

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di Laurea in Ingegneria dell'Automazione

**TEORIA DEL CONSENSO E
APPLICAZIONE AL PROBLEMA DEL
COORDINAMENTO DEL MOTO DI
ROBOT**

Tesi di Laurea in Controlli Automatici

Relatore:

**Chiar.mo Prof.
Lorenzo Marconi**

Presentata da:

Luca Furieri

II Sessione

Anno Accademico 2013-2014

*“A genuine leader is not a searcher for consensus,
but a molder of consensus.”
- Martin Luther King, Jr.*

Introduzione

Il sempre crescente numero di applicazioni di reti di sensori, robot cooperanti e formazioni di veicoli, ha fatto sì che le problematiche legate al coordinamento di gruppo diventassero tra le più studiate nell'ambito della teoria dei controlli. La sfida principale in tali applicazioni è quella di raggiungere un comportamento di gruppo prestabilito utilizzando algoritmi decentralizzati e locali, piuttosto che dei controllori centralizzati ai quali riferirsi. Molto simile al problema del coordinamento è quello dell'*agreement*, o *consensus*: l'obiettivo è quello di portare variabili di interesse ad un valore comune. Risolvendo tale problema, diventa possibile pensare ad un numero estremamente vasto di applicazioni, spaziando ad esempio dallo studio dei comportamenti di gruppo nei branchi e negli stormi di organismi viventi alla computazione distribuita o alla sorveglianza di foreste per prevenire incendi. In questo lavoro, il focus applicativo è posto sul controllo di formazione di un network di veicoli, le cui traiettorie di riferimento vengono generate utilizzando algoritmi locali di coordinamento.

La strategia che verrà studiata ha una natura distribuita: non è prevista la presenza di un leader o di un coordinatore esterno di altra natura. Piuttosto, la formazione raggiunta è ottenuta come conseguenza dell'interazione locale tra i singoli veicoli.

I sistemi dinamici, che possono essere di qualsiasi natura (biologica, fisica, ecc...) vengono solitamente rappresentati mediante i cosiddetti diagrammi a blocchi: il sistema è cioè raffigurato come un box che riceve delle variabili in ingresso e ne produce altre in uscita. Tuttavia, nel momento in cui pensiamo ad un insieme di sistemi dinamici comunicanti ed interagenti tra di loro, si viene a creare una nuova struttura più complessa,

chiamata rete di sistemi dinamici o semplicemente “gruppo”, in cui ogni sistema dinamico è rappresentato da un nodo o agente, ed esistono dei collegamenti tra i vari nodi che permettono ad un agente di interagire con i suoi vicini nella rete. Nell’ottica di una rete di tal genere, assumono grande importanza i problemi di *consenso* e *sincronizzazione*. In una rete di agenti, *consensus* indica il raggiungimento di un *agreement* relativamente ad una certa quantità di interesse che dipende dallo stato di tutti gli agenti. Un algoritmo (o protocollo) di consenso è una legge di interazione che regola lo scambio di informazioni tra un agente ed i suoi vicini nella rete al fine di raggiungere un accordo complessivo.

Struttura e finalità della tesi.

Lo scopo dell’elaborato è duplice. Da una parte si vuole fornire una trattazione teorica del problema del consenso e della sincronizzazione su sistemi ad una o più variabili di stato. Dall’altra si vogliono utilizzare tali risultati teorici per un’attività in ambiente Matlab/Simulink, con il fine applicativo di generare le traiettorie di riferimento per un gruppo di quadricotteri in formazione. Alla stesura di questo elaborato è stata affiancata l’attività di laboratorio presso il “Center for Research on Complex Automated Systems” (CASYS-DEI Università di Bologna).

Il capitolo 1 presenta tutti gli strumenti matematici e teorici che saranno ripresi e sfruttati nei capitoli seguenti. In particolare, si introducono:

- Teoria dei Grafi.
- Forma e trasformazione di Jordan.
- Il prodotto di Kronecker.
- Formalizzazione del problema del Consenso.
- Modelli nello spazio degli stati e cambi di variabile.

Nel capitolo 2, viene mostrato il comportamento di una rete di integratori connessi tra di loro e viene analizzato sotto quali condizioni il consenso viene raggiunto. Per verificare i risultati noti in letteratura, sono state svolte simulazioni in ambiente Matlab Simulink: i casi presi in esame comprendevano sia condizioni di topologie di comunicazioni fisse che casi di topologie tempo varianti.

Il capitolo 3 è dedicato ai sistemi multivariabile e alla metodologia che permette di ottenere la loro sincronizzazione. Utilizzando gli strumenti matematici introdotti nel capitolo 1, si sviluppa ulteriormente la teoria mostrando formalmente la validità delle procedure utilizzate. Il capitolo è concluso dall'applicazione al caso di un network di oscillatori armonici e da una simulazione dimostrativa su Matlab/Simulink.

Il capitolo 4 entra nel dettaglio del modello simulativo per la generazione di traiettorie, che serviranno come generatori di riferimento per un gruppo di quadricotteri. Sarà possibile impostare raggio e coordinate del centro della traiettoria target, simulando un tipico scenario di *loitering* di un sistema multi-robot. La teoria del consenso sia su sistemi integratori che oscillatori sarà sfruttata per ottenere questo risultato.

Il capitolo 5 descrive brevemente il setup di laboratorio utilizzato per realizzare il sistema di coordinamento di un gruppo di quadricotteri CrazyFlie, implementando a tale scopo il modello simulativo ottenuto nel capitolo 4. Verranno mostrati i risultati della prova sperimentale sotto forma di plot delle grandezze di interesse.

Indice

Introduzione	i
1 Teoria e strumenti matematici.	1
1.1 Teoria dei Grafi: alcune definizioni.	1
1.1.1 Descrizione tramite matrici rappresentative.	3
1.1.2 Grafi variabili nel tempo.	4
1.1.3 Componenti indipendenti di un grafo.	4
1.2 Il prodotto di Kronecker.	5
1.2.1 Proprietà del prodotto di Kronecker	6
1.3 Modelli nello spazio degli stati.	7
1.4 Rappresentazioni equivalenti e cambio di variabile.	8
1.4.1 Matrici diagonali e trasformazione di Jordan.	9
1.4.2 Forma canonica di Jordan.	10
1.5 La definizione di consenso	11
2 L’algoritmo di consenso e simulazioni su sistemi Integratori.	13
2.1 Il protocollo di consenso.	13
2.1.1 Analisi spettrale della Laplaciana.	14
2.2 Altri modi di analizzare la convergenza degli algoritmi di consenso.	16
2.2.1 Analisi di stabilità tramite funzioni di Lyapunov.	16
2.2.2 La proprietà di contrazione	19
2.3 Il valore di convergenza: l’ <i>average-consensus problem</i> .	20
2.4 Simulazioni su sistemi integratori.	23

2.4.1	Simulazione 1: una prima applicazione.	23
2.4.2	Simulazione 2: Grafi sconnessi variabili nel tempo.	26
2.4.3	Simulazione 3: Grafi complessi con componenti residuali.	30
3	Sistemi Multiagente Multivariabile.	33
3.1	L'oscillatore lineare come sistema.	34
3.2	Analisi del problema multivariabile.	35
3.2.1	L' <i>output consensus</i>	36
3.3	La convergenza del protocollo di consenso su sistemi multivariabile.	37
3.3.1	Trattazione attraverso una qualunque "Trasformazione T su L"	38
3.3.2	Sintesi della matrice K	41
3.4	Trattazione attraverso trasformata di Jordan T_J	43
3.5	Analisi attraverso trasformata di Jordan su un caso specifico.	44
3.5.1	Simulazione: output consensus su 3 oscillatori armonici.	47
4	Generazione di traiettorie circolari per il <i>loitering</i> di più robot.	49
4.1	Consenso degli oscillatori.	51
4.1.1	Sfasamento degli agenti.	53
4.2	Consenso degli integratori.	54
4.3	Simulazione della generazione di traiettorie.	55
5	Il setup di laboratorio e lo schema della Ground Station.	57
5.1	La Ground Station	60
5.2	I risultati di laboratorio.	62
	Conclusioni	65
	Bibliografia	67

Elenco delle figure

2.1	Schema Simulink	24
2.2	Grafo Connesso	25
2.3	Grafo Sconnesso	27
2.4	Modello Simulink	28
2.5	Risultato della simulazione	29
2.6	Risultato della simulazione.	31
3.1	Risultato della simulazione	48
4.1	Modello: alto livello.	51
4.2	Traiettorie prima dello sfasamento	54
4.3	Traiettorie dopo lo sfasamento.	55
4.4	Traiettorie X e Y con cambio di richiesta sul centro ogni 15 secondi.	56
5.1	Quadrirotori CrazyFlie.	58
5.2	I 3 marker e la struttura in carbonio.	59
5.3	Optitrack.	59
5.4	Periferica di trasmissione dati al nano-quad.	60
5.5	Schema della Ground Station	61
5.6	Riferimento per la X, prima e dopo sfasamento.	63
5.7	Riferimento per la Y, prima e dopo sfasamento.	63
5.8	Tracking della traiettoria di riferimento per un quadrirotore.	64

Capitolo 1

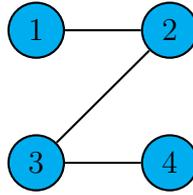
Teoria e strumenti matematici.

Nello studio di problemi di *coordinamento e sincronizzazione* di Sistemi Multiagente (MAS), la comunicazione tra agenti assume un ruolo fondamentale. Il sistema dinamico complessivo infatti, è il risultato dell'interconnessione di un certo numero di agenti, che comunicano tra loro secondo una precisa *topologia di comunicazione*.

Supponiamo dunque che più sistemi interagiscano l'uno con l'altro tramite una *rete di comunicazione* o una *rete di sensing*, oppure con una combinazione di entrambe: appare naturale modellare queste interazioni tra i sistemi mediante grafi diretti o indiretti.

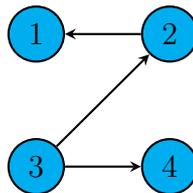
1.1 Teoria dei Grafi: alcune definizioni.

Un *grafo non diretto* \mathcal{G} di ordine n è una coppia (ν, ϵ) , dove $\nu = \{1, \dots, n\}$ è un insieme di nodi finito e non vuoto, mentre $\epsilon \subseteq \nu \times \nu$ è un *insieme di archi* corrispondenti a coppie **non ordinate** di nodi. Stabilita la coppia (ν, ϵ) , il grafo *non diretto* $\mathcal{G} = (\nu, \epsilon)$ risulta completamente definito. Se $(i, j) \in \epsilon$ oppure $(j, i) \in \epsilon$, significa che gli agenti i e j possono scambiarsi vicendevolmente informazioni. Un esempio di grafo non diretto è il seguente



dove $\nu = \{1, 2, 3, 4\}$ e $\epsilon = \{(1, 2), (2, 3), (3, 4)\}$. Si noti che le coppie di nodi non sono ordinate.

Si parla invece di *grafo diretto* nel caso in cui ϵ sia formato da coppie **ordinate** di nodi. In tal caso, la presenza dell'arco $(i, j) \in \epsilon$ denota che il sistema i può ricevere informazioni dal sistema j , ma non viceversa. Quello che segue è un esempio di grafo diretto.



dove $\nu = \{1, 2, 3, 4\}$ e $\epsilon = \{(2, 1), (3, 2), (3, 4)\}$ e le coppie di nodi sono ordinate. Archi che collegano un nodo con se stesso non sono ammessi, cioè $\forall i \in \nu, (i, i) \notin \epsilon$. Considereremo nel resto della tesi che \mathcal{G} sia un grafo **diretto**, se non esplicitato altrimenti.

In un grafo diretto, è possibile introdurre il concetto di *percorso diretto*, inteso come una sequenza ordinata di m archi del tipo $[(i_1, i_2), (i_2, i_3), \dots, (i_{m-1}, i_m)]$. Intenderemo per *ciclo* un percorso diretto che comincia e termina sullo stesso nodo.

Definizione 1 (Grafo connesso semplicemente e fortemente.). *Un grafo diretto si dice connesso se esiste un nodo dal quale è possibile raggiungere tutti gli altri nodi attraverso un percorso diretto: tale nodo viene detto centroide. Un grafo i cui nodi sono tutti dei centroidi viene detto fortemente connesso.*

Definizione 2 (Grafo connesso debolmente.). *Un grafo diretto si dice debolmente connesso se il grafo non diretto sottostante risulta connesso.*

1.1.1 Descrizione tramite matrici rappresentative.

È opportuno trovare una descrizione algebrica univoca per rappresentare i nodi e gli archi di un dato grafo diretto. A tale scopo introduciamo le seguenti *matrici rappresentative*:

- **Matrice di Adiacenza** $A \in \mathbb{R}^{n \times n}$. L'elemento a_{ij} è una quantità positiva se e solo se $(j, i) \in \epsilon$, altrimenti ha valore nullo. Considereremo solamente *grafi non pesati*, ovvero tale quantità positiva sarà sempre uguale a 1.

Definiamo *grado in entrata* del nodo i come $\sum_{j=1}^n a_{ij}$ e *grado in uscita* del nodo i come $\sum_{j=1}^n a_{ji}$: il nodo i è bilanciato se $\sum_{j=1}^n a_{ij} = \sum_{j=1}^n a_{ji}$.

- **Matrice Laplaciana diretta** $L \in \mathbb{R}^{n \times n}$. Gli elementi presenti sulla diagonale sono definiti come $l_{ii} = \sum_{j=1, j \neq i}^n a_{ij}$, mentre per gli altri elementi vale $l_{ij} = -a_{ij}$.

La proprietà di connessione.

Definite le matrici rappresentative, è possibile stabilire se un grafo diretto è connesso attraverso l'analisi degli autovalori della matrice Laplaciana associate a tale grafo. Vale infatti il seguente teorema:

Teorema 1. *Supponiamo che $\mathcal{G} = (\nu, \epsilon)$ sia un grafo diretto tale che ogni elemento della matrice di adiacenza A associata appartenga all'insieme $\{0\} \cup \{\alpha, +\infty\}$ per un certo $\alpha > 0$. Allora \mathcal{G} è connesso se e solo se il secondo autovalore della Laplaciana associata L ha parte reale positiva, cioè se $\Re(\lambda_2(L)) \geq 0$, dove per convenzione il secondo autovalore di L è quello con la seconda parte reale più piccola.*

1.1.2 Grafi variabili nel tempo.

Le definizioni finora date sono facilmente estendibili a grafi con topologia di connessione variabile nel tempo. Anche le matrici rappresentative $A(t), L(t)$ saranno dunque dipendenti dal tempo: questa osservazione ci porta a considerare la *matrice di adiacenza media nel periodo T* , definita come $A(t)^{avg} = \frac{1}{T} \int_t^{t+T} A(\tau) d\tau$.

Definizione 3. Il grafo $\mathcal{G}_T^{avg}(t)$ indotto da $A_T(t)^{avg}$ si dice uniformemente connesso se $\forall t \in \mathbb{R}, \{\exists i \in \nu, i \text{ centroide di } \mathcal{G}_T(t)\}$, cioè se $\mathcal{G}_T^{avg}(t)$ è connesso per ogni t . Si può verificare la connessione di $\mathcal{G}_T^{avg}(t)$ all'istante t sfruttando il Teorema 1.

1.1.3 Componenti indipendenti di un grafo.

È possibile definire alcune proprietà di interesse sui sottografi di un grafo complessivo $\mathcal{G}(t)$; diamo quindi la definizione di *sottografo* per poi enunciare queste proprietà:

Definizione 4. Un sottografo di \mathcal{G} è una coppia (ν_s, ϵ_s) tale che $\nu_s \subseteq \nu$ e $\epsilon_s \subseteq \epsilon \cap (\nu_s \times \nu_s)$.

Definizione 5 (Componente fortemente connessa di un grafo.). Al tempo t una *iSCC* di un grafo diretto $\mathcal{G}(t) = (\nu, \epsilon)$ è un suo sottografo $\mathcal{G}(t) = (\nu_s, \epsilon_s)$ che ha la proprietà di essere *massimo*, *fortemente connesso* e tale che $\forall i \in \nu \setminus \nu_s, (i, j) \notin \epsilon(t)$ con $j \in \nu_s$.

Ciò significa che non esiste nessun arco con coda in un nodo di $\nu \setminus \nu_s$ e testa in un nodo di ν_s ; ovvero, per quanto ci riguarda, nessun sistema dinamico rappresentato da un nodo di ν_s è influenzato da qualunque altro sistema dinamico rappresentato da un nodo all'esterno di ν_s . Questo giustifica l'aggettivo *indipendente* per un iSCC.

Definizione 6 (Componente indipendente connessa semplicemente). Una *componente indipendente connessa (iCC)* di un grafo diretto è un suo sottografo che ha la proprietà di essere **massimo** e **connesso**, ed è tale che non esiste un arco di ϵ con coda in un nodo non appartenente a ν_s e testa in ν_s .

Sia adesso $c \geq 1$ il numero di iCC di un grafo diretto. Possiamo suddividere l'intero grafo in $c + 1$ sottografi: i primi c sono le iCC stesse, l'ultimo è il sottografo composto

da quei nodi (se esistono), chiamati *residui*, che possono avere degli archi entranti provenienti da nodi di qualcuna delle iCC, ma non degli archi uscenti diretti a nodi delle iCC. Tale grafo viene detto *grafo residuo*.

Un ulteriore importante risultato di nostro interesse è il seguente:

Teorema 2. *Dato un grafo diretto $\mathcal{G}(t) = (\nu, \epsilon)$, la dimensione dell'autospazio relativo all'autovalore 0 della matrice Laplaciana $L(t)$ associata a tale grafo, i.e. $\dim[\text{Ker}(L(t))]$, è pari al numero di iSCC (o analogamente iCC) che compongono il grafo.*

1.2 Il prodotto di Kronecker.

In matematica, nel campo dell'algebra lineare, il prodotto di Kronecker, indicato con \otimes , è una operazione tra due matrici di dimensioni arbitrarie, sempre applicabile.

Se A è una matrice $m \times n$ e B una matrice $p \times q$, allora il loro prodotto di Kronecker è una matrice definita a blocchi nel modo seguente:

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \dots & \dots & \dots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}$$

cioè, esplicitando ogni termine:

$$A \otimes B = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{11} & \dots & a_{11}b_{1q} & \dots & \dots & a_{1n}b_{11} & a_{1n}b_{12} & \dots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2q} & \dots & \dots & a_{1n}b_{21} & a_{1n}b_{22} & \dots & a_{1n}b_{2q} \\ \dots & \dots \\ a_{11}b_{p1} & a_{11}b_{p2} & \dots & a_{11}b_{pq} & \dots & \dots & a_{1n}b_{p1} & a_{1n}b_{p2} & \dots & a_{1n}b_{pq} \\ \dots & \dots \\ \dots & \dots \\ a_{m1}b_{11} & a_{m1}b_{11} & \dots & a_{m1}b_{1q} & \dots & \dots & a_{mn}b_{11} & a_{mn}b_{12} & \dots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \dots & a_{m1}b_{2q} & \dots & \dots & a_{mn}b_{21} & a_{mn}b_{22} & \dots & a_{mn}b_{2q} \\ \dots & \dots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \dots & a_{m1}b_{pq} & \dots & \dots & a_{mn}b_{p1} & a_{mn}b_{p2} & \dots & a_{mn}b_{pq} \end{pmatrix}$$

1.2.1 Proprietà del prodotto di Kronecker

Bilinearità e associatività

Il prodotto di Kronecker è un caso speciale di prodotto tensoriale, dunque è bilineare e associativo:

- $A \otimes (B + C) = A \otimes B + A \otimes C.$
- $(A + B) \otimes C = A \otimes C + B \otimes C.$
- $(kA) \otimes B = A \otimes (kB) = k(A \otimes B).$
- $(A \otimes B) \otimes C = A \otimes (B \otimes C).$

Tuttavia, questo prodotto non è commutativo.

Prodotto misto

Se A, B, C, D sono matrici tali che esiste il prodotto righe per colonne tra A e C e tra B e D , allora esiste anche $(A \otimes B)(C \otimes D)$ e vale che:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$$

Ne segue che $(A \otimes B)$ è invertibile se e solo se lo sono A e B e l'inversa è data da $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$

Spettro

Siano A e B matrici quadrate di ordine rispettivamente n e q , e siano $\lambda_1, \dots, \lambda_n$ gli autovalori di A , μ_1, \dots, μ_q quelli di B . Allora gli autovalori di $A \otimes B$ sono tutti quelli della forma:

$$\lambda_i \mu_j, \quad i = 1, \dots, n; \quad j = 1, \dots, q.$$

Ne seguono le seguenti proprietà su determinante e traccia:

- $tr(A \otimes B) = tr(A)tr(B)$.
- $det(A \otimes B) = det(A)^q det(B)^n$

1.3 Modelli nello spazio degli stati.

È possibile rappresentare ogni sistema dinamico lineare d -dimensionale tramite il cosiddetto *modello nello spazio degli stati*. Esso è costituito dalle equazioni:

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \end{cases} \quad (1.1)$$

dove

$$\mathbf{x}(t) = \text{col}[x_1(t), \dots, x_d(t)]$$

è il vettore contenente le d variabili di stato del sistema,

$$\mathbf{y}(t) = \text{col}[y_1(t), \dots, y_m(t)]$$

è il vettore contenente le m uscite,

$$\mathbf{u}(t) = \text{col}[u_1(t), \dots, u_d(t)]$$

è il vettore contenente i d ingressi di controllo, e le matrici A , B , C , D sono chiamate *matrici di sistema*.

1.4 Rappresentazioni equivalenti e cambio di variabile.

La scelta del modello che rappresenta il comportamento del nostro sistema non è affatto univoca. Consideriamo infatti il modello 1.1 : se prendiamo una trasformazione lineare $T \in \mathbb{R}^{d \times d}$ biunivoca e quindi non singolare, possiamo effettuare un *cambio di variabili* definendo il vettore di stato \mathbf{z} come

$$\mathbf{z}(t) = T\mathbf{x}(t). \quad (1.2)$$

Visto che T è invertibile, tra \mathbf{x} e \mathbf{z} risulta una corrispondenza biunivoca:

$$\mathbf{x}(t) = T^{-1}\mathbf{z}(t). \quad (1.3)$$

Sostituendo la 1.3 nelle equazioni in 1.1, dopo aver moltiplicato a sinistra per T si ottiene il modello:

$$\begin{cases} \dot{\mathbf{z}}(t) = A'\mathbf{z}(t) + B'\mathbf{u}(t) \\ \mathbf{y}(t) = C'\mathbf{z}(t) + D'\mathbf{u}(t) \end{cases} \quad (1.4)$$

con

$$A' = TAT^{-1}, \quad B' = TB, \quad C' = CT^{-1}, \quad D' = D. \quad (1.5)$$

Questo sistema dinamico è *equivalente* a quello descritto dalle equazioni 1.1 nel senso che, per un ingresso $\mathbf{u}(t), t \geq t_0$, e due stati iniziali x_{t_0} e z_{t_0} correlati dalla condizione $z_{t_0} = Tx_{t_0}$, le traiettorie dello stato in 1.1 e in 1.4 sono effettivamente legati dalla condizione 1.3 e i movimenti dell'uscita sono identici. Tutto ciò è semplicemente verificabile per sostituzione. Ne ricaviamo che le quadruple di matrici (A, B, C, D) e (A', B', C', D') ,

D') sono due maniere differenti di descrivere un medesimo oggetto fisico.

Nel seguito avremo bisogno di descrivere in forma compatta un cambio di variabili su un set di n sistemi ognuno dei quali ha d variabili di stato: sfrutteremo quindi lo strumento del prodotto di Kronecker. Se $\mathbf{w} = [w_1, \dots, w_n]$ è il vettore dei vettori di stato d -dimensionali di ogni sistema ($w_i \in \mathbb{R}^d$) e vogliamo applicare il cambio di coordinate T , calcoleremo dunque:

$$\mathbf{z} = (T^{-1} \otimes I_d)\mathbf{w}$$

1.4.1 Matrici diagonali e trasformazione di Jordan.

Nel problema del consenso risulta utile effettuare un cambio di variabili sugli stati per portare la matrice Laplaciana che descrive le connessioni tra i sistemi in una forma più comoda. In particolare ci interessa ottenere una matrice *diagonale*, in modo che le equazioni che descrivono l'evoluzione di ogni sistema dipendano soltanto dalle variabili di stato del sistema stesso e non da quelle dei sistemi suoi vicini nel grafo di comunicazione.

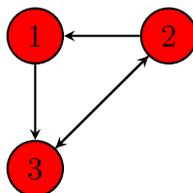
Teorema 3 (Diagonalizzabilità di una matrice). *Una matrice A si dice diagonalizzabile se e solo se $\exists V, \exists \Lambda, \Lambda = V^{-1}AV$, dove V è una opportuna matrice invertibile formata da autovettori di A , mentre Λ è una matrice diagonale. In tal caso A e Λ sono matrici simili, e cioè hanno:*

- *Stesso determinante.*
- *Stessa traccia.*
- *Stesso rango.*
- *Stessi autovalori.*

Purtroppo però non tutte le matrici sono diagonalizzabili. Prendiamo ad esempio la matrice Laplaciana

$$L = \begin{pmatrix} 2 & -1 & -1 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}$$

associata al semplice grafo



Dal momento che l'autospazio relativo al suo autovalore $\lambda = 2$ ha dimensione 1 nonostante la corrispondente molteplicità aritmetica sia pari a 2, non è possibile trovare una matrice V formata di autovettori di L che abbia rango pari a 3. Non esiste quindi una matrice diagonale simile ad L .

Introduciamo quindi un nuovo strumento matematico che ci permetta di effettuare un cambio di variabili efficace anche in presenza di Laplaciane non diagonalizzabili: la **trasformazione in forma canonica di Jordan**.

1.4.2 Forma canonica di Jordan.

In algebra lineare, la forma canonica di Jordan di una matrice quadrata A definisce una matrice triangolare J simile ad A che ha una struttura il più possibile vicina ad una matrice diagonale. La matrice è diagonale se e solo se A è diagonalizzabile, altrimenti è divisa in blocchi detti *blocchi di Jordan*.

Una matrice in forma di Jordan è una matrice diagonale a blocchi del tipo

$$J = \begin{pmatrix} J_1 & & 0 \\ & \cdots & \\ 0 & & J_k \end{pmatrix}$$

dove J_i è il blocco di Jordan relativo all'autovalore λ_i , definito come

$$J_i = \begin{pmatrix} \lambda_i & 1 & 0 & \dots \\ 0 & \dots & & \\ & \dots & 0 & \\ & \dots & \dots & 1 \\ 0 & \dots & \dots & \lambda_i \end{pmatrix}.$$

Esso contiene cioè sulla diagonale il valore λ_i ed il valore 1 in ogni posizione $(i, i+1)$. Notiamo che qualora tutti i blocchi di Jordan abbiano dimensione unitaria, la matrice in forma canonica di Jordan è diagonale.

Il risultato che ci interessa è il seguente: è sempre possibile trovare una matrice \tilde{L} in forma canonica di Jordan simile a L tale che $\exists T_J$ le cui colonne sono autovettori o autovettori generalizzati di L per cui vale $\tilde{L} = T_J^{-1} L T_J$, come vedremo in 3.4.

1.5 La definizione di consenso

Si considera il problema di coordinare tra di loro n sistemi omogenei appartenenti ad una rete, in grado di scambiarsi informazioni secondo una topologia che può efficacemente essere descritta da un grafo di comunicazione $\mathcal{G}(t)$. Mostriamo nel capitolo 2 che sotto certe ipotesi sia per quanto riguarda la topologia della rete che l'informazione scambiata tra gli agenti, le variabili di stato dei sistemi convergono asintoticamente ad un valore comune.

Consideriamo n sistemi lineari, per ognuno dei quali le variabili di stato variano secondo l'equazione

$$\dot{\mathbf{x}}_k = A\mathbf{x}_k + B\mathbf{u}_k, \quad \forall k \in \{1, \dots, n\} \quad (1.6)$$

con $\dot{\mathbf{x}}_k \in \mathbb{R}^p$ e $\mathbf{u}_k \in \mathbb{R}^m$.

Si dice che il *consenso* viene raggiunto se, in presenza di un opportuno ingresso di controllo, le variabili di stato dei diversi sistemi convergono ad un valore comune, cioè se

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, n\} \rightarrow \lim_{t \rightarrow +\infty} (\mathbf{x}_i(t) - \mathbf{x}_j(t)) = 0. \quad (1.7)$$

Raggiungere il *consenso* può essere considerato quindi un problema di convergenza, dove gli stati di equilibrio devono essere tutti valori coincidenti e la stabilità deve essere di tipo “attrattivo”.

Ci sono diversi strumenti utilizzabili per l’analisi del problema: quello a cui siamo più interessati è l’analisi delle proprietà spettrali delle matrici associate al grafo di comunicazione del sistema. Altri metodi di analisi, come vedremo nel capitolo 2, si basano sul concetto di funzione di Lyapunov.

Capitolo 2

L'algoritmo di consenso e simulazioni su sistemi Integratori.

Abbiamo definito in 1.5 il problema del consenso su sistemi dinamici; siamo adesso interessati ad illustrare quale protocollo, basato sulla topologia di comunicazione, permetta di generare un ingresso di controllo che porti alla convergenza dei sistemi su un valore comune.

2.1 Il protocollo di consenso.

Consideriamo il caso di n sistemi integratori connessi tra di loro (matrice di sistema $A = 0$ nel modello nello spazio degli stati), la cui topologia di comunicazione è descitta dal grafo $\mathcal{G}(t)$: per semplicità, consideriamo il caso di grafi non pesati in cui $a_{kl} = 1$ se esiste un arco da l a k , altrimenti $a_{kl} = 0$.

Ciascuno degli n sistemi è descritto dalla dinamica:

$$\dot{x}_k = u_k, \tag{2.1}$$

dove x_k è lo stato del sistema e u_k è l'ingresso di controllo.

Se definiamo tale ingresso come:

$$u_k = \sum_{i=1}^n a_{ki}(x_i - x_k) = - \sum_{i=1}^n l_{ki}x_i, \quad (2.2)$$

dove l_{ki} è l'elemento ki della matrice Laplaciana associata al grafo \mathcal{G} , abbiamo applicato quello che in letteratura è chiamato *protocollo di consenso*. Tale azione di controllo dipende dalla differenza relativa tra le variabili di stato (termini del tipo $x_i - x_k$) di agenti che sono effettivamente comunicanti (contributo degli elementi a_{ki} oppure l_{ki}).

In definitiva, se applichiamo ad ogni sistema il *protocollo di consenso*, possiamo scrivere in forma vettoriale che:

$$\dot{\mathbf{x}} = -L\mathbf{x} \quad (2.3)$$

dove

$$\mathbf{x} = \text{col}(x_1, x_2, \dots, x_n).$$

La soluzione nel tempo dell'equazione 2.3, è data da

$$\mathbf{x}(t) = e^{-Lt}\mathbf{x}_0 \quad (2.4)$$

dove $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$ è il valore iniziale degli n integratori.

2.1.1 Analisi spettrale della Laplaciana.

Verifichiamo adesso che l'*algoritmo di consenso* sopra descritto porta alla convergenza degli stati su valori coincidenti, attraverso l'analisi spettrale della matrice laplaciana L supposta per il momento indipendente dal tempo. Vale:

- $\lambda_0 = 0$ è **autovalore di L** . Dal momento che la somma degli elementi su ogni riga è nulla per definizione, il determinante è sicuramente nullo, cioè $\text{rank}(L) < n$. Visto che $\dim(\text{Ker}(L)) = n - \text{rank}(L)$, ne deriva che la dimensione dell'autospazio relativo all'autovalore 0 è non nulla, quindi 0 è autovalore.

Visto che $L\mathbf{1}_n = \mathbf{0}_n$, ne ricaviamo subito che $\mathbf{1}_n$ è autovettore destro di L .

- **L è una matrice con autovalori a parte reale non negativa.**

Nel caso di **grafo non diretto**, si può dimostrare che gli autovalori sono reali non negativi. Per provarlo, occorre introdurre una definizione:

Definizione 7. Si definisce Laplaciana dell'arco (i, j) la matrice $L_{(i,j)}$ che consiste dei valori non nulli relativi al solo arco (i, j) . Se l'arco è ad esempio $(1, 2)$, allora

$$L_{(1,2)} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \oplus [\text{zeri}]$$

dove con \oplus si indica la somma di Kronecker. Si può quindi scrivere che

$$L = \sum_{(i,j) \in \epsilon} L_{(i,j)}$$

Notando che

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \end{pmatrix}$$

Possiamo dire che

$$x^T L_{(1,2)} x = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (x_1 - x_2)^2$$

ed arrivare alla conclusione che

$$x^T L x = x^T \left(\sum_{(i,j) \in \epsilon} L_{(i,j)} \right) x = \sum_{(i,j) \in \epsilon} (x_i - x_j)^2$$

dimostrando così che $x^T L x$ è una forma quadratica e L è quindi semidefinita positiva per definizione (ricordiamo che una matrice semidefinita positiva ha autovalori reali non negativi).

Nel caso che il grafo sia invece **diretto**, è necessario sfruttare il *teorema dei cerchi di Greshgorin* per dimostrare che tutti gli autovalori, generalmente complessi, hanno parte reale non negativa.

Se consideriamo l'equazione 2.3 e i risultati derivanti dall'analisi spettrale appena effettuata, ci rendiamo conto che $k_n = \text{col}(k, \dots, k)$, $k \in \mathbb{R}$, è uno **stato di equilibrio** per ogni $k \in \mathbb{R}$: se ad un certo istante t^* si ha che $\mathbf{x}(t^*) = \mathbf{k}_n$ vale infatti che:

$$\dot{\mathbf{x}}(t^*) = -L\mathbf{k}_n = -kL\mathbf{1}_n = \mathbf{0}_n. \quad (2.5)$$

Dal momento che ogni vettore del tipo \mathbf{k}_n rappresenta un vettore di stati che hanno raggiunto il consenso (tutti gli stati hanno lo stesso valore k), ne deduciamo che tutti gli **stati di consenso** sono anche possibili **stati di equilibrio**.

Resta da capire se gli stati di consenso sono stati di equilibrio **asintoticamente stabili**. Abbiamo mostrato che la laplaciana L ha autovalori a parte reale non negativa e ad ogni autovalore λ di L , corrisponde l'autovalore $-\lambda$ di $-L$. Dunque gli autovalori della matrice $-L$, che governa l'evoluzione del sistema, hanno sicuramente parte reale non positiva e sappiamo dalla teoria dei controlli che ciò corrisponde a stabilità attrattiva.

2.2 Altri modi di analizzare la convergenza degli algoritmi di consenso.

Effettuare un'analisi spettrale della matrice Laplaciana non è l'unico approccio che possiamo seguire per analizzare la convergenza dell'equazione 2.3 sugli stati di consenso. Altri approcci si basano sul trovare opportune funzioni di Lyapunov che ci garantiscano la stabilità esponenziale, con alcuni vantaggi derivanti quali una descrizione qualitativa più comprensibile del comportamento dei sistemi.

2.2.1 Analisi di stabilità tramite funzioni di Lyapunov.

Introduciamo innanzitutto il concetto di *funzione di Lyapunov*.

Funzioni di Lyapunov e sistemi dinamici.

Consideriamo il sistema lineare omogeneo tempo continuo

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t), \quad \mathbf{x} \in \mathbb{R}^n \quad (2.6)$$

Associamo a questo sistema dinamico la funzione quadratica

$$V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x} \quad (2.7)$$

dove P è definita simmetrica e positiva. Questa funzione V è continua ed ha le derivate parziali prime continue. Perciò, dal momento che P è definita positiva, l'origine è l'unico punto di minimo di V . Dunque, una forma quadratica definita positiva come questa è una possibile candidata funzione di Lyapunov.

Rimane da determinare come $\dot{V}(\mathbf{x})$ sia influenzato dalle dinamiche del sistema. Si ha

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \frac{d}{dt} \mathbf{x}^T P \mathbf{x} \\ &= \dot{\mathbf{x}}^T P \mathbf{x} + \mathbf{x}^T P \dot{\mathbf{x}} \\ &= \mathbf{x}^T A^T P \mathbf{x} + \mathbf{x}^T P A \mathbf{x} \\ &= \mathbf{x}^T (A^T P + P A) \mathbf{x} \end{aligned} \quad (2.8)$$

Pertanto, definendo la matrice

$$-Q = A^T P + P A \quad (2.9)$$

si ha

$$\dot{V}(\mathbf{x}) = -\mathbf{x}^T Q \mathbf{x} \quad (2.10)$$

La funzione $\dot{V}(\mathbf{x})$ è nuovamente una forma quadratica e V sarà una funzione di Lyapunov se la matrice Q è semidefinita positiva. Infatti, se Q è semidefinita positiva si può affermare che il sistema è asintoticamente stabile e converge cioè con velocità esponenziale.

Nel caso del protocollo di consenso:

$$\dot{\mathbf{x}}(t) = -L\mathbf{x}(t), \quad \mathbf{x} \in \mathbb{R}^n$$

la convergenza può essere verificata considerando la funzione candidata di Lyapunov

$$V(\mathbf{x}, t) = \frac{\mathbf{x}^T(t)\mathbf{x}(t)}{2} = \frac{\sum_{i=1}^n x_i^2}{2}. \quad (2.11)$$

Risulta che

$$\dot{V}(\mathbf{x}, t) = \frac{\dot{\mathbf{x}}^T \mathbf{x} + \mathbf{x}^T \dot{\mathbf{x}}}{2} = \frac{-\mathbf{x}^T L^T \mathbf{x} - \mathbf{x}^T L \mathbf{x}}{2} = -\frac{\mathbf{x}^T (L^T + L) \mathbf{x}}{2}. \quad (2.12)$$

Nel caso in cui il grafo associato ad L sia **non diretto**, L risulta essere simmetrica e dunque vale

$$\dot{V}(\mathbf{x}, t) = -\mathbf{x}^T L \mathbf{x}. \quad (2.13)$$

Dal momento che L è semidefinita positiva, le traiettorie degli stati convergono ad uno dei valori per cui $\dot{V}(\mathbf{x}, t) = 0$: essendo la somma degli elementi di ogni riga di una Laplaciana nulla, vale che

$$\{\mathbf{x} \in \mathbb{R}^n \mid \dot{V}(\mathbf{x}, t) = -\mathbf{x}^T L \mathbf{x} = 0\} \supseteq \{\alpha \mathbf{1}_n \mid \alpha \in \mathbb{R}\} \quad (2.14)$$

ovvero, l'insieme di stati di consenso corrisponde all'insieme di stati di equilibrio attrattivo, come volevamo mostrare.

Nel caso che il grafo sia invece **diretto**, continua a valere la 2.12, ma non più la 2.13. D'altra parte, sotto l'ipotesi di un grafo diretto fortemente connesso, oppure debolmente connesso e bilanciato, vale che

$$\text{Ker}(L + L^T) = \text{Ker}(L) = \{\alpha \mathbf{1}_n \mid \alpha \in \mathbb{R}\} \quad (2.15)$$

e dunque abbiamo ancora che la funzione di Lyapunov 2.11 garantisce la convergenza all'insieme di consenso.

Nel prossimo paragrafo introduciamo una nuova possibile candidata di Lyapunov.

2.2.2 La proprietà di contrazione

Un modo piuttosto intuitivo di catturare le proprietà di convergenza di 2.3, è costruito su un'analisi dell'interazione elementare, che ricordiamo essere:

$$\dot{x}_k = a_{kl}(x_l - x_k).$$

Essa si basa sul fatto che ogni sistema x_k varia proporzionalmente alla distanza che corre tra lo stato di ogni sistema comunicante ed esso stesso, cioè alla quantità $x_l - x_k, \forall l$ comunicante. In altre parole, ogni stato si muove sempre in direzione degli altri stati collegati nel grafo di comunicazione. Varrà dunque che $\min\{x_1, \dots, x_n\}$ non può decrescere e $\max\{x_1, \dots, x_n\}$ non può crescere. Possiamo dunque introdurre la funzione di Lyapunov

$$V(\mathbf{x}) = \max\{x_1, \dots, x_n\} - \min\{x_1, \dots, x_n\} \quad (2.16)$$

che è definita positiva sul set degli stati di consenso $\{\mathbf{x} \in \mathbb{R}^n : x_1 = \dots = x_n\}$.

Utilizzando questa funzione di Lyapunov, sempre decrescente, siamo in grado di assicurare la convergenza sotto un'assunzione piuttosto debole di connessione su intervalli di tempo, come definito dal seguente teorema (del quale omettiamo la dimostrazione).

Teorema 4. *Si consideri il sistema lineare*

$$\dot{\mathbf{x}} = -L(t)\mathbf{x}$$

Si assuma che $L(t)$ sia una funzione limitata e quasi-continua nel tempo. Si assuma inoltre che $-L(t)$ sia Metzler con somme di righe nulle. Se esiste un indice $k \in \{1, \dots, n\}$,

un valore di soglia $\delta > 0$ e un periodo di tempo $T > 0$ tale che per ogni $t \in \mathbb{R}$ il grafo medio associato:

$$\int_t^{t+T} A(s) ds$$

ha la proprietà che tutti i nodi possono essere raggiunti dal nodo k (ed è cioè connesso), allora il set degli stati di consenso è uniformemente ed esponenzialmente stabile. In particolare, tutte le componenti di ogni soluzione $\zeta(t)$ del sistema lineare considerato convergono a un valore comune per $t \rightarrow \infty$.

La funzione di Lyapunov di cui si è parlato in questa sottosezione permette di dimostrare il teorema.

2.3 Il valore di convergenza: l'*average-consensus problem*.

Abbiamo verificato che il protocollo di consenso garantisce convergenza: ci chiediamo adesso quale sia il valore di consenso raggiunto dagli agenti. In particolare, apparirebbe naturale che tutti i sistemi convergessero su

$$k = \frac{1}{n} \sum_{i=1}^n x_i(0), \quad (2.17)$$

andando a risolvere quello che viene chiamato in letteratura *problema dell'average-consensus*. Purtroppo ciò non è garantito dal semplice fatto che il grafo sia connesso, come vediamo nel seguito.

Notiamo innanzitutto che il valore finale di convergenza deve essere quel valore k tale che:

$$k\mathbf{1}_n = \lim_{t \rightarrow +\infty} \mathbf{x}(t) = \lim_{t \rightarrow +\infty} (e^{-Lt})\mathbf{x}_0 = R\mathbf{x}_0. \quad (2.18)$$

nella quale $R = \lim_{t \rightarrow +\infty} (e^{-Lt})$. Dunque, se siamo in grado di conoscere R , sfruttando la relazione

$$k\mathbf{1}_n = R\mathbf{x}_0$$

è possibile calcolare direttamente il k di convergenza. Vale, in effetti, il seguente teorema:

Teorema 5. *Dato un grafo diretto \mathcal{G} fortemente connesso con laplaciano L tale $Lw_r = 0$, $w_l^T L = 0$ e $w_l^T w_r = 1$ (w_l e w_r sono quindi due autovettori, rispettivamente sinistro e destro, associati all'autovalore 0 della Laplaciana), si ha che $R = w_r w_l^T \in \mathbb{R}^{n \times n}$.*

Affinchè k sia la media dei valori iniziali, è sufficiente che valga una tra 3 condizioni: la dimostrazione di ciò esula dagli scopi di questa tesi. Riportiamo queste 3 condizioni nel seguente teorema, il quale afferma che sono in effetti condizioni equivalenti (di tale fatto riportiamo invece la semplice dimostrazione).

Teorema 6. *Sia \mathcal{G} un grafo diretto con matrice di adiacenza A . Tutte le seguenti affermazioni sono equivalenti:*

1. G è bilanciato (per ogni nodo, il grado di entrata è uguale a quello di uscita);
2. $w_l = \mathbf{1}$ è l'autovettore **sinistro** del Laplaciano di \mathcal{G} associato all'autovalore nullo, cioè $\mathbf{1}^T L = 0$;
3. $\sum_{i=1}^n u_i = 0, \forall x \in \mathbb{R}^n$,

Dimostrazione. • La seconda affermazione indica che la somma dei valori **per colonna** del Laplaciano è nulla. La somma degli elementi della colonna j -esima corrisponde a:

$$\sum_{i=1}^n l_{ij} = - \sum_{i=1, i \neq j}^n a_{ij} + l_{jj} = -deg_{in}(v_j) + deg_{out}(v_j).$$

e nel caso che venga posta a 0, porta a osservare che il nodo j deve essere bilanciato. Le stesse osservazioni valgono per ogni altra colonna, dimostrando l'equivalenza tra l'affermazione 1 e la 2.

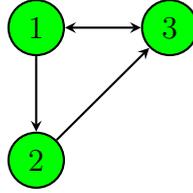
- Per quanto riguarda l'equivalenza tra la 2 e la 3, vale che:

$$0 = \sum_{i=1}^n u_i = \mathbf{1}^T \mathbf{u} = -\mathbf{1}^T L \mathbf{x},$$

se e soltanto se $\mathbf{1}^T L = 0$, e cioè $\mathbf{1}^T$ è autovettore sinistro di L associato all'autovalore nullo.

□

Per verificare la validità di quanto detto su un caso reale, prendiamo come esempio il seguente grafo:



Il grafo risulta connesso, ma il nodo 3 non è bilanciato e vale infatti che:

$$\begin{cases} u_1 = (x_2 - x_1) + (x_3 - x_1) \\ u_2 = (x_3 - x_2) \\ u_3 = x_1 - x_3 \end{cases}$$

da cui segue che $\sum_{i=1}^n u_i = x_3 - x_1 \neq 0$ se i nodi 1 e 3 non partono dallo stesso valore. Il Laplaciano risultante presenta gli autovalori 0, 2, 2 ed autovettori destro e sinistro dell'autovalore nullo pari a $w_r^T = (1, 1, 1)$ e $w_l^T = (\frac{1}{4}, \frac{1}{4}, \frac{1}{2})$. Si ha dunque che:

$$R = w_r w_l^T = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

e il valore di convergenza sarà $R\mathbf{x}_0 = \frac{1}{4}(x_1(0) + x_2(0) + 2x_3(0))$, diverso dal valor medio.

Sfruttando i risultati ottenuti, enunciamo il seguente teorema riassuntivo:

Teorema 7. *Si consideri una rete di integratori con una topologia costante rappresentata da un grafo \mathcal{G} diretto e fortemente connesso. Il protocollo del consenso è globalmente asintoticamente stabile e risolve il problema dell'average-consensus **se e soltanto se il grafo è bilanciato e debolmente connesso**. Nel caso il grafo non sia bilanciato,*

ma sia connesso, si risolve il problema del consensus, ma il valore di convergenza sarà pari a $k = w_r w_l^T \mathbf{x}_0 \neq \frac{1}{n} \sum_{i=1}^n x_i(0)$, con w_r e w_l autovettori destro e sinistro associati all'autovalore nullo della Laplaciana $L(G)$.

2.4 Simulazioni su sistemi integratori.

Si riporta in questa sezione il risultato di alcune simulazioni effettuate in ambiente Simulink su sistemi integratori.

Lo scopo delle seguenti simulazioni è, in primis, quello di trovare riscontro delle proprietà di convergenza dell'*algoritmo di consenso* descritte nella teoria, mettendo in evidenza anche i casi meno intuitivi.

La **simulazione 1** mostra una prima applicazione dell'*algoritmo di consenso* su 5 agenti interconnessi da *grafi di comunicazione* connessi interamente oppure a componenti indipendenti.

La **simulazione 2** mostra il comportamento di 5 agenti il cui *grafo di comunicazione* non è costante nel tempo, ma viene sostituito ad intervalli costanti con altri grafi appartenenti ad una collezione finita. Verranno fatte alcune considerazioni sulle proprietà di convergenza dei 5 agenti nel caso in cui i grafi appartenenti alla collezione siano tutti *sconnessi*.

La **simulazione 3** indaga le proprietà di convergenza delle componenti residuali di un *grafo di comunicazione* connesso a blocchi.

2.4.1 Simulazione 1: una prima applicazione.

Per questa simulazione abbiamo utilizzato il modello simulink di figura 2.1, che implementa l'equazione differenziale $\dot{\mathbf{x}} = -L\mathbf{x}$. Esso è composto di una matrice Laplaciana

che agisce sugli stati x_1, x_2, x_3, x_4, x_5 il cui valore iniziale è rappresentato dal vettore \mathbf{x}_0 .

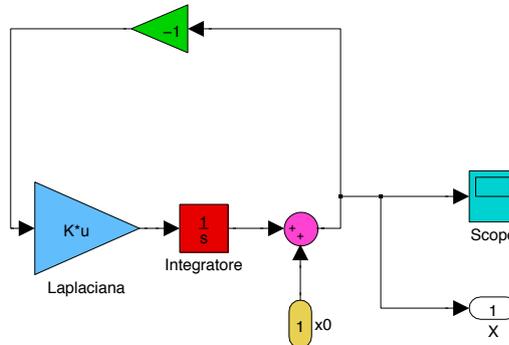
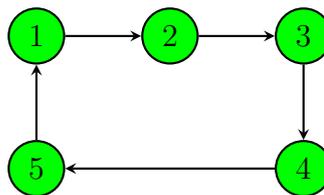


Figura 2.1: Schema Simulink

Si è quindi fatto riferimento a diverse matrici Laplaciane (ovvero a diversi *grafi di comunicazione*) e si è osservata l'evoluzione nel tempo degli stati x_1, x_2, x_3, x_4, x_5 . Riportiamo nel seguito i risultati della simulazione sia nel caso di un grafo connesso che di un grafo non connesso.

Grafo connesso

Il *grafo di comunicazione* a cui si è fatto riferimento è il seguente.



Ad esso corrisponde la matrice laplaciana

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Il risultato della simulazione è riportato in Figura 2.2.

Come previsto dalla teoria, i sistemi raggiungono il consenso, ovvero $\lim_{t \rightarrow +\infty} x_i(t) - x_j(t) = 0 \quad \forall i, \forall j$. Il valore di convergenza sarà inoltre pari a $\frac{1}{5} \sum_{i=1}^5 x_i(0)$, dal momento che il grafo è bilanciato e debolmente connesso.

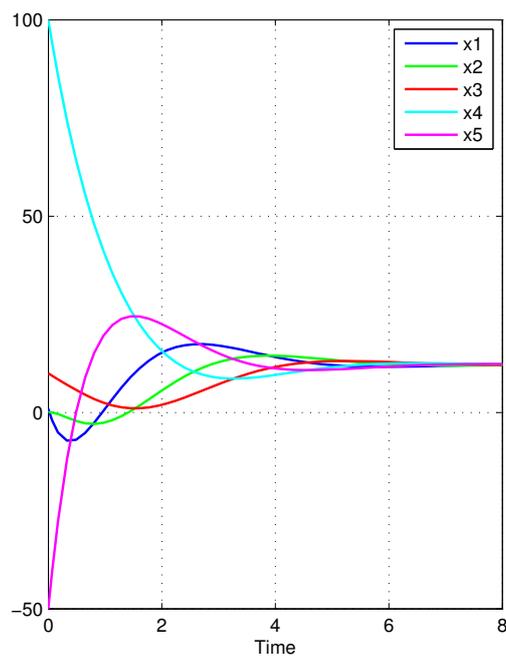
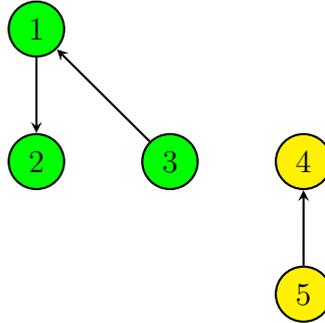


Figura 2.2: Grafo Connesso

Grafo sconnesso

Nel secondo caso si è invece fatto riferimento al seguente grafo:



Ad esso corrisponde la matrice laplaciana

$$L = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Il risultato della simulazione è riportato in Figura 2.3.

In questo caso notiamo che i sistemi $1,2,3$ e i sistemi $4,5$ raggiungono due valori di consenso differenti: nonostante il grafo complessivo risulti sconnesso, infatti, i sottografi relativi ai sistemi $1,2,3$ e a $4,5$ sono connessi.

2.4.2 Simulazione 2: Grafi sconnessi variabili nel tempo.

Fino ad ora si è fatto riferimento a matrici Laplaciane costanti nel tempo ed abbiamo riscontrato, come previsto dalla teoria, che il consenso tra tutti i sistemi viene raggiunto *se e solo se* il *grafo di comunicazione* è connesso.

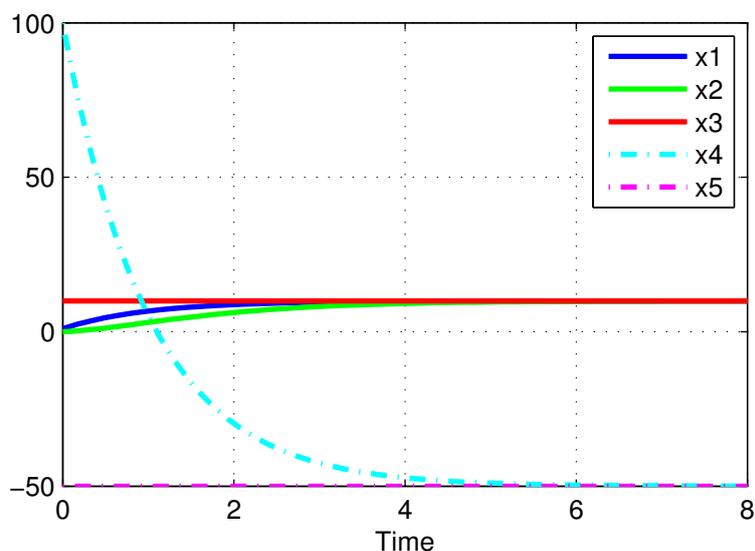
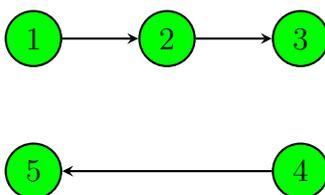


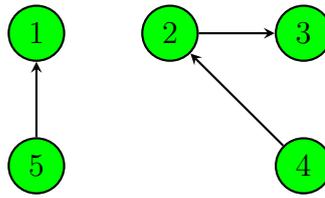
Figura 2.3: Grafo Sconnesso

Vogliamo adesso indagare il caso di matrici Laplaciane variabili nel tempo; in particolare, dato un periodo $T = \sum_{i=1}^m T_i$ ed m matrici Laplaciane L_1, L_2, \dots, L_m , durante l'intervallo $t \in [kT_{j-1}, kT_j]$, $\forall k \in \mathbb{N}$ il *grafo di comunicazione* associato al sistema sarà quello descritto dalla Laplaciana L_j .

Il caso più interessante dal punto di vista dello studio della convergenza dell'algoritmo di consenso è quello in cui tutte le matrici appartenenti al set L_1, L_2, \dots, L_m sono associate ad un *grafo di comunicazione sconnesso*: la seguente simulazione è volta a studiare proprio questa configurazione. Utilizzeremo, per semplicità, soltanto i seguenti due grafi sconnessi



e



a cui corrispondono le due laplaciane:

$$L_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \quad L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

e considereremo un periodo $T = T_1 + T_2 = 2$ secondi.

Lo schema Simulink che si è utilizzato è riportato in figura 2.4: la parte a destra dello schema ha lo scopo di passare il controllo allo script Matlab ad ogni secondo di simulazione, permettendo così la sostituzione della Laplaciana corrente con la successiva.

In figura 2.5 riportiamo il risultato della simulazione.

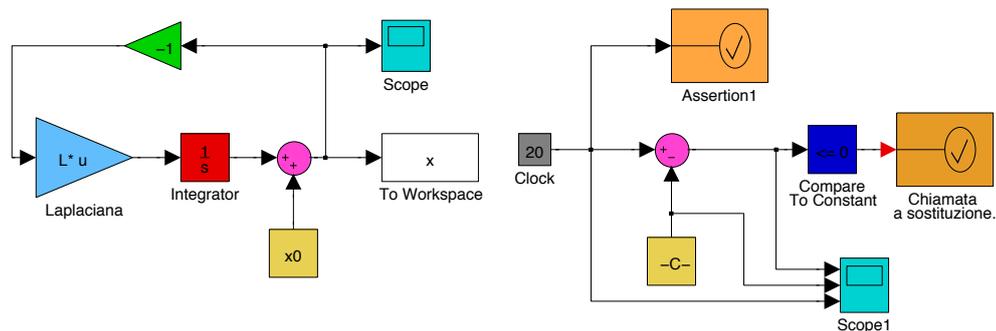


Figura 2.4: Modello Simulink

I sistemi raggiungono quindi il consenso, nonostante la Laplaciana che ne governa l'interazione sia istante per istante associata ad un *grafo di comunicazione sconnesso*.

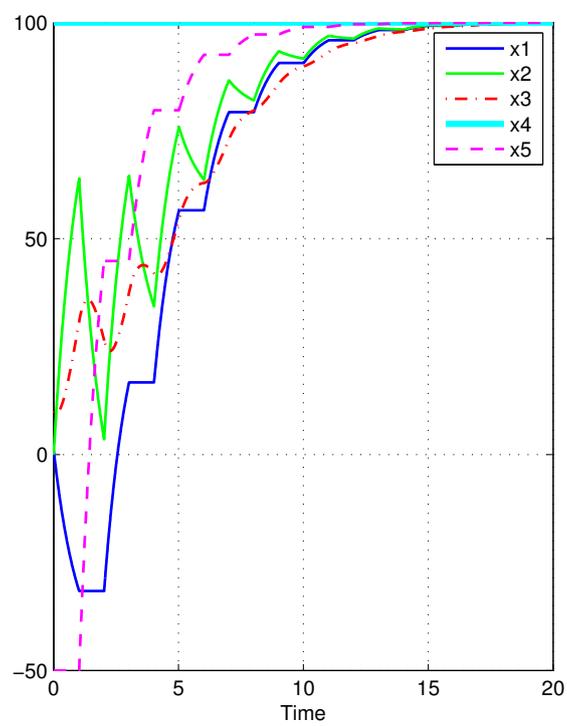


Figura 2.5: Risultato della simulazione

Per comprendere questo risultato, è possibile fare riferimento al concetto di *grafo indotto dalla matrice di adiacenza media sul periodo T* ($\mathcal{G}_T^{avg}(t)$), sul quale era possibile definire la *uniforme connessione*: tale proprietà è più debole della *connessione* in senso stretto, ma permette di ottenere la convergenza dell'algoritmo di consenso se esiste un T per cui $\mathcal{G}_T^{avg}(t)$ risulti uniformemente connesso per ogni t .

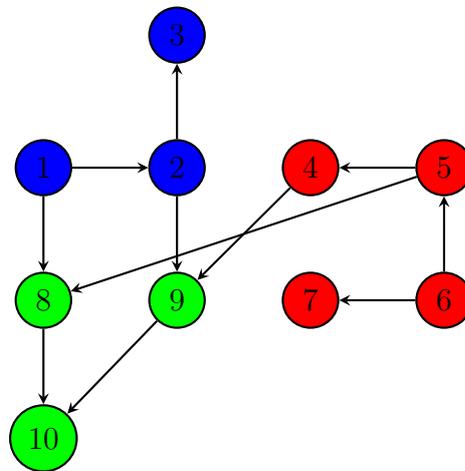
Nel nostro particolare caso il grafo $\mathcal{G}_{T=2}^{avg}(t)$, la cui matrice di adiacenza è $\mathcal{G}_{T=2}^{avg}(t) = (A_1 + A_2)/2$, risulta uniformemente connesso, giustificando così il raggiungimento del consenso.

Al fine di verificare algebricamente l'uniforme connessione della $\mathcal{G}_{T=2}^{avg}(t)$, è stata inserita nello script .m relativo alla simulazione una procedura che determina se la parte reale del secondo autovalore di $\mathcal{G}_{T=2}^{avg}(t)$ è positiva: il teorema 1, infatti, ci assicura che in tal caso $\mathcal{G}_T^{avg}(t)$ è uniformemente connesso e l'algoritmo di consenso converge.

2.4.3 Simulazione 3: Grafi complessi con componenti residuali.

Si vuole infine studiare il caso di un grafo con più componenti connesse indipendenti (iCC) ed alcuni nodi che non appartengono a nessuna componente, chiamati *residui*, come spiegato nel capitolo 1.

Nella simulazione, si è fatto riferimento al grafo



a cui corrisponde la matrice Laplaciana

$$L_{residui} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

I sistemi 1,2,3 e 4,5,6,7 formano due iCC, mentre i sistemi 8,9,10 sono dei residui e non fanno parte di una terza iCC. Il risultato della simulazione, in figura 2.6, mostra che i sistemi 8,9,10 (tratteggiati al fine di metterli in evidenza) riescono a raggiungere un proprio valore di consenso intermedio tra i valori di consenso dei sistemi 1,2,3 e dei sistemi 4,5,6,7. Ciò è in effetti conforme alla natura dei sistemi *residuali*, che pur non essendo in comunicazione tra loro ricevono le informazioni dalle stesse componenti indipendenti del grafo e finiscono quindi per raggiungere un “accordo” sul valore da assumere a regime.

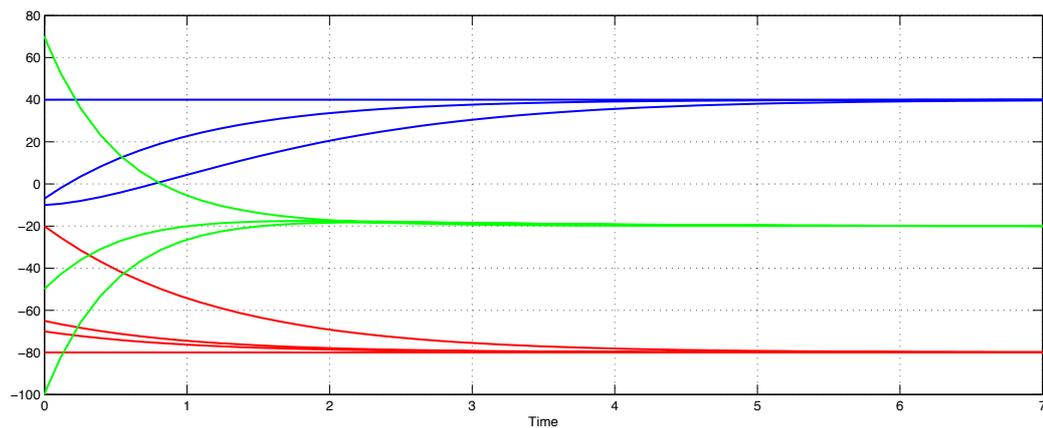


Figura 2.6: Risultato della simulazione.

Capitolo 3

Sistemi Multiagente Multivariabile.

Nelle precedenti simulazioni abbiamo considerato sistemi descritti da una sola variabile di stato, la cui risposta libera era costante ed uguale al valore di partenza: tale scelta, effettuata per una maggiore semplicità concettuale, era giustificata dal fatto che eravamo in quella sede interessati principalmente a testare l'efficacia e il campo di applicazione dell'algoritmo di consenso stesso, piuttosto che alla natura delle traiettorie sulle quali raggiungere il consenso.

Abbiamo tuttavia introdotto nel primo capitolo gli strumenti matematici che ci servono per gestire sistemi descritti da più di una variabile di stato. Per questi motivi ci accingiamo adesso a mostrare in che modo sia possibile ottenere il *consenso* su particolari sistemi multivariabile.

Abbiamo visto in 1.3 come rappresentare, tramite modelli nello spazio degli stati, sistemi con più variabili di stato. La matrice A , chiamata anche *matrice di sistema*, contiene l'informazione sull'evoluzione libera degli stati del sistema dinamico sotto esame; in particolare, se $\lambda_i, i = 1, \dots, n$ sono gli autovalori di A , la traiettoria degli stati $\mathbf{x}(t)$ sarà combinazione lineare di tutti i termini del tipo $e^{\lambda_i t}$. Ne deduciamo che:

- Sistemi dinamici con matrici di stato contenenti un autovalore a **parte reale positiva** hanno stati che divergono nel tempo, ed è pertanto poco sensato pensare al

consenso per raggiungere un accordo su traiettorie che, nonostante la coincidenza, rimarrebbero divergenti.

- Sistemi dinamici con matrici di stato con soltanto autovalori a **parte reale strettamente negativa** hanno, per loro natura, tutti gli stati che convergono al valore 0. Anche in questo caso, quindi, cercare un accordo sulle traiettorie sfruttando la teoria del consenso sembra poco significativo.
- Sistemi dinamici con matrici di stato con autovalori a **parte reale nulla** sembrano offrire maggiori prospettive per applicare la teoria del consenso. Essi espongono infatti modi che oscillano indefinitamente nel tempo, e quello di raggiungere un accordo su una traiettoria sinusoidale comune è un problema che si potrebbe presentare in molte occasioni.

Per questi motivi considereremo nel seguito il **consenso su oscillatori lineari**, ed indicheremo con S la matrice di sistema ad essi associata.

3.1 L'oscillatore lineare come sistema.

Interpretando opportunamente gli stati di più oscillatori armonici, controllati sfruttando un protocollo di consenso, sarà possibile generare delle traiettorie circolari a regime che potranno fungere da valori di riferimento per il movimento coordinato di più veicoli, come vedremo nel capitolo 4.

In fisica, viene chiamato *oscillatore armonico* qualunque sistema meccanico che reagisce ad una perturbazione dall'equilibrio con una accelerazione di richiamo proporzionale allo spostamento subito: esso è quindi descritto dall'equazione del moto $\ddot{x} = -\omega x$, dove x rappresenta la posizione dell'oscillatore ed $\omega > 0$ la pulsazione espressa in rad/sec. La soluzione dell'equazione differenziale è data da $x(t) = A \sin(\omega t + \phi)$, dove A rappresenta l'ampiezza dell'oscillazione e ϕ la fase: sia A che ϕ dipendono dalle condizioni al contorno $x(0)$ e $\dot{x}(0)$.

Dal momento che siamo interessati ad una descrizione in termini di variabili di stato, appare naturale effettuare la seguente scelta per il modello:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\omega x_1 \end{cases} \quad (3.1)$$

nel quale x_1 rappresenta la posizione dell'oscillatore e x_2 la sua velocità.

La matrice di sistema è dunque $S = \begin{pmatrix} 0 & 1 \\ -\omega & 0 \end{pmatrix}$: i suoi autovalori sono $\lambda = \pm\sqrt{\omega}i$, valori puramente immaginari che daranno quindi luogo ai modi oscillanti con pulsazione ω .

Possiamo anche considerare modelli con un numero arbitrario di stati che hanno evoluzione sinusoidale, per questo nella trattazione teorica che segue considereremo matrici S di sistemi oscillanti d -dimensionali.

3.2 Analisi del problema multivariabile.

Sia $x_k \in \mathbb{R}^p$ il vettore degli stati del k -esimo sistema del nostro network (composto da n sistemi). Come è noto, l'evoluzione libera di uno dei sistemi oscillatori è descritta dall'equazione differenziale $\dot{x}_k = Sx_k$, dunque considerando un network di n sistemi, definendo $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ vale la seguente equazione di evoluzione libera:

$$\dot{\mathbf{x}} = (I_n \otimes S)\mathbf{x}. \quad (3.2)$$

Per quanto riguarda l'ingresso di controllo, oltre a sfruttare le informazioni relative al grafo di comunicazione tramite la matrice L , sarà necessario definire una matrice di guadagni $K \in \mathbb{M}_{p \times p}$ sulla quale potere agire liberamente, ottenendo l'equazione complessiva:

$$\dot{\mathbf{x}} = [(I_n \otimes S) - (L \otimes K)]\mathbf{x} \quad (3.3)$$

in cui $[(I_n \otimes S) - (L \otimes K)]$ deve essere Hurwitz se vogliamo che il protocollo di consenso converga. Rimarchiamo il fatto che il *consensus* dovrà necessariamente avvenire su una traiettoria sinusoidale “intermedia” in ampiezza e fase tra quelle che esporrebbero naturalmente gli oscillatori del network: il protocollo di consenso non permette infatti di cambiare la “natura” dei sistemi su cui viene applicato, come è chiaro osservando la 3.3.

3.2.1 L’output consensus

Fino ad ora l’informazione scambiata tra i sistemi è sempre corrisposta alla differenza tra i loro stati interni: più realisticamente, i sistemi sono in grado di scambiarsi l’informazione relativa alla differenza tra le loro **uscite**, definite come opportune funzioni sugli stati. Ogni sistema sarà cioè descritto dal seguente set di equazioni:

$$\begin{cases} \dot{x}_k = Sx_k + u_k \\ y_k = Hx_k \end{cases} \quad (3.4)$$

con u_k ingresso di controllo definito come

$$u_k = -K \sum_{j=1}^N l_{kj}(t)y_j = - \sum_{j=1}^N l_{kj}(t)KHx_j$$

(contributo calcolato sulle uscite degli altri stati tramite la Laplaciana) e H la matrice di uscita

Dunque, in caso di output consensus l’equazione del set di sistemi controllati che consideriamo è:

$$\dot{\mathbf{x}} = [(I_n \otimes S) - (L \otimes KH)]\mathbf{x} \quad (3.5)$$

Con analogia al caso del *consensus* o *sincronizzazione* tramite differenza degli stati, se siamo in grado di scegliere K in modo che $[(I_n \otimes S) - (L \otimes KH)]$ sia Hurwitz, ci aspettiamo (come mostrato nel paragrafo 3.3.1) di raggiungere il consenso delle uscite sulla funzione:

$$\bar{\mathbf{y}}(t) = H e^{St} \bar{\mathbf{x}} \quad (3.6)$$

che espone ancora i modi di S , ma presenta il contributo della matrice H che seleziona quali stati sono disponibili in uscita per essere misurati.

3.3 La convergenza del protocollo di consenso su sistemi multivariabile.

Per riuscire a catturare e descrivere le proprietà di convergenza del protocollo di consenso nel caso multivariabile, è necessario ricondurre la matrice Laplaciana a forme più comode ed effettuare, corrispondentemente, un cambio di variabili all'interno del modello nello spazio degli stati. Definiamo tali trasformazioni:

Definizione 8 (Traformazione T sulla Laplaciana L). Si definisce come **trasformazione T sulla Laplaciana L** , una qualunque matrice T nella forma

$$T = \begin{pmatrix} 1 & \mathbf{0}_{1 \times (n-1)} \\ \mathbf{1}_{(n-1) \times 1} & I_{n-1} \end{pmatrix}$$

dove I_{n-1} è la matrice identità di dimensione $N - 1$ e conseguentemente tale che:

$$\tilde{L} = T^{-1} L T = \begin{pmatrix} 0 & \tilde{L}_{12} \\ 0 & \tilde{L}_{22} \end{pmatrix} \quad (3.7)$$

con $\tilde{L}_{12} \in \mathbb{C}^{N-1}$ vettore riga e $\tilde{L}_{22} \in \mathbb{M}_{(N-1) \times (N-1)}$.

Un caso specifico di *trasformazione T su L* , è quello che porta L nella sua forma di Jordan:

Definizione 9 (Traformazione T_J sulla Laplaciana L). Sia T_J la matrice che porta L nella sua forma canonica di Jordan \tilde{L} , cioè tale che $\tilde{L} = T_J^{-1} L T_J$: dal momento che T_J è formato da autovettori ed autovettori generalizzati di L , e che $\mathbf{1}_n$ è autovettore di L , possiamo scrivere che

$$T_J = (\mathbf{1}_n \quad \tilde{T}_J) \quad (3.8)$$

con \tilde{T}_J matrice $N \times (N - 1)$ composta di autovettori di L . Se consideriamo, senza perdita di generalità, che 0 sia il primo autovalore di L , possiamo anche scrivere

$$\tilde{L}_J = \begin{pmatrix} 0 & 0_{1 \times (N-1)} \\ 0_{(N-1) \times 1} & \tilde{\mathbf{L}}_{22_J} \end{pmatrix} \quad (3.9)$$

con $\tilde{\mathbf{L}}_{22_J}$ matrice $(N - 1) \times (N - 1)$ composta di blocchi di Jordan.

3.3.1 Trattazione attraverso una qualunque “Trasformazione T su L”

Si considera nel seguito un network di N sistemi multivariabile del tipo $\dot{x}_k = Sx_k + u_k$, $K = 1, \dots, N$ con grafo diretto di comunicazione sulle uscite descritto dalla matrice Laplaciana L . Riprendiamo la notazione introdotta in 3.3.

È possibile catturare la proprietà di convergenza del protocollo ricorrendo ad una generale “Trasformazione T su L”, come definita in *Definizione 8*.

Consideriamo il set controllato descritto dall’equazione :

$$\dot{\mathbf{x}} = [(I_n \otimes S) - (L \otimes KH)]\mathbf{x} \quad (3.10)$$

Se applichiamo adesso il cambio di coordinate $\mathbf{z} = (T^{-1} \otimes I_n)\mathbf{x}$, allora

$$\begin{aligned} \dot{\mathbf{z}} &= (T^{-1} \otimes I_n)\dot{\mathbf{x}} \\ &= (T^{-1} \otimes I_n)[(I_n \otimes S) - (L \otimes KH)]\mathbf{x} \\ &= (T^{-1} \otimes I_n)[(I_n \otimes S) - (L \otimes KH)](T \otimes I_n)\mathbf{z}. \end{aligned} \quad (3.11)$$

Sfruttando il fatto che $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ e ricordando che $\tilde{L} = T^{-1}LT$ otteniamo che

$$\begin{aligned}
 \dot{\mathbf{z}} &= (T^{-1} \otimes I_n)[(I_n \otimes S) - (L \otimes KH)](T \otimes I_n)\mathbf{z} \\
 &= (T^{-1} \otimes I_n)[(I_n \otimes S)(T \otimes I_n) - (T\tilde{L} \otimes KH)]\mathbf{z} \\
 &= (T^{-1} \otimes I_n)[(T \otimes S) - (T\tilde{L} \otimes KH)]\mathbf{z} \\
 &= [(I_n \otimes S) - (\tilde{L} \otimes KH)]\mathbf{z}
 \end{aligned} \tag{3.12}$$

ovvero un'espressione analoga a quella in (3.10) con \tilde{L} al posto di L .

Vista la particolare struttura di \tilde{L} evidenziata in (3.7), il sistema nelle coordinate \mathbf{z} diventa:

$$\begin{cases} \dot{z}_1 = Sz_1 - (L_{12} \otimes KH)\mathbf{z}_2 \\ \dot{\mathbf{z}}_2 = [(I_{n-1} \otimes S) - (\tilde{\mathbf{L}}_{22} \otimes KH)]\mathbf{z}_2 \end{cases} \tag{3.13}$$

dove $z_1 \in \mathbb{R}^p$ (p rappresenta il numero di stati del singolo sistema) e $\mathbf{z}_2 \in \mathbb{R}^{p(n-1)}$.

Dal momento che la prima colonna di T è $\mathbf{1}_n$, vale inoltre che

$$x_k = z_1 + T_k\mathbf{z}_2, \quad \forall k = 1, \dots, n. \tag{3.14}$$

per qualche matrice T_k . Vogliamo adesso mostrare che nel caso

$$[(I_{n-1} \otimes S) - (\tilde{\mathbf{L}}_{22} \otimes KH)]$$

sia Hurwitz, gli stati andranno a coincidere a regime con z_1 , analogamente al caso della trasformata di Jordan. Vale il seguente lemma

Lemma 1. *Si consideri il sistema triangolare a blocchi*

$$\begin{cases} \dot{z}_1 = Sz_1 - (L_{12} \otimes KH)\mathbf{z}_2 \\ \dot{\mathbf{z}}_2 = [(I_{n-1} \otimes S) - (\tilde{\mathbf{L}}_{22} \otimes KH)]\mathbf{z}_2 \end{cases}$$

con $z_1 \in \mathbb{R}^p$ e $\mathbf{z}_2 \in \mathbb{R}^{p(n-1)}$. Si supponga che

$$[(I_{n-1} \otimes S) - (\tilde{\mathbf{L}}_{22} \otimes KH)]$$

sia Hurwitz. Allora, esistono $M > 0$, $\alpha > 0$ ed una matrice $R_2 \in \mathbb{M}_{n_1 \times n_2}$ tali che per ogni coppia $z_1(0), \mathbf{z}_2(\mathbf{0})$ vale

$$\|z_1(t) - e^{St} \bar{z}_1\| \leq M e^{-\alpha t} \|\mathbf{z}_2(\mathbf{0})\|,$$

dove $\bar{z}_1 = z_1(0) + R_2 \mathbf{z}_2$

il quale ci assicura che z_1 tende esponenzialmente ad una traiettoria dipendente dai valori iniziali che espone i modi di S . Utilizzando il lemma (1) e considerando la (3.14), otteniamo subito la seguente conclusione:

Proposizione 1. *Si consideri il sistema controllato*

$$\dot{\mathbf{x}} = (I_n \otimes S)\mathbf{x} - (L \otimes KH)\mathbf{x} \quad (3.15)$$

e si supponga che

$$[(I_{n-1} \otimes S) - (\tilde{\mathbf{L}}_{22} \otimes KH)]$$

sia Hurwitz. Allora esistono $K > 0$, $\lambda > 0$ ed una matrice $R \in \mathbb{M}_{p \times np}$ tale che per ogni $\mathbf{x}(\mathbf{0})$, la k -esima componente dello stato \mathbf{x} dell'intero set di sistemi controllati soddisfa

$$\|x_k(t) - e^{St} \bar{x}\| \leq K e^{-\lambda t} \|\mathbf{x}(\mathbf{0})\|,$$

ovvero, in tali ipotesi anche gli stati nel primo sistema di coordinate raggiungeranno il consenso su una particolare soluzione di

$$\dot{\mathbf{x}} = S\mathbf{x},$$

soluzione che dipende da $\bar{x} = f(x_1(0), \dots, x_n(0))$ dove f è una opportuna funzione lineare.

Le eventuali uscite dei sistemi, che sono combinazione lineare degli stati tramite la matrice H , si sincronizzeranno invece lungo la funzione

$$\bar{y}(t) = He^{St}\bar{x}$$

come avevamo spiegato in precedenza.

3.3.2 Sintesi della matrice K

Come menzionato in 3.2.1, siamo interessati a rendere Hurwitz la matrice $[(I_n \otimes S) - (L \otimes KH)]$ in modo da ottenere l'*output consensus* dei nostri sistemi. È opportuno ricondurre il problema della sintesi della matrice K ad uno equivalente più semplice: riprendiamo a questo scopo la matrice

$$\tilde{L} = T^{-1}LT = \begin{pmatrix} 0 & L_{12} \\ 0 & \tilde{L}_{22} \end{pmatrix}$$

a cui ci eravamo ricondotti tramite T nel paragrafo 3.3.1. Sia poi $M \in \mathbb{M}_{(n-1) \times (n-1)}$ una matrice a valori complessi tale che

$$\Lambda = M\tilde{L}_{22}M^{-1}$$

scelta in modo che Λ sia diagonale; in tal caso i valori sulla diagonale di Λ corrispondono agli autovalori non nulli di L . La trasformazione $M^{-1} \otimes I_n$ porta la matrice 3.5 ad assumere la forma

$$(I_{n-1} \otimes S) - (\Lambda \otimes KH) \tag{3.16}$$

ed essendo Λ diagonale, otteniamo subito che la matrice (3.16) è diagonale a blocchi, con blocchi che hanno la seguente struttura:

$$S - \lambda_i(L)KH. \tag{3.17}$$

dove indichiamo con λ_i l' i -esimo autovalore di L .

Tale risultato ci porta immediatamente a formulare la seguente proposizione, grazie alla quale rendiamo più semplice il problema della sintesi di K:

Proposizione 2. *Gli autovalori della matrice $[(I_n \otimes S) - (L \otimes KH)]$ hanno tutti parte reale negativa se e solo se la matrice $S - \lambda_i(L)KH$ ha tutti gli autovalori a parte reale negativa, per ogni i .*

Utilizziamo adesso un approccio basato sulle funzioni di Lyapunov per rendere Hurwitz le matrici $A_i = S - \lambda_i(L)KH$: infatti i problemi di trovare delle funzioni di Lyapunov V_i sugli stati z_i tali che $\dot{V}_i < 0$ e quello di rendere Hurwitz le matrici $S - \lambda_i(L)KH$ sono in corrispondenza biunivoca.

La condizione da soddisfare seguendo questo approccio è che per ogni i , valga

$$z_i^T (PA_i^T + A_iP)z_i \leq -a\|z_i\|^2 \quad (3.18)$$

per un qualche $a < 0$, avendo definito come candidata funzione di Lyapunov

$$V = z_i^T P z_i$$

e chiamando P l'unica soluzione simmetrica definita positiva dell'equazione algebrica di Riccati (CARE):

$$SP + PS^T - 2\mu PH^T HP + aI_n = \mathbf{0}. \quad (3.19)$$

Tale soluzione P esiste sicuramente nel caso che S sia la matrice di un oscillatore lineare.

Si può verificare che in questo caso la scelta di $K = PH^T$ risolve il problema appena enunciato. Infatti, chiamando μ l'autovalore non nullo con la parte reale più piccola, utilizzando $K = PH^T$ nell'espressione della matrice A_i e ricordando che P è la soluzione dell'equazione CARE, vale

$$\begin{aligned}
z_i^T (PA_i^T + A_i P) z_i &= z_i^T (PS^T - \lambda_i PH^T HP^T + SP - \lambda_i PH^T HP) \\
&= z_i^T (SP + PS^T - \lambda_i PH^T H(P^T + P)) z_i \\
&= z_i^T (SP + PS^T - 2\operatorname{Re}(\lambda_i) PH^T HP) z_i \\
&\leq z_i^T (SP + PS^T - 2\mu PH^T HP) z_i = -a \|z_i\|^2
\end{aligned} \tag{3.20}$$

e dunque abbiamo soddisfatto la (3.18); come abbiamo fatto notare sopra, abbiamo congiuntamente reso Hurwitz tutte le matrici $S - \lambda_i KH$ e conseguentemente l'intera matrice $(I_{n-1} \otimes S) - (\Lambda \otimes KH)$.

Riassumiamo questi risultati nella seguente proposizione:

Proposizione 3. *Sia dato un set di N sistemi del tipo descritto in equazione (3.4). Si supponga che (S, H) sia una coppia osservabile. Si supponga che il grafo di comunicazione (tempo-invariante) sia connesso. Sia l'ingresso di controllo dato da*

$$u_k = PH^T \sum_{j=1}^N a_{kj} (y_j - y_k) \quad k = 1, \dots, N \tag{3.21}$$

In tal caso, si raggiunge l'output consensus.

3.4 Trattazione attraverso trasformata di Jordan T_J .

Se vogliamo ottenere delle equazioni semplificate rispetto a quelle in 3.13, possiamo ricorrere alla trasformazione di Jordan T_J .

Esprimiamo ancora le equazioni dei nostri sistemi controllati in forma compatta

$$\dot{\mathbf{x}} = [(I_N \otimes S) - (L \otimes KH)] \mathbf{x} \tag{3.22}$$

Effettuiamo dunque il seguente cambio di variabili sugli stati sfruttando la trasformazione di Jordan ed il prodotto di Kronecker:

$$\mathbf{z} = (T_J^{-1} \otimes I_d) \mathbf{x} \tag{3.23}$$

ed otteniamo

$$\dot{\mathbf{z}} = [(I_N \otimes S) - (\tilde{L} \otimes KH)]\mathbf{z} \quad (3.24)$$

dove, ricordiamo, \tilde{L}_J rappresenta la forma canonica di Jordan di L . Tenendo presente la particolare struttura di \tilde{L}_J evidenziata in 3.9, chiamando $\mathbf{z} = \text{col}(z_1, \mathbf{z}_2)$, $\mathbf{z}_2 = \text{col}(z_2, \dots, z_N)$, $z_i \in \mathbb{R}^d$, possiamo riscrivere la 3.24 nel seguente modo:

$$\begin{cases} \dot{z}_1 = Sz_1 \\ \dot{\mathbf{z}}_2 = [(I_{N-1} \otimes S) - (\tilde{\mathbf{L}}_{\mathbf{z}_2} \otimes KH)]\mathbf{z}_2 \end{cases} \quad (3.25)$$

Dunque z_1 ha un'evoluzione indipendente dalle altre variabili. Inoltre è possibile scegliere K in modo che il sistema con stato \mathbf{z}_2 sia *Hurwitz*, cioè che esponga modi che si annullano asintoticamente per $t \rightarrow +\infty$, ovvero che $\mathbf{z}_2(t)$ converga esponenzialmente a 0 per ogni $\mathbf{z}_2(0)$.

L'ultimo passaggio consiste nell'osservare che grazie alla particolare struttura di T_J ed al cambio di variabili effettuato vale

$$x = (1_{N \times 1} \otimes I_d)z_1 + (\tilde{T}_J \otimes I_d)\mathbf{z}_2 \quad (3.26)$$

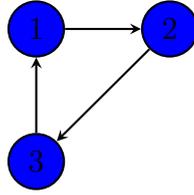
Dunque, dal momento che \mathbf{z}_2 tende ad annullarsi, si è dimostrato che tutti i x_k raggiungono il consenso sulla traiettoria di z_1 che coincide con x_1 , mettendo in evidenza l'effettiva convergenza del protocollo di consenso su una stessa traiettoria finale.

3.5 Analisi attraverso trasformata di Jordan su un caso specifico.

Consideriamo in questa sezione il caso di 3 sistemi oscillatori descritti dalla matrice di sistema

$$S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

ed interconnessi tramite il *grafo di comunicazione* ciclico



la cui matrice Laplaciana è

$$L = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

Come visto nella sottosezione 3.4, è opportuno portare L nella sua forma di Jordan \tilde{L} per capire verso cosa convergeranno gli stati di tutti i sistemi una volta applicato il controllo di consenso.

Si ha che la matrice di similitudine è

$$T_J = \begin{pmatrix} 1 & -0.5 - 0.866i & -0.5 + 0.866i \\ 1 & -0.5 + 0.866i & -0.5 - 0.866i \\ 1 & 1 & 1 \end{pmatrix}$$

e la Laplaciana diventa così

$$\tilde{L}_{22_J} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.5 - 0.866i & 0 \\ 0 & 0 & 1.5 + 0.866i \end{pmatrix}$$

Come mostrato in 3.4, è allora possibile effettuare il cambio di variabili sugli stati definito da $\mathbf{z} = (T_J^{-1} \otimes I_2)\mathbf{w}$ ed ottenere il seguente modello nello spazio degli stati

$$\begin{cases} \dot{z}_1 = Sz_1 \\ \dot{\mathbf{z}}_2 = [(I_{N-1} \otimes S) - (\tilde{L}_{22_J} \otimes K)]\mathbf{z}_2 \end{cases}$$

come avevamo già osservato in (3.25).

Se assumiamo che la matrice che descrive l'evoluzione di $\dot{\mathbf{z}}_2$ sia Hurwitz (attraverso la scelta di una opportuna K), possiamo anche osservare, riprendendo il risultato in (3.26), che

$$\mathbf{x} = (1_{N \times 1} \otimes I_d)z_1 + (\tilde{T}_J \otimes I_d)z_2 \quad (3.27)$$

Ciò significa, visto che $\dot{\mathbf{z}}_2$ tende ad annullarsi, che gli stati x_{1_i}, x_{2_i} (posizione e velocità) dei 3 oscillatori presi in considerazione andranno a consenso sulla traiettoria di $z_1 = \text{col}(z_{1_i}, z_{2_i})$ per qualunque set di valori iniziali.

Infatti, la trasformazione degli stati \mathbf{x} verso gli stati \mathbf{z} risulta essere:

$$T_J^{-1} \otimes I_2 = \begin{pmatrix} \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ -0.1667 + 0.2887i & 0 & -0.1667 - 0.2887i & 0 & \frac{1}{3} & 0 \\ 0 & -0.1667 + 0.2887i & 0 & -0.1667 - 0.2887i & 0 & \frac{1}{3} \\ -0.1667 - 0.2887i & 0 & -0.1667 + 0.2887i & 0 & \frac{1}{3} & 0 \\ 0 & -0.1667 - 0.2887i & 0 & -0.1667 + 0.2887i & 0 & \frac{1}{3} \end{pmatrix}$$

e dunque

$$z_1(t) = \frac{(x_1(t) + x_2(t) + x_3(t))}{3} \quad \forall t \in \mathbb{R}^+. \quad (3.28)$$

A regime, se il consenso è stato raggiunto (dato per assunto dal momento che la matrice che descrive l'evoluzione di $\dot{\mathbf{z}}_2$ è assunta Hurwitz), avremo, ricorrendo ad un piccolo abuso di notazione, che $x_{1_\infty} = x_{2_\infty} = x_{3_\infty}$ e dunque che $\frac{1}{3}(x_{1_\infty} + x_{2_\infty} + x_{3_\infty}) = x_{1_\infty} = x_{2_\infty} = x_{3_\infty} = z_{1_\infty}$.

Le osservazioni appena effettuate rendono evidente che per ogni sistema k , x_k va a consenso sulla traiettoria di z_1 , la cui evoluzione temporale è descritta dall'equazione

$\dot{z}_1 = Sz_1$: tale traiettoria espone i modi corrispondenti agli autovalori della matrice di sistema S , che nel nostro caso specifico sono modi oscillanti.

3.5.1 Simulazione: output consensus su 3 oscillatori armonici.

Lo scopo della seguente simulazione è verificare l'efficacia della procedura di sintesi della matrice K descritta in 3.3.2, al fine di mandare a consenso i 3 oscillatori di cui si è parlato in 3.5. Sempre in 3.5, attraverso un'analisi del sistema controllato tramite trasformata di Jordan, si era arrivati alla conclusione che gli oscillatori sarebbero andati a consenso su di una traiettoria sinusoidale di pulsazione 1 rad/sec ed ampiezza e fase dipendenti dai valori iniziali degli oscillatori; cogliamo quindi questa occasione per verificare anche tale risultato.

Il modello Simulink di riferimento è identico a quello presentato in 2.4.1, ad eccezione del fatto che al posto del blocco di guadagno corrispondente alla Laplaciana è stata posta la matrice $U = [(I_3 \otimes S) - (L \otimes KH)]$, con $K = PH^T$ (vedi paragrafo 3.3.2).

I valori iniziali di di posizione e velocità dei 3 oscillatori sono stati posti, a titolo di esempio, pari a (1, 2), (3, 3) e (5, -5). La matrice H , che come abbiamo detto rappresenta la trasformazione tra stati ed uscite, è stata posta pari a $H = \begin{pmatrix} 1 & 1 \\ 10 & 10 \end{pmatrix}$: ogni sistema avrà cioè due uscite, la prima corrispondente alla somma tra posizione e velocità, la seconda pari a 10 volte la somma tra posizione e velocità.

La procedura per calcolare K è stata implementata nel relativo script di Matlab: in particolare otteniamo che $P = \begin{pmatrix} 0.5016 & -0.4985 \\ -0.4985 & 0.5797 \end{pmatrix}$ e dunque $K = \begin{pmatrix} 0.0032 & 0.0317 \\ 0.0812 & 0.8118 \end{pmatrix}$, ponendo $a = 1$ e $\mu = 1.5$, cioè pari alla parte reale del secondo autovalore di L . Si noti che la scelta di a influisce sulla velocità di convergenza del protocollo.

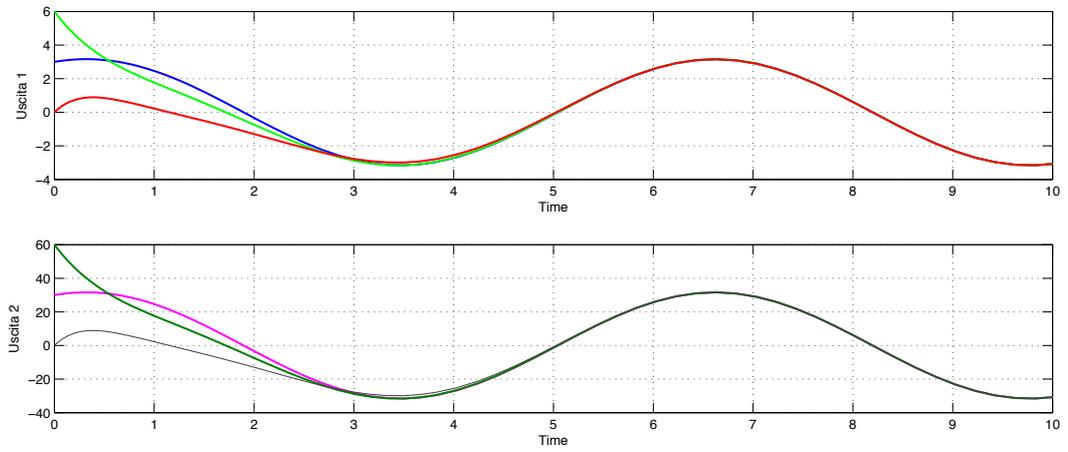


Figura 3.1: Risultato della simulazione

In figura 3.1 il risultato della simulazione: nei due plot viene riportato il consenso rispettivamente sulla prima e sulla seconda uscita.

Come previsto, le due uscite dei tre sistemi vanno a consenso su due traiettorie di pulsazione 1 rad/sec.

Capitolo 4

Generazione di traiettorie circolari per il *loitering* di più robot.

Il coordinamento di sistemi multi-robot ha come obiettivo quello di ottenere delle prestazioni migliori rispetto all'uso di singoli agenti, garantendo una maggiore robustezza e performance generale di sistema. Diventa infatti possibile risolvere sotto-problemi in parallelo, riducendo sia il tempo per il completamento del compito dell'intero sistema che il carico computazionale. Inoltre, l'assenza di un singolo "master" decisionale rende il sistema distribuito più robusto e resistente ai guasti (il fallimento di un singolo agente non implica il fallimento dell'intero network).

Tuttavia, nell'utilizzo di sistemi multi-agente emergono nuovi problemi e difficoltà:

- Coordinare un numero elevato di agenti
- Gestire un'informazione locale e non centralizzata
- Gestire l'elevata necessità di comunicazione

Il consenso fornisce alcuni possibili vantaggi relativamente a questi problemi, rispetto ad altre tecniche comunemente utilizzate in letteratura. In *primis* è completamente decentralizzato e leader-less di natura, inoltre si basa su uno scambio di informazioni semplice e contenuto.

L'idea di fondo è quella di associare ai nostri agenti fisici dei modelli di sistemi dinamici chiamati *esosistemi*, che si comporteranno come generatori di traiettorie decentralizzati per i singoli agenti; sugli esostemi si utilizzano i protocolli di consenso e mediante l'*agreement* raggiunto si ottiene il comportamento collettivo desiderato.

Come applicazione dei risultati provenienti dalla teoria sviluppata in questo lavoro, si è dunque pensato al caso del raggiungimento del consenso di più veicoli su di una traiettoria circolare. In particolare si vuole che tali agenti, qualunque siano le loro posizioni iniziali nello spazio, vadano a circolare su di una traiettoria di dato raggio e centro con velocità impostata, facendo inoltre in modo che a regime le loro posizioni rappresentino i vertici di un poligono regolare. Queste richieste sono quelle tipicamente necessarie in un contesto di *loitering* attorno ad un agente/punto dello spazio obiettivo: si richiede ai droni di pattugliare l'area circostante volando in circolo ed eventualmente di seguire l'obiettivo se questi è in moto.

In questo capitolo si vuole descrivere l'approccio seguito per realizzare la simulazione in ambiente Simulink, simulazione che è poi stata utilizzata per generare le traiettorie di riferimento di tre quadricotteri per prove sperimentali in laboratorio: i risultati di questi esperimenti sono riportati nel capitolo 5.

Il problema di controllo in questione si presenta come l'unione di due problemi di consenso, trattati nei precedenti capitoli:

- **Consenso di più oscillatori lineari:** necessario per il raggiungimento di una traiettoria circolare comune, interpretando i due stati di ogni oscillatore come posizioni X e Y sul piano. La circonferenza di regime ottenuta mediante il protocollo dipenderà dalle condizioni iniziali degli oscillatori. È però possibile modificare il protocollo inserendo un oscillatore "fittizio" che permetta di impostare a piacere raggio e velocità di regime.
- **Consenso di più integratori:** necessario per il raggiungimento di un centro

di traiettoria comune. Il valore di convergenza di questo protocollo di consenso dipenderà, anche in questo caso, dalle condizioni iniziali degli agenti. Tuttavia, è possibile modificare tale protocollo (mediante l'aggiunta di un agente "fittizio") per poter impostare il centro della circonferenza a piacere. Sarà necessario un controllo di consenso sia per la posizione X che per la posizione Y del centro.

Riportiamo in figura 4.1 il modello di simulazione visto dal livello più esterno: come anticipato, al fine di ottenere il comportamento collettivo desiderato si utilizzano due protocolli di consenso. Un primo protocollo (CONSENSO OSCILLATORI, in arancione) per impostare la traiettoria dinamica da eseguire. Un secondo protocollo di consenso (CONSENSO INTEGRATORI (CENTRO)) per imporre il centro attorno a cui gli agenti devono compiere la traiettoria dinamica (o di *loitering*) desiderata. Nel seguito si descriveranno più nel dettaglio gli accorgimenti presi per realizzare i due rami del consenso degli oscillatori e degli integratori.

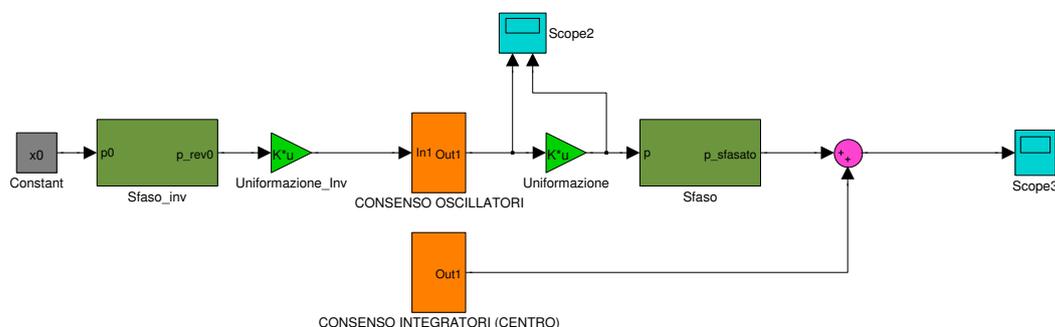


Figura 4.1: Modello: alto livello.

4.1 Consenso degli oscillatori.

Il nostro obiettivo è quello di poter realizzare una traiettoria circolare di raggio e velocità desiderate: vogliamo inoltre che questi due parametri siano imponibili a piacere, compatibilmente con i limiti cinematici degli agenti in esame.

Essendo il protocollo di consenso decentralizzato per natura, i valori/traiettorie che gli agenti raggiungono dipendono esclusivamente dalle condizioni iniziali. Per superare questo problema immaginiamo che il centro attorno al quale si compie la traiettoria circolare di *loitering* sia in realtà un agente fittizio, che “comunica” agli altri agenti raggio e velocità desiderate: in pratica, si aggiunge un *agente virtuale* al network, si rende connesso il grafo risultante avendo cura che l’agente virtuale risulti un centroide indipendente, e si estende il protocollo di consenso ad $n + 1$ agenti (dove n è il numero di droni che compongono la formazione reale). Il risultato finale sarà che tutti gli oscillatori reali andranno a consenso sull’oscillatore virtuale.

Si noti che questa è una soluzione “di comodo” che potrebbe essere in realtà ottenuta mediante altri approcci. Per imporre un raggio arbitrario sarebbe sufficiente modificare il classico oscillatore lineare, aggiungendo un termine *non-lineare* che forzi il sistema sul ciclo limite desiderato. Tale modifica condurrebbe però all’analisi del consenso per sistemi non-lineari, ambito che esula dagli obiettivi di questa tesi .

Le uscite sono state considerate coincidenti con gli stati degli oscillatori (H è quindi la matrice identità) che vengono interpretati come le posizioni X e Y degli agenti.

Fatte queste considerazioni preliminari, abbiamo supposto che gli agenti comunicassero tra loro secondo una certa topologia connessa, ed è stato creato uno script Matlab che si occupa di:

- Generare la Laplaciana L connessa associata a tale grafo.
- Effettuare la sintesi della matrice K necessaria al raggiungimento del consenso (si veda 3.3.2)
- Creare la matrice $U = [(I_n \otimes S) - (L \otimes KH)]$ da utilizzare nel modello Simulink.

All’interno del blocco CONSENSO OSCILLATORI troviamo uno schema chiuso in retroazione del tutto analogo a quelli utilizzati nelle simulazioni del capitolo 2, eccezion fatta per la matrice U al posto della matrice L .

La matrice

$$Uniformazione = I_n \otimes \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{w}} \end{pmatrix}$$

ha lo scopo di uniformare in ampiezza i due stati di ogni oscillatore. Se infatti la posizione segue una traiettoria sinusoidale di pulsazione \sqrt{w} ed ampiezza A , allora la velocità segue una traiettoria co-sinusoidale di pulsazione ancora \sqrt{w} , ma di ampiezza $\sqrt{w}A$. Dunque tra i due stati dell'oscillatore, che vengono interpretati come coordinate X e Y , corre un fattore moltiplicativo pari a \sqrt{w} che bisogna eliminare attraverso la matrice sopra introdotta. Notiamo che senza la matrice di uniformazione le traiettorie sarebbero delle ellissi anziché delle circonferenze.

4.1.1 Sfasamento degli agenti.

Per il raggiungimento di una formazione, non solo è necessario che gli agenti seguano la stessa traiettoria circolare a regime, ma si richiede anche che si dispongano secondo una ben precisa geometria. Con il solo consenso di oscillatori descritto in 4.2, infatti, gli agenti andrebbero a coincidere nelle posizioni, fatto che in una realizzazione fisica su veicoli corrisponderebbe ad uno scontro: si impongono dunque degli opportuni sfasamenti alle traiettorie dei singoli agenti, in modo che vadano a disporsi sulla circonferenza target secondo i vertici di un poligono regolare. Al fine di raggiungere questo obiettivo, è stato sufficiente osservare che a regime i due stati degli oscillatori seguono un andamento temporale del tipo

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} \sin(\omega t + \phi) \\ \cos(\omega t + \phi) \end{pmatrix} \quad (4.1)$$

e che dunque è possibile combinarli algebricamente secondo le formule di addizione/sottrazione di seno e coseno per imporre un qualunque sfasamento alle traiettorie di regime (trasformandole cioè in $\sin(\omega t + \phi + \Psi)$, $\cos(\omega t + \phi + \Psi)$ per qualunque Ψ). Si è quindi inserita una funzione embedded nel modello con lo scopo di effettuare queste manipolazioni sia a valle del blocco *CONSENSO OSCILLATORI* (trasformazione diretta sulle traiettorie) che a monte (trasformazione inversa sulle posizioni iniziali per far sì

che le traiettorie sfasate partano dalle posizioni iniziali originarie).

Potrebbe sembrare che questi sfasamenti siano realizzati in maniera centralizzata; in realtà ad ogni agente è associato un proprio sfasamento, e di conseguenza ogni agente conosce, indipendente da cosa devono fare gli altri, il valore dello sfasamento che deve imporre alla traiettoria di consenso che sta ottenendo dallo scambio di informazioni.

Nelle figure 4.2 e 4.3 riportiamo il consenso degli oscillatori prima e dopo l'applicazione dello sfasamento.

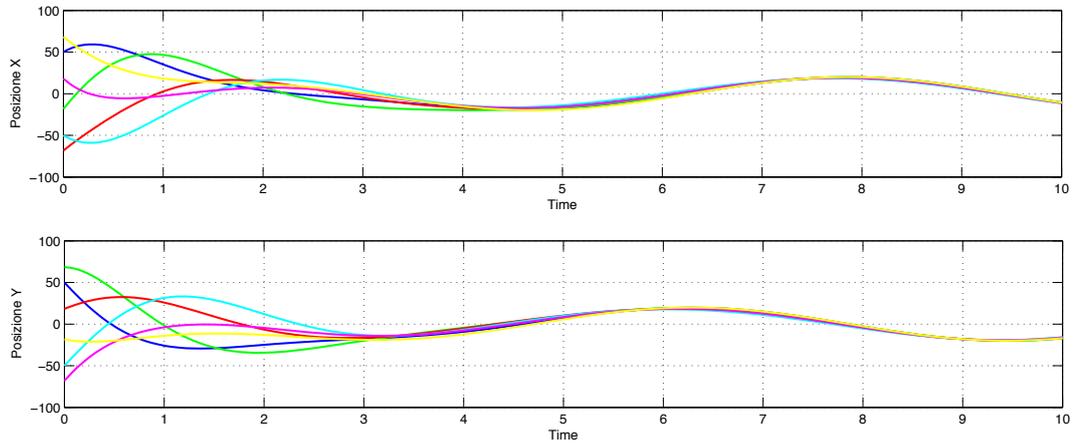


Figura 4.2: Traiettorie prima dello sfasamento

4.2 Consenso degli integratori.

Al fine di fare raggiungere alla traiettoria circolare un centro di formazione diverso da $(0,0)$, risulta necessario sommare alle traiettorie di tutti gli agenti un contributo che deriva dal consenso di integratori associati alle loro rispettive posizioni X_{centro} e Y_{centro} verso due *integratori virtuali* che contengono le informazioni sui target $(X_{centro}^{ref}, Y_{centro}^{ref})$: come nel caso dell'*oscillatore virtuale*, gli integratori virtuali dovranno comportarsi da *centroidi* indipendenti nel grafo di comunicazione complessivo, in modo che il loro valore non venga mai influenzato e che tutti gli altri integratori convergano su di loro a regime.

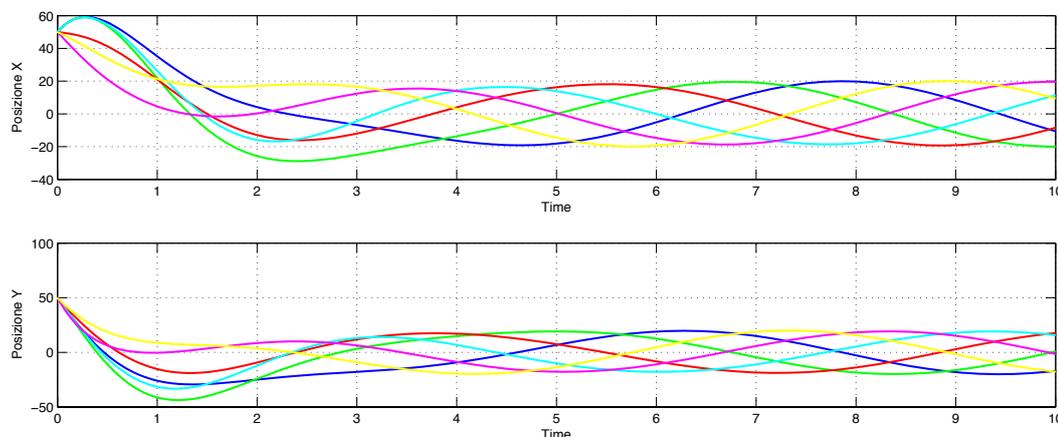


Figura 4.3: Traiettorie dopo lo sfasamento.

Come già è stato fatto nel capitolo 2, nel caso degli integratori è sufficiente implementare un loop algebrico controllato dalla matrice L , in modo da realizzare l'equazione differenziale

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \end{pmatrix} = -L \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{x}(0) = [0 \ 0 \ \dots \ x_{ref}]' \\ \mathbf{y}(0) = [0 \ 0 \ \dots \ y_{ref}]' \end{pmatrix}$$

4.3 Simulazione della generazione di traiettorie.

Per una maggiore comprensione della simulazione, si riportano in figura 4.4 le traiettorie in X e in Y di sei agenti nelle seguenti ipotesi:

- **Condizioni iniziali :** Tutti e sei gli agenti partono dalle coordinate (50,50).
- **Raggio della traiettoria 10.**
- **Pulsazione di percorrenza:** tutti gli oscillatori hanno $\omega = 1rad/sec$ dentro alla matrice di sistema.
- **Richieste sul centro:** La richiesta iniziale sulle coordinate è (100,100), per poi passare ogni 15 secondi al successivo vertice di un quadrato di lato 200. La sequenza

di richieste sul centro è quindi ciclicamente: $(100, 100)$, $(100, -100)$, $(-100, -100)$, $(-100, 100)$...

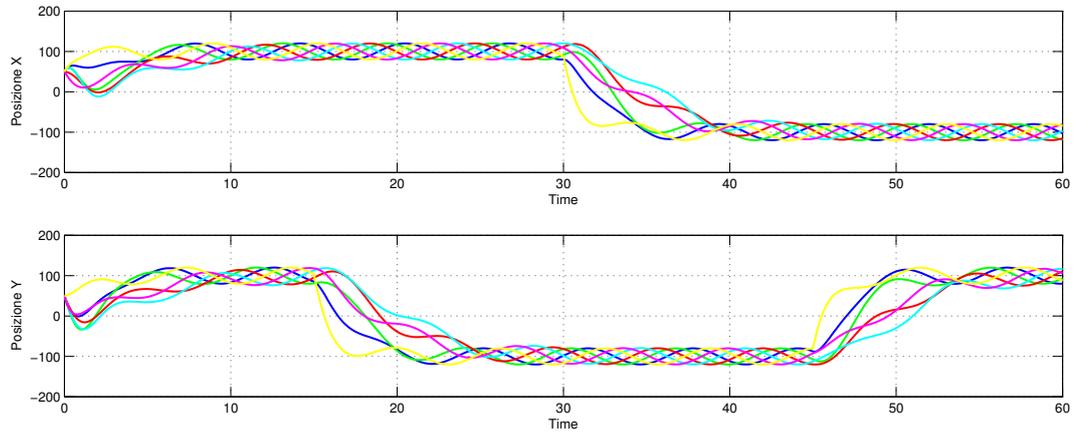


Figura 4.4: Traiettorie X e Y con cambio di richiesta sul centro ogni 15 secondi.

Capitolo 5

Il setup di laboratorio e lo schema della Ground Station.

L'obiettivo dell'esperienza di laboratorio è stato quello di realizzare il consenso di tre quadricotteri CrazyFlie su una traiettoria assoluta circolare di raggio impostato e velocità decisa a priori, utilizzando come centro mobile una macchina telecomandata o un qualunque altro agente in movimento.

Un nano-quad CrazyFlie è composto solamente da un PCB (Printed Circuit Board) dove sono saldati i quattro motori con rispettive eliche e una batteria per l'alimentazione. CrazyFlie è un sistema *open-hardware open-software*. La batteria è da $170mAh$ con un tempo di ricarica di circa 20 minuti e permette un volo di 5 minuti. Il peso è attorno ai 10 grammi.

Per realizzare l'architettura software necessaria alle prove di laboratorio si è inoltre utilizzato:

- **Sistema di acquisizione OptiTrack:** Il sistema è composto da diverse telecamere disposte lungo il perimetro del soffitto all'interno del laboratorio di volo. Utilizzando gli infrarossi emessi da ogni telecamera, **Optitrack** riesce a stimare con errori molto piccoli la posizione di un oggetto. Un limite che pone l'utilizzo di questo sistema è dato dal fatto che ogni oggetto deve essere dotato di 3 *marker*



Figura 5.1: Quadrirotori CrazyFlie.

per essere identificato. Nel nostro caso si utilizzano 3 palline tonde rivestite con un materiale riflettente, fissate ad una struttura di carbonio che viene poi attaccata ai quadrirotori. L'insieme di *marker* e struttura sorreggente, arriva a pesare qualche grammo e va quindi ad aumentare il peso del CrazyFlie del 30-40%. Tale aumento compromette sensibilmente la dinamicità dei CrazyFlie durante il volo: i motori non sono molto potenti e la portanza (spinta verticale) che generano viene ridotta dal fatto che le asticelle della struttura possono bloccare il flusso d'aria generato dalle pale. Per questo motivo l'inseguimento delle traiettorie target risulta efficace, ma le imprecisioni e gli errori parametrici fanno sì che ci sia sempre e comunque un errore a regime.

Per passare i dati dal sistema **Optitrack** a **Matlab**, una volta lanciato il software di tracking, si utilizza un programma locale chiamato *NatNet*, il quale invia i dati su una porta UDP letta da Matlab: attraverso il modello della **Ground Station**, si utilizzano le informazioni così ricevute per la generazione di traiettorie ed il controllo. (Figura 5.3).

- **Modello Matlab/Simulink della Ground Station:** si occupa di generare le



Figura 5.2: I 3 marker e la struttura in carbonio.

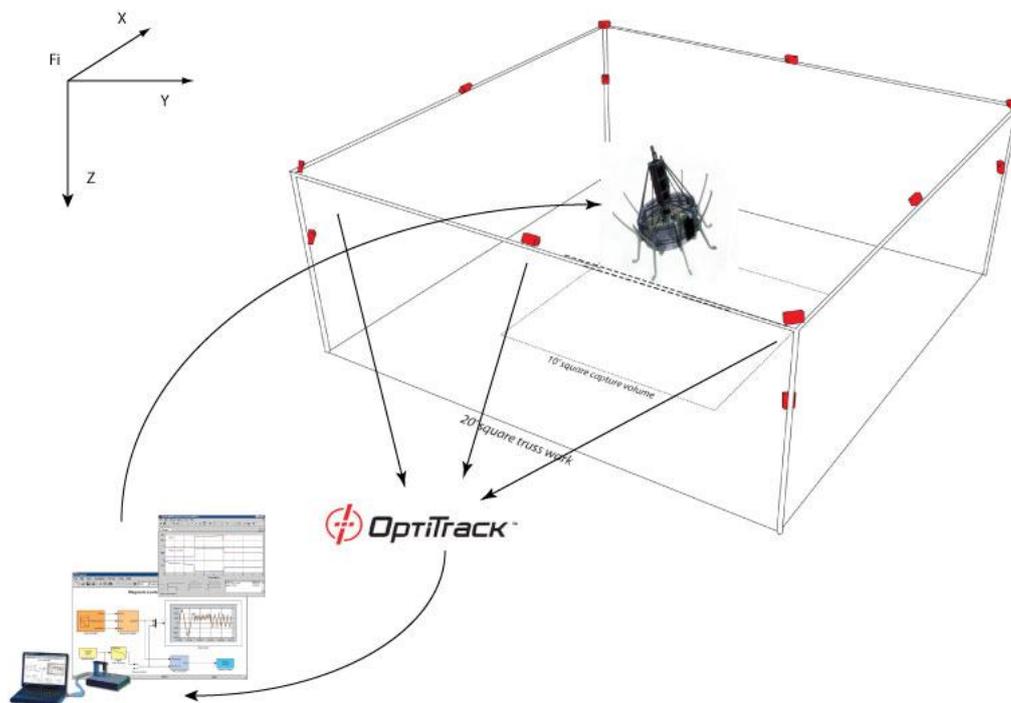


Figura 5.3: Optitrack.

traiettorie di riferimento e di garantire il controllo dei quadrotori al fine di inseguire il riferimento generato. Verrà descritto in modo più approfondito nella prossima sezione.

- **Antenne:** per inviare gli ingressi di controllo generati dalla Ground Station ai singoli quadrotori è stato necessario sfruttare delle radio-antenne con porta USB.



Figura 5.4: Periferica di trasmissione dati al nano-quad.

- **Joypad:** dopo averlo collegato ad una porta USB, bisogna lanciare un programma (chiamato *Joypad Parser*) il cui compito è quello di prendere i pacchetti di dati relativi alla pressione di pulsanti e comunicarli a Matlab mediante pacchetto UDP. Il joypad verrà sfruttato per cambiare la modalità di funzionamento: si vuole infatti che i quadrotori decollino e si portino alla stessa quota prima di entrare in modalità di inseguimento del consenso.

5.1 La Ground Station

Si è sfruttato un modello Simulink preesistente, chiamato Ground Station, che permette di utilizzare i dati acquisiti dal sistema OptiTrack e da un Joystick per generare gli ingressi di controllo dei quadrotori (figura 5.5) . Lo schema prevede la presenza di una macchina a stati il cui scopo è dare la possibilità all'utente di selezionare una mo-

- Il blocco “Consensus” (in rosso), a valle, che genera i riferimenti da inseguire.
- I blocchi “Controller” (in blu), che generano gli effettivi ingressi di controllo.
- I blocchi “Optitrack” (in verde), che forniscono ai blocchi Controller le informazioni su posizione e velocità dei Crazyflie istante per istante.
- I blocchi di uscita (in arancione), che trasmettono l’informazione sugli ingressi di controllo ai CrazyFlie tramite l’apposita radio-antenna.

5.2 I risultati di laboratorio.

Riportiamo in questa sezione alcuni grafici che mostrano le traiettorie di riferimento generate e le traiettorie effettivamente percorse dai Crazyflie.

I riferimenti

Si deve inizialmente attendere che i quadrirotori si portino stabilmente alla stessa quota (riferimenti per X e Y piatti nei primi secondi). Premendo un pulsante sul Joystick, il sistema entra in modalità “consenso” e Optitrack comunica a Matlab le posizioni iniziali degli agenti. Basandosi su questa informazione, Matlab genera le traiettorie che portano i Crazyflie a regime.

Si riportano nei plot le traiettorie di riferimento per la X e la Y, mettendo in evidenza il ruolo dello sfasamento che ogni agente impone su se stesso.

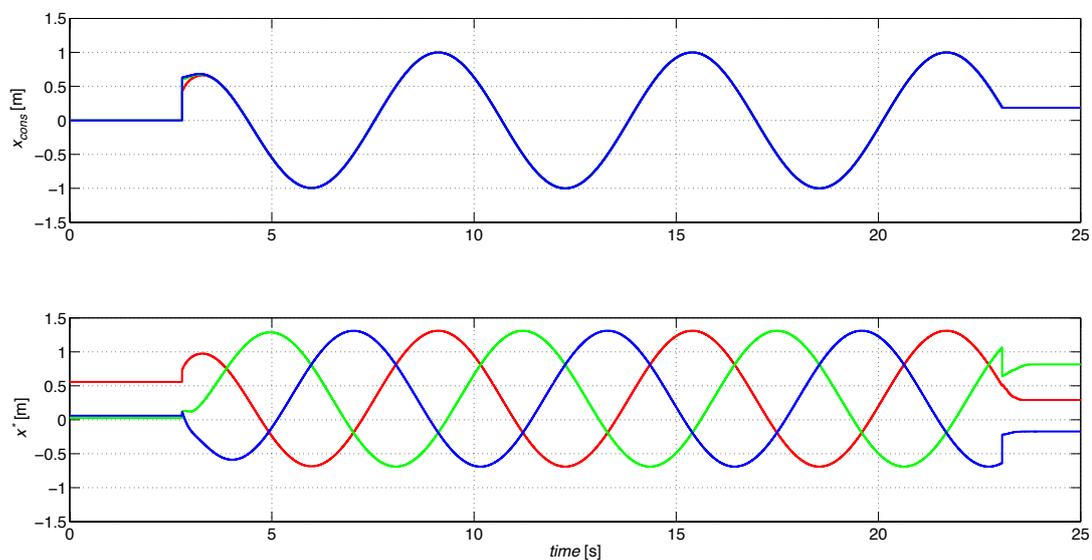


Figura 5.6: Riferimento per la X, prima e dopo sfasamento.

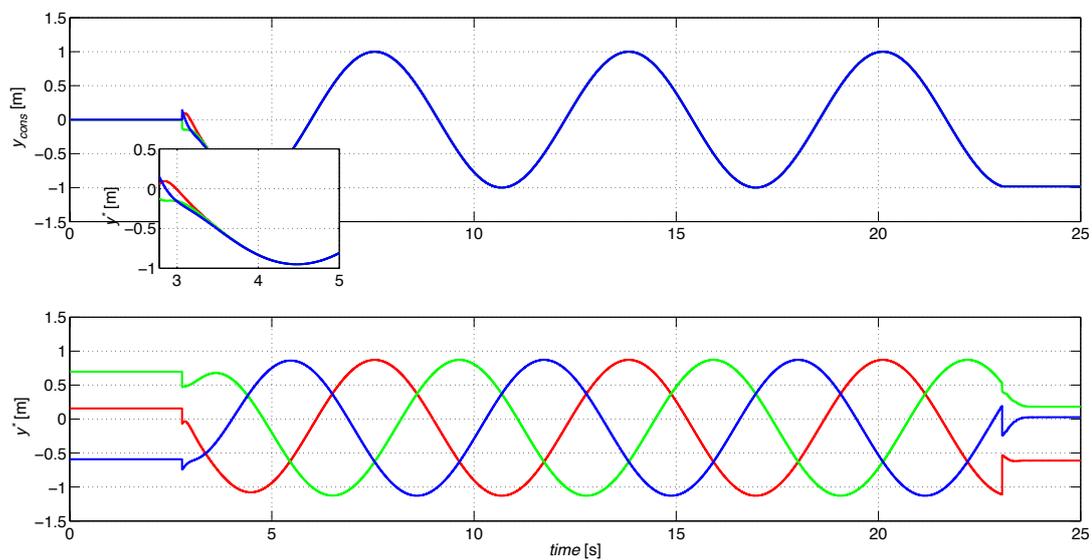


Figura 5.7: Riferimento per la Y, prima e dopo sfasamento.

L'inseguimento dei riferimenti

Come spiegato in precedenza, è impossibile ottenere un errore nullo nell'inseguimento delle traiettorie, a causa degli inevitabili errori parametrici e della presenza della struttura che sorregge i marker. Il discostamento dalla traiettoria ideale, come vediamo in figura, risulta però contenuto e la performance è nel complesso soddisfacente. Si riporta la traiettoria effettiva di un singolo quadrirotore ed il suo riferimento ideale.

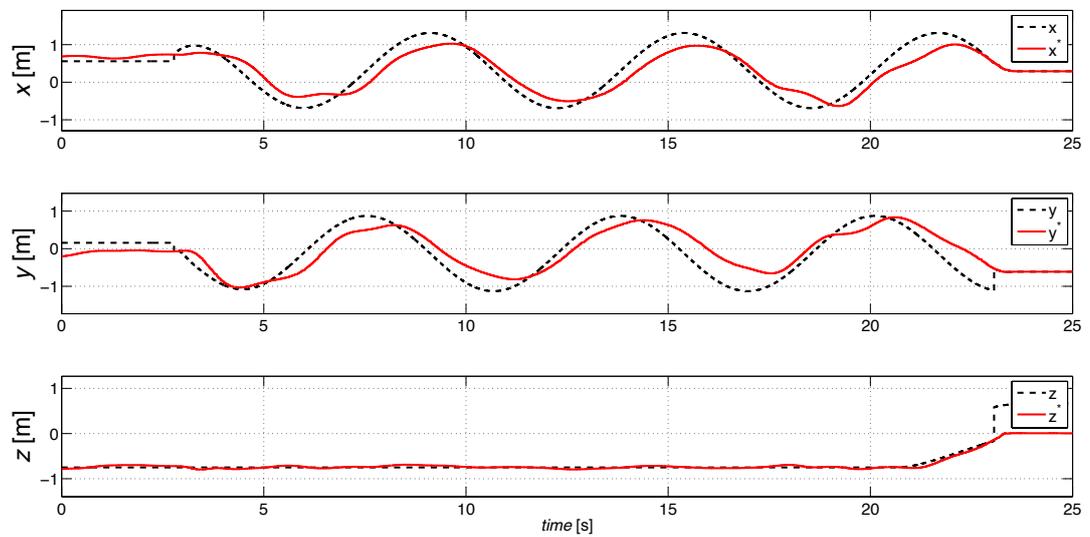


Figura 5.8: Tracking della traiettoria di riferimento per un quadrirotore.

Conclusioni

Abbiamo nel corso di questo lavoro soddisfatto gli obiettivi che ci eravamo posti.

Nei primi tre capitoli si è trattato da un punto di vista prettamente teorico il problema del consenso e della convergenza del protocollo di consenso nei casi mono e multivariabile, utilizzando un approccio controllistico basato sull'analisi tramite modelli nello spazio degli stati. Affiancando la spiegazione dell'efficacia dei metodi per risolvere i problemi mono e multivariabile, si sono effettuate varie simulazioni in ambiente Matlab/Simulink per trovare riscontro dell'effettiva realizzabilità di quanto mostrato nella teoria.

Una simulazione, non più meramente dimostrativa, è stata realizzata anche nel capitolo 4. Si è dovuto pensare ad un modo di sfruttare congiuntamente i modelli di sistemi oscillatori ed integratori per creare il riferimento di traiettorie circolari di centro instabile, avendo in mente l'applicazione al caso del *loitering* di un sistema multi-robot: l'aspetto più originale sta nell'aver semplicemente unito sistemi eterogenei per risolvere un problema che inizialmente sembrava richiedere l'introduzione di un terzo di tipo di sistema più complesso (non è stato cioè necessario trattare il consenso di oscillatori non-lineari con un offset, anche se sarebbe stato possibile agire in tal senso per ottenere gli stessi risultati).

Tale modello è stato inserito in quello della Ground Station (capitolo 5) permettendo così, prendendo opportuni accorgimenti implementativi, di sperimentare fisicamente il coordinamento di gruppo in laboratorio.

Si noti che la parte di computazione dei riferimenti è completamente realizzata dal calcolatore: sarebbe comunque possibile e semplice, se i quadrirotori avessero sufficiente potenza di calcolo, distribuire il *workload* computazionale sui singoli quadrirotori, rendendo quindi effettiva e non simulata la comunicazione tra i vari agenti.

Sviluppi futuri.

Non si è previsto un modo di evitare le eventuali collisioni tra gli agenti durante il transitorio del consenso: le traiettorie vengono generate garantendo soltanto la convergenza sulla circonferenza desiderata, senza imporre ulteriori vincoli. Per questo motivo, in laboratorio ci si è curati di posizionare i quadrirotori in posizioni iniziali “non rischiose”. Tuttavia, per rendere maggiormente implementabile ed affidabile lo schema di coordinamento proposto, sviluppi futuri dovranno affrontare questo importante problema.

L’approccio che più plausibilmente verrà seguito sarà quello di separare completamente il problema di generazione delle traiettorie di consenso da quello di *collision avoidance*. Infatti, la potenza della teoria del consenso sta proprio nella semplicità delle informazioni scambiate, e complicare il protocollo per implementare la collision-avoidance avrebbe come conseguenza la perdita dei principali vantaggi che questa tecnica di coordinamento offre.

Bibliografia

- [1] P. Wieland, “From Static to Dynamic Couplings in Consensus and Synchronization among Identical and Non-Identical Systems”, PhD thesis, Universität Stuttgart, 2010.
- [2] L. Moreau, “Stability of multi-agent systems with time-dependent communication link”, *IEEE Trans. Automat. Contr.*, 50(2), pp. 169-182, 2005.
- [3] C. Godsil, G. Royle, “Algebraic Graph Theory”, Springer.
- [4] M. Arcak. “Passivity as a design tool for group coordination”, *IEEE Trans. Aut. Contr.*, 52(8), pp. 1380-1390, 2007.
- [5] A. Isidori, L. Marconi, G. Casadei, “Robust output synchronization of a network of heterogeneous nonlinear agents via nonlinear regulation theory”, *IEEE Trans. on Aut. Contr.*
- [6] Dispense del corso di “Sistemi Robotici Distribuiti”, Università di Pisa, docente: Dott.ssa Lucia Pallottino.
- [7] Bolzern Paolo, Scattolini Riccardo, Schiavoni Nicola, “Fondamenti di Controlli Automatici”, McGraw-Hill .

