

# Sistemi FUZZY

I sistemi fuzzy possono essere definiti come “approssimatori universali”; la teoria fuzzy può essere usata quando il modello di un sistema è sconosciuto o quando non è pratico (o non è economico) adottare un metodo analitico.

Questo può essere il caso di sistemi fortemente non lineari come, ad esempio, un aereo in assetto estremo o il classico problema della palla in equilibrio su un’asta. In queste situazioni, la teoria fuzzy può fornire una **soluzione approssimata sufficientemente buona**, secondo un approccio che **esula completamente dalla teoria del controllo**.

Questo permette al progettista di realizzare buoni sistemi di regolazione senza conoscenza della teoria dei controlli automatici, ma partendo da semplici specifiche “linguistiche” del tipo **IF – THEN**. Per questo motivo, la teoria fuzzy potrebbe essere indicata come **materia prioritaria nei corsi di laurea in ingegneria informatica ed elettronica** di primo e/o di secondo livello, considerando anche il fatto che, in tali ambiti professionali, l’approccio fuzzy presenta una diffusione vasta e stabile da molti anni. Le tecniche fuzzy sono infatti utilizzate in **moltissimi ambiti dell’ingegneria informatica**, a partire dalla **logica di controllo degli elettrodomestici** fino ai **problemi di classificazione e pattern recognition**.

Al fine di meglio illustrare le potenzialità della suddetta teoria, mostreremo di seguito alcuni esempi che riguardano ambiti applicativi piuttosto semplici, ma che possono essere facilmente estesi ad altre aree di interesse.

## **1) controllo di un lucernario per regolazione effetto camino in un edificio**

Ipotizziamo di voler realizzare un sistema per il controllo dell’apertura di un lucernario nella hall di un edificio e che l’apertura debba essere regolata in funzione dell’andamento di alcune grandezze ambientali, ricavate da sensori (inizialmente, consideriamo la presenza di un solo sensore e di un solo lucernario).

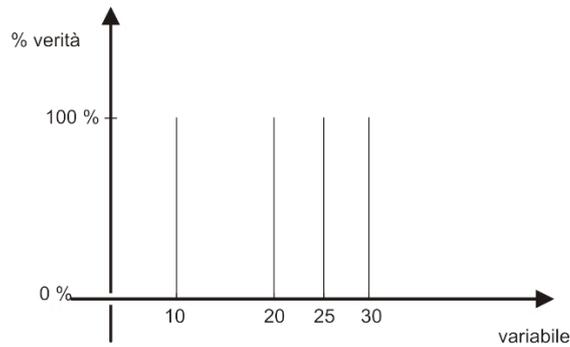
Ipotizziamo che sia necessaria una **regolazione fine** sull’apertura del lucernario e di volere configurare il sistema di controllo mediante **semplici condizioni IF – THEN**, poiché non conosciamo la funzione matematica esatta che lega le due grandezze. Diremo, ad esempio:

**IF temperatura = 10 THEN apertura = 3**

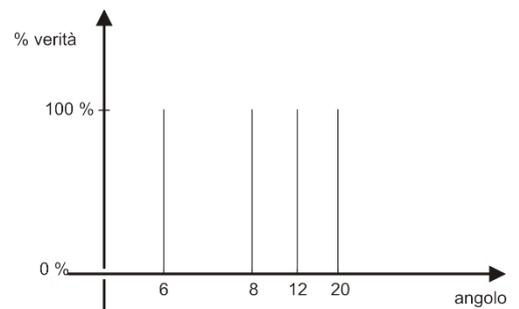
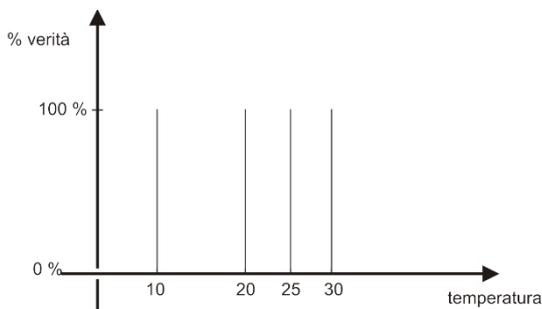
**IF temperatura = 20 THEN apertura = 8**

e così via.

Poiché ogni sistema reale non risolve infiniti valori, ciascuna delle variabili coinvolte avrà **dei valori ammessi in numero finito**, che rappresentiamo con il semplice grafico seguente:



Nella fig. seguente, abbiamo, a sinistra, il grafico della variabile indipendente o controllante (temperatura) e, a destra, il grafico della variabile dipendente (angolo di apertura del lucernario):



```

IF T = 10 THEN angolo = 6°
    |
    |
IF T = 15 THEN angolo = 8°
    |
    |
IF T = 20 THEN angolo = 12°
  
```

La regolazione fine richiederà **un numero molto grande di condizioni IF - THEN**, cioè una per ogni singola coppia (temperatura rilevata, angolo di apertura) da gestire (ricordiamo che non conosciamo la funzione matematica esatta che lega grandezze dipendenti e indipendenti).

La redazione di tali condizioni è complicata dal fatto che la relazione tra grandezza controllante (T) e controllata (angolo) è tipicamente **non lineare**: questo significa che non è possibile scrivere una relazione IF - THEN e poi ricavare tutte le altre semplicemente sommando ogni volta una stessa quantità dx alla controllante e una stessa quantità dy alla controllata (vale a dire che ad uguali variazioni dx della grandezza di ingresso non corrispondono uguali variazioni dy della grandezza di uscita).

Se aggiungiamo **più sensori di temperatura** e, inoltre, **più sensori per altre grandezze ambientali** (ad es, umidità, velocità dell'aria, irraggiamento solare) e **più lucernari o griglie di immissione**, la situazione diventa **ancora più complessa**. Avremo, infatti, regole del tipo:

**IF sensore temp 1 = 10 AND sensore umidità 1 = 30% AND irragg = 40**

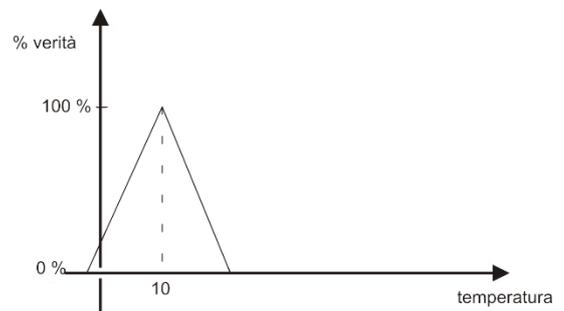
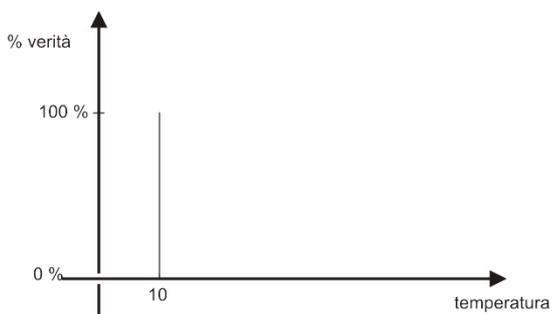
**THEN angolo griglia 1 = 10°**

e così via.

Notare che, in ogni istante temporale, **una sola** delle condizioni IF - THEN è verificata (o, al limite, nessuna se le letture dei sensori cadono fuori dal range previsto).

Da quanto esposto sopra, emerge il fatto che **sarebbe utile poter ottenere una regolazione fine dell'uscita, mantenendo semplice/minimizzando la programmazione delle regole IF (AND) – THEN**. Vedremo di seguito come ciò sia possibile utilizzando un approccio di tipo FUZZY.

Nella tecnica FUZZY, **ogni singola condizione** viene definita **come un intervallo**. Nel caso della condizione  $T = 10$ , avremo quindi (a sinistra caso classico, a destra caso FUZZY):



Nel caso classico, la **condizione  $T = 10$**  prevede il **100 % di verità** per, appunto,  $T = 10$ , mentre **prevede lo 0 % di verità per  $T$  non uguale a 10**;

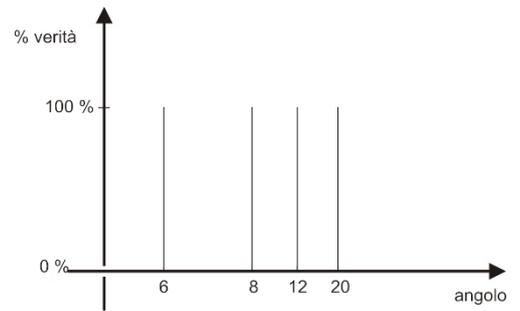
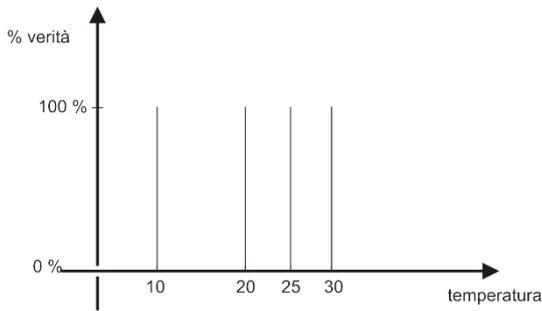
nel caso fuzzy, la condizione  $T = 10$  prevede ancora il 100 % di verità per  $T = 10$ , ma **ammette anche gradi di verità inferiori al 100 %** per valori di  $T$  **nell'intorno di 10**.

Questo significa che, se la lettura del sensore è pari a 5 gradi, avrò una percentuale di verità pari, ad esempio, al 50 % (questo valore dipende ovviamente dalla forma scelta per la funzione intervallo). Utilizzeremo uno stesso tipo di approccio per la variabile di uscita.

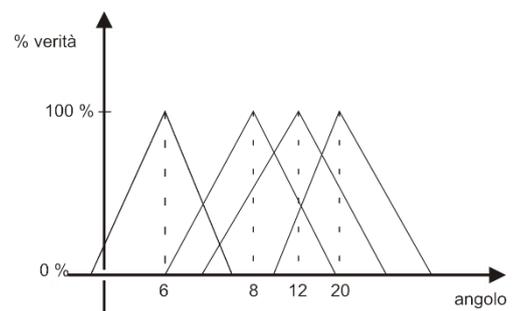
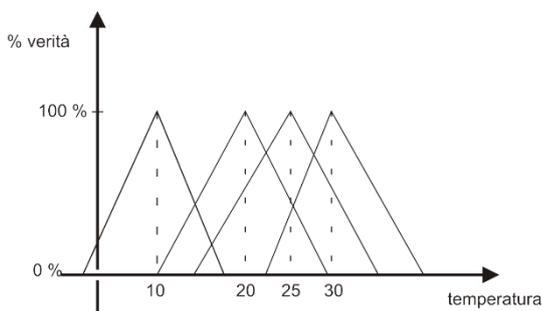
Di per sè, **queste modifiche non permettono di ottenere una regolazione fine dell'uscita**, volendo minimizzare il numero delle regole IF (AND) – THEN; infatti, sarei sempre costretto a scrivere una regola per ogni valore nel dominio di interesse (ad es, sarebbe comunque necessaria una regola aggiuntiva per il caso di  $T = 5^\circ$  prima esposto).

**La situazione cambierebbe se il sistema di controllo fosse in grado di valutare più regole IF (AND) – THEN alla volta** (in ogni istante temporale); questo è quello che accade nei sistemi fuzzy.

Rivediamo la situazione iniziale (classica):

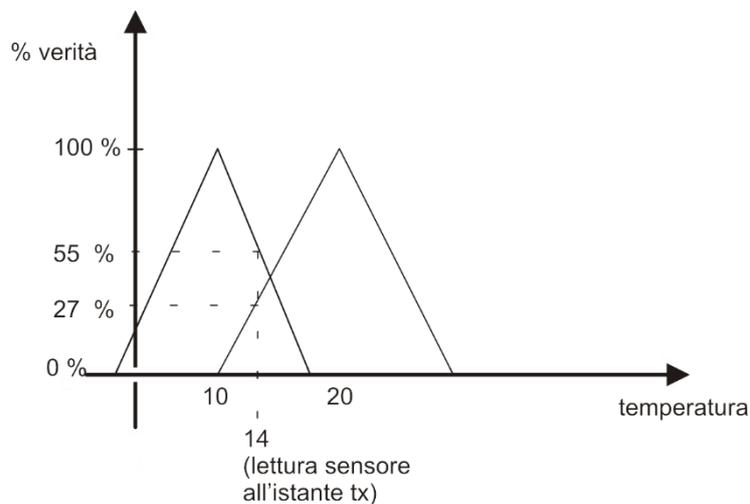


e trasformiamola secondo gli intervalli di tipo fuzzy:



In generale, gli intervalli a triangolo possono non avere la stessa larghezza e persino essere asimmetrici rispetto al valore che li definisce; l'importante è che gli **intervalli contigui presentino un'intersezione non nulla**.

Un'intersezione tra due intervalli sta a significare che la variabile ha la probabilità di assumere due diversi valori contemporaneamente, seppure in diversa percentuale, e che quindi **saranno applicate due regole IF – THEN nello stesso istante temporale**, pesate con **diverse percentuali**.



Nella figura precedente, si nota come una lettura del sensore pari a 14 gradi ricada al 55 % nel caso  $T = 10$  e al 27% nel caso  $T = 20$ . Pertanto, il sistema fuzzy attiverà **contemporaneamente** le due regole (se definite):

**IF  $T = 10$  THEN apertura = angolo 1 (attivazione al 55 %)**

**IF  $T = 20$  THEN apertura = angolo 2 (attivazione al 27 %)**

Tali percentuali determineranno **una diversa percentuale di validità degli intervalli “angolo 1” e “angolo 2”** (anch'essi fuzzy) sulla **variabile d'uscita**. Il valore effettivo dell'angolo di apertura sarà determinato dal sistema tramite un criterio di defuzzificazione (tra i più adottati, quello del “centro di massa” o centroide).

Appare chiaro che, **con le due sole regole precedenti in un sistema fuzzy** determinato dagli intervalli come sopra definiti, **abbiamo ottenuto una regolazione fine nel range da 10 a 19 gradi** (intervallo di intersezione) che in un sistema classico avrebbe richiesto moltissime regole; infatti, **al variare della lettura dal sensore in tale intervallo, variano con continuità le percentuali di attivazione delle regole precedenti e di conseguenza varia con continuità il valore di uscita tra i valori angolo 1 e angolo 2.**

**Con l'adozione di poche regole (tipicamente meno di 10), è possibile definire comportamenti estremamente accurati.**

Lo stesso principio può essere applicato al caso di **regole scritte per sensori e attuatori multipli.**

Va notato che il sistema fuzzy, una volta definito, **esegue autonomamente i calcoli necessari**; l'utente deve solo definire gli intervalli per le variabili e le regole.

Modificando la posizione relativa dei singoli intervalli, è possibile effettuare un fine-tuning del sistema, in ogni caso, un sistema fuzzy costituisce, ovviamente, una **soluzione approssimata** del caso classico di regolazione mediante utilizzo della funzione di trasferimento del sistema fisico in oggetto.

## ***2) ball and beam problem (2 dimensioni)***

Riportiamo un caso applicativo del **ball and beam problem** per il quale abbiamo realizzato un prototipo secondo un approccio a due dimensioni: una palla deve essere mantenuta al centro di una piattaforma, nonostante la presenza di forze disturbanti esterne.

La piattaforma è fissata su un giunto cardanico e può essere mossa in due dimensioni mediante due servoattuatori, controllati dal software che implementa l'algoritmo fuzzy.

La **posizione** della palla è letta tramite una **griglia ad infrarossi** parallela alla piattaforma; la griglia è realizzata mediante diodi emettitori IR e fototransistor recettori IR, pilotati ad impulsi sincronizzati con le letture effettuate dal software, il quale reagisce anche alla **derivata prima della posizione (velocità)** della palla.

L'algoritmo potrebbe essere realizzato secondo i dettami della **teoria della regolazione (utilizzando un regolatore PID)** ma vogliamo mostrare che è anche possibile risolvere il problema in maniera

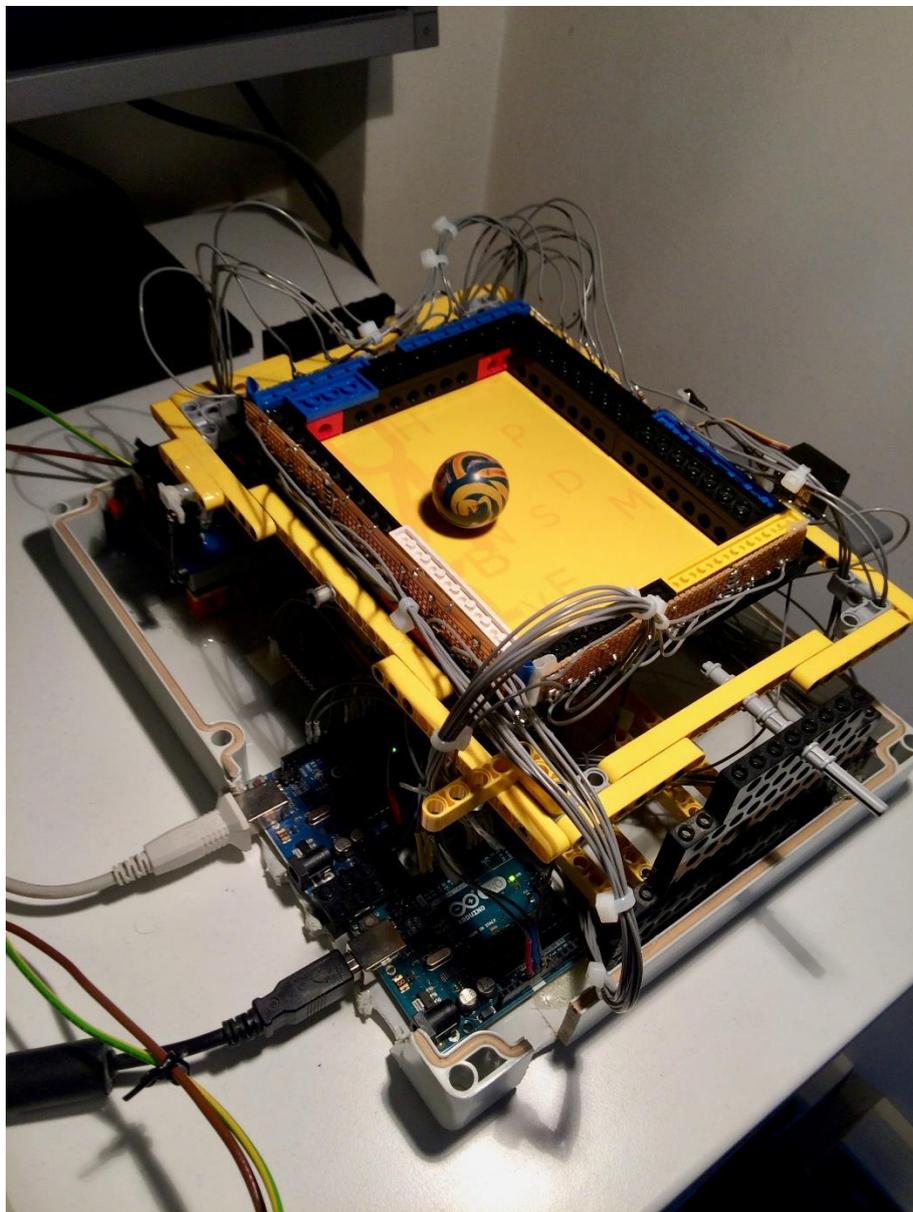
sufficientemente buona mediante un approccio fuzzy, utilizzando solo specifiche verbali, senza nessuna conoscenza della teoria della regolazione.

Il sistema è programmato secondo regole del tipo:

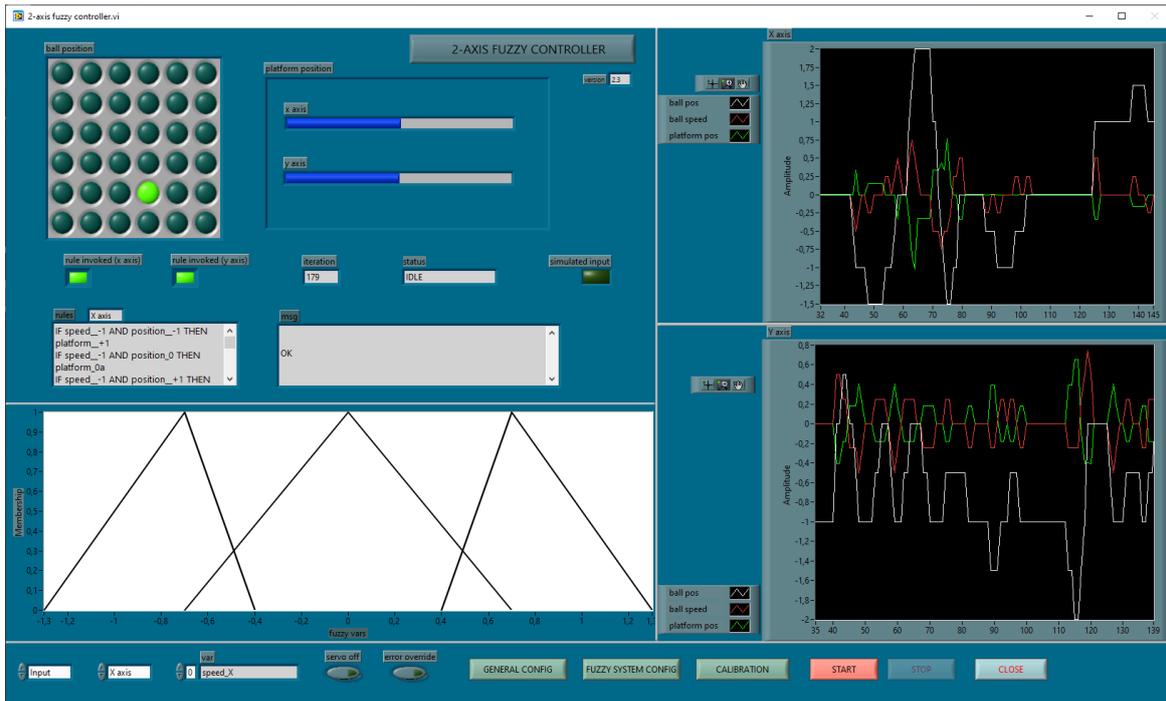
**IF velocità palla = 2 AND posizione palla = -3 THEN angolo piattaforma = 10°**

usando coppie di regole equivalenti per le componenti dei vettori su ciascun asse (è stato trascurato, per semplicità, il rapporto tra l'angolo della piattaforma e la coppia applicata, approssimazione tollerabile per determinati valori di massa della palla).

La dinamica dell'utilizzo delle regole da parte del sistema fuzzy è la stessa illustrata nell'esempio precedente.



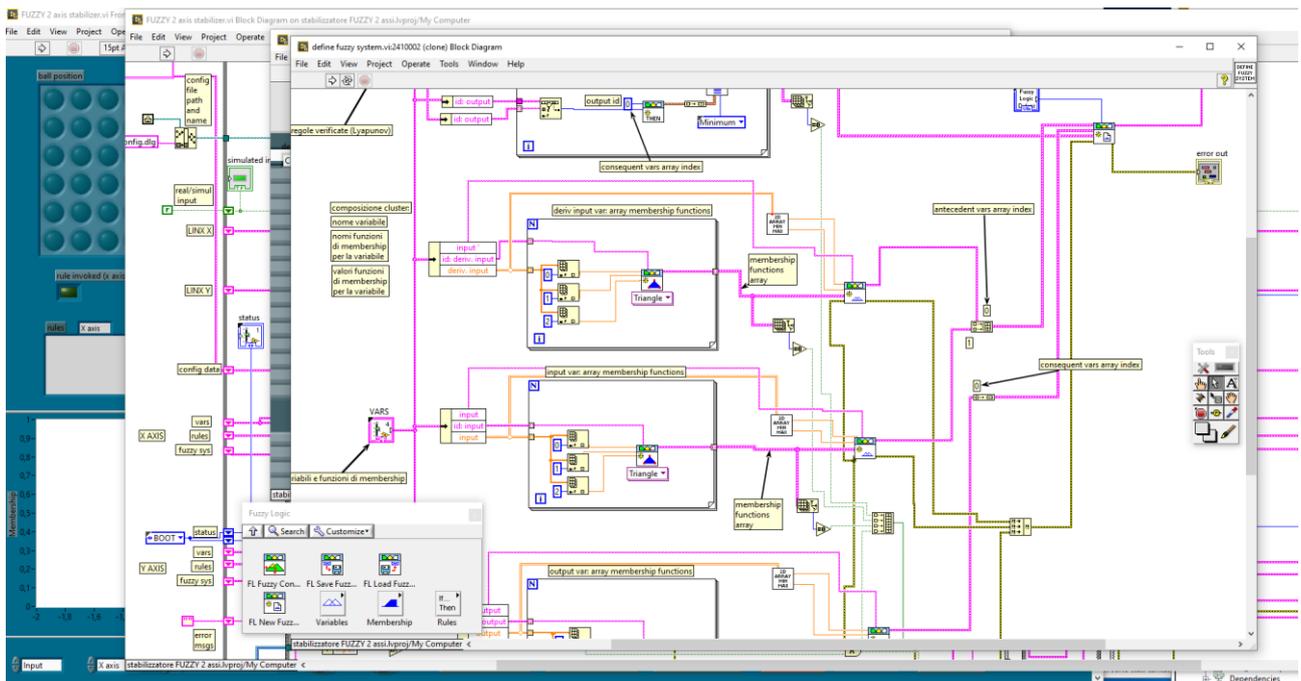
allestimento sperimentale per il controllore fuzzy per il ball and beam problem



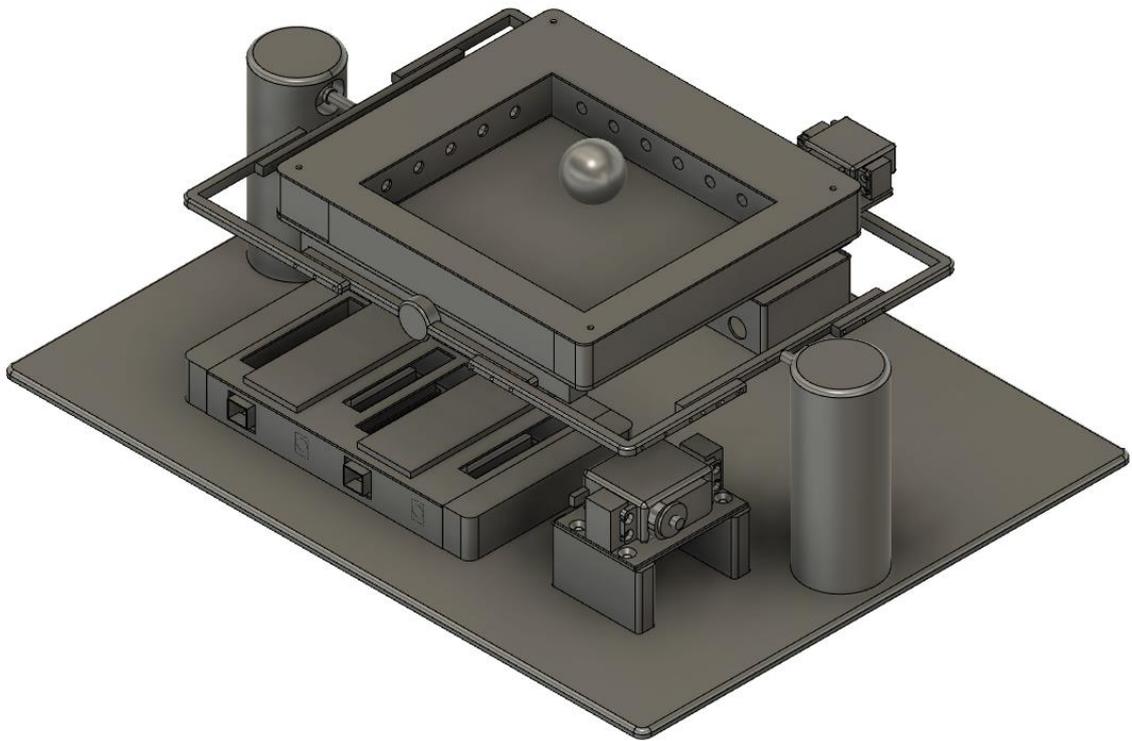
interfaccia software per il controllore fuzzy (ball and beam problem)

altre immagini e video disponibili all'indirizzo : [https://www.studiorossi2.it/fuzzy\\_controller.php](https://www.studiorossi2.it/fuzzy_controller.php)

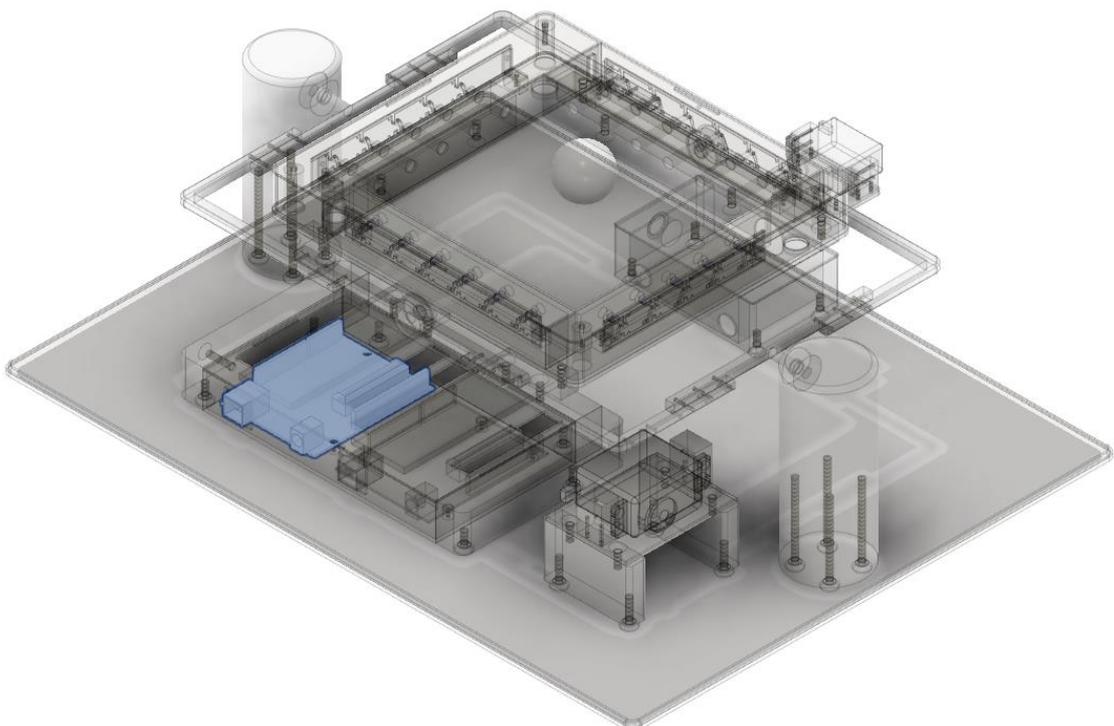
Il software è stato realizzato mediante l'ambiente Labview della National Instruments, che contiene un toolbox per lo sviluppo dei sistemi fuzzy (l'autore è sviluppatore certificato CLD) che è **disponibile gratuitamente** per scopi non commerciali all'indirizzo <https://www.ni.com/it-it/shop/labview/select-edition/labview-community-edition.html> .



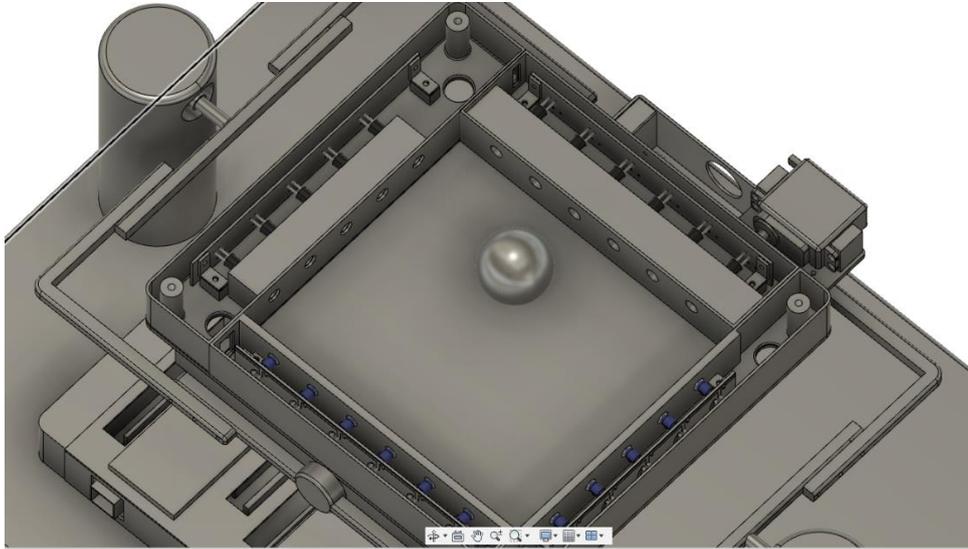
frammento di codice del regolatore fuzzy



Progetto al CAD della versione in metallo su cuscinetti a sfera



Progetto al CAD della versione in metallo (trasparenza)



Progetto al CAD della versione in metallo (dettaglio)

### 3) software per ottimizzazione con fattori di preferenza

L'esempio qui illustrato riguarda un software che permette di scegliere la migliore di tre soluzioni alternative (A, B e C) sulla base di parametri di costo, prestazioni e flessibilità, oltre che in base alle preferenze che un ipotetico committente assegna ai suddetti parametri.

Il software, realizzato in PHP, implementa l'algoritmo min-max nella variante che tiene conto dei fattori di preferenza sui parametri e può essere testato all'indirizzo:

[https://www.studiorossi2.it/fuzzy\\_decision\\_making\\_software.php](https://www.studiorossi2.it/fuzzy_decision_making_software.php)

studio ROSSI - fuzzy multioject: X

← → ↻ 🏠 🔍 ☆ 🗑️ ⋮

https://www.studiorossi2.it/fuzzy\_decision\_making\_software.php

FUZZY OPTIMIZATION WITH PREFERENCE FACTORS

You have to choose between three alternatives: A, B and C (they can be of any kind).

Three objectives impact the decision: cost, performance, flexibility to future changes of the requirements:

so, you must define quality factors for each objective (in the scale from 0.1 (bad) to 0.9 (optimum)).

Moreover, your client has defined preferences for each objective (in the scale from 0.1 (small preference) to 0.9 (full preference)).

Having these preconditions, the algorithm must calculate the best solution among the three alternatives.

Each parameter has a certain degree of validity, so this is a typical fuzzy problem.

The three alternatives may be of any kind: you define them by quality factor values only.

A cost: 0.2 | A performance: 0.8 | A flexibility: 0.4 | B cost: 0.8 | B performance: 0.6 | B flexibility: 0.6 | C cost: 0.9 | C performance: 0.2 | C flexibility: 0.5 | client preference: cost: 0.5 | client preference: performance: 0.8

client preference: flexibility: 0.7

CALCULATE

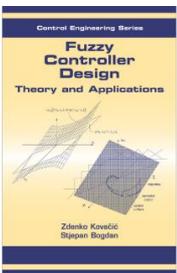
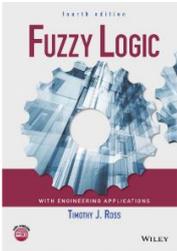
RESULT

**B choice**

result coeff.  
A: 0.4 --- B: 0.6 --- C: 0.2

## Testi di approfondimento

Di seguito riportiamo due testi di approfondimento della materia:



## Conclusioni

La teoria fuzzy permette al progettista di realizzare buoni sistemi di regolazione in moltissimi ambiti applicativi (dalla logica di controllo degli elettrodomestici fino ai problemi di classificazione e pattern recognition) evitando un approccio analitico e partendo da semplici specifiche “linguistiche” del tipo IF – THEN. I sistemi fuzzy hanno raggiunto una diffusione vasta e stabile da molti anni e sono dunque **ben rivendibili** nel mercato del lavoro che compete agli ingegneri del software.

ing. Stefano Rossi

ingegnere informatico iunior

(iscriz. albo ingegneri prov. Roma n° B 39534)

Certified Labview Developer



Web: [www.studiorossi2.it](http://www.studiorossi2.it)

Social: [profilo LinkedIn](#)