

Strategie di Sensor Fusion applicate al tracking

Gennaro Azzollini

matr: 570022-IAMs mail: gennaro.azzollini@libero.it

Luciano Lunardi

matr: 566877-IAMs mail: ingdj77@gmail.com

Sebastiano Segantin

matr: 566746-IAMs mail: sebastiano.segantin@gmail.com

Abstract—In questo lavoro si vuole realizzare un sistema per la localizzazione ed il tracking di uno o più veicoli mobili all'interno di un ambiente strutturato sul quale sono predisposti a tale scopo molteplici sensori di diversa natura, quali una rete di sensori in radiofrequenza ed una rete di videocamere. Il progetto viene scomposto in due specifici sottoproblemi appartenenti entrambi al paradigma generale del *Sensor Fusion*: il primo consiste nella coordinazione dei singoli sensori, mentre il secondo sottoproblema consiste nella fusione dei dati da essi provenienti. Parallelamente, per permettere il funzionamento in modalità multitasking, viene affrontata anche la questione dell'associazione dei dati in un sistema multi-sensore quale è quello considerato.

Il filtro di Kalman in forma d'informazione ed il filtro particellare sono le due tecniche messe in pratica per la realizzazione dello stadio di fusione dei dati. Algoritmi ad-hoc sono implementati per la coordinazione delle videocamere e per lo stadio di associazione dei dati.

Index Terms—Filtro d'Informazione, Filtro di Kalman, Filtro Particellare, Sensor Cooperation, Data Association, Data Fusion.

I. INTRODUZIONE

QUANDO si parla di *Sensor Fusion* ci si riferisce generalmente ad un insieme di specifiche tecniche sviluppate allo scopo di organizzare e gestire in modo intelligente una certa moltitudine di sensori e la grande mole di dati da essi prodotta. Il gran numero di pubblicazioni sull'argomento sono un chiaro indice del crescente interesse per la disciplina da parte di diverse aree di ricerca: interesse certamente spinto dall'attuale disponibilità di sensori e unità di elaborazione distribuita a basso costo.

I campi applicativi che hanno contribuito allo sviluppo di queste tecniche sono diversi e, solo per fare alcuni esempi, si citano le applicazioni aerospaziali e militari nelle quali le tecniche di *Sensor Fusion* sono sfruttate per realizzare complessi sistemi di navigazione, inseguimento o puntamento [16] [25] nonché di gestione

intelligente delle risorse nel campo di battaglia; le applicazioni di automotive e della guida assistita nelle quali si integrano e si fondono le informazioni provenienti da diversi sistemi di monitoraggio e navigazione presenti a bordo del veicolo [5] [28]; ultime, ed in grande diffusione oggi, le applicazioni di videosorveglianza e fornitura di servizi multimediali nelle quali, se pur con scopi evidentemente diversi, si realizzano reti di videocamere e sensori wireless che interagiscono tra loro e con i soggetti presenti all'interno dell'area di copertura [6] [8] [10].

Un sistema di *Sensor Fusion* è caratterizzato principalmente da due entità: la rete di sensori e la struttura del sistema di fusione.

Le motivazioni che spingono alla realizzazione di una rete di sensori sono molteplici: in generale, si sceglie di riunire in una rete un certo numero di sensori indipendenti allo scopo di ottenere potenzialità che un sensore non può essere in grado di possedere singolarmente. Le reti di sensori realizzabili possono avere diversi livelli di complessità e possono essere basate su svariate architetture specifiche che determinano i requisiti di capacità computazionale e di comunicazione che l'intero sistema deve soddisfare.

Esistono diverse modalità secondo le quali i sensori possono interagire tra loro all'interno delle reti da essi stessi costituite, e da queste modalità dipendono le specifiche strategie da adottare per l'operazione di fusione. Le modalità di interazione cui si fa comunemente riferimento, e che verranno ora descritte nel dettaglio, sono la *Complementarietà*, la *Concorrenza* e la *Cooperazione*, come discusso in [7].

- *Sensori Complementari*: La complementarietà si può manifestare in due diverse modalità: sono complementari un insieme di sensori indipendenti e di natura diversa che forniscono informazioni relativamente a diverse grandezze fisiche, oppure sono complementari anche sensori distinti ma dello stesso tipo che agiscono in diverse zone di rilevamento. In entrambi i casi, comunque, l'obiettivo è

combinare le misurazioni dei sensori della rete allo scopo di ottenere una descrizione più completa del fenomeno osservato.

Secondo le due modalità appena introdotte, dunque, una rete di sensori complementari è particolarmente utile quando si vogliono monitorare diversi aspetti dello stesso fenomeno al fine di descriverlo in modo più preciso, oppure quando si ha l'esigenza di acquisire misure in un'area molto vasta che non potrebbe mai essere coperta da un solo sensore.

Esempi di sensori complementari secondo la prima modalità sono l'odometro ed il tachimetro in un veicolo autonomo, la cui combinazione è parte integrante di qualsiasi sistema di navigazione automatica. Alla seconda modalità, invece, appartengono ad esempio delle videocamere con aree visive contigue la cui combinazione permette di ottenere un sistema con campo visivo molto vasto.

- *Sensori Concorrenti*: insieme di sensori distinti che forniscono informazioni indipendenti relativamente alla stessa grandezza fisica o alla stessa zona di rilevamento. Sensori concorrenti possono essere identici o possono sfruttare metodi differenti per misurare la stessa grandezza fisica. La concorrenza tra sensori è generalmente sfruttata per aumentare l'affidabilità del sistema di misura e per aumentarne il livello di tolleranza al malfunzionamento di qualche singolo sensore.

Per esempio, appartengono a questa categoria di sensori delle videocamere che inquadrano la stessa porzione di ambiente, o più termometri che misurano la temperatura della stessa stanza.

- *Sensori Cooperanti*: una rete di sensori cooperanti è costituita da un insieme di sensori indipendenti la cui osservazione è combinata allo scopo di ottenere informazioni che non potrebbe essere dedotte dall'informazione prodotta da un singolo sensore. Questo tipo di interazione tra sensori è particolarmente difficile da realizzare e, al contrario dell'interazione di concorrenza, causa generalmente un deterioramento dell'affidabilità e della robustezza del sistema di misura.

Un buon esempio dell'interazione cooperativa tra sensori è costituito da due videocamere configurate per la stereo-visione: dalla visione bidimensionale delle due è possibile ricostruire l'informazione di profondità della regione osservata ottenendone una rappresentazione tridimensionale.

La realizzazione del sistema di fusione dell'informazione, invece, è generalmente nota in letteratura come il problema del *Data-Fusion*. Gli

approcci proposti sono molti e della natura più varia ma, tra tutti, si è scelto di considerare nella trattazione attuale solo i due maggiormente sviluppati: il metodo del filtraggio statistico secondo la teoria di Kalman ed il metodo dei filtri particellari, i quali verranno introdotti rispettivamente nella Sezione II e nella Sezione III.

A prescindere dall'approccio utilizzato, inoltre, è possibile organizzare il sistema di data-fusion in modo tale che l'elaborazione dell'informazione avvenga secondo una *struttura centralizzata* o secondo una *struttura distribuita*, le quali vengono ora descritte in riferimento ad un sistema di tracking.

- *Struttura Centralizzata*: Le misure acquisite da ogni sensore vengono inviate ad un'unità centrale nella quale vengono opportunamente processate al fine di ottenere la stima dello stato per il modello del veicolo di cui si sta eseguendo il tracking.

Il principale vantaggio degli algoritmi centralizzati consiste nella minor perdita di informazione intrinseca: che è conseguenza dell'invio di "dati grezzi"; per contro, però, è richiesta una notevole capacità computazionale a livello dell'unità centrale e, inoltre, i fenomeni di errori di comunicazione e perdita di pacchetti influiscono in modo considerevole sulle prestazioni del sistema.

- *Struttura Distribuita*: Ogni sensore è dotato di capacità computazionale sufficiente a processare le proprie misure (*Smart Sensor*) al fine di ottenere una stima locale del vettore di stato del veicolo, la quale viene poi inviata all'unità centrale che si occupa di combinare in modo opportuno le diverse stime locali ricevute ricavandone una stima globale. Il fatto che una parte dell'elaborazione dei dati avvenga a livello di singolo sensore comporta una minor necessità di capacità computazionale a livello dell'unità centrale. Inoltre, una struttura distribuita è di particolare interesse qualora si voglia dotare un insieme di sensori complementari della capacità di auto-organizzarsi comunicando tra loro le proprie informazioni locali.

Per quanto riguarda i fenomeni di errori di comunicazione e perdita di pacchetti un sistema distribuito gode chiaramente di una certa robustezza dovuta semplicemente al fatto che ogni stato stimato locale trasmesso condensa in sé tutta l'informazione relativa al passato, indipendentemente dagli eventuali errori di comunicazione eventualmente manifestatisi nelle precedenti comunicazioni.

Un sistema a struttura distribuita, comunque, è generalmente caratterizzato da un grado di complessità maggiore rispetto ad un sistema centraliz-

zato. Ad esempio, basti pensare al problema teorico legato al fatto che le stime locali sono ottenute come soluzioni di problemi di ottimizzazione locale spesso tra loro non indipendenti ed il cui carattere di ottimalità locale non si traduce in modo chiaro in un carattere di ottimalità globale durante l'operazione di data-fusion.

Nel presente lavoro, per studiare e mettere in pratica il paradigma del *Sensor Fusion*, si è scelto di affrontare una delle sue più tipiche applicazioni: il tracking di veicoli mobili all'interno di un ambiente disseminato di opportuni sensori. Nello specifico, l'ambiente è dotato sia di una rete di sensori in radiofrequenza fissi in grado di comunicare tra loro e con un sensore mobile montato a bordo del veicolo mobile, sia di una rete di videocamere ognuna delle quali è in grado di inquadrare il veicolo mobile solo in una determinata porzione dell'ambiente. L'utilità di uno scenario così strutturato sta nel rendere possibile lo sviluppo e lo studio di un sistema dotato di sensori concorrenti e complementari, nonché lo studio e l'implementazione di algoritmi di fusione dei dati a struttura centralizzata o distribuita.

Inoltre, il sistema è organizzato in modo modulare e decentralizzato. Ne risulta, dunque, una struttura complessiva facilmente estendibile e riconfigurabile, il tutto senza la necessità di un'unità centrale con una elevata potenza di calcolo.

II. DATA FUSION MEDIANTE FILTRO DI KALMAN

Come noto, il Filtro di Kalman costituisce un algoritmo ricorsivo ottimo, nel senso della minimizzazione della varianza d'errore di stima, utilizzabile per stimare lo stato di un sistema lineare. Dunque, è naturale pensare di applicare questa teoria anche ai sistemi di tracking multi sensore basati sulle tecniche di Sensor Fusion, previo un opportuno adattamento dell'algoritmo al contesto specifico.

Tale adattamento consiste nell'utilizzare il cosiddetto *Filtro di Kalman in forma d'Informazione* (o più semplicemente Filtro d'Informazione) al posto del Filtro di Kalman classico. Il Filtro d'Informazione può essere dedotto formalmente dalla sua formulazione classica, e si rimanda all'Appendice B per i dettagli del procedimento.

Ci si limita qui a riportare le espressioni finali dell'algoritmo ricorsivo del filtro:

$$\hat{z}_{k+1|k} = Z_{k+1|k} A Z_{k+1|k}^{-1} \hat{z}_{k|k} \quad (1a)$$

$$Z_{k+1|k} = (A Z_{k|k} A^T + Q)^{-1} \quad (1b)$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + C^T R^{-1} y_{k+1} \quad (2a)$$

$$Z_{k+1|k+1} = Z_{k+1|k} + C^T R^{-1} C \quad (2b)$$

L'interesse principale verso il Filtro d'Informazione appena introdotto sta nella possibilità di adattarlo facilmente al funzionamento in un sistema multi-sensore all'interno del quale esso assolve in modo naturale al ruolo di operatore di data-fusion. Inoltre, esso permette di svolgere l'operazione di data-fusion sia in modo centralizzato sia in modo distribuito.

A. Filtro d'Informazione centralizzato

Volendo realizzare un sistema di tracking centralizzato che elabora la stima ottima globale sulla base delle informazioni ricevute dai sensori, risulta che il Filtro d'Informazione centralizzato si basa semplicemente sulla somma dei singoli contributi d'informazione prodotti dai sensori.

Si supponga che il sistema sia costituito da un numero complessivo N di sensori e si faccia l'ipotesi fondamentale che essi siano soggetti a rumori di misura tra loro scorrelati, allora, definendo:

$$y_k \triangleq [y_k^1 \dots y_k^N]^T$$

è immediato verificare che vale la relazione:

$$C^T R^{-1} y_k = \begin{bmatrix} C^{1T} & \dots & C^{N^T} \end{bmatrix} \begin{bmatrix} R^{1^{-1}} & & \\ & \ddots & \\ & & R^{N^{-1}} \end{bmatrix} \begin{bmatrix} y_k^1 \\ \vdots \\ y_k^N \end{bmatrix} \quad (3)$$

dalla quale si ricava immediatamente:

$$C^T R^{-1} y_{k+1} = \sum_{i=1}^N C^{iT} R^{i^{-1}} y_{k+1}^i \quad (4)$$

e, partizionando allo stesso modo anche il termine $C^T R^{-1} C$, si ottiene anche la scrittura:

$$C^T R^{-1} C = \sum_{i=1}^N C^{iT} R^{i^{-1}} C^i \quad (5)$$

Allora, sostituendo le due relazioni appena trovate nelle equazioni di aggiornamento del Filtro d'Informazione (2), si ottiene una struttura per l'algoritmo ricorsivo del Filtro d'Informazione centralizzato che è la seguente:

$$\hat{z}_{k+1|k} = Z_{k+1|k} A Z_{k+1|k}^{-1} \hat{z}_{k|k} \quad (6a)$$

$$Z_{k+1|k} = (A Z_{k|k} A^T + Q)^{-1} \quad (6b)$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + \sum_{i=1}^N C^{iT} R^{i^{-1}} y_{k+1}^i \quad (7a)$$

$$Z_{k+1|k+1} = Z_{k+1|k} + \sum_{i=1}^N C^{iT} R^{i^{-1}} C^i \quad (7b)$$

Dunque, è ora evidente la comodità del Filtro d'Informazione in quanto esso si basa sulla semplice somma dei contributi di misura apportati da ogni singolo sensore per ottenere la stima globale ottima.

In conclusione, grazie alla struttura diagonale a blocchi della matrice R , ovvero grazie all'ipotesi di scorrelazione dei rumori di misura dei singoli sensori, si è ottenuta una struttura particolarmente semplice per il Filtro d'Informazione centralizzato. Tuttavia bisogna osservare che la suddetta ipotesi può essere inaccettabile in determinate applicazioni reali nelle quali, di conseguenza, il filtro esibisce scarse prestazioni. Si può già anticipare che esiste una soluzione a questo problema basata su una riformulazione del filtro che tenga conto di questa correlazione tra i rumori di misura, ed essa verrà discussa nel seguito della trattazione.

B. Filtro d'Informazione parallelo

Avendo introdotto il Filtro d'Informazione centralizzato nel paragrafo precedente, si vuole ora derivarne una struttura in parallelo. Si tratta sostanzialmente di dedurre una formulazione equivalente del filtro nella quale l'elaborazione centralizzata avviene ricevendo dai sensori una stima locale dello stato anziché la semplice misura effettuata; ciò significa che i sensori utilizzati devono essere necessariamente dotati di una capacità computazionale sufficiente a permettere l'implementazione di uno stimatore locale al loro interno.

Per ricavare la struttura in parallelo del filtro si consideri innanzitutto lo stimatore locale del generico sensore i -esimo: la sue equazioni coincidono con la relazioni generali (1) e (2) alle quali si aggiunge il consueto apice i indicativo del sensore locale. Dall'equazione di aggiornamento dello stato (2a) dopo un semplice riordino dei termini si ottiene la relazione:

$$C^{iT} R^{i-1} y_{k+1}^i = \hat{z}_{k+1|k+1}^i - \hat{z}_{k+1|k}^i \quad (8)$$

mentre dalla (2b) si ottiene allo stesso modo:

$$C^{iT} R^{i-1} C^i = Z_{k+1|k+1}^i - Z_{k+1|k}^i \quad (9)$$

In fine, sostituendo direttamente nelle (7) le due relazioni appena trovate si ottiene la formulazione in parallelo del Filtro d'Informazione cercata, il cui algoritmo ricorsivo può essere riassunto dalle espressioni seguenti:

$$\hat{z}_{k+1|k} = Z_{k+1|k} A Z_{k+1|k}^{-1} \hat{z}_{k|k} \quad (10a)$$

$$Z_{k+1|k} = (A Z_{k|k} A^T + Q)^{-1} \quad (10b)$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + \sum_{i=1}^N \left(\hat{z}_{k+1|k+1}^i - \hat{z}_{k+1|k}^i \right) \quad (11a)$$

$$Z_{k+1|k+1} = Z_{k+1|k} + \sum_{i=1}^N \left(Z_{k+1|k+1}^i - Z_{k+1|k}^i \right) \quad (11b)$$

È opportuno sottolineare che dal punto di vista prestazionale la struttura parallelo è equivalente alla struttura centralizzata, tuttavia può essere preferibile sia nella realizzazione di sistemi distribuiti sia nell'utilizzo in sistemi nei quali la comunicazione è soggetta a perdita di pacchetti. Quest'ultima considerazione è di carattere del tutto generale ed è collegata al fatto che il vettore di stato condensa in sé tutta l'informazione riguardante il passato del sistema dinamico cui è riferito; ciò significa che anche se avviene una perdita di pacchetto, l'informazione persa con esso viene recuperata intrinsecamente con il primo pacchetto successivamente ricevuto in modo corretto dall'unità di data-fusion.

C. Filtro d'Informazione con rumori di misura correlati

Come già anticipato alla fine del Paragrafo II-A nelle applicazioni pratiche è spesso troppo restrittivo supporre che i rumori di misura dei sensori siano tra loro scorrelati. Per fare un esempio, si consideri il caso di due videocamere concorrenti che inquadrano un target in una particolare condizione di illuminazione che crea un effetto d'ombra o un riflesso tale da pregiudicare l'azione delle videocamere stesse.

Quella della correlazione tra misure (o tra stime locali) nei sistemi distribuiti multi-sensore è una questione particolarmente importante perchè può condizionare in modo anche pesante le proprietà statistiche del sistema, alterandone il carattere di ottimalità. Le soluzioni che vengono proposte in letteratura sono molteplici e della natura più varia: comunque, verrà qui discussa nel dettaglio solo quella elaborata in [23] essendo essa la soluzione che può essere dedotta in modo più naturale dai ragionamenti fatti fino a questo punto.

Formalmente, supporre che il sistema sia costituito da un numero complessivo N di sensori soggetti a rumori di misura non necessariamente scorrelati tra loro, significa considerare che la matrice R^{-1} che compare nella relazione (3) non sia necessariamente diagonale a blocchi: di conseguenza, non si possono supporre valide le decomposizioni in sommatorie date dalle uguaglianze (4) e (5), le quali sono alla base di entrambi gli algoritmi di Data-Fusion discussi fino ad ora. Ciò nonostante, sfruttando in modo opportuno le proprietà strutturali della relazione (3) e applicandovi semplici ragionamenti

algebrici, è ancora possibile ricavare espressioni interessanti sia per il Filtro d'informazione centralizzato che per il Filtro d'informazione parallelo.

Per capire in modo chiaro i passaggi algebrici eseguiti e le espressioni che se ne ricavano conviene ragionare su un esempio pratico. Si consideri un sistema di tracking multisensore con le seguenti caratteristiche dimensionali:

- $N = 3$ sensori, dei quali il primo ed il terzo si suppongono caratterizzati da rumori di misura correlati.
- n sia la dimensione del vettore di stato dei modelli utilizzati.
- $p = p_1 + p_2 + p_3$ sia la dimensione del vettore di misura complessivo, costituito dalla somma delle dimensioni dei vettori di misura dei singoli sensori del sistema.

Allora, la relazione (3) può essere scritta esplicitamente come:

$$C^T R^{-1} y_k = \underbrace{\begin{bmatrix} C^{1T} & C^{2T} & C^{3T} \end{bmatrix}}_{\in R^{n \times p}} \underbrace{\begin{bmatrix} R^1 & 0 & R^{13} \\ 0 & R^2 & 0 \\ R^{13} & 0 & R^3 \end{bmatrix}}_{\in R^{p \times p}} \underbrace{\begin{bmatrix} y_k^1 \\ y_k^2 \\ y_k^3 \end{bmatrix}}_{\in R^{p \times 1}} \quad (12)$$

da cui, eseguendo un prodotto e lasciando scritta per brevità in modo non scomposta la matrice C^T :

$$\begin{aligned} C^T R^{-1} y_k &= C^T \begin{bmatrix} R^1 y_k^1 + R^{13} y_k^3 \\ R^2 y_k^2 \\ R^{13} y_k^1 + R^3 y_k^3 \end{bmatrix} \\ &= C^T \left(\begin{bmatrix} R^1 \\ 0 \\ R^{13} \end{bmatrix} y_k^1 + \begin{bmatrix} 0 \\ R^2 \\ 0 \end{bmatrix} y_k^2 + \begin{bmatrix} R^{13} \\ 0 \\ R^3 \end{bmatrix} y_k^3 \right) \\ &= C^T \sum_{i=1}^3 R_{*i}^{-1} y_k^i \end{aligned} \quad (13)$$

dove $R_{*i}^{-1} \in R^{p \times p_i}$ indica la i -esima sezione colonna della matrice R^{-1} .

Generalizzando la (13) si ottiene la relazione che costituisce la naturale estensione della (4) al caso di rumori di misura correlati:

$$C^T R^{-1} y_{k+1} = C^{iT} \sum_{i=1}^N R_{*i}^{-1} y_{k+1}^i \quad (14)$$

ed in modo analogo, in fine, si generalizza la (5) ottenendo la relazione:

$$\begin{aligned} C^T R^{-1} C &= C^T \sum_{i=1}^N R_{*i}^{-1} C^i \\ &= C^T \sum_{i=1}^N R_{*i}^{-1} R^i (C^{iT})^\dagger C^{iT} R^{i-1} C^i \end{aligned} \quad (15)$$

nella quale l'apice \dagger indica l'operazione pseudo-inversione¹ e la seconda equivalenza, pur sembrando inutilmente più complicata della prima, ha il vantaggio di permettere l'applicazione immediata della relazione (9) per la deduzione del filtro d'informazione parallelo dal filtro centralizzato, grazie alla presenza esplicita del termine $C^{iT} R^{i-1} C^i$.

A questo punto è immediato ripercorrere i ragionamenti che hanno portato alla deduzione degli algoritmi ricorsivi del filtro d'informazione centralizzato e parallelo avendo l'accortezza di utilizzare le relazioni (14) e (15) al posto delle (4) e (5). Se ne riporta qui il risultato finale relativo al solo stadio di aggiornamento dello stato, essendo lo stato di predizione non influenzato direttamente dalle misure dei sensori. Dunque, per quanto riguarda la formulazione centralizzata del filtro, si ottiene:

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + C^{iT} \sum_{i=1}^N R_{*i}^{-1} y_{k+1}^i \quad (16a)$$

$$Z_{k+1|k+1} = Z_{k+1|k} + C^T \sum_{i=1}^N R_{*i}^{-1} R^i (C^{iT})^\dagger C^{iT} R^{i-1} C^i \quad (16b)$$

mentre per quanto riguarda la formulazione parallelo il risultato è il seguente:

$$\begin{aligned} \hat{z}_{k+1|k+1} &= \hat{z}_{k+1|k} + \\ &C^T \sum_{i=1}^N R_{*i}^{-1} R^i (C^{iT})^\dagger \left(\hat{z}_{k+1|k+1}^i - \hat{z}_{k+1|k}^i \right) \end{aligned} \quad (17a)$$

$$\begin{aligned} Z_{k+1|k+1} &= Z_{k+1|k} + \\ &C^T \sum_{i=1}^N R_{*i}^{-1} R^i (C^{iT})^\dagger \left(Z_{k+1|k+1}^i - Z_{k+1|k}^i \right) \end{aligned} \quad (17b)$$

Dal punto di vista algoritmico, quindi, la complicazione dovuta all'eventuale presenza di rumori di misura non scorrelati è assolutamente accettabile.

D. Applicazione del Filtro d'Informazione

Prima di implementare il Filtro d'Informazione all'interno del sistema di tracking realizzato, si vuole illustrarne le prestazioni con l'ausilio di semplici simulazioni. Dato che il sistema che si vuole realizzare è di tipo distribuito si è interessati all'utilizzo del filtro d'informazione nella sua formulazione in parallelo;

¹ L'espressione ha senso se vale che $C^i (C^i)^\dagger = I$ e condizione necessaria affinché tale uguaglianza sia verificata è che la matrice C^i abbia rango pieno per righe.

pertanto le simulazioni effettuate si basano sull'utilizzo dell'algoritmo introdotto nel precedente Paragrafo II-B.

Lo scenario considerato è costituito semplicemente da due sensori generici indipendenti e aventi regioni di copertura parzialmente sovrapposte, cosicché esista una porzione di ambiente nella quale essi sono concorrenti. Nel dettaglio, si suppone che entrambi i sensori coprano interamente l'ambiente lungo la direzione y , mentre lungo l'asse x il primo sensore copre una regione limitata all'intervallo di punti $[r1_i, r1_f]$, e il secondo sensore copre una regione corrispondente all'intervallo $[r2_i, r2_f]$ con il vincolo $r1_f > r2_i$ necessario a garantire l'esistenza di una zona di sovrapposizione. Il target è costituito da un veicolo che percorre una traiettoria di tipo sinusoidale all'interno dell'ambiente monitorato dai due sensori.

Entrambi i sensori implementano il filtro d'informazione standard secondo l'algoritmo dato dalle relazioni (1) e (2); le stime locali ottenute in questo modo vengono poi elaborate nel blocco di data-fusion dall'algoritmo corrispondente alle equazioni (10) e (11). Per modellare la dinamica del veicolo i due filtri locali hanno a disposizione lo stesso modello di stato pilotato da rumore bianco (si veda l'Appendice A-A) e si assume che sia lo stesso anche il rumore di modello (matrice Q), mentre per quanto riguarda il rumore di misura (matrice R) si suppone che quello a cui è soggetto il primo sensore sia inferiore del 20% rispetto a quello cui è soggetto il secondo sensore.

Il tracking del veicolo che si ottiene da questa procedura è illustrato in Figura 1 assieme alla traiettoria vera stabilita per la simulazione.

Più interessanti, comunque, sono i risultati che si possono osservare in Figura 2. Innanzitutto bisogna distinguere due modalità con le quali è possibile inizializzare gli algoritmi di stima locale. La modalità più semplice, corrispondente alla Figura 2(a), prevede di inizializzare il i -esimo stimatore locale con valori predefiniti, per esempio nel modo seguente:

$$\begin{cases} \hat{z}_{0|-1}^i = 0 \\ \hat{Z}_{0|-1}^i = I \end{cases} \quad (18)$$

In alternativa, con riferimento alla Figura 2(b), se si dispone di un canale di comunicazione bidirezionale tra i sensori ed il centro di data-fusion, una soluzione più efficiente consiste nell'inviare allo stimatore locale i -esimo la stima globale disponibile a livello di data-fusion, ovvero lo si inizializza ponendo:

$$\begin{cases} \hat{z}_{0|-1}^i = \hat{z}_{-1|-1} \\ \hat{Z}_{0|-1}^i = \hat{Z}_{-1|-1} \end{cases} \quad (19)$$

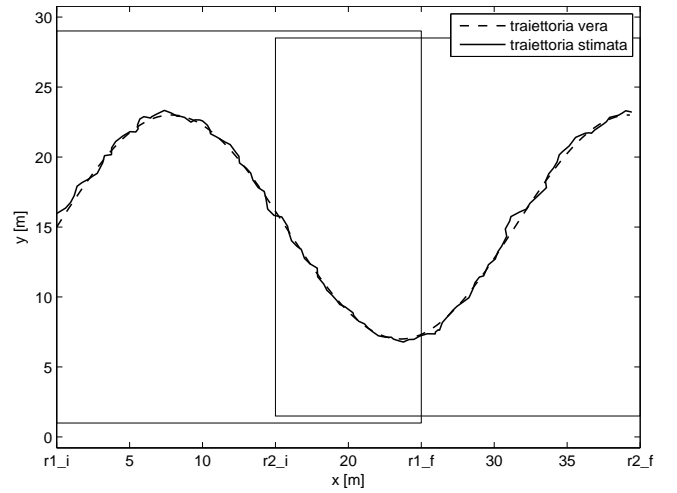


Fig. 1. Traiettoria reale a stimata del veicolo mobile nell'ambiente monitorato da due sensori. I due rettangoli a tratto più fine danno un'indicazione delle regioni di copertura dei sensori.

Confrontando tra loro i due grafici della Figura 2, comunque, è evidente che le diverse modalità di inizializzazione locale non influenzano in alcun modo le prestazioni globali del sistema (dimostrato analiticamente in [23] anche nel caso limite in cui la stima globale $\hat{z}_{t-1|t-1}$ venga utilizzata negli stadi di predizione locale ad ogni iterazione e non solo in quella di inizializzazione).

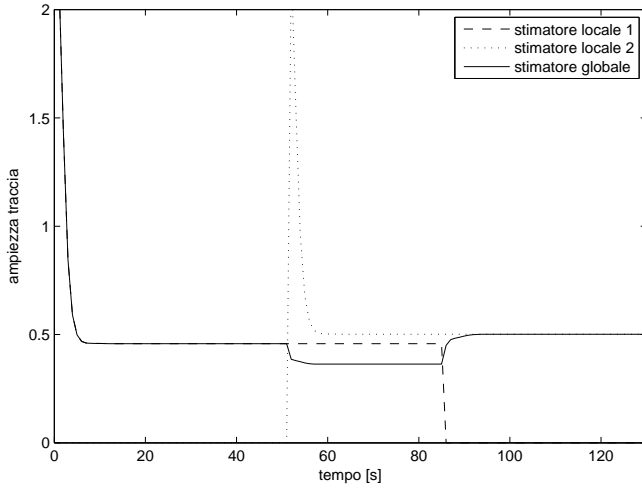
A prescindere dalla strategia di inizializzazione adottata, è significativo osservare che la varianza d'errore dello stimatore globale esibisce una significativa diminuzione in corrispondenza del transito del target nella regione monitorata da entrambi i sensori. Si tratta di una conferma evidente della capacità del filtro globale di estrapolare informazione in modo ottimale a partire da dati provenienti da più sensori.

III. DATA FUSION MEDIANTE FILTRO PARTICELLARE

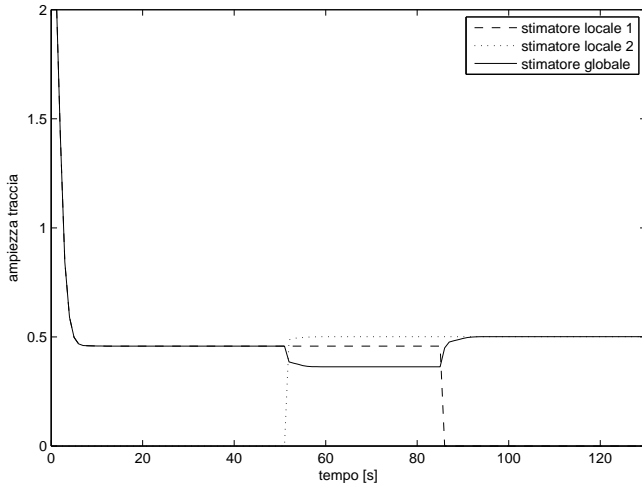
A volte, situazioni controllistiche particolari possono richiedere modellizzazioni del problema in spazio di stato con comportamento non lineare, e di conseguenza l'utilizzo del filtro di Kalman diventa impossibile. Si ricorrono quindi ad approssimazioni particolari, ovviamente non ottime, come il filtro di Kalman esteso. Un metodo alternativo al filtro di Kalman esteso da utilizzare per il tracking di oggetti in movimento nel caso non lineare è il cosiddetto Particle filter (o filtro particellare).

Questo tipo di filtro, profondamente diverso a livello implementativo dal filtro di Kalman esteso, appartiene alla famiglia dei metodi Monte Carlo.

I metodi Monte Carlo consistono in algoritmi volti a risolvere problemi computazionali utilizzando la generazione di numeri pseudocasuali (quelli generati da un



(a) Inizializzazione locale secondo le relazioni (18).



(b) Inizializzazione locale secondo le relazioni (19).

Fig. 2. Traccia della matrice Varianza d'errore di stima dei filtri locali e del filtro globale di data-fusion per due diverse strategie di inizializzazione degli stimatori locali.

qualsiasi elaboratore).

Approcci probabilistici di questo tipo sono ampiamente usati nella soluzione dei problemi di integrazione ed ottimizzazione di sistemi modellizzati in spazi di stato con un grande numero di dimensioni.

Un algoritmo Monte Carlo viene solitamente scelto quando algoritmi standard non raggiungono risultati soddisfacenti a causa della complessità del problema: infatti capita che, al contrario di altri metodi, l'efficienza cresce con l'aumentare delle dimensioni del problema.

Riconducendosi ad un problema di stima generale, e assumendo che gli stati del sistema siano Markoviani, non lineari e non Gaussiani; indicando con $x_{0:k} := \{x_0, x_1, \dots, x_k\}$ e $z_{1:k} := \{z_1, z_2, \dots, z_k\}$ l'evoluzione dello stato e la sequenza delle osservazioni fino all'istante k , si cerca di stimare ricorsivamente

l'aspettazione di una qualsiasi funzione della distribuzione congiunta a posteriori $p(x_{0:k}|z_{1:k})$, ossia:

$$\begin{aligned} I(f_k) &= E_{p(x_{0:k}|z_{1:k})} [f_k(x_{0:k})] \\ &:= \int f_k(x_{0:k}) p(x_{0:k}|z_{1:k}) dx_{0:k}. \end{aligned} \quad (20)$$

Per fare questo si ricorre alla generazione e propagazione nel tempo di una versione approssimata della densità di probabilità ottenuta "campionando" la densità stessa, e manipolandola come fosse la distribuzione vera e propria.

Se consideriamo quindi questi campioni i.i.d $\{x_{0:k}^{(i)}\}_{i=1}^N$ distribuiti secondo la densità voluta, si può approssimare:

$$p(x_{0:k}|z_{1:k}) \approx p_N(x_{0:k}|z_{1:k}) = \frac{1}{N} \sum_{i=1}^N \delta(x_{0:k} - x_{0:k}^{(i)}), \quad (21)$$

e sostituendola nella formulazione di $I(f_k)$ ci si riporta ad un comodo risultato, approssimato $I_N(f_k)$:

$$I_N(f_k) = \frac{1}{N} \sum_{i=1}^N f_k(x_{0:k}^{(i)}). \quad (22)$$

Ciò è vantaggioso in quanto sostituendo la pdf di partenza con l'approssimazione discreta si può calcolare una qualsiasi stima $I_N(f_k)$ di $I(f_k)$.

A. Il tracking Bayesian non lineare

L'approccio Bayesian, cerca di ricavare teoricamente una stima della dinamica degli stati tramite la funzione densità di probabilità a posteriori (**PDF Posterior**), generandola in base a tutte le informazioni disponibili, misure comprese, in un intervallo temporale qualsiasi. Purtroppo nella realtà si hanno a disposizione informazioni solamente causali, cioè presenti nel passato, e nemmeno tutte, in quanto non si dispone di una memoria infinita.

La disponibilità di elaboratori, a volte non eccezionali, spinge quindi ad una categoria di algoritmi, detti on-line, che sfruttano le possibilità di implementazioni algoritmiche ricorsive, aventi a disposizione un numero limitato di informazioni (buffer, esecuzioni real time, etc.).

Nel nostro caso avremo a disposizione dei dati sequenziali (per questo il particle filter viene anche detto *Sequential Monte Carlo*), le misure, con le quali potremo elaborare, volta per volta, l'uscita del nostro filtro.

La natura discreta della situazione struttura quindi l'evoluzione in due momenti: l'istante in cui arriva la misura, ed il tempo che trascorre fra una misura e l'altra. Il nostro filtro avrà quindi due fasi principali:

- una fase di predizione, in cui useremo il modello per determinare la funzione densità di probabilità degli stati all'istante successivo (funzione densità di probabilità a priori [**PDF Prior**]),
- una fase di aggiornamento in cui utilizzeremo l'ultima misura disponibile per modificare la funzione densità di probabilità a priori **PDF Posterior**.

Nel dettaglio, il tracking Bayesiano non lineare, partendo dal sistema reale²:

$$\begin{cases} x_{k+1} = f(x_k, w_k) \\ z_k = h(x_k, v_k) \end{cases} \quad (23)$$

si propone di stimare ricorsivamente x_k dalle misure z_k rese disponibili dal metodo di rilevamento. In altre parole, cerchiamo di definire $p(x_k|z_{1:k})$ ³. Partendo dalle ipotesi di avere al passo $k-1$, $p(x_{k-1}|z_{1:k-1})$, tramite l'equazione di Chapman-Kolmogorov, e la regola di Bayes si ricava la PDF al passo successivo:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (24)$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (25)$$

Dove inoltre si considera la costante di normalizzazione:

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k \quad (26)$$

Ma questo metodo non è di immediata applicazione, infatti è una soluzione concettuale, per cui, per arrivare all'implementabilità serviranno altre ipotesi.

B. Le approssimazioni nel Particle Filter: l'Importance-Sampling

Le ipotesi di modello markoviano, la non gaussianità del rumore di modello, e la successiva ipotesi che faremo, cioè la rappresentazione della funzione densità di probabilità a posteriori tramite un set di campioni casuali con peso associato (tralaltro anch'essa lungi dall'essere gaussiana), rende l'utilizzo del particle filter una soluzione non ottima, ma al contempo, una soluzione di facile implementazione.

In letterature esistono moltissime varianti sul tema: Sequential Monte Carlo Method, Sampling-Importance-Resampling, Bootstrap, Condensation, Sequential-Importance-Sampling, Auxiliary Particle Filter, Regularized Particle Filter, 'Likelihood' Particle Filter, tra le quali esistono sinonimi e implementazioni

²Nel particle filter utilizziamo z come simbolo relativo alle misure, mentre w_k, v_k sono rumori.

³Si vedrà poi che sotto le ipotesi di modello markoviano, basterà calcolare $p(x_k|z_k)$.

che differiscono dall'originale solo di poco.

L'implementazione utilizzata sarà un derivato dell'Importance-Sampling, che andiamo a descrivere: l'approssimazione precedente della funzione di distribuzione $p_N(x_{0:k}|z_{1:k})$ era effettuata tramite una serie di "impulsi" di eguale probabilità $1/N$, ora introduciamo il concetto di "peso" che, tramite gli stessi campioni a disposizione, si riesce a rappresentare una diversa funzione densità di probabilità, aumentando o diminuendo tramite il peso la partecipazione degli impulsi stessi nella densità globale.

Infatti, se \mathcal{X} è l'alfabeto della variabile aleatoria discreta e w_i sono i pesi, si può di ottenere la distribuzione e la fdp della variabile continua equivalente rispettivamente come:

$$F_x(x) = \sum_{x_i \in \mathcal{X}} w_i \delta(x - x_i), \quad (27)$$

$$f(x) = \sum_{i=1}^N w_i \delta(x - x_i). \quad (28)$$

Si introduce ora il concetto della distribuzione Importance-Sampling: rappresentiamo la funzione densità di probabilità a posteriori tramite un set di campioni casuali con peso associato $\{x_{0:k}^i, w_k^i\}$.

Questi denotano una misura (realizzazione) casuale della variabile aleatoria con PDF-Posterior $p(x_{0:k}|z_{1:k})$. I punti $\{x_{0:k}^i, i = 0, \dots, N_s\}$ (dove N_s è il numero di particelle totali) mi rappresentano dei punti di supporto, con dei pesi associati $\{w_k^i, i = 1, \dots, N_s\}$ e $x_{0:k} = \{x_j, j = 0, \dots, k\}$ è il set dei punti di supporto all'istante k .

I pesi associati ad ogni stato sono inoltre normalizzati ad ogni istante: $\sum_i w_k^i = 1$.

La PDF-Posterior sarà quindi approssimata (con dei pesi discreti), come:

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_{0:k} - x_{0:k}^i). \quad (29)$$

A questo punto, supponiamo che $p(x) \propto \pi(x)$ sia una densità di probabilità dalla quale è difficile estrarne dei campioni, ma per il quale $\pi(x)$ può essere valutato (e così in modo proporzionale a $p(x)$).

Poi, siano x_i realizzazioni della v.a. $q(x)$, $i = 1, \dots, N_s$, cioè campioni facilmente generati da una densità proposta $q(\cdot)$, chiamata *proposal distribution*. Così, un'approssimazione pesata della densità $p(\cdot)$ è data da

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i), \quad (30)$$

dove $w^i \propto \frac{\pi(x^i)}{q(x^i)}$ è il peso normalizzato della i -esima particella. Riprendendo quindi $I(f_k)$ e tramite una serie di sostituzioni illustrate in appendice C-A si ottiene una formulazione di questo tipo:

$$I(f_k) = \frac{\int f_k(x_{0:k}) w_k(x_{0:k}) q(x_{0:k}|z_{1:k}) dx_{0:k}}{\int w_k(x_{0:k}) q(x_{0:k}|z_{1:k}) dx_{0:k}}, \quad (31)$$

che può essere utilmente approssimata, tramite , ma interpretata per $q(x)$, come:

$$\begin{aligned} \widehat{I}_N(f_k) &= \frac{\frac{1}{N} \sum_{i=1}^{N_s} f_k(x_{0:k}^{(i)}) w(x_{0:k}^{(i)})}{\frac{1}{N} \sum_{j=1}^{N_s} w(x_{0:k}^{(j)})} \\ &= \sum_{i=1}^N f_k(x_{0:k}^{(i)}) \tilde{w}_k^{(i)}, \end{aligned} \quad (32)$$

dove si usano i pesi d'importanza normalizzati (stimati):

$$\tilde{w}_k^{(i)} = \frac{w(x_{0:k}^{(i)})}{\frac{1}{N} \sum_{j=1}^N w(x_{0:k}^{(j)})}. \quad (33)$$

C. La ricorsione nel Particle Filter, il Sequential-Importance-Sampling

Come accennato prima, i pesi vengono scelti usando il principio dell'Importance Sampling.

Per creare una versione ricorsiva del filtro dobbiamo avere a disposizione le ipotesi valide del passo precedente, cioè avere una rappresentazione in "forma particellare" della pdf $p(x_{0:k-1}|z_{1:k-1})$ e di voler calcolare una approssimazione $\widehat{P}_N(x_{0:k}|z_{1:k})$ tramite una nuova serie di particelle⁴. Sfruttando quindi due serie di sostituzioni, tramite la formula di Bayes e il fatto che abbiamo a che fare con un processo markoviano:

$$\begin{aligned} q(x_{0:k}|z_{1:k}) &= q(x_{0:k-1}|z_{1:k-1}) q(x_k|x_{0:k-1}, z_{1:k}) \\ &= q(x_0) \prod_{j=1}^k q(x_j|x_{0:j-1}, z_{1:j}), \end{aligned} \quad (34)$$

$$p(x_{0:k}) = p(x_0) \prod_{j=1}^k p(x_j|x_{j-1}), \quad (35)$$

$$p(z_{1:k}|x_{0:k}) = \prod_{j=1}^k p(z_j|x_j), \quad (36)$$

è possibile ricavare una formulazione ricorsiva del calcolo dei pesi:

⁴si ricordano le ipotesi di Markov, cioè il passato non viene influenzato.

$$\begin{aligned} w_k &= \frac{p(z_{1:k}|x_{0:k}) p(x_{0:k})}{q(x_{0:k-1}|z_{1:k-1}) q(x_k|x_{0:k-1}, z_{1:k})} \\ &= w_{k-1} \frac{1}{w_{k-1}} \frac{p(z_{1:k}|x_{0:k}) p(x_{0:k})}{q(x_{0:k-1}|z_{1:k-1}) q(x_k|x_{0:k-1}, z_{1:k})} \\ &= w_{k-1} \frac{p(z_{1:k}|x_{0:k})}{p(z_{1:k-1}|x_{0:k-1})} \frac{p(x_{0:k})}{p(x_{0:k-1})} \frac{1}{q(x_k|x_{0:k-1}, z_{1:k})} \\ &= w_{k-1} \frac{p(z_k|x_k) p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}, z_{1:k})}. \end{aligned} \quad (37)$$

Ricapitolando si arriva ad una formulazione più pratica della PDF-Posterior:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_k - x_k^{(i)}), \quad (38)$$

dove i pesi sono proprio definiti come:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)}) p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)}. \quad (39)$$

Resta infine da scegliere una distribuzione adeguata di q .

D. Problemi di degenerazione: scelta di q e resampling

Il fenomeno della generazione del Sequential-Importance-Sampling consiste nella concentrazione del peso su di una singola particella a discapito di altre che hanno peso quasi nullo, dopo qualche iterazione.

Si possono porre soluzioni a questo problema introducendo dapprima un numero efficace di campioni, per avere una metrica di riscontro nel caso la degenerazione sia eccessiva.

Questa metrica tralatro è difficilmente calcolabile, ma viene approssimata tramite dei boundary:

$$N_{eff} = \frac{N}{1 + var(w^*(x_{0:k}))}. \quad (40)$$

Questo valore è quindi impostabile come threshold per il numero di particelle e l'attivazione di metodi anti-degenerazione.

Nella nostra implementazione non avremo problemi di questo tipo, in quanto il filtro patricellare avrà due metodi per contrastare questa situazione: la scelta mirata di q e l'applicazione del resampling ad ogni passo (implementazione del Sampling-Importance-Resampling).

- 1) **La scelta di q** viene quindi ad essere una delle più importanti in quanto può anch'essa definirni una soluzione ottimale ma difficile da calcolare. Nell'articolo [3], si evidenzia che la soluzione ottima sarebbe:

$$q(x_k|x_{k-1}, z_k)_{opt} = \frac{p(z_k|x_k, x_{k-1}^{(i)}) p(x_k|x_{k-1}^{(i)})}{p(z_k|x_{k-1}^{(i)})}, \quad (41)$$

che ridurrebbe i pesi in questa forma:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \int p(z_k|x'_k)p(x'_k|x_{k-1}^{(i)})dx'_k. \quad (42)$$

Ma essendo calcolabile solo in alcuni casi particolari (x_k set finito di stati o $p(x_k|x_{k-1}^{(i)}, z_k)$ gaussiano), si trascura, e si prende come *proposal distribution* la semplice PDF-Prior (si vedano i dettagli in [3]):

$$q(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)}), \quad (43)$$

che è facilmente implementabile e intuitiva, anche se in realtà appartiene alle soluzioni sub-ottime del problema.

2) **Il resampling** ha come idea base l'eliminazione delle particelle di poco peso, e concentrarsi su quello che ne hanno molto.

In altre parole, ad ogni passo si genera un nuovo set di particelle $\{x_k^{i*}\}_{i=1}^{N_s}$ ricampionando con rimpiazzamento N_s volte da una rappresentazione approssimazione discreta di $p(x_k|z_{1:k})$ data da:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i), \quad (44)$$

in modo che $p(x_k^{i*} = x_k^j) = w_k^j$. I campioni risultanti sono infatti delle realizzazioni i.i.d della densità discreta $p(x_k|z_{1:k})$, e così i pesi sono resettati a $w_k^i = 1/N_s$. Questo metodo implica ulteriori problemi di impoverimento ma che non saranno approfonditi qui.

E. Implementazione: Sampling-Importance-Resampling

Le ipotesi con le quali si opera sono semplici: basta avere a disposizione delle realizzazioni dell'evoluzione probabilistica del sistema (cioè delle realizzazioni della v.a. $p(x_k|x)$, la quale viene implementata tramite l'applicazione nel modello dello stato x_{k-1} e l'estrapolazione di una realizzazione di w_{k-1} , per il calcolo di x_k), una realizzazione di v_{k-1} (per ricavarsi la Likelihood), ed il modello di misura.

Partendo dall'algoritmo Sequential-Importance-Sampling, basta considerare $p(x_k|x_{k-1})$ come *Importance density*, ed eseguire il resampling ad ogni passo.

Grazie alla scelta mirata di $q(\cdot)$, riduciamo i pesi alla forma

$$w_k^i \propto w_{k-1}^i p(z_k|x_k^i), \quad (45)$$

e dopo il resampling:

$$w_k^i \propto p(z_k|x_k^i), \quad (46)$$

dato che abbiamo $w_{k-1}^i = 1/N \forall i$.

F. Il primo approccio di sensor fusion

Come primo approccio di fusione, consideriamo il caso semplice, e non ottimo, di ricondurre ad una sola misura z_k di varianza minore, quelle di partenza, e di applicare il particle filter nella sua forma base, il quale considera appunto una sola rilevazione.

Per fondere le varie misure candidate a disposizione, utilizziamo il principio introdotto in [26], in una versione più semplice, trascurando il bias e continuando ad utilizzare una sensitività uguale fra le misure su cui applicare il sensor fusion, considerando poi un semplice passo k e omettendone il pedice.

La modellizzazione della misura è effettuata come segue:

$$z_t = s_t \theta + \sigma w_t, \quad t = 1, \dots, Q \quad (47)$$

dove S , che lega la misura originale θ alla rilevata è un vettore $[1, 1, \dots, 1]^T$, che rappresenta le misure rilevate dalla wsn, dalla cam1, etc., e la varianza σ varia in funzione del sensore che rileva la misura.

La matrice che definisce la correlazione fra i rumori, essendo i rumori indipendenti è diagonale:

$$R = \begin{bmatrix} \sigma_{wsn}^2 & 0 & 0 & \dots \\ 0 & \sigma_{cam1}^2 & 0 & \dots \\ 0 & 0 & \sigma_{cam2}^2 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}. \quad (48)$$

Il migliore stimatore possibile è quindi dato dallo stimatore di Markov $\hat{\theta}(z) = [S'R^{-1}S]^{-1}S'R^{-1}z$.

La misura finale $z(n)$, ricavata dall'insieme delle Q misure z^i globali, è data perciò dalla somma pesata:

$$z(n) = \sum_{i=1}^Q W_i z^i, \quad (49)$$

dove il peso W_i può calcolato se la varianza (l'incertezza) delle misure in ingresso è noto.

Ipotizzando un insieme di rilevazioni ricavato dalla stessa misura, ma ognuna modellizzata come una gaussiana⁵ $\mathcal{N}(0, \Sigma_{z^n})$; si può ricavare la varianza di fusione:

$$\Sigma_z = \frac{\prod_{i=1}^Q \Sigma_{z^i}}{\sum_{i=1}^Q \left(\frac{\prod_{j=1}^Q (\Sigma_{z^j})}{\Sigma_{z^i}} \right)}, \quad (50)$$

e i pesi corrispondenti come:

$$W_i = \frac{\Sigma_z}{\Sigma_{z^i}}. \quad (51)$$

In appendice si riporta uno pseudoalgoritmo del filtro implementato.

⁵e quindi la stima è non fisheriana.

G. Il secondo approccio di sensor fusion

Il primo approccio risulta abbastanza semplicistico, e non sfrutta appieno le possibilità che la struttura del particle filter ha a disposizione.

Si può così incrementare la prestazione del particle filter cercando di esplicitare la relazione fra la struttura del modello e la natura delle informazioni nelle varie modalità di misura.

Si riprenda la situazione di partenza del fusion, assumendo di avere ancora a disposizione M misure, e di disporle in un vettore $z = (z^1, z^2, \dots, z^M)$.

Se si assume che le misure siano condizionatamente indipendenti fra di loro, dato lo stato, allora la Likelihood può essere fattorizzata come segue:

$$p(z|x) = \prod_{m=1}^M p(z^m|x). \quad (52)$$

Se questa ipotesi venisse introdotta nell'algoritmo precedente (apportando un prodotto di M Likelihood minori), porterebbe già dei leggeri miglioramenti, in quanto definirebbe meglio la partecipazione di ogni rilevazione, ma per ora si limiti a considerarla come il primo passo per il raggiungimento di una versione più elaborata.

Si introduce quindi una assunzione particolare: considerando lo stato al momento precedente x' e il momento stesso x , si supponga che l'evoluzione di stato e la proposal distribution possa essere espressa come segue:

$$p(x|x') = \int p_M(x|x^{M-1})p_{M-1}(x^{M-1}|x^{M-2})\dots \times p_1(x^1|x')dx^1dx^2\dots dx^{M-1}, \quad (53)$$

$$q_{M-1}(x^{M-1}|x^{M-2}, z^{M-1})q_1(x^1|x', z^1)dx^1dx^2\dots dx^{M-1}, \quad (54)$$

dove gli x^1, x^2, \dots, x^{M-1} sono dei vettori dello stato "ausiliari".

Si suppone cioè di spezzettare l'evoluzione originale del sistema in M passi successivi, e di calcolare quindi le pdf cercate cercando di riferirsi alla traiettoria fatta dall'evoluzione di stato tra un campione e l'altro.

Ciò può essere fatto, ad esempio, quando lo stato è M -dimensionale, e la modellizzazione dell'evoluzione con elevato numero di variabili di stato corrispondente sono indipendenti, oppure quando l'evoluzione del modello è gaussiana e lineare e può così essere facilmente frammentata in M passi successivi con varianza minore.

Se inoltre accettiamo l'approssimazione in cui la Likelihood dell' m -esima misura $p(z^m|x)$ viene fatta partecipare dopo aver applicato l' m -esimo stato del modello di

evoluzione $p_m(x^m|x^{m-1})$, possiamo ricavarci una nuova formulazione ricorsiva della distribuzione obiettivo, che prende la forma:

$$\pi^m(x^m) \propto$$

$$\int w_m(x^m, x^{m-1})q_m(x^m|x^{m-1}, z^m)\pi^{m-1}(x^{m-1})dx^{m-1} \quad (55)$$

con, $m=1, \dots, M$, e

$$w_m(x^m, x^{m-1}) = \frac{p(z^m|x^m)p(x^m|x^{m-1})}{q_m(x^m|x^{m-1}, z^m)} \quad (56)$$

dove π^0 e π^M sono rispettivamente la distribuzione di filtraggio precedente, e quella nuova, $x^0 = x'$ e $x^M = x$. Questa ricorsione può quindi essere approssimata con la strategia *Layered Sampling*, dove all' m -esimo livello vengono simulati nuovi campioni forniti da un'approssimazione Monte Carlo della distribuzione $q_m(x^m|x^{m-1}, z^m)\pi^{m-1}(x^{m-1})$, con associati dei pesi di importanza proporzionali a $w_m(x^m, x^{m-1})$, per mantenere una pesatura corretta dei campioni.

Così com'è illustrato, l'approccio *Layered Sampling* non sembra fornire dei vantaggi ovvii rispetto l'applicazione del particle filter iniziale; in realtà l'efficacia si fa notare quando le misure che apportano ulteriore informazione differiscono nella qualità della rilevazione dello stato stesso.

Se le misure vengono quindi ordinate in modo tale che la prima apportante informazione sia la più rozza (varianza massima), e l'ultima sia la più fine (varianza minima), questo approccio guiderà efficacemente la ricerca negli spazi di stato, e ad ogni livello rifinirà il risultato fornito dal livello precedente.

In qualche applicazione speciale, la Likelihood, e il modello dell'evoluzione di stato sono indipendenti rispetto ogni singolo componente degli spazi di stato:

$$p(z|x) = \prod_{m=1}^M p(z^m|x_m) \quad (57)$$

$$p(x|x') = \prod_{m=1}^M p(x_m|x'_m) \quad (58)$$

con $x = (x_1, x_2, \dots, x_M)$.

Per i modelli di questa natura, il *Layered Sampling*, con

$$p_m(x^m|x^{m-1}) = p(x_m^m|x_m^{m-1}) \prod_{k \neq m} \delta_{x_k^{m-1}}(x_k^m), \quad (59)$$

sostituito nella rispettivamente formulazione di $p(x|x')$ e di $w_m(x^m, x^{m-1})$ è esatto, ed è conosciuto come *Partitioned Sampling*.

Esso effettivamente rimpiazza la ricerca nello spazio di stato totale con una successione di M problemi di ricerca

più facili, ognuno in uno spazio dimensionale minore. A questo punto basterà seguire l'algoritmo descritto in appendice per avere un'idea dell'implementazione da seguire per la seconda versione del filtro particellare.

H. Applicazione del particle filter

Si considerino le condizioni di simulazione apportate nel paragrafo II-D: due sensori diversi con un'area in cui sono concorrenti ed un segnale sinusoidale da inseguire.

Il primo approccio di fusione riconduce le varie rilevazioni ad una sola, di varianza minore, da porre in ingresso al filtro particellare base (Sampling-importance-resampling).

Questo criterio centralizzato viene preferito in quanto il particle filter, a seconda del numero di particelle, risulta computazionalmente oneroso, e quindi poco proponibile alla basse prestazioni dei sensori.

L'inizializzazione viene eseguita seguendo un criterio di generalità: si dispongono le particelle iniziali seguendo la distribuzione di una normale con media la prima misura rilevata e con varianza tale che la variazione massima del segnale da inseguire sia a grandi linee all'interno del 95% dei valori possibili della normale di partenza (questo valore è approssimativo in quanto l'evoluzione delle particelle che permettono il tracking "raggiungono" la misura in un numero limitato di passi: maggiormente sono note le condizioni iniziali di misura, minore sarà il transitorio di varianza iniziale).

I pesi delle varie particelle possono quindi essere impostati a seconda della distanza dalla prima misura disponibile.

Al passo successivo, prima della rilevazione della seconda misura, le particelle evolvono secondo il modello lineare di appendice A-A, a seconda del parametro di variazione del modello impostato.

Vengono quindi associati i pesi alle varie particelle tramite il criterio della massima verosimiglianza, viene effettuato il resampling per evitare la degenerazione e viene iterato il tutto finché sono disponibili rilevazioni del movimento.

In figura 3(a) viene riportato il risultato dell'inseguimento, mentre si può notare in figura 3(b) l'evoluzione della varianza: nonostante il transitorio iniziale e gli ovvii picchi caratteristici dovuti dall'aleatorietà dell'evoluzione del modello, la varianza di inseguimento sembra rispecchiare una proporzionalità nei confronti della varianza di misura, e la varianza nei campioni in cui le rilevazioni sono concorrenti, è minore.

Devono essere però segnalati alcuni problemi:

- 1) L'aleatorietà della media della varianza d'inseguimento non è ben definita in quanto si basa su di un numero limitato di campioni, per cui si è deciso di riportare nelle figure i casi tipici, anche se possono esserci casi in cui la media della varianza di inseguimento con più misure disponibili può essere maggiore di quelle con misure singole.
- 2) La taratura del parametro di evoluzione "probabilistica" del sistema dev'essere fatta euristicamente a seconda della velocità massima con la quale potrebbe variare il segnale: un valore insufficiente non permetterebbe l'inseguimento, e la stima diverrebbe inutilizzabile, mentre un valore troppo alto ne ridurrebbe troppo la precisione, se in presenza di un basso numero di particelle.
- 3) Il numero di particelle deve essere sufficientemente grande da permettere che almeno una particella, a seconda dell'evoluzione, cada in prossimità della misura successiva di quanto lo pretende la tolleranza di inseguimento; in caso contrario il particle filter sarebbe computazionalmente pesante.

Si nota poi che, dalla teoria, non sarebbe corretto utilizzare un particle filter in presenza di un modello lineare, dato che il filtro di Kalman è la soluzione ottima da utilizzare in queste applicazioni.

Tuttavia da un particle filter ben tarato, come da figura, può risultare un inseguimento migliore in termini di varianza, mentre se si ricerca un comportamento più omogeneo è utile ricorrere al filtro di Kalman.

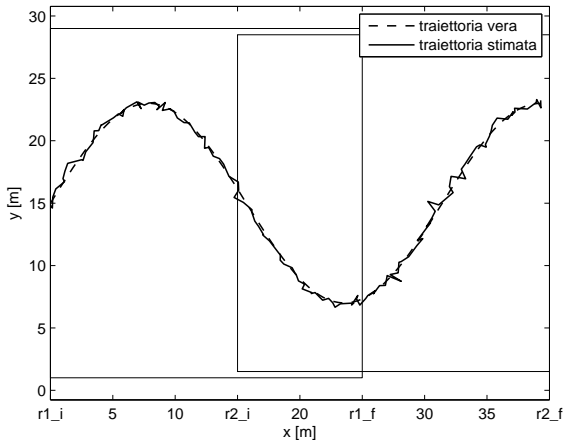
Nella seconda serie di simulazioni, non consideriamo più la fusione precedente l'applicazione del particle filter, ma immettiamo le misure nell'algoritmo direttamente, rinunciando quindi all'accento di decentralizzazione del tracking.

Consideriamo poi la possibilità di eseguire la simulazione basandosi sia su di un modello lineare, sia su di un modello non lineare (vedi appendice A-B), giustificando così l'impiego di un particle filter in un framework dove il filtro di Kalman, così com'è, non può essere utilizzato (si utilizzi il filtro di Kalman esteso).

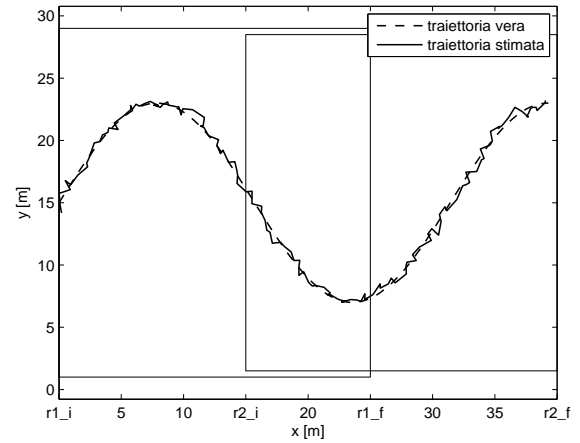
L'utilizzo di un modello non lineare permette di rappresentare in maniera diversa il movimento dell'inseguito, La struttura riportata nella secondo criterio di fusione, si ricorda, esegue n passi elaborativi intermedi fra un istante di campionamento e l'altro, permettendo di "fondere" n misure.

L'inizializzazione viene effettuata quindi nella stessa maniera, nella massima generalità, e l'esecuzione attuata allo stesso modo.

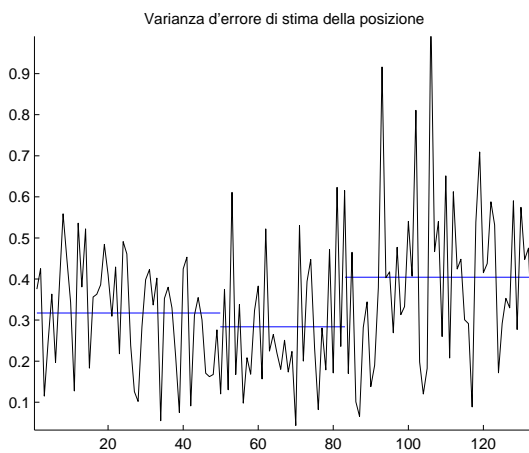
Il risultato delle simulazioni con il secondo criterio di



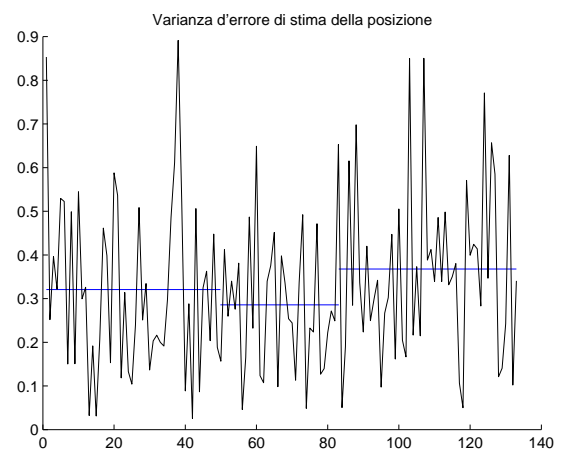
(a) Traiettoria reale e stimata del veicolo mobile nell'ambiente monitorato, nel caso di Particle Filter con implementato il primo approccio di fusione.



(a) Traiettoria reale e stimata del veicolo mobile nell'ambiente monitorato da due sensori nel caso del Particle Filter Layered Sampling, con modello di evoluzione del sistema lineare.



(b) Traiettoria reale e stimata del veicolo mobile nell'ambiente monitorato, nel caso di Particle Filter con implementato il primo approccio di fusione.



(b) Andamento temporale della varianza d'errore, per ogni istante di campionamento, nel caso del Particle Filter Layered Sampling, con modello di evoluzione del sistema lineare. Vengono inoltre messe in rilievo le medie negli istanti temporali in cui si verifica la diversa disponibilità delle misure.

Fig. 3. Primo approccio di fusione.

Fig. 4. secondo approccio di fusione, modello lineare.

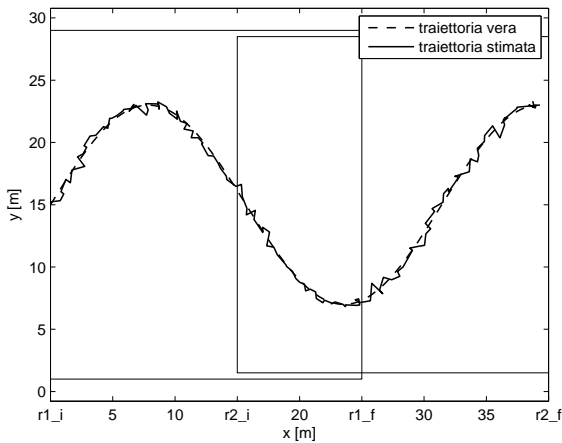
fusione non differiscono molto a seconda del diverso modello implementato, risultando qualitativamente equivalenti, ma non a livello computazionale.

IV. ARCHITETTURA DEL SISTEMA

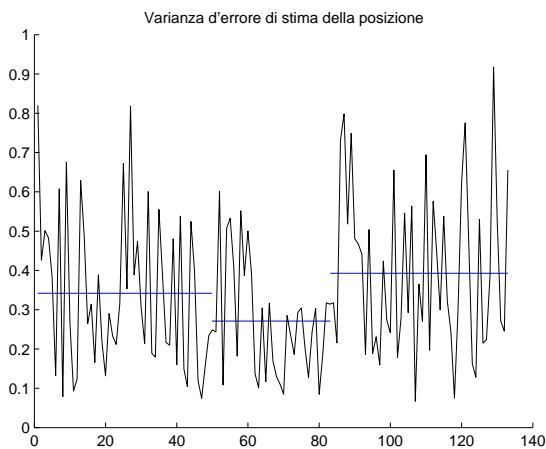
Se nelle sezioni precedenti sono stati introdotti i concetti matematici che sono alla base del procedimento di data-fusion, nella sezione attuale si vuole introdurre l'architettura funzionale del sistema di tracking. Tale struttura è pensata sia per permettere l'effettiva implementazione dei precedenti algoritmi di data-fusion sia per soddisfare gli obiettivi di decentralizzazione e modularità che ci si è posti all'inizio del lavoro.

Come già accennato nella sezione introduttiva, nello scenario preso in considerazione i sensori a disposizione

sono dati da una rete di $N - 1$ videocamere (ognuna delle quali viene assunta essere un sensore) e da una rete di dispositivi in radiofrequenza (considerati nell'insieme come un unico sensore che si conviene di indicare con l'acronimo WSN). Nella realtà pratica la rete WSN sfrutta il sistema TESEO e, pertanto, si assume di avere a disposizione un sistema a scatola chiusa del quale sia solo possibile acquisire i dati prodotti assimilandoli a misure effettuate dal sensore WSN. comunque, il lettore interessato può trovare maggiori dettagli riguardanti il funzionamento della rete WSN ed il sistema TESEO in [6], [17]. Si sottolinea, comunque, che nel contesto sperimentale reale l'estensione della WSN su tutta l'area



(a) Traiettoria reale e stimata del veicolo mobile nell'ambiente monitorato da due sensori nel caso del Particle Filter Layered Sampling, con modello di evoluzione del sistema non lineare.

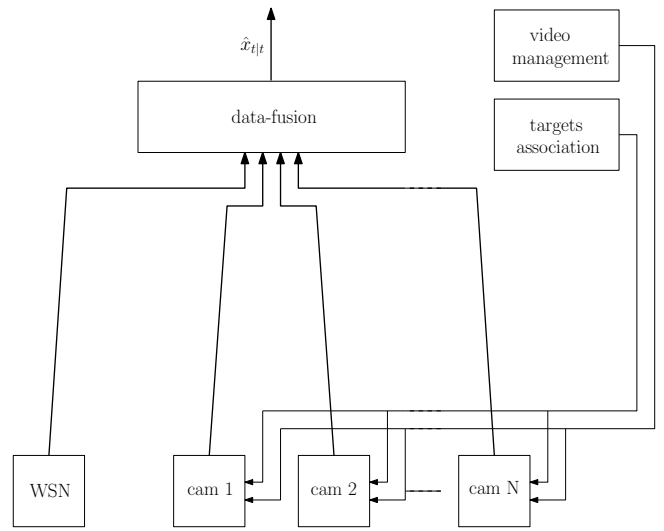


(b) Andamento temporale della varianza d'errore, per ogni istante di campionamento, nel caso del Particle Filter Layered Sampling, con modello di evoluzione del sistema non lineare. Vengono inoltre messe in rilievo le medie negli istanti temporali in cui si verifica la diversa disponibilità delle misure.

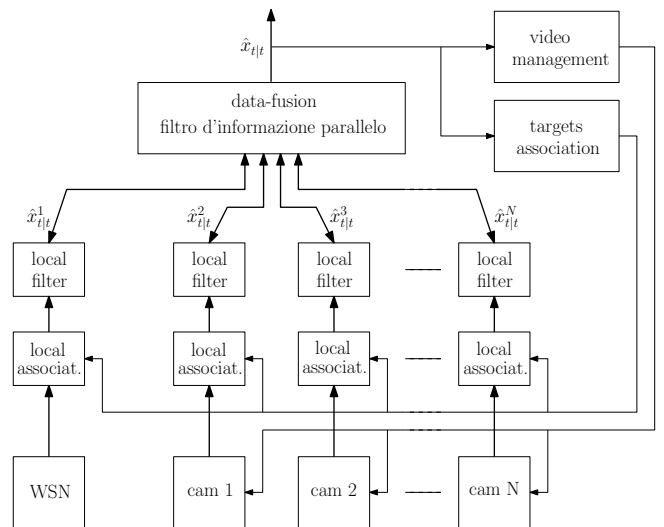
Fig. 5. secondo approccio di fusione, modello non lineare..

sottoposta al controllo fa sì che questa diventi un indicatore fondamentale per il rilevamento della posizione dei targets nel caso di aree occluse al campo visivo della rete di videocamere e, inoltre, permette di non perdere mai il *labeling* relativo ai targets inseguiti, il tutto anche a costo di una precisione inferiore nel tracking. Per quanto riguarda la rete di videocamere, invece, il sistema di misura e gestione della rete è stato appositamente realizzato in tutte le sue parti, e nella Sezione V se ne forniranno tutti i dettagli implementativi.

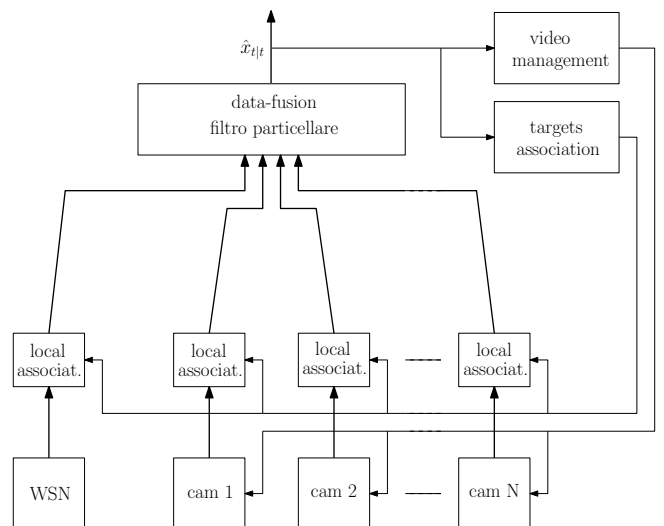
Il funzionamento del sistema in modalità multitasking è gestito da uno specifico blocco il cui funzionamento verrà descritto in modo approfondito, relativamente alla



(a) Struttura base del sistema.



(b) Struttura distribuita con filtro d'informazione parallelo.



(c) Struttura centralizzata con filtro particellare.

Fig. 6. Varianti dell'architettura del sistema di tracking multi-sensore realizzato. Le frecce più spesse indicano il flusso dati relativo al tracking, le frecce più fine indicano i segnali di controllo del sistema.

rete di videocamere, nella Sezione VI. Nella rete WSN, invece, la modalità multitracking è gestibile in modo molto più semplice grazie al già citato sistema TESEO.

Concettualmente, quindi, l'architettura base del sistema è riconducibile a quella di Figura 6(a), la quale viene poi estesa e particolareggiata in funzione delle specifiche potenzialità di cui si vuole dotare il sistema di tracking e dell'approccio scelto per la realizzazione dell'unità di data-fusion.

Dunque, volendo realizzare l'unità di data-fusion secondo l'approccio di Kalman e volendo implementare il sistema in modo distribuito, si deve particolareggiare l'architettura base appena introdotta dotando ogni sensore di un filtro d'informazione che provveda all'attività di tracking locale e implementando a livello dell'unità di data-fusion l'algoritmo definito nella Sezione II-B. La struttura che ne risulta è quella riportata in Figura 6(b). Nell'architettura che si ottiene, inoltre, se la comunicazione tra l'unità centrale di data-fusion e i filtri locali è bidirezionale, è possibile usare la stima ottima calcolata dall'unità di data-fusion per inizializzare lo stadio di predizione dei filtri locali introducendo, di fatto, una sorta di feedback nel sistema.

Realizzando l'unità di data-fusion secondo l'approccio del filtro particellare, invece, si fa riferimento ad una struttura riconducibile a quella di Figura 6(c). Il sistema non prevede l'utilizzo di filtri decentralizzati a livello dei singoli sensori perchè, pur essendo essi dotati di capacità di calcolo, raramente essa è sufficiente a garantire l'implementazione di un filtro particellare la cui complessità computazionale è notoriamente elevata. In ogni caso, non è esclusa la possibilità di realizzare un sistema ibrido caratterizzato da un'unità di data-fusion basata su filtri particellari interfacciata ad un sistema distribuito dotato di filtri d'informazione locali come implementato, ad esempio, in [28].

V. RETE DI VIDEOCAMERE

L'utilizzo di una rete di videocamere porta ad ovvi vantaggi rispetto al tracking con un'unica camera. Basti considerare l'estensione del campo visivo offerto dalla rete, che permette di controllare nello stesso tempo anche ambienti molto estesi considerandoli come un tutt'uno. La disponibilità di molteplici punti di vista consente inoltre di migliorare la robustezza del tracking in quanto zone parzialmente occluse in alcune camere possono risultare perfettamente visibili da altri punti di vista. Il problema delle occlusioni diviene molto rilevante soprattutto nei sistemi di videosorveglianza (considerabili come un'estensione del caso considerato) nei quali vi è il bisogno di inseguire più oggetti simultaneamente

mantenendo l'identità di ciascuno e considerando che, in questi sistemi, le occlusioni sono causate dagli stessi oggetti tracciati. In letteratura sono state proposte diverse soluzioni per risolvere il problema succitato, tra le quali l'utilizzo dell'omografia dell'immagine acquisita risulta essere il metodo più robusto, si vedano per riferimento [27], [11].

Nello specifico in questa sezione verranno trattati la struttura ed il funzionamento della rete di telecamere.

A. Virtualizzazione

Nella fase sperimentale si è avuta a disposizione una sola camera, di conseguenza la rete di telecamere, necessaria a valutare i risultati degli algoritmi proposti, è stata virtualizzata partendo dall'unica vista disponibile. Si considera ciascuna camera (virtuale) posizionata sul soffitto con l'asse ottico ortogonale al pavimento. Questa scelta non risulta essere vincolante ai fini implementativi in quanto, pur partendo da una serie di viste anche prospettiche è sempre possibile effettuare una omografia delle immagini e riportarsi nelle condizioni succitate. La scelta di lavorare su di un'immagine rettificata oltre a rendere il sistema ininfluente alle diverse angolazioni di ripresa delle camere, minimizza anche gli errori legati alla non perfetta conoscenza della loro orientazione. La posizione di ciascuna camera rispetto all'ambiente circostante pu essere calcolata utilizzando l'algoritmo presentato in [12] che prevede un metodo automatico per l'autolocalizzazione delle camere con viste sovrapposte utilizzando tutte le viste simultaneamente. Utilizzando l'algoritmo in ambienti aperti la notevole distanza che può esserci tra gli oggetti inseguiti e la camera non fa che aumentare l'incidenza degli errori legati alla stima dei parametri intrinseci, rendendo inopportuno il ricorso a metodi rigorosamente geometrici per la proiezione. Si ritiene quindi che una precalibrazione della camera, atta all'individuazione dell'opportuna matrice di proiezione prospettica tramite l'individuazione del piano del moto e la sua successiva proiezione sul piano dell'immagine, semplifichi le operazioni di localizzazione a vantaggio di una maggior robustezza, si veda a tal fine [27] che mostra un metodo robusto per l'identificazione della matrice di proiezione utilizzando il frame iniziale.

Nella fase di inizializzazione del programma è fornita la possibilità di scegliere il numero di camere e impostare i relativi campi visivi che saranno utilizzati ad ogni iterazione per individuare il frame relativo a ciascuna camera. Al fine di rendere più realistiche le condizioni simulate ciascuna camera fornisce le misure con un rumore gaussiano bianco con una varianza dell'ordine di 4 pixel, ritenuta sufficiente a simulare gli eventuali

errori di calibrazione e localizzazione considerando la ridotta estensione dell'intera area di lavoro.

Si ricorda, per completezza, che l'omografia è una trasformazione lineare⁶ non singolare del piano proiettivo in se stesso. La sua realizzazione prevede la conoscenza dei parametri intrinseci ed estrinseci della camera [13], alternativamente la matrice di proiezione H può essere calcolata utilizzando un set di corrispondenze tra i due piani [27] che utilizza quattro coppie di punti non collineari.

B. Detection

La fase di detection rappresenta l'estrapolazione dal frame acquisito della misura della posizione del robot. Per ottimizzare le risorse (anche energetiche) le camere sono gestite da un sistema centrale di coordinamento che abilita ciascuna camera al funzionamento solo se il robot può transitare nel suo campo visivo. Per velocizzare l'algoritmo si individuano dapprima le zone soggette al motion, tramite un'operazione di *background subtraction* e *thresholding*, e solo su queste zone si ricercano le features rappresentanti i robots. Esistono varie tecniche di background subtraction di cui una panoramica viene mostrata in [19]. Nel nostro caso la fase di learning del background è molto semplice. Non essendoci la possibilità che le condizioni della pedana possano cambiare durante la prova, si inizializza il background prima dell'inizio di ogni esperienza con l'immagine della pedana senza i robot e si fa riferimento a quest'immagine per individuare le zone di motion. Questo espediente permette di rendere l'algoritmo invariante alle diverse condizioni ambientali di ciascuna prova⁷, che possono provocare differenze di luminosità sulla pedana legate a riflessi del sole o delle luci. Nell'applicazione sviluppata l'individuazione di ciascun robot è stata ottenuta utilizzando come feature due led luminosi posti sopra il robot. Questa scelta permette di ottenere, nel caso specifico, una misura della posizione più precisa rispetto a quella che si otterrebbe considerando l'intera immagine del robot. Una giustificazione a questa considerazione viene dal fatto che se si fosse considerata l'immagine completa del robot, in fase di motion detection sarebbero state selezionate anche le ombre che lo stesso proietta sul pavimento, di conseguenza l'individuazione della posizione sarebbe stata falsata da questi fenomeni e sarebbe risultata più imprecisa. Per enfatizzare la feature

⁶La linearità è garantita descrivendo il piano in coordinate proiettive (omogenee), risulta invece non lineare se si utilizzano le coordinate cartesiane.

⁷Si suppone che le condizioni ambientali restino pressoché invariate durante la prova data la sua esigua durata.

cercata ciascun frame è stato acquisito ad alto contrasto e bassa luminosità. Dopo la fase di motion si è optato per una sogliaatura, al fine di escludere dal processo di feature extraction eventuali rumori presenti nell'immagine. Successivamente, utilizzando funzioni native di Matlab⁸ si raggruppano tutti i punti interessati dal motion con distanza tra loro e numerosità tali da essere classificabili come led. Il passo successivo è quello di calcolare il baricentro dei blob ottenuti, lo scopo si raggiunge pesando la posizione di ogni singolo pixel con la sua luminosità al fine di ottenere una osservazione più precisa della posizione. Grazie agli accorgimenti messi in atto si è riusciti ad avere un errore nella misura dei singoli punti di interesse dell'ordine del sub-pixel in riferimento alla sola estrazione delle feature dall'immagine acquisita.

VI. ASSOCIATION

Il punto cruciale dei problemi di tracking multitarget è quello del processo di associazione dell'osservazione al tracking, raccolte le osservazioni vengono associate alla relativa traccia attraverso un algoritmo di data association. Dopo una breve panoramica su alcune tecniche principalmente utilizzate verrà spiegato l'algoritmo realizzato e le motivazioni che hanno portato a tale scelta. La regola del *Nearest Neighbors* è forse l'approccio più semplice per determinare quale sia l'oggetto tracciato relativo ad una data osservazione dei sensori [14]. Quando si riceve una nuova osservazione, tutte le tracce esistenti vengono proiettate all'istante di tempo della nuova misura ottenuta. Quindi, viene calcolata la distanza tra la posizione osservata e ciascuna delle posizioni stimate di ogni traccia. L'osservazione viene quindi associata alla traccia più vicina. In questo modo un problema di tracciamento multiplo può essere ridotto ad un insieme di problemi di tracciamento singolo. Un approccio più sofisticato allo *splitting* è chiamato *multiplehypothesis tracking*, che mantiene una storia delle ramificazioni delle tracce, così che ogni qualvolta un ramo viene confermato, tutti i rami alternativi possono essere scartati [14]. Secondo questo schema, il sistema di tracciamento non deve necessariamente assegnare l'osservazione ad una singola traccia, ma se l'osservazione è altamente correlata a più tracce può essere creata una copia aggiornata di tali tracce; con le successive osservazioni la traccia corretta convergerà con la reale traiettoria del target, mentre le altre tracce saranno sempre meno correlate alle osservazioni successive. Questa tecnica è chiamata *track splitting* [4]. Le soluzioni mostrate prevedono tutte l'utilizzo di un ulteriore filtro per l'associazione che deve

⁸Le funzioni native di Matlab garantiscono una maggiore velocità di esecuzione, e pertanto sono da preferire.

essere attuata in maniera centralizzata, questo tipo di approccio diviene però tanto più oneroso computazionalmente quanto più il sistema si espande. Per ovviare a questo inconveniente è stato studiato un algoritmo decentralizzato per l'associazione delle osservazioni basato su un approccio di tipo *Nearest Neighbors*, che prevede l'ausilio di un funzionale di costo per sopperire alle errate associazioni nel caso di robot molto vicini.

A. Algoritmo utilizzato

Nel sistema analizzato il problema dell'associazione investe principalmente la rete delle camere in quanto per la WSN il problema si è risolto utilizzando canali di trasmissione differenti per ciascun sensore.

La soluzione adottata per questo problema è stata quella di realizzare un sistema di associazione decentralizzato, utilizzando un approccio di tipo *Nearest Neighbors* la cui riduzione a singoli problemi di tracciamento si presta meglio alla decentralizzazione. Un problema della regola del *nearest-neighbors* è dovuta alle assegnazioni errate delle osservazioni alle tracce, come succede inevitabilmente se gli oggetti tracciati sono vicini. L'introduzione di un funzionale di costo che migliorasse l'associazione nel caso di robot vicini, ha permesso di migliorarne notevolmente le prestazioni. A tale scopo si è attribuito un maggior peso alla distanza dei robot, mentre la direzione⁹ diviene rilevante quando i robot sono molto vicini.

$$F = distance^{exp_func} + \frac{\Delta direction}{fract_func}$$

Rispetto al caso generale, nel caso specifico non si dispone di una informazione univoca relativa ai robot, poiché esistono differenti interpretazioni delle feature estratte. Per ciascun robot vi sono infatti due possibilità:

- entrambi i led sono visibili.
- solo un led è visibile.

Per ogni possibile robot si calcola l'associazione che minimizza il funzionale rispetto a quelli precedentemente assegnati, per fare ciò è necessario che ad ogni sensore sia retroazionata la stima dei robot come mostrato, per esempio, in Figura 6(b).

La scelta di considerare anche i robot con un solo led visibile è stata fatta per sfruttare tutta l'informazione disponibile, e permette di localizzare i robot sia all'ingresso del campo visivo delle camere che nelle posizioni più esterne della pedana, da cui i led paiono poco luminosi e difficilmente contemporaneamente visibili. Così facendo è garantita anche la generalità a

⁹Per direzione si intende la differenza di direzione tra quella del robot localizzato tramite fusione e quella del robot da associare.

cui si è fatto riferimento precedentemente, che prevede la possibilità di avere occlusioni in qualsiasi punto del campo visivo dell'immagine.

Vediamo nel dettaglio come funziona l'algoritmo nel caso in cui più led si trovano contemporaneamente ad una distanza tale da poter essere considerati come robot e come viene risolto il problema legato alla mancata conoscenza del numero esatto di robot da trovare nel frame.

Si consideri l'esempio semplificato mostrato in Figura 7, è evidente che potremmo trovarci in uno qualsiasi dei casi mostrati. Per far fronte a questo problema si utilizza una matrice delle adiacenze con tante righe e colonne quanti i led trovati, in cui ciascuna coppia di punti¹⁰ è inizializzata con un peso pari alla direzione del robot individuato dall'adiacenza. Per ogni riga della matrice delle adiacenze si calcola l'indice di costo tra tutte le possibili associazioni e i robot trovati nella fusione. L'associazione di uno o due led che minimizza il funzionale è classificata come robot e viene associato al suo corrispondente. Alle successive adiacenze è preclusa la possibilità di essere associate a robot già vincolati, in tal caso sarà considerato il successivo minimo. Non considerando contemporaneamente tutte le possibili adiacenze di tutti i led, la minimizzazione del funzionale che ne deriva rappresenta un minimo locale, che rappresenterebbe un problema se ad ogni iterazione i robot assumessero, posizioni molto differenti rispetto alle precedenti. Sperimentalmente si è verificato che la suddetta condizione non è mai verificata poiché il tempo necessario all'esecuzione di un passo di campionamento (0.2 secondi¹¹) e la velocità dei robot fanno sì che le assunzioni fatte risultino valide garantendo nel complesso buoni risultati.

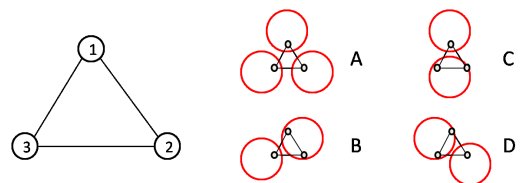


Fig. 7. Associazioni possibili tra tre led.

¹⁰Vengono considerati solo i punti della matrice delle adiacenze che hanno una distanza compatibile con le dimensioni del robot.

¹¹Tempo misurato considerando l'esecuzione completa dell'algoritmo utilizzando come metodo di fusione l'algoritmo proposto nella Sezione II-B, 11 sensori e 4 robot (0.1 secondi senza fondere le informazioni). Si noti che, a causa dell'hardware a disposizione, i sensori non vengono chiamati in parallelo ma in serie.

B. Confronto Algoritmo standard con algoritmo modificato

Si presentano adesso la successione di alcuni frame esplicativi dell'importanza dell'informazione relativa alla direzione nell'algoritmo di associazione proposto. La sequenza proposta in Figura 8 mostra lo scontro di due robot, condizione molto critica a causa delle variazioni improvvise di direzione e velocit. In verde e azzurro sono evidenziate le tracce vere dei due robot, i cerchi rappresentano le stime di posizione in funzione della sola distanza, mentre le croci rappresentano ad ogni iterazione la stima con l'algoritmo proposto. L'esperienza è stata fatta con un tempo di campionamento di 0.3 secondi.

Nello specifico caso limite si è visto che aumentando il tempo di campionamento la distanza tra i due frame successivi diventa rilevante, in condizioni meno critiche il tempo di campionamento può aumentare senza che questo pregiudichi il risultato dell'associazione ma considerando questo come caso peggiore si ritiene che una buona associazione per il sistema in esame possa essere garantita con il tempo di campionamento limite ricavato.

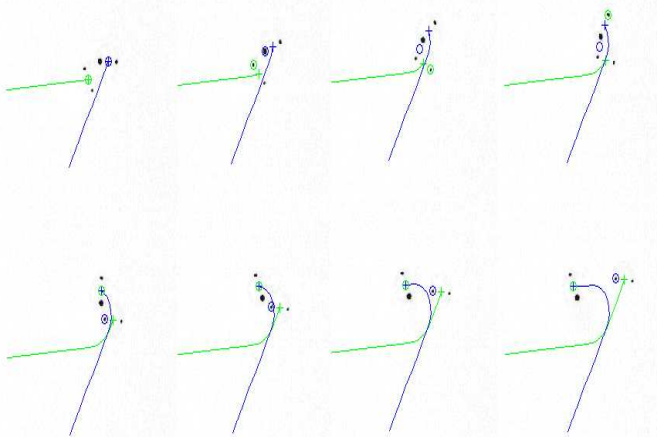


Fig. 8. In verde e blu le traiettorie vere del tracking

VII. SPERIMENTAZIONE

In questa sezione si illustrano i principali risultati ottenuti, permettendo un confronto tra i metodi di sensor fusion proposti. Il testing è stato effettuato utilizzando la strumentazione del laboratorio NAVLAB ed eseguendo tutti gli algoritmi del sistema di tracking mediante il software Matlab versione 2007b su un elaboratore con potenza di calcolo di 2GHZ: è importante specificare questo dato in quanto, essendo la rete delle camere simulata, di fatto l'elaborazione attuata da ogni smart sensor grava completamente sul computer in uso. Gli oggetti del tracking sono rappresentati dagli *e-puck*: robot

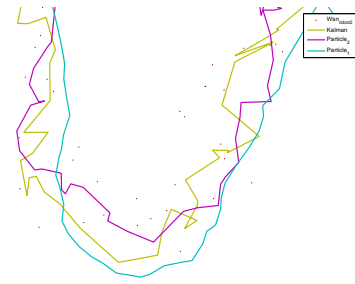


Fig. 9. Particolare di una traiettoria di tracking basata solo su misure provenienti dalla rete WSN.

interfacciabili tramite bluetooth e collegabili con un *T-mote Sky* alla rete wireless. L'algoritmo di localizzazione WSN utilizza il sistema *TESEO*, che sfrutta una rete di trenta sensori *T-mote Sky* e un sistema di acquisizione ed elaborazione implementato in linguaggio *Java*.

Si delineano quindi, i principali punti sui quali bisogna porre attenzione nella realizzazione dei test sperimentali:

- Le variabili aleatorie che rappresentano il rumore di misura possono essere simili a gaussiane, come nel caso delle videocamere, o completamente differenti, come bimodali o uniformi, o con media non necessariamente nulla, nel caso della WSN.
- La presenza di alcuni punti corrispondenti a misure vistosamente falsate (outlier).
- Il sistema operativo del computer utilizzato non è real-time, questo fa sì che durante l'esperienza alcuni frame possano essere persi o non sincronizzati pur avendo opportunamente tarato il passo di campionamento.

Prima dell'esecuzione, i filtri d'informazione cosiccome il filtro particellare, necessitano di essere accordati sul sistema reale. Per quel che riguarda i filtri d'informazione (sia quelli locali che quello di data-fusion) la taratura si riduce sostanzialmente alla scelta delle matrici di varianza relative al rumore di modello e di misura che sono deducibili da considerazioni pratiche o dai dati caratteristici dei sensori. In particolare, tutti i filtri locali vengono tarati con un significativo rumore di modello, mentre il rumore di misura delle videocamere risulta essere molto inferiore rispetto a quello caratteristico della rete WSN. Per quanto riguarda la taratura del filtro particellare, invece, a seconda di quale delle due implementazioni venga considerata, il numero di particelle e il parametro di varianza dell'evoluzione probabilistica del sistema vengono scelti euristicamente in modo tale da poter garantire l'inseguimento anche in presenza di salti dovuti ad un'erronea acquisizione delle

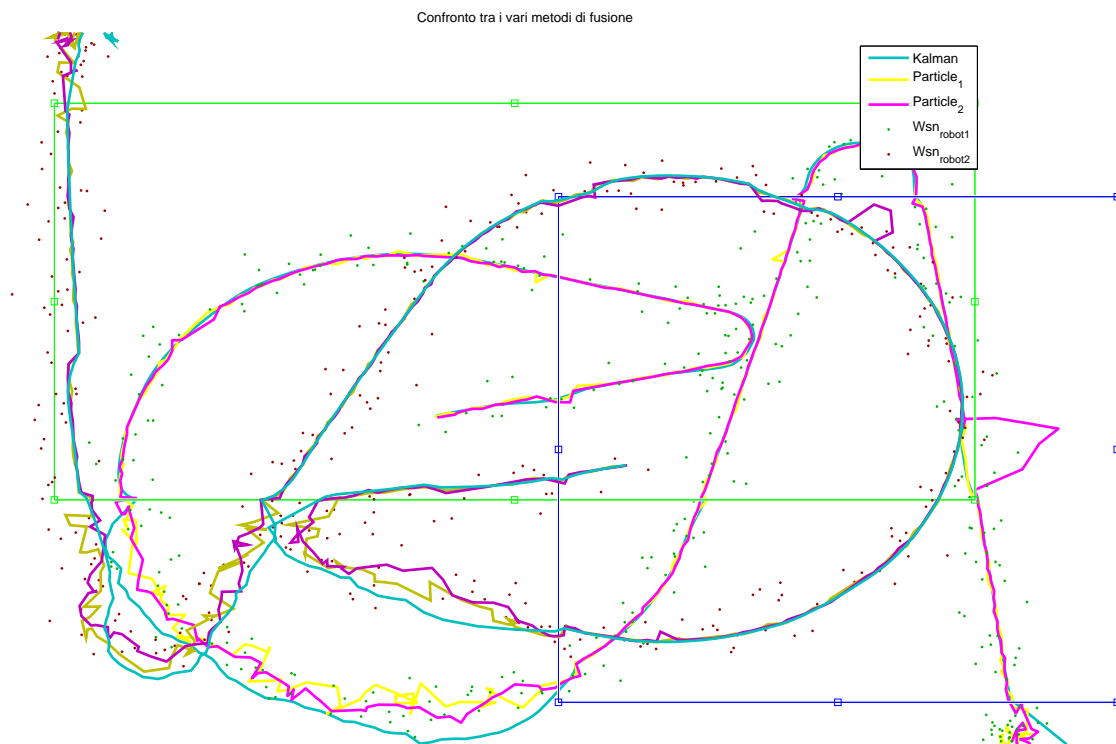


Fig. 10. Tracking ottenuto con le diverse tecniche di Sensor Fusion in uno scenario caratterizzato da due videocamere a copertura parziale dell'ambiente e da rete WSN a copertura totale. Due target con punto di partenza del moto rispettivamente nell'angolo superiore sinistro ed inferiore destro dell'ambiente.

misure.

Nella Figura 10 sono illustrati i risultati ottenuti in un semplice scenario reale nel quale il tracking di due veicoli mobili viene eseguito dal sistema basato sul filtro d'informazione parallelo o sul filtro particellare nelle sue due varianti. Quando il target è nel campo visivo di almeno una videocamera il sistema dispone di misure molto precise ed è evidente che le prestazioni dei due approcci sono sostanzialmente equivalenti. Quando le uniche misure disponibili provengono dalla rete WSN, invece, il comportamento dell'unità di data-fusion basata sul filtro particellare peggiora drasticamente, come si vede nella Figura 9 che illustra un particolare di una traiettoria eseguita. Ciò è sintomatico della difficoltà che si riscontra nel tarare il filtro particellare nel quale non si ha una conoscenza precisa del legame esistente tra i parametri di taratura ed i corrispondenti risvolti funzionali.

In ogni caso, a prescindere dalla strategia di data-fusion considerata, la differente prestazione del tracking in presenza o assenza delle misure rilevate dalle videocamere è indicativa della capacità degli algoritmi di pesare in modo diverso e adattativo le misure più o meno precise ricevute.

VIII. CONCLUSIONI

In questo articolo è stato mostrato come sia possibile realizzare un sistema di tracking multi-sensore nel quale si utilizzano tecniche di Sensor Fusion per la gestione e l'elaborazione dell'informazione fornita da sensori eterogenei quali una rete di videocamere ed una rete di sensori in radiofrequenza.

Per la realizzazione dell'unità di data-fusion, fulcro del sistema multi-sensore, sono stati proposti ed analizzati i due approcci più estesamente trattati nella letteratura specifica: il *Filtro d'Informazione* ed il *Filtro Particellare*. In particolare, si propone una formulazione in parallelo del Filtro d'Informazione, la quale è particolarmente adatta all'implementazione di sistemi ad architettura distribuita. Per il Filtro Particellare, invece, si propone una formulazione orientata alla realizzazione di sistemi di tracking ad architettura centralizzata, motivando tale scelta con l'onere computazionale richiesto da tale filtro che, pertanto, poco si adatta ad essere implementato all'interno di sensori che tipicamente non dispongono di grande capacità di calcolo.

La gestione ed il coordinamento dei sensori sono operazioni fondamentali nella realizzazione di sistemi multi-sensore e le tecniche da adottare a tal scopo dipendono

evidentemente dalla tipologia di sensori utilizzati e dalla struttura del sistema. Per la gestione delle videocamere viene proposto un semplice algoritmo che sfrutta la stima elaborata dall'unità di data-fusion sia per l'attivazione delle videocamere stesse, sia per l'inizializzazione dei filtri locali implementati su di esse.

Il funzionamento in modalità multi-tracking è stato realizzato studiando un algoritmo decentralizzato per l'associazione delle osservazioni basato su un approccio di tipo *Nearest Neighbors*, che prevede l'ausilio di un funzionale di costo per sopperire alle errate associazioni che possono manifestarsi quando i robots sono molto vicini tra loro.

Nel complesso, grazie all'utilizzo delle tecniche di Sensor Fusion e data-association, il sistema realizzato e testato in laboratorio è caratterizzato da un buon grado di affidabilità e robustezza. L'odierna disponibilità di svariati sensori a basso costo associata alle tecniche di filtraggio descritte in questo lavoro, dunque, permettono di suggerire l'impiego delle tecniche di *Sensor Fusion* per una grande gamma di applicazioni pratiche.

APPENDIX A

MODELLO DI STATO DEL TARGET

A. Modello dinamico lineare

La dinamica dell'oggetto su cui eseguire il tracking viene modellata secondo un modello di stato discreto. Nella rappresentazione utilizzata, il vettore di stato $x = (p_x, v_x, p_y, v_y)^T \in \mathbb{R}^4$, è costituito dalle componenti di posizione (p_x, p_y) e velocità istantanee (v_x, v_y) del target. Gli ingressi del modello sono assimilabili a forze generalizzate agenti in modo indipendente sulle due direzioni del moto e, trattandosi di grandezze incognite, vengono modellate come variabili aleatorie gaussiane.

La dinamica discretizzata (tempo di campionamento T), quindi, è data da:

$$\begin{cases} x_{k+1} = Ax_k + w_k & w_k \sim \mathcal{N}(0, Q) \\ y_k = Cx_k + v_k & v_k \sim \mathcal{N}(0, R) \end{cases}$$

Dove w_k ingloba sia il rumore di modello sia gli ingressi del sistema; v_k rappresenta l'errore di misura e le matrici indicate sono date da:

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = qT \begin{bmatrix} \frac{T^3}{4} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{4} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$R = r \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \end{bmatrix}$$

e i coefficienti q ed r sono dei semplici coefficienti di taratura da impostare a seconda dei singoli casi.

B. Modello dinamico non lineare

Il modello non lineare con il quale approssimiamo il movimento del robot ha un vettore di stato del tipo $\mathbf{x}(t) = \{x(t), v(t), y(t), \theta(t)\}$ e le equazioni di stato:

$$\begin{aligned} x_{t+1} &= v(t)\cos\theta(t) + N_x \\ v_{t+1} &= v(t) + N_v \\ y_{t+1} &= v(t)\sin\theta(t) + N_y \\ \theta_{t+1} &= \theta(t) + N_\theta, \end{aligned}$$

dove N_x, N_y rappresentano i rumori di modello, e N_v, N_θ rappresentano gli ingressi, modellizzati come rumori gaussiani, tutti con media nulla e con varianza

caratteristica.

Con questo modello, possiamo andare ad influenzare direttamente l'angolo e il modulo della velocità, e i parametri della normale relativi ad essi descriveranno l'evoluzione del movimento in un passo.

L'uscita del modello è descritta dalle equazioni lineari:

$$Z(t) = HX(t) + N_m$$

dove H è:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

e N_m è il vettore dei rumori di misura.

APPENDIX B

DERIVAZIONE DEL FILTRO D'INFORMAZIONE

La trattazione completa della teoria di Kalman esula chiaramente dagli scopi attuali e si rimanda per approfondimenti ai testi specifici [2], [20]; qui ci si limita a dare una semplice introduzione formale alle formule che costituiscono l'algoritmo del filtro di Kalman per usarle come punto di partenza nella deduzione del Filtro d'Informazione.

Si consideri il modello lineare stocastico tempo-invariante:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = Cx_k + v_k \end{cases}$$

in cui il rumore di modello w_k ed il rumore di misura v_k sono variabili aleatorie con distribuzione rispettivamente:

$$w_k \sim \mathcal{N}(0, Q) \quad v_k \sim \mathcal{N}(0, R)$$

e soddisfano le seguenti proprietà:

$$\begin{aligned} \mathbb{E}[w_k w_h] &= Q\delta(k-h) \\ \mathbb{E}[v_k v_h] &= R\delta(k-h) \\ \mathbb{E}[v_k w_h] &= 0 \quad \forall k, h \end{aligned}$$

Il filtro di Kalman è definito formalmente come la stima:

$$\hat{x}_{k|k} = \mathbb{E}[x_k | y_0, \dots, y_k]$$

alla quale viene associata la varianza d'errore di stima:

$$P_{k|k} = \mathbb{E}\left[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T | y_0, \dots, y_k\right]$$

Si dimostra che, per il modello di stato precedente e sotto opportune ipotesi, tale espressione può essere valutata ricorsivamente mediante un algoritmo in due stadi (predizione e aggiornamento) esprimibile nel modo seguente:

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} \quad (60a)$$

$$P_{k+1|k} = AP_{k|k}A^T + Q \quad (60b)$$

$$K_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1} = P_{k+1|k}C^T\Lambda_{k+1}^{-1} \quad (61)$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) \quad (62a)$$

$$P_{k+1|k+1} = (I - K_{k+1}C)P_{k+1|k} \quad (62b)$$

dove K_{k+1} è il guadagno di Kalman tempo-variante e Λ_{k+1} è la matrice di covarianza del processo d'innovazione.

Sostanzialmente, per dedurre le espressioni del Filtro d'Informazione da quelle del classico Filtro di Kalman, bisogna eseguire su queste ultime una trasformazione dal consueto spazio di stato al cosiddetto *Spazio d'Informazione*. Indicando con $Z_{k|k}$ la matrice del cambiamento di base, essa definisce lo Spazio d'Informazione se viene scelta come:

$$Z_{k|k} \triangleq P_{k|k}^{-1} \quad (63)$$

e di conseguenza il vettore di stato nella nuova base, chiamato *vettore d'informazione* sarà dato da:

$$z_{k|k} = Z_{k|k}x_{k|k} \quad (64)$$

Prima di procedere nella deduzione del Filtro d'Informazione, comunque, conviene introdurre due identità che verranno sfruttate nei passaggi successivi; la prima di esse è data semplicemente da una riscrittura dell'equazione (62b):

$$I - K_{k+1}C = P_{k+1|k+1}P_{k+1|k}^{-1} \quad (65)$$

la seconda identità, invece, proviene da una rielaborazione dell'espressione del guadagno di Kalman (61):

$$\begin{aligned} K_{k+1} &= P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1} \\ &= (I - K_{k+1}C)P_{k+1|k}C^TR^{-1} \\ &= P_{k+1|k+1}C^TR^{-1} \end{aligned} \quad (66)$$

in cui l'ultima uguaglianza è ottenuta applicando la (65).

Considerando ora la classica espressione di aggiornamento dello stato (62a) e rielaborandola in modo opportuno, si perviene alla seguente scrittura nella quale sono poste in evidenza le identità appena ricavate, ovvero:

$$\begin{aligned} \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) \\ &= \underbrace{(I - K_{k+1}C)}_{\text{eq. (65)}}\hat{x}_{k+1|k} + \underbrace{K_{k+1}}_{\text{eq. (66)}}y_{k+1} \end{aligned}$$

da cui, sostituendo le espressioni (65) e (66) come indicato, dopo qualche passaggio algebrico ed una premoltiplicazione di ambo i membri per $P_{k+1|k+1}^{-1}$ si ottiene:

$$P_{k+1|k+1}^{-1}\hat{x}_{k+1|k+1} = P_{k+1|k}^{-1}\hat{x}_{k+1|k} + C^TR^{-1}y_{k+1}$$

In fine, ricordando la definizione di vettore d'informazione (64) si riconosce immediatamente l'equivalenza di questa espressione con la relazione:

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + C^T R^{-1} y_{k+1} \quad (67)$$

che assume il ruolo di equazione di aggiornamento dello stato nel filtro d'informazione.

Per quanto riguarda l'equazione di aggiornamento della varianza d'errore di stima (62b), invece, utilizzando ancora le identità (65), (66) e premoltiplicando due volte ambo i membri dell'equazione per $P_{k+1|k+1}^{-1}$ si ottiene, dopo qualche passaggio algebrico, l'espressione:

$$P_{k+1|k+1}^{-1} = P_{k+1|k}^{-1} + C^T R^{-1} C$$

la quale, applicando la definizione di matrice d'informazione (63), è esprimibile semplicemente come:

$$Z_{k+1|k+1} = Z_{k+1|k} + C^T R^{-1} C \quad (68)$$

In conclusione, se lo stadio di aggiornamento del filtro d'Informazione è definito dalle equazioni (67) e (68), lo stadio di predizione si ricava direttamente dallo stesso stadio del filtro di Kalman classico eseguendo solo il cambio di base già discusso. Dunque, l'algoritmo in due stadi del filtro d'informazione è riassumibile nelle relazioni ricorsive seguenti:

$$\hat{z}_{k+1|k} = Z_{k+1|k} A Z_{k+1|k}^{-1} \hat{z}_{k|k} \quad (69a)$$

$$Z_{k+1|k} = (A Z_{k|k} A^T + Q)^{-1} \quad (69b)$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + C^T R^{-1} y_{k+1} \quad (70a)$$

$$Z_{k+1|k+1} = Z_{k+1|k} + C^T R^{-1} C \quad (70b)$$

APPENDIX C

DETTAGLI SUL FILTRO PARTICELLARE

A. Estrapolazione delle equazioni dell'importance sampling

Si introduce ora il concetto della distribuzione Importance-Sampling: rappresentiamo la funzione densità di probabilità a posteriori tramite un set di campioni casuali con peso associato $\{x_{0:k}^i, w_k^i\}$.

Questi denotano una misura (realizzazione) casuale della variabile aleatoria con PDF-Posterior $p(x_{0:k}|z_{1:k})$. I punti $\{x_{0:k}^i, i = 0, \dots, N_s\}$ (dove N_s è il numero di particelle totali) mi rappresentano dei punti di supporto, con dei pesi associati $\{w_k^i, i = 1, \dots, N_s\}$ e $x_{0:k} = \{x_j, j = 0, \dots, k\}$ è il set dei punti di supporto all'istante k .

I pesi associati ad ogni stato sono inoltre normalizzati ad ogni istante: $\sum_i w_k^i = 1$.

La PDF-Posterior sarà quindi approssimata (con dei pesi discreti), come:

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_{0:k} - x_{0:k}^i). \quad (71)$$

A questo punto, supponiamo che $p(x) \propto \pi(x)$ sia una densità di probabilità dalla quale è difficile estrarne dei campioni, ma per il quale $\pi(x)$ può essere valutato (e così in modo proporzionale a $p(x)$).

Poi, siano x_i realizzazioni della v.a. $q(x)$, $i = 1, \dots, N_s$, cioè campioni facilmente generati da una densità proposta $q(\cdot)$, chiamata *proposal distribution*. Così, un'approssimazione pesata della densità $p(\cdot)$ è data da

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i), \quad (72)$$

dove $w^i \propto \frac{\pi(x^i)}{q(x^i)}$ è il peso normalizzato della i -esima particella. Riprendendo quindi $I(f_k)$ e sostituendo la densità approssimata si ottiene:

$$\begin{aligned} I(f_k) &= \int f_k(x_{0:k}) \frac{p(x_{0:k}|z_{1:k})}{q(x_{0:k}|z_{1:k})} q(x_{0:k}|z_{1:k}) dx_{0:k} \\ &= \int f_k(x_{0:k}) w^*(x_{0:k}) q(x_{0:k}|z_{1:k}) dx_{0:k}, \end{aligned} \quad (73)$$

dove $w^*(x_{0:k})$ è definito come:

$$w^*(x_{0:k}) = \frac{p(x_{0:k}|z_{1:k})}{q(x_{0:k}|z_{1:k})} = \frac{p(z_{1:k}|x_{0:k})p(x_{0:k})}{q(x_{0:k}|z_{1:k})p(z_{1:k})} \quad (74)$$

, e rappresenta i pesi effettivi.

Siccome non è possibile conoscere la costante di normalizzazione $p(z_{1:k})$, si cerca di aggirare il problema utilizzando dei pesi non normalizzati, proporzionali a quelli di partenza:

$$w(x_{0:k}) = \frac{p(z_{1:k}|x_{0:k})p(x_{0:k})}{q(x_{0:k}|z_{1:k})} \propto w^*(x_{0:k}), \quad (75)$$

potendo così riformulare il problema in questo modo:

$$I(f_k) = \int f_k(x_{0:k}) \frac{w(x_{0:k})}{p(z_{1:k})} q(x_{0:k}|z_{1:k}) dx_{0:k}, \quad (76)$$

esprimendo $p(z_{1:k})$ come:

$$\begin{aligned} p(z_{1:k}) &= \int p(z_{1:k}|x_{0:k})p(x_{0:k}) dx_{0:k} = \\ &= \int \frac{p(z_{1:k}|x_{0:k})p(x_{0:k})}{q(x_{0:k}|z_{1:k})} q(x_{0:k}|z_{1:k}) dx_{0:k} \\ &= \int w_k(x_{0:k}) q(x_{0:k}|z_{1:k}) dx_{0:k}, \end{aligned} \quad (77)$$

ottenendo quindi:

$$I(f_k) = \frac{\int f_k(x_{0:k}) w_k(x_{0:k}) q(x_{0:k} | z_{1:k}) dx_{0:k}}{\int w_k(x_{0:k}) q(x_{0:k} | z_{1:k}) dx_{0:k}}, \quad (78)$$

che può essere utilmente approssimata come:

$$\begin{aligned} \widehat{I}_N(f_k) &= \frac{\frac{1}{N} \sum_{i=1}^{N_s} f_k(x_{0:k}^{(i)}) w(x_{0:k}^{(i)})}{\frac{1}{N} \sum_{j=1}^{N_s} w(x_{0:k}^{(j)})}} \\ &= \sum_{i=1}^N f_k(x_{0:k}^{(i)}) \tilde{w}_k^{(i)}, \end{aligned} \quad (79)$$

dove si usano i pesi d'importanza normalizzati (stimati):

$$\tilde{w}_k^{(i)} = \frac{w(x_{0:k}^{(i)})}{\frac{1}{N} \sum_{j=1}^N w(x_{0:k}^{(j)})}. \quad (80)$$

B. Note implementative

Nel filtro particellare, il passo di predizione nell'algoritmo ricorsivo richiede di avere a disposizione la rappresentazione della pdf Likelihood $p(z_k | x_k^{(i)})$ in forma particellare. Per averla è necessario avere la media e la corrispondente varianza. Considerando che

$$p(z_k | x_k^{(i)}) = p(Cx_k^{(i)} + v_k^{(i)} | x_k^{(i)})$$

quindi sfruttando la struttura di C, e il fatto che i rumori di misura negli assi x e y sono indipendenti:

$$\begin{aligned} p(Cx_k^{(i)} + v_k^{(i)} | x_k^{(i)}) &= \\ p(x1^{(i)} + v1_k^{(i)} | x1_k^{(i)}) p(x3^{(i)} + v3_k^{(i)} | x3_k^{(i)}), \end{aligned}$$

dove $x1$ e $x3$ sono la prima e la terza componente dello stato¹².

Prendendole singolarmente, $p(xn^{(i)} + vn_k^{(i)} | xn_k^{(i)})$ indica semplicemente qual è la possibilità che si estragga la misura $xn^{(i)} + vn_k^{(i)}$ sapendo lo stato $xn^{(i)}$, quindi è semplicemente la probabilità che, dato $xn^{(i)}$, avrebbe una realizzazione di gaussiana $\mathcal{N}(xn^{(i)}, \sum_{v_n})$. Perciò risulterà $p(z_k | x_k^{(i)}) = \mathcal{N}(z1_k - x1^{(i)}, \sum_{v1}) \mathcal{N}(z2_k - x3^{(i)}, \sum_{v2})$.

APPENDIX D

PSEUDO-ALGORITMI RICORSIVI PER L'IMPLEMENTAZIONE DEL PARTICLE FILTER.

A. Pseudo-algoritmo ricorsivo, data fusion 1

Il programma relativo al primo metodo di fusione del filtro particellare è strutturato principalmente in 3 fasi:

- 1) Estrazione di N_s particelle $x_k^{(i)} \sim (x_k^{(i)} | x_{0:k-1}^{(i)}, z_{1:k}) = p(x_k | x_{k-1}^{(i)})$, e aggiornamento delle traiettorie $x_{1:k}^{(i)} := \{x_{1:k-1}^{(i)}, x_k^{(i)}\}$. Sarà cioè implementato un modello in spazio di stato in cui le particelle, rappresentanti i campioni della pdf, verranno applicati al modello per rappresentare così la pdf dell'istante successivo, vincolato dalla condizione delle particelle stesse all'istante precedente.
- 2) Aggiornamento del peso di ogni particella $w_k^{(i)} = w_{k-1}^{(i)} p(z_k | x_k^{(i)})$. Le particelle "predette" di un passo saranno quindi aggiornate in base alla pdf likelihood, dando maggior importanza a quelle più vicine alla misura rilevata. Sarà in questo punto che il sensor fusion sarà implementato, adoperando la sola e semplice misura fusa z_k .
- 3) Normalizzazione dei pesi $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^N w_k^{(i)}}$. Ossia ci si riporta alla proprietà base delle pdf, cioè che la somma delle possibilità sia 1, in quanto vengono modificate in maniera proporzionale ai pesi "reali".

B. Pseudo-algoritmo ricorsivo, data fusion 2

Rappresentando con $\{x_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^{N_p}$ l'insieme di particelle del passo precedente, al passo n si esegue:

- 1) Si inizializza dapprima associando le particelle del passo precedente come le prime della sequenza: $\{x^{0(i)}, w^{0(i)}\}_{i=1}^{N_p} = \{x_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^{N_p}$.
- 2) Si ripete questo punto per un numero di volte pari al numero di misure a disposizione, cioè: for $m=1,2,\dots,M$
 - a) Proposal distribution: estraggo i particelle $x^{m(i)} \sim q_m(x^m | x^{m-1(i)}, y_n^m)$.
 - b) Aggiorno i pesi: $w^{m(i)} \propto \frac{w^{m-1(i)} p(z_n^m | x^{m(i)}) p_m(x^{m(i)} | x^{m-1(i)})}{q_m(x^{m(i)} | x^{m-1(i)}, z_n^m)}$, e li normalizzo, con $\sum_{i=1}^{N_p} w^{m(i)} = 1$.
 - c) Nel caso si voglia fare resampling anche all'interno del ciclo, cioè negli stati intermedi ad un campionamento bisogna: estrarre delle particelle nuove $a_i \sim \{w^{m(k)}\}_{k=1}^{N_p}$, e rimpiazzare così: $\{x^{m(i)}, w^{m(i)}\} \leftarrow \{x^{m(a_i)}, 1/N_p\}$.

¹² $x1$ e $x3$ sono le coordinate in x,y; si veda il modello.

3) Terminare predisponendo per l'istante successivo:

$$\left\{ x_n^{(i)}, w_n^{(i)} \right\}_{i=1}^{Np} = \left\{ x^{M(i)}, w^{M(i)} \right\}_{i=1}^{Np}.$$

Nell'implementazione del programma per confrontare le prestazioni viene adoperata, come nel primo caso, al posto della $q_m(x^m | x^{m-1(i)}, y_n^m)$ l'applicazione dello stato (particella), al modello, simulando quindi la distribuzione $p_m(x^{m(i)} | x^{m-1(i)})$.

In questo modo l'aggiornamento dei pesi viene effettuato semplicemente tramite:

$w^{m(i)} = w^{m-1(i)} p(z_n^m | x^{m(i)})$, e la Likelihood viene sviluppata come nel primo caso.

REFERENCES

- [1] F. V. Andrea Soliman and D. Zilio, "Metodi per il visual tracking," Progetto per il corso di Progettazione di Sistemi di Controllo, a.a. 2005-2006.
- [2] A. P. Andrews and M. S. Grewal, *Kalman Filtering: Theory and Paractice using MATLAB*, 2nd ed. John Wiley and Sons Inc., 2001.
- [3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE transaction on signal processing*, vol. 50, no. 2, pp. 174-188, Febbraio 2002.
- [4] Y. Bar-Shalom and X.-R. Li, *Multitarget - Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.
- [5] J. C. Becker, "Fusion of heterogeneous sensors for the guidance of an autonomous vehicle," in *3rd International Conference on Information Fusion*, Paris (France), Luglio 2000, pp. WED5/11-WED5/18.
- [6] M. Bertinato, G. Ortolan, F. Maran, R. Marcon, A. Marcassa, F. Zanella, P. Zambotto, L. Schenato, and A. Cenedese, "Rf localization and tracking of mobile nodes in wireless sensors networks: Architectures, algorithms and experiments," Department of Information Engineering, University of Padua, Italy, Tech. Rep., 2008.
- [7] R. R. Brooks and S. Iyengar, *Multi-Sensor Fusion: fundamentals and applications with software*. Prentice Hall Inc., 1997.
- [8] A. F. Cattoni, A. Dore, and C. S. Regazzoni, "Video-radio fusion approach for target tracking in smart spaces," in *10th International Conference on Information Fusion*, Quebec, Que (Canada), Luglio 2007, pp. 1-7.
- [9] C.-Y. Chong, S. Mori, W. H. Barker, and K.-C. Chang, "Architectures and algorithms for track association and fusion," *IEEE AES Systems Magazine*, pp. 5-13, Gennaio 2000.
- [10] W. Du and J. Piater, "Data fusion by belief propagation for multi-camera tracking," in *9th International Conference on Information Fusion*, Florence (Italy), Luglio 2006, pp. 1-8.
- [11] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, Giugno 2008.
- [12] J.-M. Frahm, K. Koser, and R. Koch, "Pose estimation for multi-camera systems," in *Proceedings on DAGM*, Tuebingen (Germania), 2004.
- [13] A. Fusiello, *Visione Computazionale: appunti delle lezioni*. Dipartimento di Informatica, Università di Verona, 2007.
- [14] D. L. Hall and J. Llinas, *Handbook for multisensor data fusion (Electrical engineering and applied signal processing)*. CRC Press LLC, 2001.
- [15] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel kalman filtering," *IEEE transaction on automatic control*, vol. 33, no. 1, pp. 88-94, Gennaio 1988.
- [16] Z. Liu, X. Waing, and M. Palaniswami, "Improving tracking accuracy using information of dissimilar sensors," in *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing*, Gennaio 2005, pp. 94-99.
- [17] A. Marcassa, "Localizzazione e tracking con rete si sensori wireless," Master's thesis, Università degli studi di Padova, Aprile 2008.
- [18] J. V. Patrick Prez and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE*, vol.92, No.3, Marzo 2004.
- [19] M. Piccardi, "Background subtraction techniques: a review," *Proceedings of IEEE International Conference Systems, Man and Cybernetics*, vol. 4, pp. 3099-3104, Ottobre 2004.
- [20] G. Picci, *Filtraggio Statistico (Wiener, Levinson, Kalman) e applicazioni*. Padova: Edizioni Libreria Progetto, 2006.
- [21] H. F. Robert Collins, Alan J. Lipton and T. Kanade, "Algorithm for cooperative multisensor surveillance," *D.A.R.P.A.*, Ottobre 2001.
- [22] M. D. Rocchi, A. Manente, A. Mucciardi, and G. Paccagnella, "Localizzazione e tracking tramite sensor fusion," Progetto per il corso di Progettazione di Sistemi di Controllo, a.a. 2007-2008.
- [23] E. Song, Y. Zhu, J. Zhou, and Z. You, "Optimal kalman filtering fusion with cross-correlated sensor noises," *Automatica*, no. 43, pp. 1450-1456, 2007.
- [24] B.-N. Vo and S. Challa, "Multi-sensor data fusion for tracking in clutter," *University of Melbourne, Department of Electrical and Electronic Engineering*, 2001.
- [25] M. Wheeler, B. Schrick, W. Whitacre, M. Campbell, R. Rysdyk, and R. Wise, "Cooperative tracking of moving targets by a team of autonomous uavs," in *25th Digital Avionics Systems Conference*, Portland (Oregon), Ottobre 2006, pp. 1-9.
- [26] E. C. Yeh and C. S. Wang, "An auto calibration scheme for sensor fusion," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Las Vegas (Nevada), Ottobre 1994, pp. 136-142.
- [27] Z. Yue, S. K. Zhou, and R. Chellappa, "Robust two-camera tracking using homography," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, no. 3, pp. 1-4, Maggio 2004.
- [28] Y. Zhai, M. B. Yeary, J. P. Havlicek, and G. Fan, "A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems," *IEEE transaction on instrumentaion and measurement*, vol. 57, no. 10, pp. 2377-2387, Ottobre 2008.



Gennaro Azzollini



Luciano Lunardi



Sebastiano Segantin