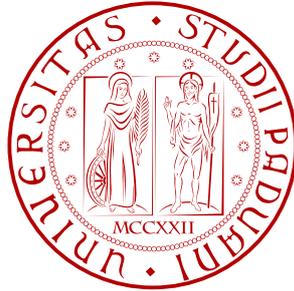


UNIVERSITÀ DEGLI STUDI DI PADOVA



Facoltà di Ingegneria  
Corso di Laurea Magistrale in Ingegneria dell'Automazione

**Corso di Progettazione di Sistemi di Controllo  
A.A. 2009/10**

**Coordinazione distribuita di telecamere  
per patrolling di perimetri e tracking**

Mauro Baseggio, Pierangelo Merlo, Mauro Pozzi  
603399-IAM 607368-IAM 603717-IAM

Padova, 15 febbraio 2010

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Stato dell'arte</b>	<b>5</b>
<b>3</b>	<b>Analisi teorica del problema di patrolling</b>	<b>7</b>
3.1	Definizione delle variabili del problema . . . . .	8
3.2	Criterio di ottimalità . . . . .	9
3.3	Dinamica di controllo . . . . .	10
3.3.1	Tipo di minimizzazione . . . . .	10
3.3.2	Modello probabilistico dell'evento . . . . .	11
<b>4</b>	<b>Calcolo della traiettoria ottima</b>	<b>13</b>
4.1	Traiettoria ottima senza vincoli di copertura . . . . .	13
4.2	Traiettoria ottima con vincoli di copertura . . . . .	18
4.2.1	Analisi di ammissibilità della soluzione ottima non vincolata . .	18
4.2.2	Calcolo della traiettoria ottima vincolata . . . . .	19
4.3	Problemi e limiti implementativi . . . . .	21
4.3.1	Analisi della soluzione ottima in termini distribuiti . . . . .	21
4.3.2	Grafo di comunicazione del sistema . . . . .	21
4.3.3	Comunicazione passo-passo: calcolo del periodo ottimo . . . . .	22
4.3.4	Analisi della complessità dell'algoritmo passo-passo . . . . .	22
<b>5</b>	<b>Algoritmo distribuito per determinare la copertura ottima</b>	<b>24</b>
5.1	Paragone con un sistema fisico meccanico . . . . .	24
5.2	Equivalenza della soluzione di equilibrio con il calcolo delle sezioni ottime	26
5.3	Analisi in spazio di stato . . . . .	27
5.4	Vantaggi della soluzione distribuita adottata . . . . .	28
5.4.1	Robustezza e flessibilità . . . . .	28
<b>6</b>	<b>Implementazione: struttura base e patrolling</b>	<b>29</b>
6.1	Comunicazione tra le camere . . . . .	29
6.2	Forza sugli estremi di patrolling . . . . .	31
6.3	Modello degli estremi di patrolling . . . . .	32
6.4	Movimento della telecamera e potenziale degli estremi . . . . .	34
6.5	Messa in fase delle camere . . . . .	34
<b>7</b>	<b>Tracking di eventi: modellizzazione ed implementazione</b>	<b>39</b>
7.1	Dinamica dell'evento . . . . .	39
7.2	Filtro di Kalman e coordinazione tra le camere . . . . .	40
7.3	Implementazione . . . . .	41

<b>8</b>	<b>Simulazioni</b>	<b>44</b>
8.1	Dimensionamento dei parametri ed analisi del transitorio . . . . .	44
8.1.1	Stima della velocità di convergenza . . . . .	44
8.1.2	Dipendenza dai parametri . . . . .	45
8.2	Simulazioni con comunicazione frequente . . . . .	46
8.3	Verifica del tempo di assestamento e limiti del sistema . . . . .	47
8.4	Simulazioni in presenza di evento . . . . .	51
<b>9</b>	<b>Conclusioni e Sviluppi futuri</b>	<b>54</b>
9.1	Estensione al 1.5D . . . . .	54
9.2	Osservazioni conclusive . . . . .	55
<b>A</b>	<b>Approfondimenti sulla traiettoria ottima con vincoli</b>	<b>56</b>
<b>B</b>	<b>Ulteriori simulazioni</b>	<b>60</b>
<b>C</b>	<b>L'approccio iniziale: i funzionali</b>	<b>64</b>
<b>D</b>	<b>Codice Matlab</b>	<b>69</b>
D.1	Dati.m . . . . .	69
D.2	Funzione Evento.m . . . . .	71
D.3	StimaTempoConvergenza.m . . . . .	72
<b>E</b>	<b>Schemi simulink</b>	<b>73</b>
<b>F</b>	<b>Rendering 3D delle simulazioni</b>	<b>75</b>

# Abstract

*L'obiettivo del progetto è implementare degli algoritmi distribuiti per realizzare patrolling e tracking per un sistema di videosorveglianza con vincoli di comunicazione, di risorse e di velocità di inseguimento. È stato implementato un algoritmo in grado di trovare le sezioni di copertura ottime, anche in presenza di vincoli fisici di copertura. Le telecamere eseguono il patrolling secondo quanto dettato da tale algoritmo; per il tracking, inteso non come problema di visione ma di inseguimento coordinato, si è fatto uso di un filtro di Kalman implementato in ogni telecamera. Le simulazioni, che hanno reso ragione all'approccio teorico sviluppato, sono state fatte in ambiente Simulink, sfruttando un'analogia fisica per ottenere un comportamento intrinsecamente distribuito. L'utilizzo di Simulink ha consentito, tra l'altro, di impostare vincoli di comunicazione ed asincronismo. Si è realizzato anche un ambiente per delle simulazioni 3D, che rendono ancora più visibile l'efficacia del metodo proposto.*

## 1 Introduzione

Effettuare il patrolling di un perimetro significa continuare a percorrere il percorso assegnato in modo tale da visitarne ogni punto il più spesso possibile, per proteggerlo o semplicemente per osservarne lo stato [1]. Da ciò si deduce immediatamente che una buona strategia di patrolling consiste nel minimizzare l'intervallo di tempo tra due successive visite di uno stesso punto. Tale fatto in letteratura è generalmente dato per scontato data la sua immediatezza, tuttavia in questo progetto se ne è comunque fornita prova (vedere sezione 4, proposizione 1). Il problema del patrolling, per come è posto, trova naturale applicazione in molteplici campi: sistemi di difesa militari, veicoli sottomarini o robot mobili atti al monitoraggio di certe aree o al perimeter detection, industrie chimiche o nucleari (tracking di perdite chimiche o radiattive) etc. Si deduce altresì come il problema in questione trovi naturale applicazione nel campo della videosorveglianza, dove sono utilizzate reti di videocamere per monitorare perimetri, aree esterne o interne. Dal monitorare un'area ad inseguire un evento (fare cioè il tracking) il passo è decisamente breve; le camere devono dunque essere in grado di fornire immagini a varie risoluzioni (importante per l'identificazione di persone o targhe), possibilmente da diverse prospettive e con campi di visione (field of view) sufficientemente ampi in modo tale da seguire agevolmente il moto degli eventi. Le videocamere che si adattano meglio a tutte queste necessità sono le PTZ (Pan Tilt Zoom). Una rete di tali camere sarà composta da un numero minore di agenti rispetto ad una rete con camere fisse ([3]), infatti una singola camera PTZ ha più potenziale di copertura di un'altra impossibilitata a muoversi. Nasce però il problema di controllare tali telecamere. La soluzione più comunemente adottata è il controllo visivo da parte di un operatore umano, soluzione che presenta una serie di svantaggi: l'operatore deve essere costantemente concentrato su più monitor, col rischio di perdere particolari rilevanti; quando si verifica un evento la sua attenzione si concentra su di esso, trascurando il

resto dell'area da sorvegliare; se quest'ultima inoltre è di dimensioni notevoli, non è sufficiente un solo operatore. Queste problematiche inducono la possibilità di migliorare il sistema di controllo passando ad un operatore automatico (calcolatore). Esso ovviamente difetterà delle caratteristiche peculiarmente umane, come il riconoscimento di certi eventi particolari, ma in questo progetto non è richiesto di considerare la visione come problematica. Dando dunque per buone le capacità percettive di un controllore, si presenta un'ulteriore questione: approccio centralizzato o distribuito? Convenzionalmente, l'obiettivo di patrolling è implementato da una struttura centralizzata, con un calcolatore centrale che gestisce le informazioni e comanda il movimento di ogni agente coinvolto. Il controllo centralizzato si basa su ottimizzazione e comunicazione globale, in tal modo la complessità computazionale cresce col numero degli agenti utilizzati; non sarà poi possibile gestire contemporaneamente problematiche complesse, dato che al calcolatore devono essere assegnate delle priorità. Il controllo decentralizzato consiste invece nel fatto che ogni telecamera, o agente, è "intelligente" (dotato di microprocessore) e decide come comportarsi in funzione di ciò che vede e delle informazioni che gli arrivano dai vicini. Il meccanismo di controllo è dunque basato su informazioni locali. Tale sistema inoltre è più robusto, sia nel caso di eventuale perdita di agenti, sia per la gestione di eventi complessi: grazie alla comunicazione tra vicini, le camere sono in grado di ottimizzare la loro azione di patrolling anche qualora qualcuna di esse interrompesse tale azione per seguire altre priorità, come appunto il tracking di un evento. Altro aspetto che si darà per scontato in queste pagine (oltre a quello già menzionato della visione) è la corretta dislocazione delle telecamere (o più in generale di sensori) nell'ambiente al fine di realizzare al meglio la copertura dell'area e permettere agli algoritmi sviluppati di funzionare appropriatamente (tale problematica è nota in letteratura anche col nome di "art gallery problem"); verranno fatte delle ipotesi di buon senso che saranno anche rilassate, per maggiori dettagli si vedano [4, 21]. Viene tuttavia ipotizzata una certa sovrapposizione delle zone di copertura delle telecamere, per giustificare la ricerca di un algoritmo di ottimizzazione del patrolling: se le zone da controllare fossero tutte disgiunte, non ci sarebbe problema di coordinazione: semplicemente ogni camera controllerebbe la frazione di perimetro a lei assegnata. L'ottimizzazione potrà essere vincolata o meno, a seconda della presenza di vincoli fisici imposti alle telecamere. L'ottimalità di una legge di controllo per il patrolling può essere valutata sia rispetto all'obiettivo in sé sia rispetto alla robustezza ([10]): l'algoritmo proposto in questo progetto lo è sotto entrambi gli aspetti, infatti è distribuito e dunque scalabile e robusto, ma soddisfa anche un criterio di ottimalità; essendo inoltre basato su un'implementazione asincrona si adatta molto bene alle situazioni di comunicazione reale: può essere implementato in una rete di agenti che si muovono a velocità diverse, con capacità computazionali e di comunicazione differenti.

Un buon sistema di videosorveglianza, come già accennato, dev'essere anche in grado di effettuare il tracking, ovvero l'inseguimento di un evento. Essendo le telecamere intelligenti, la cosa più naturale è implementare in esse un filtro di Kalman: esse sono così in grado di seguire un evento, qualora ciò sia consentito dai loro vincoli fisici e dinamici. Tuttavia limitarsi a seguire l'evento da parte delle singole camere sarebbe riduttivo: a buon senso l'inseguimento di un intruso è operazione della massima impor-

tanza, per la quale si può anche sacrificare l'ottimalità della contemporanea azione di patrolling. È stato fatto in modo quindi che la telecamera impegnata nel tracking passi alle vicine tale informazione, in modo tale che esse sappiano cosa succede e siano pronte ad inseguire l'evento qualora esso entri nella loro porzione di perimetro da sorvegliare.

Nella sezione successiva viene presentato lo stato dell'arte, nella sezione 3 si affronta il problema dell'ottimizzazione del patrolling dal punto di vista strettamente teorico, nella sezione 5 viene presentata un'interessante analogia con un sistema di molle al fine di rendere più semplice l'implementazione con Simulink, nella sezione 6 viene spiegato nel dettaglio il procedimento di implementazione, nella sezione 7 si spiega come è stato realizzato il tracking e infine nella sezione 8 vengono riportate le simulazioni più significative.

## 2 Stato dell'arte

La tematica generale nella quale si può inquadrare un sistema di telecamere è quella della rete di sensori. Tale campo di studi è stato debitamente indagato ed approfondito in letteratura, sviluppando problematiche di implementazione, scheduling, controllo, copertura di obiettivi selezionati. Per un'esauriente trattazione, abbastanza generale, si veda [16]. Il problema principale che si presenta facendo uso di una rete di sensori è la copertura di aree di interesse. Gli strumenti matematici che vengono utilizzati in questi casi sono i più disparati: per un esempio di controllo distribuito, si veda [17], nel quale lo spazio da coprire è modellato con una funzione densità rappresentante la frequenza del manifestarsi degli eventi. Un altro problema di copertura, in questo caso per reti wireless di sensori, è affrontato in [18] e viene risolto ricorrendo a celle di Voronoi e grafi. Un particolare sottoinsieme delle reti di sensori è composto dagli agenti mobili, in genere robot in grado di muoversi (reti multi-veicolo) o UAV (Unmanned air vehicle): per essi si presentano problemi come il coordinamento e lo scambio di informazioni, dal momento che in genere non vi è un sistema centralizzato ma si vuole comunque ottenere una copertura globale dell'obiettivo. In letteratura si trovano vari algoritmi che prevedono approccio distribuito, robustezza in caso di perdita di un agente, limiti di comunicazione e sensibilità: si veda ad esempio [15] nel quale si fa uso degli algoritmi di Loyd e delle celle di Voronoi per realizzare un algoritmo di controllo asincrono, distribuito ed adattativo.

Le tecniche adottate qui e in altri casi possono essere degli ottimi spunti per il campo della videosorveglianza, nel quale si riscontra buona parte delle problematiche viste finora: gli agenti sono le camere e presentano vincoli di vario genere, prima di tutto sul movimento. Dal momento che le analogie con le reti multi-veicolo sono molte, quanto presente in letteratura sulle seconde può essere riportato, con i dovuti accorgimenti, sulle prime.

Il tema del progetto in questione si articola lungo due filoni principali, il patrolling ed il tracking.

Il patrolling, come già detto, trova applicazione non solo in sistemi di telecamere. In

[1] viene effettuata un'analisi teorica del problema di patrolling, chiedendosi se esistono strategie ottime o sub-ottime ed algoritmi in grado di realizzarle, e si affrontano altri problemi teorici inerenti il patrolling, senza curarsi degli aspetti implementativi. L'analisi svolta viene effettuata basandosi sui grafi. In [2] gli autori introducono un algoritmo per il multi-agent patrolling basato su un'analogia fisica: dei palloni riempiti di gas. In tale algoritmo gli agenti partizionano dinamicamente il grafo rappresentante l'area da monitorare ed ogni agente diventa responsabile del proprio sottografo. Gli autori intuiscono che esiste un bilanciamento equilibrato dei sottografi ma non riescono a dimostrarne l'esistenza, nonostante le simulazioni la confermino. Molto interessante si è rivelato [5], nel quale viene presentato e risolto in maniera distribuita un problema di sorveglianza perimetrale cooperativa multi-agente; la soluzione è robusta sia perchè si adatta a variazioni del perimetro, sia perchè riesce a sopperire all'eventuale perdita di un agente. Il punto cruciale di tale soluzione è la comunicazione a due a due tra gli agenti che avviene quando essi si incontrano, fatto reso possibile effettuando il patrolling in opposizione di fase. I risultati dell'articolo vengono poi adattati ad un gruppo di UAV. Essi vengono utilizzati come agenti per monitorare aree ed effettuare tracking di perimetri anche in [23]. Ugualmente interessante si è dimostrato l'articolo di Yang et al. [10] che, oltre ad implementare un approccio distribuito per il patrolling, suggeriscono l'originale analogia con un sistema di masse e molle: tale intuizione nasce dal fatto che il modello di controllo utilizzato rappresenta sostanzialmente una molla virtuale, da tale fatto si passa poi al confronto con un sistema fisico molla-massa-molla. In tale algoritmo il patrolling viene applicato ad un sistema biologico (vengono infatti studiati i linfociti T); tale analogia è evidentemente adatta all'applicazione del problema di patrolling in quanto viene effettuata anche in [13], nel quale si basa sulle tracce di feromone, utilizzate come potenziale attraente per gli agenti verso le aree che non sono state visitate recentemente: si utilizza dunque un funzionale locale crescente col tempo. Altro esempio di utilizzo di potenziali è fornito da [22] nel quale si ricorre a funzioni potenziali che uniscono pianificazione cinematica e dinamica per la pianificazione del moto di una flotta di robot. In [7, 8] viene proposto un algoritmo per coprire dinamicamente una regione usando degli agenti mobili: il problema viene affrontato in maniera rigorosa dal punto di vista teorico e porta alla definizione di leggi di controllo che garantiscono la copertura effettiva ed evitano il flocking e le collisioni tra agenti. Tali leggi sono locali, ovvero ogni agente utilizza le informazioni degli agenti con cui è in grado di comunicare. Per quanto riguarda l'individuazione di perimetri inizialmente incogniti ed il loro successivo tracking, ovvero il pattugliamento "dinamico", nel senso che si adatta alle variazioni del perimetro in questione, si veda anche [6], interessante per l'utilizzo di potenziali e di variabili d'ambiente discrete e continue.

Come detto in precedenza, le tecniche viste finora si possono adattare al problema delle reti di telecamere e dunque della videosorveglianza. La letteratura riguardante strettamente le telecamere è vasta ed eterogenea, e spesso sfiora nel campo della visione, dal quale era specificamente richiesto di mantenere le distanze. Per avere un'idea generale di quanto è già stato affrontato finora in letteratura in tale campo, si veda [12], interessante review di come siano evoluti nel tempo i sistemi di videosorveglianza intelligenti, sotto vari punti di vista: algoritmi di visione per image processing, inserimento di tali

algoritmi in sistemi distribuiti, strategie implementative per la distribuzione e la comunicazione, anche a seconda che il sistema di visione sia stereo o monoculare. Gli autori di [3] propongono un algoritmo distribuito (basato unicamente sulla comunicazione tra vicini) per coordinare un sistema di telecamere preposto all'identificazione di persone. In un primo approccio si utilizzano camere fisse ed il problema viene risolto con tecniche di visione; successivamente si passa alle telecamere PTZ. Viene inoltre posto il problema di scegliere quale evento seguire, qualora se ne presentino più di uno nella stessa scena; vengono tuttavia fatte delle assunzioni un pò troppo specifiche, come disporre di camere PTZ molto più veloci dell'evento e che non ci sia significativo overlap tra le regioni viste dalle camere. Si veda anche [9] per un'applicazione con camere fisse e PTZ coordinate in maniera gerarchica. Le telecamere presentano anche altri aspetti da tenere in considerazione: in [19] viene proposto un modello di pattugliamento virtuale per operazioni di sorveglianza in reti di sensori; l'obiettivo è mantenere un buon livello di sorveglianza anche impostando alcuni sensori in modalità "sleep" per risparmiare energia, dal momento che ad un buon sistema di sorveglianza si richiede, tra l'altro, di operare a lungo termine gestendo sapientemente il passaggio dei sensori da "working mode" a "sleeping mode".

Per quanto concerne il tracking, è difficile trovare in letteratura qualcosa di completamente estraneo al campo della visione: spesso gli articoli propongono soluzioni per inseguire facce o oggetti specifici, oppure sfruttano il movimento degli oggetti per capire la dislocazione delle camere nello spazio. Qualche soluzione tuttavia esiste, ed in genere si basa sul filtro di Kalman. Un esempio è fornito da [11], nel quale viene realizzato un sistema autonomo di videosorveglianza che però fa capo ad un processore centrale, il quale permette all'operatore umano di visualizzare quanto ripreso dagli agenti (telecamere PTZ) ed eventualmente di comandarne alcuni. Altra applicazione del filtro di Kalman si trova in [20], nel quale esso viene applicato con successo a telecamere PTZ. L'articolo si concentra anche sull'azionamento ed il controllo della camera.

Per concludere, una curiosità: si veda [14] per le implicazioni legali dell'utilizzo di sistemi distribuiti ed intelligenti di videosorveglianza, con riferimento alla legislazione del Regno Unito.

### 3 Analisi teorica del problema di patrolling

In queste sezioni si illustrerà il metodo proposto per la risoluzione del problema in questione: si mostrerà come, procedendo per piccoli passi, si è giunti a concepire un algoritmo distribuito, implementabile ed efficace nel risolvere il problema di patrolling anche in presenza di un evento da monitorare. Tale algoritmo è stato concepito per la risoluzione di problemi di copertura monodimensionali (1D) ma tuttavia presenta una naturale estensione a situazioni più complesse modellabili in 1.5D.

### 3.1 Definizione delle variabili del problema

Nel seguito del progetto si farà uso delle seguenti variabili:

1.  $L = [0, L_{tot}] \subset \mathbb{R}^+$ , lunghezza complessiva da monitorare
2.  $N$ , numero delle telecamere (numero di agenti)
3.  $C_i = [C_{i,inf}, C_{i,sup}] \subset L$ , range di copertura fisica della telecamera i-esima
4.  $v_i \in [-V_{i,max}, +V_{i,max}]$ , range di velocità ammissibile per la telecamera i-esima
5.  $z_i(t) : \mathcal{C}[0, +\infty] \rightarrow C_i$ , posizione del centro camera della telecamera i-esima
6.  $A_i \subseteq C_i$ , sezione di copertura a regime della telecamera i-esima

Ci si è posti in un caso del tutto generale: la copertura fisica e la velocità di monitoraggio sono parametri propri di ciascun agente e possono essere molto diversi tra camera e camera.

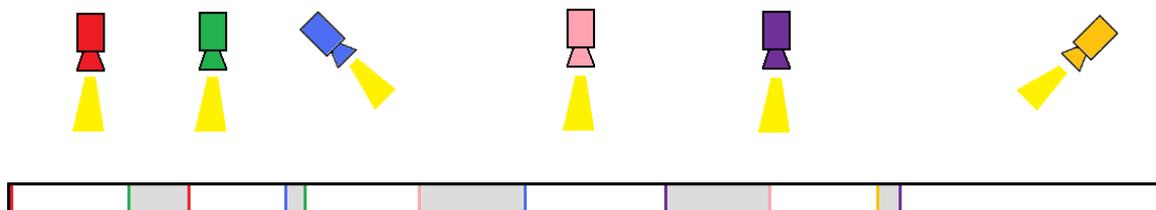


Figura 1: Percorso da monitorare con i range di copertura fisici delle camere

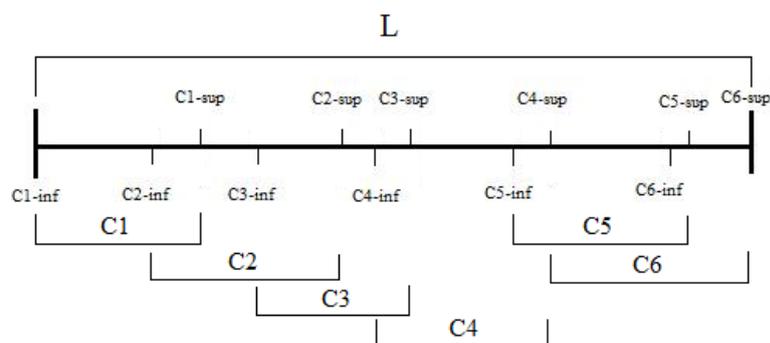


Figura 2: Schema di un percorso da monitorare con i range di copertura fisici delle camere

### 3.2 Criterio di ottimalità

Per definire un criterio di copertura ottimale si deve innanzitutto scegliere un indice adeguato rispetto a cui operare un certo tipo di minimizzazione o massimizzazione. Si è scelto il seguente indice:

$$J(x, t, z(t)) : L \times \mathbb{R}^+ \times \mathcal{C}^N(\mathbb{R}^+) \rightarrow \mathbb{R}^+$$

che vale:

$$J(x, t, z(t)) = \begin{cases} 0 & \text{se all'istante } t \exists i \mid z_i(t) = x \\ g(\bar{t}(x)) & \text{altrimenti} \end{cases}$$

con:

- $z(t) = [z_1(t) \dots z_N(t)]'$
- $x \in L$
- $t \in \mathbb{R}^+$
- $\bar{t}(x) : L \rightarrow \mathbb{R}^+$ , tempo trascorso dall'ultima visita in  $x$  di una telecamera (tempo trascorso dall'ultima volta in cui  $z_i(t) = x$ , per un qualche  $i$ )
- $g(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ , funzione monotona strettamente crescente tale che  $g(0) = 0$

Le condizioni iniziali per  $t = 0$  sono:

- $z_i(0)$  variabili aleatorie con distribuzione uniforme in  $C_i$
- $\bar{t}(x) = 0, \forall x \in L$

In questo modo il funzionale  $J(\cdot, \cdot, \cdot)$  risulta inizializzato:

$$J(x, 0, z(0)) = 0, \quad \forall x \in L$$

**Osservazione 1.** Il significato del funzionale  $J(\cdot, \cdot, \cdot)$  è il seguente: ad ogni istante di tempo  $t$  nel percorso in posizione  $x$  esso assume il valore 0 se il punto è osservato da una telecamera, mentre assume un valore reale positivo tanto più grande quanto più tempo è trascorso dall'ultima visita di una qualsiasi telecamera in quel punto ( $g(\cdot)$  è una funzione crescente). Questa scelta è strettamente legata all'obiettivo del progetto che è quello di stabilire un criterio di copertura ottimale con l'ulteriore necessità di effettuare il tracking di eventuali eventi.

**Osservazione 2.** Ci si è posti implicitamente nell'ipotesi di *field of view* (cono di visione delle telecamere) *puntiforme* ( $fov(i) = 0, \forall i$ ): a ciascuna camera infatti è associato il moto di un punto materiale ( $z_i(t)$ ) e si considera che un punto del percorso  $x$  è visto da una telecamera se e solo se  $\exists i \mid z_i(t) = x$ . Tale ipotesi tuttavia non toglie generalità al procedimento esposto: noti i coni di visione (che possono in generale essere diversi da camera a camera) ci si riconduce al caso puntiforme considerando che

il percorso da monitorare, invece di avere lunghezza  $L_{tot}$ , abbia una lunghezza effettiva pari a:

$$L_{eff} = L_{tot} - \sum_{i=1}^N fov(i)$$

e si effettuano sul percorso di lunghezza  $L_{eff}$  le medesime considerazioni che si faranno nel seguito.

### 3.3 Dinamica di controllo

L'obiettivo è quello di determinare un'opportuna legge di controllo che garantisca di operare un certo tipo di minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$ . Si consideri la dinamica delle telecamere:

$$\dot{z}_i(t) = v_i(t), \quad \forall i$$

Una scelta naturale per il sistema in questione è quella di considerare come ingresso di controllo le velocità  $v_i$ :

$$U(t) = [u_1(t) \dots u_N(t)]' = [v_1(t) \dots v_N(t)]', \quad t \in [0 + \infty)$$

Si è quindi deciso di impostare la dinamica di controllo secondo il seguente criterio di minimizzazione:

$$\bar{U}(\cdot) = \arg \min_{U(\cdot)} \sup_t \max_x J(x, t, z(\cdot)) \quad (1)$$

con vincoli:

$$u_i(t) = v_i(t) \in [-V_{i,max}, +V_{i,max}] \quad (2)$$

$$z_i(t) \in C_i \quad (3)$$

Il problema appare molto complesso: tuttavia si farà vedere come, procedendo per piccoli passi, si è riusciti a determinare una traiettoria a regime che soddisfa tale criterio di ottimalità. Prima di illustrarla tuttavia conviene soffermarsi sulla scelta del tipo di minimizzazione effettuata, in relazione anche alla natura dell'evento.

#### 3.3.1 Tipo di minimizzazione

Si è scelto di operare la minimizzazione (1), che concettualmente assomiglia ad una minimizzazione in  $\|\cdot\|_\infty$  del funzionale  $J(\cdot, \cdot, \cdot)$ , poichè essa corrisponde intuitivamente, per come è stato definito il funzionale, a garantire un *worst case* di tempo di non visita  $T_{max}$  per ciascun punto del percorso, scelta ragionevole per il problema in questione. Si noti come sarebbe stato possibile procedere anche cercando di operare una minimizzazione del tipo  $\|\cdot\|_1$  (o anche  $\|\cdot\|_2$ ): in tal caso si sarebbe ottimizzato il tempo medio di visita di ciascun punto; tale soluzione però è stata scartata per due motivi: innanzitutto intuire e dimostrare quale sarebbe stata la traiettoria ottima (e quindi il controllo da applicare) era molto più difficile rispetto al caso scelto ed in secondo luogo, motivazione non meno importante, si intuiva il rischio di ottenere una traiettoria a regime in cui alcuni punti del percorso (ad esempio gli estremi) avrebbero avuto

tempo massimo di non visita maggiore, e questo avrebbe potuto essere un problema per cercare di risolvere in modo efficiente anche la questione del tracking dell'evento, come si mostrerà nella prossima sezione in cui si avvalora la scelta di minimizzazione effettuata.

### 3.3.2 Modello probabilistico dell'evento

Si supponga che per l'evento  $e$  valgano le seguenti ipotesi:

1. la probabilità a priori che si verifichi  $e$  è la stessa in ogni punto del percorso (l'evento è uniformemente distribuito nello spazio  $L$ );
2. l'evento  $e$  è raro (cioè si verifica con una probabilità piccola nell'intervallo di tempo considerato);
3. l'evento  $e$  non ha dinamica nel tempo (resta fermo) ed è puntiforme.

Mentre le prime due sono ipotesi molto ragionevoli la terza è restrittiva, tuttavia è necessaria per una prima analisi rigorosa. La prima ipotesi sottolinea il fatto che non ci sono parti del percorso privilegiate in cui è meno probabile si verifichi un evento (e viceversa zone in cui l'evento è a priori più probabile); la seconda invece permette di modellare  $\forall x \in L$  il numero di volte in cui si verifica un evento come un *processo contatore di Poisson*:

$$P_x(N(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}, \quad \forall x \in L$$

Indicando con  $N(t)$  il numero di occorrenze dell'evento  $e$  in posizione  $x$  nell'arco di tempo  $t$ .

Quindi ad ogni punto del percorso è associata una descrizione probabilistica del numero di eventi che si verificano come un processo contatore di Poisson di intensità  $\lambda$ : tale parametro è lo stesso  $\forall x$  per l'ipotesi 1 e dipende dal tipo di evento che si sta considerando ( $\lambda$  può essere interpretato come la media e la varianza del processo per unità di tempo e questi due parametri sono gli stessi per ogni punto del percorso perchè a priori l'evento può verificarsi con pari probabilità in ogni punto). Per poter adoperare il processo contatore di Poisson sono inoltre necessarie le seguenti ipotesi tecniche che nel problema specifico si possono ritenere soddisfatte:

1. non ci sono collisioni ( $N(0) = 0$ );
2.  $N(t)$  ha incrementi indipendenti ( $N(t)$ ,  $N(t, t_1)$ ,  $N(t_1, t_2)$ , ... sono mutuamente indipendenti);
3. stazionarietà di  $N(t)$  ( $N(t_k, t_{k+1}) = N(t_{k+1} - t_k)$ ).

I processi di Poisson godono di queste due importanti proprietà:

1. tempi di interarrivo con distribuzione esponenziale:

$$P[V_k < t | T_{k-1} = t_{k-1}] = P[V_k < t] = 1 - e^{-\lambda t}$$

è la probabilità che si verifichi in meno di  $t$  il  $k$ -esimo evento noto che il  $k-1$ -esimo si è verificato all'istante  $t_{k-1}$  (si noti che in realtà tale probabilità non dipende da  $t_{k-1}$  ma solo dal tempo  $t$  che si sta considerando);

2. assenza di memoria:

$$P[V_k < t + z | V_k > z] = 1 - e^{-\lambda t}$$

In queste ipotesi è possibile fare il seguente ragionamento. Si supponga di scegliere la funzione  $g(\bar{t}(x))$  in questo modo:

$$g(\bar{t}(x)) = 1 - e^{-\lambda \bar{t}(x)}$$

Tale funzione soddisfa le ipotesi necessarie per la definizione dell'indice  $J(\cdot, \cdot, \cdot)$ : è crescente rispetto a  $\bar{t}(x)$  e vale  $g(0) = 0$ . Supponiamo ora di considerare un qualsiasi  $x_0 \in L$ ; per tale punto vale che:

$$g(\bar{t}(x_0)) = 1 - e^{-\lambda \bar{t}(x_0)}$$

La probabilità che in un intervallo di tempo  $\bar{t}(x_0)$  si verifichi almeno un evento in  $x_0$  è pari a:

$$P[N(\bar{t}(x_0)) \geq 1] = 1 - P[N(\bar{t}(x_0)) = 0] = 1 - e^{-\lambda \bar{t}(x_0)} \quad (4)$$

Con questa scelta della funzione  $g(\cdot)$  nelle ipotesi in cui ci si è posti, il valore assunto dal funzionale  $J(\cdot, \cdot, \cdot)$  in  $x_0$  nel tempo è pari alla probabilità che si sia verificato in  $x_0$  almeno un evento dall'ultimo istante di visita di una telecamera e tale considerazione si può estendere a tutti i punti del percorso. Siccome condizione necessaria per poi poter effettuare un buon tracking è come prima cosa accorgersi dell'arrivo di un evento, la scelta di effettuare la legge di controllo così come mostrato nella sezione precedente corrisponde, in questo semplice caso, a minimizzare la probabilità che si verifichi un evento senza che il sistema se ne accorga.

**Evento reale con dinamica.** Si supponga ora di rilassare l'ipotesi 3 e di mettersi nel caso più realistico in cui per *evento* si intende qualcosa che ha una certa dinamica nel tempo (e dunque nello spazio) di cui il sistema si deve accorgere il prima possibile. Per quanto riguarda tale fase di individuazione dell'evento in movimento, si può in ogni caso ricondursi al modello semplificato descritto precedentemente considerando che il numero di eventi  $e$  associato al processo contatore di Poisson in posizione  $x$  è il numero di volte che l'*evento* assume il valore  $x$  durante la sua dinamica; allora valgono gli stessi ragionamenti fatti in precedenza ed imponendo  $g(\cdot)$  della stessa forma di prima si ha che la probabilità in un punto  $x$  che all'istante  $t$  si sia verificato un evento senza che il sistema se ne sia accorto è:

$$P[N(\bar{t}(x_0)) \geq 1] = 1 - P[N(\bar{t}(x_0)) = 0] = 1 - e^{-\lambda \bar{t}(x_0)}$$

Pertanto il controllo che agisce minimizzando l'indice  $J(\cdot, \cdot, \cdot)$  col metodo proposto, anche in questo caso minimizza la probabilità che si verifichi lungo il percorso un evento senza che il sistema se ne accorga.

Vanno sottolineati tuttavia i limiti di questo modello nel caso reale appena indagato; infatti, anche mantenendo valida l'ipotesi che l'evento abbia una distribuzione spaziale uniforme (nel senso che non vi sono a priori zone del percorso in cui è più probabile che esso si trovi), vi possono essere realizzazioni particolarmente sfortunate dell'evento in cui la sua dinamica è tale da nascondersi alle telecamere togliendo quindi un pò di valore all'analisi probabilistica effettuata. Tuttavia è possibile risolvere questo tipo di problematica inducendo nella traiettoria di copertura delle telecamere un'opposizione di fase: in questo modo, supponendo che il moto a regime sia periodico di periodo  $T$ , qualunque sia la dinamica dell'evento si ha la certezza che il sistema se ne accorga in un tempo che al massimo è pari al periodo stesso  $T$ ; resta quindi significativa l'analisi probabilistica effettuata e risulta che la probabilità nel *worst case* (l'evento nasce in un estremo appena dopo il passaggio delle due telecamere e resta non visibile fino al loro passaggio successivo) che il sistema non si accorga dell'evento è:

$$p = 1 - e^{-\lambda T} \quad (5)$$

pertanto minimizzare questa probabilità è ancora una volta equivalente alla minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$  proposta perchè, come si farà vedere nella prossima sezione, tale scelta conduce a determinare una traiettoria ottima periodica di periodo minimo  $T$  (e minimizzare il periodo  $T$  significa minimizzare  $p$  come si può osservare nella formula 5).

## 4 Calcolo della traiettoria ottima

Risolvere in forma chiusa il problema di minimizzazione posto per determinare la legge di controllo ottima risulta essere, come detto precedentemente, complicato. Tuttavia procedendo per piccoli passi è stato possibile intuire quale fosse la traiettoria ottima (e di conseguenza la legge di controllo da applicare in termini di velocità) e dimostrarne l'ottimalità rispetto a tutte le altre possibili con una logica per assurdo.

### 4.1 Traiettoria ottima senza vincoli di copertura

**Proposizione 1.** *La copertura ottima rispetto alla minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$  con il criterio (1), nel caso di una sola telecamera che deve pattugliare la sezione  $L$ , si ottiene imponendo alla telecamera un moto periodico alla sua velocità massima con periodo  $T$  pari a:*

$$T = \frac{2L_{tot}}{V_{max}}$$

*In tal caso il valore massimo assunto dal funzionale nel tempo si ha negli estremi ( $x = 0 \vee x = L_{tot}$ ) e vale:*

$$\max_{x,t} J(x, t, z(t)) = \max_t \{J(0, t, z(t)), J(L_{tot}, t, z(t))\} \leq 1 - e^{-\lambda T}$$

Si suppone quindi di aver utilizzato la funzione  $g(\cdot)$  nella forma:

$$g(\bar{t}(x)) = 1 - e^{-\lambda \bar{t}(x)}$$

per i motivi esposti precedentemente.

*Dimostrazione.* Consideriamo per assurdo un qualsiasi moto diverso da quello supposto ottimale che copra l'intero percorso. Supponiamo inoltre per semplicità di partire dall'estremo sinistro ( $x = 0$ ), caso che non toglie generalità alla dimostrazione in quanto si potrebbe sempre iniziare a fare le seguenti considerazioni quando la telecamera raggiunge tale estremo per la prima volta. Vale allora senz'altro che il tempo massimo di non visita dell'estremo destro  $\bar{T}$  (durante la prima passata) è in tal caso:

$$\bar{T} > \frac{T}{2} = \frac{L_{tot}}{V_{max}}$$

in quanto l'imporre una velocità minore della  $V_{max}$  o compiere una traiettoria di moto non rettilineo uniforme, fa perdere tempo alla telecamera (all'andata ed in modo simmetrico al ritorno) e quindi almeno per gli estremi del percorso il valore massimo assunto nel tempo dal funzionale  $J(\cdot, \cdot, \cdot)$  è maggiore di quello che si otterrebbe con la traiettoria proposta, andando cioè a velocità costante pari a  $V_{max}$  e compiendo un moto di periodo  $T$ .  $\square$

Quanto appena dimostrato in questo semplice caso, induce alla seguente considerazione: la copertura ottima del percorso, anche nel caso generale, si otterrà imponendo alle telecamere di andare ciascuna alla sua velocità massima, compiendo un moto periodico nelle loro sezioni di copertura  $A_i$ . In questo modo il problema della minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$ , si riduce alla scelta delle  $A_i$  a regime per ciascuna telecamera. Questo concetto è ben espresso dalla seguente proposizione.

**Proposizione 2** (Algoritmo 1). *La copertura ottima del percorso rispetto alla minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$  con il criterio espresso nell'equazione 1 senza i vincoli di copertura (3), nell'ipotesi che ciascuna telecamera vada alla sua velocità massima ( $V_{i,max}$ ) e pattugli la propria sezione con un moto di periodo  $T_i$ , si ottiene imponendo a tutte le telecamere uno stesso periodo  $T$  e delle sezioni di copertura  $A_i$  disgiunte:*

$$T = 2T_{ott} = \frac{2L_{tot}}{\sum_{i=1}^N V_{i,max}}$$

e con:

$$|A_i| = V_{i,max} T_{ott}$$

Dove con  $|A_i|$  si intende la misura delle sezioni ottime. Il percorso è quindi suddiviso in  $N$  sezioni disgiunte, ciascuna pattugliata da una telecamera alla sua velocità massima in un tempo  $T_{ott}$  pari a metà periodo.

**Osservazione 3.** Nelle ipotesi in cui ci è messo l'azione di controllo nelle velocità serve solo per determinare i bordi delle sezioni  $A_i$  di copertura effettiva a regime, in quanto le  $v_i$  sono fissate al loro valore massimo.

**Osservazione 4.** E' fondamentale osservare come in questo calcolo per determinare la traiettoria ottima, non si tenga conto dei vincoli fisici di copertura delle telecamere: l'unica cosa implicitamente richiesta è che la sezione di copertura della telecamera  $i$ -esima segua quella della camera  $(i-1)$ -esima,  $\forall i$ . Nella prossima sezione si vedrà come si è generalizzata la soluzione proposta, tenendo conto dei vincoli (3).

**Idea della soluzione ottima proposta.** Prima di procedere con la dimostrazione, può essere utile intuire la logica che conduce alla soluzione proposta.

Sia  $\bar{T}$  il tempo massimo necessario per visitare a regime tutti i punti del percorso (ciò significa che a partire da una situazione a regime e da un generico istante  $T_0$ , in  $T_0 + \bar{T}$  si visitano tutti i punti). Senz'altro in  $\bar{T}$  vale allora che la somma degli spazi percorsi dalle telecamere è maggiore (o almeno uguale) all'intero percorso da monitorare:

$$\sum_{i=1}^N s_i \geq L_{tot}$$

Posto che a regime ciascuna telecamera si muova a velocità pari alla sua  $V_{max}$  si ottiene:

$$\sum_{i=1}^N V_{i,max} \bar{T} \geq L_{tot} \quad \Rightarrow \quad \bar{T} \geq \frac{L_{tot}}{\sum_{i=1}^N V_{i,max}}$$

ma minimizzare l'indice  $J(\cdot, \cdot, \cdot)$  con il criterio 1 intuitivamente è equivalente a minimizzare proprio il tempo  $\bar{T}$  necessario per visitare tutti i punti. Dunque

$$T_{ott} = \frac{L_{tot}}{\sum_{i=1}^N V_{i,max}}$$

che tornando indietro significa far valere l'uguaglianza nella relazione iniziale:

$$\sum_{i=1}^N s_i = L_{tot}$$

Si analizzi attentamente l'ultima uguaglianza. Si sta dicendo che la somma degli spazi percorsi in  $\bar{T} = T_{ott}$  è proprio pari all'intera lunghezza da monitorare; questo può voler dire solo una cosa: all'ottimo ciascuna telecamera monitora una parte del percorso ( $A_i$ ) che le è preposta, con  $T_{ott}$  pari al tempo necessario per effettuare una spazzolata completa da un estremo all'altro; vale quindi:

$$T = T_{oscill} = 2 T_{ott}$$

con  $T_{max} = T$  (il caso peggiore in termini di tempo di visita si verifica sempre agli estremi di copertura delle telecamere).

Ricapitolando: esiste un ottimo a regime per il problema di minimizzazione posto, in cui ciascuna telecamera monitora a velocità massima una sezione  $A_i$  ad essa preposta, le  $A_i$  hanno intersezione nulla e da questo punto di vista non c'è bisogno di alcuna coordinazione tra le telecamere (nel senso che non è necessario indurre un comportamento in fase o in opposizione di fase).

*Dimostrazione.* Si riepilogano brevemente le ipotesi del problema:

1. Le telecamere hanno un *field of view* puntiforme associato al loro moto  $z_i(t)$
2. Le telecamere si muovono con velocità pari in modulo a  $V_{i,max}$ , compiendo un moto periodico nella loro sezione  $A_i$  di copertura
3. Non si considerano i vincoli di copertura fisici delle telecamere

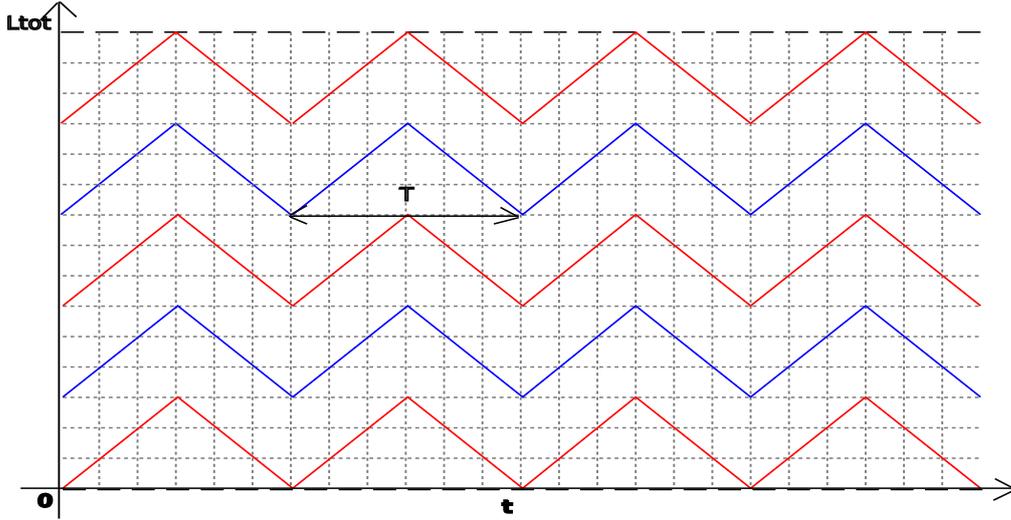


Figura 3: Traiettoria ottima (caso particolare: camere in fase e con la stessa  $V_{max}$ )

Supponiamo per assurdo di apportare una qualsiasi modifica alla traiettoria ottima proposta; è possibile allora:

1. Modificare le sezioni  $A_i$  mantendole però disgiunte
2. Modificare le sezioni  $A_i$  creando sovrapposizione

Nel primo dei due casi si conlude immediatamente con un assurdo: una modifica così fatta delle  $A_i$  produce una sezione  $A_j$  il cui pattugliamento per la telecamera j-esima necessita un tempo  $\bar{T}$  maggiore del  $T_{ott}$  della soluzione proposta, pertanto almeno agli estremi di tale sezione si è certi che il funzionale  $J(\cdot, \cdot, \cdot)$  assuma valori maggiori di prima (l'analisi è la stessa della proposizione 1).

Nel secondo caso, come suggerito dall'euristica della dimostrazione proposta precedentemente, avere delle sezioni  $A_i$  in parte sovrapposte obbliga le telecamere a percorrere uno spazio totale  $S > L_{tot}$  e questo non può che far aumentare il  $T_{max}$  di visita in qualche punto del percorso, con conseguente aumento del valore massimo assunto dal funzionale nel tempo.

Si analizza ora più nel dettaglio come una qualsiasi modifica delle  $A_i$  porti effettivamente ad un valore massimo dell'indice peggiore del valore precedente. Infatti, come già sottolineato, per modificare le  $A_i$  si deve necessariamente fare in modo che una di esse abbia misura superiore a quella precedente (se una delle  $A_i$  viene ridotta, ce ne sarà un'altra che dovrà crescere per ottenere alla fine la copertura totale). Si indichi con  $\{\bar{A}_i\}$  la nuova partizione:  $\exists i \mid |\bar{A}_i| > |A_i|$ ; sia  $\delta = |\bar{A}_i| - |A_i|$ . Si supponga di aver incrementato l'estremo superiore della sezione  $\bar{A}_i$ , e che tale estremo superiore venga raggiunto dalla camera  $i$  durante il suo moto periodico a  $V_{i,max}$ . Il fatto di aver incrementato l'estremo superiore della sezione  $\bar{A}_i$  comporta che all'estremo inferiore non si riesce a ritornare entro il tempo ottimo di copertura. Si riesce ad assicurare la copertura entro il tempo ottimo solo in una fascia di altezza  $|\bar{A}_i|$ , in quanto la base del triangolo che si ottiene in questo modo misura proprio  $2|\bar{A}_i|/V_{i,max} = T$ . Bisogna richiedere quindi che al di sotto della base di tale triangolo cada un punto della traiettoria della camera  $i-1$ . Ciò può accadere in due modi:

- traslando la  $\bar{A}_{i-1}$  verso l'alto di  $\delta$ , ovvero incrementando di  $\delta$  entrambi i suoi estremi, se il periodo della camera  $i - 1$  è minore di quello della  $i$  (figura 4);
- allargando la  $\bar{A}_{i-1}$  verso l'alto di  $\delta$  (figura 5), ovvero incrementando di  $\delta$  solo l'estremo superiore.

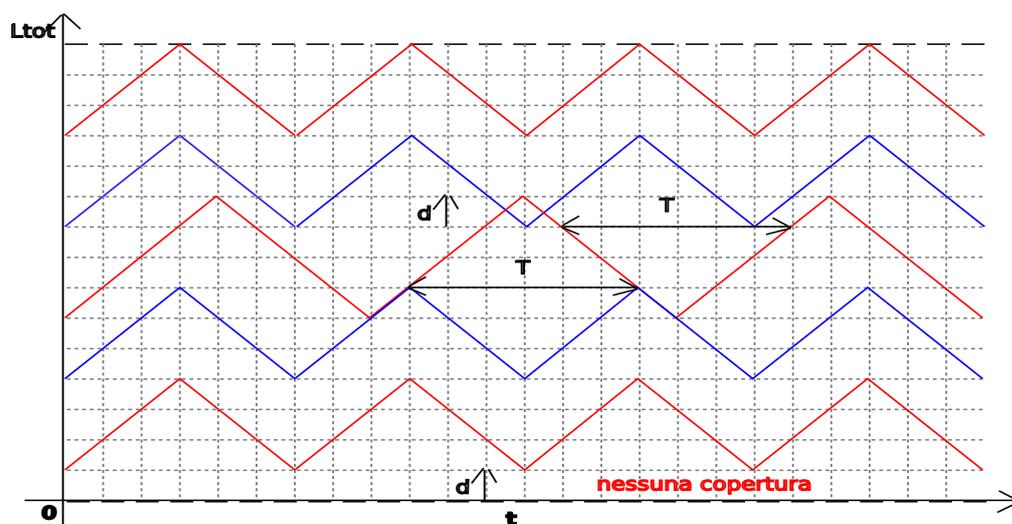


Figura 4: Traiettoria non ottima con traslazione delle camere inferiori

Entrambe le soluzioni proposte ripresentano il problema precedente alla base della  $A_{i-1}$ , dove si ha una carenza di copertura in una fascia di ampiezza  $\delta$ . Procedendo in questo modo si propaga questa carenza verso l'estremo inferiore del percorso, dove però non è possibile compensare in nessun modo: al bordo si avrà una fascia di ampiezza

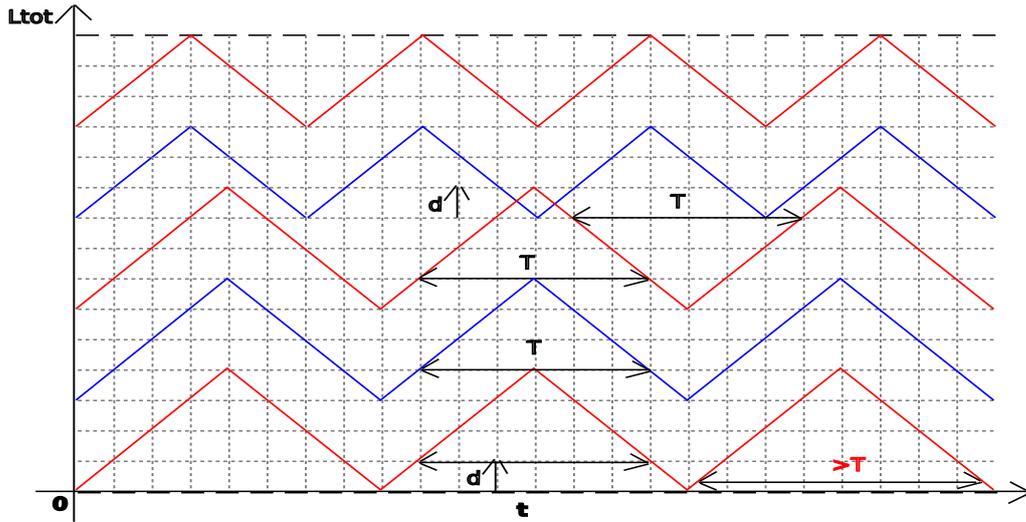


Figura 5: Traiettorie non ottima con dilatazione delle camere inferiori

$\delta$  (eventualmente riscalata nel caso di velocità diversa tra le camere) in cui il tempo massimo tra una copertura e l'altra supera il valore ottimo pari a  $T$ .  $\square$

## 4.2 Traiettorie ottima con vincoli di copertura

Prima di determinare la traiettoria della copertura ottima tenendo conto dei vincoli di copertura fisici delle camere, si analizzano in modo esplicito le condizioni che tale traiettoria deve soddisfare rispetto a quella ottenuta senza vincoli.

### 4.2.1 Analisi di ammissibilità della soluzione ottima non vincolata

La traiettoria ottima non vincolata è ammissibile se e solo se valgono i seguenti vincoli lineari:

- $C_{i+1,inf} \leq L_i, \quad i = 1 \dots N$
- $C_{i,sup} \geq L_i, \quad i = 1 \dots N$

con  $L_i = \sum_{j=1}^i |A_j|$ .

Se tali vincoli sono soddisfatti allora la traiettoria ottima con i vincoli di copertura è la stessa di quella ottenuta senza vincoli.

Si può inoltre osservare che condizione necessaria affinché ciò avvenga è che:

$$A_i \subseteq C_i$$

**Osservazione 5.** L'ammissibilità della traiettoria ottima ottenuta nel problema non vincolato, può essere verificata completamente offline e questo permette già a priori di

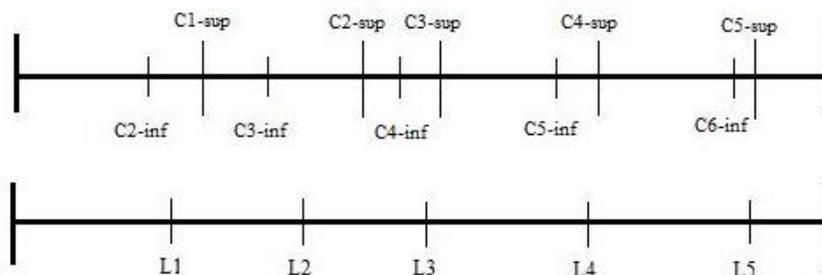


Figura 6: Esempio in cui i vincoli di copertura sono soddisfatti

sapere se è possibile o meno imporre tale traiettoria e, qualora fosse possibile, modificare in modo appropriato i vincoli di copertura affinché risulti ammissibile.

#### 4.2.2 Calcolo della traiettoria ottima vincolata

Come analizzato nella sezione precedente è possibile che la soluzione ottima del problema non vincolato non sia a posteriori compatibile con i vincoli di copertura delle telecamere: in tal caso si pone il problema di stabilire quale sia la copertura a regime, compatibile con i vincoli, ottima rispetto al criterio proposto di minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$ .

Il procedimento adottato è il seguente: a partire dalla traiettoria ottima ottenuta senza tener conto dei vincoli di copertura, si cercherà di adattarla a tali vincoli dimostrando che la scelta fatta è la migliore possibile rispetto al problema di ottimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$ . Di seguito si illustrerà come adattare la traiettoria nel caso di vincolo violato per la prima camera: il caso in cui invece la sezione di copertura non fosse ammissibile per una telecamera nel mezzo o per l'ultima è concettualmente analogo; per approfondimenti, si rimanda all'appendice A in cui si dimostra come determinare la traiettoria ottima anche in questi casi.

Per la camera 1 si può verificare solo un caso sfavorevole:

- $C_{1,sup} < L_1$

Il che significa che la telecamera 1 dovrebbe sfiorare il limite di copertura di cui dispone: è necessario quindi che essa percorra il maggior spazio possibile compatibile con il proprio vincolo  $C_{1,sup}$ , al fine di garantire l'ottimo di copertura del residuo del percorso. Sarà quindi poi possibile reimpostare il calcolo della copertura ottima escludendo la telecamera 1 dal calcolo e considerando solo le restanti  $N - 1$ . La formalizzazione di quanto detto si ha con la seguente proposizione:

**Proposizione 3.** *La copertura ottima del percorso nel caso in cui la telecamera 1 non possa coprire quanto calcolato senza tener conto dei vincoli, si ottiene attribuendo*

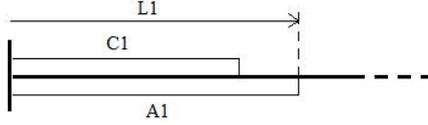


Figura 7: Vincolo violato nella telecamera 1

alla telecamera 1 la massima lunghezza da coprire compatibilmente ai suoi vincoli, e riapplicando l'algoritmo 1 (proposizione 2) alle  $N - 1$  telecamere rimaste sul tratto di percorso  $L \setminus C_1$ . In formule:

1.  $A_1 = C_1$
2.  $|A_i| = V_{i,max} T_{ott.c.v} \quad i \neq 1$

con

$$T_{ott.c.v} = \frac{L_{tot} - |C_1|}{\sum_{i=2}^N V_{i,max}}$$

con  $T_{ott.c.v} > T_{ott}$  (dove per  $T_{ott.c.v}$  si intende il  $T$  ottimo con vincoli.)

*Dimostrazione.* Supponiamo per assurdo che si faccia una scelta diversa di  $A_1$ , con  $A_1 \subset C_1$ ). Allora necessariamente le restanti  $N - 1$  telecamere dovranno coprire una lunghezza  $L_{tot} - |A_1| > L_{tot} - |C_1|$ , ma quindi:

$$\bar{T} = \frac{L_{tot} - |A_1|}{\sum_{i=2}^N V_{i,max}} > T_{ott.c.v} \quad \Rightarrow \quad 2\bar{T} > 2T_{ott.c.v}$$

con  $T_{ott.c.v}$  calcolato come sopra.

Ma il criterio di ottimalità scelto stabilisce di minimizzare il  $T_{max}$  di non visita per i punti del percorso (globalmente), pertanto con la disuguaglianza  $2\bar{T} > 2T_{ott.c.v}$  (il periodo di patrolling delle telecamere da 2 a  $N$  peggiora) si è dimostrato che per ogni scelta di  $A_1 \neq C_1$  si fa peggio.  $\square$

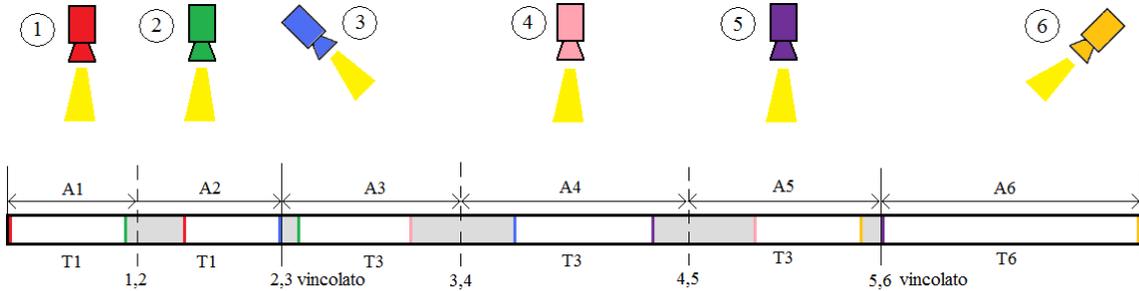


Figura 8: Possibili sezioni ottime (vincolate) in un percorso

### 4.3 Problemi e limiti implementativi

Nella definizione del funzionale di ottimizzazione  $J(\cdot, \cdot, \cdot)$  e nella successiva determinazione della traiettoria ottima (con o senza vincoli) non si è tenuto conto del fatto che l'implementazione della soluzione adottata deve essere distribuita. Come per la determinazione del controllo ottimo per la minimizzazione dell'indice, si è ritenuto utile procedere per passi successivi: innanzitutto determinare la soluzione ottima nel problema centralizzato (generalmente più facile) e poi cercare di adattarla (con al più qualche modifica) alle esigenze di un'implementazione distribuita. In questa sezione si illustreranno le problematiche che si sono dovute affrontare, mentre nella successiva si esporrà la soluzione distribuita implementata dimostrando l'equivalenza con la soluzione ottima determinata in modo centralizzato.

#### 4.3.1 Analisi della soluzione ottima in termini distribuiti

Assumendo che ogni telecamera conosca la lunghezza totale  $L_{tot}$  del percorso da monitorare e che i field of view ( $fov$ ) siano puntiformi (come detto in precedenza, anche per l'analisi distribuita questa ipotesi non complica in particolar modo la questione), il tempo ottimo  $T_{ott}$  di patrolling è (nel caso non vincolato):

$$T_{ott} = \frac{L_{tot}}{\sum_{i=1}^N V_{i,max}}$$

In tale formula si vede chiaramente che l'unico parametro variante da camera a camera che influenza il calcolo del tempo ottimo è la velocità massima delle singole camere. Questa sarà dunque l'informazione che le camere devono trasmettersi a due a due, sfruttando un approccio distribuito. Tra le varie possibilità si è pensato anche di ricorrere ad una tecnica basata sull'average consensus (ciascun agente deve conoscere la somma delle velocità che altro non è che la media moltiplicata per il numero totale di agenti); tuttavia nel problema specifico sfruttando il fatto che il grafo di comunicazione associato agli agenti è fortemente connesso è possibile adottare una soluzione molto semplice (che verrà chiamata *comunicazione passo passo*) che in esattamente  $N$  passi permette a ciascun agente di disporre della somma delle velocità; inoltre la procedura passo passo è necessaria per verificare l'ammissibilità ai vincoli della soluzione ottima.

#### 4.3.2 Grafo di comunicazione del sistema

Supponiamo di associare ad ogni agente (telecamera) un nodo e di stabilire se vi è comunicazione tra due di essi collegandoli con un arco orientato; allora nel problema specifico si può considerare il grafo di comunicazione riportato in figura 9.

Dall'immagine si osserva immediatamente che esso è fortemente connesso, quindi ciascun agente può comunicare con un altro in al più  $N$  passi (l'agente  $i$ -esimo invia a  $i + 1$  la richiesta di comunicare con  $j$ , al passo successivo  $i + 1$  la inoltra a  $i + 2$  e così via fino all'agente  $j$ , da cui il nome "comunicazione passo passo").

Il grafo di comunicazione utilizzato è sensato per il problema in questione: si può infatti supporre che ciascuna telecamera possa comunicare almeno con le due adiacenti,

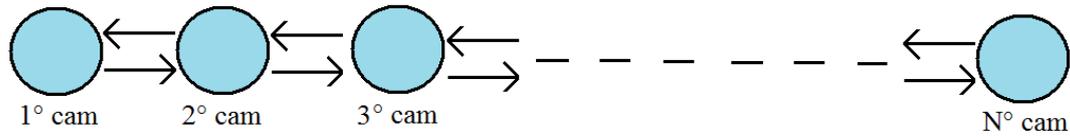


Figura 9: Grafo base di comunicazione del sistema

condizione indispensabile affinché complessivamente il sistema possa convergere verso un qualche tipo di coordinazione; eventuali archi di comunicazione in più sono superflui ai fini della procedura che si esporrà anche se ovviamente permettono un passaggio di informazione più rapido e quindi una maggiore velocità di convergenza dell'algoritmo.

#### 4.3.3 Comunicazione passo-passo: calcolo del periodo ottimo

La comunicazione passo-passo, come già accennato, consiste nello scambiarsi informazioni solo con i vicini, ad ogni istante di tempo. Supponiamo di iniziare tale procedura di comunicazione dalle camere poste ai due estremi del percorso: in tal modo dopo esattamente  $N$  passi ogni camera dispone dell'informazione richiesta riguardante tutte le altre (nello specifico esempio le velocità) e dunque è in grado di calcolare il tempo ottimo di patrolling. Ulteriore e non banale problema sarà quello di calcolare le sezioni ottime ( $A_i$ ) di ogni telecamera, questione affrontata nella prossima sezione.

#### 4.3.4 Analisi della complessità dell'algoritmo passo-passo

Risolto abbastanza facilmente il problema del calcolo distribuito del  $T_{ott}$ , ci si pone il problema di determinare per ogni telecamera la sezione di copertura a regime  $A_i$ , compatibilmente ai vincoli. L'algoritmo proposto è una naturale estensione in termini distribuiti della procedura adottata per determinare la traiettoria ottima vincolata (sezione 4.2.2).

#### Algoritmo 2.

1. Calcolo del  $T_{ott}$  in  $N$  passi utilizzando la comunicazione passo passo;
2. a partire da un estremo (ad esempio dalla telecamera 1) si verifica l'ammissibilità della sezione  $A_i$  rispetto ai vincoli di copertura  $C_i$ :
  - se è soddisfatta si prosegue con l'analisi di ammissibilità della sezione successiva  $A_{i+1}$ ;
  - altrimenti si fissa la copertura  $A_i$  compatibilmente con i vincoli e si rilancia il calcolo delle sezioni ottime nel resto del percorso con l'algoritmo 1 come mostrato nella sezione 4.2.2 comunicandolo alle adiacenti.

3. La procedura è conclusa quando ogni telecamera ha calcolato la propria sezione di copertura compatibilmente ai vincoli del sistema.

Analizzando attentamente quest'algoritmo risulta evidente la sua logica ricorsiva (il percorso ad ogni passo viene partizionato in sottopercorsi) e la sua convergenza verso l'ottimo vincolato: tuttavia sono altrettanto evidenti le difficoltà che si possono ottenere in pratica nell'implementarlo ed i lunghi tempi di convergenza soprattutto nei casi più sfortunati; in particolar modo emerge una notevole differenza tra il caso in cui l'ottimo non vincolato sia ammissibile e quello in cui non lo sia.

**Ottimo non vincolato ammissibile.** In questo caso l'algoritmo proposto procede come segue:

1. ad ogni passo la telecamera  $i$ -esima comunica le velocità che ha in memoria alle telecamere adiacenti (in  $N$  passi tutte le telecamere hanno le velocità delle altre);
2. ciascuna telecamera calcola il  $T_{ott}$  e in ulteriori  $N$  passi (a partire dalla prima) le telecamere conoscono ciascuna la propria sezione di copertura all'ottimo  $A_i$  che verificano essere compatibile con i vincoli di copertura.

Dunque in  $2N$  passi il sistema converge alla soluzione ottima e l'algoritmo proposto risulta quindi avere una complessità accettabile dell'ordine del numero delle telecamere,  $O(N)$ .

**Ottimo non vincolato non ammissibile.** L'analisi di questo caso è fatta nel *worst case*. Supponiamo che il problema di copertura si manifesti solo nell'ultima telecamera: quindi  $A_n$  non è ammissibile. In tal caso l'algoritmo richiede di fissare compatibilmente ai vincoli la copertura  $A_n (= C_n)$  e di rilanciare il calcolo dell'ottimo sul resto del percorso considerando di avere solo  $N - 1$  telecamere a disposizione; ma il problema poi si potrebbe verificare nella camera numero 1 e così via:

1. problema che si verifica nella telecamera  $N$ -esima ( $N$  passi);
2. calcolo dell'ottimo rilanciando l'algoritmo 1 (proposizione 2) sulle prime  $N - 1$  telecamere con problema nella prima ( $N - 1$  passi);
3. fissata la copertura della prima ( $A_1 = C_1$ ) si esegue un nuovo calcolo dell'ottimo sul residuo del percorso per le telecamere da 2 a  $N - 1$  ( $N - 2$  passi);
4. il problema si verifica alla  $N - 1$ -esima, si fissa tale copertura ecc...

Non è difficile quindi intuire come la convergenza alla soluzione ottima vincolata richieda  $N!$  passi e quindi la complessità sia nel caso peggiore:

$$O(N!)$$

il che rende praticamente impossibile adoperare rigorosamente questa tecnica per determinare l'insieme delle sezioni ottime vincolate ( $\{A_i\}$ ): già solo per  $N = 10$  il numero

di passi necessari è nel caso peggiore dell'ordine di  $10^6$ . Va sottolineato come sia molto improbabile incappare in questo caso sfortunato (che comunque è possibile), tuttavia le difficoltà nello stimare la complessità del caso medio (che potrebbe non essere  $O(N!)$  ma nemmeno  $O(N)$ ) e le difficoltà di un'implementazione efficiente, hanno spinto verso la ricerca di una tecnica implementativa alternativa.

## 5 Algoritmo distribuito per determinare la copertura ottima

Nella sezione precedente sono emerse tutte le difficoltà di un'implementazione distribuita basata sulla comunicazione passo passo dell'algoritmo di calcolo delle sezioni nel problema vincolato (che è il caso reale). Il limite di questo approccio consiste nel cercare di realizzare la soluzione ottima calcolata in modo centralizzato sfruttando il fatto che il grafo di comunicazione è fortemente connesso e che quindi ciascuna telecamera può sapere tutto delle altre a patto di aspettare un certo numero di passi; questa logica quindi rallenta notevolmente il calcolo delle sezioni, come è stato dimostrato precedentemente, perchè non è intrinsecamente distribuita.

L'idea della soluzione distribuita implementata ha avuto origine dall'analogia del sistema in questione con uno fisico molto intuitivo e dall'osservazione fondamentale che l'algoritmo di calcolo delle sezioni così come è stato concepito, sia nel caso non vincolato che in quello vincolato, cerca sempre di distribuire la copertura del percorso, compatibilmente ai vincoli, imponendo l'uguaglianza dei periodi di patrolling tra le telecamere adiacenti (l'equilibrio dei periodi  $T$ ).

### 5.1 Paragone con un sistema fisico meccanico

Consideriamo un sistema costituito da un insieme di  $N$  molle (di massa trascurabile) in cui tra gli estremi comuni sono attaccate delle masse  $m$  ed in cui la prima e l'ultima molla sono fissate a telaio:

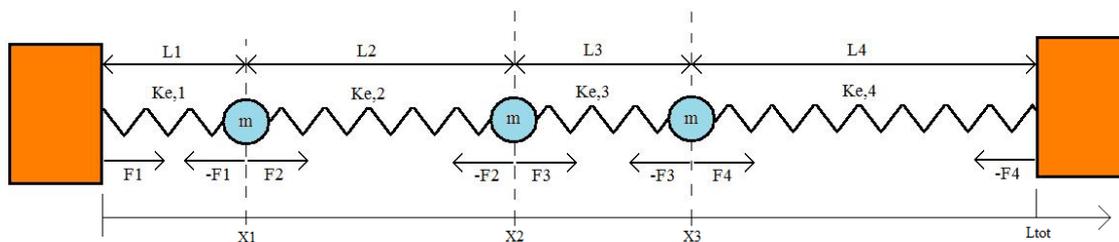


Figura 10: Sistema fisico meccanico in equilibrio, con  $N = 4$

Per il sistema in questione esiste un punto di equilibrio stabile, ottenuto imponendo che la forza risultante applicata ad ognuna delle masse  $m$  sia nulla:

$$|F_1| = |F_2| \quad |F_2| = |F_3| \quad \dots \quad \Rightarrow \quad |F_1| = \dots = |F_N|$$

In altri termini la lunghezza di ciascuna molla  $L_i$  nel punto di equilibrio dovrà essere inversamente proporzionale alla sua costante elastica  $K_{e,i}$ :

$$L_1 K_{e,1} = L_2 K_{e,2} = \dots = L_N K_{e,N}$$

L'equilibrio in questione è senz'altro stabile (è un minimo dell'energia potenziale totale del sistema) e diventa asintoticamente stabile quando nel sistema agisce un attrito viscoso. Queste considerazioni fisicamente molto intuitive possono essere ben formalizzate ad esempio con il *secondo teorema di Lyapunov* prendendo l'energia meccanica  $E_{mecc} = E_K + E_P$  come funzione di Lyapunov: in tal caso evidentemente  $E_{mecc}$  ha un minimo nella configurazione sopra mostrata (velocità nulle ( $E_K = 0$ ) e minimo di energia potenziale) ed in presenza di attrito viscoso vale  $\dot{E}_{mecc} < 0$ , pertanto si conclude immediatamente per l'asintotica stabilità del punto di equilibrio considerato. Ciò significa che a partire da una qualsiasi configurazione ammissibile, il sistema tende spontaneamente verso la soluzione di equilibrio analizzata con una velocità che è funzione delle masse  $m$ , dell'attrito viscoso  $\mu$  e delle costanti elastiche delle molle. Per farsi un'idea di come questi parametri determinano la velocità di convergenza al punto di equilibrio si consideri il sistema semplice *molla-smorzatore*: in questo caso è possibile definire il coefficiente

$$\xi = \frac{\mu}{2\sqrt{K_e m}} \quad (6)$$

che, a seconda dei valori che assume, caratterizza l'evoluzione libera del sistema (un valore appropriato di  $\xi$  è 1, in tal caso il sistema converge velocemente alla soluzione di equilibrio senza sovralongazione). La prontezza del sistema è data invece dal rapporto

$$\omega_0 = \sqrt{\frac{K_e}{m}} \quad (7)$$

Riepiloghiamo brevemente le ipotesi del modello considerato:

- $N$  molle aventi costanti elastiche  $K_{e,i}$ , anche diverse;
- $N - 1$  masse  $m$  attaccate tra una molla e la successiva (le masse  $m$  potrebbero anche essere diverse);
- presenza di attrito viscoso  $\mu$ .

In queste ipotesi il sistema converge spontaneamente verso la soluzione di equilibrio in cui le forze interne applicate alle masse  $m$  sono pari ed opposte: si sottolinea l'uso dell'aggettivo *interne* per sottolineare che l'equilibrio per questo sistema fisico è legato solo ad interazioni locali, il che concettualmente equivale ad affermare che il sistema meccanico converge alla condizione di equilibrio in modo distribuito.

Fatte queste premesse è possibile interpretare il sistema di telecamere in questione mediante quello fisico appena illustrato, adattando le grandezze di quest'ultimo a quelle del primo:

1. la lunghezza di ciascuna molla  $L_i$  corrisponde alla lunghezza della  $A_i$  di ciascuna telecamera;
2. la costante elastica della molla  $i$ -esima ( $K_{e,i}$ ) è pari al reciproco della velocità massima della telecamera corrispondente ( $V_{i,max}$ ):

$$K_{e,i} = \frac{1}{V_{i,max}}$$

3. le masse puntiformi  $m$  attaccate tra una molla e l'altra modellano la dinamica dell'estremo comune tra le telecamere che si adatta nel tempo;
4. il coefficiente di attrito  $\mu$  è fissato dalla scelta del coefficiente di smorzamento  $\xi$  (formula 6) desiderato.

Interpretando in questo modo le grandezze del sistema in esame si dimostra che la soluzione di equilibrio coincide con quella determinata ottimizzando l'indice  $J(\cdot, \cdot, \cdot)$  secondo la minimizzazione proposta.

**Osservazione.** Il sistema in questione è più complesso del sistema semplice molla-smorzatore visto nella sezione precedente; tuttavia come scelta iniziale dei parametri  $\xi$  e  $\omega_0$  ci si è basati sulle formule (6) e (7) impostando il valore di  $\xi$  pari a 1 e calcolando il valore di  $\mu$  corrispondente, utilizzando successivamente le simulazioni per una taratura più corretta di tale parametro.

## 5.2 Equivalenza della soluzione di equilibrio con il calcolo delle sezioni ottime

Adattando come proposto le grandezze del problema, la soluzione di equilibrio per il sistema fisico si traduce nel sistema in esame come segue:

$$F_i = F_j \quad \Rightarrow \quad \frac{1}{V_{i,max}} |A_i| = \frac{1}{V_{j,max}} |A_j| \quad \Rightarrow \quad T_i = T_j \quad , \quad \forall i, j$$

Ciò dimostra perfettamente l'equivalenza tra la soluzione di equilibrio e le sezioni calcolate all'ottimo con il metodo proposto: l'equilibrio a cui si giunge, infatti, consiste nell'imporre periodi  $T_i$  di patrolling uguali tra le telecamere che è proprio quanto si impone con l'algoritmo proposto. In assenza di vincoli (o equivalentemente se la soluzione ottima è ammissibile) si giunge pertanto ad una situazione a regime in cui ciascuna telecamera monitora la propria sezione  $A_i$ , a lei preposta, con periodo  $T$  pari a:

$$T = 2T_{ott} = \frac{2L_{tot}}{\sum_{i=1}^N V_{i,max}}$$

Anche per il caso vincolato si dimostra immediatamente l'equivalenza dei due approcci: in tal caso la procedura richiede di fissare i limiti delle  $A_i$  compatibilmente ai vincoli di copertura fisica imposti dalle  $C_i$ , per poi rilanciare l'algoritmo 1 sulle restanti telecamere; la soluzione proposta si comporta esattamente allo stesso modo: fissa gli estremi di  $A_i$  compatibilmente al vincolo e poi automaticamente le altre telecamere si assestano in modo tale da coprire la restante parte del percorso in maniera ottima con periodi  $T_j$  possibilmente uguali (sempre compatibilmente ai vincoli del sistema, figura 8).

**Osservazione.** Come si approfondirà nella sezione relativa agli aspetti implementativi, la soluzione proposta è davvero molto semplice da un punto di vista realizzativo e probabilmente rappresenta il miglior approccio distribuito possibile che permette di convergere al calcolo delle sezioni ottime secondo il criterio di minimizzazione dell'indice  $J(\cdot, \cdot, \cdot)$  scelto: a ciascuna camera è semplicemente richiesto di conoscere il limite inferiore e quello superiore di copertura effettiva e di esercitare una forza proporzionale al periodo di patrolling necessario per coprire tale sezione in modo tale da ridurla se il periodo è troppo grande o viceversa aumentarla (compatibilmente ai vincoli).

### 5.3 Analisi in spazio di stato

Può essere interessante effettuare un'analisi in spazio di stato del sistema naturale per capirne a fondo il funzionamento ed avere una stima della velocità di convergenza al punto di equilibrio analizzando gli autovalori della matrice  $F$ . Le equazioni che regolano la dinamica delle masse  $m$ , nel caso di un sistema con 4 molle e 3 masse (figura 10), sono:

$$\begin{cases} ma_1 = -K_{e,1}\Delta l_1 + K_{e,2}\Delta l_2 - \mu v_1 \\ ma_2 = -K_{e,2}\Delta l_2 + K_{e,3}\Delta l_3 - \mu v_2 \\ ma_3 = -K_{e,3}\Delta l_3 + K_{e,4}\Delta l_4 - \mu v_3 \end{cases} \Rightarrow \begin{cases} \dot{x}_1 = v_1 \\ \dot{v}_1 = -\frac{K_{e,1}}{m}x_1 + \frac{K_{e,2}}{m}(x_2 - x_1) - \mu v_1 \\ \dot{x}_2 = v_2 \\ \dot{v}_2 = -\frac{K_{e,2}}{m}(x_2 - x_1) + \frac{K_{e,3}}{m}(x_3 - x_2) - \mu v_2 \\ \dot{x}_3 = v_3 \\ \dot{v}_3 = -\frac{K_{e,3}}{m}(x_3 - x_2) + \frac{K_{e,4}}{m}(L - x_3) - \mu v_3 \end{cases}$$

Definendo lo stato del sistema come  $X = [x_1 \ v_1 \ x_2 \ v_2 \ x_3 \ v_3]'$  si ottiene la seguente matrice  $F$ :

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K_{e,1}}{m} - \frac{K_{e,2}}{m} & -\mu & \frac{K_{e,2}}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{K_{e,2}}{m} & 0 & -\frac{K_{e,2}}{m} - \frac{K_{e,3}}{m} & -\mu & \frac{K_{e,3}}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{K_{e,3}}{m} & 0 & -\frac{K_{e,3}}{m} - \frac{K_{e,4}}{m} & -\mu \end{bmatrix}$$

Calcolando gli autovalori di  $F$  è possibile avere un'ottima stima della velocità di convergenza alla soluzione di equilibrio, data dall'autovalore dominante (ovviamente è possibile trasportare l'analisi effettuata al sistema in esame valutando quindi la velocità di convergenza del sistema nel calcolo delle sezioni ottime  $A_i$  confrontando tale risultato con quello delle simulazioni).

## 5.4 Vantaggi della soluzione distribuita adottata

I vantaggi del metodo proposto rispetto all'implementazione basata su una comunicazione passo passo, come già accennato, sono molteplici e si provvede ora ad una loro elencazione:

- approccio intrinsecamente distribuito (ogni agente conosce solo i suoi parametri ed interagisce solo in modo locale con gli agenti adiacenti);
- maggiore semplicità di implementazione;
- convergenza all'ottimo con continuità di adattamento delle sezioni;
- maggiore velocità di convergenza alla soluzione ottima fissati in modo adeguato i parametri  $m$  e  $\mu$  (e possibilità di stimarla calcolando l'autovalore dominante della matrice  $F$ );
- maggiore flessibilità dell'algoritmo in caso di rottura di una telecamera (robustezza) o di inseguimento di un evento (tracking).

### 5.4.1 Robustezza e flessibilità

Tra i diversi vantaggi della soluzione adottata, l'analisi della robustezza in caso di rottura di una telecamera e l'analisi della flessibilità nel caso in cui una sia impegnata nel tracking meritano un'attenzione particolare data la loro importanza in un sistema di controllo come quello in esame.

L'algoritmo proposto nella sua implementazione distribuita è estremamente robusto: infatti in caso di rottura di una telecamera  $i$  automaticamente le due adiacenti fissano rispettivamente a  $C_{i-1,sup}$  e  $C_{i+1,inf}$  i propri estremi di pattugliamento mentre gli altri due estremi si assestano dinamicamente in una nuova soluzione di equilibrio. Anche nel caso di tracking dell'evento da parte della telecamera  $i$ , le due adiacenti aumentano al massimo, compatibilmente ai loro vincoli fisici, l'estremo di copertura condiviso con l'agente  $i$ -esimo per ottimizzare in ogni caso il patrolling del resto del percorso (mentre gli altri due estremi dinamicamente si riasserteranno) ed al contempo prepararsi a subentrare nel tracking qualora l'evento entrasse nella loro sezione di competenza, data la priorità del tracking (per maggior chiarezza su tale argomento si veda la sezione 7 dedicata appunto al problema del tracking).

**Osservazione.** In caso di rottura della telecamera  $i$  sarebbe anche possibile riassegnare i limiti fisici di copertura delle due adiacenti ( $C_{i-1,sup}$  e  $C_{i+1,inf}$ ) per garantire comunque la copertura dell'intero percorso: in tal caso gli agenti  $i-1$  e  $i+1$  condividono il nuovo estremo ed il sistema si comporta normalmente adattandosi dinamicamente al fatto che la copertura totale ora è a carico delle  $N-1$  telecamere rimaste.

## 6 Implementazione: struttura base e patrolling

Come si è visto, le caratteristiche peculiari del sistema di camere che ci si propone di implementare sono:

- capacità di calcolo distribuita e non centralizzata;
- comunicazione sporadica e solo lungo i rami del grafo di comunicazione.

Tali proprietà possono essere efficacemente messe in risalto con uno schema a blocchi dove ogni blocco rappresenta la singola telecamera. La struttura interna di tali blocchi si ripete in maniera modulare, nel senso che non vi sono differenze essenziali tra le telecamere, se non quelle legate ai loro parametri caratteristici quali velocità massima di patrolling, range fisico, field of view. I blocchi delle telecamere sono poi connessi da dei bus di comunicazione, organizzati in modo conforme al grafo di comunicazione.

Una struttura di questo tipo può essere resa in modo molto chiaro e suggestivo con uno strumento quale *Mathworks Simulink*, compreso nella suite *Mathworks MATLAB R2009a*. Si è abbandonata un'implementazione iniziale basata puramente su codice in quanto, via via che con l'evoluzione del progetto le dinamiche interne e di interazione tra le camere diventavano più complesse, le difficoltà di stesura del codice aumentavano in modo ingovernabile. Lo strumento Simulink consente invece di mantenere una struttura chiara e modulare, facilmente interpretabile già a prima vista da chi non l'ha realizzata e quindi particolarmente adatta al lavoro di gruppo. Si possono facilmente implementare blocchi dinamici quali funzioni di trasferimento; le variabili di stato sono chiaramente localizzabili in quanto rappresentate dai blocchi di memoria quali ritardi ed integratori. Infine, è sempre possibile implementare segmenti di codice attraverso appositi blocchi che applicano all'ingresso uno specifico algoritmo e restituiscono il risultato alle uscite.

Il sistema è stato realizzato interamente a tempo discreto (simulazione *fixed-step discrete*) in quanto chiaramente i sistemi di calcolo presenti nelle camere lavorano a tempo discreto alla frequenza di clock del loro microprocessore. In appendice E è riportata una vista globale degli schemi; qui di seguito si analizzeranno invece i vari sottosistemi, per descriverne la funzione e le strategie implementative che essi realizzano.

### 6.1 Comunicazione tra le camere

Le camere adiacenti sono interconnesse da un bus di comunicazione che collega gli ingressi *com\_su\_in* (comunicazione dalla camera successiva) e *com\_pr\_in* (comunicazione dalla camera precedente) alle uscite *com\_pr\_out* e *com\_su\_out*. Ogni camera dunque è connessa alle due camere ad essa adiacenti. La comunicazione tra ogni coppia di camere avviene inoltre in modo sporadico: è definita una probabilità  $p_{com}$  (*p\_com* nel file *Dat1.m*, appendice D) che la comunicazione possa avvenire nell'istante corrente della simulazione. Il consenso alla comunicazione tra la coppia di camere viene dato dal segnale *tcom*, che si trova al livello logico alto quando la comunicazione può avvenire. Quando essa avviene, è supposta essere bidirezionale. Nel caso della simulazione con

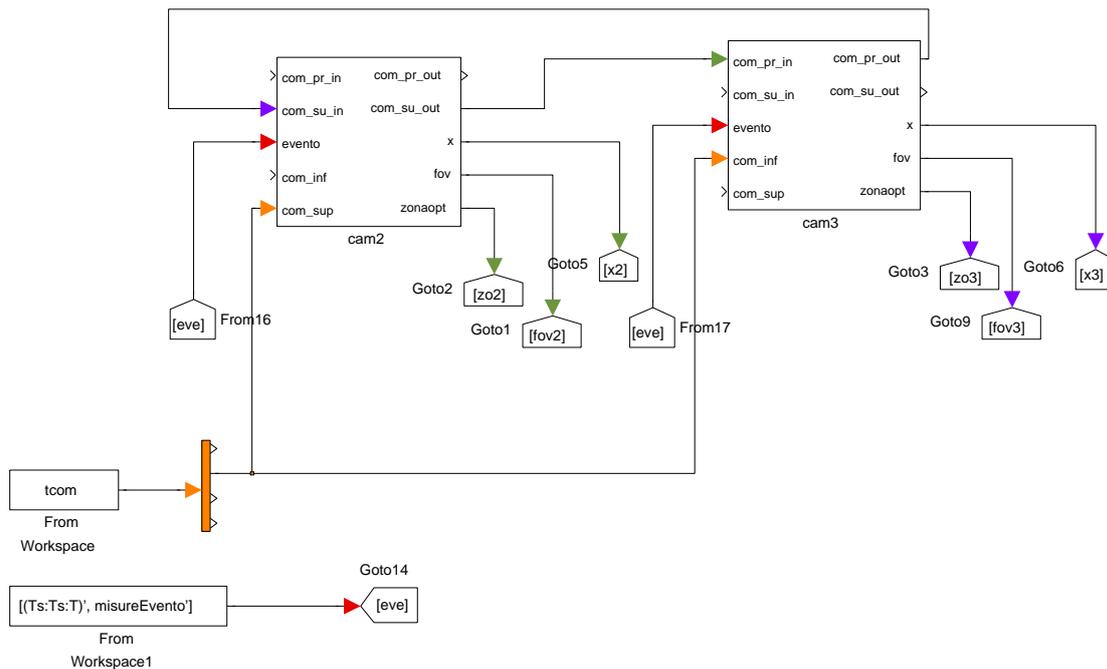


Figura 11: Schema di comunicazione tra due camere  
 I blocchi *Goto* servono a raccogliere l'output della simulazione e possono essere ignorati.

5 camere, *tcom* è un segnale vettoriale con 4 elementi, per consentire in modo asincrono la comunicazione tra le 4 adiacenze. Quando la comunicazione non avviene, viene mantenuto il valore precedente dei segnali ricevuti. Ciò viene modellato dalla presenza di blocchetti di Sample and Hold nei segnali di ingresso alla camera (*SeH Modificato*, figura 12). Non si è utilizzato il blocchetto di Sample and Hold presente di default in Simulink in quanto questo non consentiva la comunicazione in due istanti consecutivi (esso effettuava il sampling solo sui fronti di salita del segnale di comunicazione), mentre la comunicazione in due istanti consecutivi nel modello implementato non è a priori esclusa.

Nella figura 12 si può vedere il dettaglio dei segnali che viaggiano sul bus di comunicazione, che vengono qui elencati:

1. forza che la camera esercita sugli estremi di patrolling;
2. sincronizzazione della posizione dell'estremo comune tra due camere adiacenti;
3. non connesso. Si tratta di una linea di comunicazione rimasta inutilizzata a seguito di alcune semplificazioni sulla struttura implementativa e concettualmente può essere rimossa;
4. segnale di messa in fase delle camere durante il patrolling;
5. tempo di arrivo dell'evento alla camera adiacente.

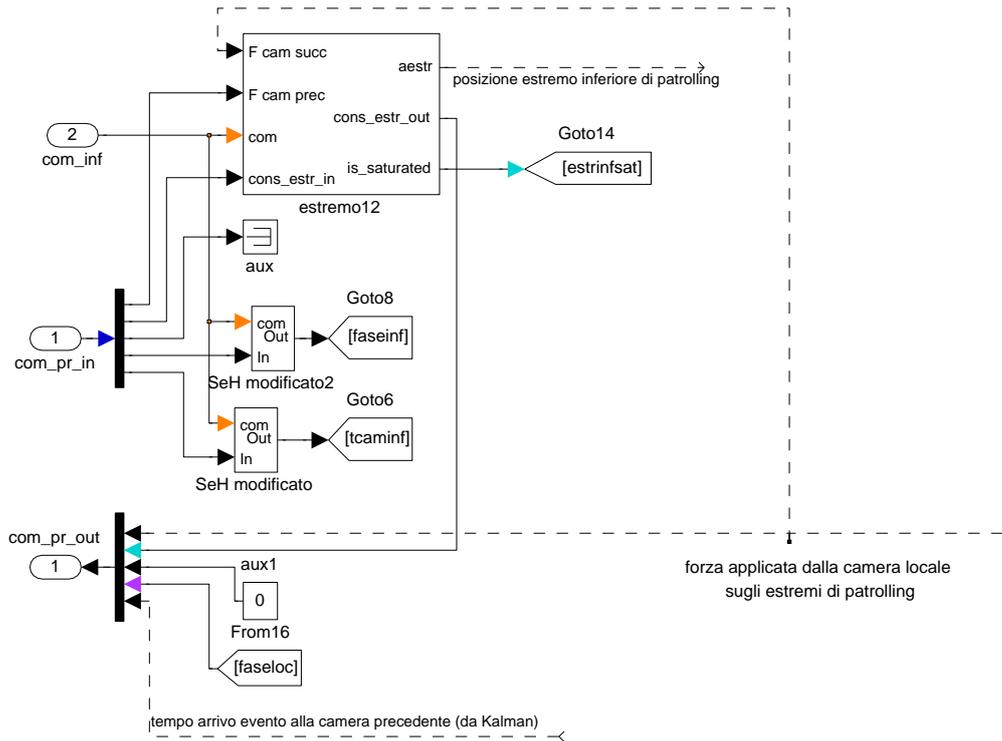


Figura 12: Interno di una camera: componenti del bus di comunicazione, e blocco dell'estremo di patrolling

## 6.2 Forza sugli estremi di patrolling

La forza che la camera esercita sugli estremi di patrolling (cfr.sez. 5) è essenzialmente proporzionale alla distanza tra gli estremi secondo la costante elastica  $K_{e,i}$  (figura 13). Il blocco `adatt_forza` aumenta la forza sugli estremi di patrolling quando la camera sta facendo il tracking in modo da restringerli il più possibile, si veda la sezione 7 per comprendere in quali occasioni il segnale `do_track`, che introduce tale bias sulla forza, venga attivato. La forza applicata sugli estremi di patrolling viene resa nota alle camere adiacenti per mezzo della linea 1 di comunicazione.

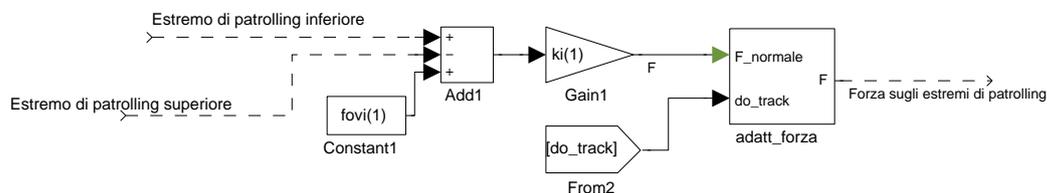


Figura 13: Forza sugli estremi di patrolling

### 6.3 Modello degli estremi di patrolling

Come si è visto nell'analisi teorica, si desidera modellare l'estremo di patrolling come una massa con attrito, alla quale viene applicata la risultante delle forze elastiche delle due camere. Sorge il problema di decidere quale camera, tra due adiacenti, debba calcolare la dinamica dell'estremo e trasmettere la posizione dello stesso alla vicina. Si è deciso di adottare una strategia che presenta una certa ridondanza ma che offre numerosi vantaggi: ogni camera elabora un modello fisico di entrambi i suoi estremi, superiore ed inferiore, e si basa sulla posizione degli estremi fornita da questi due modelli locali per decidere quale sia il suo range di patrolling e quale forza debba applicare sugli stessi estremi. Di conseguenza, esistono due modelli dello stesso estremo di patrolling. Ad esempio l'*estremo23* comune tra camera 2 e camera 3 è modellato come estremo superiore nella camera 2 e come estremo inferiore nella camera 3. Si capisce come sia fondamentale che le due camere mantengano il consenso sulla posizione dell'estremo. Questo è ottenuto in due modi:

1. se le due versioni dell'estremo differiscono, al primo istante in cui si ha la comunicazione tra le due camere adiacenti la posizione dello stesso estremo viene resettata alla media delle due versioni esistenti. Questo avviene sfruttando la linea 2 di comunicazione;
2. quando non c'è comunicazione, le camere applicano sull'estremo la differenza tra forza elastica locale e forza elastica della camera adiacente calcolata all'ultimo istante di comunicazione.

L'accorgimento 2 assicura che, una volta che le camere giungano al consenso sull'estremo sfruttando la strategia fornita dall'accorgimento 1, esso venga poi sempre mantenuto se, come ci si aspetta, le due versioni degli estremi hanno le stesse caratteristiche dei blocchi dinamici. Infatti, su entrambi i modelli viene applicata la stessa forza, che varia negli istanti di comunicazione e che altrimenti viene tenuta costante. Il vantaggio di questo accorgimento è evidente: la dinamica dell'estremo può essere fatta evolvere alla massima frequenza di campionamento, che è quella legata al clock del  $\mu P$  della camera. In questo modo, la camera ha a disposizione un valore di posizione per l'estremo di patrolling che varia in modo "dolce" anche durante i periodi in cui non c'è comunicazione, portandosi verso il nuovo valore di equilibrio dato dalla forza presente all'istante precedente di comunicazione con la dinamica di un sistema con massa e attrito. Il valore della forza applicata viene aggiornato sporadicamente e la simulazione fornirà un'euristica di quale sia il bound inferiore sulla frequenza di comunicazione che serve a mantenere una dinamica stabile degli estremi. Si vedrà che con sole 2-3 comunicazioni nel tempo di salita dell'estremo si ha una dinamica più che soddisfacente. Eventualmente, se la comunicazione dovesse essere molto sporadica, è possibile aumentare la massa degli estremi, in modo da rallentarli. Questo è stato fatto modificando il valore di massa degli estremi in modo coerente con la probabilità di comunicazione definita nel file `Dati.m`, cercando la massa minima degli estremi che consenta una dinamica stabile con la probabilità di comunicazione imposta. Una possibile linea di

sviluppo di questo approccio è un algoritmo che regoli in modo adattativo la massa degli estremi di patrolling a seconda della frequenza di comunicazione disponibile. La massa può essere variata liberamente, infatti il suo valore influenza solo la velocità di convergenza all'equilibrio ma non modifica la configurazione finale di equilibrio, che rimane fissata dalle equazioni proposte in sezione 5. Un approccio adattativo è particolarmente adatto a strutture con comunicazione wireless, dove la probabilità che la comunicazione possa andare a buon fine può dipendere dalle condizioni ambientali e variare nel tempo.

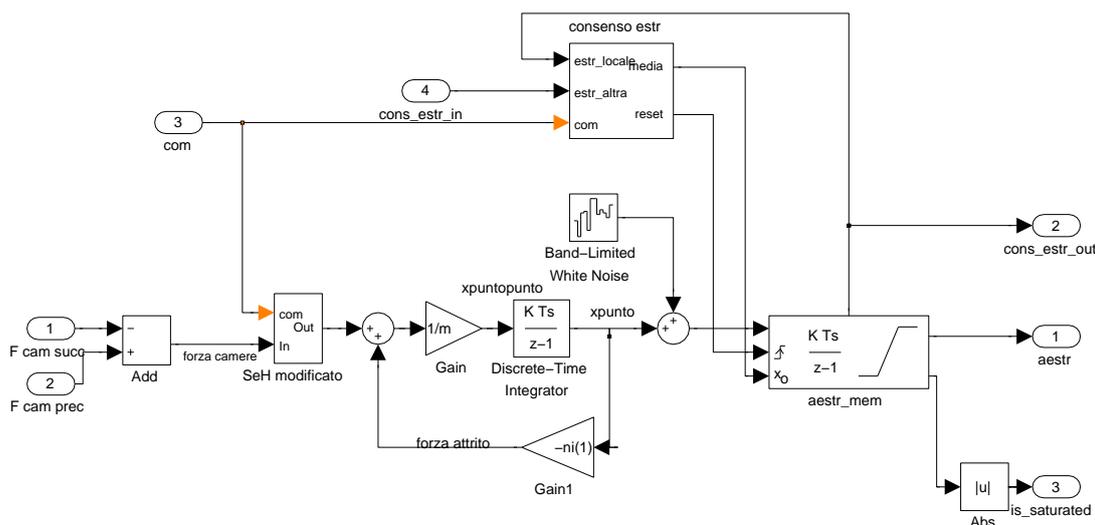


Figura 14: Modello dell'estremo di patrolling

In figura 14 si riporta il dettaglio del modello degli estremi di patrolling implementato all'interno delle camere. I blocchi essenziali sono gli integratori che permettono di ottenere velocità e posizione degli estremi. La posizione è soggetta ad una saturazione in maniera tale da far variare l'estremo di patrolling solo all'interno della zona dove c'è sovrapposizione tra i range fisici delle due camere adiacenti. Il blocco di rumore che va ad agire sulla linea della velocità è normalmente disattivato e permette di apprezzare in che modo le due versioni degli estremi si riportino al consenso qualora i valori assunti dovessero andare a differenziarsi per qualche motivo (ad es. diversa quantizzazione nella struttura di calcolo delle due camere adiacenti, si veda la simulazione di figura 37, appendice B). Per semplicità si è deciso di resettare al consenso solo l'integratore di posizione, mentre a rigore sarebbe necessario resettare l'intero stato dell'estremo, ovvero anche l'integratore di velocità. Tuttavia grazie alla presenza di un termine di smorzamento, si può vedere che dopo alcuni istanti di comunicazione si raggiunge comunque il consenso entro una tolleranza prefissata effettuando il reset del solo integratore di posizione. La posizione dell'estremo di patrolling è fornita dal segnale *aestr*.

## 6.4 Movimento della telecamera e potenziale degli estremi

Nell'approccio iniziale con il quale si era strutturato il patrolling (cfr. appendice C), la camera disponeva di un potenziale di posizione per ogni punto del suo range fisico, che cresceva in maniera monotona al crescere del tempo di non visita di tale punto del percorso. Tale potenziale veniva modificato in modo da forzare la messa in fase tra le camere, e la camera andava in direzione del massimo del funzionale risultante. Questo comportava una notevole complessità di calcolo e determinava un comportamento complessivo del sistema difficile da interpretare e da analizzare teoricamente e completamente inaffidabile in caso di comunicazione sporadica. Con il nuovo approccio invece il range ottimo di patrolling è noto alla camera in ogni istante, ed essa non dovrà fare altro che muoversi alla massima velocità tra i due estremi di patrolling. Dall'analisi teorica, la messa in fase delle camere è risultata essere ininfluenza per quanto riguarda la minimizzazione del tempo di non visita ai punti del percorso, mentre un comportamento in opposizione di fase consente di assicurare un lower bound sulla probabilità di intercettare un evento in movimento (cfr. sez. 3). La messa in fase verrà dunque indotta con un meccanismo a priorità inferiore ad ogni altra azione della telecamera e indipendente dai potenziali di posizione legati al patrolling. Tale meccanismo verrà analizzato nella sezione successiva. Ciò che qui conta è la perfetta equivalenza in una situazione di regime tra memorizzare un potenziale di non visita per ogni punto del percorso e dirigere la camera verso il punto dove il potenziale è massimo, rispetto a calcolare il valore del potenziale di non visita solo agli estremi di patrolling per decidere verso quale estremo dirigere la camera (cfr. proposizione 1, sez. 4). Il sistema qui proposto, e rappresentato in figura 15, adotta proprio questa strategia. Due integratori fanno crescere linearmente nel tempo il valore del potenziale associato ai due estremi e vengono resettati quando il field of view della camera raggiunge l'estremo. La camera si dirige verso l'estremo dove il potenziale è maggiore, alla velocità massima. La limitazione di velocità è imposta dal blocco *sat\_velocità\_dinamica*, che come si vedrà nella sezione successiva consente di ottenere la messa in fase tra le camere, e quando la messa in fase è raggiunta non fa altro che limitare la velocità della camera a  $V_{i,max}$ . Vi sono due segnali prioritari che possono interrompere il normale comportamento di patrolling e sono legati alla necessità di fare il tracking di eventi. Il primo (*priorità\_catch*) impone alla camera di dirigersi verso il bordo con un evento in arrivo, e si attiva quando il tempo di arrivo dell'evento è inferiore al tempo che la camera impiega a raggiungere il bordo; il secondo fa dirigere la camera verso la stima di posizione dell'evento fornita dal filtro di Kalman qualora l'evento si trovi nel suo field of view.

## 6.5 Messa in fase delle camere

La sincronizzazione della fase delle telecamere è una caratteristica che si è deciso di aggiungere verso la fine del progetto, pensando che potesse essere una ragionevole richiesta di un ipotetico committente del sistema. Ciò che si mira a realizzare è in realtà la sincronizzazione in opposizione di fase delle telecamere, in quanto consente di ottenere un bound inferiore sulla probabilità di catturare un evento in movimento

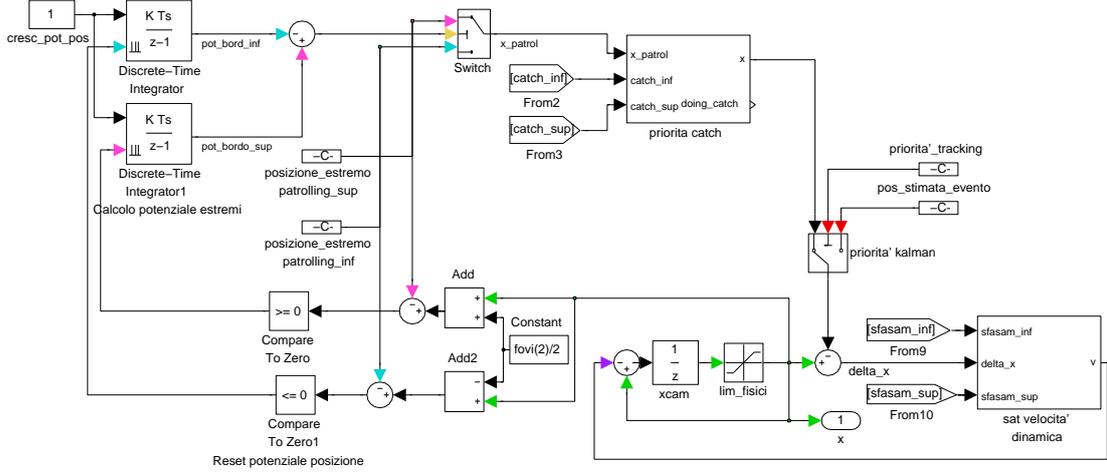


Figura 15: Dinamica di patrolling della camera

(sez. 3) nella zona dove le camere hanno lo stesso  $T_{ott}$  di patrolling. Parleremo nel seguito di messa in fase delle camere, essendo del tutto banale la modifica per indurre invece l'opposizione di fase. Si è deciso di realizzarla come un sistema a priorità minima, ovvero bypassato da tutte le altre dinamiche prioritarie legate alla comparsa dell'evento, e disattivabile a piacere dall'utilizzatore.

Per ottenere la messa in fase delle camere con una struttura indipendente dai potenziali associati agli estremi e facilmente integrabile con lo schema di patrolling già implementato, si è deciso di andare ad agire sulla velocità della camera, anziché intervenire sul potenziale legato agli estremi. Assunto che le telecamere abbiano un limite superiore alla loro velocità, pari a  $V_{i,max}$ , l'unico modo in cui si possa ottenere la messa in fase è quello di far rallentare alcune delle camere perché si sincronizzino con le altre. Chiaramente dunque si perderà qualcosa in termini di tempo massimo di non visita ai punti del percorso, almeno in un transitorio iniziale. A nostro avviso però la possibilità di assicurare l'aggancio di un evento in movimento è vincente in questo trade-off.

La camera  $i$ -esima dunque calcola un valore di fase  $\phi_i$  di questo tipo:

$$\phi_i(t) = 0.5 \frac{x_i(t) - A_{i,inf}}{A_{i,sup} - A_{i,inf}} \quad \text{se } J(A_{i,inf}, t) < J(A_{i,sup}, t),$$

$$\phi_i(t) = 1 - 0.5 \frac{x_i(t) - A_{i,inf}}{A_{i,sup} - A_{i,inf}} \quad \text{altrimenti}$$

dove con un piccolo abuso si è indicato con  $J(\cdot, \cdot)$  il potenziale associato agli estremi, e chiaramente  $A_i = [A_{i,inf}, A_{i,sup}]$  è un intervallo i cui estremi variano per assestarsi alla situazione di regime nel modo precedentemente descritto. La fase dunque è definita in modo da discriminare il fatto che la camera stia andando in una direzione o nell'altra, e cresce monotonamente nel tempo modulo l'unità. Per analogia, si pensi alla fase di un

vettore complesso la cui punta ruota monotonamente lungo una circonferenza di centro l'origine: essa cresce monotonamente modulo  $2\pi$ , ed il moto di un sistema periodico può essere una funzione armonica che è la proiezione di tale vettore sull'asse reale. Allo stesso modo intendiamo qui la fase, con la sola differenza che il moto della camera non è descritto da legge armonica.

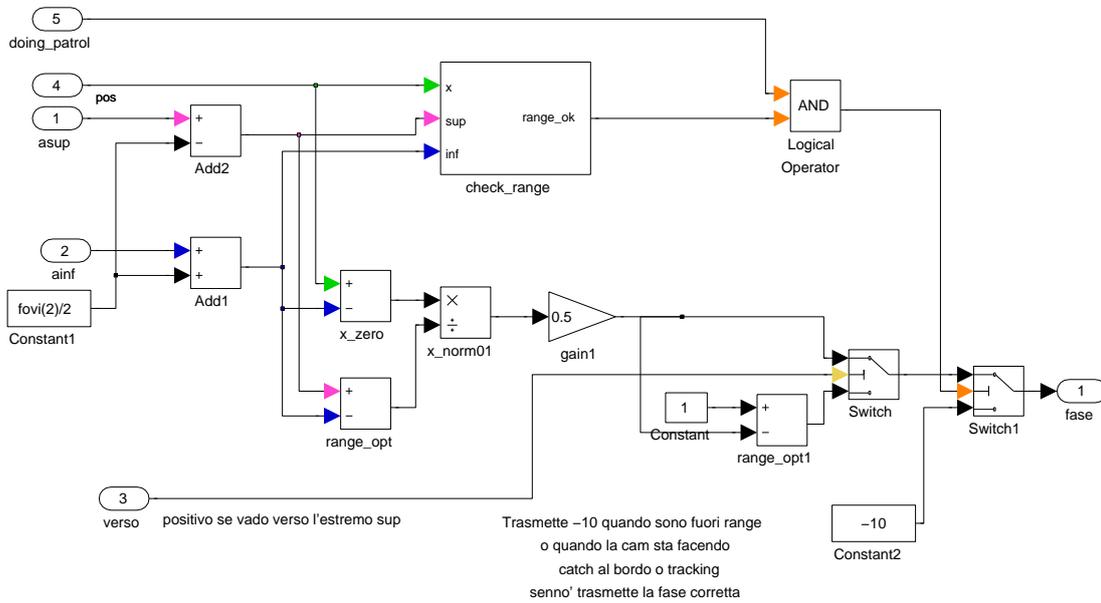


Figura 16: Calcolo della fase della camera

Le camere adiacenti si scambiano l'informazione sulla fase. La camera  $i$  si trova in *anticipo di fase* rispetto alla camera  $j = i - 1$  o  $j = i + 1$  quando

$$\begin{aligned} \phi_i > \phi_j \quad & \text{e} \quad |\phi_i - \phi_j| < 0.5 \\ \text{oppure} \\ \phi_i < \phi_j \quad & \text{e} \quad |\phi_i - \phi_j| > 0.5 \end{aligned}$$

Nei casi sopra citati, individuabili con un'operazione logica di *or esclusivo*, la camera  $i$  deve rallentare per sincronizzarsi con la  $j$ , e lo farà di un fattore proporzionale allo sfasamento; negli altri casi invece continuerà ad andare alla  $V_{i,max}$  in quanto è la camera vicina che deve rallentare per sincronizzarsi. La scelta di ottenere un rallentamento proporzionale allo sfasamento e con un fattore di proporzionalità abbastanza piccolo rispetto all'unità assicura che la messa in fase non penalizza eccessivamente il patrolling e conduce, dopo un certo numero di oscillazioni periodiche, alla corretta messa in fase anche in condizioni di comunicazione sporadica, sempreché ci siano almeno due comunicazioni nel corso di un periodo di patrolling.

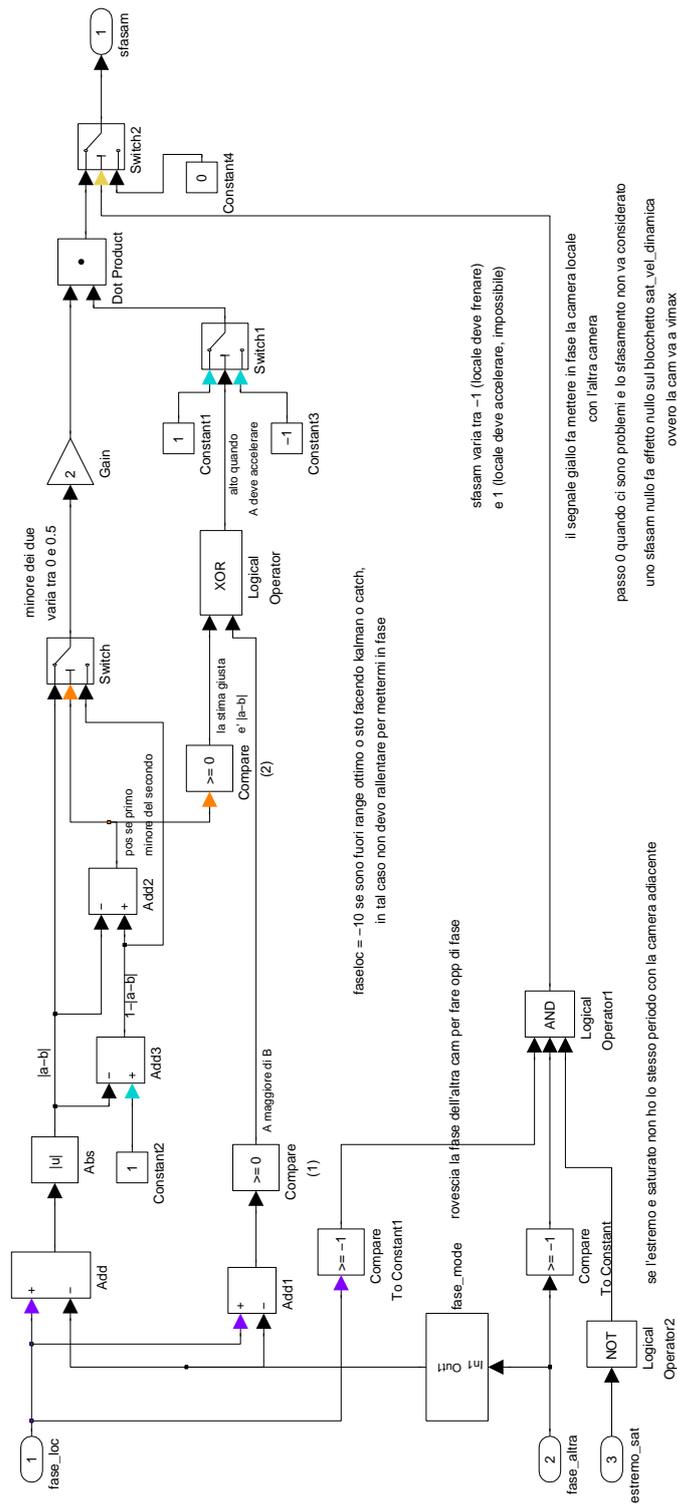


Figura 17: Calcolo dello sfasamento rispetto alla camera adiacente

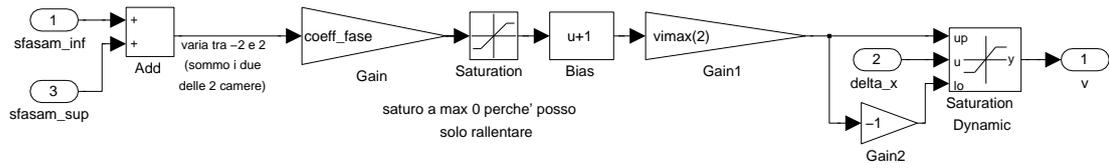


Figura 18: Saturazione dinamica della velocità

Si riportano gli schemi di calcolo della fase (figura 16), di calcolo dello sfasamento (figura 17) e di riduzione della velocità massima della camera (figura 18) che implementano quanto descritto. Una nota importante: quando l'estremo tra due camere si trova al valore di saturazione imposto dagli estremi fisici del range della camera, chiaramente le due camere non hanno lo stesso periodo di patrolling. In questo caso, la messa in fase viene disattivata per mezzo del segnale *estremo\_sat*.

## 7 Tracking di eventi: modellizzazione ed implementazione

### 7.1 Dinamica dell'evento

Per effettuare il tracking di un evento, è necessario prima modellizzare l'evento in questione. Vista la volontà di utilizzare un filtro di Kalman, la modellizzazione richiesta è ovviamente quella in spazio di stato. Come è noto, vengono assunti come stato di un oggetto in movimento la sua posizione e la sua velocità. La posizione futura dell'evento dipende dalla posizione e dalla velocità attuali:

$$x_e(t+1) = x_e(t) + v_e(t) \quad (8)$$

La velocità invece è stata modellata secondo la seguente equazione:

$$v_e(t+1) = av_e(t) + bn_1(t) \quad (9)$$

Il parametro  $a$ , a seconda che sia maggiore o minore di 1, fa sì che l'evento acceleri o freni, il rumore  $n_1(t)$ , bianco e gaussiano, serve per ottenere un andamento più o meno casuale della velocità, consentendo ad essa di cambiare anche di segno, facendo sì che l'evento si diriga verso un estremo o l'altro del perimetro in cui esso si muove. Il modello in spazio di stato sarà dunque

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & a \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ b & 0 \end{bmatrix} \begin{bmatrix} n_1(t) \\ n_2(t) \end{bmatrix} \\ y(t) &= [1 \ 0] x(t) + [0 \ d] \begin{bmatrix} n_1(t) \\ n_2(t) \end{bmatrix} \end{aligned} \quad (10)$$

L'uscita considera unicamente la posizione perchè le telecamere acquisiscono in realtà immagini in successione; esse dunque non sono in grado di misurare la velocità dell'evento, ma si limiteranno a stimarla. In altre parole, le telecamere sono sensori solamente di posizione. Il rumore  $n_2(t)$  è rumore bianco e gaussiano di misura. I parametri  $b$  e  $d$  saranno scelti in maniera opportuna a seconda di quali aspetti si vorranno far risaltare dalle simulazioni.

## 7.2 Filtro di Kalman e coordinazione tra le camere

Modellato l'evento, si può passare al filtro di Kalman. Come è noto dalla teoria, esso si ottiene con le seguenti equazioni:

$$\begin{aligned}
 \left. \begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k} \\ P_{k+1|k} &= AP_{k|k}A^T + Q \end{aligned} \right\} \text{predizione} \\
 \\
 \left. \begin{aligned} \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) \\ P_{k+1|k+1} &= P_{k+1|k} - P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1}CP_{k+1|k} \end{aligned} \right\} \text{aggiornamento} \\
 \\
 K_{k+1} &\triangleq P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1} \} \text{guadagno di Kalman} \\
 \\
 \left. \begin{aligned} \hat{x}_{0|-1} &= \bar{x}_0 \\ P_{0|-1} &= P_0 \end{aligned} \right\} \text{inizializzazione}
 \end{aligned} \tag{11}$$

La matrice  $P_{0|-1}$  di varianza iniziale è stata creata con grande incertezza sulla velocità ed incertezza non eccessiva sulla posizione, dal momento che le telecamere, la prima volta che si accorgono della presenza di un evento, non hanno alcuna informazione sulla sua velocità, mentre la posizione è per forza contenuta all'interno del field of view e dunque non ha senso assegnarle varianza iniziale troppo elevata:

$$P_{0|-1} = \begin{bmatrix} 10 & 0 \\ 0 & 1000 \end{bmatrix}$$

La condizione iniziale  $\bar{x}_0$  è invece la prima misura dell'evento che la telecamera effettua, ovvero la posizione nel primo frame in cui esso appare. Si osservi come ci si trovi effettivamente nelle ipotesi per applicare il filtro di Kalman: i rumori  $n_1(\cdot)$  e  $n_2(\cdot)$  sono bianchi, gaussiani, a media nulla e scorrelati, la matrice  $R$  è definita positiva, mentre  $Q$  è semidefinita:

$$R = DD^T = [d^2] > 0 \quad Q = BB^T = \begin{bmatrix} 0 & 0 \\ 0 & b^2 \end{bmatrix} \geq 0$$

Implementare il filtro di Kalman all'interno degli schemi Simulink sviluppati è molto semplice: è bastato inserire un opportuno blocco funzione dentro ogni telecamera. La cosa interessante e che dà adito a diverse strategie implementative è come fare in modo che le telecamere si "passino" l'informazione riguardo al tracking per ottimizzare l'inseguimento dell'evento. Le possibilità sono svariate ed ognuna presenta aspetti positivi e negativi. Si è implementata quella a buon senso ritenuta più corretta, ragionando comunque a tutto tondo sul problema.

Una cosa fondamentale da tenere in considerazione è che esiste un trade-off tra patrolling e tracking: una camera impegnata nel tracking non effettua il patrolling del suo perimetro. D'altro canto, il tracking è un obiettivo prioritario ed è lecito attendersi un peggioramento dell'attività di patrolling pur di essere certi di non perdere l'evento, che potenzialmente può essere pericoloso. Per tale motivo è buona norma che l'evento

non vada perso mai, neanche nei passaggi tra le zone di copertura di due telecamere adiacenti. Si rende dunque necessaria una strategia di coordinazione per far sì che le camere confinanti con una impegnata in un'azione di tracking siano pronte a “ricevere” l'evento. Tenuto conto di questi ragionamenti, si è deciso di procedere nel modo seguente:

1. la telecamera impegnata nel tracking esercita ai suoi estremi un forza maggiore; in tal modo essa fa estendere verso di sé i bordi di copertura delle camere adiacenti fino al loro limite fisico (e ciò si ripercuote a cascata su tutto il sistema) per consentire il patrolling in parte della sua zona di competenza, che altrimenti sarebbe del tutto trascurata;
2. la camera impegnata nel tracking segue l'evento fino ai suoi bordi fisici;
3. la camera che esegue il tracking calcola ad ogni istante di comunicazione il tempo stimato nel quale l'evento raggiungerà i bordi fisici ed invia tali informazioni alle camere confinanti;
4. le camere adiacenti, ricevute tali informazioni, le confrontano con i tempi a loro necessari per arrivare al bordo dove si presenterà l'evento ed eventualmente si dirigono verso il bordo in questione.

Gli ultimi tre punti appena riportati garantiscono che l'evento non venga perso: infatti le camere adiacenti a quella impegnata nel tracking effettuano il patrolling e si muovono verso i bordi solo se necessario, ovvero se l'evento è sufficientemente vicino al bordo o si muove con velocità elevata. Qualora esse fossero troppo lontane dal bordo e non facessero in tempo ad arrivare, l'evento comunque non andrebbe perso perchè la telecamera precedentemente impegnata a seguirlo lo “abbandona” solamente quando raggiunge il proprio limite fisico, garantendo dunque robustezza al metodo implementato. Tale robustezza si manifesta anche dal fatto che, in certi istanti di tempo, potrebbe accadere che due camere effettuino contemporaneamente il tracking, ognuna in maniera indipendente; ciò è un fatto positivo se si pensa, ad esempio, alla situazione in cui l'evento si trova in prossimità del bordo ottimo senza muoversi significativamente: è corretto che entrambe le camere lo monitorino perchè, qualora esso acquisti improvvisamente velocità, si riesca a seguirlo sia che vada nella zona di copertura di una camera sia che vada nell'altra; inoltre si pone rimedio al fatto che l'evento si trova all'estremo di copertura della telecamera e potrebbe accadere che la qualità di visione in tali condizioni peggiori.

### 7.3 Implementazione

Per semplicità, l'evento viene generato fuori linea tramite un'opportuna funzione che realizza il modello in spazio di stato 10. L'evento viene dunque inserito nello schema Simulink e dato in ingresso alle telecamere. Ognuna di esse al suo interno presenta un opportuno filtro (figura 19) che sostituisce il sistema di visione: esso valuta, tramite semplici operazioni logiche, se l'evento si trova o meno nel field of view della telecamera.

In caso di risposta affermativa, viene attivato un valore logico (che potremmo definire “bit di presenza”) che comanda alla camera di avviare il filtro di Kalman per seguire l’evento ed interrompere l’azione di patrolling. Il filtro di Kalman è in realtà un blocco

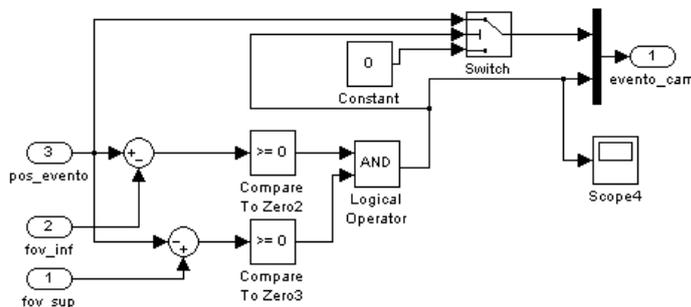


Figura 19: Filtro per l’individuazione dell’evento

più complesso (figura 20) che non si limita ad eseguire la funzione *kalmanfilter.m* ma calcola anche i tempi di arrivo previsti ai bordi ottimi. Per fare ciò vengono estratte le componenti dello stato: la posizione viene utilizzata per calcolare la distanza dai bordi e la velocità, opportunamente filtrata, viene infine usata per il calcolo del tempo di arrivo previsto (figura 21). La velocità viene filtrata per tre motivi principali:

- l’obiettivo è stimare l’andamento medio dell’evento nel tempo, trascurando eventuali piccole oscillazioni locali;
- la comunicazione non avviene ad ogni istante temporale;
- si vogliono evitare indecisioni da parte delle telecamere adiacenti nell’avvicinarsi al bordo per “ricevere” l’evento in arrivo. Tali indecisioni sarebbero presenti qualora si trasmettesse una stima del tempo di arrivo troppo variabile.

I tempi così calcolati vengono passati alle telecamere vicine, mentre il bit di presenza resta alto per tutto il tempo in cui l’evento è presente per poter comandare alla telecamera di interrompere il patrolling ed eseguire il tracking. Il bit di presenza comanda

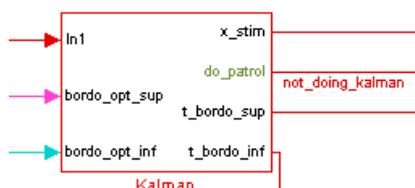


Figura 20: Blocco di Kalman

anche l’adattamento della forza (punto 1 dell’elenco nella precedente sezione, figura 22), per far sì che la telecamera impegnata nel tracking abbia più peso nel calcolo del bordo ottimo. Come detto in precedenza, ciò si ottiene sommando alla normale forza

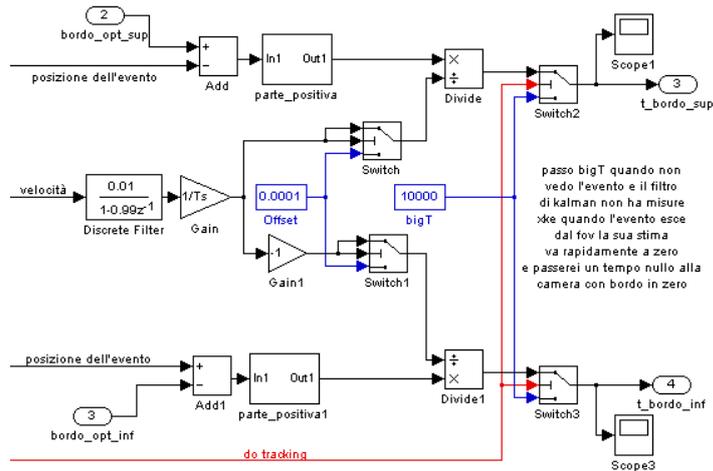


Figura 21: Particolare del blocco di Kalman, calcolo dei tempi di arrivo ai bordi

di attrazione esercitata dalla camera una forza sufficientemente grande da far spostare i bordi di copertura delle camere adiacenti ai loro limiti fisici. Si è scelto di sommare un bias e non di moltiplicare la forza per un qualche valore per evitare fenomeni oscillatori indesiderati: sommare vuol dire imporre un vincolo, moltiplicare significa invece modificare le caratteristiche della molla e ciò potrebbe indurre a vibrazioni indesiderate che potrebbero anche portare all'instabilità. Per far sì che la camera effettui il tracking fino

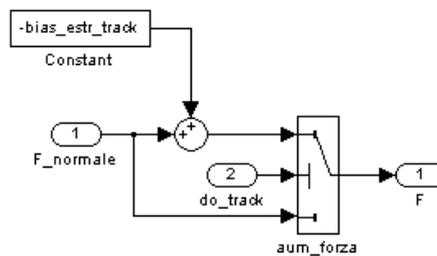


Figura 22: Adattamento della forza in presenza di tracking

ai suoi bordi fisici (punto 2 dell'elenco) è sufficiente inserire una semplice saturazione su tali limiti. Una volta che la camera li raggiunge, se l'evento continua a muoversi, esso uscirà dal field of view e dunque il bit di presenza per la camera in questione si resetta e la telecamera riprende la normale azione di patrolling, in un'area ottima che subisce l'influenza del fatto che la camera vicina è impegnata nel tracking.

## 8 Simulazioni

### 8.1 Dimensionamento dei parametri ed analisi del transitorio

In questa sezione si valuta l'influenza dei parametri dinamici del sistema sulle caratteristiche del transitorio degli estremi ottimi, integrando i risultati della simulazione con ulteriori considerazioni che estendono l'analisi teorica effettuata nella sezione 5.3.

#### 8.1.1 Stima della velocità di convergenza

Si sfrutterà qui l'analisi in spazio di stato effettuata in precedenza per stimare la velocità di convergenza del sistema nel calcolo delle sezioni ottime di copertura. La dinamica studiata in precedenza si basava su un modello a tempo continuo; tuttavia il sistema in questione è digitale, cioè opera a tempo discreto con frequenza  $f_s$ ; supponendo che ci sia comunicazione ad ogni istante di campionamento ( $p_{com} = 1$ ), tramite uno *zero-order-hold* si applicano le stesse grandezze tra due istanti successivi ( $t_i, t_i + T_s$ ): il sistema andrebbe quindi discretizzato e si dovrebbe poi considerare il modello a tempo continuo ottenuto tramite l'utilizzo dello *zero-order-hold*; tale analisi non è complessa, tuttavia nel caso specifico il periodo  $T_s$  può essere considerato sufficientemente piccolo (dell'ordine del centesimo di secondo) pertanto si è ritenuto lecito stimare la velocità di convergenza sulla base del modello a tempo continuo (sapendo comunque di compiere un'approssimazione). Pertanto si è operato come segue:

- calcolo degli autovalori  $\{\lambda_i\}$  di  $F$ ;
- determinazione dell'autovalore dominante  $\lambda_{dom}$ ;
- stima del tempo di convergenza  $t_{ass}$ , considerando di essere a regime con un errore del 1%:

$$e^{\lambda_{dom}t} < 0.01 \quad \Rightarrow \quad t_{ass} > \frac{\ln(0.01)}{\lambda_{dom}}$$

La stima ottenuta è corretta in quanto il sistema in esame è in evoluzione libera: pertanto, una volta determinato l'autovalore dominante, il tempo necessario per andare alla soluzione di equilibrio con un errore inferiore ad una percentuale fissata è calcolabile semplicemente come indicato nel punto 3.

**Osservazione 1.** La soluzione di equilibrio del modello nella forma in sezione 5.3 non è nell'origine; tuttavia è sempre possibile effettuare una traslazione delle coordinate in spazio di stato per portare il sistema considerata ad avere il punto di equilibrio nell'origine e la dinamica associata al sistema traslato è la stessa di prima, pertanto le considerazioni effettuate per la stima del  $t_{ass}$  restano valide.

**Osservazione 2.** Un ulteriore importante parametro legato alle prestazioni del sistema è la *comunicazione asincrona*: tuttavia nell'analisi del tempo di assestamento questo parametro dovrebbe avere un peso minore, in quanto più che gli istanti in cui avviene effettivamente la comunicazione è importante il numero di volte in cui essa avviene nell'arco di tempo  $t$  considerato, e questo dipende solo dai parametri  $f_s$  e  $p_{com}$ .

### 8.1.2 Dipendenza dai parametri

I parametri che intervengono direttamente nel determinare le caratteristiche della convergenza verso la situazione ottima a regime come già detto sono:  $K_e$ ,  $m$ ,  $\mu$  e  $f_s$ ,  $p_{com}$ . E' utile eseguire questa analisi considerando prima l'effetto dei primi tre, e successivamente quello degli ultimi due.

**Transitorio in funzione di  $K_e$ ,  $m$ ,  $\mu$ .** Il modo in cui questi tre parametri intervengono nel caratterizzare il transitorio è già stato accennato nelle sezioni precedenti in cui si mostrava l'analogia con il sistema meccanico di riferimento; i valori delle  $K_e$  sono fissati univocamente dalle velocità massime della camere, mentre il valore di  $m$  può essere scelto per garantire un'appropriata prontezza del sistema. La scelta più importante e delicata è tuttavia quella di  $\mu$ : come già detto basandosi sul sistema semplice molla-smorzatore si è visto come una prima scelta del parametro potesse essere quella per la quale  $\xi = 1$ . Si è deciso di imporre smorzamento critico al sistema composto dalla coppia di molle agganciata alla singola massa con smorzatore. Ragionando per coppie successive, si è giunti al seguente criterio per determinare i coefficienti di attrito:

$$\mu_{i,i+1} = 2\xi\sqrt{m(k_i + k_{i+1})}$$

Con questa scelta si è visto che per valori di  $m$  ragionevoli, anche per il sistema in questione, così come in quello semplice di riferimento molla-smorzatore, i poli risultano essere tutti reali, il che garantisce per la dinamica a tempo continuo un transitorio senza oscillazioni. Sperimentalmente si è visto che per valori di  $f_s$  sufficientemente elevati tale proprietà rimane presente; inoltre la presenza di soli poli reali rende particolarmente semplice l'individuazione del polo dominante e di conseguenza il calcolo del tempo di assestamento, anch'esso parametro che trova riscontro nelle simulazioni effettuate.

**Transitorio in funzione di  $f_s$  e  $p_{com}$ .** Mentre i parametri precedenti (escluse le costanti elastiche  $K_{e,i}$ ) rappresentano dei gradi di libertà nel progetto della dinamica di controllo,  $f_s$  e  $p_{com}$  sono invece due parametri che si possono ritenere fissati dalle particolari condizioni in cui opera il sistema; come già detto si può ritenere che la frequenza a cui opera il sistema di controllo  $f_s$  sia sufficientemente elevata da permettere di approssimare la dinamica complessiva del sistema con quella a tempo continuo (valori verosimili utilizzati nelle simulazioni variavano da 10 a 100 il che significa  $T_s$  dell'ordine del decimo o centesimo di secondo). Il parametro  $p_{com}$  è stato scelto a sua volta in modo da rendere significative le simulazioni rispetto a quelle che potevano essere le problematiche della comunicazione reale.

E' molto interessante l'effetto di questi parametri sulla convergenza verso la soluzione di equilibrio: valori critici di  $p_{com}$  e di  $f_s$  inducono un transitorio con molte oscillazioni degli estremi delle sezioni di copertura (fatto che con la scelta appropriata dei parametri  $m$  e  $\mu$  si riusciva ad evitare in presenza di valori non problematici di  $p_{com}$  e  $f_s$ ) ma non peggiorano in modo significativo il tempo di assestamento  $t_{ass}$  che risulta quindi con buona approssimazione essere legato principalmente alle scelte dei tre parametri precedenti; tale risultato è positivo se si pensa che, in caso di  $p_{com}$  bassa, si possono regolare i tre parametri precedenti in modo da migliorare le prestazioni del transitorio.

## 8.2 Simulazioni con comunicazione frequente

Si propone una prima simulazione in cui la sovrapposizione tra i range fisici delle camere è tale che la soluzione ottima senza vincoli e la soluzione ottima con vincoli coincidano. In tutte le simulazioni riportate, la numerazione delle camere va dal basso all'alto, le linee nere rappresentano i bordi ottimi di copertura, le linee tratteggiate sono i limiti fisici della telecamera del loro medesimo colore, in rosso è rappresentato l'evento, in arancione lo stato misurato (quest'ultimo è rappresentato solo nelle ultime due simulazioni riportate in appendice B). Gli estremi di patrolling si assestano ad un valore interno alla regione sovrapposta, e le camere hanno tutte lo stesso periodo di patrolling. La messa in fase è disattivata in modo da consentire di apprezzare come effettivamente i periodi tra le camere siano uguali per effetto del corretto posizionamento degli estremi di patrolling secondo il criterio ottimo, e non è invece legata alla scelta di sincronizzare o meno la fase delle camere. Per soddisfare alcuni requisiti implementativi (check di raggiungimento dell'estremo di patrolling con il *field of view* della camera, e poi tracking dell'evento), è importante che nelle simulazioni il *fov* non sia puntiforme ma abbia dimensione fisica. Si propongono dunque i seguenti valori per la simulazione, con le lunghezze intese in termini relativi rispetto a un'unità di misura che si può fissare liberamente per dimensionare il sistema. In un problema realistico l'unità di lunghezza potrebbe corrispondere a 10 m, scegliamo qui per semplicità di usare una lunghezza base di 1 m. La simulazione è impostata con 5 camere, con un percorso che si estende dalla posizione assoluta 0 m alla posizione assoluta 5 m. Il risultato è riportato in figura 23. Con questi parametri il sistema si assesta correttamente alla situazione di regime desiderata, con la lunghezza delle sezioni di patrolling conforme alla proposizione 2.

- $fov$  delle camere: [0.2 0.2 0.2 0.2 0.2] m
- velocità massima delle camere: [0.18 0.23 0.15 0.25 0.16] m/s
- limiti inferiori zone sovrapposte, posizione assoluta: [0.8 1.8 2.8 3.8] m
- limiti superiori zone sovrapposte, posizione assoluta: [1.2 2.2 3.2 4.2] m
- $T_s = 0.05$  s,  $p_{com} = 0.4$
- massa degli estremi di patrolling  $m = 2$  kg

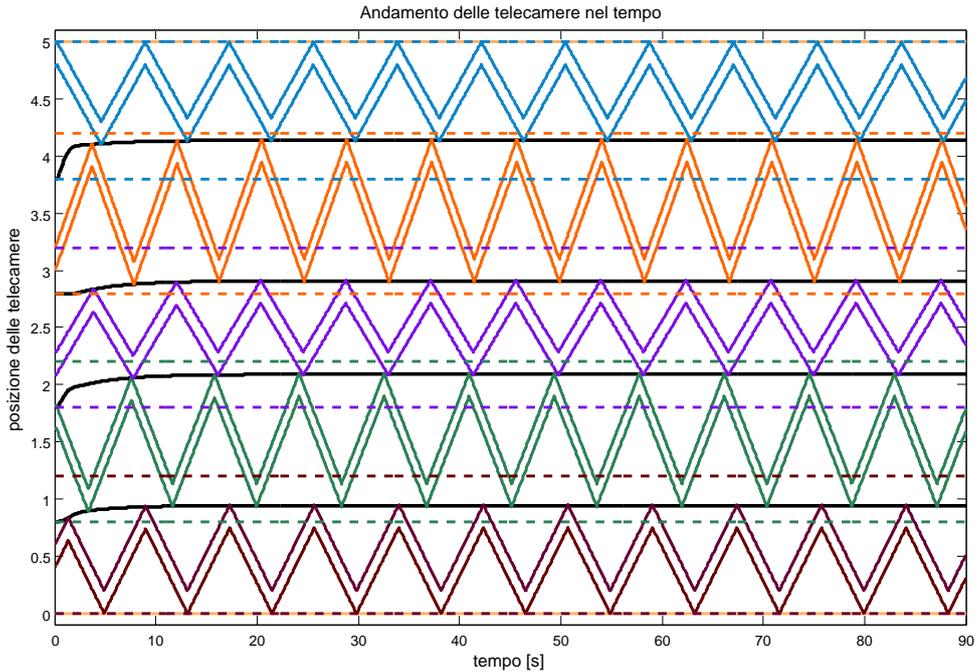


Figura 23: Situazione di regime con ottimo di patrolling strettamente entro i vincoli

Si introduce ora un vincolo più stringente nella zona centrale in modo da indurre due periodi diversi tra le prime due camere e le ultime tre. Viene attivata la sincronizzazione in opposizione di fase, ed a metà simulazione una delle camere viene bloccata per mostrare come gli estremi di patrolling si riadattino per compensare la perdita di una telecamera. Per indurre questo comportamento si è introdotto un evento fittizio nel range della camera 2, in quanto non è stata implementata nelle camere una struttura che permetta loro di capire quando una delle loro vicine si guasta. Tuttavia, introducendo alcuni check sui parametri ricevuti dalla camera vicina e sul tempo trascorso dalla ultima comunicazione avvenuta con successo, è facile integrare questa peculiarità in un sistema volto ad applicazioni reali, che risulta così essere robusto alla rottura di una telecamera (figura 24).

I parametri variati rispetto alla precedente simulazione sono:

- limiti inferiori zone sovrapposte, posizione assoluta: [0.8 1.5 2.3 3.8] m
- limiti superiori zone sovrapposte, posizione assoluta: [1.2 1.8 2.8 4.2] m

### 8.3 Verifica del tempo di assestamento e limiti del sistema

Si applica la stima teorica del tempo di convergenza alla dinamica con gli stessi dati della figura 23. Gli autovalori risultano essere:

$$\begin{array}{cccc} -1.2042 & -0.1099 & -0.7361 & -0.4796 \\ -9.1612 & -8.0083 & -8.2806 & -8.6011 \end{array}$$

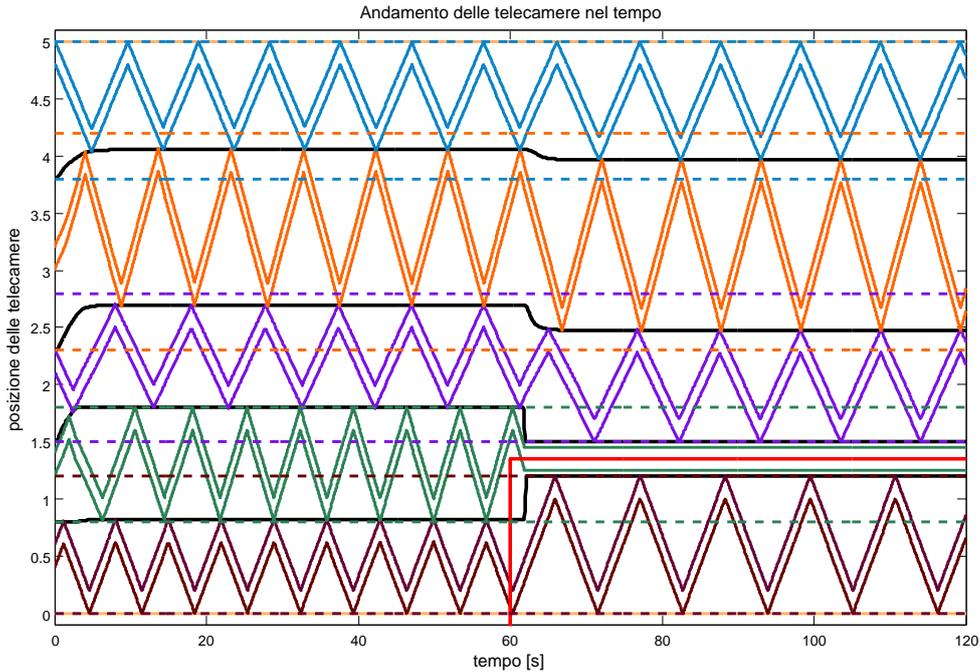


Figura 24: Soluzione ottima con vincoli stringenti e perdita di una camera

Rispetto all'autovalore dominante, la stima del tempo di salita risulta essere 41.9 s; rispetto al primo autovalore non dominante risulta invece essere 9.6 s. La simulazione mostrava la convergenza entro circa 20 s.

**Osservazioni.** Da questa ed altre simulazioni emerge che la stima del tempo di assestamento sull'autovalore dominante tende ad essere troppo pessimistica; questo si può spiegare pensando al fatto che all'autovalore dominante è associato un autovettore nello spazio di stato, e la condizione iniziale non è detto sia tale da avere proiezione sul sottospazio relativo. Per questo motivo, nell'esempio precedente, si è mostrata anche la stima basata sul secondo autovalore. In generale, si è osservato che se il sistema tende ad avere una dinamica veloce (valori di  $m$  piccoli rispetto al caso precedente) la stima effettuata è più precisa in quanto gli autovalori tendono a essere più vicini tra loro. Viceversa, se la dinamica è estremamente lenta ( $m$  grande) al sistema è associato un autovalore dominante che si discosta molto dagli altri autovalori, e molto vicino a 0; come appena sottolineato comunque non è detto che la condizione iniziale sia tale da eccitare tale modo.

Vi è un ultimo aspetto che tende a rendere troppo pessimistica la stima effettuata: il sistema reale tra due istanti di comunicazione successivi mantiene costante la forza elastica applicata agli estremi. Nel modello a tempo continuo invece la forza si riduce con continuità man mano che ci si avvicina al punto di equilibrio: pertanto, il sistema in esame tende ad avere una dinamica più veloce, a scapito della possibile comparsa di oscillazioni.

Proponiamo ora un'altra simulazione con dinamica più veloce (figura 25); per fare ciò, si è impostata la massa degli estremi a  $m = 1$  kg. Lasciamo disattivata per ora la messa in fase, per riprenderla poi nei casi con tracking dell'evento.

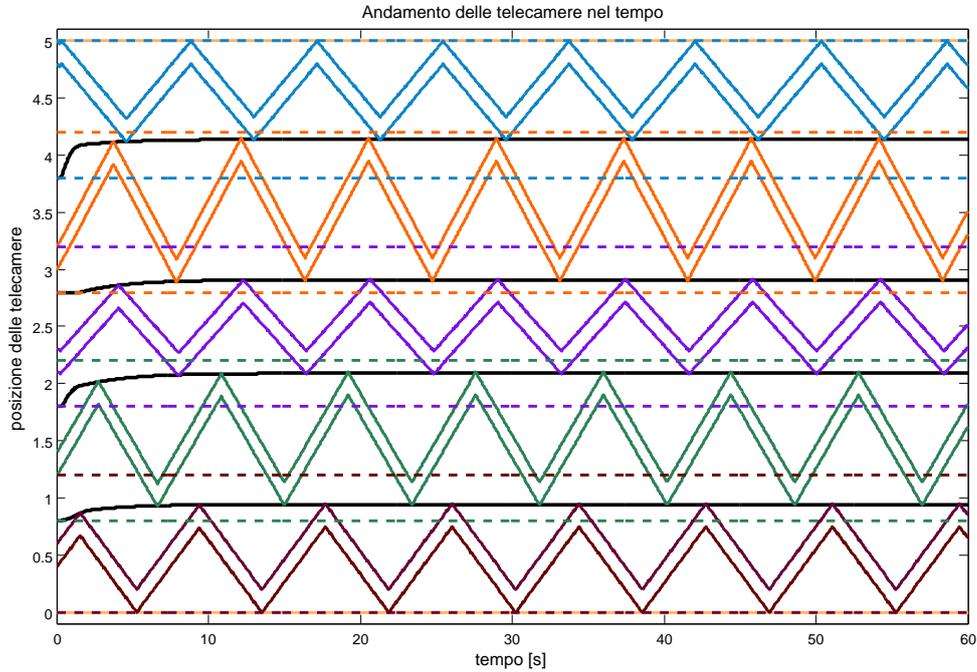


Figura 25: Dinamica degli estremi più veloce, massa inferiore

La stima del tempo di salita sull'autovalore dominante risulta essere  $14.3$  s, in buon accordo con il risultato sperimentale.

Esaminiamo ora un caso con comunicazione poco frequente (figura 26).

- $p_{com} = 0.05$
- $m = 2$  kg

Essendo  $T_s = 0.05$  s, il tempo medio tra una comunicazione e l'altra è 1 s. Ci si spinge verso valori di  $p_{com}$  molto bassi; si osservi tuttavia come la cosa realmente importante sia l'effettivo tempo medio tra due comunicazioni. In un sistema reale, più verosimilmente si tenterà la comunicazione più raramente (es ogni  $n$  istanti di campionamento) e si richiederà una  $p_{com}$  maggiore.

Si manifesta un comportamento oscillatorio indesiderato, che può essere rimosso aumentando la massa degli estremi (figura 27):

- $p_{com} = 0.05$
- $m = 10$  kg

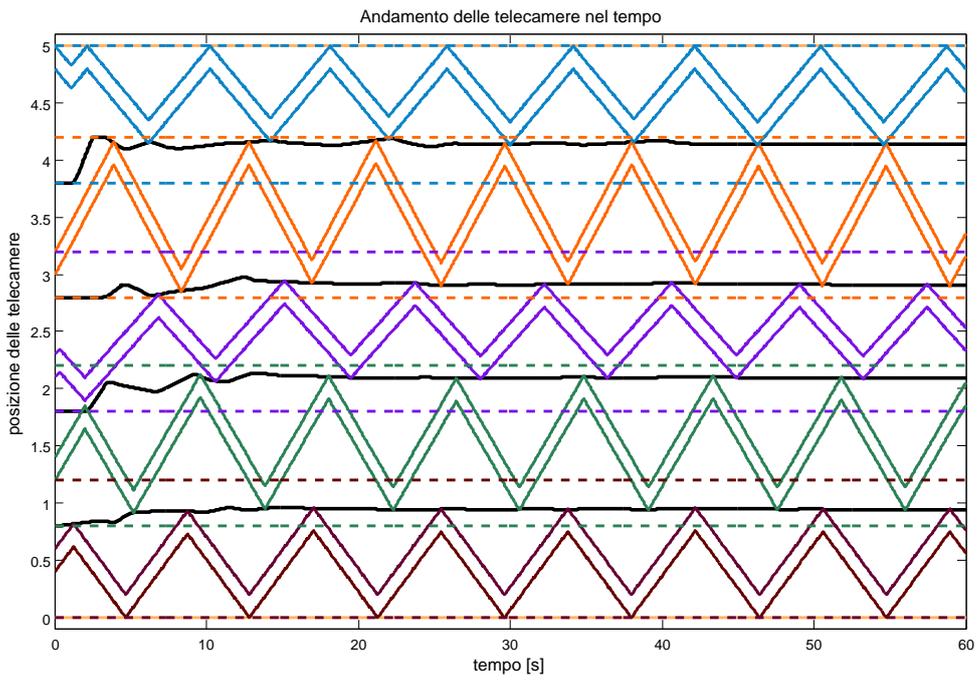


Figura 26: Dinamica degli estremi con scarsa comunicazione

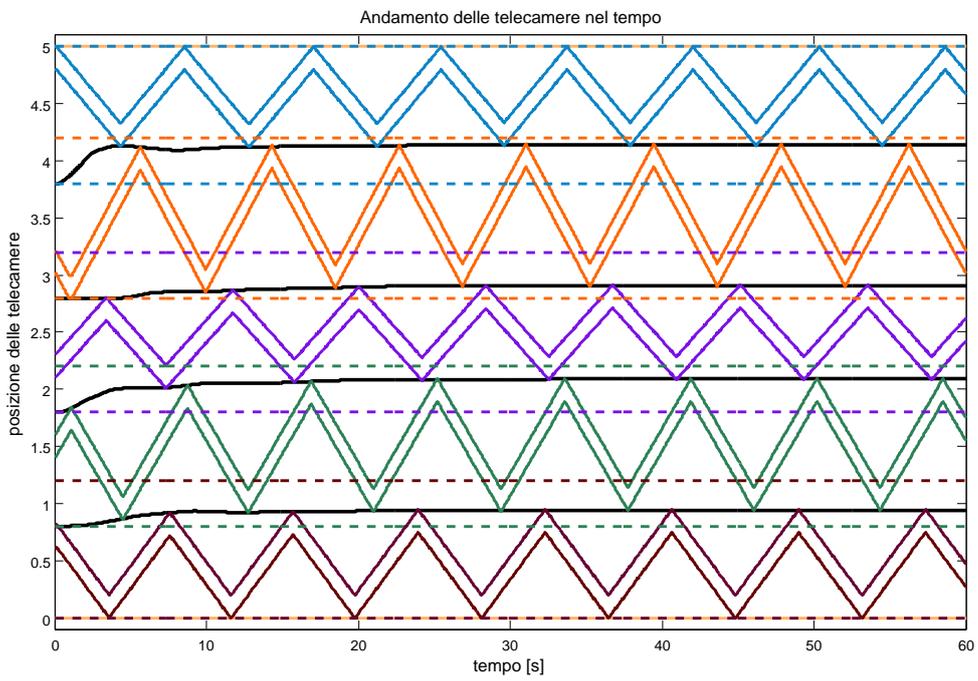


Figura 27: Scarsa comunicazione e massa elevata degli estremi

## 8.4 Simulazioni in presenza di evento

Per quanto riguarda la modellizzazione dell'evento (cfr. sez. 7), sono stati utilizzati i seguenti parametri:

$$a = 0.999 \quad b = 0.0005 \quad d = 0.002$$

Il valore di  $a$  fa sì che l'evento non acceleri ma al tempo stesso rimanga in movimento per la durata della simulazione; il parametro  $b$  introduce una certa variazione nella velocità dell'evento, ma soprattutto nel suo verso, il parametro  $d$  è la varianza dell'errore di misura e non è stata supposta troppo elevata. Si veda l'appendice B per delle simulazioni con parametri diversi da quello qui riportati. Per la prima simulazione, sono stati mantenuti i limiti fisici e le velocità già utilizzate per la simulazione di figura 23.

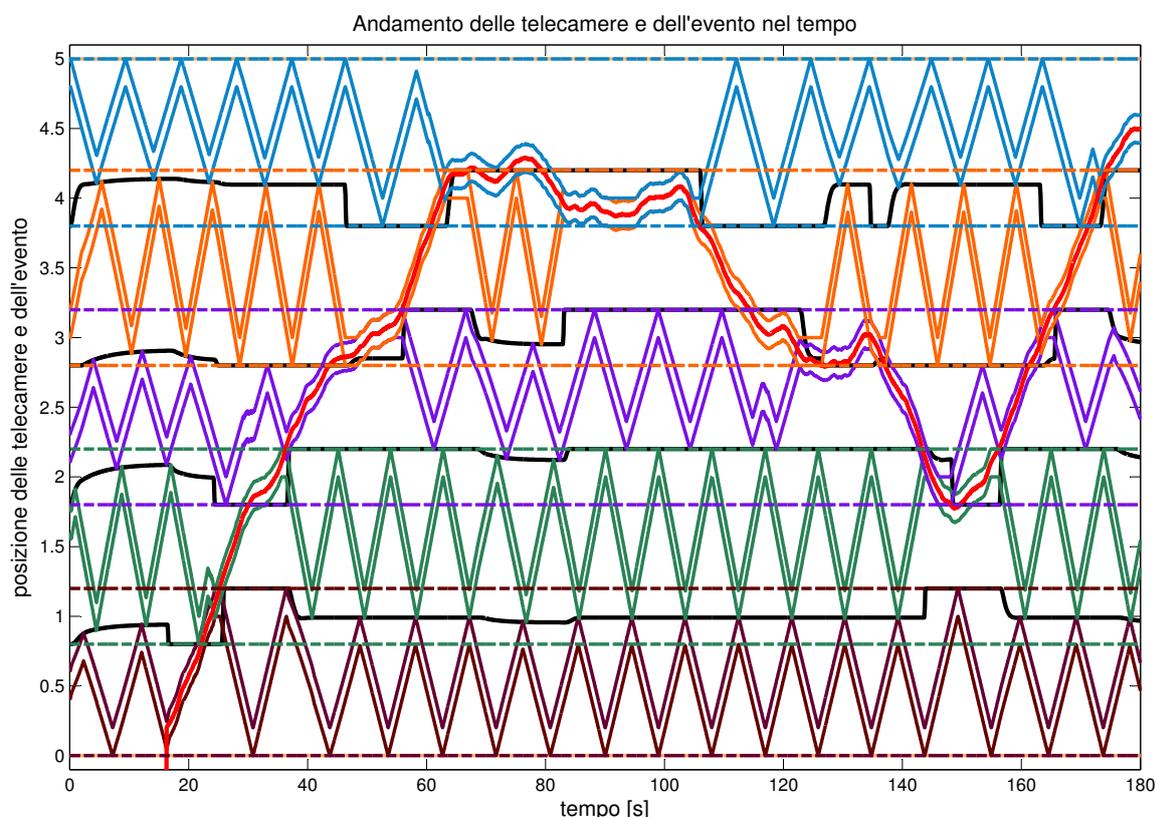


Figura 28: Tracking di evento

Dalla simulazione emergono chiaramente le ipotesi che sono state adottate e si verifica come il sistema progettato risponda all'obiettivo richiesto. Innanzitutto si nota come il tracking sia prioritario: dal momento che l'evento entra nel field of view di una telecamera, questa non lo abbandona mai, fintantochè esso si trova all'interno dei limiti fisici della camera in questione. Si noti inoltre l'azione di "richiamo" da parte della camera impegnata nel tracking nei confronti delle altre: quando, ad esempio, l'evento

è in procinto di entrare nella zona preposta alla seconda camera, la prima trasmette l'informazione sul tempo stimato, chiamando così a sè la seconda che, interrotta la propria azione di patrolling, si dirige verso il suo bordo fisico per ricevere l'evento. Si noti inoltre come la camera impegnata nel tracking eserciti una forza maggiore sui bordi ottimi, avvicinandoli a sè, per consentire alle camere adiacenti di effettuare il patrolling in parte della propria zona di competenza. Si cerca dunque di ottimizzare il patrolling, ciò emerge anche dal fatto che le camere, qualora non impegnate nel tracking, cerchino comunque di portarsi in opposizione di fase per migliorare la loro azione di patrolling (si vedano le camere 1 e 2, dopo che hanno abbandonato l'evento).

Il sistema progettato si comporta in maniera egregia anche in condizioni più sfavorevoli per le telecamere, ovvero con sezioni fisiche molto differenti, dovute a una maggior differenza di velocità tra le camere. Si veda a tal scopo la figura 29 dove sono stati utilizzati i seguenti parametri:

- velocità massima delle camere: [0.12 0.23 0.14 0.25 0.19] m/s
- limiti inferiori delle zone sovrapposte, posizione assoluta: [0.6 2.1 2.6 3.8] m
- limiti superiori delle zone sovrapposte, posizione assoluta: [0.7 2.5 2.8 4] m

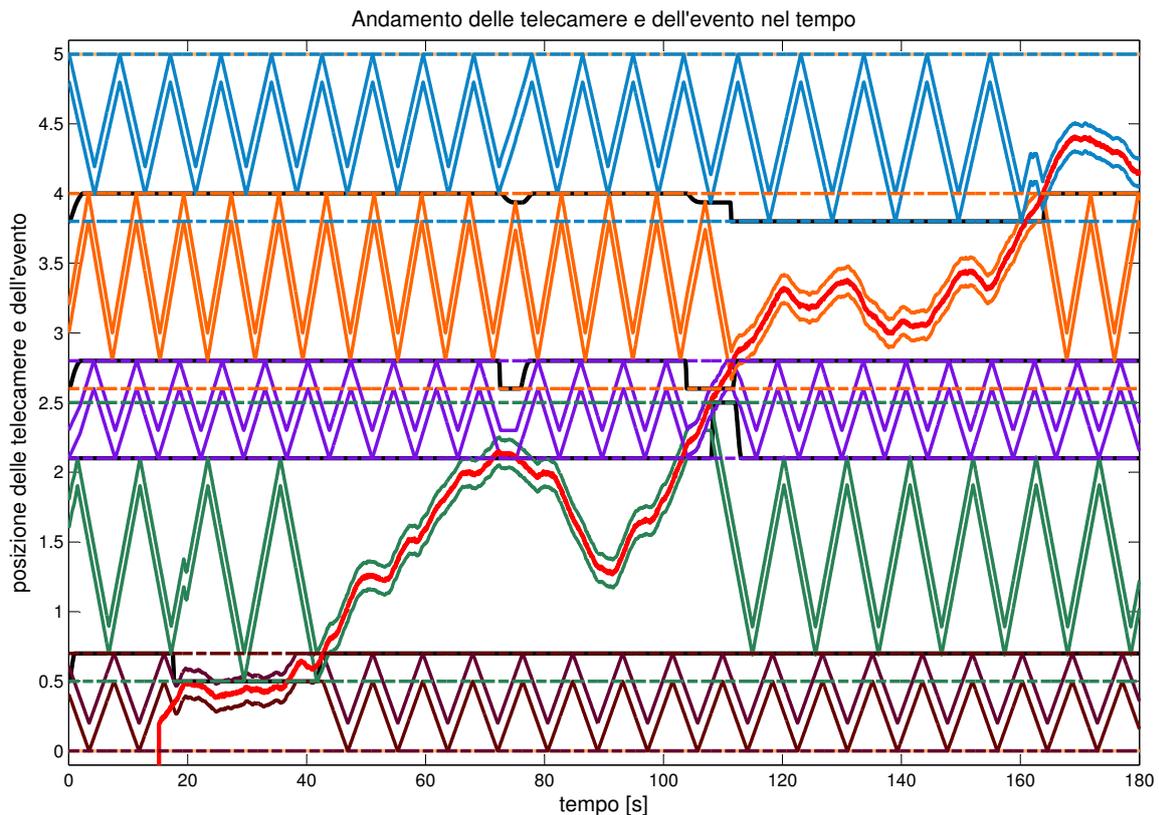


Figura 29: Tracking di evento con vincoli più stringenti e velocità differenti

Anche in questo caso, con la telecamera 2 sensibilmente più veloce della 1 e della 3, si ottengono comunque gli obiettivi desiderati, prima di tutto il tracking prioritario dell'evento che non viene mai perso. Anche stavolta le camere si chiamano l'un l'altra quando necessario, si noti a tal proposito l'incertezza all'istante  $t = 20$  s della camera 2: l'evento è in prossimità del bordo fisico e dunque la camera 1 la "chiama" affinché l'evento non vada perso neanche per un solo istante, tuttavia subito dopo l'evento cambia direzione e dunque la camera 2 riprende la sua normale azione di patrolling. In questo caso non si ottiene l'opposizione di fase, date le velocità molto diverse e i limiti fisici, tuttavia il comportamento del sistema risponde alle specifiche richieste.

## 9 Conclusioni e Sviluppi futuri

### 9.1 Estensione al 1.5D

L'algoritmo distribuito implementato è stato concepito per risolvere problemi di copertura e tracking monodimensionali (1D) tuttavia presenta una naturale estensione a situazioni più complesse modellabili in 1.5D.

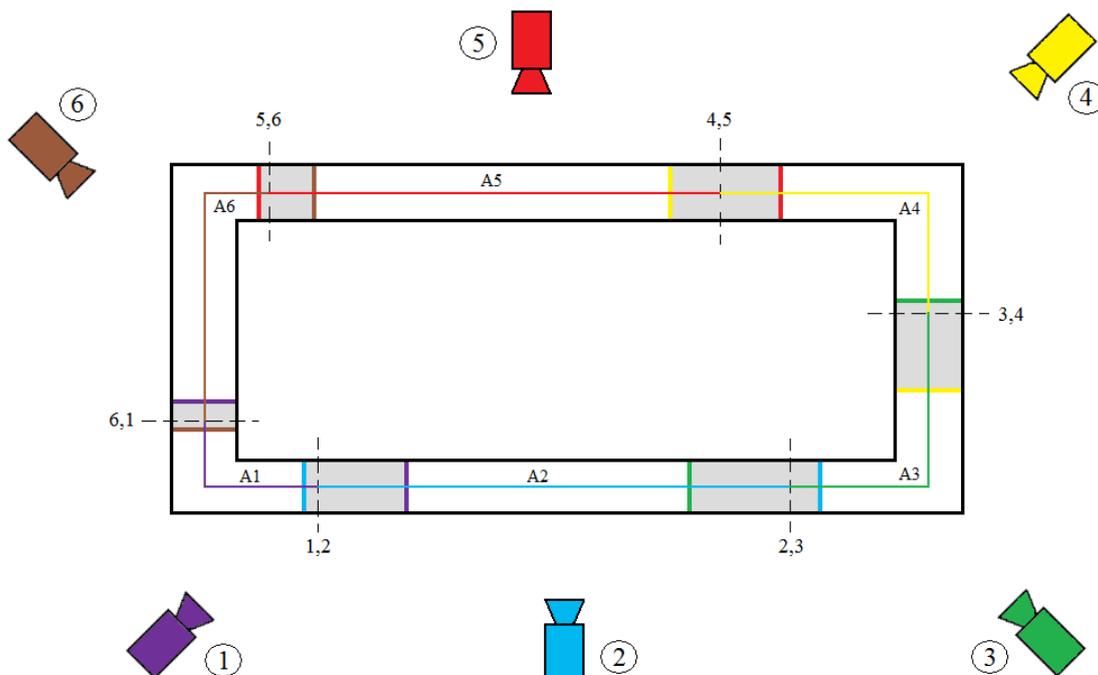


Figura 30: Perimetro rettangolare: applicazione del metodo proposto

Come si può intuire osservando la figura in tali contesti (perimetri chiusi con forma arbitraria) si può applicare direttamente la soluzione proposta senza particolari accorgimenti.

Situazioni reali più complesse possono essere quelle in cui il percorso da monitorare abbia una struttura con degli incroci, creando così delle zone condivise da più di due camere; in queste situazioni l'algoritmo proposto potrebbe essere applicato ma necessita di alcune modifiche. Un modo semplice ma efficace per risolvere il problema è quello mostrato in figura 31; si potrebbe quindi affidare la copertura dell'incrocio alla telecamera delle tre che sta coprendo la sua sezione di copertura con periodo inferiore, scelta coerente con il criterio di ottimizzazione proposto, fissando agli estremi dell'incrocio la zona di copertura delle altre due telecamere. Ovviamente questa è solo una delle possibili soluzioni adottabili; si potrebbe pensare anche a delle soluzioni più sofisticate che anche per la zone d'incrocio (qualora ad esempio queste fossero particolarmente ampie) tengano conto di un trade off nella copertura fra le telecamere interessate.

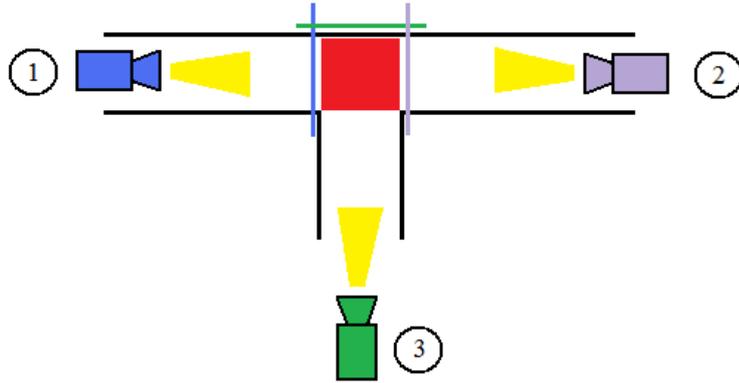


Figura 31: Possibile incrocio: caso con  $T_3 < T_1$ ,  $T_3 < T_2$

## 9.2 Osservazioni conclusive

La soluzione adottata per risolvere il problema in questione è stata ottenuta fissando un criterio di ottimalità, scelto in funzione di tale problema, e cercando di determinare una traiettoria che lo soddisfacesse; nonostante ci si fosse posti in ipotesi molto generali sia in termini di velocità che in termini di potenzialità di copertura degli agenti (e quindi il problema risultasse notevolmente vincolato) si è implementato un algoritmo notevolmente semplice, distribuito e molto robusto che risolve ottimalmente il problema posto, sfruttando un'analogia con un sistema meccanico per dimostrare le proprietà di convergenza verso il calcolo delle sezioni ottime.

La soluzione proposta è stata implementata in Simulink e ciò ha reso possibile realizzare un'implementazione complessa, che tenesse conto di una serie di problematiche reali che il sistema deve affrontare quali la possibilità di perdita di pacchetti durante la comunicazione e l'asincronismo della stessa tra i diversi agenti, constatando attraverso le simulazioni come la dinamica del sistema fosse molto robusta anche rispetto a variazioni di tali parametri.

Il problema del tracking è stato affrontato in modo efficace: all'evento si è attribuita una dinamica verosimile e, con l'implementazione di un filtro di Kalman su ciascun agente, è stato possibile effettuare in modo appropriato non solo la dinamica di inseguimento, ma anche quella di coordinamento tra due telecamere adiacenti per garantire la copertura dell'evento per tutti gli istanti di tempo in cui agisce lungo il percorso. Il tracking è stato quindi ragionevolmente concepito in modo prioritario, ciò nonostante le telecamere non impegnate in tale operazione si redistribuiscono in maniera ottimale la copertura delle restanti parti del percorso. Come si è detto precedentemente, la soluzione adottata ha una naturale estensione a casi reali modellabili con un 1.5D e ciò la rende effettivamente implementabile nella pratica in molte situazioni reali (monitoraggio di perimetri aperti o chiusi, di catene di montaggio anche complesse, di corridoi e percorsi stradali anche in presenza di incroci). Eventuali estensioni a casi ancora più complessi bidimensionali (2D) richiederebbero invece notevoli modifiche della soluzione proposta per adattarla a tali esigenze di copertura.

# APPENDICE

## A Approfondimenti sulla traiettoria ottima con vincoli

La generalizzazione di quanto dimostrato al caso in cui sia una telecamera nel mezzo (con  $i \neq 1, N$ ) o l'ultima a non soddisfare i vincoli di copertura all'ottimo è immediata.

**Telecamera  $i$ -esima nel mezzo:** In tal caso si possono avere diversi casi sfavorevoli:

1.  $C_{i,inf} > L_{i-1} \quad \wedge \quad C_{i,sup} < L_i$
2.  $C_{i,inf} > L_{i-1}$
3.  $C_{i,sup} < L_i$

**1° caso:** è forse quello più semplice da trattare; la scelta ottima c.v. risulta:

1.  $A_i = C_i$
2.  $|A_j| = V_{j,max} T_{ottS.c.v.}, \quad j < i$
3.  $|A_j| = V_{j,max} T_{ottD.c.v.}, \quad j > i$

con

$$T_{ottS.c.v.} = \frac{L_S}{\sum_{j=1}^{i-1} V_{j,max}} \quad T_{ottD.c.v.} = \frac{L_D}{\sum_{j=i+1}^N V_{j,max}}$$

dove  $L_S$  e  $L_D$  sono le lunghezze dei percorsi alla destra e alla sinistra della sezione ottima  $A_i$  fissata. La dimostrazione che questa è la scelta giusta discende immediatamente da quella per il caso  $n = 1$ : una qualsiasi altra scelta di  $A_i \subset C_i$  porterebbe, o per le telecamere alla sua destra o per quelle alla sua sinistra, a coprire un tratto di percorso maggiore ( $\bar{L}_S > L_S \vee \bar{L}_D > L_D$ ) con conseguente perdita di ottimalità sulla metà in questione e quindi perdita di ottimalità globale.

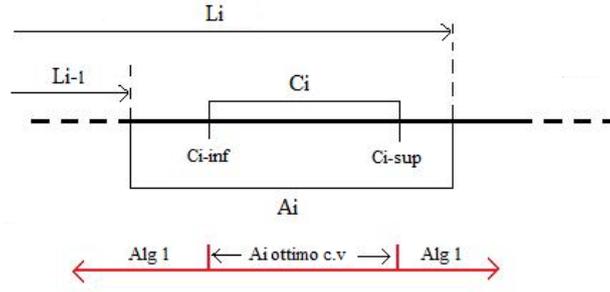


Figura 32: Scelta ottima c.v. in caso di vincolo violato (tipo 1) in una telecamera nel mezzo

**2° caso:** in questa situazione non vi è più simmetria come nel caso precedente. Per le telecamere a sinistra di  $i$  la scelta ottima c.v. risulta:

- $|A_j| = V_{j,max} T_{ottS.c.v.}, \quad j < i$

con

$$T_{ottS,c.v} = \frac{L_S}{\sum_{j=1}^{i-1} V_{j,max}}$$

Mentre per quelle a destra ( $i$  compresa) risulta:

- $|A_j| = V_{j,max} T_{ottD.c.v.}, \quad j \geq i$

con

$$T_{ottD,c.v} = \frac{L_D}{\sum_{j=i}^N V_{j,max}} = \frac{L_{tot} - L_S}{\sum_{j=i}^N V_{j,max}}$$

In questo modo l'ottimalità c.v. è garantita: se il margine sinistro fosse stato scelto maggiore del limite di copertura inferiore ( $L_{i-1} > C_{i,inf}$ ) allora le telecamere a sinistra della  $i$ -esima avrebbe coperto una lunghezza maggiore di quella invece da coprire con questa scelta (dunque si peggiora il  $T_{max}$ ); anche per  $i$  e le telecamere alla sua destra si deve riapplicare l'algoritmo 1 per la scelta delle  $A_i$ : in questo modo si è certi dell'ottimalità anche per le telecamere  $j \geq i$  (rispetto alla copertura della parte restante di percorso) e si può verificare anche il caso intuitivamente probabile che il nuovo margine destro di  $i$  sia maggiore di quello precedente ( $L_{i,ott.c.v} > L_{i,ott}$ ).

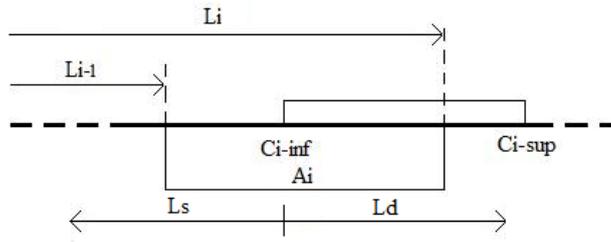


Figura 33: Vincolo violato (tipo 2) in una telecamera nel mezzo

**3° caso:** è analogo a quello precedente soltanto simmetrizzato; per le telecamere a destra si ha:

- $|A_j| = V_{j,max} T_{ottD.c.v.}, \quad j > i$

con

$$T_{ottD.c.v.} = \frac{L_D}{\sum_{j=i+1}^N V_{j,max}}$$

Mentre per quelle a sinistra ( $i$  compresa) risulta:

- $|A_j| = V_{j,max} T_{ottS.c.v.}, \quad j \leq i$

con

$$T_{ottS.c.v.} = \frac{L_S}{\sum_{j=1}^i V_{j,max}} = \frac{L_{tot} - L_D}{\sum_{j=1}^i V_{j,max}}$$

la dimostrazione dell'ottimalità è la stessa del caso precedente.

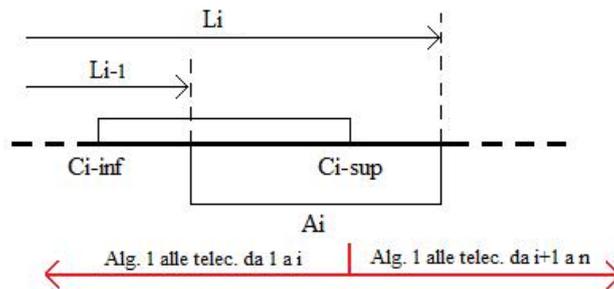


Figura 34: Scelta ottima c.v. in caso di vincolo violato (tipo 3) in una telecamera nel mezzo

**Telecamera N-esima finale:** In tal caso ci si trova nella situazione simmetrica rispetto alla prima telecamera. L'unico caso sfavorevole si manifesta quando:

$$C_{N,inf} > L_{N-1}$$

e cioè all'ultima telecamere è richiesto di coprire una sezione del percorso che precede il suo limite inferiore di copertura.

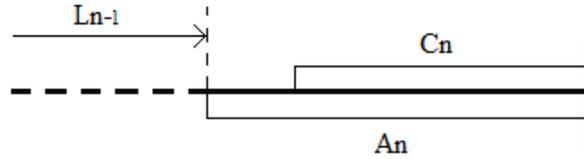


Figura 35: Vincolo violato nella telecamera N

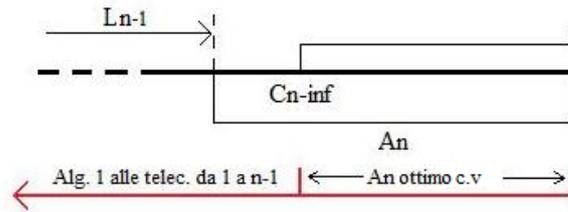


Figura 36: Scelta ottima c.v. in caso di vincolo violato nella telecamera N

La scelta ottima c.v. di copertura (dimostrazione del tutto analoga al caso  $n = 1$ ) si ottiene imponendo alla telecamera  $N$ -esima di coprire il maggior spazio possibile e ricalcolando le sezioni  $A_i$  per le altre telecamere nel percorso ridotto:

1.  $A_N = C_N$
2.  $|A_i| = V_{i,max} T_{ott.c.v.}, \quad i < N$

con

$$T_{ott.c.v.} = \frac{L - |C_N|}{\sum_{i=1}^{N-1} V_{i,max}}$$

## B Ulteriori simulazioni

**Consenso tra gli estremi.** Come è stato detto nella sezione 6, esistono due versioni dello stesso estremo di patrolling, una in ciascuna delle due camere adiacenti a tale estremo. Si mostra in figura 37 come il sistema di reset al consenso degli estremi funzioni nel modo corretto quando si va ad aggiungere del rumore nel modello dinamico per il calcolo degli estremi. Essi possono differenziarsi solo fintanto che non c'è comunicazione, ma mantengono un andamento medio comune in quanto la posizione viene riportata al consenso quando la comunicazione può avvenire. I parametri di simulazione sono gli stessi di figura 23

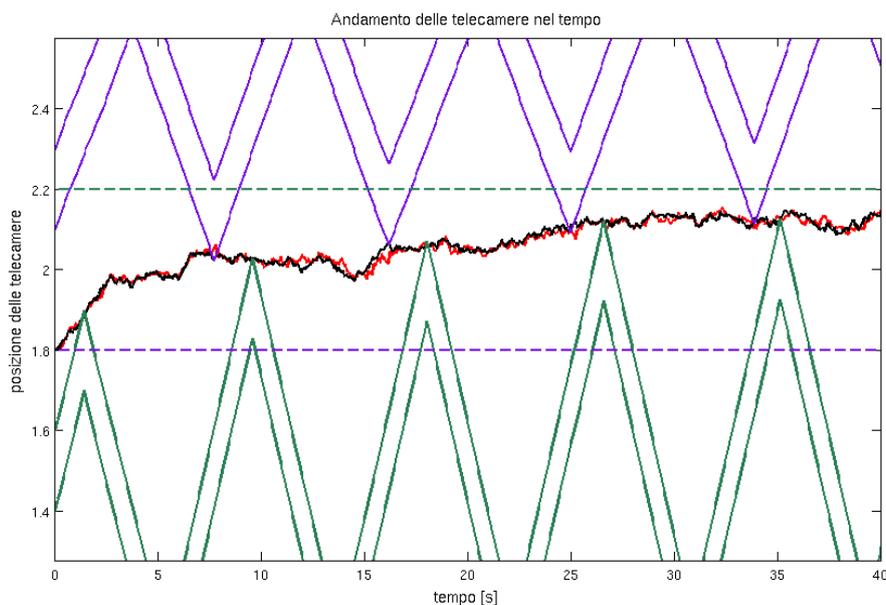
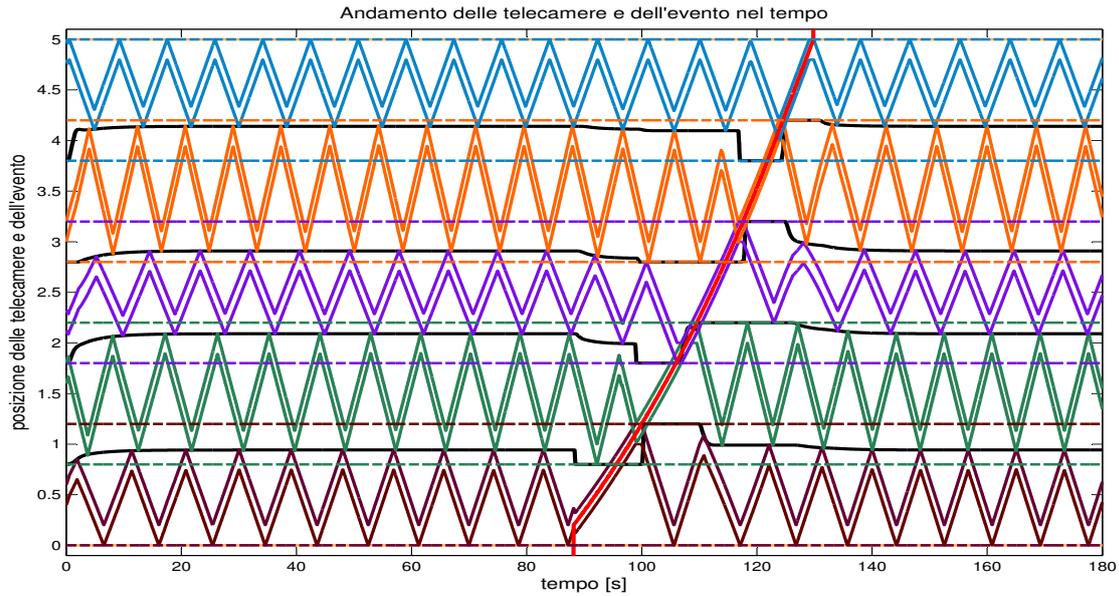


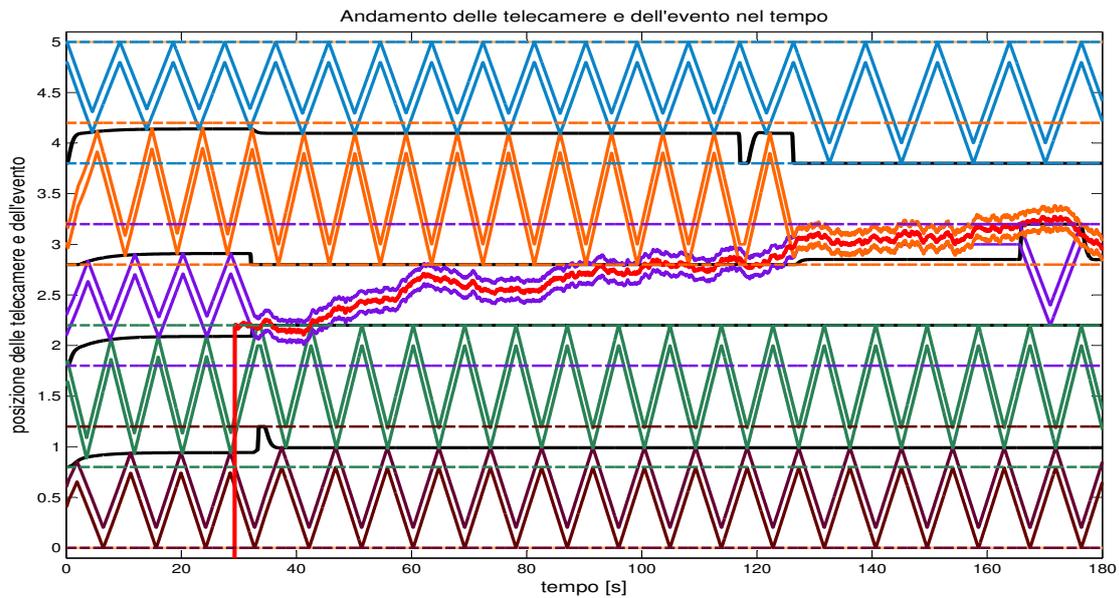
Figura 37: Reset al consenso degli estremi con dinamica alterata da rumore

**Tracking.** Si riportano ora ulteriori simulazioni in presenza di evento, ottenute modificando i parametri assegnati in sezione 8.4. I valori delle velocità e i bordi ottimi sono quelli tipici già utilizzati nelle precedenti simulazioni.

L'importanza del parametro  $a$  emerge con la simulazione di figura 38a. In essa è stato posto  $a = 1.001$  ed è stato tolto il rumore sulla velocità ( $b = 0$ ), mentre rimane quello standard di misura. Si vede come l'evento acceleri col passare del tempo, il valore di  $a$  utilizzato è quello limite per non perdere l'evento: con valori anche di poco maggiori il sistema di telecamere non riesce a mantenere agganciato l'evento. Ciò evidenzia la scelta sensata, in fase di simulazione, di vincolare l'evento a rispettare i limiti fisici di velocità delle camere.



(a) Tracking di evento che accelera

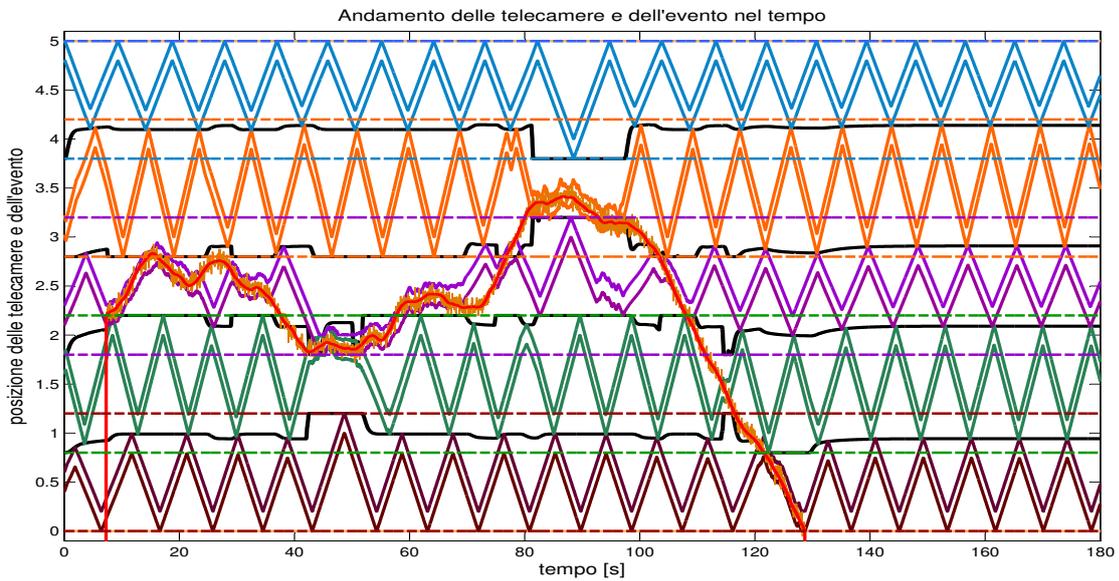


(b) Tracking di evento con dinamica molto variabile

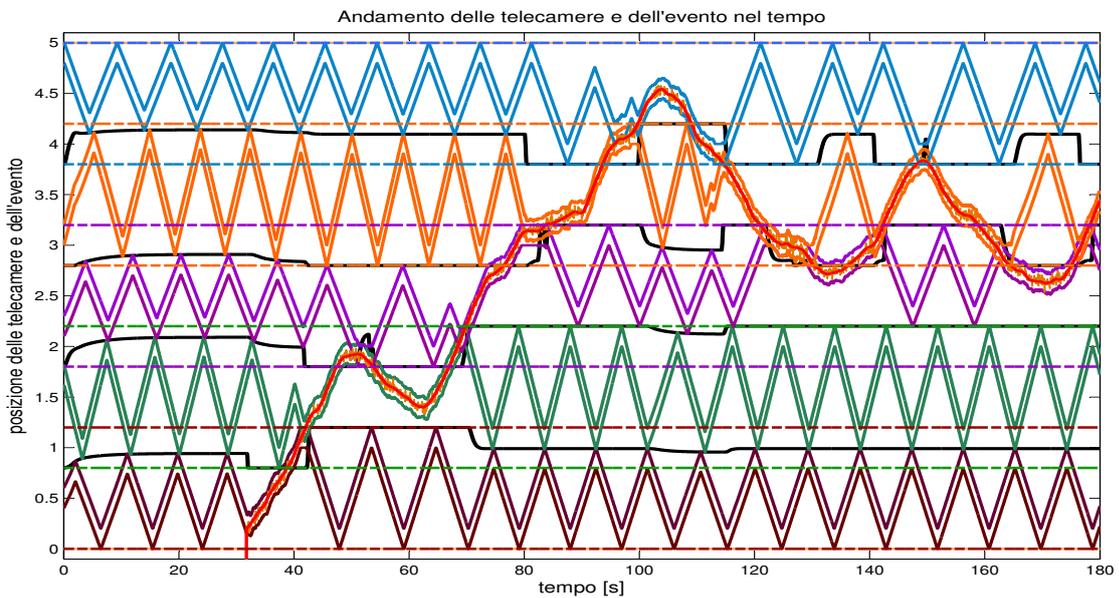
Figura 38: Alcuni casi di tracking

In figura 38b si è invece aumentata la varianza del rumore di velocità, imponendo  $b = 10$  e lasciando inalterati gli altri valori. Si nota come l'evento abbia dinamica molto variabile, tuttavia il filtro di Kalman non ha problemi a seguirlo: le camere impegnate nel tracking oscillano con l'evento, mantenendolo sempre al centro del loro field of view. Per un sistema reale, la trasmissione di tali vibrazioni alle camere potrebbe determinare un inaccettabile peggioramento della qualità video. A questo problema si

può porre facilmente rimedio introducendo una zona morta nel centro del fov per il tracking dell'evento, effettuando un filtraggio passa-basso sul comando di posizione alla telecamera, o variando i parametri del filtro di Kalman; questi accorgimenti non sono stati qui adottati per evitare di introdurre ritardo nella dinamica di tracking.



(a) Rumore di misura a varianza 0.03



(b) Rumore di misura a varianza 0.05

Figura 39: Tracking con rumore di misura elevato

Nella figura 39 è stata invece aumentata la varianza dell'errore di misura, ponendo  $d = 0.03$  e  $d = 0.05$ . Nel primo caso il sistema riesce comunque a seguire l'evento nonostante la misura di posizione (plottata per chiarezza) sia sostanzialmente diversa da quella reale. Nel secondo caso la differenza è troppo elevata (in certi punti l'escursione delle misure è pari al  $fov$ ) e l'evento viene perso in più di qualche punto.

Si mostra infine un caso di tracking con scarsa comunicazione tra le camere. I parametri sono quelli di figura 27 con in più l'introduzione della coordinazione in opposizione di fase. Le camere mediamente comunicano una volta al secondo. E' interessante notare come anche in queste condizioni gli accorgimenti adottati e la struttura del sistema di comunicazione assicurino una buona coordinazione globale delle camere nel tracking.

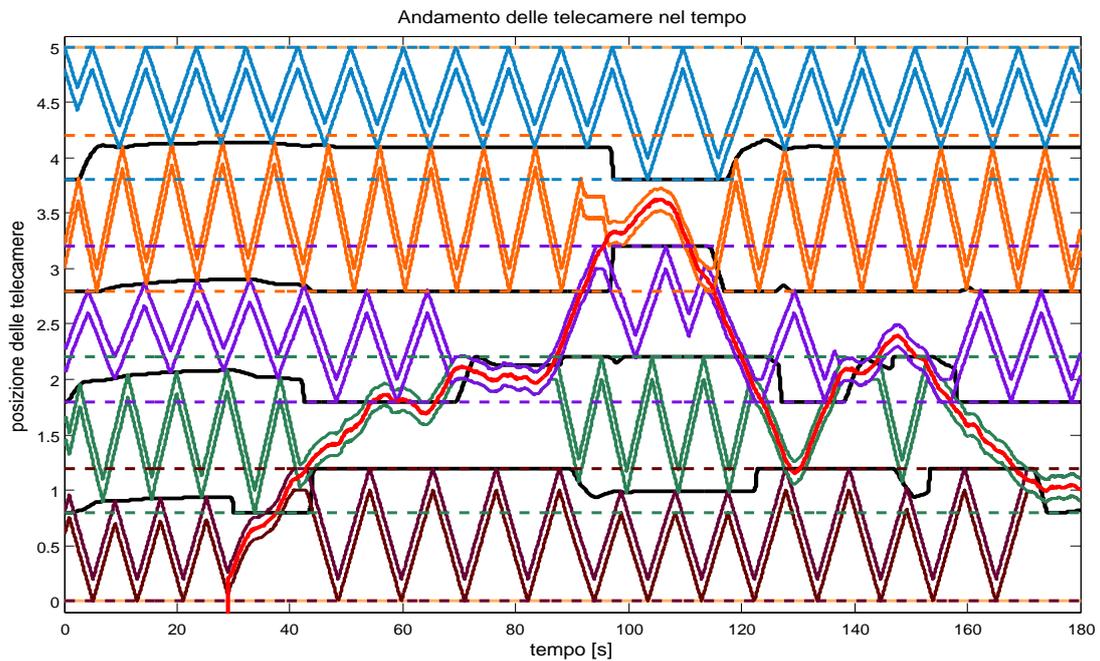


Figura 40: Tracking con poca comunicazione

## C L'approccio iniziale: i funzionali

Prima di affrontare il problema nella sua forma più generale (ipotesi iniziali riportate in sezione 3) si è cercato di affrontarne una versione semplificata, al fine di comprendere a fondo quelle che potevano essere le problematiche da affrontare ed avere una base implementativa da cui partire in seguito. Questo approccio iniziale è stato successivamente abbandonato a causa dei limiti realizzativi, delle difficoltà legate al raggiungimento del partizionamento ottimo vincolato e del fatto che, tramite l'approccio più profondo sviluppato nelle sezioni 3, 4, 5, si è reso obsoleto. Viene comunque riportato nel seguito dal momento che ha comportato impegno e ragionamento e, almeno per la parte del funzionale di posizione, si è rivelato utile nell'implementazione seguente.

Le ipotesi di quello che si può definire “caso base” sono:

1. telecamere aventi tutte la stessa velocità massima,  $V_{max}$ ;
2. zone di copertura costanti ed uguali per ogni telecamera ( $C_i = C_j = L$ ,  $\forall i, j$ );
3. zone di copertura disgiunte ( $C_i \cap C_j = \emptyset$ ).

I dati di cui si è fatto uso sono i seguenti:

- $C_i$  range di copertura delle telecamera  $i$ -esima (tutti lunghi  $L$ );
- $S_i(t)$  sezione vista dalla telecamera  $i$ -esima all'istante  $t$  ( $S_i \subseteq C_i$ );
- $pos(t)$  vettore che contiene la posizione di ciascuna telecamera all'istante  $t$ ;
- $pos(0)$  variabile aleatoria uniformemente distribuita tra 0 e 1000 per ogni telecamera;
- $V_{max} = 10$  (ad ogni istante la telecamera si sposta al più di 10 passi).

**Osservazione.** Il percorso da monitorare è stato discretizzato: ciascuna  $C_i$  ha una certa risoluzione (posta uguale a 1000); la posizione di ciascuna telecamera assume un valore intero compreso tra 1 e 1000 ed il range di visione (field of view,  $fov$ , lunghezza di  $S_i$ ) è stato posto pari a 100 (10 % del segmento  $C_i$  da esplorare); la velocità massima come scritto sopra vale 10.

**Obiettivo:** Cercare di coordinare le telecamere nella loro normale azione di patrolling, ottenendo la miglior copertura possibile in assenza di eventi.

**Idea della soluzione proposta** Si è pensato di adoperare due funzionali per descrivere la dinamica di ciascuna telecamera: uno che tenga conto del tempo trascorso dall'ultima visita di un punto da parte della telecamera stessa, l'altro che permetta di coordinare le telecamere secondo l'ottimo da noi presupposto:

$$1. J_{pos_i}(x, t) = \begin{cases} 0, & \text{se } x \in S_i \\ (t - t_{ultimaVisita})^2, & \text{altrimenti} \end{cases}$$

$$2. J_{coor_i}(x, t) = -k(x - pos_{i-1}(t))^2 - k(x - pos_{i+1}(t))^2$$

con  $x \in [1, \dots, 1000]$  intero.

Ad ogni istante si calcola  $J_i(x, t) = J_{pos_i}(x, t) + J_{coor_i}(x, t)$  e la posizione ottima per la telecamera  $i$ -esima all'istante  $t + 1$  sarà:

$$pos_i(t + 1) = \arg \max_x [J_i(x, t)]$$

con  $k$  parametro su cui tarare l'ampiezza del funzionale di coordinazione.

In questo modo si cerca di minimizzare il massimo assunto dal funzionale  $J_i$  ad ogni passo (si veda la formalizzazione più rigorosa nel seguito), scelta che sembra ragionevole e che porterà a regime le telecamere ad effettuare il patrolling alla velocità massima assegnata.

L'idea originale quindi era quella di coordinare le telecamere per la funzione di patrolling in modo tale che a regime esse fossero tutte a distanza pari a  $L$ , supponendo che questo potesse essere utile per un'eventuale generalizzazione futura in cui le velocità delle telecamere fossero diverse e si avessero anche zone di copertura sovrapposte (ipotesi 1 e 3 rilassate, mentre per come era stato concepito il funzionale di coordinazione il rilassamento dell'ipotesi 2 avrebbe probabilmente creato qualche problema in più).

**Scelta del funzionale di coordinazione** Per raggiungere l'obiettivo prefissato è stato creato il funzionale di coordinazione come la somma di due parabole con la concavità rivolta verso il basso il cui centro (e massimo) fosse rispettivamente la posizione della telecamera  $i - 1$  e  $i + 1$  (in questo modo lo spostamento ottimo suggerito ad  $i$  da entrambe le telecamere adiacenti è di trovarsi a distanza  $L$  da loro); la scelta della parabola è legata al fatto che il minimo (o massimo in questo caso) di una parabola è ben definito e la somma di due parabole con vertice sull'asse  $x$  è ancora una parabola con ascissa del vertice situata nella media delle due di partenza:

$$\begin{cases} y_1 = k(x - x_1)^2 \\ y_2 = k(x - x_2)^2 \end{cases} \Rightarrow y_3 = y_1 + y_2 = (2k)x^2 + (-2k(x_1 + x_2))x + (k(x_1^2 + x_2^2))$$

$y_1$  ha vertice  $(x_1, 0)$ ,  $y_2$  ha vertice  $(x_2, 0)$  mentre il vertice di  $y_3$  ha ascissa che vale proprio  $(x_1 + x_2)/2$ .

Grazie a questa proprietà non solo si è certi che il massimo di questo funzionale sia unico, ma anche che sotto l'azione del solo funzionale di coordinazione la posizione all'istante  $t + 1$  della telecamera  $i$ -esima tenderà (compatibilmente al vincolo di velocità

massima a cui è sottoposta) alla posizione intermedia suggerita dalle telecamere adiacenti: questo algoritmo è senz'altro convergente (e cioè porta le telecamere a disporsi a distanza  $L$  ciascuna dall'altra) perchè la prima e l'ultima hanno posizione che dipende solo da una telecamera adiacente; esse quindi dopo pochi passi si dispongono a distanza  $L$  (rispettivamente dalla seconda e dalla penultima) e poi a cascata in pochi passi anche quelle intermedie fanno altrettanto.

**Scelta del funzionale di patrolling** La scelta quadratica di  $J_{pos_i}(x, t)$  è legata al fatto che si desiderava ottenere due funzionali che fossero confrontabili per somma diretta.

**Scelta dell'ottimo all'istante  $t+1$**  Come detto precedentemente l'ottimo all'istante  $t+1$  per la telecamera  $i$ -esima è:

$$x_{ott_i}(t+1) = \arg \max_x [J_{pos_i}(x, t) + J_{coord_i}(x, t)] \quad (12)$$

Il massimo risultante dalla somma dei due è quindi funzione di entrambi i funzionali: a seconda di quale dei due prevalga, il sistema tende a coordinarsi oppure a monitorare il percorso.

Il grado di libertà su cui agire per fare in modo che nel primo transitorio prevalga il funzionale di coordinazione (per allineare le telecamere) ed a regime invece quello di patrolling è il parametro  $k$ .

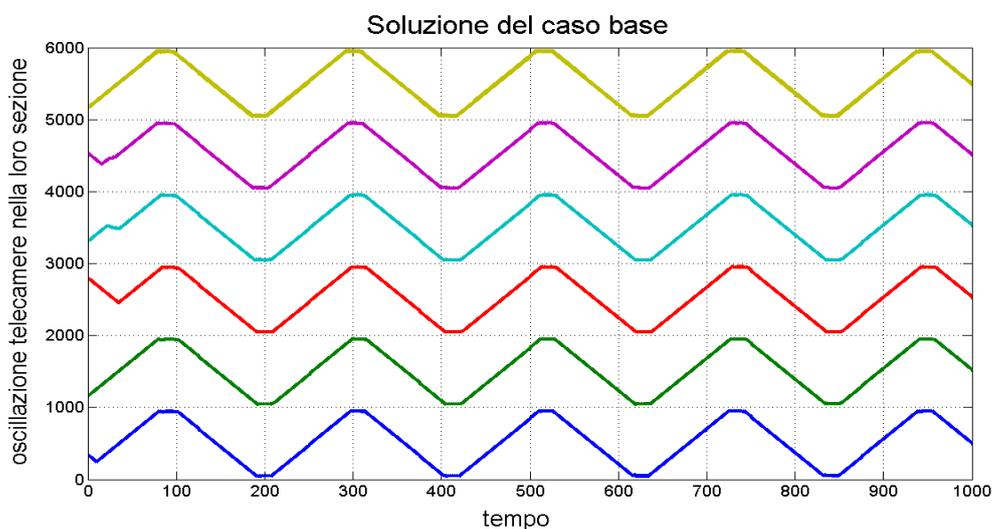


Figura 41: Esempio di simulazione del caso base

**Euristica del parametro  $k$ :** Valori di  $k$  troppo piccoli impediscono al sistema di coordinarsi, in quanto il funzionale di patrolling assume troppo presto valori dominanti rispetto a quello di coordinazione (il massimo assoluto cioè è imposto da  $J_{pos}(\cdot, \cdot)$ ); ma

valori di  $k$  troppo grandi provocano delle oscillazioni a regime intorno agli estremi della sezione da monitorare: ciò accade perchè dopo ogni spazzolata è necessario aspettare un  $\Delta T$  prima che il funzionale di patrolling torni a dominare su quello di coordinazione. Si intuisce il fatto che ci sia un trade off tra le due esigenze, con la speranza di poter determinare nel primo caso una condizione esplicita del tipo  $k > c_1$  (condizione 1), nel secondo invece del tipo  $k < c_2$  (condizione 2): se il  $c_1$  trovato fosse maggiore di  $c_2$ , significherebbe che la procedura ideata avrebbe il limite intrinseco di coordinare le telecamere a patto di produrre delle piccole oscillazione agli estremi durante il patrolling a regime. Tuttavia si riesce a produrre una relazione esplicita solo nel secondo caso: nel primo purtroppo il fatto che il funzionale di patrolling non abbia ancora una forma di ramo di parabola (si forma tale ramo solo dopo una spazzolata completa) rende molto difficile un'analisi analitica; tuttavia da uno studio accurato dell'algoritmo e dall'osservazione di molte simulazioni sembra proprio che “ $c_1$  sia maggiore di  $c_2$ ” e non esista quindi un valore limite che soddisfi entrambe le condizioni.

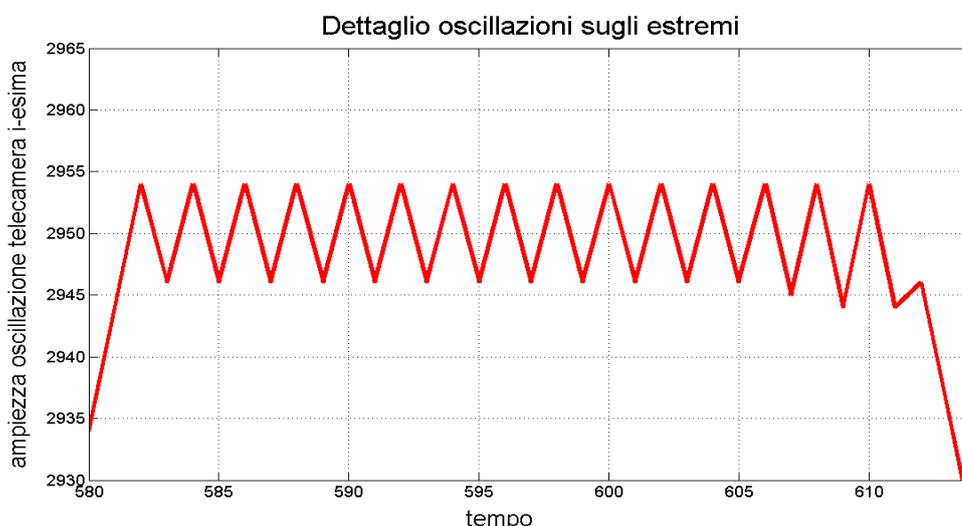


Figura 42: Dettaglio di oscillazione ad un estremo: caso con  $k$  troppo elevato

**Analisi della condizione 2:** A regime il problema delle oscillazioni si crea agli estremi. Come detto questo è dovuto al fatto che il massimo del potenziale di coordinazione “vince” su quello di patrolling per un  $\Delta T$ .

Si supponga di trovarsi nell'estremità destra di pattugliamento in posizione  $x = 950$ . Il periodo ottimo a regime in assenza di oscillazioni vale:

$$T = 2(900/V_{max}) = 2(900/10) = 180$$

Affinchè ciò si verifichi è necessario che:

$$J_i(50, t) = (T/2)^2 - 2k(50 - 950)^2 \geq 0$$

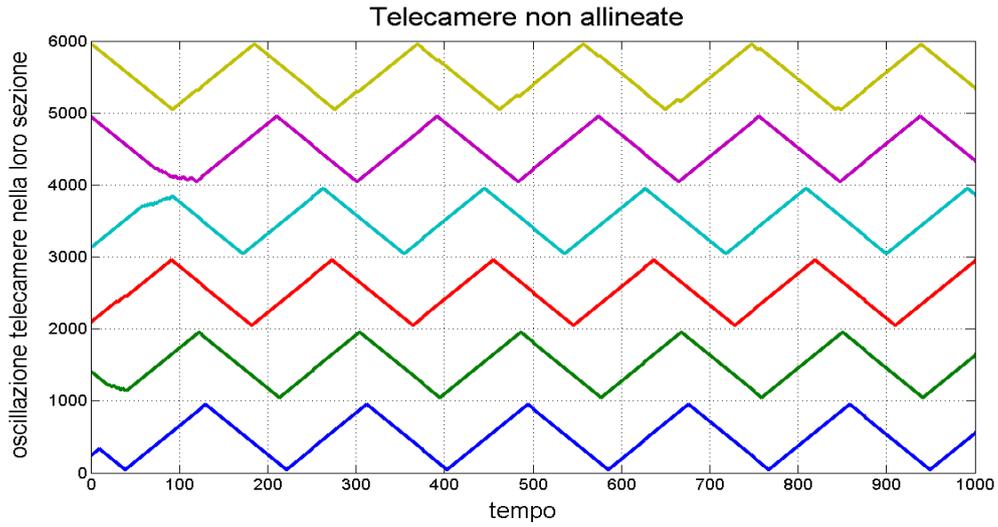


Figura 43: Telecamere non allineate: caso con  $k$  troppo piccolo

In questo modo il massimo all'istante  $t+1$  sarà proprio dalla parte opposta (in posizione 50) e questa è la condizione che a regime fa spostare il sistema. Risolvendo risulta:

$$k \leq 1/200 = 0.005$$

valore che si riscontra precisamente nelle simulazioni.

## D Codice Matlab

### D.1 Dati.m

```
% estremi fisici del percorso totale
Linf=0;
Lsup=5;
T=120;
Ts=0.05;

p_com=0.4; % probabilita' di comunicazione agli estremi

vmax=[0.18 0.23 0.15 0.25 0.16]; % m/s

fovi=[0.2 0.2 0.2 0.2 0.2];

vimax=vmax*Ts; % distanza da coprire in un istante di campionamento

t=(Ts:Ts:T)';

tcom=zeros(T/Ts,4);

for i=1:T/Ts
    for n=1:4
        num = rand(1);
        if num<p_com, tcom(i,n)=1; end
    end
end

tcom=[t tcom];

coeff_fase=0.5; % 0.5 ok, convergono abbastanza alla fase in 3-4 periodi
                % Impostare a zero per disattivare messa in fase

fase_mode=-1; % 1=in fase, -1=opp fase.

FdT_chiamata=tf([0.05 0],[1 -0.95],Ts); % gain statico 1
% filtro la velocità in uscita da kalman
% per calcolare e passare all'altra camera il
% tempo di arrivo dell'evento. Bilanciare il gain statico e la posizione
% del polo reale, che deve comunque essere vicino a uno.

% varianza rumore sul calcolo dell'estremo nelle camere. Permette di vedere
% il modo in cui gli estremi si riportano al consenso se la modellizzazione
% dell'estremo tra 2 camere dovesse per qualche motivo andare a differenziarsi.
noise_estremo=0%.00001;

% differenza tollerata tra le 2 versioni degli estremi prima di resettare
% l'integratore
tolleranza_estremo=0.001;

m=2;
% kg. La massa ottima e' 0.1kg con Ts=0.025. Provo a aumentarla x rallentare la
```

```

% convergenza e vedere influenza andamento bordi sul zig-zag della
% telecamera

ki=1./(vmax); % N/m (Delta-l in metri)
psi=1; % damping factor. Ottimo=2. Sottosmorzo x vedere effetto oscillazioni
% psi = 2/radice di 2 dà massima velocità di salita con oscillaz
% psi = 2 dà massima velocità senza oscill

% aumento la forza tra gli estremi quando faccio tracking. Calcolo la forza
% media che mi aspetto interessare gli estremi, e sommo un offset pari al
% quadruplo di tale forza quando faccio tracking
Fopt=(Lsup-Linf)/sum(vmax);
bias_estr_track=4*Fopt;

for i=1:numel(ki)-1
    kiestr(i)=ki(i)+ki(i+1)
end
% l'estremo ha due molle attaccate, quindi considero una kiestr equivalente
% pari alla somma delle due costanti elastiche. Questa kiestr equivalente
% mi serve solo per calcolare il coefficiente di attrito ni ottimo per
% l'estremo

ni=psi*2*sqrt(kiestr*m); % ATTENZIONE, MODIFICARE psi NON QUESTO
% coeff di attrito viscoso Ns/m, tale da assicurare
% damping factor=psi per il singolo oscillatore armonico
% smorzato

% Vincoli superiori e inferiori in coordinate assolute alle variazioni
% degli estremi aestr delle sezioni.

vincolo_inf=[0.8 1.8 2.8 3.8];
vincolo_sup=[1.2 2.2 3.2 4.2];

x0cam=[0.5 1.7 2.2 3.1 4.9] % posizione iniziali telecamere

% Generazione evento
velMin=min(vmax); %indipendente dal Ts

% EVENTO OFF
% statoEvento = ones(2,T/Ts+1)*(-1);
% misureEvento = ones(1,T/Ts)*(-1);

% EVENTO ON
[statoEvento, misureEvento] = Evento(Lsup-Linf,T,Ts,velMin);
plot(misureEvento)

limiteinf = [Linf vincolo_inf];
limitesup = [Lsup vincolo_sup];
limiti = [limiteinf limitesup];

```

## D.2 Funzione Evento.m

```
function [statoEvento, misureEvento] = Evento(Ltot,Ttot,Tsample,velMin)

global a;
global b;
global d;

Ttot=floor(Ttot/Tsample);
velMin=velMin*Tsample;

x = ones(2,Ttot+1)*(-1);
u = zeros(2,Ttot);
y = ones(1,Ttot)*(-1);
posEvento = ones(1,Ttot+1)*(-1);
velEvento = ones(1,Ttot+1)*(-1);
x = [posEvento;velEvento];
a = 0.999;
b = 0.0005; % 0.4 ok...
d = 0.002;
n1 = randn(1,Ttot);
n2 = randn(1,Ttot);
u = [n1 ; n2];
A = [1 1 ; 0 a];
B = [0 0; b 0];
C = [1 0];
D = [0 d];
Q = B*B';
R = D*D';

startEvento = floor(rand(1)*Ttot/4)+1;
% posEvento(startEvento) = rand(1)*Ltot;
posEvento(startEvento) = 0.2; % forzo cond iniz
velEvento(startEvento) = velMin/2;
x(:,startEvento) = [posEvento(startEvento);velEvento(startEvento)];
y(startEvento) = C*x(:,startEvento) + D*u(:,startEvento);

stopEvento=0;
for (t=startEvento:Ttot)
    if stopEvento==0
        x(:,t+1) = A*x(:,t) + B*u(:,t); % dinamica di stato
        y(t) = C*x(:,t) + D*u(:,t); % dinamica di stato
        if (x(1,t+1)>Ltot), stopEvento=1; end
        if (x(1,t+1)<0), stopEvento=-1; end
    else
        if stopEvento==-1
            x(:,t+1) = [-1;0];
            y(t)=-1;
        else
            x(:,t+1) = [6;0];
            y(t)=6;
        end
    end
end
```

```

end
if abs(x(2,t+1))>velMin, x(2,t+1)=velMin*sign(x(2,t+1)); % limito la vel
end

statoEvento = x(:,2:Ttot+1); %INVECE DI X LEVO L PRIMA COLONNA
misureEvento = y;

end

```

### D.3 StimaTempoConvergenza.m

```

% matrice F a tempo continuo per N = 5 camere
F_1i = [0 1 0 0 0 0 0 0];
F_2i = [(-(1/m)*(1/vmax(1)+1/vmax(2))) (-ni(1)) (1/(m*vmax(2))) 0 0 0 0 0];
F_3i = [0 0 0 1 0 0 0 0];
F_4i = [(1/(m*vmax(2))) 0 (-(1/m)*(1/vmax(2)+1/vmax(3))) ...
        (-ni(2)) (1/(m*vmax(3))) 0 0 0];
F_5i = [0 0 0 0 0 1 0 0];
F_6i = [0 0 (1/(m*vmax(3))) 0 (-(1/m)*(1/vmax(3)+1/vmax(4)))...
        (-ni(3)) (1/(m*vmax(4))) 0];
F_7i = [0 0 0 0 0 0 0 1];
F_8i = [0 0 0 0 (1/(m*vmax(4))) 0 (-(1/m)*(1/vmax(4)+1/vmax(5))) (-ni(4))];

F = [F_1i; F_2i; F_3i; F_4i; F_5i; F_6i; F_7i; F_8i];
eig(F)
Lambda_max = max(eig(F))
t_ass = log(0.01)/Lambda_max

```

## E Schemi simulink

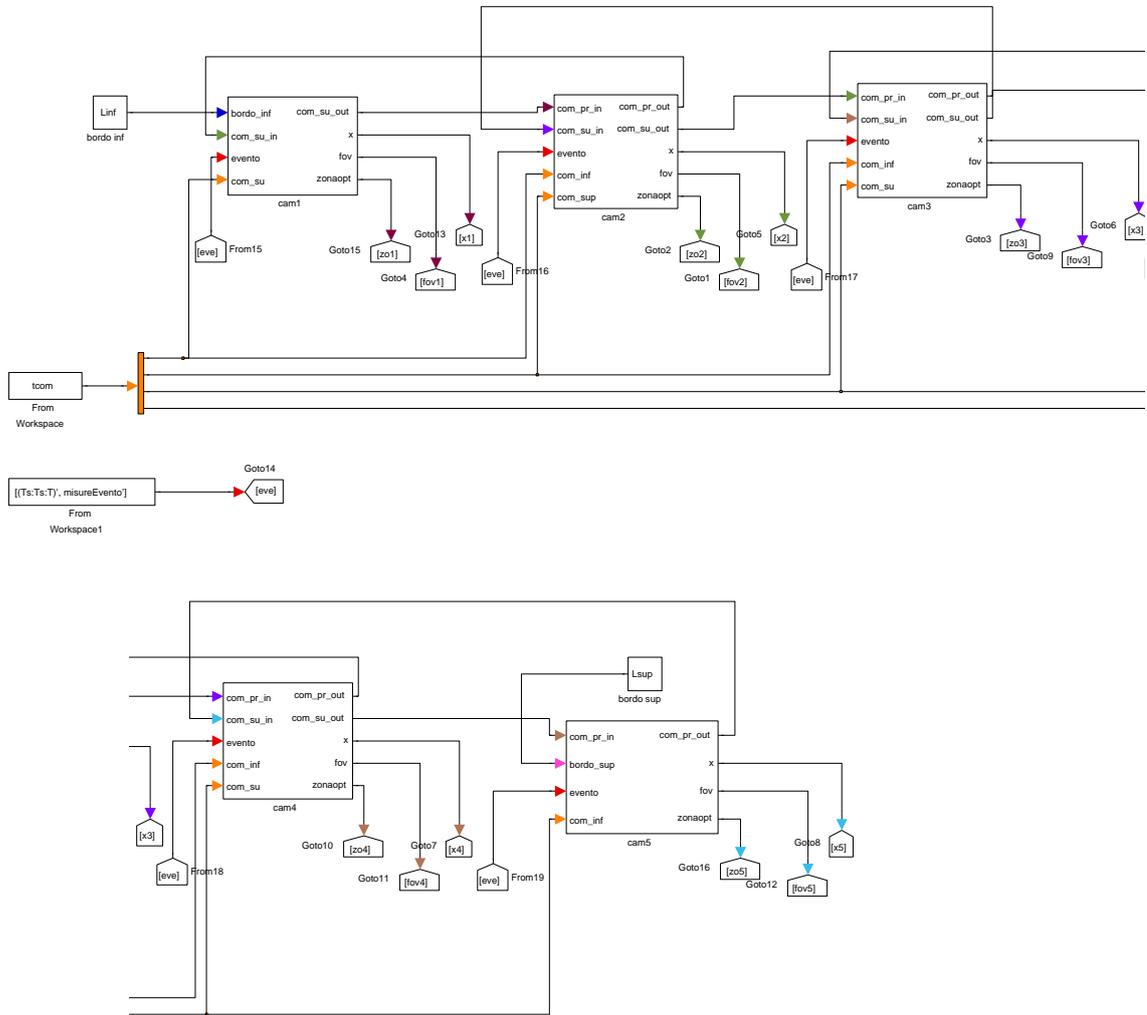


Figura 44: Schema globale della struttura a blocchi con 5 camere, e bus di comunicazione

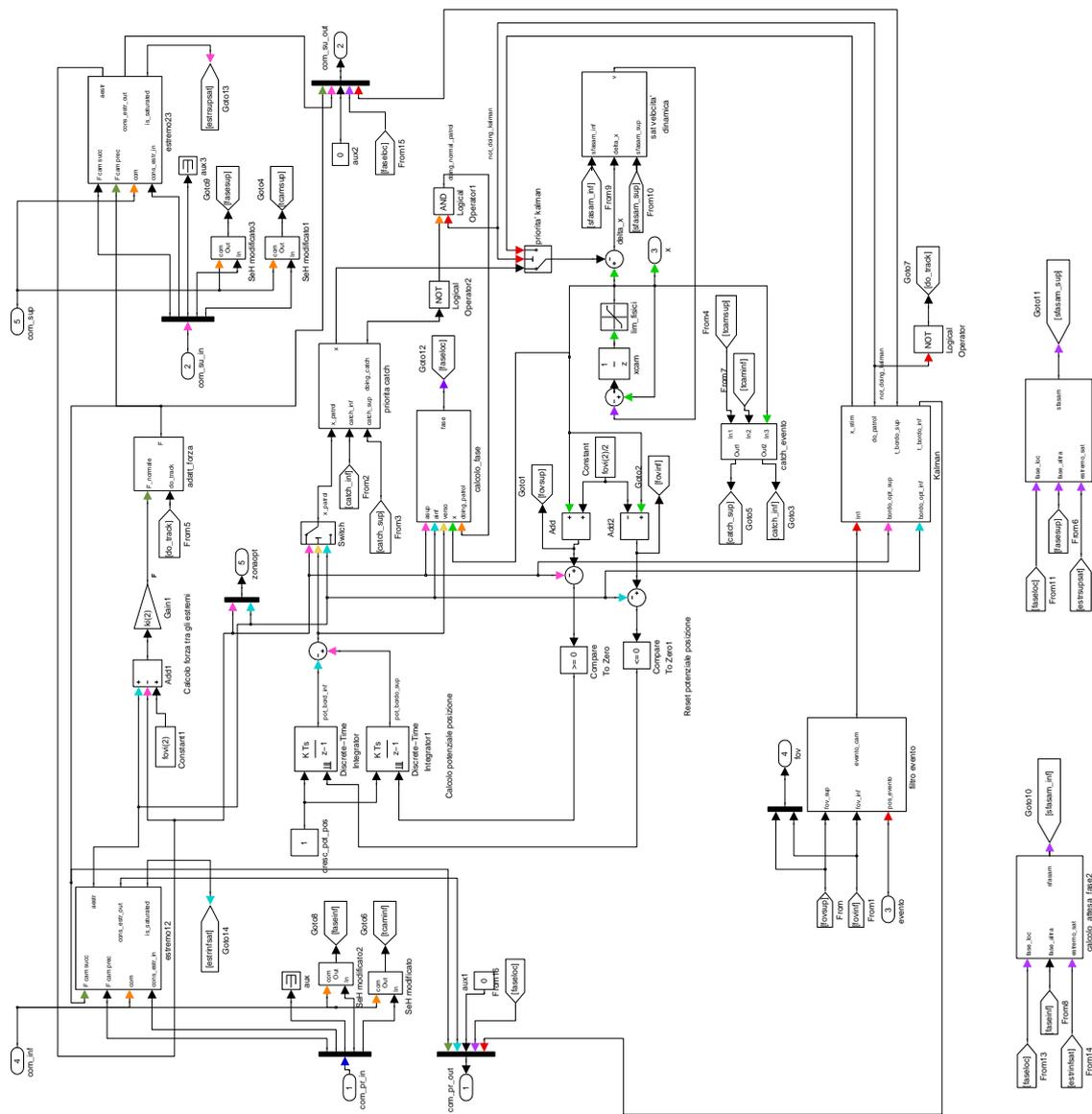


Figura 45: Schema interno di una telecamera

Si è qui proposta una vista globale degli schemi simulink del sistema complessivo composto di 5 camere e dello schema interno relativo a una camera. Le immagini sono di tipo vettoriale e possono essere apprezzate nel dettaglio nel PDF di questa relazione. Ai lettori della copia cartacea si consiglia invece di riferirsi alla sezione 6 dove i sottosistemi sono riportati a una scala visibile dalla stampa e viene analizzato nel dettaglio il loro ruolo.

## F Rendering 3D delle simulazioni

I video mostrati durante la presentazione di questo progetto sono stati realizzati con l'impiego della toolbox *Simulink 3D*, compresa nella suite utilizzata per le simulazioni. Questo strumento consente di interfacciare con facilità i segnali ottenuti dalle simulazioni con un particolare *sink* che li traduce in proprietà di oggetti tridimensionali. Il linguaggio utilizzato per disegnare il mondo 3D è *VRML97*, che consente di definire forme primitive quali cilindri, sfere e parallelepipedi; tali forme costituiscono la struttura essenziale dell'ambiente grafico, e l'animazione è stata ottenuta sfruttando la possibilità di linkare le proprietà di posizione, scala e rotazione degli oggetti 3D ai segnali provenienti dalle simulazioni. Questo ha consentito di apprezzare il comportamento "realistico" del sistema implementato, e ha consentito di individuare proprietà (ed errori) non facilmente visibili dai plot bidimensionali.

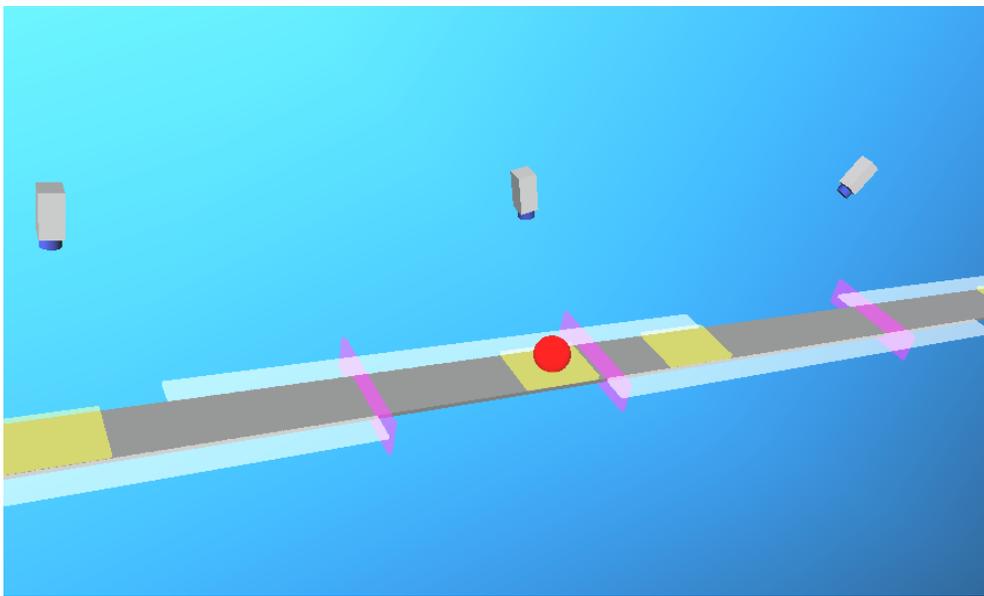


Figura 46: Un fotogramma della scena 3D

## Riferimenti bibliografici

- [1] Yann Chevaleyre, *Theoretical Analysis of the Multi-agent Patrolling Problem*, Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)
- [2] Yotam Elor and Alfred M. Bruckstein, *Multi-A(ge)nt Graph Patrolling and Partitioning*, 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops
- [3] Cash J. Costello and I-Jeng Wang, *Surveillance Camera Coordination Through Distributed Scheduling*, Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 Seville, Spain, December 12-15, 2005
- [4] Ugur Murat Erdem, Stan Sclaroff, *Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements*, Computer Vision and Image Understanding 103 (2006) 156-169
- [5] Derek Kingston, Ryan Holt, Randal Beardy, Timothy McLain, and David Casbeer, *Decentralized Perimeter Surveillance Using a Team of UAVs*, Brigham Young University, Provo, UT, 84602
- [6] Justin Clark and Rafael Fierro, *Cooperative Hybrid Control of Robotic Sensors for Perimeter Detection and Tracking*, 2005 American Control Conference June 8-10, 2005. Portland, OR, USA
- [7] Islam I. Hussein, Dusan M. Stipanovi´c, *Effective Coverage Control using Dynamic Sensor Networks*, Proceedings of the 45th IEEE Conference on Decision & Control Manchester Grand Hyatt Hotel San Diego, CA, USA, December 13-15, 2006
- [8] Islam I. Hussein, Dusan M. Stipanovi´c, *Effective Coverage Control using Dynamic Sensor Networks with Flocking and Guaranteed Collision Avoidance*, Proceedings of the 2007 American Control Conference Marriott Marquis Hotel at Times Square New York City, USA, July 11-13, 2007
- [9] C. Micheloni, G. L. Foresti and L. Snidaro, *A cooperative multicamera system for video-surveillance of parking lots*, Proceedings of the IEE Symposium on Intelligence Distributed Surveillance Systems, 26 Feb. 2003
- [10] Ruoting Yang, Sharon Bewick, Mingjun Zhan, William R. Hamel, Tzyh-Jong Tarn *Adaptive immune system inspired perimeter patrol control strategy*, Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, October 11-15, 2009
- [11] Hugo Dias, Joao Rocha, Paulo Silva, Carlos Leao and Luis Paulo Reis *Distributed Surveillance System*, Proceeding of the Portuguese conference on Artificial intelligence, Covilha, Portugal, December 5-8, 2005

- [12] M. Valera and S.A. Velastin *Intelligent distributed surveillance systems: a review*, Vision, Image and Signal Processing, IEE Proceedings - Volume 152, Issue 2, 8 April 2005 Page(s): 192 - 204
- [13] James Guo Ming Fu and Marcelo H. Ang Jr *Probabilistic Ants (PAnts) in Multi-Agent Patrolling*, Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Suntec Convention and Exhibition Center, Singapore, July 14-17, 2009
- [14] D. M. G. Knight *The Legal Implications of Intelligent Distributed Surveillance Systems*, IEE Symposium on Intelligence Distributed Surveillance Systems, 26 Feb. 2003 Pages: 2/1 - 2/5
- [15] Jorge Cortes, Sonia Martinez, Timur Karatas, Francesco Bullo *Coverage control for mobile sensing networks*, IEEE Transactions on Robotics and Automation, April 2004, Volume 20, on page(s): 243- 255
- [16] Christos G. Cassandras and Wei Li *Sensor Networks and Cooperative Control*, 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 12-15 Dec. 2005
- [17] Wei Li and Christos G. Cassandras *Distributed Cooperative Coverage Control of Sensor Networks*, 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 12-15 Dec. 2005
- [18] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B. Srivastava *Coverage Problems in Wireless Ad-hoc Sensor Networks* , Proceedings of the INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. 2001, Anchorage, AK, USA
- [19] Chao Gui and Prasant Mohapatra *Virtual Patrol: A New Power Conservation Design for Surveillance Using Sensor Networks*, Fourth International Symposium on Information Processing in Sensor Networks, 2005. 15 April 2005
- [20] S. Suryadarma, T. Adiono, C. Machbub, T. L. R. Mengko *Camera Object Tracking System*, Proceeding of the International Conference on Information, Communication and Signal Processing, ICICS '97, Singapore, 9-12 September 1997
- [21] Alan T. Murray , Kamyoun Kim , James W. Davis, Raghu Machiraju, Richard Parent *Coverage optimization to support security monitoring*, Computers Environment and Urban Systems (2007). Volume: 31, Issue: 2, Pages: 133-147
- [22] Elon Rimon, Daniel E. Kodischek *Exact Robot Navigation Using Artificial Potential Functions*, IEEE Transation on Robotics and Automation, Vol. 8, No. 5, October 1992

- [23] David W. Casbeer, Derek B. Kingston, Randal W. Beard, Timothy W. McLain  
*Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles*, International Journal of Systems Science, Vol. 00, No. 00, January 2005, 1-18