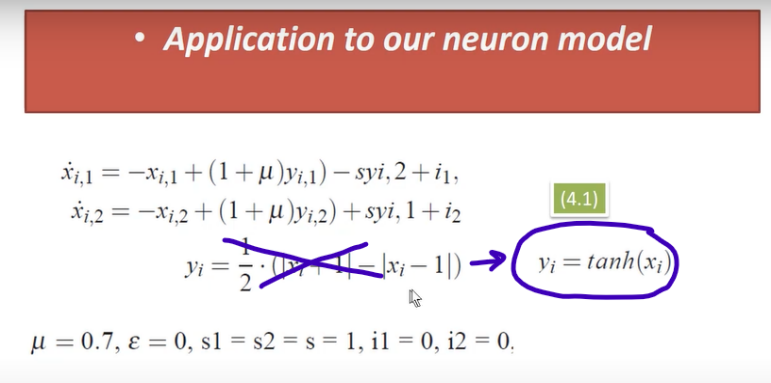
**PARTIAL CONTRACTION IN CPG (Central Pattern Generator)**

Vediamo ora l’applicazione della “partial contraction” in un “tree graph” (grafo che non contiene loop). Qualora ci fossero loop, si dovrebbe obbedire a condizioni periodiche che non garantirebbero la selezione arbitraria delle fasi tra gli oscillatori che controllano il robot.

Consideriamo ora le equazioni del nostro neurone. Qualora i1 e i2 fossero diverse da zero, otterremmo uno slow fast limit cycle. In questo caso consideriamole nulle al fine di creare qualcosa come un oscillatore armonico. Inoltre sostituiamo la classica non linearità utilizzata con la tangente iperbolica (smooth non linear function).

A picture containing text, clock

Description automatically generatedVediamo ora la Partial Contraction Theory implementata in un network CPG formato da 3 neuroni (undirected diffusive graph without loop).

NB: ogni neurone è caratterizzato dalla posizione “i” nel grafo.

Vediamo alcuni parametri associati a questa rete:

* Nn = numero di neuroni nella rete;
* Nc = ordine del singolo neurone (nel nostro caso è pari a 2);
* Nt = Nn\*Nc (numero totale di variabili di stato nella rete, nel nostro caso 6);

Mostriamo l’equazione che rappresenta **l’intera dinamica** della rete:

Diagram

Description automatically generated

La funzione “f” è la dinamica slow fast del singolo sistema di neurone disaccoppiato. La seconda parte è la parte diffusiva, controllata dal gain “k”. Rij è la matrice di rotazione (attorno all’asse “z”, uscente dallo schermo).

Analizziamo la parte diffusiva, in particolare il feedback error. Quando accade che xi è pari a xj (ruotato tramite la matrice R), otteniamo phase shift synchronization.

NB: xj è ruotato di un angolo theta tramite la matrice R. Se la relazione diffusiva è vera (differenza nulla, dopo che xj è stato ruotato), allora xj e xi sono shiftati di theta gradi.

NB2: possiamo imporre questa convergenza a theta gradi qualora il sistema sia “contrattivo”.

NB3: la sincronizzazione phase shifted può applicarsi nei leg dei robot quadrupedi.

NB4: lo shift è presente tra i neuroni, i quali controllano i leg.

Dall’equazione vista prima, otteniamo la seguente (forma compatta):

* Diagram

  Description automatically generatedf(x) è il vettore colonna delle variabili di stato di tutti i neuroni.
* L è la matrice dei coupling 🡪 dimensione NtxNt (pari a 6)
* Lij è il singolo laplacian coupling per il singolo neurone (dimensione NcxNc, pari a 2).

Text

Description automatically generated

Analizziamo le porzioni del laplaciano viste sopra. Il primo caso indica che fisicamente due neuroni sono connessi con un preciso phase shift. La seconda riga è obbligatoria nel nostro caso poiché analizziamo strutture undirected (quindi ci serve la connessione diretta e quella opposta per implementare il concetto di undirected balanced network). Nella terza riga vediamo che se non vi sono connessioni tra neuroni, il laplaciano è nullo. Sulla diagonale del laplaciano, si esplica la classica regola che dice che otteniamo il valore del grado del nodo i-esimo nel grafo.

A picture containing text

Description automatically generatedEsempio di Laplaciano nella struttura vista precedentemente.

Tramite questi concetti vogliamo implementare il gait. Tutto ciò è controllato da una rete di oscillatori, i quali controllano lo sfasamento tra i leg. Ogni tipologia di passo ha un preciso phase shift da implementare in maniera stabile. Il problema è cercare il valore corretto di “k” (fissato L) tale che si possa ottenere un pattern di gait stabile (vogliamo far convergere il sistema ad uno dei passi desiderati: walk, amble, pace, trot, run).

//slide on partial contraction analysis applicata: matematichese eccessivo inutile per ricercare “K”.

//precedente slide riassunta: imporre il phase shift tra le variabili di stato dei rispettivi oscillatori (in una rete di oscillatori). Il valore di Kmin ci fornisce la condizione sufficiente per rispettare la sincronizzazione phase locked. Generalmente, Kmin dipende dal gait imposto.

NB: nella tesi, Kmin è indicato pari a 3 e viene spiegato come un fattore di accoppiamento necessario per ottenere il coupling tra le dinamiche disaccoppiate.

L’obiettivo del nostro controllo del gait è migrare tra diversi flow invariant subspace (i quali rappresentano i gait), mantenendo la condizione sufficiente per la stabilità della partial contraction theory. Tutto ciò viene fatto ipotizzando il grafo fisso per ogni gait (la combinazione dei neuroni non varia). E’ dimostrabile che il valore di K che stabilizza il sistema è fisso per qualunque gait (deve essere superiore a Kmin=2.4 quindi 3 è corretto).

**HEXAPODAL NETWORK**

Diagram

Description automatically generatedVediamo la struttura topologica dei neuroni di un esapode. Definiamo la topologia come un tree graph. R0 indica che non vi è sfasamento nella backbone del robot (B1,B2,B3 sono sincronizzati). Il gait è invece imposto tramite le matrici rotazionali che inducono il phase shift. L1, R2 e L3 sono sincronizzati con zero delay tra di loro (allo stesso modo R1,L2,R3). Lo sfasamento si impone tra i due gruppi di leg (ES: L1,R2,L3 =0 VS R1,L2,R3 = 180 degree phase shift).

NB: il phase shift è imposto rispetto alla backbone (ES: L1 rispetto a B1 = shift nullo, R1 rispetto a B1 = 180 gradi shift).

//sulla parte teorica, qui saltata, le lezioni dell’anno scorso sono più accurate (e più lunghe)

//Arina spiega che K si può considerare come unico per tutti i gait (lo dimostra in maniera farfugliata al solito).

//Utilizzando K=3 si soddisfa la condizione sufficiente per il gait control stabile. Stiamo dunque adoperando un valore unico di K tale che si ottenga la convergenza globale a qualsiasi tipo di flow invariant subspace (ovvero qualunque gait).

La variazione del gait, dunque, NON è ottenuta tramite K (il quale resta fisso) ma tramite il laplaciano. Per come è costruito il laplaciano, esso ha una dipendenza dagli sfasamenti tra i leg. Fissato il laplaciano, si fissa il gait!