

**ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA  
SEDE DI CESENA  
SECONDA FACOLTÀ DI INGEGNERIA CON SEDE A CESENA  
CORSO DI LAUREA TRIENNALE IN INGEGNERIA ELETTRONICA  
E DELLE TELECOMUNICAZIONI**

**ALGORITMI PER L'INSEGUIMENTO DI  
TARGET MULTIPLI IN SISTEMI RADAR  
DISTRIBUITI**

Tesi in: TELECOMUNICAZIONI

Relatore:

**Chiar.mo Prof. Ing. Marco Chiani**

Correlatori:

**Dott. Ing. Enrico Paolini**

Presentata da:

**Iacopo Mambelli**

Sessione I

Anno Accademico 2011-2012



# **PAROLE CHIAVE**

**Radar Multistatico**

**Banda Ultra-Larga (UWB)**

**Algoritmi di Tracking**

**Filtro di Kalman**

**Filtro SHC**



# Indice

<b>Introduzione</b>	<b>xiii</b>
<b>1 Descrizione del sistema</b>	<b>1</b>
1.1 Radar Multistatico . . . . .	1
1.2 Stima del <i>TOA</i> . . . . .	3
1.3 Potenze Trasmesse e Ricevute . . . . .	5
1.4 Rimozione del clutter . . . . .	7
1.5 Imaging, Detection, e Tracking . . . . .	8
<b>2 Filtro di Kalman e Tracking singolo Target</b>	<b>11</b>
2.1 Introduzione al filtro . . . . .	11
2.2 Filtro di Kalman . . . . .	12
2.2.1 Modello di Processo . . . . .	12
2.2.2 Modello di misura . . . . .	13
2.3 Algoritmo ricorsivo . . . . .	15
2.3.1 Inizializzazione . . . . .	15
2.3.2 Predizione . . . . .	16
2.3.3 Correzione . . . . .	16
2.4 Stazionarietà del filtro . . . . .	17
2.5 Formulazione di Bayes del Filtro di Kalman . . . . .	18
2.5.1 Riassumendo . . . . .	20
2.6 Tracking Singolo Target . . . . .	20
2.6.1 Esempio di veicolo in due dimensioni . . . . .	23
2.6.2 Risultato delle Simulazioni . . . . .	26
<b>3 Tracking Multitarget</b>	<b>35</b>
3.1 Introduzione all'approccio Multitarget . . . . .	35

---

---

3.2	Associazione Traccia-Misura . . . . .	37
3.2.1	Global Association Distance . . . . .	37
3.3	Single-Hypotesis Correlation . . . . .	38
3.3.1	Introduzione agli Algoritmi di Correlazione . . . . .	38
3.3.2	Algoritmo SHC . . . . .	39
3.3.3	SHC con False Misure e Mancati Allarmi . . . . .	41
3.4	Introduzione al Simulatore . . . . .	41
3.4.1	Tracking Multiplo . . . . .	43
3.5	Simulazioni al Calcolatore del Sistema di Tracking Multitar- get Basato su Algoritmo SHC . . . . .	45
3.5.1	Simulazioni Effettuate al Variare di $\sigma_z$ . . . . .	46
3.5.2	Simulazioni Effettuate al Variare di $\sigma_a$ . . . . .	48
3.5.3	Simulazioni Effettuate al Variare del Passo Tempora- le dt . . . . .	50
3.6	Difetti legati all' univoca Correlazione . . . . .	53
<b>4</b>	<b>Prestazioni del Sistema Implementato</b>	<b>57</b>
4.1	Conclusioni . . . . .	60
<b>A</b>	<b>Richiami di Teoria della Probabilità</b>	<b>63</b>
A.1	Matrice di covarianza . . . . .	63
A.2	Variabili aleatorie congiunte Gaussiane . . . . .	65
A.3	Processi stocastici tempo-discreti . . . . .	66
	<b>Bibliografia</b>	<b>69</b>

---

# Elenco delle figure

1.1	Schema di funzionamento del sistema. . . . .	1
1.2	Schema a blocchi di una coppia trasmettitore-ricevitore. . . . .	2
1.3	Stima del TOA . . . . .	3
1.4	ovali di Cassini . . . . .	4
1.5	Percorso diretto ( $R_o$ ) e riflesso ( $R_{Tx} + R_{Rx}$ ) di un radar bistatico. . . . .	6
1.6	Schema di principio della tecnica Empty-Room . . . . .	7
1.7	Esempio di creazione dell'immagine. . . . .	8
1.8	Schema a blocchi del sistema di Imaging, Detection, e Tracking. . . . .	9
2.1	Schema dell'algoritmo ciclico ricorsivo di Predizione-Correzione. . . . .	15
2.2	Traiettoria rettilinea con $x$ e $y$ crescenti, si é considerata una simulazione di 400 s con passo $dt$ unitario. . . . .	21
2.3	Traiettoria rettilinea con $y$ costante, si é considerata una simulazione di 400 s con passo $dt$ unitario. . . . .	21
2.4	Traiettoria circolare centrata in $x = 250 (m)$ , $y = 250 (m)$ , di raggio ( $Radius$ ) = $50 (m)$ , con $w = 0.1 rad/s$ e sfasamento nullo. . . . .	22
2.5	Risultato della simulazione per Traiettoria rettilinea con $y = 250 m$ , assumendo $\sigma_z = 5 m$ , velocità del target $vel = 2m/s$ , passo temporale di scansione $dt = 1$ . . . . .	26
2.6	Risultato della simulazione per Traiettoria rettilinea con $y = 250 m$ , assumendo $\sigma_z = 10 m$ , velocità del target $vel = 2m/s$ , passo temporale di scansione $dt = 1$ . . . . .	27

---

2.7	Risultato della simulazione per Traiettorie rettilinea con $y = 250\text{ m}$ , assumendo $\sigma_z = 5\text{ m}$ , velocità del target = $6\text{ m/s}$ , passo temporale di scansione $dt = 1\text{ s}$ . . . . .	28
2.8	Risultato della simulazione per Traiettorie rettilinea con $y = 250\text{ m}$ , assumendo $\sigma_z = 1\text{ m}$ , velocità del target $vel = 2\text{ m/s}$ , passo temporale di scansione $dt = 1$ . . . . .	29
2.9	Risultato della simulazione per Traiettorie rettilinea con $x$ e $y$ variabili, assumendo $\sigma_z = 5\text{ m}$ , velocità del target $vel = 2\text{ m/s}$ , passo temporale di scansione $dt = 1\text{ s}$ . . . . .	30
2.10	Risultato della simulazione per Traiettorie rettilinea con $x$ e $y$ variabili, assumendo $\sigma_z = 10\text{ m}$ , velocità del target $vel = 2\text{ m/s}$ , passo temporale di scansione $dt = 1\text{ s}$ . . . . .	30
2.11	Risultato della simulazione per Traiettorie circolare, assumendo $\sigma_z = 5\text{ m}$ , pulsazione $w = 0.1\text{ rad/sec}$ , durata della simulazione pari a $100\text{ s}$ , sfasamento iniziale nullo, passo temporale di scansione $dt = 1\text{ s}$ , scelta del vettore iniziale di coordinate $p_{0_x} = p_{0_y} = 250\text{ (m)}$ . . . . .	31
2.12	Risultato della simulazione per Traiettorie circolare, assumendo $\sigma_z = 5\text{ m}$ , pulsazione $w = 0.1\text{ rad/sec}$ , durata della simulazione pari a $100\text{ s}$ , sfasamento iniziale nullo, passo temporale di scansione $dt = 1\text{ s}$ , scelta del vettore iniziale di coordinate $p_{0_x} = p_{0_y} = 0\text{ (m)}$ . . . . .	32
2.13	Risultato della simulazione per Traiettorie circolare, assumendo $\sigma_z = 10\text{ m}$ , pulsazione $w = 0.1\text{ rad/sec}$ , durata della simulazione pari a $100\text{ s}$ , sfasamento iniziale nullo, passo temporale di scansione $dt = 1\text{ s}$ , scelta del vettore iniziale di coordinate $p_{0_x} = p_{0_y} = 250\text{ (m)}$ . . . . .	33
3.1	Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di $\sigma_a = 0.2\text{ m/s}$ e $\sigma_z = 1\text{ m}$ . . . . .	46
3.2	Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di $\sigma_a = 0.2\text{ m/s}$ e $\sigma_z = 5\text{ m}$ . . . . .	46
3.3	Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di $\sigma_a = 0.2\text{ m/s}$ e $\sigma_z = 10\text{ m}$ . . . . .	47
3.4	Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di $\sigma_a = 0.5\text{ m/s}$ e $\sigma_z = 5\text{ m}$ . . . . .	48

---

---

3.5	Tracking multiplo costituito da traiettorie rettiline con l'assunzione di $\sigma_a = 1$ m/s e $\sigma_z = 5$ m. . . . .	49
3.6	Tracking multiplo costituito da traiettorie rettiline con l'assunzione di durata = 400 s e passo $dt = 0.5$ s. . . . .	50
3.7	Tracking multiplo costituito da traiettorie rettiline con l'assunzione di durata = 300 s e passo $dt = 0.2$ s. . . . .	51
3.8	Dettaglio di Figura 3.2. . . . .	51
3.9	Dettaglio di Figura 3.6. . . . .	52
3.10	Dettaglio di Figura 3.7. . . . .	52
3.11	Primo caso di errata identificazione. . . . .	54
3.12	Caso di giusta identificazione. . . . .	56
4.1	Esempio di simulazione di incrocio dei target correttamente riuscito. . . . .	58
4.2	Esempio di simulazione di incrocio dei target in cui avviene lo scambio di associazione fra tracce e misure. . . . .	59

---



## Elenco delle tabelle

3.1	Principali parametri utilizzati dal simulatore per descrivere l'ambiente. . . . .	42
3.2	Principali parametri utilizzati dal simulatore per caratterizzare il target. . . . .	43
3.3	Restanti parametri utilizzati dal simulatore. . . . .	43
3.4	Principali parametri utilizzati nel simulatore per il filtraggio multitarget. . . . .	44
3.5	Principali parametri utilizzati dal simulatore attraverso la correzione. . . . .	45
3.6	Precisazioni sui colori usati nelle simulazioni. . . . .	45
3.7	Descrizione dei valori iniziali dei vettori di stato dei singoli target. . . . .	45
3.8	Principali parametri utilizzati per simulare i diversi comportamenti del filtro cambiando la deviazione standard del rumore di misura. . . . .	48
3.9	Principali parametri utilizzati per simulare i diversi comportamenti del filtro cambiando la deviazione standard del rumore di processo. . . . .	49
3.10	Principali parametri utilizzati per simulare i diversi comportamenti del filtro cambiando il passo temporale. . . . .	53
3.11	Principali parametri utilizzati per inizializzare i vettori necessari per simulare una delle errate identificazioni. . . . .	54
3.12	Principali parametri utilizzati per simulare i diversi comportamenti dovuti all'errata identificazione. . . . .	55
3.13	Principali parametri utilizzati per inizializzare i vettori necessari per simulare una delle errate identificazioni. . . . .	56

---

4.1	Principali parametri utilizzati dal simulatore per eseguire una generica simulazione di incrocio. . . . .	58
4.2	Principali parametri utilizzati dal simulatore per eseguire un ciclo for di 100 simulazioni di incrocio. . . . .	59
4.3	Principali parametri utilizzati dal simulatore per eseguire un ciclo for di 1000 simulazioni di incrocio. . . . .	60

# Introduzione

I sistemi di anti-intrusione permettono la rilevazione di oggetti non identificati all'interno di un'area da sorvegliare. Fino ad ora i sistemi realizzati per ambienti esterni non consentivano di poter tracciare lo spostamento di oggetti all'interno di questa area, ne' tantomeno la localizzazione con precisione del target.

Questo elaborato ha lo scopo di studiare e realizzare un sistema radar multistatico per il tracking di molteplici oggetti in uno spazio da sorvegliare, basandosi su segnali *UWB*(Ultra-WideBand), cioè impulsi con banda molto grande nelle frequenze e durata nel tempo dell'ordine dei nanosecondi. Questa nuova tecnologia si differenzia dalle precedenti, consentendo di ottenere una elevata risoluzione e precisione di localizzazione. Questo sviluppo avrà il beneficio di essere applicato a molti settori delle telecomunicazioni quali per uso civile e militare.

Nel capitolo 1 daremo una descrizione complessiva del sistema e del suo funzionamento soffermandoci sugli elementi principali che compongono il radar multistatico.

Nel capitolo 2 analizzeremo il filtro di Kalman a singolo target, studiandone le equazioni e il principio di funzionamento implementando il filtro stesso. Faremo diverse simulazioni e commenteremo i risultati.

Nel capitolo 3 verrà descritto e implementato un filtraggio di tipo multitarget con filtro SHC (Single-Hypothesis-Correlation), commentandone le prestazioni e i risultati derivati da diversi tipi di simulazioni.

Nel capitolo 4 analizzeremo le prestazioni del filtro implementato e trarremo le conclusioni di questo elaborato in base alle simulazioni e ipotesi considerate.

Infine, nell'Appendice A, vengono richiamati alcuni concetti di statistica, utili in questo elaborato.

---



# Capitolo 1

## Descrizione del sistema

### 1.1 Radar Multistatico

Il radar multistatico é composto da un trasmettitore e nel caso in esame, da tre ricevitori (devono essere almeno tre per identificare in modo univoco la posizione del target) disposti generalmente sui bordi dell'area da sorvegliare.

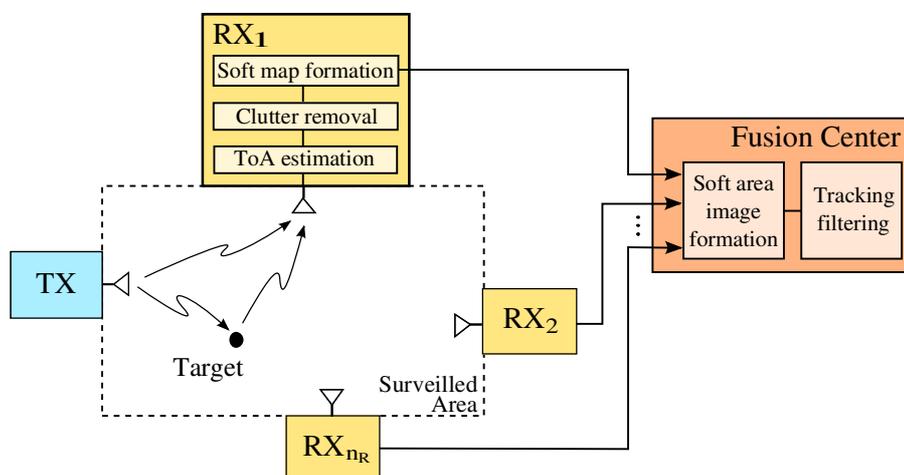


Figura 1.1: Schema di funzionamento del sistema.

Come si vede dalla Figura 1.1, la rivelazione e la localizzazione del target comprende una serie di fasi che possono essere riassunte in: stima del TOA, rimozione del clutter, creazione delle mappe soft, imaging, detection, e tracking.

---

Il radar multistatico può essere visto come formato da più coppie di trasmettitori-ricevitori, perciò per capirne meglio il suo funzionamento si può considerare una delle generiche coppie il cui schema a blocchi è del tipo di Figura 1.2.

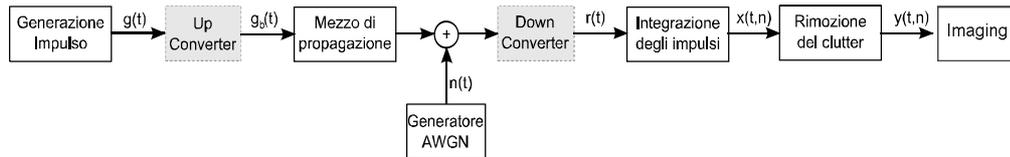


Figura 1.2: Schema a blocchi di una coppia trasmettitore-ricevitore.

Il trasmettitore si occupa della trasmissione del segnale generando un impulso a forma di monociclo gaussiano, il quale a sua volta verrà inviato al blocco di *Up-converter* in modo da traslarne lo spettro in banda, essenzialmente lo moduliamo per mandarlo a radiofrequenza.

Il mezzo di propagazione simula tutto ciò che avviene nel canale tenendo conto di vari fattori di propagazione ambientale. Successivamente viene aggiunto del rumore gaussiano bianco attraverso il blocco *Generatore AWGN* che considera tutte le rumorosità presenti nel canale e negli apparati di ricezione.

Il ricevitore come primo compito deve riconvertire il segnale in banda base attraverso un blocco di *down-converter* e integrare successivamente gli impulsi, ovvero sommiamo fra loro più repliche di impulsi ricevuti in modo che i monocicli si sommino coerentemente mentre il rumore si attenua. Nel fare questa operazione bisogna stare molto attenti perché essa risulta vantaggiosa se sommiamo fra loro impulsi molto vicini nel tempo per i quali non perdiamo informazioni sulla realtà del target mentre può essere dannoso per l'efficacia del sistema se l'intervallo temporale considerato fosse troppo ampio rispetto alla velocità del target, non considerando il fatto che il target in quel lasso temporale potrebbe essersi spostato.

Il blocco successivo si occupa della rimozione del clutter, cioè di tutto ciò che non sia l'oggetto da monitorare. La rimozione del clutter è un tema molto importante nell'ambito radar poiché esso si occupa di pulire il segnale generato dal target da tutto ciò che potrebbe essere rumore oppure riflesso di oggetti che sono nell'ambiente che stiamo monitorando (se

l'ambiente fosse interno alcuni elementi che gerano del clutter potrebbero essere sedie, tavoli ecc).

Il blocco finale di *imaging* si occupa di creare una matrice (mappa soft) nella quale venga indicata attraverso dei *pixel* la probabilità di trovare il target in quel determinato punto. Queste mappe vengono create da ciascun ricevitore indipendentemente che il target sia presente o meno all'interno dell'area. Esse vengono inviate ad un nodo centrale detto *fusion node* al quale spetta la decisione sulla presenza o meno del target e, in caso positivo, indicarne la posizione (*blocco di detection*) osservando la mappa soft *totale* creata, basandosi, per determinare la posizione, sui pixel a probabilità più alta.

Queste informazioni verranno passate al blocco di *tracking* che è l'oggetto del nostro elaborato.

## 1.2 Stima del TOA

Consideriamo  $d$  la distanza diretta fra trasmettitore-ricevitore la quale è nota,  $d_1$  la distanza fra trasmettitore-target e  $d_2$  la distanza target-ricevitore.

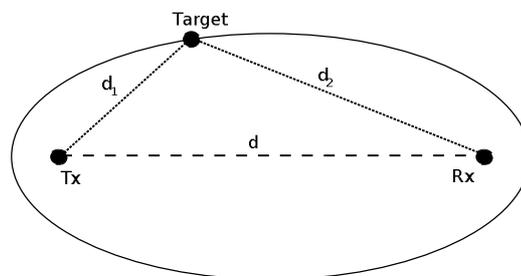


Figura 1.3: Stima del TOA

Il trasmettitore invia un segnale composto da un monociclo il quale arriva al ricevitore attraverso due percorsi distinti: uno diretto (*direct path*) e uno dovuto alla riflessione del target. Il ricevitore si sincronizza con l'impulso diretto e valuta il ritardo con cui il segnale riflesso dal target viene ricevuto.

Il contributo del cammino diretto subisce un ritardo di propagazione proporzionale alla distanza  $d$ . Tale ritardo temporale si calcola come:

$$\Delta_{T_{DP}} = \frac{d}{c} \quad (1.1)$$

Dato che il radar sfrutta il ritardo temporale del segnale ricevuto per identificarne la posizione, bisogna tenere conto che i segnali provenienti dai cammini riflessi subiranno un ritardo di propagazione diverso, dipendente dalla distanza  $d_1 + d_2$ . Tale ritardo temporale si calcola come:

$$\Delta_{T_{SP}} = \frac{d_1 + d_2}{c} \quad (1.2)$$

Con questo tempo, e grazie alla conoscenza di  $d$  il sistema é in grado di identificare la somma  $d_1 + d_2$ . Essa é un'ellisse che ha come fuochi i nodi stessi. Nel caso del radar multistatico si avrà un'ellisse per ogni coppia trasmettitore-ricevitore in modo tale che se il sistema possiede un numero di ricevitori pari almeno a tre, esso identifica in modo univoco la posizione.

La particolare geometria del radar bistatico permette di ragionare oltre che sul *TOA* anche sulle potenze riflesse dal target e ricevute dai nodi. La potenza ricevuta risulta essere costante se il target si muove lungo la particolare curva detta *ovale di Cassini*, Figura 1.4.

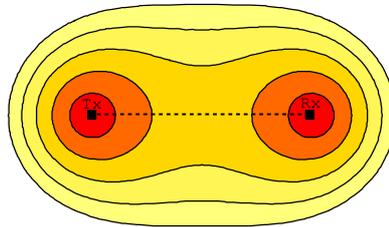


Figura 1.4: ovali di Cassini

Per il radar monostatico la potenza ricevuta é proporzionale  $1/d^4$ , dove con  $d$  intendiamo la distanza del target, mentre nel bistatico essa é

proporzionale a  $1/(d_1 d_2)^2$ , cioè al prodotto delle due distanze al quadrato.

Si può notare che fissando per il target una curva (TOA oppure potenza ricevuta costante) l'altra curva risulta variare. Si può notare inoltre che fissando la potenza del trasmettitore, nello scenario in cui il prodotto  $d_1 d_2$  aumentano si ha una corrispondente decrescita della potenza ricevuta con un conseguente aumento dell'ovale di cassini. Si deve tener conto inoltre del fatto che il target può essere individuato quando si trova all'interno dell'ovale massimo di cassini e fuori dall'ellisse minima. Per questa trattazione richiamo [1].

Analizziamo ora le potenze in gioco per ogni coppia di trasmettitore-ricevitore.

### 1.3 Potenze Trasmesse e Ricevute

Considerando una generica coppia trasmettitore-ricevitore e usando un segnale *UWB* possiamo determinare la potenza ricevuta lungo il percorso diretto (vedi Figura 1.5 nella pagina successiva),  $P_{R_{DP}}$  come:

$$P_{R_{DP}} = \int_{f_L}^{f_L+B} \frac{S_T(f) G_T(f) G_R(f) \lambda^2}{(4\pi)^2 (R_o)^2} df \quad (1.3)$$

e quella ricevuta a causa della riflessione del target  $P_{R_{SP}}$  come:

$$P_{R_{SP}} = \int_{f_L}^{f_L+B} \frac{S_T(f) G_T(f) G_R(f) \lambda^2 \sigma}{(4\pi)^2 (R_{Tx} R_{Rx})^2} df \quad (1.4)$$

dove  $S_T(f)$  è la densità di potenza trasmessa,  $G_T(f)$  e  $G_R(f)$  sono i guadagni d'antenna dipendenti dalla frequenza di trasmissione,  $\lambda = c/f$  è la lunghezza d'onda,  $R_o$  è la distanza trasmettitore-ricevitore,  $R_{Tx}$  è la distanza trasmettitore-target,  $R_{Rx}$  è la distanza target-ricevitore,  $B = f_U - f_L$  è la banda a radiofrequenza, e infine  $A_S$  è l'attenuazione supplementare

---

che si considera unitaria. Tra gli oggetti che riflettono il segnale oltre al clutter ed il target, é presente anche la riflessione sul terreno, che se considerato perfettamente liscio ed uniforme, può essere rappresentato da un unico oggetto puntiforme riflettente con sezione radar costante pari a  $\sigma_f$ .

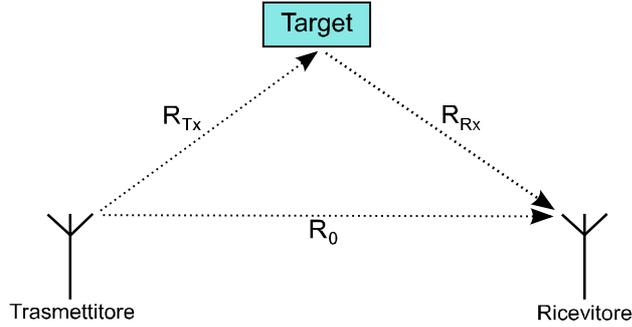


Figura 1.5: Percorso diretto ( $R_o$ ) e riflesso ( $R_{Tx} + R_{Rx}$ ) di un radar bistatico.

Considerando un guadagno di antenna costante all'interno della banda di interesse (questa ipotesi risulta essere veritiera per alcune antenne UWB), e uno spettro bianco per il segnale trasmesso possiamo scrivere le potenze come:

$$P_{RDP} = \frac{S_T G_T G_R c^2}{(4\pi)^2 (R_o)^2} \left( \frac{1}{f_L} - \frac{1}{f_L + B} \right) \quad (1.5)$$

$$P_{RSP} = \frac{S_T G_T G_R c^2 \sigma}{(4\pi)^3 (R_{Tx} R_{Rx})^2} \left( \frac{1}{f_L} - \frac{1}{f_L + B} \right) \quad (1.6)$$

L'ipotesi della costanza di  $\sigma$ , e della densità spettrale di potenza indipendente dalla frequenza, semplifica l'analisi senza alterarne lo scopo finale.

## 1.4 Rimozione del clutter

Quando il ricevitore ha ricevuto un certo numero di impulsi opportuni esso deve effettuare l'integrazione di questi al fine di ridurre l'effetto del rumore termico in modo da ottenere una misura la piú precisa possibile. A valle di tale integrazione viene eseguita la rimozione del clutter.

Esistono vari algoritmi di rimozione del clutter le cui caratteristiche principali debbono essere l'affidabilit , il ritardo temporale e la qualit  dei segnali a valle del sistema. Il ritardo temporale pu  fare la differenza nel caso in cui il nostro sistema ha bisogno di misure nell'immediato (quali applicazioni balistiche) cos  come   importante che il segnale in uscita da questo blocco non venga ne' distorto ne' attenuato.

Un esempio semplice ed intuitivo di rimozione del clutter lo si pu  dare attraverso il metodo *Empty-Room*. Questa ha il grosso vantaggio di essere una tecnica *real-time* poich  non introduce ritardo temporale. Essa consiste nell'effettuare un *set-up* iniziale in cui viene fatta una scansione di riferimento in assenza del target. Questa ha lo scopo di ottenere un segnale che descriva il clutter generato dall'ambiente circostante in modo da sottrarre al segnale ottenuto dal ricevitore quello ottenuto durante il *set-up* per ottenere idealmente il solo contributo del target.

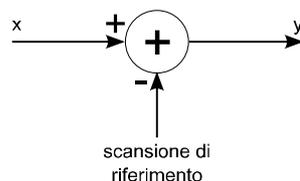


Figura 1.6: Schema di principio della tecnica Empty-Room

Come si capisce esso risulta efficiente per quei tipi di ambienti in cui l'area da sorvegliare non subisce delle variazioni durante l'osservazione e che tutte le volte in cui la scena cambia bisogna effettuare il *set-up* (esempio ambienti interni).

Esempi di rimozione del clutter per ambienti esterni possono essere:

- *Moving Target Indicator (MTI)*;
- *Fast Fourier Transform (FFT)*.

i quali sono filtri piú complessi che permettono di funzionare anche in ambienti esterni il cui svantaggio però é l'introduzione di un ritardo di breve o lunga durata a seconda del grado di complessità del filtro [6].

## 1.5 Imaging, Detection, e Tracking

Dopo la rimozione del clutter, ogni ricevitore crea una matrice (mappa soft) di pixel, il cui valore rappresenta la probabilità di trovare il target in quel pixel. Le mappe soft dei tre ricevitori vengono poi combinate a formare un'immagine. In Figura 1.7 vi é un esempio di creazione dell'immagine.

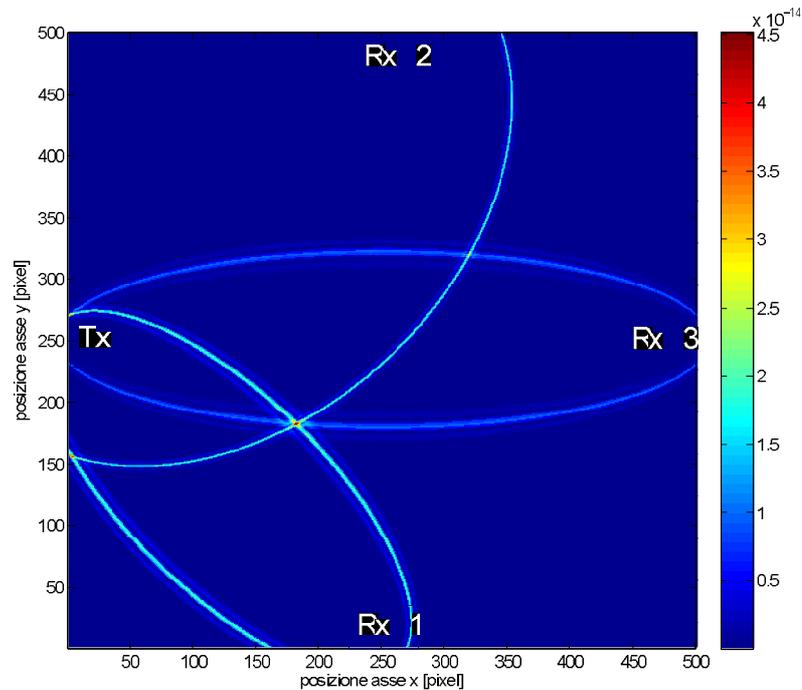


Figura 1.7: Esempio di creazione dell'immagine.

Successivamente l'immagine viene data in ingresso ad un blocco di detection (vedi schema a blocchi di Figura 1.8 nella pagina successiva) che, nel caso in questione, decide circa la presenza di uno o piú target e circa la loro posizione.

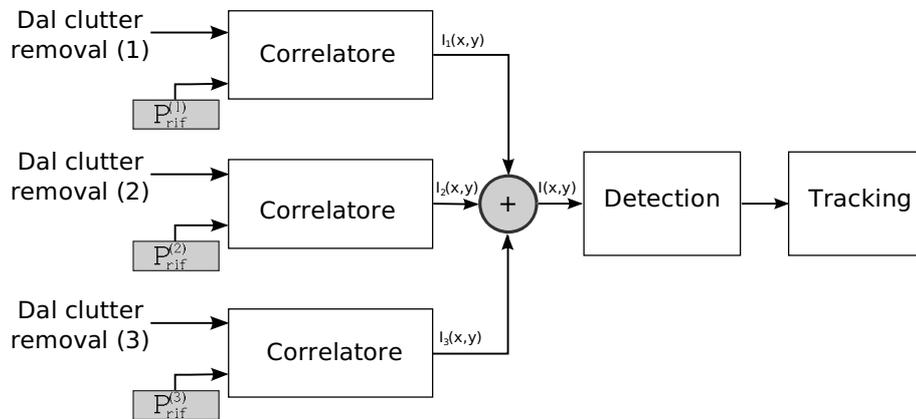


Figura 1.8: Schema a blocchi del sistema di Imaging, Detection, e Tracking.

Per creare ogni singola mappa soft, ogni ricevitore effettuata la correlazione tra il segnale ricevuto e un impulso di riferimento opportunamente traslato, per vedere se il ritardo introdotto dall'impulso di riferimento é comparabile con quello del segnale ricevuto. Dato che si suppone che il canale non introduce distorsione nel monociclo trasmesso, per calcolare l'impulso di riferimento bisogna conoscere l'impulso di partenza, il ritardo temporale, e in che modo il filtraggio del clutter modifica il segnale trasmesso.

Una volta create le tre mappe soft, l'immagine é ottenuta attraverso una combinazione non lineare delle mappe, ovvero si ottiene la *somma delle correlazioni (o immagine)  $I(x, y)$*  [2, 4]:

$$\begin{aligned}
I(x, y) &= \underbrace{\ln \cosh \left[ \int_{T_o} r_1(t) p_{rif}(t - \tau_1) dt \right]}_{I_1(x, y)} + \\
&\quad \underbrace{\ln \cosh \left[ \int_{T_o} r_2(t) p_{rif}(t - \tau_2) dt \right]}_{I_2(x, y)} + \\
&\quad \underbrace{\ln \cosh \left[ \int_{T_o} r_3(t) p_{rif}(t - \tau_3) dt \right]}_{I_3(x, y)} \\
&= I_1(x, y) + I_2(x, y) + I_3(x, y)
\end{aligned} \tag{1.7}$$

Il blocco di detection ricevendo all'ingresso *l'immagine*, decide qual é il punto piú probabile in cui si trova il target, prendendo semplicemente come *misura* della posizione del target  $z = (z_x, z_y)$ , il pixel relativo al massimo valore di  $I(x, y)$ , ovvero, per il caso di un singolo target:

$$z = (z_x, z_y) = \underset{(x, y)}{\operatorname{arg\,max}} I(x, y) \tag{1.8}$$

Il blocco di Detection fornirá le misure al blocco di Tracking il quale é l'oggetto del nostro elaborato. Inizialmente costruiremo un Filtro di Kalman per il Tracking di un singolo oggetto e terremo traccia del suo cambiamento di posizione, inseguendolo nel tempo, poi effettueremo il Tracking di oggetti multipli.

Ad ogni scansione il sistema fornirá in uscita una *stima* della posizione del target  $(\hat{x}, \hat{y})$ , basandosi sull'ultima misura  $(z_x, z_y)$  fornita dal blocco di Detection.

## Capitolo 2

# Filtro di Kalman e Tracking singolo Target

### 2.1 Introduzione al filtro

Il Filtro di Kalman é uno dei filtri piú usati in applicazioni di Tracking (dove per Tracking si intende l'inseguimento, il tracciamento della posizione dell'oggetto nel tempo) sia per la sua semplicitá sia per la sua efficienza. Esso si basa sul fatto che tutte le distribuzioni in gioco risultino essere Gaussiane a valor medio nullo e che il modello di evoluzione del sistema sia lineare. In questi casi il sistema risulta essere molto efficiente, in altri, dove queste ipotesi non vengono rispettate, possono portare ad una stima non errata . Esso é un particolare filtro *Bayesiano*, dove come detto tutte le distribuzioni risultano gaussiane. I filtri *Bayesiani* stimano probabilisticamente lo stato di un sistema dinamico a partire da una serie di osservazioni, essi si basano su densitá di probabilitá e sono piú complessi computazionalmente del filtro di Kalman, il quale risulta essere piú facilmente implementabile. Sperimentalmente si vede che il Bayesiano risulta essere migliore di quello di Kalman in tutte quelle situazioni in cui il target si muove in modo troppo rapido rispetto al campionamento temporale (poiché inizialmente Kalman dá fiducia alla stima accorgendosi poi, con l'arrivo di un certo numero di misure, che deve dare piú peso ad esse), oppure in quei casi in cui il SNR sia basso.

Vedremo poi nel Tracking Multitarget la creazione di tanti filtri di Kal-

---

man in parallelo, corrispondenti al numero di target rilevati nell'area da sorvegliare e vedremo come in alcuni casi (esempio: target vicini fra loro e con SNR non elevati), il filtro si possa agganciare all'oggetto sbagliato provocando una errata identificazione, la quale ovviamente é da evitare (basta pensare ad applicazioni balistiche in cui invece che lanciare un missile contro il nemico lo si manda contro l'alleato).

## 2.2 Filtro di Kalman

Lo scopo del filtraggio é di determinare una stima sul vettore di stato  $x_{k|k}$  il quale contiene informazioni sul target, quali la posizione, la velocità, eventualmente anche l'accelerazione. Il filtro in questione attribuirá un certo peso alla misura rilevata in funzione del rumore determinando un guadagno  $K$  detto *guadagno di Kalman*, in base ad esso la stima sará condizionata fortemente dalla misura oppure dalla predizione fatta.

La stima del *vettore di stato*  $x_{k|k}$  la si ottiene attraverso le equazioni che governano il sistema:

- *Modello di processo;*
- *Modello di misura.*

Approfondiamone gli aspetti.

### 2.2.1 Modello di Processo

L'equazione del modello di processo é la seguente:

$$x_k = F_{k-1} x_{k-1} + B_{k-1} u_{k-1} + w_{k-1} \quad (2.1)$$

dove:

- $k$  é il generico istante di campionamento temporale;
-

- $x_k \in \mathbb{R}^{n \times 1}$  é il vettore di stato, i cui elementi costituiscono una stima sulle caratteristiche del target;
- $u_k \in \mathbb{R}^{r \times 1}$  é il vettore degli ingressi deterministico cioè noto a priori, presente solo se vi é un controllo sul sistema;
- $w_k \in \mathbb{R}^{n \times 1}$  é il vettore del rumore di processo (o sullo stato), e rappresenta come gli ingressi agiscono in maniera incontrollabile sullo stato del sistema; é un rumore gaussiano a valore medio nullo con matrice di covarianza  $Q_k$ , ovvero  $w_k \sim N(0, Q_k)$ ;
- $Q_k = cov(w_k) = \mathbb{E}[w_k w_k^T] \geq 0 \in \mathbb{R}^{n \times n}$ , é la matrice di covarianza del vettore del rumore di processo, semidefinita positiva, e rappresenta la potenza del disturbo introdotto nel sistema che devia l'andamento delle variabili di stato, rispetto a quello prevedibile dalla conoscenza dell'ingresso e dalla legge lineare che ne governa l'evoluzione;
- $F_k \in \mathbb{R}^{n \times n}$  é la matrice di transizione degli stati, descrive il modello del moto;
- $B_k \in \mathbb{R}^{n \times r}$  é la matrice di controllo degli ingressi, e descrive l'evoluzione forzata della variabile di stato a partire dal valore del vettore degli ingressi  $u_k$ , nel caso in cui esso sia presente.

### 2.2.2 Modello di misura

La seconda equazione del sistema, chiamata *modello di misura* é la seguente:

$$z_k = H_k x_k + v_k \quad (2.2)$$

dove:

- $z_k \in \mathbb{R}^{m \times 1}$  é il vettore delle misure (o osservazioni) dello stato;
  - $v_k \in \mathbb{R}^{m \times 1}$  é il vettore del rumore di misura, e rappresenta la differenza tra il vettore delle misure ed il valore che le uscite avrebbero in conseguenza del solo valore attuale dello stato; é un rumore
-

gaussiano a valore medio nullo con matrice di covarianza  $R_k$ , ovvero  $v_k \sim N(0, R_k)$ ;

- $R_k = cov(v_k) = \mathbb{E}[v_k v_k^T] \geq 0 \in \mathbb{R}^{m \times m}$ , é la matrice di covarianza del vettore del rumore di misura, semidefinita positiva, e rappresenta la potenza del disturbo sulla misura;
- $H_k \in \mathbb{R}^{m \times n}$  é la matrice di sensibilità della misura, e descrive il valore assunto dalle variabili misurate in funzione del valore attuale della variabile di stato.

Definiamo inoltre delle quantità necessarie per sviluppare il filtro:

- $x_0 \in \mathbb{R}^{n \times 1}$ , é il vettore di stato iniziale che dovremo inizializzare nel filtro ( é la posizione di partenza del vettore stima);
  - $P_0 = cov(x_0) \in \mathbb{R}^{n \times n}$  é la matrice di covarianza del vettore di stato iniziale, semidefinita positiva (all'inizio bisogna inizializzare anch'essa, per semplicitá il valore iniziale scelto é una matrice di zeri);
  - $\hat{x}_{k|k-1} = \mathbb{E}[x_k | z_{k-1}, \dots, z_0, x_0] \in \mathbb{R}^{n \times 1}$ , é la stima a priori del vettore di stato all'istante  $k$  date le osservazioni fino all'istante  $k-1$  e lo stato iniziale  $x_0$ ;
  - $\hat{x}_{k|k} = \mathbb{E}[x_k | z_k, \dots, z_0, x_0] \in \mathbb{R}^{n \times 1}$ , é la stima a posteriori del vettore di stato all'istante  $k$  date le osservazioni fino all'istante  $k$  e lo stato iniziale  $x_0$ ;
  - $e_{k|k-1} = x_k - \hat{x}_{k|k-1} \in \mathbb{R}^{n \times 1}$ , é l'errore di stima a priori del vettore di stato;
  - $e_{k|k} = x_k - \hat{x}_{k|k} \in \mathbb{R}^{n \times 1}$ , é l'errore di stima a posteriori del vettore di stato;
  - $P_{k|k-1} = cov(e_{k|k-1}) = \mathbb{E}[e_{k|k-1} e_{k|k-1}^T | z_{k-1}, \dots, z_0, x_0] \geq 0 \in \mathbb{R}^{n \times n}$ , é la matrice di covarianza dell'errore di stima a priori del vettore di stato, semidefinita positiva;
  - $P_{k|k} = cov(e_{k|k}) = \mathbb{E}[e_{k|k} e_{k|k}^T | z_k, \dots, z_0, x_0] \geq 0 \in \mathbb{R}^{n \times n}$ , é la matrice di covarianza dell'errore di stima a posteriori del vettore di stato, semidefinita positiva;
-

- $K_k \in \mathbb{R}^{m \times n}$ , é la matrice del guadagno di Kalman e indica il livello di fiducia assegnato alla misura rispetto a quello assegnato alla stima dello stato;
- $\tilde{y}_k \in \mathbb{R}^{m \times 1}$ , é il vettore del residuo di misura (o innovazione), e rappresenta la differenza tra la misura attuale  $z_k$  e una previsione della misura data dal prodotto  $H_k \hat{x}_{k|k-1}$ ;
- $S_k = cov(\tilde{y}_k) = \mathbb{E}[\tilde{y}_k \tilde{y}_k^T] \geq 0 \in \mathbb{R}^{m \times m}$ , é la matrice di covarianza del vettore del residuo di misura, semidefinita positiva.

Per chiarimenti sulla definizione di matrice di covarianza e di processo stocastico, si vedano gli appendici A.1 e A.3.

## 2.3 Algoritmo ricorsivo

Il filtro di Kalman si basa su un meccanismo ricorsivo composto da due fasi fondamentali che sono la predizione e correzione.

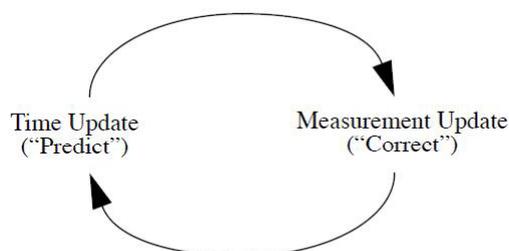


Figura 2.1: Schema dell'algoritmo ciclico ricorsivo di Predizione-Correzione.

Dobbiamo inoltre fornire al sistema uno stato iniziale che serve per inizializzare il vettore stima.

### 2.3.1 Inizializzazione

Come detto dobbiamo scegliere il vettore  $x_{k|k}$  all'istante iniziale ovvero  $x_{0|0}$ . Questa scelta nel caso di singolo target può essere arbitraria, nel senso che in base alle caratteristiche del vettore di stato vedremo che prima o

---

poi il filtro inizia ad inseguire e ad agganciarsi al target. L'inizializzazione deve essere svolta anche per  $P_{0|0}$  che rappresenta la covarianza dell'errore stima iniziale.

### 2.3.2 Predizione

Siamo all'istante  $k + 1$ , nel quale é arrivata una nuova misura. Prima di confrontarci con essa bisogna effettuare la stima  $x_{k+1|k}$  poiché il target fra l'istante  $k$  e l'istante  $k + 1$  si é spostato (si prende la stima a posteriori dell'istante  $k$ ) seguendo, nella teoria, l'andamento della matrice  $F$ . In altre parole si esegue l'equazione (2.1):

$$x_{k+1|k} = F_k x_{k|k} + B_k u_k \quad (2.3)$$

In questa fase bisogna fare anche la stima sulla covarianza del rumore la quale servirá per il calcolo del guadagno di Kalman:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \quad (2.4)$$

### 2.3.3 Correzione

Una volta effettuata la stima della traccia all'istante  $k + 1$  questa deve essere corretta in funzione della misura in arrivo dal blocco di Detection. Nel simulatore realizzato il blocco di Detection verrà simulato attraverso *misure vere* sporcate da rumore in modo da simulare il processo reale. La misura richiama l'equazione (2.2), inoltre verrà calcolato il residuo  $y_{k+1}$  e la covarianza del residuo  $S_{k+1}$  nel seguente modo:

---

$$\tilde{y}_{k+1} = z_{k+1} - H_{k+1} x_{k+1|k}. \quad (2.5)$$

Tale residuo se il modello é accurato deve soddisfare il seguente criterio:

$$\mathbb{E}[\tilde{y}_{k+1}] = 0. \quad (2.6)$$

Definiamo anche la *covarianza del residuo di misura*  $S_{k+1}$ :

$$S_{k+1} = \text{cov}(\tilde{y}_{k+1}) = H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}. \quad (2.7)$$

## 2.4 Stazionarietà del filtro

Alcuni parametri come le matrici  $F_k$ ,  $H_k$ ,  $B_k$ ,  $R_k$ ,  $Q_k$ , possono essere ritenute costanti durante il filtraggio, in questo caso si parla di processo stazionario. Il guadagno di Kalman risulta essere comunque tempo-variante perciò definire costanti quelle matrici permette di ridurre la complessità dell'algoritmo poiché esse possono essere calcolate una volta sola come per esempio, nel filtro in questione, in cui esse sono state calcolate all'inizio. Naturalmente queste sono ipotesi, poiché per esempio si suppone che il moto del target non cambia apprezzabilmente nel tempo (per esempio se la matrice  $F$  descrive un moto rettilineo si suppone che nel tempo l'oggetto mantenga più o meno questo moto e non inizi a variare sensibilmente). Discorsi analoghi si possono fare per le matrici  $Q$  e  $R$ , per le quali non ha senso modificare quei valori durante il filtraggio poiché si presume che il rumore introdotto dal sensore che effettui la misura sia mediamente lo stesso per ogni misura e che quindi  $Q$  non cambi apprezzabilmente.

---

## 2.5 Formulazione di Bayes del Filtro di Kalman

Come abbiamo detto in precedenza, il filtro di Bayes opera su delle densità di probabilità. Possiamo mettere a confronto l'equazione del modello (2.1) per il filtro di Kalman con quella relativa al modello per il filtro di Bayes:

$$f_{k+1|k}(x|x') = N_{Q_k}(x - F_k x') \quad (2.8)$$

dove  $f_{k+1|k}(x|x')$  rappresenta una funzione densità di probabilità della stima del vettore di stato dato lo stato  $x'$  precedente, come una Gaussiana di matrice di covarianza  $Q_k$ , con valor medio  $F_k x'$  che rappresenta la stima. Questa funzione sarà *piccola* se  $x$  è distante dalla stima predetta altrimenti avrà un valore alto.

Allo stesso modo possiamo definire la funzione densità che ingloba le caratteristiche del vettore di stato e della matrice di covarianza dell'errore come:

$$f_{k|k}(x|Z^k) = N_{P_{k|k}}(x - x_{k|k}) \quad (2.9)$$

che rappresenta la funzione densità di probabilità all'istante  $k$  del vettore di stato  $x$  condizionato dal vettore delle misure fino all'istante stesso, cioè in base alla misura ricevuta trova la confidenza che essa ha con il vettore di stato. A partire dalla funzione  $f_{k|k}(x'|Z^k)$  che rappresenta la p.d.f (Probability Density Function) [5] che il target abbia lo stato  $x'$  data la misura, all'istante  $k$ , il prodotto  $f_{k+1|k}(x|x') \cdot f_{k|k}(x'|Z^k)$  è la p.d.f che il target abbia vettore di stato  $x$  al tempo  $k + 1$  a partire dal vettore di stato  $x'$  all'istante  $k$  condizionato dalle misure fino quell'istante. Integrando tutti gli stati a priori  $x'$  si ottiene la p.d.f totale di  $f_{k+1|k}(x)$  che rappresenta il vettore di stato  $x$  del target all'istante  $k + 1$ .

---

Si può dimostrare che [7]

$$f_{k+1|k}(x) = f_{k+1|k}(x|Z^k) \quad (2.10)$$

rappresenta una formulazione matematica equivalente della predizione del filtro di Kalman in termini di probabilità.

Per quanto riguarda la formulazione dell'equazione di correzione con Bayes rispetto all'equazione (2.2) si ha che:

$$f_{k+1}(z|x) = N_{R_{k+1}}(z - H_{k+1}x) \quad (2.11)$$

detta *likelihood function* che indica la p.d.f della misura dato il vettore di stato come una congiunta gaussiana con matrice di covarianza  $R_{k+1}$  e con valor medio  $H_{k+1}x$ .

Lo scopo è quello di calcolare una p.d.f a posteriori, che avrà la seguente espressione:

$$f_{k+1|k+1}(x|Z^{k+1}) = \frac{f_{k+1}(z_{k+1}|x)f_{k+1|k}(x|Z^k)}{f_{k+1}(z_{k+1}|Z^k)}. \quad (2.12)$$

Possiamo notare al numeratore il primo membro è una p.d.f che rappresenta una fiducia da attribuire alla misura nuova dato il vettore di stato e al secondo membro la p.d.f calcolata prima che rappresenta la stima (2.10). Al denominatore vediamo invece il fattore di normalizzazione di Bayes che tiene conto di tutte le osservazioni fatte fino a quell'istante.

Le equazioni (2.10), (2.12) definiscono per un singolo sensore, singolo target, il filtro ricorsivo di Bayes. Attraverso criteri quali il metodo *MAP*

---

(Maximum a Posteriori) e criterio *MLE* (Maximum Likelihood Estimator) il filtro eseguirá la stima sul vettore di stato. [7].

### 2.5.1 Riassumendo

Possiamo mettere a confronto le fasi di predizione e correzione mediante il filtro di Kalman (2.13) e quello di Bayes (2.14), evidenziando gli step fondamentali:

$$x_{k|k} \xrightarrow{\text{predizione}} x_{k+1|k} \xrightarrow{\text{correzione}} x_{k+1|k+1} \quad (2.13)$$

$$f_{k|k} \xrightarrow{\text{predizione}} f_{k+1|k} \xrightarrow{\text{correzione}} f_{k+1|k+1} \quad (2.14)$$

## 2.6 Tracking Singolo Target

É stato sviluppato un simulatore che potesse interfacciarsi il piú possibile con l'utente in modo da soddisfare diversi tipi di simulazioni. L'area da sorvegliare é stata scelta come quadrata di lato 500 metri ma che puó essere cambiata all'occorrenza. Vediamo alcune possibili traiettorie *vere* percorse dal target:

---

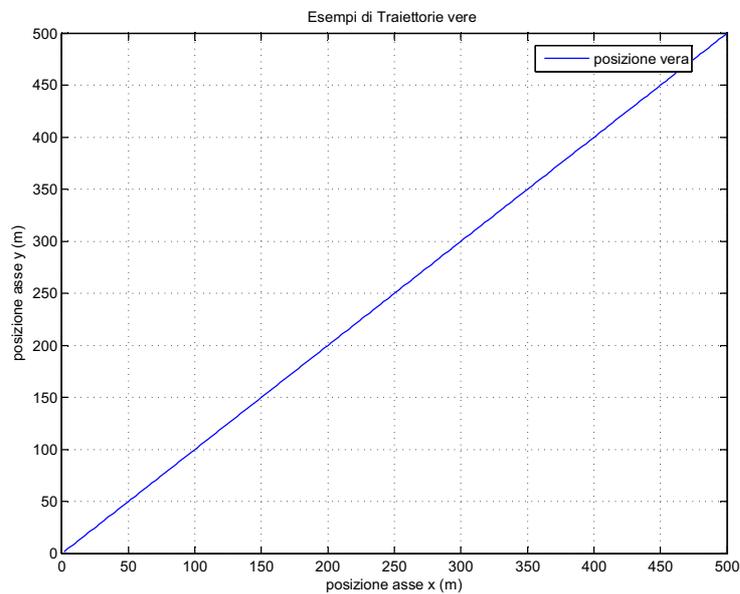


Figura 2.2: Traiettoria rettilinea con  $x$  e  $y$  crescenti, si é considerata una simulazione di 400 s con passo  $dt$  unitario.

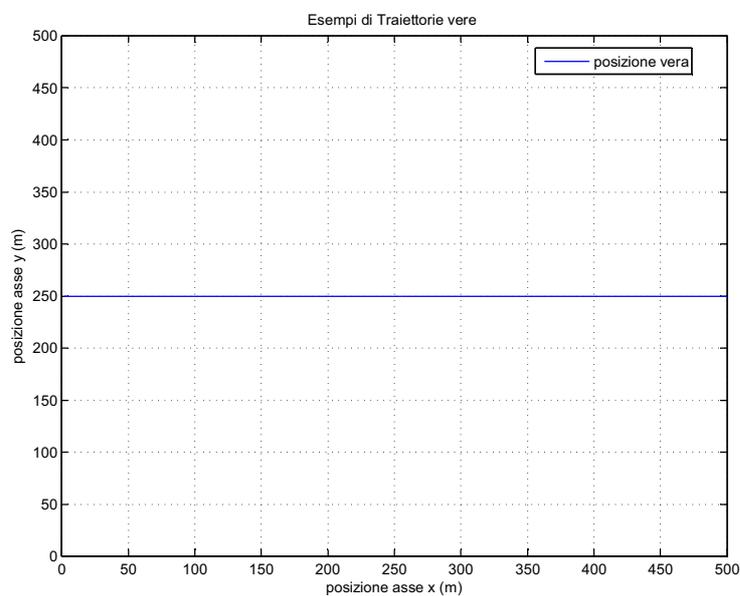


Figura 2.3: Traiettoria rettilinea con  $y$  costante, si é considerata una simulazione di 400 s con passo  $dt$  unitario.

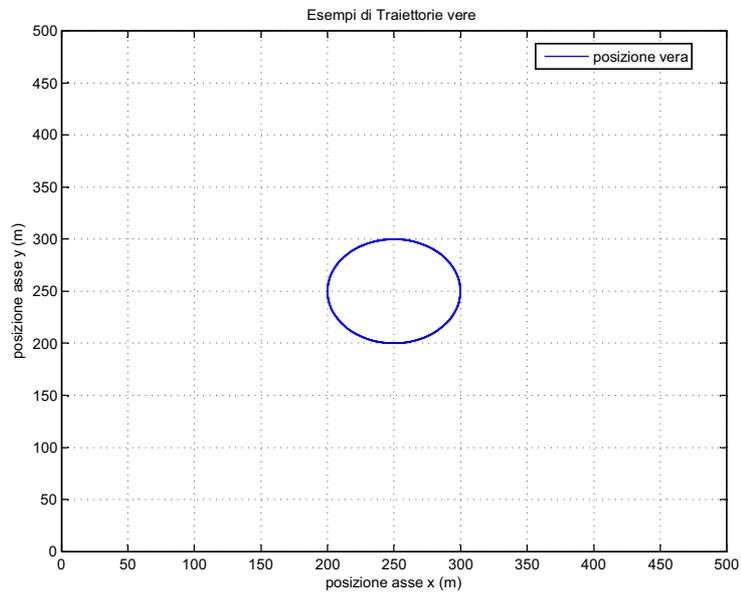


Figura 2.4: Traiettoria circolare centrata in  $x = 250 (m)$ ,  $y = 250 (m)$ , di raggio ( $Radius$ ) =  $50 (m)$ , con  $w = 0.1 rad/s$  e sfasamento nullo.

### 2.6.1 Esempio di veicolo in due dimensioni

Consideriamo un veicolo che si muove di velocità costante all'interno dell'area da sorvegliare, con accelerazione nulla (di conseguenza la matrice  $B$  in questo caso non influenza il moto). Ipotizziamo che esso entri nell'area stessa a partire da una cordinata  $(x, y)$  che può essere impostata a piacere e che nella descrizione del moto seguente corrisponderà  $(p_{0_x}, p_{0_y})$ . Il modello del moto non è perfetto poiché la velocità non è perfettamente costante causa per esempio dell'attrito. Assumiamo pertanto che le possibili variazioni di velocità (il moto è rettilineo uniforme) lungo i due assi  $x$  e  $y$ , siano da considerarsi dei rumori modellati da due componenti aleatorie  $(a_{k_x}, a_{k_y}) \in \mathbb{R}$  Gaussiane incorrelate ed identicamente distribuite (i.i.d) a valore medio nullo con deviazione standard  $\sigma_a$  che nelle simulazioni successive abbiamo scelto pari a  $0.2 \text{ m/s}$ . Tali componenti  $(a_{k_x}, a_{k_y})$  rappresentano la variazione aleatoria di velocità del veicolo dall'istante  $k - 1$  all'istante  $k$ , lungo i due assi  $x$  e  $y$ . Sappiamo inoltre che la posizione del veicolo è misurata con i rumori di misura  $(v_{k_x}, v_{k_y}) \in \mathbb{R}$  Gaussiani incorrelati ed identicamente distribuiti (i.i.d), a valore medio nullo con deviazione standard  $\sigma_z$  anch'essa impostabile.

Le equazioni del moto di Newton riferite all'asse  $x$  per il caso in questione, sono la posizione dell'auto  $p_{k_x}$  all'istante  $k$ , e la velocità dell'auto  $\dot{p}_{k_x}$  all'istante  $k$ :

$$p_{k_x} = p_{k-1_x} + dt \dot{p}_{k-1_x} \quad \dot{p}_{k_x} = \dot{p}_{k-1_x} + a_{k-1_x} \quad (2.15)$$

La velocità come detto è soggetta ad una componente aleatoria perciò la posizione lungo  $x$  sarà:

$$p_{k_x} = p_{k-1_x} + dt \dot{p}_{k-1_x} + dt a_{k-1_x}. \quad (2.16)$$

Per l'asse  $y$  in modo analogo avremo:

---

$$p_{k,y} = p_{k-1,y} + dt \dot{p}_{k-1,y} + dt a_{k-1,y} \quad \dot{p}_{k,y} = \dot{p}_{k-1,y} + a_{k-1,y}. \quad (2.17)$$

Possiamo quindi definire il vettore di stato  $x_k \in \mathbb{R}^{4 \times 1}$  nel modo seguente:

$$x_k = \begin{bmatrix} p_{k,x} \\ p_{k,y} \\ \dot{p}_{k,x} \\ \dot{p}_{k,y} \end{bmatrix}$$

Ricordando le espressioni del modello di processo (2.1) e del modello di misura (2.2), per la dinamica del moto del target in questione:

$$x_k = \underbrace{\begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_F \cdot x_{k-1} \quad (2.18)$$

$$z_k = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_H \cdot x_k + \underbrace{\begin{bmatrix} v_{k,x} \\ v_{k,y} \end{bmatrix}}_{v_k} \quad (2.19)$$

dove  $F \in \mathbb{R}^{4 \times 4}$  é la matrice di transizione degli stati,  $H \in \mathbb{R}^{2 \times 4}$  é la matrice di sensibilità della misura,  $z_k \in \mathbb{R}^{2 \times 1}$  é la posizione misurata del veicolo lungo l'asse  $x$  e l'asse  $y$  (eventualmente si potrebbe modificare  $H$  in modo che la misura tenga conto anche di informazioni sulla velocità lungo  $x$  e lungo  $y$ ), mentre  $v_{k,x}$  e  $v_{k,y}$  rappresentano i rumori lungo l'asse  $x$  e l'asse  $y$  che influenzano la misura. La matrice di covarianza del rumore di processo, ritenendo incorrelate le variabili aleatorie fra l'asse  $x$  e l'asse  $y$ , considerando il caso stazionario, risulta essere  $Q \in \mathbb{R}^{4 \times 4}$ :

---

$$Q = \mathbb{E} \begin{bmatrix} a_x^2 dt^2 & 0 & a_x^2 dt & 0 \\ 0 & a_y^2 dt^2 & 0 & a_y^2 dt \\ a_x^2 dt & 0 & a_x^2 & 0 \\ 0 & a_y^2 dt & 0 & a_y^2 \end{bmatrix} = \sigma_a^2 \begin{bmatrix} dt^2 & 0 & dt & 0 \\ 0 & dt^2 & 0 & dt \\ dt & 0 & 1 & 0 \\ 0 & dt & 0 & 0 \end{bmatrix}.$$

La matrice di covarianza di  $R$  risulta essere:

$$R = \text{cov}(v) = \mathbb{E} \begin{bmatrix} v_x^2 & 0 \\ 0 & v_y^2 \end{bmatrix} = \sigma_z^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Ci rimane da inizializzare il vettore di stato per dire all'algoritmo da quale posizione iniziale partire:

$$\hat{x}_{0|0} = \begin{bmatrix} p_{0x} \\ \dot{p}_{0x} \\ p_{0y} \\ \dot{p}_{0y} \end{bmatrix}$$

Nelle nostre simulazioni abbiamo cambiato le condizioni di partenza del filtro per vedere il suo comportamento.

Allo stesso modo dobbiamo inizializzare la matrice di covarianza dell'errore, la quale inizialmente é sempre stata supposta una matrice di zeri.

$$P_{0|0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

## 2.6.2 Risultato delle Simulazioni

Mostriamo il risultato di due diverse simulazioni nelle quali abbiamo impostato una diversa deviazione di rumore  $\sigma_z$ .

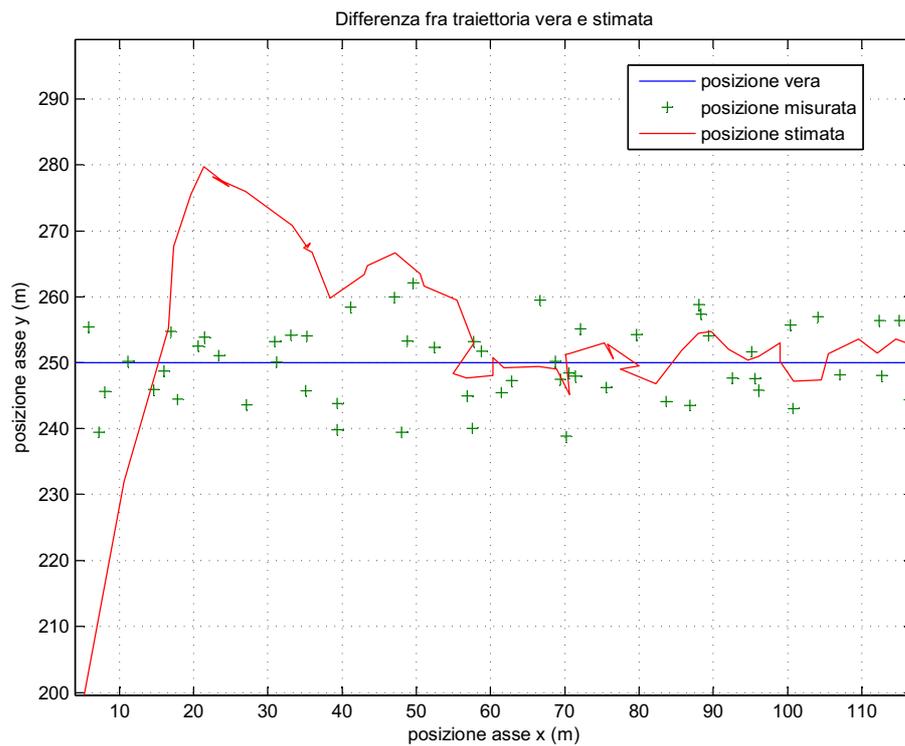


Figura 2.5: Risultato della simulazione per Traiettoria rettilinea con  $y = 250\text{ m}$ , assumendo  $\sigma_z = 5\text{ m}$ , velocità del target  $vel = 2\text{ m/s}$ , passo temporale di scansione  $dt = 1$ .

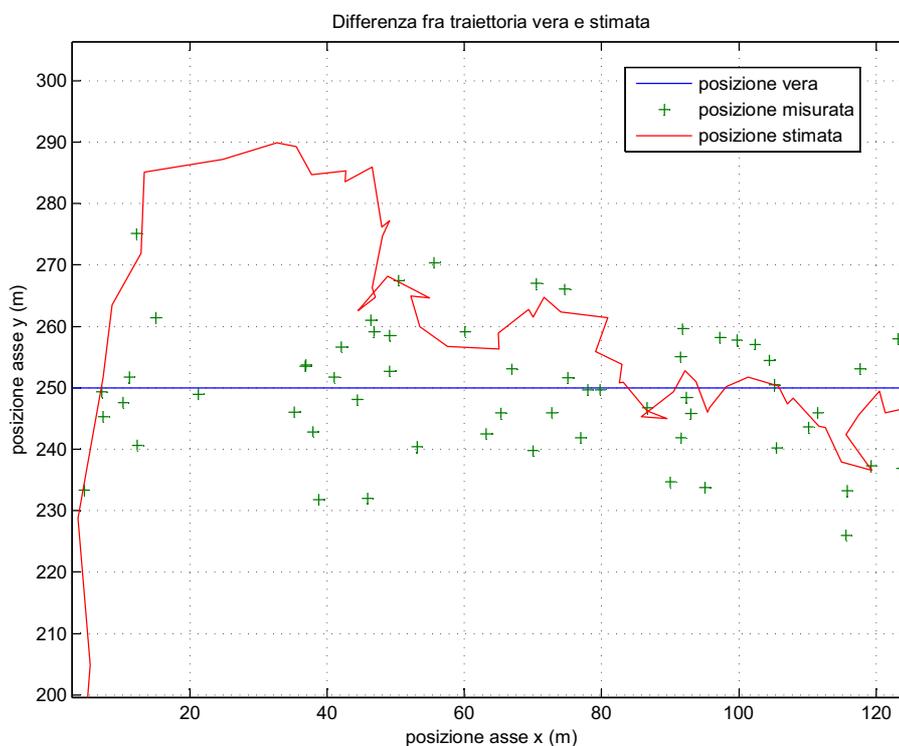


Figura 2.6: Risultato della simulazione per Traiettoria rettilinea con  $y = 250\text{ m}$ , assumendo  $\sigma_z = 10\text{ m}$ , velocità del target  $vel = 2\text{ m/s}$ , passo temporale di scansione  $dt = 1$ .

Si può notare come in Figura 2.6 il filtro si agganci ad una quota  $x$  maggiore rispetto alla figura Figura 2.5 nella pagina precedente. Questo è dipeso da una deviazione sul rumore di misura maggiore nel secondo caso rispetto al primo poiché il filtro avrebbe bisogno, a parità di velocità del target, di maggiori misure oppure un passo temporale più corto per agganciarsi prima. In entrambi i casi lo stato iniziale del filtro è stato fatto partire da una quota  $p_{0_x} = p_{0_y} = 0$ . Si può notare come appena incroci la traiettoria reale, la traccia generata dal filtro venga riportata verso il basso per effetto delle misure.

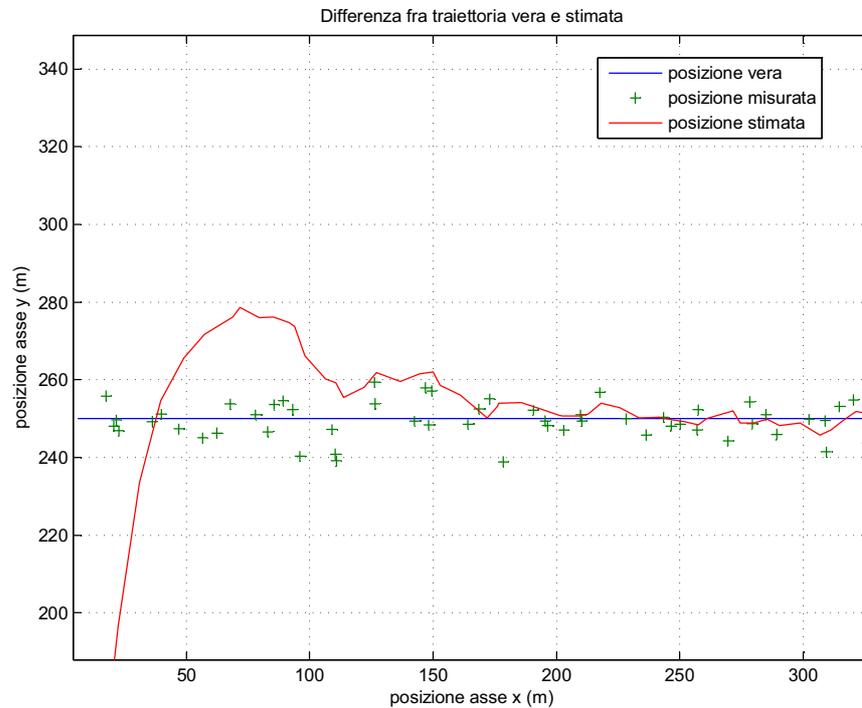


Figura 2.7: Risultato della simulazione per Traiettoria rettilinea con  $y = 250$  m, assumendo  $\sigma_z = 5$  m, velocità del target = 6 m/s, passo temporale di scansione  $dt = 1$  s.

Dalla Figura 2.7 si può osservare come a causa di una velocità elevata rispetto al passo temporale, il filtro si aggancia molto in ritardo nonostante una  $\sigma_z$  non eccessiva (il filtro per correggersi ha bisogno di misure). Una volta agganciato, il filtro segue bene il target.

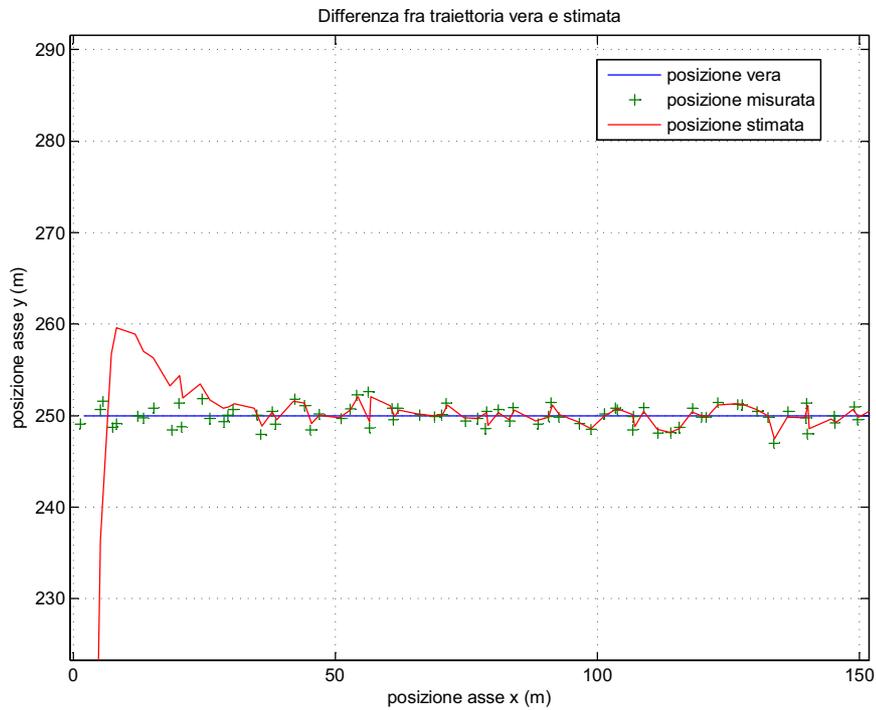


Figura 2.8: Risultato della simulazione per Traiettoria rettilinea con  $y = 250\text{ m}$ , assumendo  $\sigma_z = 1\text{ m}$ , velocità del target  $vel = 2\text{ m/s}$ , passo temporale di scansione  $dt = 1$ .

In Figura 2.8 si vede come una  $\sigma_z$  e una velocità del target bassa fanno sì che il filtro si agganci presto ed in modo efficiente alla traiettoria vera, seguendo il target lungo il suo moto, è sicuramente la situazione migliore. Vediamo ora un'esempio di traiettoria crescente sia in  $x$  che  $y$ . Andiamo a graficare due diverse simulazioni tenendo conto di diversi valori della deviazione del rumore di misura  $\sigma_z$ .

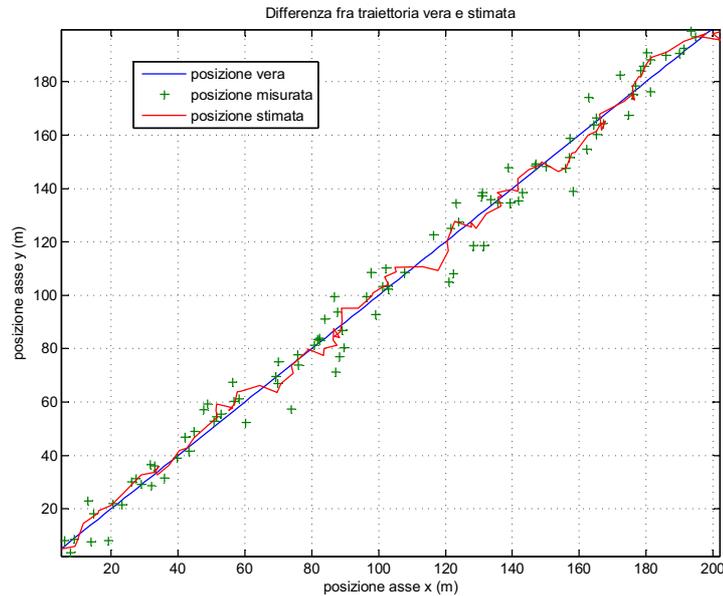


Figura 2.9: Risultato della simulazione per Traiettoria rettilinea con  $x$  e  $y$  variabili, assumendo  $\sigma_z = 5 m$ , velocità del target  $vel = 2m/s$ , passo temporale di scansione  $dt = 1 s$ .

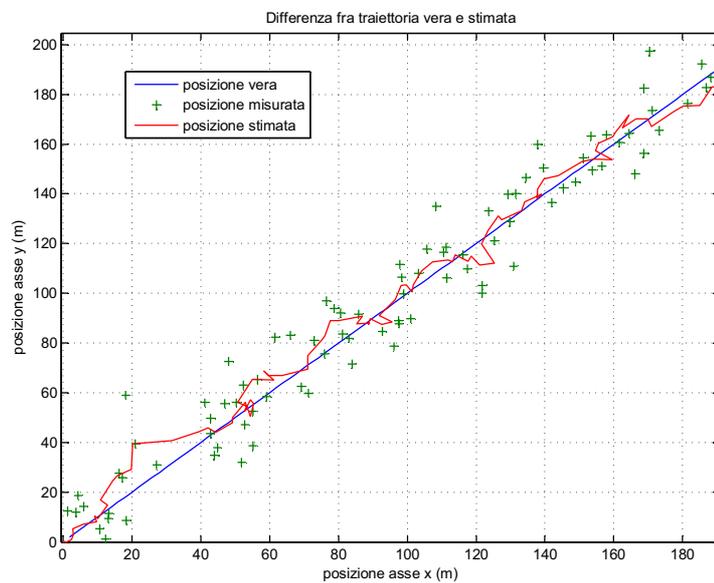


Figura 2.10: Risultato della simulazione per Traiettoria rettilinea con  $x$  e  $y$  variabili, assumendo  $\sigma_z = 10 m$ , velocità del target  $vel = 2m/s$ , passo temporale di scansione  $dt = 1 s$ .

Causa di una diversa deviazione standard del rumore si vede come l'inseguimento sia ostacolato da una dispersione maggiore delle misurazioni, presente in Figura 2.10 a fronte.

Vengono di seguito riportati altri esempi con traiettorie circolari. Abbiamo simulato diverse situazioni per le quali cambi o la posizione iniziale del filtro di Kalman oppure la deviazione del rumore di misura  $\sigma_z$ .

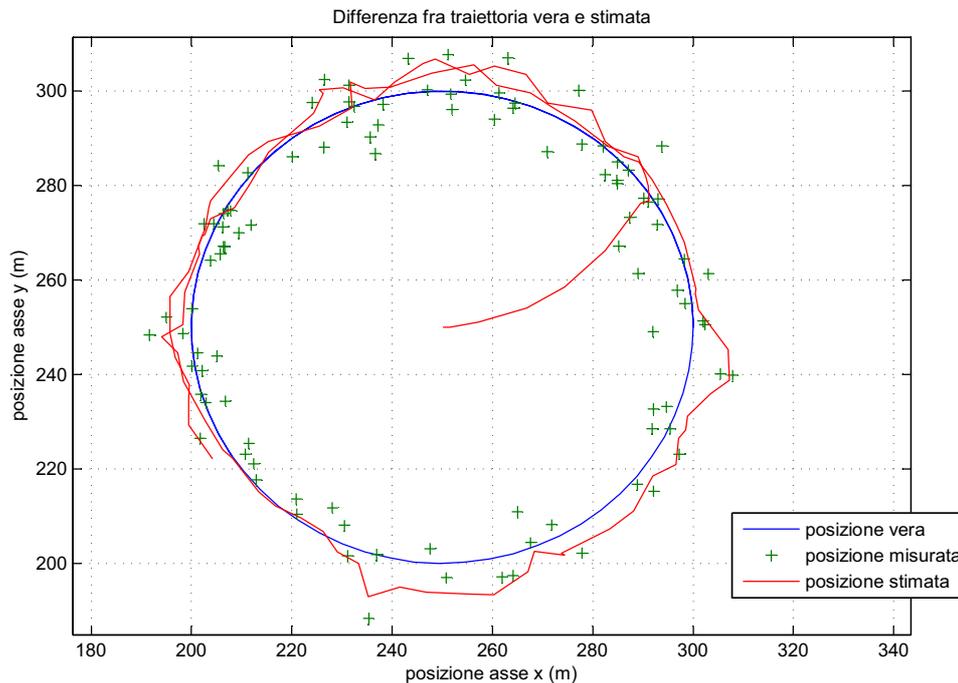


Figura 2.11: Risultato della simulazione per Traiettoria circolare, assumendo  $\sigma_z = 5 m$ , pulsazione  $w = 0.1 rad/sec$ , durata della simulazione pari a 100 s, sfasamento iniziale nullo, passo temporale di scansione  $dt = 1 s$ , scelta del vettore iniziale di coordinate  $p_{0_x} = p_{0_y} = 250 (m)$ .

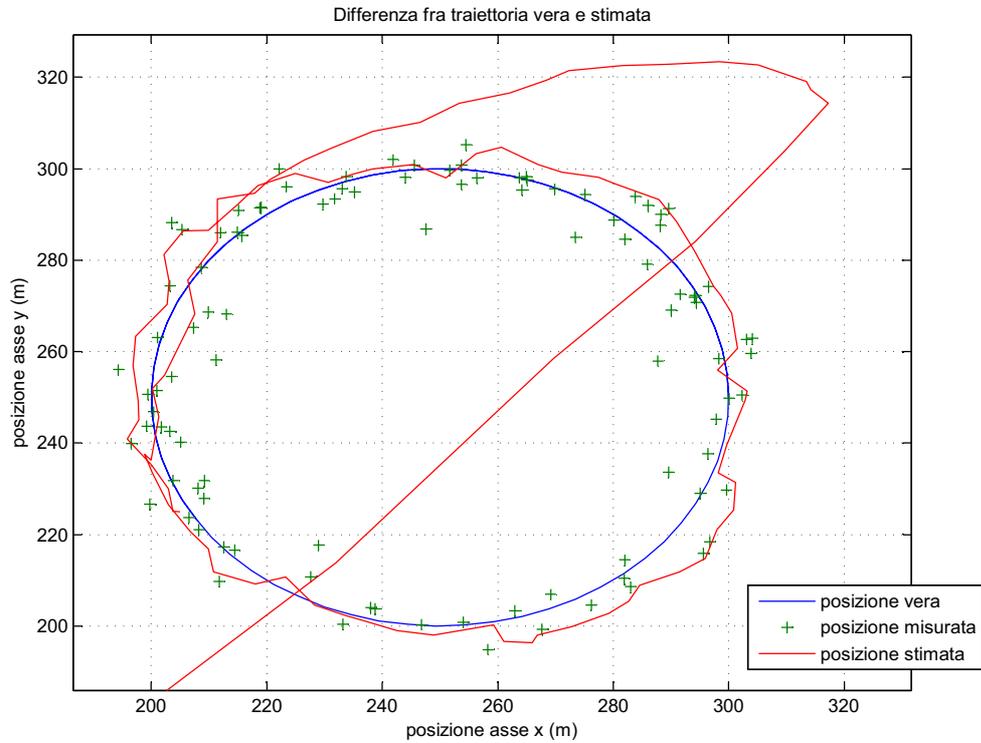


Figura 2.12: Risultato della simulazione per Traiettoria circolare, assumendo  $\sigma_z = 5\text{ m}$ , pulsazione  $w = 0.1\text{ rad/sec}$ , durata della simulazione pari a 100 s, sfasamento iniziale nullo, passo temporale di scansione  $dt = 1\text{ s}$ , scelta del vettore iniziale di coordinate  $p_{0_x} = p_{0_y} = 0\text{ (m)}$ .

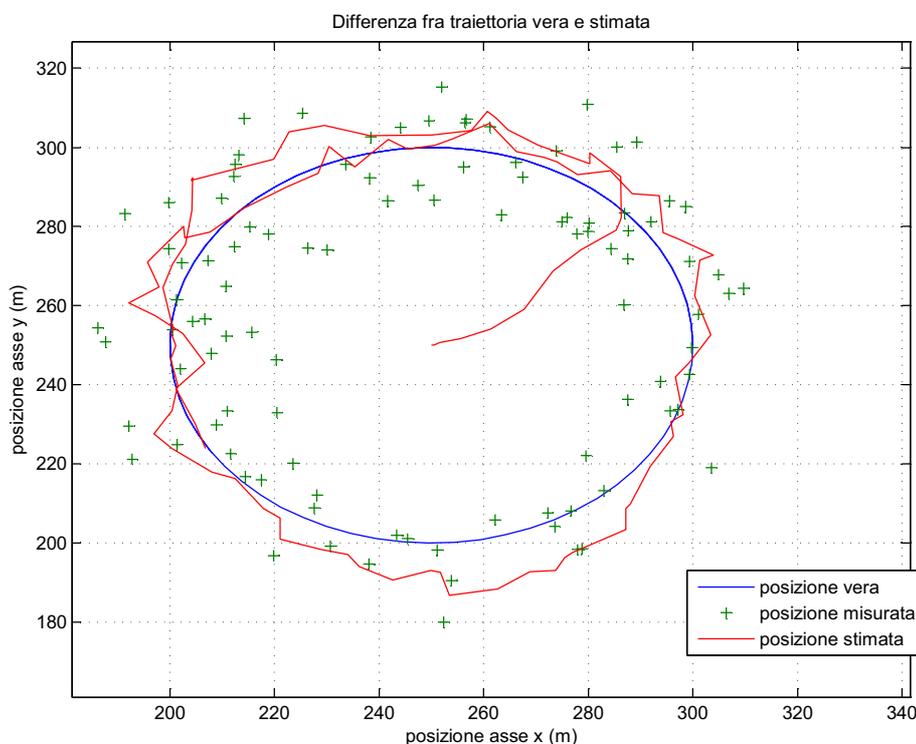


Figura 2.13: Risultato della simulazione per Traiettoria circolare, assumendo  $\sigma_z = 10\text{ m}$ , pulsazione  $w = 0.1\text{ rad/sec}$ , durata della simulazione pari a  $100\text{ s}$ , sfasamento iniziale nullo, passo temporale di scansione  $dt = 1\text{ s}$ , scelta del vettore iniziale di coordinate  $p_{0_x} = p_{0_y} = 250\text{ (m)}$ .

Dalla Figura 2.12 nella pagina precedente si può osservare come il filtro tenda a seguire un moto rettilineo dando maggior peso alla matrice  $F$  e, ad un certo punto, accorgersi del fatto che il target sia rimasto indietro. Una volta agganciato da più peso alle misure. In entrambi i casi (di Figura 2.11 a pagina 31 e Figura 2.12 nella pagina precedente) si può osservare che nonostante il filtro lavori con un modello dinamico di tipo 'moto rettilineo', esso si comporti comunque abbastanza bene, agevolato dal fatto che la dispersione delle misure non è eccessiva. In Figura 2.13 si osserva una maggiore dispersione delle misure, causata da una deviazione di rumore  $\sigma_z$  maggiore rispetto ai due casi precedenti. Essa fa sì che la traccia generata dal filtro è peggiore rispetto ai due casi precedenti.



# Capitolo 3

## Tracking Multitarget

### 3.1 Introduzione all'approccio Multitarget

Lo scopo dell'elaborato consiste nell'implementare un sistema di tracking per l'inseguimento di piú oggetti all'interno dell'area da sorvegliare.

Successivamente alla rimozione del clutter ogni ricevitore crea una matrice (mappa soft) di pixel, il cui valore rappresenta la probabilità di trovare il target in quel pixel misurata dal ricevitore. Siccome ci saranno piú oggetti all'interno di quest'area, ogni ricevitore avrà piú pixel a probabilità alta (esempio: nel caso di un buon filtraggio mi aspetto tre pixel ad alta probabilità nella situazione in cui abbia tre target dentro l'area).

Una volta create le mappe soft, l'immagine  $I(x, y)$  é ottenuta attraverso una combinazione non lineare di esse. Il blocco di detection, a questo punto, riceve all'ingresso l'immagine e decide in base ai punti di probabilità piú alta, prendendo come misura della posizione del target generico la cordinata  $z(i) = (z_x(i), z_y(i))$ , che corrisponde ad un  $i$ -esimo pixel di valore alto, secondo opportuni algoritmi non approfonditi nella presente tesi.

Durante lo svolgimento di questo elaborato é stato supposto che il blocco che esegue la detection sia ideale, cioé che il sistema fornisca al blocco di tracking un numero di misure corrispondenti al numero di tracce evitando quindi problemi di *False Alarms* e di *Missed Detections*. Ci si é invece concentrati sul comprendere come gestire un tracking multitarget di oggetti nascenti (*Target Birth*)(esempio: passaggio da un'unica traccia a piú

---

tracce), e morenti (*Target Death*)(esempio contrario) all'interno dell'area da sorvegliare, partendo dal risolvere il problema di associazione delle tracce con le misure ricevute. Questo é stato uno degli aspetti fondamentali, poiché il blocco di detection fornisce le misure ma é compito del filtro di tracking capire come associare queste alla traccia  $i$ -esima.

Assumeremo che a ciascun target siano applicabili equazioni del tipo (2.1) e (2.2) , cioè le equazioni di Kalman per il singolo target. Ad ogni istante avremo, per il modello di misura l'equazione del tipo:

$$Z_i = H x_i + v_i \quad (3.1)$$

dove  $Z_i$  é la misura generata dall' $i$ -esimo target presente in quell'istante, per  $i = 1, \dots, n$  target e dove  $v_i$  é il rumore, Gaussiano, a valor medio nullo con matrice di covarianza  $R$ . Siccome siamo nell'ipotesi in cui non ci sono Missed Detection e False Alarms avremo un numero di elementi del vettore  $Z$  ad ogni istante che sará pari al numero delle tracce in quell'istante, cioè:

$$Z = \{z_1, \dots, z_n\} \quad (3.2)$$

Per quanto riguarda il modello di moto abbiamo assunto che:

- Ogni target sia governato dall'equazione di Kalman del tipo (2.1);
- Ogni target si muova con moto statisticamente indipendente rispetto agli altri;
- Nuovi target appaiano sulla scena mediante una distribuzione di Poisson nel tempo, mentre in modo uniforme nella posizione.

Il modello che tiene conto di eventuali *nascite* o *morti* é il seguente:

$$X = X(x'_1) \cup \dots \cup X(x'_{n'}) \cup \{(b_1, \dots, b_\nu)\} \quad (3.3)$$


---

dove  $X$  rappresenta il vettore di predizione totale degli stati,  $X(x'_i)$  può assumere valore nullo (scomparsa dell' $i$ -esimo target) oppure valore  $x_i$  (valore predetto) nel passaggio da un'istante temporale a quello successivo, mentre  $(b_1, \dots, b_\nu)$  rappresentano la comparsa di nuovi target in quell'istante.

## 3.2 Associazione Traccia-Misura

Per ipotesi, in ogni istante temporale il blocco di detection fornisce un numero di misure pari al numero di target presenti nella scena. Ciascuna di queste misure si riferiscono ad un target preciso e poiché il blocco di tracking opera con tanti filtri di Kalman in parallelo quante sono le tracce, è necessario associare al filtro corrispondente alla  $i$ -esima traccia la misura corretta. Si rende quindi necessario un algoritmo che date le misure le *diriga* nel filtro giusto. Si potrebbe pensare di usare, al fine di questa associazione, la distanza Euclidea, calcolare questa fra la generica misura ricevuta e la predizione di una generica traccia, valutando il modulo e scegliendo di associare la misura alla predizione più vicina. Questa scelta non è tuttavia ottima poiché occorre tenere conto in qualche modo sia della matrice  $R$ , sia dalla matrice  $P$ . Basti pensare che se  $R$  e  $P$  sono entrambe *piccole* rispetto al modulo della distanza euclidea fra la misura e la predizione generica, con buona probabilità la misura non è la responsabile di quella traccia nonostante il fatto che la distanza euclidea in modulo, sia minima. Vediamo l'algoritmo implementato nell'ambito della tesi.

### 3.2.1 Global Association Distance

Il metodo basato sulla *Global Association Distance* è uno di quelli più comunemente usati; esso tiene conto di tutte le possibili permutazioni fra le misure e le tracce. Fissata una possibile delle  $n!$  permutazioni (dove  $n$  è il numero di target), esso calcola l'association distance  $d$ , cioè valuta l'*influenza* della misura rispetto alla generica traccia stimata, in termini di densità di probabilità, lavorando con le matrici  $R$  e  $P$ . In base al criterio Maximum a Posteriori (MAP) tale distanza viene calcolata come [7]:

---

$$d(x'_i|z)^2 = (z - Hx'_i)^T (HP_iH^T + R)^{-1} (z - Hx'_i). \quad (3.4)$$

Vediamo come data la misura, la distanza  $d$  relativa alla predizione  $i$ -esima sia fortemente dipendente dalle matrici che condizionano il processo. Poiché fissata una permutazione ci sono  $n$  associazioni fra tracce e misure, la distanza globale  $d_\sigma^2$  calcolata su una singola permutazione, riprendendo la (3.4) é data da:

$$d_\sigma^2 = \sum_{i=1}^n d(z_{\sigma_i}, Hx'_i)^2 = \sum_{i=1}^n (z_{\sigma_i} - Hx'_i)^T (HP_iH^T + R)^{-1} (z_{\sigma_i} - Hx'_i) \quad (3.5)$$

L'associazione ottima risulterà essere quella che minimizza  $d_\sigma^2$ . Questi algoritmi aumentano notevolmente di complessità all'aumentare di  $n$  poiché il numero delle permutazioni é pari  $n!$ .

Esistono alcuni algoritmi quali ad esempio quelli descritti in [8] e [9] che tentano di massimizzare l'efficienza computazionale. Tali algoritmi sono tuttavia al di fuori degli scopi di questa tesi.

### 3.3 Single-Hypothesis Correlation

#### 3.3.1 Introduzione agli Algoritmi di Correlazione

L'approccio *SHC* utilizzato in questo elaborato é uno dei possibili metodi con cui il sistema di tracking multitarget effettua l'associazione fra misure e tracce, dando una certa *fiducia* all'associazione fatta. Più precisamente il filtro, una volta fatte le permutazioni e trovata la scelta che minimizza la distanza globale  $d_\sigma$ , associa la misura alla corrispondente traccia. Questa associazione potrebbe, sotto alcune condizioni, essere sbagliata. Pertanto, se il filtro avesse una sorta di memoria in modo da accorgersi a posteriori, dopo per esempio alcun scansioni, che la scelta in un determinato istante era stata sbagliata, esso sarebbe in grado di correggere la scelta, viceversa da quell'istante di tempo in poi il filtro potrebbe sbagliare l'associazione e per esempio invertire le tracce.

---

Gli algoritmi che effettuano l'associazione tra tracce e misure tenendo in memoria la scelta precedentemente fatta sono chiamati *Multihypotesis Correlation* (MHC). L'approccio SHC é un approccio diretto, effettua cioè un'associazione fra tracce e misure delle quali si perde istantaneamente la memoria. Vedremo attraverso simulazioni di varie casistiche come il filtro si comporti bene a patto che i target non siano molto vicini fra di loro e che il rapporto segnale-rumore (SNR) sia sufficientemente alto. Come possiamo dedurre, se ciò non si verificasse, si potrebbe compiere un errore dovuto per esempio a tracce molto vicine fra di loro e, una volta commesso, invertire irreversibilmente le tracce.

### 3.3.2 Algoritmo SHC

Con l'assunzione fatta precedentemente per la quale non possono verificarsi mancate rilevazioni e falsi allarmi, ci troviamo, a partire da un istante iniziale, fino all'istante  $k + 1$ , con una sequenza di misure  $Z_1, \dots, Z_{k+1}$ , costituite all'istante  $i$ -esimo dall'insieme di  $n$  misure  $Z_i = \{z_i^1, \dots, z_i^n\}$ . Definiamo a quest'istante una *track table*:

$$T_{k|k} = \{(l_{k|k}^1, x_{k|k}^1, P_{k|k}^1), \dots, (l_{k|k}^n, x_{k|k}^n, P_{k|k}^n)\} \quad (3.6)$$

Questa tabella contiene, per ogni traccia, l'identificazione della traccia stessa attraverso il parametro  $l$ , la stima sullo stato della traccia  $x_{k|k}^i$  e la relativa matrice di covarianza  $P_{k|k}^i$ . Ad ogni passo, il sistema a partire da una *Track Table* iniziale effettua una predizione e correzione in modo da arrivare ad una tabella a posteriori che contenga i valori aggiornati.

Per prima cosa si prende la tabella ad un certo istante  $k$  e si applicano le equazioni di predizione (2.3) e (2.4) la per ogni traccia:

$$x_{k+1|k}^i = F_k x_{k|k}^i \quad (3.7)$$

$$P_{k+1|k}^i = F_k P_{k|k}^i F_k^T + Q_k. \quad (3.8)$$

Il valore di  $l_{k|k}$  rimane invariato poiché esso identifica la traccia a cui sono associati i valori del vettore di stato e della matrice di covarianza  $i$ -esima. La predizione genera la track table seguente:

$$T_{k+1|k} = \{(l_{k+1|k}^1, x_{k+1|k}^1, P_{k+1|k}^1), \dots, (l_{k+1|k}^n, x_{k+1|k}^n, P_{k+1|k}^n)\}. \quad (3.9)$$

Nello step successivo si considera l'arrivo della misura. In questo caso bisogna decidere come effettuare l'associazione traccia-misura. Nel simulatore é stato implementato un algoritmo che scegliesse fra le varie permutazioni quella che consente di minimizzare la  $d_\sigma$ , vedi Capitolo 3.2.1.

Una volta determinata la misura associata alla traccia si passa infine alla correzione della track table:

$$x_{k+1|k+1}^i = x_{k+1|k}^i K_{k+1}^i \tilde{y}_{k+1}^i \quad (3.10)$$

$$P_{k+1|k+1}^i = (I - K_{k+1}^i H) P_{k+1|k}^i \quad (3.11)$$

dove  $I$  é la matrice identità e  $H$  é stata considerata nel caso stazionario.

Si arriva infine alla track table a posteriori che contiene i valori aggiornati:

$$T_{k+1|k+1} = \{(l_{k+1|k+1}^1, x_{k+1|k+1}^1, P_{k+1|k+1}^1), \dots, (l_{k+1|k+1}^n, x_{k+1|k+1}^n, P_{k+1|k+1}^n)\} \quad (3.12)$$

Questa tabella é ottenuta attraverso un'unica ipotesi di correlazione traccia-misura, come detto in precedenza. Si puó notare come, attraverso le equazioni usate, il filtro SHC possa essere considerato un sistema composto da  $n$  filtri di Kalman che lavorano in parallelo.

Di seguito, diamo una breve descrizione del filtro nel caso in cui avessimo False Alarm e Missed Detection.

### 3.3.3 SHC con False Misure e Mancati Allarmi

In questo caso il vettore delle misure puó contenere un numero di elementi diverso dal numero dei target, cioè il vettore  $Z_{k+1} = \{z_{k+1}^1, \dots, z_{k+1}^m\}$ , dove  $m \neq n$ . L'elemento generico  $z_{k+1}^i$  potrebbe essere associato a diverse casistiche:

- $z_{k+1}^i$  potrebbe essere la misura di una traccia che era già presente all'interno dall'area;
- $z_{k+1}^i$  potrebbe essere la misura generata da una nuova traccia che entra nel perimetro;
- $z_{k+1}^i$  potrebbe essere dovuta ad un qualche fenomeno che non si riconduce né a nuove tracce, né a tracce che si mantengono fra un'istante temporale precedente ed il successivo (esempio: é un falso allarme).

Il filtro SHC ha una logica interna, che permette di gestire, in base al tipo di applicazione richiesta, la presenza o meno di nuove misure. Se l'applicazione é del tipo *Real-Time*, dove la rilevazione del target é cruciale, la logica attribuirá alla nuova misura subito una traccia (esempio: settore bellico). Se l'applicazione invece si puó *permettere* di aspettare alcune scansioni allora la presenza di una nuova misura verrá gestita come un falso allarme e se, nelle scansioni successive continua a presentarsi, essa verrá inizializzata ad una nuova traccia.

## 3.4 Introduzione al Simulatore

Descriviamo brevemente alcuni parametri usati dal simulatore per il Radar Multistatico in ambiente Matlab.

---

Abbiamo supposto in quest'elaborato uno scenario di forma quadrata lungo le due dimensioni  $x, y$ . É possibile scegliere le dimensioni dell'area da sorvegliare, andando a modificare opportunamente le potenze di trasmissione in gioco. Nel nostro caso abbiamo scelto un'area quadrata di lato  $500 \times 500$  m ovvero costituito da un numero di pixel pari a  $2500 \times 2500$  poiché ogni pixel é un quadrato di lato  $d_{pix} = 20$  cm. Questo valore deriva da un trade-off di due fattori: Risoluzione massima ottenibile con gli impulsi UWB, e massima incertezza del sistema rispetto alla posizione vera del target tollerabile.

Vi sono un trasmettitore (TX) e tre ricevitori (RX) disposti ai bordi dell'area da sorvegliare. Il  $Tx$  ha coordinate (1,250) metri,  $Rx1$  ha coordinate (250,500),  $Rx2$  ha coordinate (500,250) e  $Rx3$  ha coordinate (250,1).

La velocità del target é impostabile a piacere; essa é indicata attraverso il comando *vel*. Il movimento dei target all'interno dell'area é rettilineo. Si possono scegliere diversi moti rettilinei (per esempio con  $x$  crescenti,  $x$  decrescenti oppure con le  $x$  costanti), attraverso i comandi *incrementa*, *decrementa* e *costante*. Si può scegliere il tempo della simulazione ed il passo temporale attraverso i comandi *durata* e *dt*.

Riassumiamo nella tabella successiva i principali parametri usati per descrivere l'ambiente circostante.

Parametro	Valore	Descrizione
$pixel\_size$	0.2	dimensioni del pixel $d_{pix}$ (m)
$Bmin$	0	ordinata minima del lato (m)
$Bmax$	500	ordinata massima del lato (m)
$Tx\_x, Tx\_y$	1, 250	coordinate del trasmettitore ( $x, y$ ) (metri)
$Rx1\_x, Rx1\_y$	250, 500	coordinate dei ricevitori ( $x, y$ ) (metri)
$Rx2\_x, Rx2\_y$	500, 250	
$Rx3\_x, Rx3\_y$	250, 1	

Tabella 3.1: Principali parametri utilizzati dal simulatore per descrivere l'ambiente.

Per caratterizzare il target é necessario l'inserimento di altri parametri, riassunti in Tabella 3.2 nella pagina successiva e Tabella 3.3 a fronte.

Parametro	Valore	Descrizione
<i>vel</i>	"a scelta"	velocità $\frac{1}{2}$ target (m/s)
$[x_{vero}, y_{vero}]$	"a scelta"	coordinata reale del target (metri)
<i>movimento</i>	"a scelta"	movimenti possibili lungo le traiettorie
	<i>incrementa</i>	se in linea retta con <i>x</i> crescenti
	<i>decrementa</i>	se in linea retta con <i>x</i> decrescenti
	<i>costante</i>	se in linea retta con <i>x</i> costante

Tabella 3.2: Principali parametri utilizzati dal simulatore per caratterizzare il target.

Parametro	Valore	Descrizione
<i>durata</i>	"a scelta"	durata della simulazione (secondi)
<i>dt</i>	"a scelta"	passo temporale (secondi)
<i>n</i>	"a scelta"	numero dei target che interagiranno nella simulazione
<i>tracce</i>	<i>zeros(1, n)</i>	parametro che uso per capire quali tracce entrano o spariscono

Tabella 3.3: Restanti parametri utilizzati dal simulatore.

### 3.4.1 Tracking Multiplo

Abbiamo simulato in ambiente Matlab il caso di tre target che, a partire da un istante iniziale in cui nessuno di essi è presente all'interno dell'area da sorvegliare, entrano nella scena uno alla volta. Successivamente escono dalla scena uno alla volta per tornare alla configurazione iniziale in cui nessuna traccia è presente.

Nel simulatore è stato realizzato un blocco di detection che fornisce al blocco di tracking le misure solo quando i target entrano all'interno dell'area (detection ideale).

Analogamente allo svolgimento dell'esempio del Paragrafo 2.6.1, consideriamo degli oggetti muoversi di moto rettilineo uniforme con velocità impostabile attraverso il parametro *vel*, con accelerazione nulla. Tutti i target sono soggetti ad un rumore di processo dovuto al fatto che la velocità istante per istante risulta essere soggetta a componenti aleatorie lungo *x* e lungo *y*  $(a_{k_x}, a_{k_y}) \in \mathbb{R}^2$ , le quali sono modellate come variabili aleatorie

Gaussiane indipendenti ed identicamente distribuite (i.i.d) a valor medio nullo e con deviazione standard pari a  $\sigma_a$  in m/s.

Allo stesso modo le misure saranno soggette ad un rumore  $(v_{k_x}, v_{k_y}) \in \mathbb{R}^2$ , le cui componenti sono anch'esse modellate come Gaussiane i.i.d a valor medio nullo e con deviazione standard pari a  $\sigma_z$  in metri.

Dobbiamo definire le matrici che identificano il processo e la misura. Riassumiamole in Tabella 3.4.

Parametro	Valore	Descrizione
$\sigma_a$	"a scelta"	dev standard rumore di processo (m/s)
$\sigma_z$	"a scelta"	dev standard rumore di misura (metri)
$Z$	$[z_1, z_2, z_3]$	vettore delle misure con $n=3$
$F$	$\begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	matrice di transizione degli stati $F$
$H$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	matrice di sensibilità della misura $H$
$Q$	$\sigma_a^2 \begin{bmatrix} dt^2 & 0 & dt & 0 \\ 0 & dt^2 & 0 & dt \\ dt & 0 & 1 & 0 \\ 0 & dt & 0 & 1 \end{bmatrix}$	covarianza del rumore di processo $Q$
$R$	$\sigma_z^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	covarianza del rumore di misura $R$

Tabella 3.4: Principali parametri utilizzati nel simulatore per il filtraggio multitarget.

Il filtraggio ci consente di ottenere una stima a posteriori sia del vettore di stato, sia della matrice di covarianza ad esso associata. Inoltre vengono calcolate alcune grandezze necessarie per la determinazione di ciascun vettore di stato a posteriori. Riassumiamo queste ultime in Tabella 3.5 a fronte.

Parametro	Descrizione
$\tilde{y}$	<i>Residuo di misura (o innovazione)</i>
$S$	<i>covarianza del residuo di misura</i>
$K$	<i>guadagno di Kalman ottimo</i>
$\hat{x}$	<i>stima dello stato del target</i>
$P_x$	<i>covarianza dell'errore di stima</i>

Tabella 3.5: Principali parametri utilizzati dal simulatore attraverso la correzione.

### 3.5 Simulazioni al Calcolatore del Sistema di Tracking Multitarget Basato su Algoritmo SHC

Prima di procedere ad illustrare i risultati é necessario premettere le convenzioni utilizzate.

Colore traccia	Descrizione
rosso	prima traccia che entra nel perimetro da sorvegliare
blu	seconda traccia che entra nel perimetro da sorvegliare
verde	terza traccia che entra nel perimetro da sorvegliare

Tabella 3.6: Precisazioni sui colori usati nelle simulazioni.

In Tabella 3.6 vengono riassunte le associazioni fra i colori e le tracce a seconda dell'ordine nel quale entrino nell'area da sorvegliare. Utilizzando tali convenzioni sui colori sono state eseguite diverse simulazioni, specificando di volta in volta quali parametri vengono variati. Le tracce hanno coordinate iniziali come specificato in Tabella 3.7.

Traccia	Colore	Vettore iniziale
1	rosso	[100;-15;0;0]
2	blu	[-30;120;0;0]
3	verde	[550;200;0;0]

Tabella 3.7: Desrizione dei valori iniziali dei vettori di stato dei singoli target.

### 3.5.1 Simulazioni Effettuate al Variare di $\sigma_z$

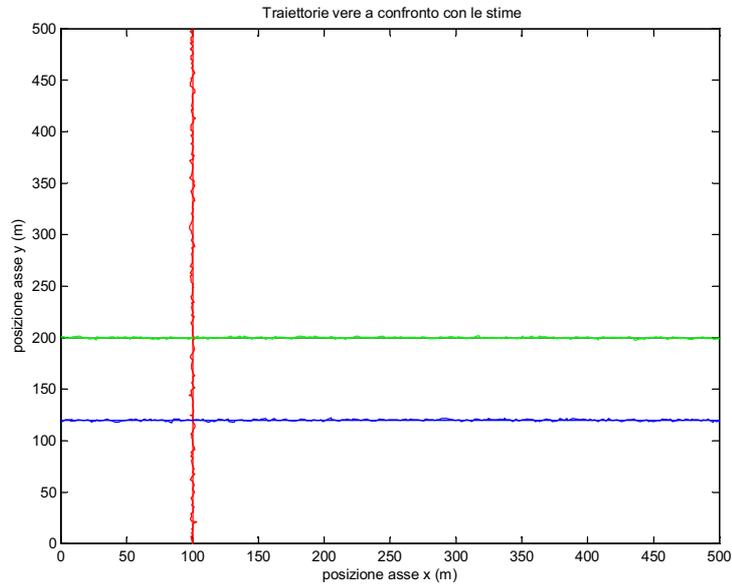


Figura 3.1: Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di  $\sigma_a = 0.2$  m/s e  $\sigma_z = 1$  m.

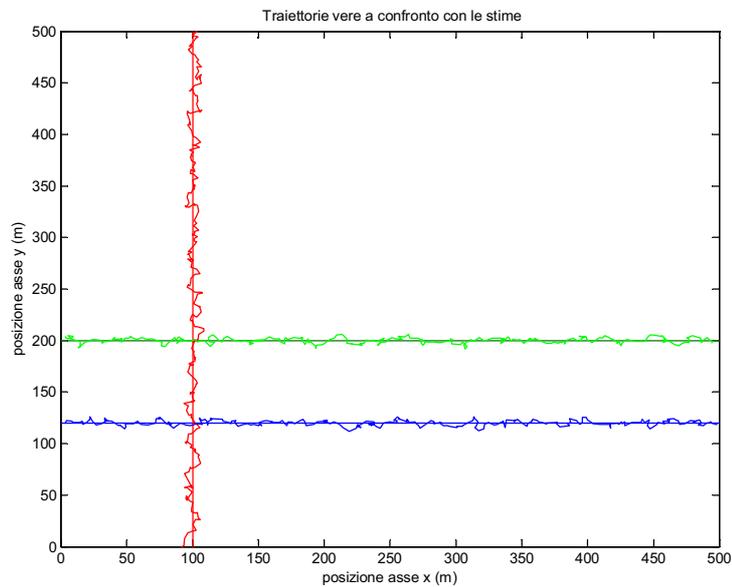


Figura 3.2: Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di  $\sigma_a = 0.2$  m/s e  $\sigma_z = 5$  m.

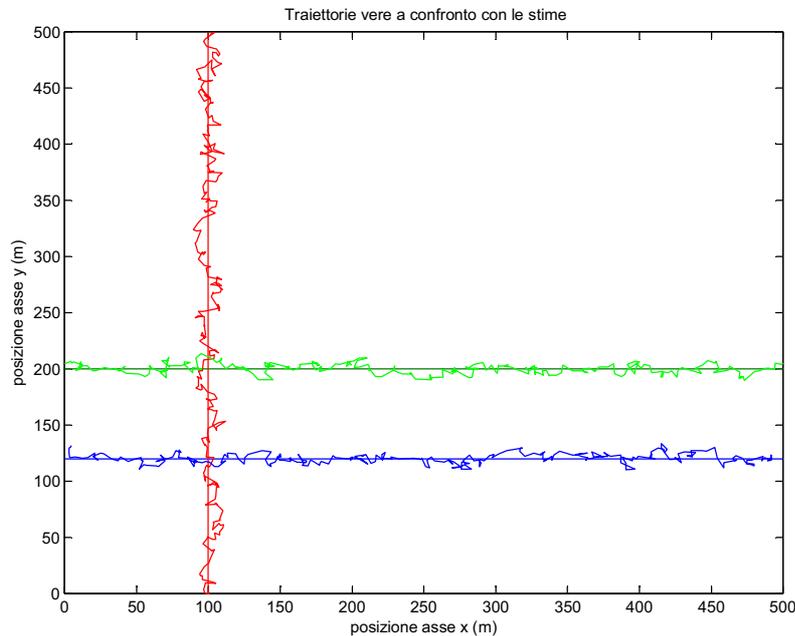


Figura 3.3: Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di  $\sigma_a = 0.2$  m/s e  $\sigma_z = 10$  m.

La Figura 3.1 mostra come il filtro segua in modo quasi perfetto l'andamento della traiettoria reale di ciascun target. Questo é dovuto ad una deviazione standard del rumore di misura relativamente bassa, e ad una velocità dei target anch'essa bassa, pari a 2 m/s, rispetto al passo  $dt$ . In Figura 3.2 si può notare come un aumento di  $\sigma_z$  si traduce in una traiettoria filtrata più discontinua. Nel caso  $\sigma_z = 10$ m, per effetto di una maggiore dispersione delle misure la traiettoria dovuta al filtraggio risulta essere 'peggiore' rispetto alle precedenti.

In Tabella 3.8 nella pagina successiva sono riassunti i parametri utilizzati nelle simulazioni illustrate nelle Figure 3.1, 3.2, 3.3.

Simulazione	$\sigma_a$ (m/s)	$\sigma_z$ (m)	vel (m/s)	durata (s)	dt (s)
Figura 3.1	0.2	1	2	700	1
Figura 3.2	0.2	5	2	700	1
Figura 3.3	0.2	10	2	700	1

Tabella 3.8: Principali parametri utilizzati per simulare i diversi comportamenti del filtro cambiando la deviazione standard del rumore di misura.

### 3.5.2 Simulazioni Effettuate al Variare di $\sigma_a$

Vengono ora presentati i risultati di simulazioni nelle quali, fissata la deviazione standard del rumore di misura  $\sigma_z$  (in particolare ponendo il valore di  $\sigma_z = 5$  m) si aumenta la deviazione standard sul rumore di processo  $\sigma_a$ . Con l'aumentare di  $\sigma_a$  ci si attende un output che 'peggiora' progressivamente.

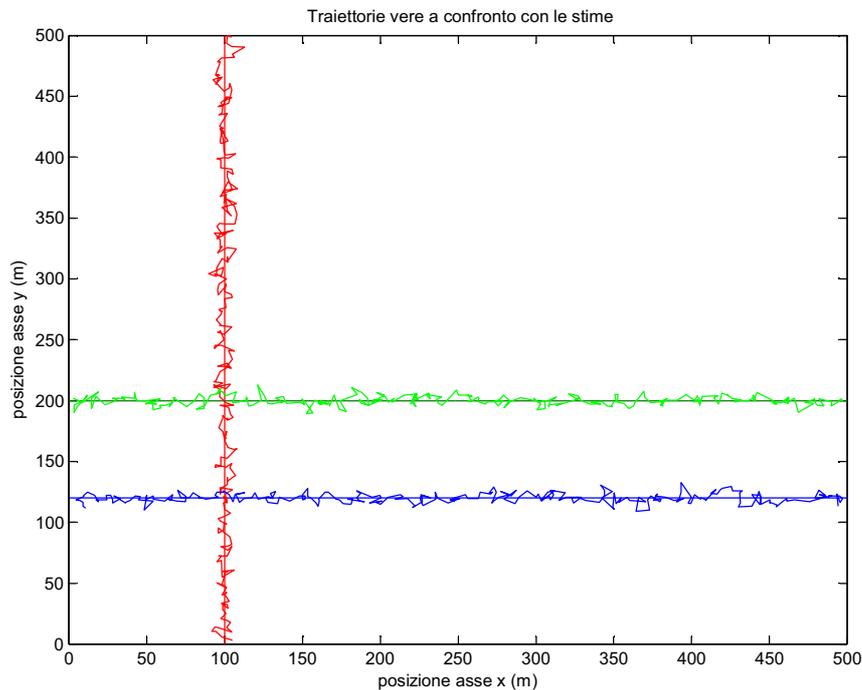


Figura 3.4: Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di  $\sigma_a = 0.5$  m/s e  $\sigma_z = 5$  m.

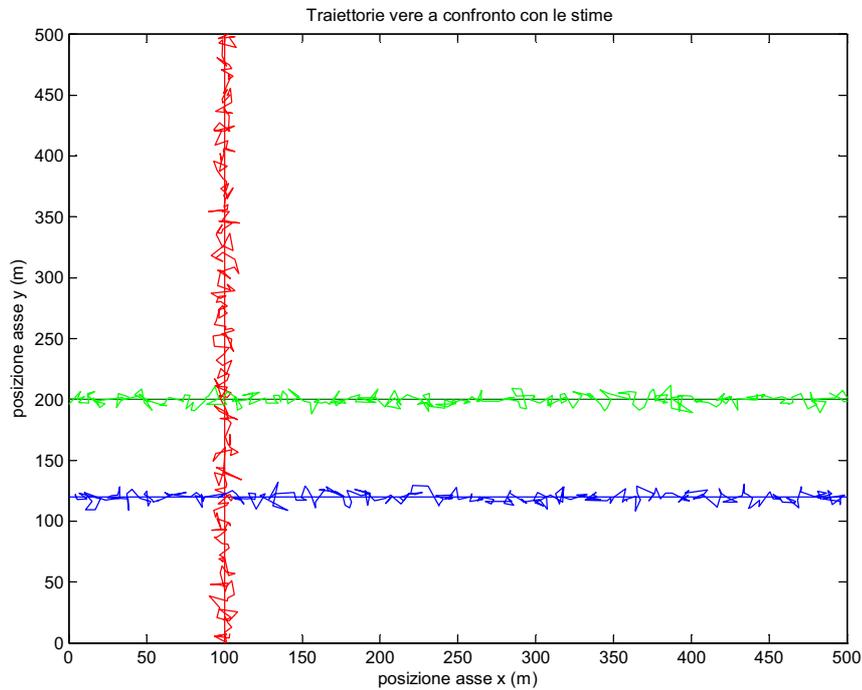


Figura 3.5: Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di  $\sigma_a = 1$  m/s e  $\sigma_z = 5$  m.

I grafici riportati nelle Figure 3.2, 3.4 e 3.5 confermano le aspettative. Riassumendo:

Simulazione	$\sigma_a$ (m/s)	$\sigma_z$ (m)	vel (m/s)	durata (s)	dt (s)
Figura 3.2	0.2	5	2	700	1
Figura 3.4	0.5	5	2	700	1
Figura 3.5	1	5	2	700	1

Tabella 3.9: Principali parametri utilizzati per simulare i diversi comportamenti del filtro cambiando la deviazione standard del rumore di processo.

### 3.5.3 Simulazioni Effettuate al Variare del Passo Temporale $dt$

Si é anche analizzato l'impatto del passo temporale  $dt$  nel comportamento del sistema di tracking. Mantenendo la stessa velocità per i target, una diminuzione del passo  $dt$  comporta una maggiore acquisizione di campioni, ovvero di misure.

I risultati delle simulazioni sono illustrate nelle Figure 3.6 e 3.7 (oltre che in Figura 3.2). Nelle Figure 3.8, 3.9 e 3.10 sono invece mostrati dettagli delle figure precedenti, relativamente alla traccia di colore rosso.

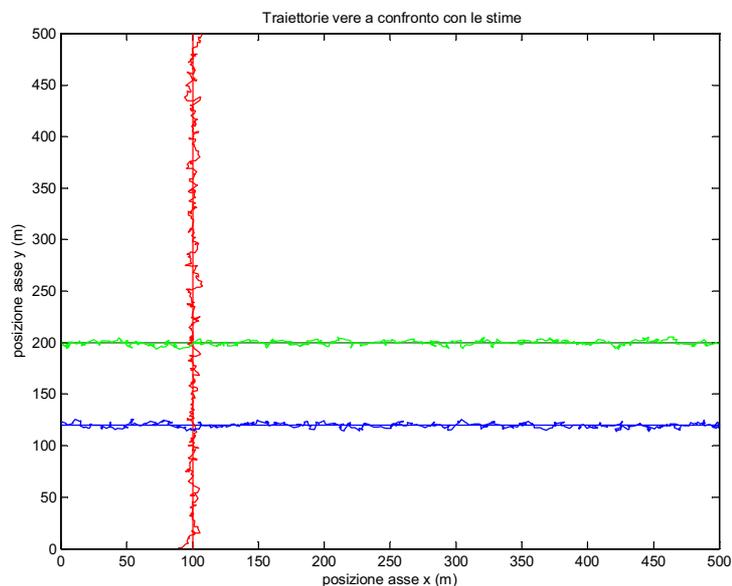


Figura 3.6: Tracking multiplo costituito da traiettorie rettilinee con l'assunzione di durata = 400 s e passo  $dt = 0.5$  s.

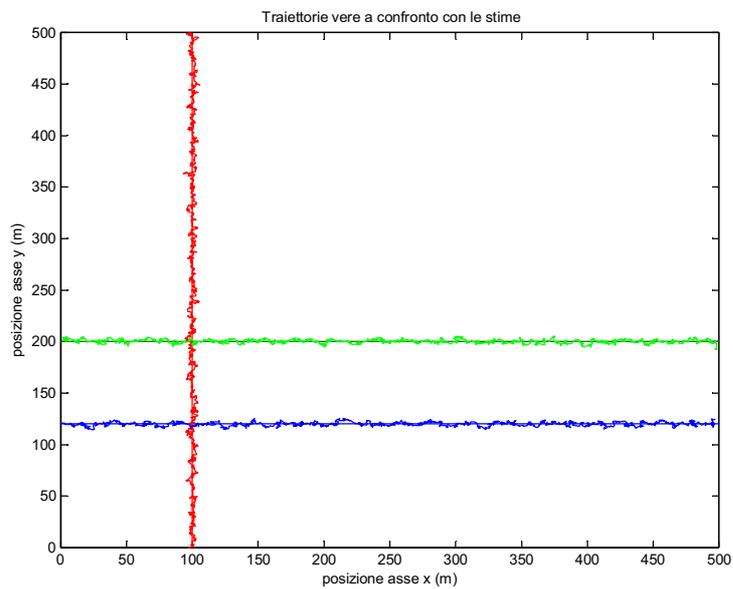


Figura 3.7: Tracking multiplo costituito da traiettorie rettiline con l'assunzione di durata = 300 s e passo  $dt = 0.2$  s.

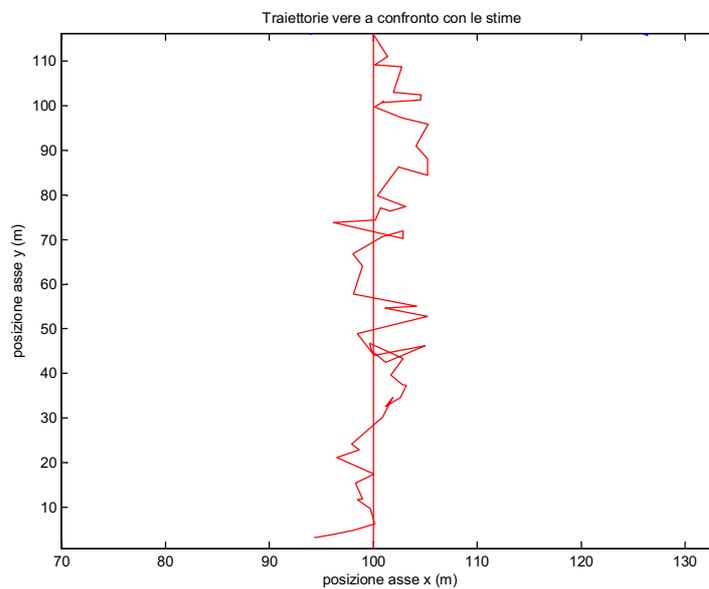


Figura 3.8: Dettaglio di Figura 3.2.

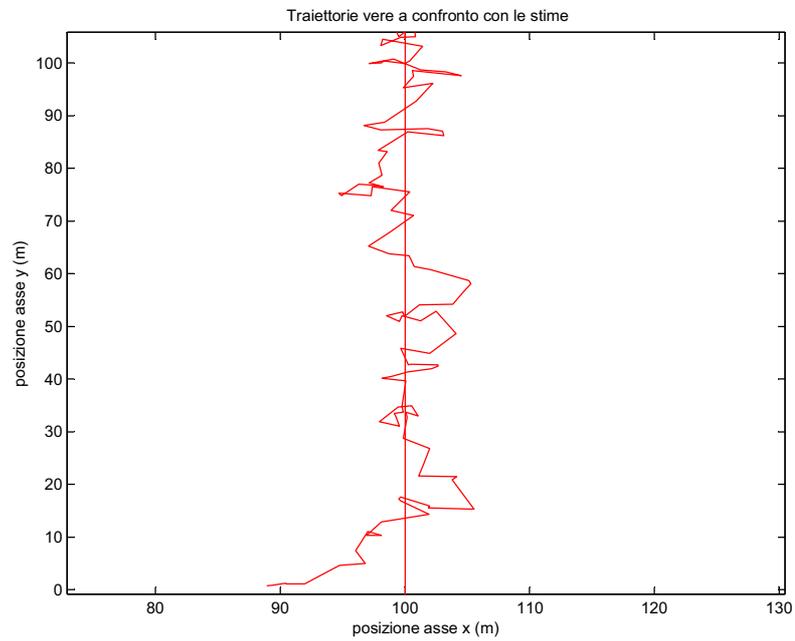


Figura 3.9: Dettaglio di Figura 3.6.

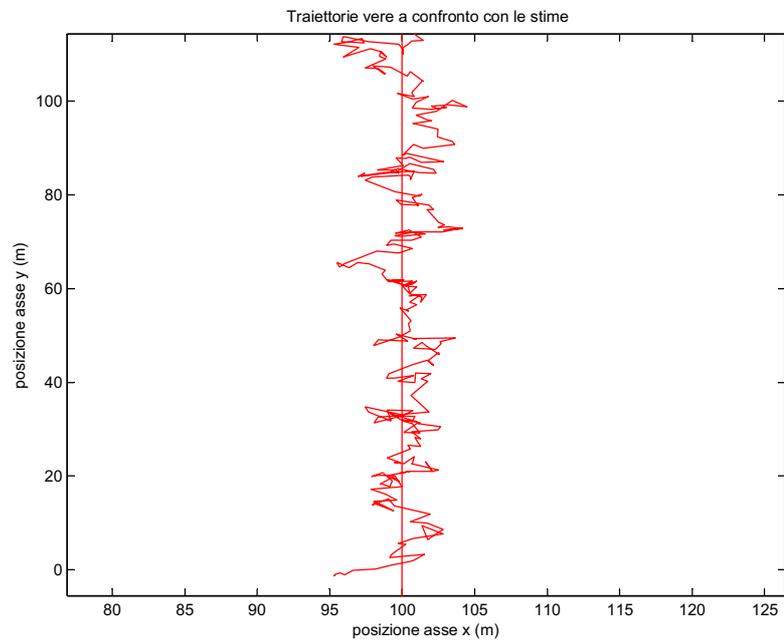


Figura 3.10: Dettaglio di Figura 3.7.

É possibile osservare come una diminuzione del passo temporale di scansione produca un aumento delle misure acquisite a parit  di intervallo temporale di osservazione. Mano mano che il passo temporale cala si percepiscono aumentare le ripide variazioni del filtraggio dovute appunto da un numero doppio di misure, come nel caso di Figura 3.9 (poich  il passo   dimezzato), e di numero quintuplo come nel caso di Figura 3.10 (poich  il passo   1/5).

Soprattutto in Figura 3.10 si nota come la traccia generata dal filtro, grazie ad un numero elevato di misure e date le caratteristiche delle variabili in gioco, abbia un valor medio che si avvicina molto a come il target si muove lungo la traiettoria reale. I cambiamenti repentini lungo la traiettoria filtrata potrebbero indurre ad inserire un filtro *passa alto*, in modo da 'smussare' la curva e farle inseguire meglio la traiettoria reale.

Riassumendo:

Simulazione	$\sigma_a$ (m/s)	$\sigma_z$ (m)	vel (m/s)	durata (s)	dt (s)
Figura 3.2	0.2	5	2	700	1
Figura 3.6	0.2	5	2	400	0.5
Figura 3.7	0.2	5	2	300	0.2

Tabella 3.10: Principali parametri utilizzati per simulare i diversi comportamenti del filtro cambiando il passo temporale.

### 3.6 Difetti legati all' univoca Correlazione

Come anticipato precedentemente, l'algoritmo *SHC* si comporta bene quando i target sono sufficientemente distanziati fra loro e quando il rapporto segnale-rumore sia sufficientemente buono. Siccome il filtro associa in modo univoco istante per istante la misura ad una traccia, in presenza di target vicini e misure rumorose a causa di un basso rapporto segnale-rumore, pu  capitare che questa associazione sia errata. In particolare l'algoritmo *SHC* potrebbe prendere decisioni errate sul caso di target che si incrociano durante il moto. Per analizzare questo aspetto abbiamo considerato due target presenti all'interno dell'area da sorvegliare (fin dall'istante iniziale),

---

dei quali uno si muove con traiettoria rettilinea e l'altro di moto circolare. Non consideriamo piú l'associazione fra le tracce ed i colori poiché entrambe le tracce sono presenti all'istante iniziale. I parametri scelti per la simulazione sono riassunti in Tabella 3.11.

Moto	Posizione iniziale	Descrizione
rettilineo	[0; 250; 0; 0]	$x$ crescenti
circolare	[250; 250; 0; 0]	$Radius = 50$ m, $w = 0.4$ rad/s, sfasamento nullo

Tabella 3.11: Principali parametri utilizzati per inizializzare i vettori necessari per simulare una delle errate identificazioni.

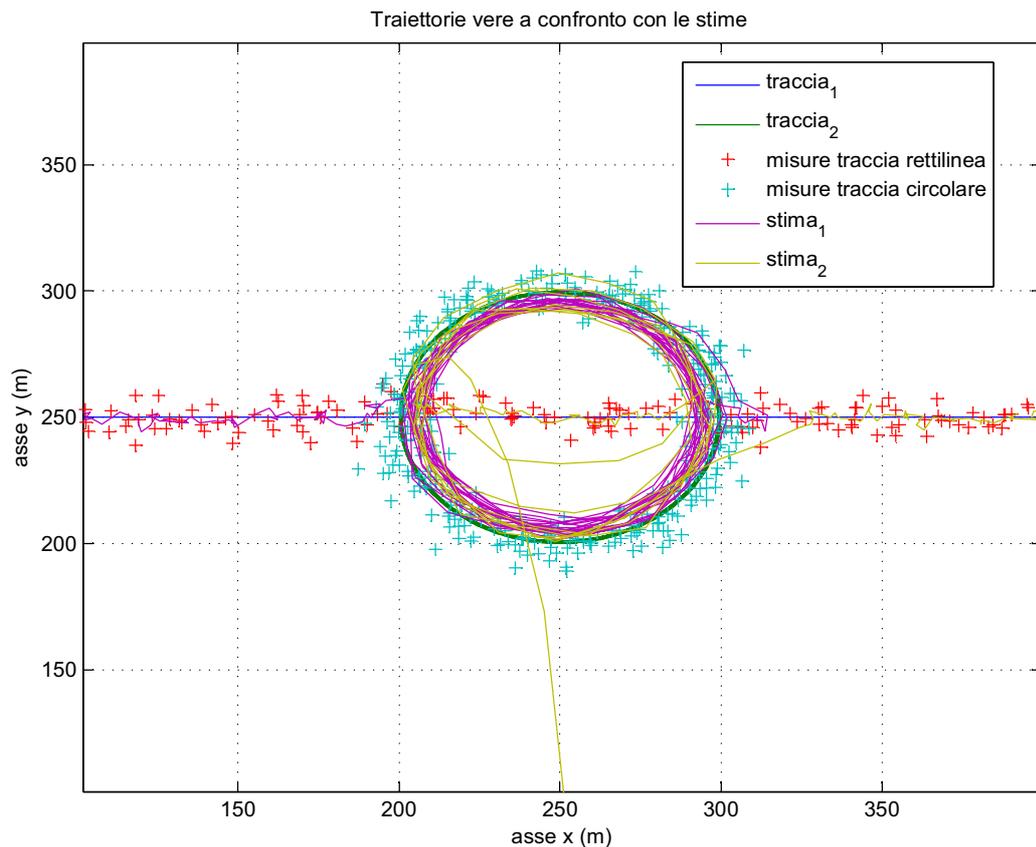


Figura 3.11: Primo caso di errata identificazione.

caso	$\sigma_a$ (m/s)	$\sigma_z$ (m)	$vel_{target-rett}$ (m/s)	durata (s)	dt (s)
Figura 3.11	0.2	5	2	400	1

Tabella 3.12: Principali parametri utilizzati per simulare i diversi comportamenti dovuti all'errata identificazione.

Come é possibile osservare dalla Figura 3.11, inizialmente entrambe le traiettorie filtrate restano associate alle misure generate dai rispettivi target. All'intersezione fra la traiettoria rettilinea e quella circolare avviene uno scambio di associazione. Esso é dovuto alla generazione di misure vicine fra loro (il target circolare ruota molto rapidamente), cosicché il filtro in quel determinato istante puó effettuare una errata associazione. Si osserva come essa faccia sí che la traiettoria filtrata che era inizialmente associata al target in moto rettilineo (traccia di colore viola) inizi ad inseguire il target che si muove di moto circolare. Viceversa la traccia filtrata che inizialmente era associata al target in moto circolare (traccia di colore giallo) inizia a inseguire le misure (di colore rosso in Figura 3.11) generate dal target che si muove di moto rettilineo. Nel caso di pulsazione del target circolare piú bassa, per esempio pari a 0.2 rad/s, si puó invece osservare come lo scambio non avvenga poiché il segnale filtrato relativo alle due tracce non si incontra nel punto di intersezione fra le curve, eseguendo un corretto Tracking. Questo si verifica sia nell'intersezione di entrata del percorso rettilineo su quello circolare, sia in uscita. Di fatto il punto debole del sistema é rappresentato dal caso in cui i segnali filtrati siano vicini fra loro e che la dispersione delle misure sia tale da renderle indistinguibili.

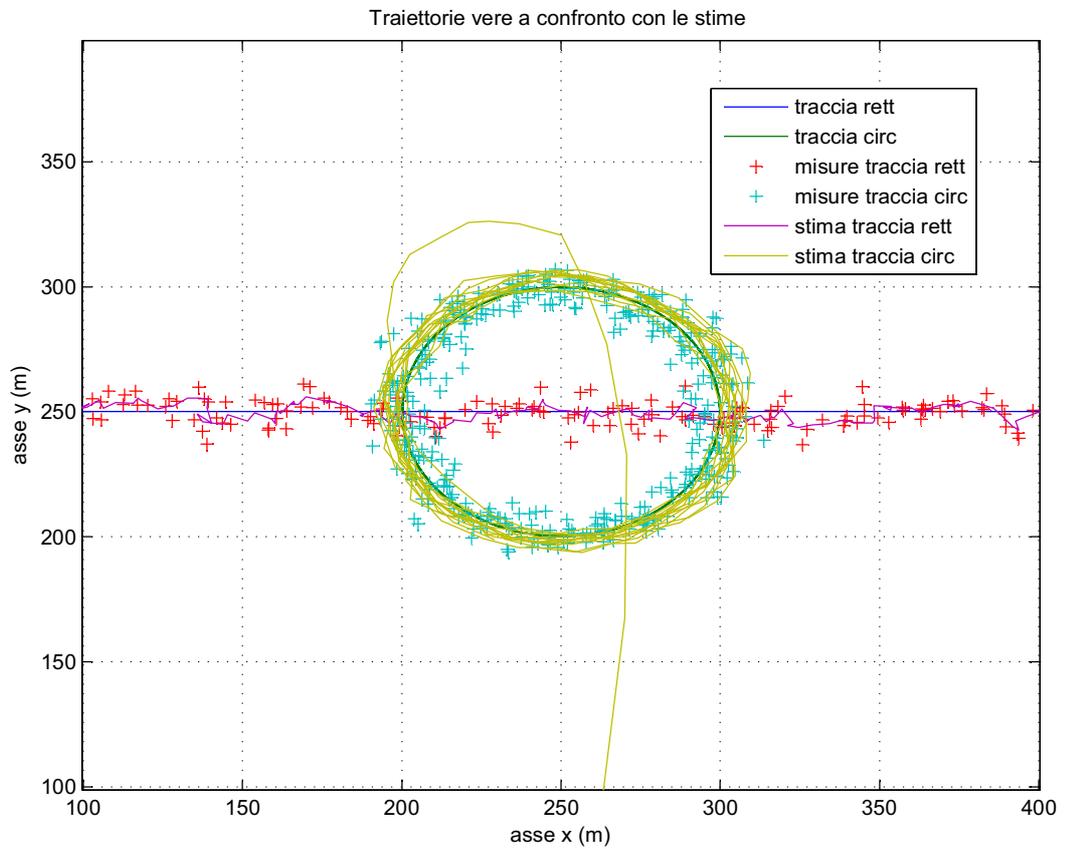


Figura 3.12: Caso di giusta identificazione.

Moto	Posizione iniziale	Descrizione
rettilineo	[0; 250; 0; 0]	$x$ crescenti
circolare	[250; 250; 0; 0]	$Radius = 50$ m, $w = 0.2$ rad/s, sfasamento nullo

Tabella 3.13: Principali parametri utilizzati per inizializzare i vettori necessari per simulare una delle errate identificazioni.

# Capitolo 4

## Prestazioni del Sistema Implementato

L'obiettivo di questo capitolo é quello di valutare le prestazioni dell'algoritmo *SHC* implementato, in termini di probabilità di falsa associazione fra misure e tracce. A tale scopo sono state effettuate diverse simulazioni di incrocio delle traiettorie di due target, uno dei quali si muove con  $x$  crescenti mentre l'altro con  $x$  decrescenti. Le rispettive quote  $y$  sono state mantenute costanti per entrambi gli oggetti. La probabilità di errore del sistema é stata valutata modificando nelle diverse simulazioni la quota  $\delta$  dalla quale si scostano in modo positivo o negativo le traiettorie reali rispetto ad una quota di riferimento dei target (esempio il target con  $x$  crescente ha quota  $(250 + \delta)$  mentre il target con  $x$  decrescente ha quota  $(250 - \delta)$ ), fissati i valori dei parametri  $\sigma_a$  e  $\sigma_z$ . Consideriamo due target, uno in posizione iniziale  $(0, 250 + \delta)$  il quale si muove verso  $x$  crescenti di moto rettilineo ed uniforme e un secondo target di posizione iniziale  $(501, 250 - \delta)$  il quale si muove verso  $x$  decrescenti di moto rettilineo ed uniforme e con la stessa velocità. In base al valore  $\delta$ , fissati i valori delle deviazioni standard sui rumori di misura e di processo, ci si attende che il sistema si comporti in modo diverso. Più precisamente all'aumentare di  $\delta$ , cioè ad un aumento della quota  $y$  di separazione fra i target, ci si attende che il sistema offra prestazioni migliori. In Tabella 4.1 nella pagina seguente mostriamo quanto detto precedentemente.

Fissato il valore dei parametri  $\delta$ ,  $\sigma_a$  e  $\sigma_z$  abbiamo eseguito  $n$  volte la si-

---

N° target	Cordinate	Descrizione
1	$[0, 250 + \delta]$	traccia che entra per prima con $x$ crescente
2	$[501, 250 - \delta]$	traccia che entra per seconda con $x$ decrescente

Tabella 4.1: Principali parametri utilizzati dal simulatore per eseguire una generica simulazione di incrocio.

mulazione di incrocio dei target, dove questo parametro puó essere scelto dall'utente; si ricorda che all'aumentare di  $n$  corrisponde un tempo di simulazione maggiore e contestualmente una maggiore affidabilità del risultato in termini statistici. Ogni volta che il sistema verifica lo scambio di associazione delle tracce nella generica simulazione di incrocio, viene incrementata la variabile *ERROR*, la quale alla fine degli  $n$  cicli impostati conterrà il numero totale degli errori. Alla fine del ciclo, tale variabile viene divisa per il numero totale dei cicli stessi per fornirci la probabilità di errore del sistema, dati i parametri fissati. Per le simulazioni si é scelto di usare  $\sigma_a = 0.2$  m/s e  $\sigma_z = 7$  m. I casi riscontrati nelle simulazioni sono stati due, rappresentati nelle Figure 4.1 e 4.2.

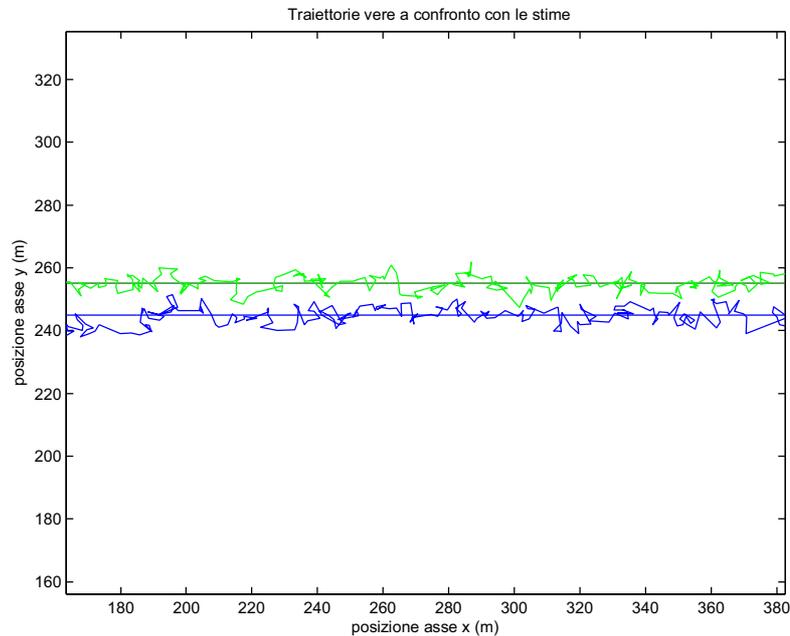


Figura 4.1: Esempio di simulazione di incrocio dei target correttamente riuscito.

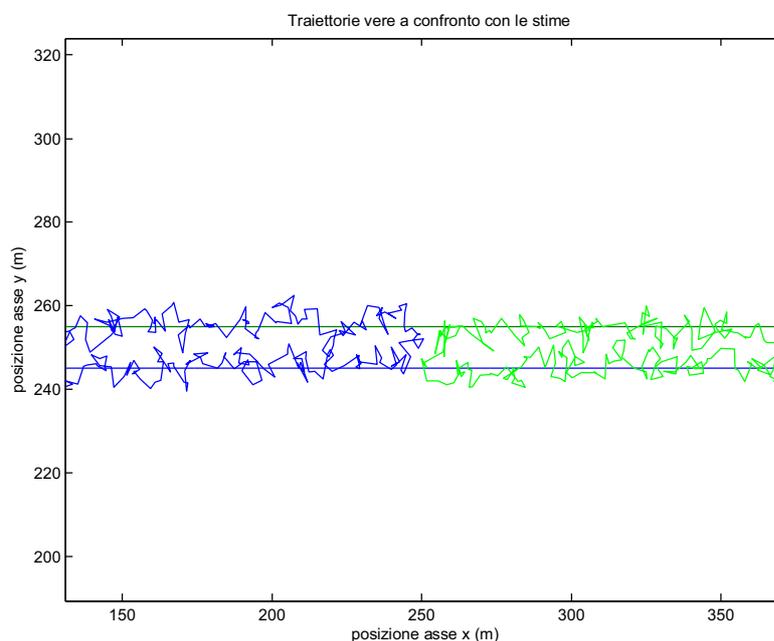


Figura 4.2: Esempio di simulazione di incrocio dei target in cui avviene lo scambio di associazione fra tracce e misure.

Si nota come in Figura 4.1 nella pagina precedente non avviene lo scambio di associazione e ciascuna delle tracce filtrate rimane associata al medesimo target durante l'intera simulazione. In Figura 4.2 si nota invece come ci sia uno scambio di associazione il quale fa sì che il segnale filtrato si agganci al target sbagliato (lo si vede bene poiché c'è il cambiamento di quota). Abbiamo eseguito delle simulazioni con due diversi valori di  $n$ . In Tabella 4.2 sono riportati gli ulteriori parametri necessari per la simulazione e la probabilità di errore del sistema per un numero di cicli pari a 100, così come in Tabella 4.3 in cui si è scelto un numero di cicli pari a 1000.

Probabilità di errore	$\delta$ (m)	cicli tot	vel (m/s)	durata (s)	dt (s)
0.3	2	100	1	550	1
0.18	3	100	1	550	1
0.15	5	100	1	550	1
0	10	100	1	550	1

Tabella 4.2: Principali parametri utilizzati dal simulatore per eseguire un ciclo for di 100 simulazioni di incrocio.

Probabilità di errore	$\delta$ (m)	cicli tot	vel (m/s)	durata (s)	dt (s)
0.234	2	1000	1	550	1
0.225	3	1000	1	550	1
0.145	5	1000	1	550	1
0.01	10	1000	1	550	1

Tabella 4.3: Principali parametri utilizzati dal simulatore per eseguire un ciclo for di 1000 simulazioni di incrocio.

Si può notare come un aumento di  $n$  comporti una precisione maggiore sulla probabilità di errore del sistema; si ricorda che questo avviene a scapito della durata temporale delle simulazione. La Tabella 4.2 e la Tabella 4.3 confermano quello che ci aspettavamo; ovvero che più i target si distanziano, cioè più  $\delta$  aumenta, e più il sistema riesce ad essere efficiente attribuendo alle tracce filtrate le giuste misure. Si può concludere che per ottenere una probabilità d'errore inferiore a  $10^{-1}$  occorre che i target siano distanziati dalla quota di riferimento di almeno  $\sigma_z$  (infatti fissata  $\sigma_z = 7$  m si osserva che si ottiene la probabilità d'errore richiesta quando  $\delta$  passa da 5 m a 10 m).

## 4.1 Conclusioni

Lo scopo di questo elaborato è stato quello di implementare e analizzare un tracking di oggetti multipli all'interno di un'area da sorvegliare. Siamo partiti dall'analisi e realizzazione di un filtro di Kalman a singolo target effettuando diverse simulazioni. Esso richiede determinate ipotesi di lavoro, quali che le distribuzioni in gioco siano Gaussiane a valore medio nullo, e che il modello del sistema sia lineare. Se queste ipotesi vengono a meno, le prestazioni del filtro degradano. Abbiamo visto come il filtro tenda a seguire bene la traiettoria vera in condizioni di velocità del target non elevate e in condizioni di rumore non eccessivo.

Abbiamo realizzato a partire da questa implementazione un filtro che gestisca un tracking multiplo, basato sull'ipotesi di univoca correlazione *SHC*, soffermandoci in modo particolare sull'algoritmo che gestisce le associazioni tracce-misure. Abbiamo effettuato diverse simulazioni e discusso il comportamento di questo filtro al variare dei parametri conclu-

dendo che il filtro si comporta bene quando i target sono sufficientemente distanziati fra loro e il rapporto  $SNR$  sia sufficientemente buono.

Le misure date in ingresso al sistema di tracking non sono state ottenute da un caso pratico di laboratorio ma simulate attraverso un blocco di detection, creato nell'algoritmo di filtraggio. Uno step successivo potrebbe consistere nello sfruttare questo algoritmo al fine di eseguire delle misure in laboratorio e valutare i risultati ottenuti. Dalle nostre simulazioni e dall'analisi sulle prestazioni esso é comunque risultato efficiente.

---



# Appendice A

## Richiami di Teoria della Probabilità

### A.1 Matrice di covarianza

Data una variabile aleatoria vettoriale  $X = (X_1, X_2, \dots, X_n)$  con valore medio dato dal vettore  $\mu = \mathbb{E}[X] = (\mu_1, \mu_2, \dots, \mu_n) \in \mathbb{R}^{n \times 1}$  dove  $\mu_i = \mathbb{E}[X_i]$ , si definisce funzione di covarianza  $C \in \mathbb{R}^{n \times n}$  di  $X$ , la matrice [10, 5]:

$$C = \text{cov}(X) = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T]$$

con elementi:

$$\begin{aligned} C_{i,j} &= \text{cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] \\ &= \mathbb{E}[X_i, X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j]. \end{aligned}$$

La matrice di covarianza ha in posizione  $i, i$  sulla diagonale principale la varianza della componente  $i$ -esima  $\sigma_i^2$  del vettore aleatorio, e in posizione  $i, j$  la covarianza  $\text{cov}(X_i, X_j)$  tra la componente  $i$ -esima e la  $j$ -esima, ovvero:

---

$$C = \begin{bmatrix} \sigma_1^2 & C_{1,2} & \dots & C_{1,n} \\ C_{2,1} & \sigma_2^2 & \dots & C_{2,n} \\ \vdots & \vdots & \ddots & \dots \\ C_{n,1} & C_{n,2} & \dots & \sigma_n^2 \end{bmatrix}$$

Dato che  $cov(X_i, X_j) = cov(X_j, X_i)$ , la matrice di covarianza é simmetrica.

Nel caso la variabile aleatoria vettoriale  $X$  sia formata da variabili aleatorie incorrelate, cioè tali che  $\mathbb{E}[X_i X_j] = \mathbb{E}[X_i] \mathbb{E}[X_j]$ , allora la matrice di covarianza  $C \in \mathbb{R}^{n \times n}$  di  $X$ , avrà elementi:

$$C_{i,j} = cov(X_i, X_j) = \mathbb{E}[X_i X_j] - \mathbb{E}[X_i] \mathbb{E}[X_j] = \begin{cases} \sigma_i^2, & \text{se } i = j \\ 0, & \forall i \neq j \end{cases}$$

dove  $\sigma_i^2 = \mathbb{E}[X_i^2] - \mathbb{E}^2[X_i]$  é la varianza della variabile aleatoria  $X_i$ .

Nel caso la variabile aleatoria vettoriale  $X$  sia formata da variabili aleatorie incorrelate e a valore medio nullo, allora la matrice di covarianza  $C \in \mathbb{R}^{n \times n}$  di  $X$  diventa:

$$C = cov(X) = \mathbb{E}[X X^T]$$

con elementi:

$$C_{i,j} = cov(X_i, X_j) = \mathbb{E}[X_i X_j] = \begin{cases} \sigma_i^2, & \text{se } i = j \\ 0, & \forall i \neq j \end{cases}$$

dove  $\sigma_i^2 = \mathbb{E}[X_i^2]$  é la varianza della variabile aleatoria  $X_i$ . Si noti che la traccia della matrice di covarianza  $C$ , é data da:

$$tr(C) = \mathbb{E}[X^T X] = \sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2$$


---

## A.2 Variabili aleatorie congiunte Gaussiane

Data una variabile aleatoria vettoriale  $X = (X_1, X_2, \dots, X_n)$  le variabili  $X_1, X_2, \dots, X_n$  sono dette congiuntamente Gaussiane se la funzione densità di probabilità (p.d.f) congiunta di  $X$  si scrive come [11, 5]:

$$p(x_1, x_2, \dots, x_n) = p(x) = \frac{1}{\sqrt{(2\pi)^n \det(C)}} \left\{ -\frac{1}{2} [(x - \mu)C^{-1}(x - \mu)^T] \right\}$$

dove:

- $C = \{C_{i,j}\} \in \mathbb{R}^{n \times n}$  é la matrice di covarianza di  $X$  costituita dagli elementi  $C_{i,j} = \text{cov}(X_i, X_j)$ ;
- $\mu = \mathbb{E}[X] = (\mu_1, \dots, \mu_n) \in \mathbb{R}^{n \times 1}$  é il valore medio di  $X$  ed é costituito dagli elementi  $\mu_i = \mathbb{E}[X_i]$ .

É possibile utilizzare la notazione compatta:

$$X \sim \mathcal{N}(\mu, C).$$

Le variabili aleatorie congiuntamente Gaussiane godono delle seguenti proprietà:

- la distribuzione statistica é determinata dalla sola conoscenza di  $\mu$  e  $C$ ;
- se le variabili aleatorie sono tra loro incorrelate, allora sono anche statisticamente indipendenti e  $C$  é una matrice diagonale;
- se  $X_1, X_2, \dots, X_n$  sono variabili aleatorie congiuntamente Gaussiane, allora ogni variabile  $X_i$  con  $1 \leq i \leq n$ , é una variabile aleatoria Gaussiana;
- una qualunque combinazione lineare di variabili aleatorie congiuntamente Gaussiane é ancora Gaussiana.

Ad esempio, se  $X_1, X_2, \dots, X_n$  sono variabili aleatorie congiuntamente Gaussiane, la variabile aleatoria  $Y = AX + B$ , con  $A, B$  matrici di dimensioni opportune, é un vettore di variabili aleatorie congiuntamente

---

Gaussiane, che si può scrivere come:

$$Y \sim \mathcal{N}(\mu_Y, C_Y)$$

con:

$$\mu_Y = A\mu_X + B \quad , \quad C_Y = AC_XA^T$$

dove  $C_X$  e  $C_Y$  sono le matrici di covarianza di  $X$  e  $Y$ , e dove  $\mu_X = \mathbb{E}[X]$  e  $\mu_Y = \mathbb{E}[Y]$  sono i valori medi di  $X$  e  $Y$ .

### A.3 Processi stocastici tempo-discreti

Una variabile aleatoria discreta é rappresentata da una funzione  $X$  definita per ogni realizzazione  $s$  di un esperimento aleatorio. Se assegnamo ad ogni realizzazione  $s$  una funzione tempo-discreta  $X_k$ , otteniamo una successione di funzioni chiamata processo stocastico tempo-discreto [12, 5, 13]:

$$\{ X_k \}, \quad k = 1, 2, \dots$$

Ogni valore del processo stocastico  $X_k$  ad un particolare istante  $k = k_0$ , costituisce una variabile aleatoria. Nel caso  $X_k$  sia un vettore  $\in \mathbb{R}^{n \times 1}$  allora, ogni valore del processo stocastico ad un'istante  $k = k_0$ , costituisce una variabile aleatoria vettoriale. Ad ogni istante  $k$ , possiamo definire il valore medio della variabile aleatoria  $X_k$  dato da  $\mathbb{E}[X_k]$ , che nel caso vettoriale costituisce un vettore di valori medi  $\mathbb{E}[X_k] \in \mathbb{R}^{n \times 1}$ .

Dato un processo stocastico  $X_k \in \mathbb{R}^{n \times 1}$ , si definisce funzione di autocorrelazione  $R_X(i, j) \in \mathbb{R}^{n \times n}$  la matrice:

$$\mathcal{R}_X(i, j) = \text{corr}(X_k, X_j) = \mathbb{E}[X_k X_j^T]$$

e funzione di autocovarianza  $C_X(i, j) \in \mathbb{R}^{n \times n}$  la matrice:

---

$$\begin{aligned} C_X(i, j) &= \text{cov}(X_k, X_j) = \mathbb{E}[(X_k - \mathbb{E}[X_k])(X_j - \mathbb{E}[X_j])^T] \\ &= \mathcal{R}_X(i, j) - \mathbb{E}[X_i]\mathbb{E}[X_j^T] \end{aligned}$$

In particolare, se il processo é a valor medio nullo, autocorrelazione ed autocovarianza coincidono, ovvero:

$$C_X(i, j) = \mathcal{R}_X(i, j) = \mathbb{E}[X_k X_j^T]$$

Un processo  $X_k$  é definito incorrelato, se:

$$C_X(i, j) = \mathbb{E}[(X_k - \mathbb{E}[X_k])(X_j - \mathbb{E}[X_j])^T] = C_k \delta_{k-j}$$

dove  $C_k \in \mathbb{R}^{n \times n}$  rappresenta la matrice di covarianza di  $X_k$ , e  $\delta_k$  rappresenta la funzione delta di Kronecker definita come:

$$\delta_k = \begin{cases} 1 & \text{se } k = 0 \\ 0 & \text{altrimenti.} \end{cases}$$

Un processo stocastico bianco, é un esempio di processo incorrelato.

Assegnati due processi stocastici  $X_k \in \mathbb{R}^{n \times 1}$  e  $Y_k \in \mathbb{R}^{m \times 1}$ , si definisce funzione di crosscorrelazione  $R_{XY}(i, j) \in \mathbb{R}^{n \times m}$  la matrice:

$$R_{XY}(i, j) = \text{corr}(X_k, Y_j) = \mathbb{E}[X_k Y_j^T]$$

e funzione di crosscovarianza  $C_{XY}(i, j) \in \mathbb{R}^{n \times m}$  la matrice:

$$\begin{aligned} C_{XY}(i, j) &= \text{cov}(X_k, Y_j) = \mathbb{E}[(X_k - \mathbb{E}[X_k])(Y_j - \mathbb{E}[Y_j])^T] \\ &= R_{XY}(i, j) - \mathbb{E}[X_i]\mathbb{E}[Y_j^T]. \end{aligned}$$

In particolare, se i processi sono a valor medio nullo, crosscorrelazione e crosscovarianza coincidono, ovvero:

$$C_{XY}(i, j) = R_{XY}(i, j) = \mathbb{E}[X_k Y_j^T].$$


---

Due processi  $X_k$  e  $Y_k$  si dicono incorrelati se la relativa crosscovarianza é nulla, ovvero:

$$C_{XY}(i, j) = 0$$

# Bibliografia

- [1] E. Paolini, A. Giorgetti, M. Chiani, R. Minutolo, and M. Montanari, "*Localization Capability of Cooperative Anti-Intruder Radar Systems*", EURASIP Journal on Advances in Signal Processing, Vol. 2008, Article ID 726854, 14 pages, 2008.
  - [2] M. Chiani, A. Giorgetti, M. Mazzotti, R. Minutolo, and E. Paolini, "*Target Detection Metrics and Tracking for UWB Radar Sensor Networks*", Ultra-Wideband, 2009. ICUWB 2009. IEEE International Conference on, 9-11 September 2009 Page(s): 469 - 474.
  - [3] A. Urbini, "*Sistemi di localizzazione UWB: Algoritmi di rimozione del clutter per radar multistatico*", Tesi di laurea specialistica in Ingegneria Elettronica e delle Telecomunicazioni, Università di Bologna, II Facoltà di Ingegneria, A. A. 2006-2007.
  - [4] A. Facondini, "*Sistemi di localizzazione UWB: Algoritmi di rivelazione per radar multistatico*", Tesi di laurea specialistica in Ingegneria Elettronica e delle Telecomunicazioni, Università di Bologna, II Facoltà di Ingegneria, A. A. 2006-2007.
  - [5] L. Calandrino, M. Chiani, "*Lezioni di Comunicazioni Elettriche*", Pitagora Editrice Bologna, 2 ed. 2006.
  - [6] E. Macrelli, "*Sistemi Di Localizzazione UWB: Algoritmi Di Tracking Con Filtro di Kalman Per Radar Multistatico*", Tesi di laurea specialistica in Ingegneria Elettronica e delle Telecomunicazioni, Università di Bologna, II Facoltà di Ingegneria, A. A. 2008-2009.
  - [7] Ronald P.S. Mahler, "*Statistical Multisource-Multitarget, Information Fusion*", Artech House, 2007.
-

- [8] S.S. Blackman, *"Multiple hypothesis tracking for multiple target tracking"*, IEEE Aerospace and Electr.Sys.Mag., Part 2:Tutorials, Vol. 19 No. 1, pp. 1-18,2004
  - [9] S.S. Blackman and S.Popoli, *"Design and Analysis of Modern Tracking Systems, Norwood, MA: "*, Artech House, 2000
  - [10] A. Bicchi, *"Fondamenti di Automatica 3-parte"*, Corso di Fondamenti di Automatica, 2007, Università di Pisa.
  - [11] L. Schenato, *"Il filtro di Kalman"*, Progettazione di sistemi di controllo, 2006, Università degli Studi di Padova.
  - [12] Mohinder S. Grewal, Angus P. Andrews, *"Kalman Filtering: Theory and Practice Using MATLAB"*, 3rd Edition September 2008, John Wiley and Sons, Inc.
  - [13] M. Fossi, M. Gherardelli *"Appunti di Teoria dei Segnali: Segnali aleatori"*, Corso di teoria dei segnali, 2008, Università degli studi di Firenze.
-

# Ringraziamenti

*È finito per me questo primo periodo universitario, ricco di soddisfazioni e di prove che fanno maturare una persona. È doveroso da parte mia ringraziare alcune persone.*

*Un grazie va al mio relatore prof. ing. Marco Chiani che ha dato la disponibilità e l'approvazione nel farmi realizzare questo elaborato, è stato per me un onore.*

*Un grazie sentito va al mio correlatore ing. Enrico Paolini il quale mi ha concretamente accompagnato in questi mesi di lavoro, infondendo in me lo spirito nell'affrontare nuove sfide e nuovi problemi che emergevamo durante la stesura di questo elaborato, persone così arricchiscono la nostra università.*

*Un grazie a Francesca, la cui presenza mi infonde serenità e felicità; in questi anni ha cercato sempre di non farmi pesare nulla soprattutto nei momenti per me più difficili; ricordo attendere insieme il risultato degli esami, con ansia crescente fino alla fine; questa laurea è come se fosse anche sua, un grazie di cuore.*

*Ringrazio i miei cari amici, molti dei quali oramai li conosco da anni e con i quali condivido grossa parte del tempo libero, in particolar modo il gabbo e zampo con i quali ho maggiormente legato e aspettato con gioia questo momento.*

*Ringrazio tutti gli amici conosciuti all'università, con i quali abbiamo condiviso, per chi più o meno, tempo insieme; sicuramente ricorderò per le avventure passate e per i tanti viaggi insieme il vinz, les, marco, paolo, vero, simo, raffa, marino, feltro, stefania e tanti altri ancora con i quali in questi anni abbia condiviso una risata o momenti di gioia.*

*Infine ringrazio tutti coloro che hanno avuto fiducia in me, quali i miei genitori; la motivazione e la voglia di raggiungere determinati obiettivi mi rendono oggi una persona fiera e orgogliosa di me stesso.*

Iacopo

---