(l)}d^{(l)}}$ è la covarianza incrociata tra lo stato del modello ridotto $x_k^{(l)}$ e il vettore d'ingresso locale $d_k^{(l)}$. Infine, la matrice di informazione locale $Z_{k+1|k}^{(l)}$, necessaria per convertire le stime dalla variabile $\hat{x}_{k+1|k}^{(l)}$ in $\hat{z}_{k+1|k}^{(l)}$, è ricavata attraverso l'opportuna inversione di $P_{k+1|k}^{(l)}$, che necessita esclusivamente di comunicazioni locali con i vicini.

VI. APPLICAZIONE: CONDUZIONE DEL CALORE IN UNA BARRA UNIDIMENSIONALE CIRCOLARE

In questa sezione, i risultati teorici sin qui ottenuti, si andranno ad applicare a un particolare processo fisico, la conduzione del calore. Questo fenomeno può essere

descritto tramite un'equazione differenziale alle derivate parziali (PDE), che modella l'andamento della funzione temperatura, in una particolare regione dello spazio e sotto opportune condizioni, in funzione del tempo. Per una funzione u , definita nelle coordinate spaziali (x,y,z) , l'equazione si presenta nella forma:

$$\frac{\partial u}{\partial t} = k \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (41)$$

dove k rappresenta la conduttività termica del materiale; per dettagli circa la derivazione dell'equazione si rimanda a [12] e [13]. In questo caso si è scelto di esaminare la conduzione del calore in una barra unidimensionale di forma circolare, essendo tale struttura particolarmente indicata per eseguire i calcoli relativi alla distribuzione del modello. Tuttavia la descrizione microscopica di diffusione che abbiamo dato per l'equazione del calore, al fine di essere implementata per via numerica, necessita di una rispettiva versione discretizzata. Nel caso in esame è stata ottenuta usando il metodo delle differenze finite, secondo la seguente equazione:

$$\begin{aligned} u_j^{n+1} &= u_j^n + \frac{k\Delta t}{\Delta x^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \\ &= q (u_{j+1}^n + u_{j-1}^n) + (1 - 2q)u_j^n \end{aligned} \quad (42)$$

Ciascuna sezione del conduttore, all'istante temporale seguente, cede a ciascuno dei propri due vicini una percentuale q del calore che possiede, e mantiene la rimanente parte $1-2q$. La temperatura in una certa sezione della barra, al passo successivo, è pertanto una media pesata della propria temperatura e di quella dei vicini al passo temporale corrente. Ne consegue che la matrice A che si ottiene, che caratterizza la dinamica del modello, ha struttura circolante.

$$A = \begin{bmatrix} 1-2q & q & 0 & 0 & q \\ q & 1-2q & q & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & q & 1-2q & q \\ q & 0 & 0 & q & 1-2q \end{bmatrix} \quad (43)$$

In fig.4 seguente viene data una rappresentazione sul piano xy della barra circolare (in blu) e della distribuzione iniziale di temperatura scelta per le simulazioni (in rosso). Si è scelto di simulare una situazione nella quale la barra ha una lunghezza di 10m e ciascuno stato corrisponde ad una sezione di barra lunga 0.5m : si ricava pertanto una dimensione dello stato di $n = 20$. La distribuzione iniziale di temperatura scelta è di tipo sinusoidale; all'intera lunghezza della barra viene fatto corrispondere un periodo della sinusoide.

Per effetto della conduzione del calore si ha che dopo 4 secondi, la durata prevista delle simulazioni, il calore si propaga lungo la barra fino ad avere un valore pressoché

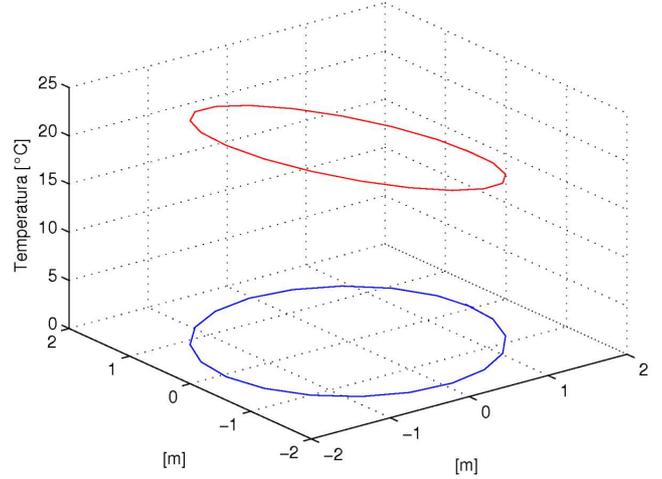


Fig. 4. Distribuzione di temperatura iniziale nella barra di conduzione

uniforme. Nelle simulazioni è stato scelto un valore di conduttività termica pari a $2 \text{ W/m} \cdot \text{K}$, pertanto si può pensare che la barra possieda delle proprietà di conduzione paragonabili a quelle di un materiale vetroso. Si suppone che nel processo non agiscano ingressi esterni, ma siano solamente presenti dei disturbi di modello e di osservazione di tipo gaussiano. Il sistema evolve quindi in evoluzione libera e il calore si propaga solamente per l'effetto della conduzione attraverso la barra. In fig.5 si riporta l'andamento della temperatura nella barra circolare in funzione del tempo. Per comodità la barra è visualizzata svolta.

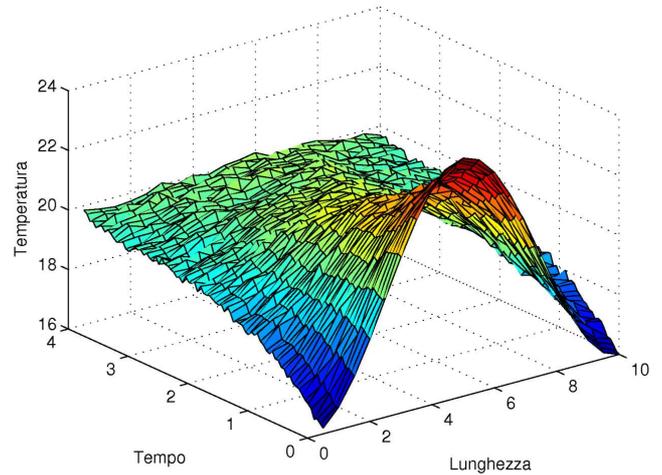


Fig. 5. Evoluzione della temperatura nella barra di conduzione

Il processo viene osservato tramite sensori che sono spazati in maniera uniforme lungo tutta la barra. Affinché sia possibile realizzare la distribuzione del modello è necessario supporre che ogni intervallo Δx nel quale è suddivisa la barra sia osservabile da almeno un sensore (si veda [6]). E' da notare che tali intervalli rappresentano gli stati x del

sistema. Quindi, nel caso generale, il numero di sensori è inferiore al numero di stati, pertanto si ha che ogni sensore osserva più di uno stato. In fase di implementazione, per semplicità, si è scelto di imporre che tutti i sensori osservino lo stesso numero di stati, tuttavia in generale è possibile considerare qualunque scelta. In tal caso è stato scelto di attribuire un peso maggiore, uguale di default a 0.5, alla misura eseguita sullo stato più vicino al luogo fisico dove viene piazzato il sensore; mentre dei pesi che decrescono progressivamente della metà, per l'osservazione di ciascuno stato in più tra quelli adiacenti verso destra e verso sinistra. Si noti che, per semplicità, la somma di tutti questi pesi è stata posta uguale a uno, in questo modo lo stato x e la misura y si esprimono entrambi direttamente in gradi centigradi, senza dover aggiungere ulteriori costanti moltiplicative. Un ulteriore vincolo che è stato imposto nella modellizzazione dei sensori è quello che ogni stato viene osservato al massimo da due sensori. Questa ipotesi non è strettamente necessaria al fine di implementare il metodo con stimatore di ordine ridotto, che è del tutto generale, risulta comunque logica e semplificativa nel caso in considerazione, permettendo così di sviluppare delle simulazioni ad hoc, senza doversi preoccupare di creare del software completamente generale. Si ricordi che per la comunicazione tra i vari sensori si è ipotizzato un grafo di tipo circolare indiretto. Ci si è quindi volutamente posti in una condizione in cui dinamica e comunicazione tra i sensori evolvono di pari passo. Per l'esempio scelto, che presenta un naturale disaccoppiamento dinamico tra stati lontani tra loro e proprio per come evolve la comunicazione, risulta particolarmente interessante confrontare, la possibilità di utilizzare un filtro di Kalman locale in forma di Informazione con consensus (Sezione III), con uno stimatore di ordine ridotto (Sezione IV).

Per ulteriori dettagli sulla implementazione si faccia riferimento al codice riportato in Appendice C

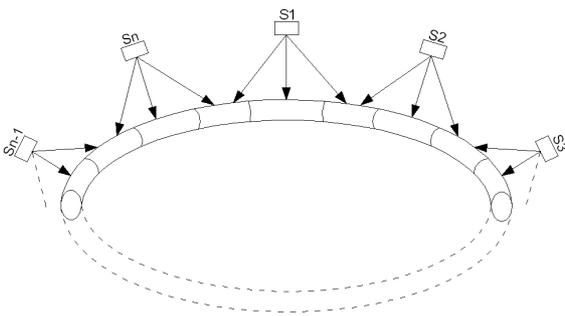


Fig. 6. Distribuzione dei sensori lungo la barra di conduzione

VII. CONFRONTO TRA LE DIVERSE IMPLEMENTAZIONI DEI FILTRI DI KALMAN IN FORMA DI INFORMAZIONE

In queste simulazioni, l'interesse è posto sul confronto dei vari metodi di stima presentati nelle sezioni precedenti, supponendo un ridotto numero di comunicazioni per intervallo di campionamento. Si inizia col presentare i risultati ottenuti

con i filtri di Kalman locali in forma di informazione che utilizzano la tecnica del consensus descritti in Sezione III. Nell'esempio utilizzato per le simulazioni, dove si fa riferimento ad un grafo di comunicazione tra i vari sensori di tipo circolare e indiretto, questo approccio non ha dato buoni risultati, come del resto ci si poteva attendere. Confrontando le stime ottenute con l'effettiva evoluzione dello stato, ci si accorge di una discrepanza che tende a rimanere costante. Per comprendere la natura di questo errore, si è implementato allora anche il filtro centralizzato in forma di informazione (riportato in Sezione II), e in particolare si è evidenziato come la stima $\hat{x}_{k|k}^c$, derivi dalla somma di due contributi: uno legato a $\hat{x}_{k|k-1}^c$ ed uno legato alla sommatoria dei termini $C^{iT} R^{i-1} y_k^i$, ognuno di questi premoltiplicato per un'opportuno "guadagno". Pensando di scomporre in questo senso anche la stima del filtro locale, dopo il consensus, ci si aspetta di trovare che dopo un sufficiente numero di iterazioni di consensus la stima

$$\hat{x}_{k|k}^i \rightarrow (I - K_{k|k} C) \frac{1}{N} \sum_{j=1}^N \hat{x}_{k|k-1}^j + N \cdot P_{k|k} \frac{1}{N} \sum_{j=1}^N C^{jT} R^{j-1} y_k^j. \quad (44)$$

Tenendo presente che si vuole ridurre al minimo il numero di comunicazioni, ricordando che i pesi della matrice di consensus Q per il grafo circolare seguono la logica di Metropolis, si giunge ad una stima ben diversa. Per questo semplice caso, è possibile calcolare facilmente gli autovalori della matrice di consensus Q (si veda la lezione 19 in [3]) tramite l'equazione:

$$\lambda_i = 1 + 2q \left[\cos \left(\frac{2\pi}{N} i \right) - 1 \right], \quad i = 0, \dots, N-1 \quad (45)$$

Come è noto, l'autovalore λ_1 , che è il maggiore in modulo, escludendo l'autovalore $\lambda_0 = 1$, dà un'indicazione riguardo alla velocità di convergenza del consensus. Nelle condizioni in cui ci si è posti questo risulta $\lambda_1 = 0,9107$. Appare dunque evidente che il problema principale nell'utilizzo di questo algoritmo è dovuto al fatto che per un limitato numero di comunicazioni, si ha che:

$$\sum_{j \in \mathcal{N}(i)} Q_{ij} C^{jT} R^{j-1} y_k^j \rightarrow \frac{1}{N} \sum_{j=1}^N C^{jT} R^{j-1} y_k^j \quad (46)$$

In fig.7, si può verificare il diverso contributo portato alla stima dai differenti vettori di informazione nei due casi. In secondo luogo, anche la somma pesata delle stime aggiornate, limitata dalle poche comunicazioni, non converge alla media di queste. Tutto questo si ripercuote sull'errore di stima, che viene compensato nel passo di aggiornamento, il quale sarà riferito ad una informazione differente da quella del caso centralizzato, portando così la stima ad avere

una sorta di “errore a regime”. La fig.7 fa riferimento al caso in cui sia effettuata una sola iterazione di consensus, esclusivamente per evidenziare il fenomeno, che continua a mantenere andamenti simili, anche se meno accentuati, fino a che non si considera un sufficiente numero di iterazioni di consensus per passo di campionamento.

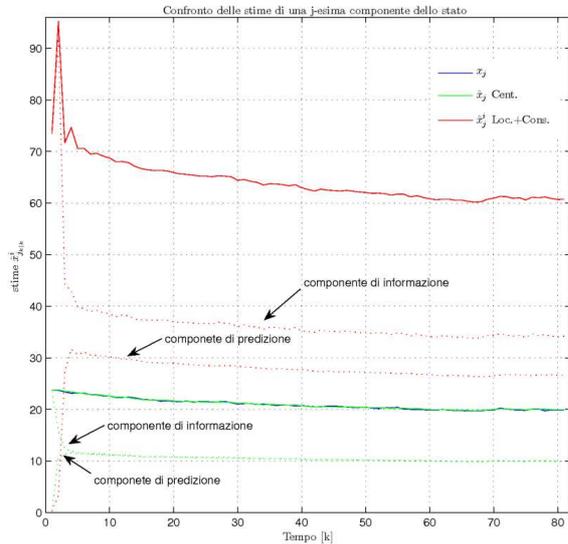


Fig. 7. Confronto tra filtro di Kalman centralizzato e filtri locali con consensus

Considerando di utilizzare le stesse condizioni iniziali $\hat{x}_{0| -1}^i = \bar{x}_0$, $i = 1 \dots N$ per ciascun filtro locale, può venire in mente di modificare l’algoritmo in modo che il consensus sia eseguito esclusivamente sulla parte relativa al vettore di informazione, per risolvere almeno il secondo problema. Questo però non migliora le cose, infatti per quanto visto in (46), si ottiene per la stima un comportamento analogo (fig.8).

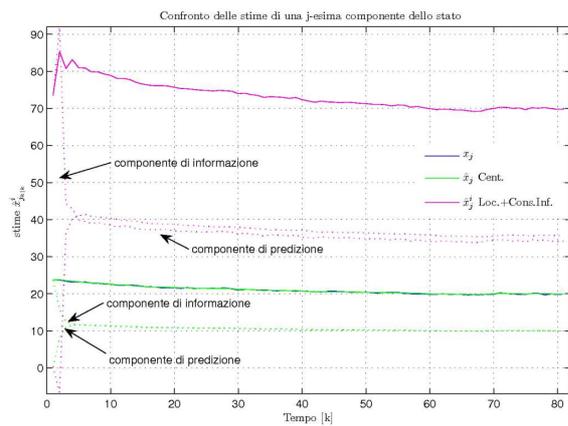


Fig. 8. Confronto tra filtro di Kalman centralizzato e filtri locali con consensus solo sul vettore di informazione

Si noti che volendo ottenere per questi metodi delle caratteristiche che bene approssimano quelle ottenibili con il filtro centralizzato è necessario aumentare notevolmente

il numero di iterazioni di consensus per ogni passo di aggiornamento. Nel caso in esame, ad esempio, in un sistema con dimensione dello stato $n = 20$, osservato da $N = 10$ sensori, ciascuno dei quali osserva 3 stati, il numero di iterazioni di consensus da eseguire ad ogni passo di aggiornamento risulta pari a 60. Questo dato, dà subito un’idea di quanto possa essere computazionalmente oneroso tale algoritmo, in particolare su sistemi di vaste dimensioni, sia a livello di mole di dati, sia a livello di capacità di calcolo richiesta ad ogni nodo. Tuttavia, va detto che questo metodo, a differenza di quelli che verranno successivamente discussi, è l’unico che permette di avere ad ogni sensore, una stima aggiornata dell’intero vettore di stato del sistema, equivalente a quella che si avrebbe nel caso centralizzato, sin dai primo istante di campionamento. Come si vede in fig.9, le caratteristiche del filtro centralizzato e quelle dei due filtri locali che utilizzano il consensus tendono a confondersi. Da questo ne deriva che le covarianze dell’errore di stima ricavate per via empirica si sovrappongono con quella ottima calcolata per via teorica (si veda la fig.10), mentre quella del filtro centralizzato corrisponde con quelle dei filtri locali.

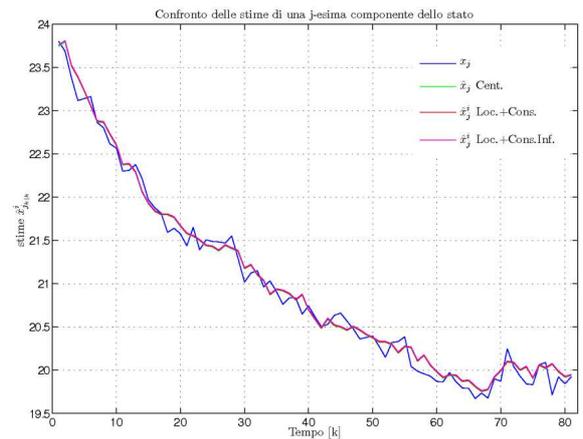


Fig. 9. Confronto tra filtro di Kalman centralizzato e le due tipologie di filtri locali con consensus

In considerazione del limitato numero di comunicazioni e della particolare struttura del grafo, un’alternativa più ragionevole è quella di implementare, per ogni nodo, un filtro di Kalman in forma di informazione, che utilizza ancora una volta un algoritmo di consensus, ma che tiene conto della limitata informazione che riceve rispetto al filtro centralizzato. Ovviamente, una scelta del genere porta a dover rinunciare ad ottenere sin dai primi passi di campionamento prestazioni paragonabili al caso centralizzato, come invece avveniva nel caso precedente. Seguendo questa logica, cioè sfruttare un limitato numero di comunicazioni, per stimare al meglio almeno le componenti dello stato direttamente osservate dal sensore, il passo di aggiornamento della stima locale si può esprimere come:

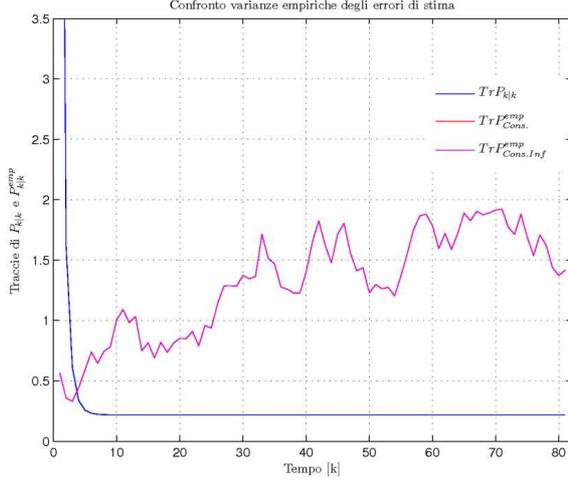


Fig. 10. Confronto prestazioni tra filtro di Kalman centralizzato e le due tipologie di filtri locali con consensus

$$\hat{x}_{k|k}^i = (I - N \cdot K_{k|k}^i C^i) \hat{x}_{k|k-1}^i + N \cdot P_{k|k} C^i R^{i-1} y_k^i \quad (47)$$

Dove N in questo caso, non è più l'intero numero dei sensori che osservano il sistema, ma dei soli sensori che comunicano tra loro in un passo (in questo caso $N = 3$). K^i è invece la i -esima colonna del guadagno di Kalman del filtro centralizzato. L'utilizzo della sola colonna i appare ragionevole, nella logica di considerare l'informazione al sensore i come l'unica disponibile. Si noti che in questo caso, ad essere moltiplicato per N è l'intero errore di stima locale $e^i = y^i - C^i \hat{x}_{k|k-1}^i$, con C^i i -esima riga della matrice C . Ogni filtro locale cerca quindi di compensare al meglio il proprio errore di stima. Dopo l'iterazione di consensus, che coinvolge esclusivamente il nodo precedente e quello successivo, si ottiene che la stima aggiornata risulta:

$$\hat{x}_{k|k}^i = \frac{1}{N} \sum_{j=i-1}^{i+1} \hat{x}_{k|k-1}^j + N \cdot \frac{1}{N} \sum_{j=i-1}^{i+1} K_{k|k}^j (y_k^j - C^j \hat{x}_{k|k-1}^j) \quad (48)$$

Rimanendo nella logica di ottimizzare ciascuno stimatore rispetto alla sola informazione y^i che riceve, si può pensare di usare il guadagno ottimo K_i (del filtro che considera un unico sensore i) al posto della colonna K^i di K . Il confronto tra i due approcci e le covarianze degli errori di stima ricavate per via empirica sono riportate nelle fig.11 e 12. Si nota immediatamente, dalla fig.11, che nonostante le forti oscillazioni iniziali (eliminabili considerando da subito un guadagno K_i tempo-invariante, equivalente a quello a regime), le stime della componente dello stato direttamente osservata dal sensore in esame, operate dai filtri locali, convergono ad una approssimazione della variabile da

stimare. Ovviamente, in questo caso, la convergenza a delle stime accurate, quanto nel caso precedente, richiederà un certo numero di passi. In particolare, ad un certo sensore i , ad essere poco accurate, all'inizio, saranno tutte quelle stime che riguardano le componenti dello stato non osservate direttamente da i , tanto più se queste sono anche lontane e dinamicamente disaccoppiate da quelle viste da i . Appare evidente dai grafici, come le prestazioni valutate per via empirica, su $M = 20$ esperimenti indipendenti, a regime risultino dello stesso ordine di quelle dei filtri locali con consensus. La velocità, con cui la covarianza dell'errore di stima, converge a questi valori, è sensatamente paragonabile al numero di iterazioni di consensus necessarie nel caso precedente, ed ovviamente non poteva essere altrimenti. La cosa interessante è che a differenza di quanto avveniva in precedenza, dove anche a regime si effettuava un elevato numero di comunicazioni per intervallo di campionamento, in questo caso è sufficiente una sola comunicazione per mantenere uguali prestazioni. I motivi di questo risultato vanno probabilmente ricercati nelle forti ipotesi di somiglianza tra l'evoluzione della dinamica e della comunicazione in cui ci si è posti. Tuttavia, i risultati ottenuti appaiono interessanti nell'ottica di una possibile generalizzazione, per sistemi di larga scala in cui la dinamica continua a poter essere disaccoppiata in sottosistemi, ed è possibile solo un limitato numero di comunicazioni con i sottosistemi vicini. Proprio in considerazione di questo, è possibile osservare, aldilà della dimensione dello stato stimato localmente, delle somiglianze tra quest'ultimo metodo a ridotte comunicazioni e quello discusso in Sezione IV, argomento che verrà ripreso in seguito. In fine, è interessante notare che, questi due ultimi approcci risultano "solidi" (nel senso che è comunque garantita la convergenza), a differenza di quelli precedenti, sia con una sola comunicazione e $N = 3$, sia con un'elevato numero di comunicazioni, ma utilizzando N uguale al numero di sensori.

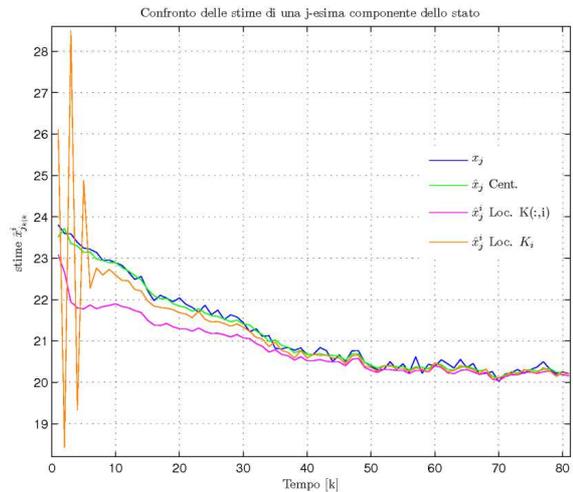


Fig. 11. Confronto tra filtro di Kalman centralizzato e le due tipologie di filtri locali a ridotte comunicazioni

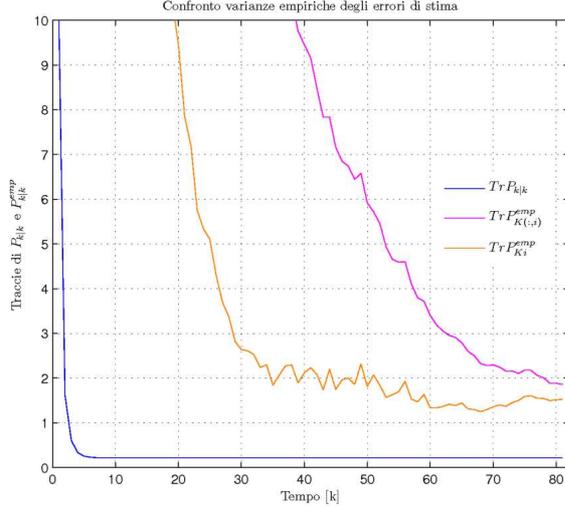


Fig. 12. Confronto prestazioni tra filtro di Kalman centralizzato e le due tipologie di filtri locali a ridotte comunicazioni

VIII. CONSIDERAZIONI SULLA STIMA DISTRIBUITA TRAMITE DECOMPOSIZIONE DEL MODELLO

Si noti, che a livello di implementazione si sono riscontrate una serie di difficoltà legate alla non semplice realizzazione di questo metodo, in modo automatizzato, su un modello del tutto generale. Tralasciando quindi l'ottimizzazione e la scrittura di codice totalmente generale si è preferito concentrarsi sul caso particolare in considerazione. In particolare, per quanto riguarda la disposizione dei sensori, ci si è posti nelle condizioni descritte dalle ipotesi riportate in VI, che hanno permesso uno sviluppo molto più semplice del codice necessario alle simulazioni. A livello implementativo si sono rilevate significative discrepanze tra quanto esposto nella Sezione V-D e quanto si ritiene essere una più corretta interpretazione del metodo di stima sui modelli ridotti. In particolare, c'è una differenza sostanziale che riguarda il passo di aggiornamento, che si ripercuote poi anche in quello di predizione. Scrivendo l'aggiornamento nella variabile di stato locale $x^{(l)}$, come un filtro di Kalman in forma di informazione che si basa esclusivamente sulla misura $y^{(l)}$, ed utilizzando espressioni in funzione della sola varianza $P_{k|k-1}^{(l)}$, si ha che:

$$\hat{x}_{k|k}^{(l)} = \left(P_{k|k-1}^{(l)-1} + C^{(l)T} R^{(l)-1} C^{(l)} \right)^{-1} P_{k|k-1}^{(l)-1} \hat{x}_{k|k-1}^{(l)} + \left(P_{k|k-1}^{(l)-1} + C^{(l)T} R^{(l)-1} C^{(l)} \right)^{-1} C^{(l)T} R^{(l)-1} y_k^{(l)} \quad (49)$$

dove $P_{k|k-1}^{(l)-1}$ è l'inversa della matrice locale di covarianza dell'errore $P_{k|k-1}^{(l)}$ calcolata in (20), che non è equivalente a $Z_{k|k-1}^{(l)}$ che si può estrarre da $Z_{k|k-1} = P_{k|k-1}^{-1}$, come si è fatto in una prima fase. Infatti, per trascurare il problema dell'inversione distribuita di una matrice si sono calcolate fuori linea tutte le matrici di covarianza P e le

inverse Z , dalle quali si sono, volta per volta, opportunamente estratte le sottomatrici necessarie. Analogamente, $\tilde{P}_{k|k}^{(l)} = \left(P_{k|k-1}^{(l)-1} + C^{(l)T} R^{(l)-1} C^{(l)} \right)^{-1} \neq P_{k|k}^{(l)}$ che si può estrarre da $P_{k|k}$. È fondamentale notare come ciascun vettore di informazione locale $i_k^{(l)} = C^{(l)T} R^{(l)-1} y_k^{(l)}$ sia moltiplicato per un differente fattore $\tilde{P}_{k|k}^{(l)}$. Dividendo ambo i membri per $\tilde{P}_{k|k}^{(l)}$, per ritornare ad una scrittura nella variabile z , si ottiene, nel caso di uno stimatore che considera la sola informazione proveniente dal sensore l :

$$\hat{z}_{k|k}^{(l)} = \hat{z}_{k|k-1}^{(l)} + i_k^{(l)} \quad (50)$$

Pensando di tornare ora ad una stima che utilizza anche l'informazione proveniente dai vicini, per quanto visto nei passaggi precedenti, si intuisce come non sia formalmente corretto scrivere l'equazione di aggiornamento nel modo in cui è presentata in (35). Infatti dalla (49) si capisce che la fusione sui vettori di informazione non va effettuata direttamente sugli $i_k^{(l)}$, bensì sul risultato dei prodotti $\tilde{P}_{k|k}^{(l)} \cdot i_k^{(l)}$. Ciò che si è effettivamente implementato per il passo di aggiornamento è quindi la (50), dalla quale si passa a $\hat{x}_{k|k}^{(l)} = \tilde{P}_{k|k}^{(l)} \hat{z}_{k|k}^{(l)}$. Da quest'ultima, attraverso un'algoritmo di consensus con i vicini, che riguarda le sole variabili dello stato ridotto che sono in comune, si ottiene l'effettiva stima $\hat{x}_{k|k}^{(l)}$. Infine, analogamente a quanto detto in precedenza, si noti che anche nel passo di predizione (40), al momento della conversione nella variabile in z , si utilizza $P_{k+1|k}^{(l)-1}$ con $P_{k+1|k}^{(l)}$ calcolata in (20) e non $Z_{k+1|k}^{(l)}$ estratta da $Z_{k+1|k} = P_{k+1|k}^{-1}$. Nonostante la probabile banalità di tali note, si è ritenuto necessario darne traccia, in corpo alla relazione, per chiarire sin da subito come si è operato. Queste precisazioni riguardo all'implementazione, risultano utili oltre che per evidenziare un modo corretto per utilizzare questo algoritmo, anche per notare una evidente similitudine tra questo approccio e quanto proposto in (48), di cui questo può in qualche modo essere considerato come un caso particolare. Si passa ora all'analisi delle prestazioni di questo metodo ed al confronto, oltre che con il filtro ottimo centralizzato, anche con il filtro distribuito in forma di informazione che effettua il consensus solo con i vicini ed utilizza un guadagno K_i . Prima di procedere è opportuno precisare che, in considerazione del fatto che gli stimatori basati sulla distribuzione del modello stimano solo un numero ridotto di variabili, ciò che ha senso confrontare è la prestazione che riguarda esclusivamente un ridotto set di componenti dello stato. Per una prima analisi, si è considerata la stima, ricavata al nodo l , della componente dello stato osservata con maggiore peso da questo e la si è confrontata con quella ottenuta dal filtro ottimo centralizzato. Dalla fig.13 si può apprezzare come vi sia un'effettiva convergenza della stima basata sulla distribuzione del modello a quella del filtro centralizzato. Si noti che questo è vero per tutte le componenti osservate

al sensore l . Ovviamente la motivazione sta nel fatto che data la particolare struttura del sistema globale, per la stima a livello locale, la conoscenza di componenti dello stato che dinamicamente non influiscono in un solo passo su quelle locali non ha rilevanza. Per quanto riguarda un confronto in termini di prestazioni, ovvero varianze degli errori di stima, occorre prima decidere cosa si vuol intendere. Nel nostro caso, si è scelto di definire come errore di stima sul sistema globale, il vettore colonna e_k (di dimensioni $n_l \times 1$), di componenti $e_l = x_j - \hat{x}_i^{(l)}$, che raccoglie quindi ad ogni riga l la differenza tra l'effettiva componente j dello stato e la sua stima che si trova alla componente i del vettore di stima locale al sensore l . Usando coerenti definizioni per gli errori di stima nel caso del filtro centralizzato e filtro distribuito a ridotte comunicazioni, si è ricavato per via empirica il confronto tra le varianze degli errori di stima. Tale confronto riportato in fig.14 evidenzia come le prestazioni del filtro basato sulla distribuzione del modello (TrP_{Dist}^{emp}), siano assolutamente confrontabili con quelle del filtro centralizzato (TrP_{Cent}^{emp}) e del filtro distribuito con ridotte comunicazioni (TrP_{Kl}^{emp}), convergendo più velocemente di quest'ultimo alle prestazioni del caso centralizzato.

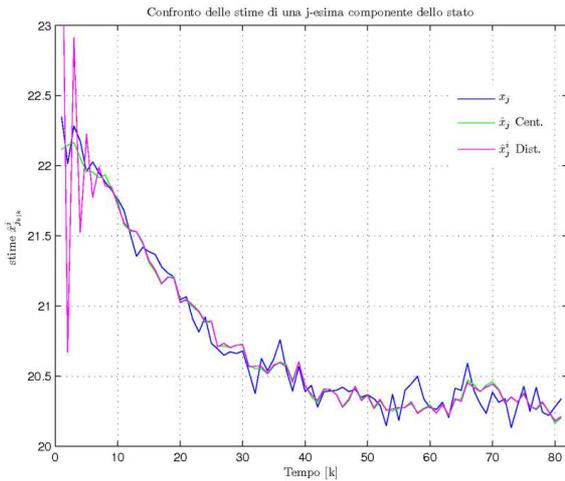


Fig. 13. Confronto tra filtro di Kalman centralizzato e filtro distribuito sul modello ridotto

IX. CONCLUSIONI E SVILUPPI FUTURI

In conclusione si riassumono i risultati e gli aspetti fondamentali di quanto visto finora. Si è mostrato come in particolari casi, sia possibile applicare tecniche di stima distribuita, sia dell'intero vettore di stato, sia di una sua porzione riferita a un sottosistema di ordine ridotto, utilizzando un numero ridotto di comunicazioni ed ottenendo a regime prestazioni assolutamente confrontabili con quelle del caso centralizzato. Il prezzo da pagare, come ovvio, consiste nel dover attendere un transitorio iniziale prima di raggiungere le caratteristiche del filtro centralizzato o dei filtri di Kalman distribuiti che utilizzano il consensus con un alto numero di comunicazioni, che oltretutto non sono

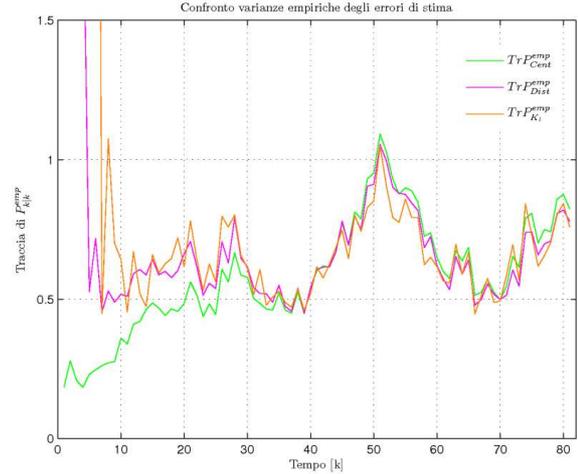


Fig. 14. Confronto prestazioni tra filtro di Kalman centralizzato e filtro distribuito sul modello ridotto

influenzati dalla particolare struttura del sistema, a differenza dei primi. Per avere fin da subito le stesse prestazioni del filtro centralizzato, nel caso in cui si possa trascurare il transitorio iniziale, il sistema abbia una dinamica lenta e non ci siano ingressi che possono far cambiare repentinamente qualche componente dello stato del sistema, si può pensare di ricorrere a filtri locali sullo stato globale che effettuano la sola comunicazione con i vicini. Attraverso di essi è possibile infatti avere, a ogni nodo e in ogni istante di campionamento, l'informazione sull'intero stato del sistema, utilizzando il minimo numero di comunicazioni. Ovviamente questa visione presenta dei forti limiti nell'inseguimento della dinamica. Per seguire sin da subito la caratteristica del filtro centralizzato si potrebbe pensare di eseguire al passo iniziale un elevato numero di comunicazioni, che verrà ridotto successivamente, in quanto non più necessarie per seguire l'evoluzione, una volta raggiunto il regime. Ci si pone nelle condizioni di utilizzare un approccio basato sulla distribuzione del modello, ma avendo comunque la volontà di avere, a ogni istante e a ogni nodo, l'informazione sullo stato globale, senza però l'onere computazionale richiesto dai filtri locali che utilizzano il consensus. Si può pensare che nello stesso tempo in cui vengono effettuate le numerose iterazioni di consensus su vettori di stato di grandi dimensioni, le informazioni relative alle stime locali di ciascun nodo riescano ad arrivare a tutti gli altri nodi della rete. In questo modo, utilizzando un minor numero di comunicazioni e con un minor peso computazionale a livello locale, si potrebbe ottenere il duplice vantaggio di avere a disposizione ad ogni nodo la stima dell'intero stato del sistema (che potrebbe essere utile ad esempio per scopi di controllo ottimo) e la capacità di seguire la dinamica, al pari del filtro centralizzato, anche in presenza di ingressi con caratteristiche di tipo "a gradino" su qualche componente dello stato. A questo punto diventerebbe interessante provare a caratterizzare le prestazioni di questi metodi in funzione

della particolare scelta effettuata per il monitoraggio del sistema, oltre che della struttura di esso, sulla quale appare necessario tuttavia porre delle forti condizioni.

APPENDICE A
DIMOSTRAZIONE DEL LEMMA DI INVERSIONE DI MATRICE

Lemma 1 (di Inversione di Matrice): Siano A e C matrici quadrate invertibili. Si ha:
 $[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}$

Dimostrazione 1: Si tratta di una verifica:

$$\begin{aligned} & (A + BCD)(A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}) = \\ & = I + BCDA^{-1} - (B + BCDA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}DA^{-1} = \\ & = I + BCDA^{-1} - BC(C^{-1} + DA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}DA^{-1} = I \end{aligned} \quad (51)$$

□

APPENDICE B
ALGORITMO DIC1

Si tralasciano momentaneamente, per maggior chiarezza, le indicazioni temporali; per ulteriori approfondimenti si rimanda a [7].

B.1 Centralized Jacobi Overrelaxation Algorithm (JOR)

L'algoritmo JOR centralizzato risolve problemi matriciali del tipo:

$$ZP = T \quad (52)$$

dove la matrice P è incognita, mentre T e Z sono note. Definita $M = \text{diag}(Z)$, si ha che per alcuni $\gamma > 0$

$$P_{t+1} = ((1 - \gamma)I_{n \times n} + \gamma M^{-1}(M - Z)) P_t + \gamma M^{-1}T \quad (53)$$

converge a P , soluzione della (52), dove γ è chiamato parametro di rilassamento. Ponendo $T = I_{n \times n}$ si trova $ZP = I_{n \times n} \Rightarrow P = Z^{-1}$, e se Z è nota, l'iterazione convergerà a Z^{-1} .

$$P_{t+1} = S_\gamma P_t + \gamma M^{-1} \quad (54)$$

dove S_γ , detta matrice moltiplicatrice, è definita come:

$$S_\gamma = (1 - \gamma)I_{n \times n} + \gamma M^{-1}(M - Z) \quad (55)$$

B.2 Distributed JOR Algorithm

Per la determinazione delle covarianze d'errore locali, è sufficiente determinare la fascia che comprende le L diagonali superiori e inferiori di $P = Z^{-1}$. Distribuendo l'algoritmo *JOR*, per determinare gli elementi p_{ij} appartenenti alla fascia d'interesse di P_{t+1} , si utilizzano le seguenti relazioni:

$$\begin{aligned} p_{ij,t+1} &= s_i p_t^j & i \neq j \\ p_{ij,t+1} &= s_i p_t^j + m_{ii}^{-1} & i = j \end{aligned} \quad (56)$$

Dove il vettore riga s_i è la i -esima riga della matrice S_γ ; il vettore colonna p_t^j è la j -esima colonna di P_t , e m_{ii} è l' i -esimo elemento sulla diagonale di M . Data una Z che sia L -banded, anche S_γ sarà L -banded. La i -esima riga della matrice moltiplicatrice S_γ conterrà elementi non nulli al più in $2L + 1$ locazioni di indice $k = \{i - L, \dots, i, \dots, i + L\}$. Questi elementi si combineranno con quelli corrispondenti della j -esima colonna p_t^j di P_t . Grazie alla proprietà della matrice S_γ di essere L -banded, questo algoritmo può essere facilmente distribuito con opportune comunicazioni tra sensori vicini. Ciò nonostante, si può dimostrare che una singola iterazione di questo algoritmo impone la scansione di un'intera riga di P , richiedendo la conoscenza di elementi al di fuori della L -band, risultando quindi di complessità di ordine n .

B.3 Distributed Iterate Collapse Overrelaxation Algorithm (DIC1-OR)

L'algoritmo *DIC1-OR* si suddivide in due passaggi: uno iterativo ed uno di crollo.

Il passo iterativo, per gli elementi p_{ij} che appartengono alla fascia delle L sopra e sottodiagonali (tali che $|i - j| \leq L$), si può scrivere come:

$$\begin{aligned}
p_{ij,t+1} &= s_i p_t^j & i \neq j, |i-j| \leq L \\
p_{ij,t+1} &= s_i p_t^j + m_{ii}^{-1} & i = j, |i-j| \leq L
\end{aligned}
\tag{57}$$

con significato dei simboli precedentemente definito.

L'implementazione del passo iterativo (57), come si è visto in precedenza, richiederebbe anche elementi non appartenenti alla L -band. Per ovviare a questo problema si introduce il passaggio di crollo. Si assume che P_t sia l'inversa di una matrice L -banded, e si utilizza il seguente risultato contenuto in [10]. Si dimostra infatti, che se P è l'inversa di una matrice L -banded, allora le sottomatrici che non appartengono alla L -band di P , possono essere calcolate a partire da quelle della L -band di P . In questo modo per calcolare l'inversa $P = Z^{-1}$ della matrice L -banded Z è sufficiente calcolare le sottomatrici appartenenti alla L -band della sua inversa P . Quando $L = 1$, un elemento non appartenente alla L -band, (p_{ij} tali che $|i-j| > L$), è ricavabile a partire da quelli nella L -band (p_{ij} tali che $|i-j| \leq L$) come:

$$p_{ij} = p_{i,j-1} p_{i+1,j-1}^{-1} p_{i+1,j} \quad |i-j| > L \tag{58}$$

In questo modo, si evita di eseguire il passo iterativo per elementi che non appartengono alla L -band. Le condizioni iniziali per l'algoritmo *DICI-OR* sono date da:

$$\begin{aligned}
S_\gamma^{(l)} &= (1-\gamma)I_{n \times n}^{(l)} + \gamma (M^{(l)})^{-1} (M^{(l)} - Z^{(l)}) \\
P_0^{(l)} &= (Z^{(l)})^{-1}
\end{aligned}
\tag{59}$$

Si noti che le (59) non richiedono alcuna comunicazione e possono essere calcolate direttamente ad ogni sensore partendo dalla matrice locale di osservazione $Z^{(l)}$. Questo perché la matrice M è diagonale e la sottomatrice $M^{(l)}$ è l'esatta inversa degli elementi che formano la diagonale di $Z^{(l)}$. L'algoritmo *DICI* (senza il parametro di rilassamento γ), può essere ottenuto a partire dal *DICI-OR* ponendo $\gamma = 1$. L'onere computazionale per l'algoritmo *DICI* è indipendente da n . Per ulteriori approfondimenti e la generalizzazione al caso $L > 1$ si rimanda a [9] e [10].

APPENDICE C CODICE MATLAB

Si riportano i listati MATLAB delle principali funzioni implementate:

```

%% Funzione per la stima distribuita dello stato basata sulla scomposizione
% del modello ridotto
function [x_l_kk,P_l_kk,i_flk]=StimDistrFurboMod2A
(A,A_l,x_l,D_l,d_l,C_l,Rl_inv,I_fl,sens,s_oss,Q,Q_l,P0,X0,I,P0_l,y)
% INPUT:
% A
% Q
% Matrici e vett. dei modelli ridotti
% A_l
% x_l
% D_l
% d_l
% C_l
% Rl_inv
% Q_l
% I_fl
% I
% sens= vettore che raccoglie le componenti centrali su cui e centrato
% ciascun sensore
% s_oss=numero di stati visti per sensore
% X0 = x_0|-1,
% P0 = P_0|-1 condizioni iniziali
% P0_l
% Y

% OUTPUT:
% x_l_kk = x(:,k,l) dimensioni: s_oss*T*n_l, stima di x cioè x^k|k
% P_l_k|k Varianza dell'errore di stima locale
% i_flk vettore fusione delle osservazioni

```

```

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FUSIONE INFORMAZIONI
ns= length(sens); % numero di sensori
T = size(y,2);    % numero di istanti temporali

for(l=1:ns)
    % scandisco ciascun sensore
    i_lk(:, :, l) = C_l(:, :, l)'*Rl_inv(l)*y(l, :); % vettore informazione locale
    % i_l,k istante k=1.....T
    % dim: s_oss*T*ns
I_l(:, :, l) = C_l(:, :, l)'*Rl_inv(l)*C_l(:, :, l); % I_l covarianza informazione
    % dim s_oss*s_oss*ns
end
% FUSIONE dei vettori di informazione i_lk nelle componenti
% che rappresentano la stessa componente del vettore di stato.
% Le matrici di informazione I_fl sono a gratis,
% arrivano dalla distribuzione del modello, perchè il modello è statico.

for(l=1:ns)
    % scandisco ciascun sensore
% per come configurato il sistema possono avere che una componente di stato
% sia oss. da l lo sia anche da un'altro sensore, solo se vicino
% ovvero attiguo l+1 e l-1
    lp= l+1;
    if(lp>ns)
        lp=lp-ns;
    end
    lm= l-1;
    if(lm<0)
        lm=ns+lm;
    end
    i_flk(:, :, l)=i_lk(:, :, l); % inicializzo la fusione al valore
    % del vettore d'informazione del sensore
    for(jj=1:s_oss)
        % scandisco gli stati oss dal sens
        for(hh=1:s_oss)
            % scandisco gli stati oss dai sens vicini
            if(x_l(:, jj, l)==x_l(:, hh, lp)) % se vedono la stessa componete
                %aggiorno la fusione.
                i_flk(jj, :, l)=i_flk(jj, :, l)+i_lk(hh, :, lp);
            end
            if((x_l(:, jj, l))==x_l(:, hh, lm))
                i_flk(jj, :, l)=i_flk(jj, :, l)+i_lk(hh, :, lm);
            end
        end
    end
end
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INIZIALIZZAZIONE:
%%FURBO%%
Z_ks(:, :, 1)=P0^(-1);
P_ks(:, :, 1)=P0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for(l=1:ns)
    X0_l(:, l)=X0(x_l(:, :, l)); % condizioni iniz. stimatori locali x^_l_0|-1
    % scrivo l'inversione in modo diretto solo perchè le P0(:, :, l) sono
    % matrici diagonali (idnetità)
    Z0_l(:, :, l) = P0_l(:, :, l)^(-1); % Z_l_0|-1=(P_l_0|-1)^-1
    z0_l(:, l)=Z0_l(:, :, l)*X0_l(:, l); % z_l_0|-1=(Z_l_0|-1)*x_l_0|-1
    Z_l_ks(:, :, l, l)=Z0_l(:, :, l); % Z_l_k|k-1(1)
    z_l_ks(:, l, l)=z0_l(:, l); % z_l_k|k-1(1)
    x_l_ks(:, l, l)=X0_l(:, l); % x_l_k|k-1(1)
    P_l_ks(:, :, l, l)=P0_l(:, :, l); % P_l_k|k-1(1)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% EVOLUZIONE Delle matrici di Covarianza
%%% FURBO
% Aggiornamento
Avanz=10; % iterazioni preliminari per la varianza
F=T+Avanz; % Una decina di cicli alla varianza prima di
% inizializzarne il valore iniziale
for(k=1:F-1) % per l'indice k = 1, ....., T-1
    % corrispondente ai tempi t = 0, ....., T-2
    Z_kk(:, :, k)=Z_ks(:, :, k)+I;
    P_kk(:, :, k)=Z_kk(:, :, k)^(-1);
end

```

```

% Predizione
P_ks(:, :, k+1)=A*P_kk(:, :, k)*A'+Q;
Z_ks(:, :, k+1)=(P_ks(:, :, k+1))^( -1);
end
% ultimo passo di aggiornamento
Z_kk(:, :, F)=Z_ks(:, :, F)+I;
P_kk(:, :, F)=Z_kk(:, :, F)^( -1);

% RIPRISTINO INDICI CORRETTI
Z_kk=Z_kk(:, :, Avanz+1:F);
Z_ks=Z_ks(:, :, Avanz+1:F);
P_kk=P_kk(:, :, Avanz+1:F);
P_ks=P_ks(:, :, Avanz+1:F);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EVOLUZIONE Stima dello stato dei sistemi ridotti
for(k=1:T-1) % per l'indice k = 1, ..., T-1
    % corrispondente ai tempi t = 0, ..., T-2

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AGGIORNAMENTO:
% 1)
    for(l=1:ns) % ad ogni istante temporale scandisco ciascun sensore

        Z_l_kk(:, :, k, l)=Z_l_ks(:, :, k, l)+I_fl(:, :, l); % Z_l_k|k
        z_l_kk(:, k, l)=z_l_ks(:, k, l)+i_lk(:, k, l); % z^_l_k|k

%INVERSIONE di Z_l_kk(:, :, k, l)=====>P_l_kk(:, :, k, l)
% uso P_l_kk(:, :, k, l) ricavata solo da P_l_ks(:, :, k, l) e matrici locali
%RIPORTO in X
% x_l_kk(:, k, l)=(inversa(Z_l_kk(:, :, k, l)))*z_l_kk(:, k, l);
% x_l_kk(:, k, l)=
(C_l(:, :, l)'*Rl_inv(l)*C_l(:, :, l)+P_l_ks(:, :, k, l)^( -1))^( -1)*z_l_kk(:, k, l);
% utilizzando variabili del modello ridotto
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2) CONSENSUS TRA LE STIME AGGIORNATE IN X
for(l=1:ns) % scandisco ciascun sensore
% per come configurato il sistema possono avere che una componente di stato
% che sia oss. da l lo sia anche da un'altro sensore, solo se vicino
% ovvero attiguo l+1 e l-1
    lp= l+1;
    if(lp>ns)
        lp=lp-ns;
    end
    lm= l-1;
    if(lm<0)
        lm=ns+lm;
    end
    for(jj=1:s_oss) % scandisco gli stati oss dal sens
        x_l_kkp(jj, k, l)=x_l_kk(jj, k, l);
        for(hh=1:s_oss) % scandisco gli stati oss dai sens vicini
            if(x_l(:, jj, l)==x_l(:, hh, lp)) % se vedono la stessa componente
                % aggiorno la fusione.
                x_l_kkp(jj, k, l)=x_l_kkp(jj, k, l)*0.5+x_l_kk(hh, k, lp)*0.5;
            end
            if((x_l(:, jj, l))==x_l(:, hh, lm))
                x_l_kkp(jj, k, l)=x_l_kkp(jj, k, l)*0.5+x_l_kk(hh, k, lm)*0.5;
            end
        end
    end
end
x_l_kk(:, k, :)=x_l_kkp(:, k, :);% ridefinisco le stime con quelle dopo il
% consensus

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PREDIZIONE:
% 1)
    for(l=1:ns) % ad ogni istante temporale scandisco ciascun sensore
% DETERMINAZIONE delle stime di d_l_kk(:, k, l) dei sensori vicini
% per come configurato il sistema possono avere componenti di
% stato di interesse come ingresso solo nei sensori vicini

```

```

% ovvero attigui l+1 e l-1
lp= l+1;
if(lp>ns)
    lp=lp-ns;
end
lm= l-1;
if(lm<0)
    lm=ns+lm;           % OK con le ipotesi che:
                        % ogni componente dello
                        % stato è vista al max da 2 sensori,
                        % e le componenti in ingresso come d_l
                        % viste dal sensore prossimo
end
d_l_kk(:,k,l)=zeros(length(d_l(:, :, l)),1);
for(jj=1: length(d_l(:, :, l)))
    % scandisco le componenti di d_l(:, :, l)
    for(hh=1:s_oss)
        % scandisco gli stati oss dai sens vicini
        if(d_l(:, jj, l)==x_l(:, hh, lp))
            % se questi stimano la componente di interesse
            d_l_kk(jj,k,l)=x_l_kk(hh,k,lp); % la inserisco nel
            % vettore stime degli ingressi
        end
        if((d_l(:, jj, l))==x_l(:, hh, lm))
            d_l_kk(jj,k,l)=x_l_kk(hh,k,lm);
        end
    end
end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% % 2) Passo di PREDIZIONE di P_l_k+1|k
for(l=1:ns) % ad ogni istante temporale scandisco ciascun sensore
%% % Calcolo P_l_k+1|k(in funzione delle P_l_k|k dei vari sensori)
%% % Devo determinare P_dl_kk e P_xdl_kk covarianze di dl e di xl con dl
%% % all'istante k|k
    % NO: è un casino ricavare P_dl_kk e P_xdl_kk
    if(D_l==0)
        P_dl_kk(:, :, k, l)=0;
    else
        P_dl_kk(:, :, k, l)=zeros(size(d_l(:, :, l), 2), size(d_l(:, :, l), 2));
        for(jj=1:size(d_l(:, :, l), 2))
            for(hh=1:size(d_l(:, :, l), 2))
                P_dl_kk(jj, hh, k, l)= P_kk(d_l(:, jj, l), d_l(:, hh, l), k);
            end
        end
    end
end

if(D_l==0)
    P_xdl_kk(:, :, k, l)=0;
else
    P_xdl_kk(:, :, k, l)=zeros(size(x_l(:, :, l), 2), size(d_l(:, :, l), 2));
    for(jj=1:size(x_l(:, :, l), 2))
        for(hh=1:size(d_l(:, :, l), 2))
            P_xdl_kk(jj, hh, k, l)= P_kk(x_l(:, jj, l), d_l(:, hh, l), k);
        end
    end
end

% estraggo la P_l_kk estraendola da quella precalcolata
[P_l_kk(:, :, k, l)]=Dist_PoZ(P_kk(:, :, k), sens(l), s_oss);
% P_l_k+1|k
P_l_ks(:, :, k+1, l)=A_l(:, :, l)*P_l_kk(:, :, k, l)*A_l(:, :, l)'+
+(A_l(:, :, l)*P_xdl_kk(:, :, k, l)*D_l(:, :, l)')+
+(A_l(:, :, l)*P_xdl_kk(:, :, k, l)*D_l(:, :, l)')+
+D_l(:, :, l)*P_dl_kk(:, :, k, l)*D_l(:, :, l)'+ Q_l(:, :, l);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% % 3) PASSO DI PREDIZIONE di Z_l_k+1|k x_l_k+1|k e z_l_k+1|k
for(l=1:ns) % ad ogni istante temporale scandisco ciascun sensore
%% % INVERSIONE di P_l_k+1|k trovando Z_l_k+1|k
%% % inversa(P_l_ks(:, :, k+1, l)+comunicazioni locali-->Z_l_ks(:, :, k+1, l)

```

```

% Calcolo inversa locale estraendola dalla precalcolata
% Z_l_ks(:, :, k+1, l) = (inversa di (P_l_k|k-1));
[Z_l_ks(:, :, k+1, l)] = Dist_PoZ(Z_ks(:, :, k+1), sens(l), s_oss);

% Recuperate delle stime di d_l_kk(:, k, l) dai sensori vicini
% calcolo di x^_l_k+1|k e z^_l_k+1|k

x_l_ks(:, k+1, l) = A_l(:, :, l) * x_l_kk(:, k, l) + D_l(:, :, l) * d_l_kk(:, k, l);

% calcolo di z^_l_k+1|k
% z_l_ks(:, k+1, l) = Z_l_ks(:, :, k+1, l) * (A_l(:, :, l) * x_l_kk(:, k, l) +
% D_l(:, :, l) * d_l_kk(:, k, l));
% xò al posto di Z_l_ks(:, :, k+1, l) estratta da Z_k+1|k = (P_k+1|k)^-1
% si usa per Z_l_k+1|k l'espressione (P_l_k+1|k)^-1 quindi
% ricavandola dall'inversione della sola matrice locale
z_l_ks(:, k+1, l) = P_l_ks(:, :, k+1, l)^(-1) * (A_l(:, :, l) * x_l_kk(:, k, l) +
+ D_l(:, :, l) * d_l_kk(:, k, l));
end
end % Fine ciclo for dei tempi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% AGGIORNAMENTO FINALE elementi T-esimi:
% 1)
for(l=1:ns)
    Z_l_kk(:, :, T, l) = Z_l_ks(:, :, T-1, l) + I_l(:, :, l); % Z_l_k|k
    z_l_kk(:, T, l) = z_l_ks(:, T, l) + i_lk(:, T, l); % z^_l_k|k

% INVERSIONE di Z_l_kk(:, :, k, l) ==> P_l_kk(:, :, k, l)
% uso P_l_kk(:, :, k, l) ricavata solo da P_l_ks(:, :, k, l) e matrici locali
% RIPORTO in x
% x_l_kk(:, k, l) = (inversa(Z_l_kk(:, :, k, l))) * z_l_kk(:, k, l);
% x_l_kk(:, T, l) = P_l_kk(:, :, T, l) * z_l_kk(:, T, l); % converto da z a x
x_l_kk(:, T, l) = (C_l(:, :, l)' * R_l_inv(l) * C_l(:, :, l) +
+ P_l_ks(:, :, T, l)^(-1))^(-1) * z_l_kk(:, T, l);
% utilizzando variabili del modello ridotto
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2) CONSENSUS TRA LE STIME AGGIORNATE IN X
for(l=1:ns) % scandisco ciascun sensore
% per come configurato il sistema possono avere che una componente di stato
% sia oss. da l lo sia anche da un'altro sensore, solo se vicino
% ovvero attiguo l+1 e l-1
    lp = l+1;
    if(lp > ns)
        lp = lp - ns;
    end
    lm = l-1;
    if(lm <= 0)
        lm = ns + lm;
    end
    for(jj=1:s_oss) % scandisco gli stati oss dal sens
        x_l_kkp(jj, T, l) = x_l_kk(jj, T, l);
        for(hh=1:s_oss) % scandisco gli stati oss dai sens vicini
            if(x_l(:, jj, l) == x_l(:, hh, lp)) % se vedono la stessa componente
                % aggiorno
                x_l_kkp(jj, T, l) = x_l_kkp(jj, T, l) * 0.5 + x_l_kk(hh, T, lp) * 0.5;
            end
            if((x_l(:, jj, l)) == x_l(:, hh, lm))
                x_l_kkp(jj, T, l) = x_l_kkp(jj, T, l) * 0.5 + x_l_kk(hh, T, lm) * 0.5;
            end
        end
    end
end
end
x_l_kk(:, T, :) = x_l_kkp(:, T, :); % ridefinisco le stime con quelle dopo
% il consensus
return

```

RIFERIMENTI BIBLIOGRAFICI

- [1] Rudolf E. Kalman. *A new approach to linear filtering and prediction problems*. Journal of Basic Engineering, vol.82, no.2, pp. 35-45, 1960.
- [2] Giorgio Picci. *Filtraggio Statistico (Wiener, Levinson, Kalman) e Applicazioni*. Libreria Progetto, Padova.
- [3] Luca Schenato. *Appunti del corso di Progettazione di Sistemi di Controllo*. <http://www.dei.unipd.it/~schenato>
- [4] R. Olfati-Saber e J. S. Shamma. *Consensus filters for sensor networks and distributed sensor fusion*. Proceedings of the 44th Conference on Decision and Control, 2005.
- [5] R. Carli, A. Chiuso, L. Schenato e S. Zampieri, *Distributed kalman filtering based on consensus strategies*. Dipartimento di Ingegneria dell'Informazione, Università di Padova, 2007.
- [6] Usman A. Khan and José M. F. Moura. *Model Distribution For Distributed Kalman Filters: A Graph Theoretic Approach*.
- [7] Usman A. Khan and José M. F. Moura. *Distributing the Kalman filters for large-scale systems*. Submitted to IEEE Trans. on Signal Processing, Aug. 2007. <http://arxiv.org/pdf/0708.0242>
- [8] N. Balram and José M. F. Moura. *Noncausal Gauss Markov random fields: Parameter structure and estimation*. IEEE Trans. on Information Theory, vol. 39, no.4, pp. 1333-1355, Jul. 1993.
- [9] A. Kavcic and José M. F. Moura. *Matrices with banded inverses: Inversion algorithms and factorization of Gauss-Markov processes*. IEEE Trans. on Information Theory, vol.46, no.4, pp. 1495-1509, Jul. 2000.
- [10] A. Asif and José M. F. Moura. *Inversion of blockmatrices with L-Block banded inverse*. IEEE Trans. on Sig. Proc., vol.53, no.2, pp. 630-642, Feb. 2005.
- [11] Usman A. Khan e José M. F. Moura. *Distributed Kalman filters in sensor networks: Bipartite fusion graphs*. 15th IEEE Workshop on Statistical Sig. Proc., Madison, WI, Aug. 26-29 2007, pp. 700-704.
- [12] Jean Baptiste Joseph Fourier. *The analytical theory of heat*. Cambridge University Press, 1878.
- [13] John H. Lienhard IV e John H. Lienhard V. *A Heat Transfer Textbook, 3rd edition*. Phlogiston Press, Cambridge MT.