AGENTI CHE RAGIONANO LOGICAMENTE

Logica fuzzy nella navigazione di robot autonomi

E.Mumolo

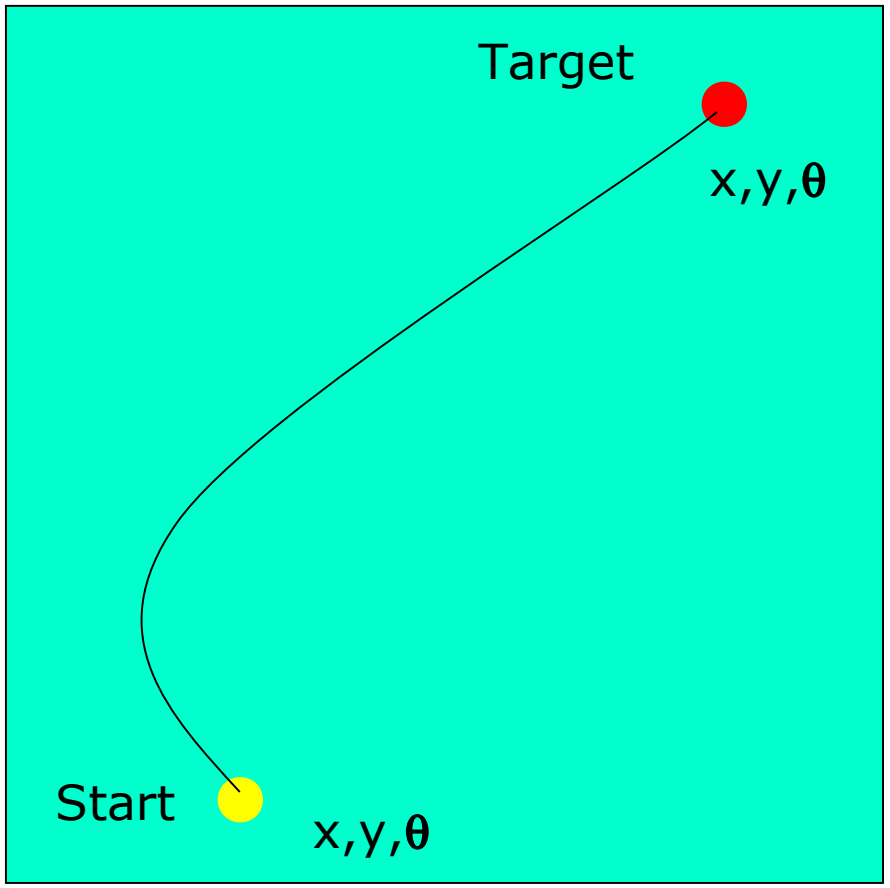
mumolo@units.it

Navigazione di robot autonomi

- Comportamenti reattivi
- Inseguimento della traiettoria (trajectory tracking)
  - Percorso assegnato → comandi attuatori (*Speed e Jog*)
  - Euristica: raggiungimento di punti intermedi
- Esempi:
  - Manovra di parcheggio
  - Allineamento all'obiettivo
  - Navigazione punto-punto
  - Evitare collisioni

Navigazione di un singolo robot

- Moto Piano
- Robot “ideale”
  - puntiforme
  - variazioni di direzione istantane
  - angolo di sterzata limitato
  - velocità costante in modulo
- Robot “reale”
  - corpo rigido
  - vincoli anolonomi
  - angolo di sterzata limitato
  - velocità variabile

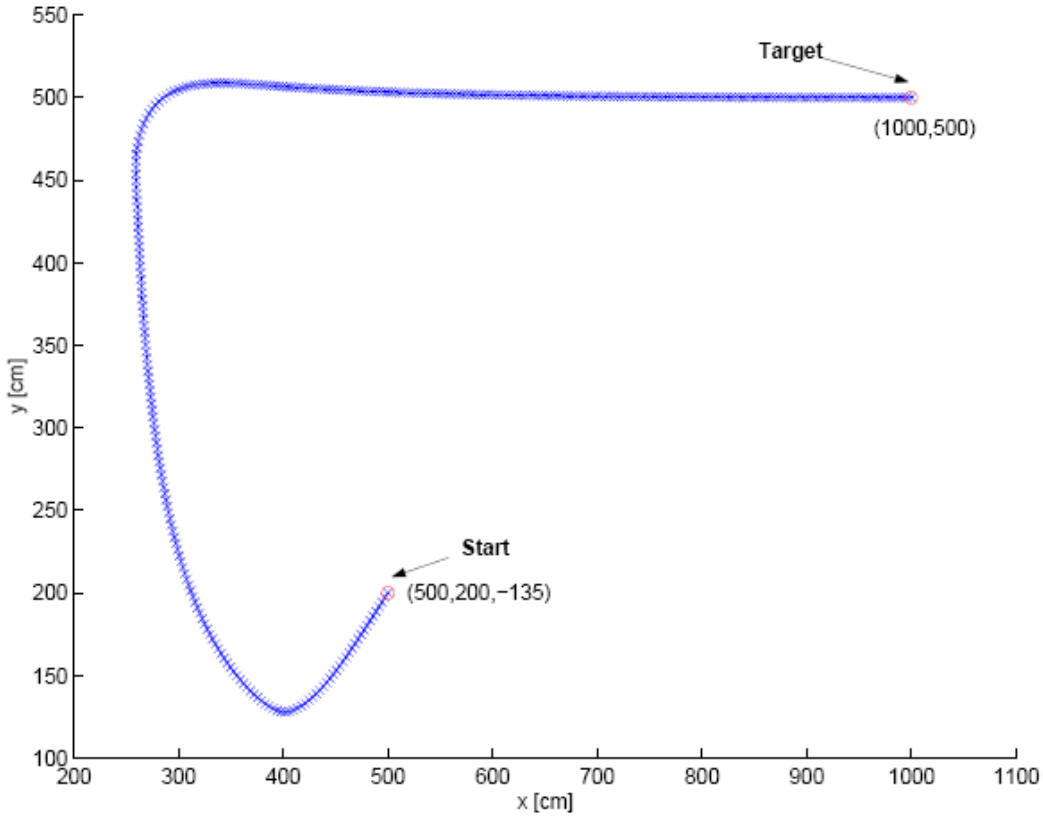


Manovra di parcheggio

- Problema classico
  - Nguyen e B. Widrow, The truck backer-upper: An example of self learning in neural networks, in International Joint Conference on Neural Network, vol. 2, IEEE Press, 1989, pp. 357–363.
  - S.-G. Kong e B. Kosko, Adaptive fuzzy systems for backing up a truckand-trailer, IEEE Transactions on Neural Networks, 3 (1992), pp. 211–223.
- Ingressi: coordinate (x,y) e orientamento θ del veicolo
- Uscita: angolo ϕ dello sterzo
- Inisiemi fuzzy:
- $x \in [0 \div 10]$ metri → Sinistra, Mediamente Sinistra, Centrale, Mediamente Destra e Destra
- $x \in [0 \div 10]$ metri → Bassa, Mediamente Bassa, Centrale, Mediamente Alta e Alta
- $\theta \in [-180 \div 180]$ gradi → Sinistra, Basso, Destra e Alto
- $\phi \in [-30 \div 30]$ gradi →Sinistra, Mediamente Sinistra, Dritto, Mediamente Destra e Destra

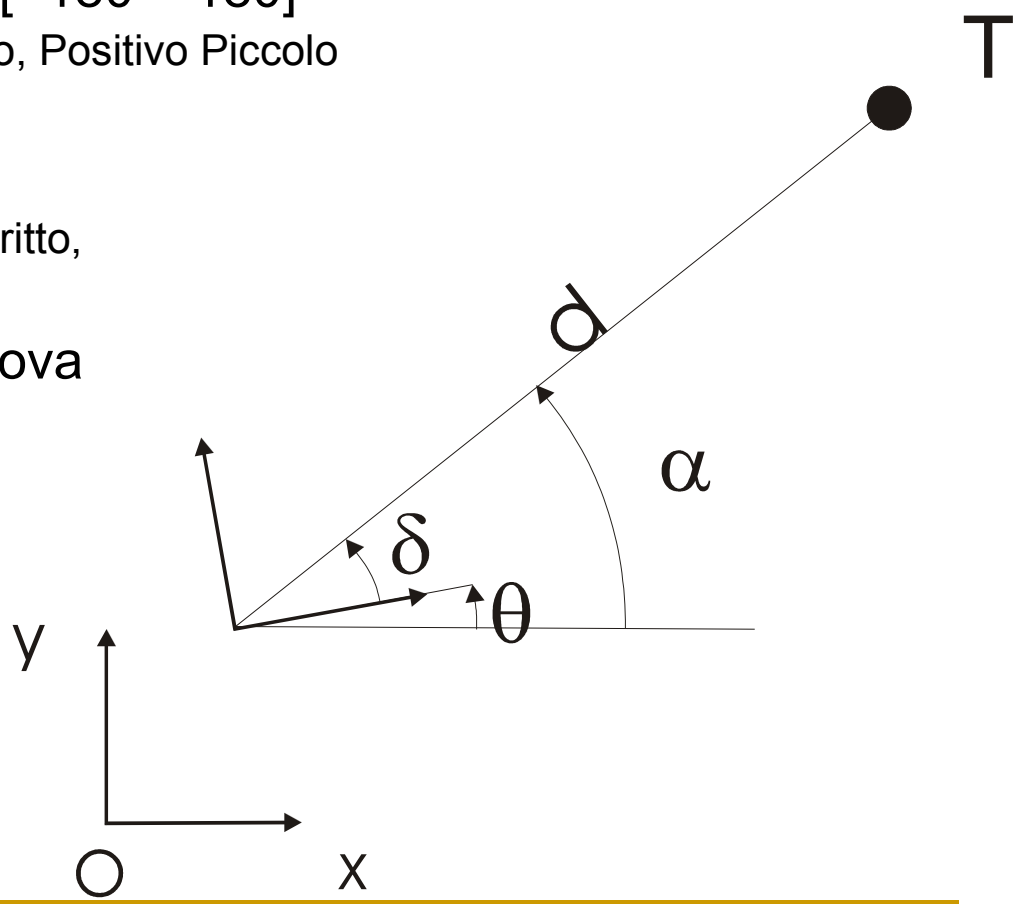
Manovra di parcheggio

- 100 regole IF … THEN … ELSE
- Robot puntiforme, sterzata max 30° e velocità costante 0.2 m/s
- Risultati:



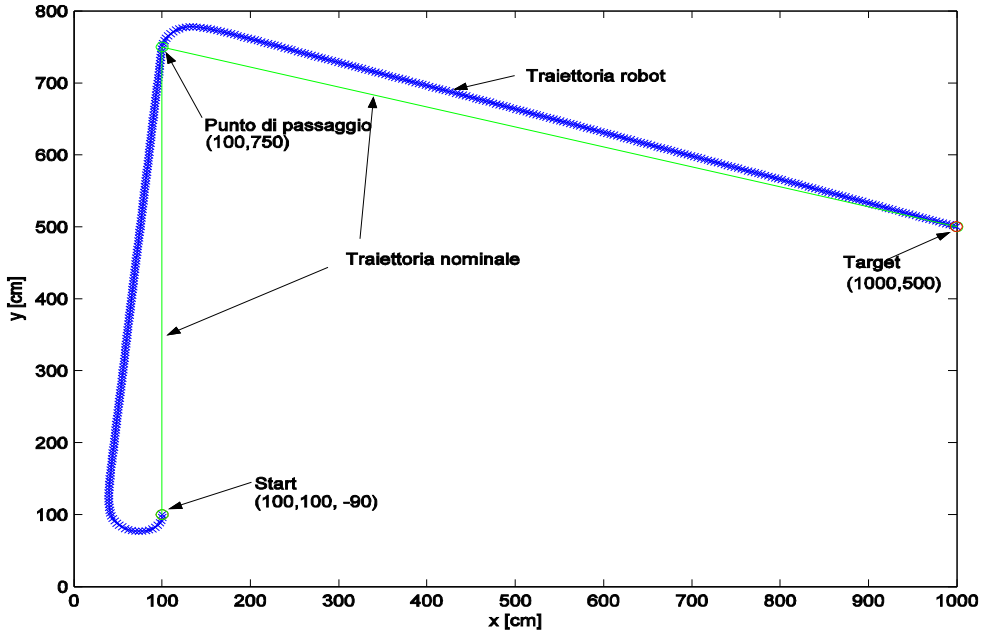
Allineamento all'obiettivo

- Robot ideale
- Ingresso : disallineamento δ [−180 ÷ 180]
  - Negativo Grande, Negativo, Zero, Positivo Piccolo e Positivo Grande
- Uscita: sterzo θ [−30 ÷ 30]
  - Sinistra, Mediamente Sinistra, Dritto, Mediamente Destra e Destra
- Se δ negativo, l'obiettivo si trova alla destra del robot
  → sterzare a destra
- Se δ positivo
  →sterzare verso sinistra
- Se δ nullo
  →allineato



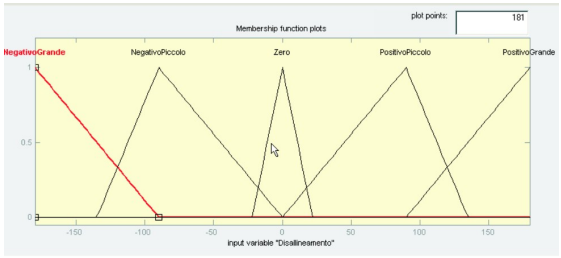
Allineamento all'obiettivo

1	(**IF** δ is *Negativo Grande*)	**THEN**	ϕ is *Destra*
2	(**IF** δ is *Negativo Piccolo*)	**THEN**	ϕ is *Mediamente Destra*
3	(**IF** δ is *Zero*)	**THEN**	ϕ is *Dritto*
4	(**IF** δ is *Positivo Piccolo*)	**THEN**	ϕ is *Mediamente Sinistra*
5	(**IF** δ is *Positivo Grande*)	**THEN**	ϕ is *Sinistra*

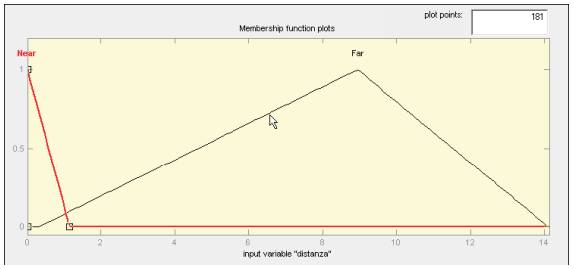


Navigazione punto a punto

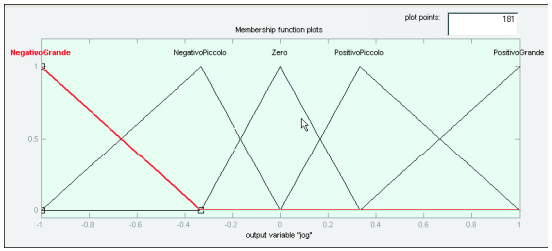
- Ingressi:
  - “*disallineamento*”:



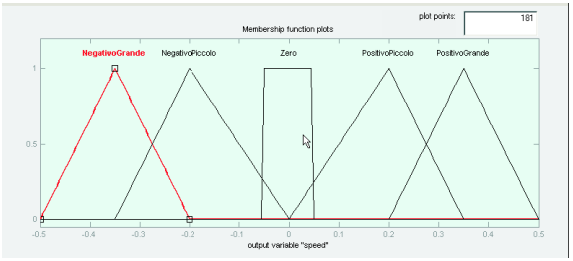
  - “*distanza obiettivo*”:



- Uscite:
  - *Jog*



  - *speed*

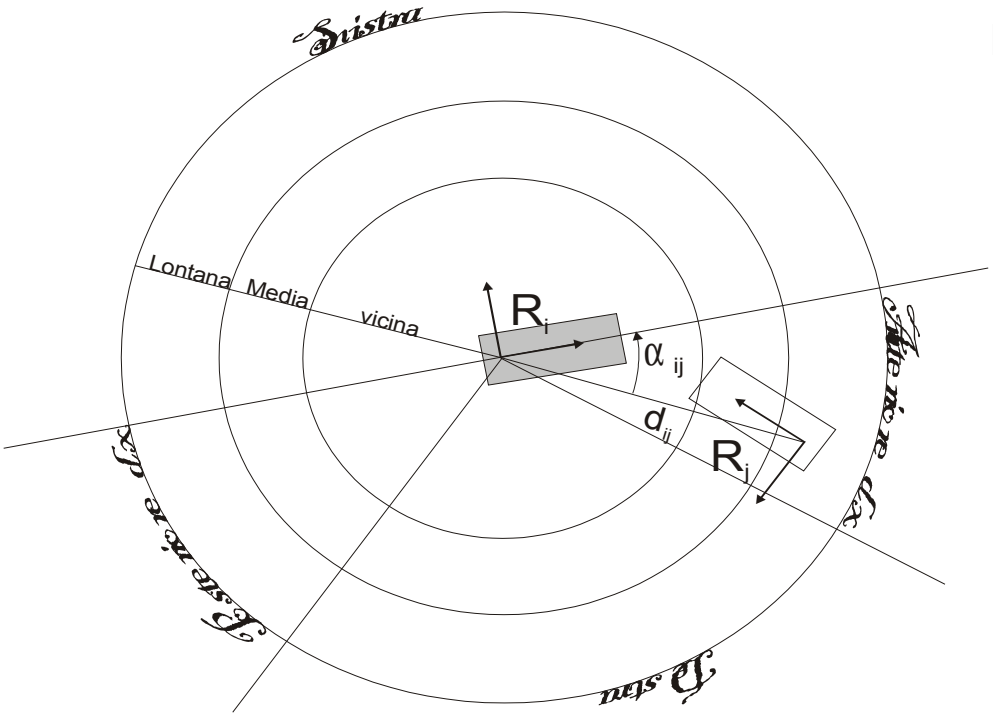


Navigazione punto a punto

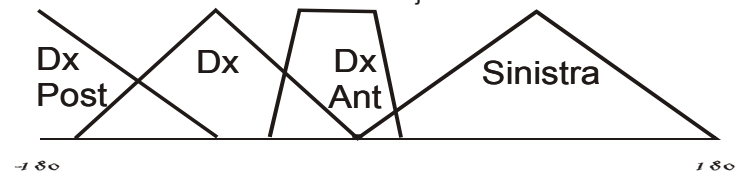
- 10 Regole linguistiche: segue l'allineamento all'obiettivo

1	(**IF** δ is NG)	**AND**	(d is V)	**THEN**	jog is NG	$speed$ is Z
2	(**IF** δ is NP)	**AND**	(d is V)	**THEN**	jog is NP	$speed$ is Z
3	(**IF** δ is Z)	**AND**	(d is V)	**THEN**	jog is Z	$speed$ is Z
4	(**IF** δ is PP)	**AND**	(d is V)	**THEN**	jog is PP	$speed$ is Z
5	(**IF** δ is PG)	**AND**	(d is V)	**THEN**	jog is PG	$speed$ is Z
6	(**IF** δ is NG)	**AND**	(d is L)	**THEN**	jog is NG	$speed$ is PP
7	(**IF** δ is NP)	**AND**	(d is L)	**THEN**	jog is NP	$speed$ is PP
8	(**IF** δ is Z)	**AND**	(d is L)	**THEN**	jog is Z	$speed$ is PG
9	(**IF** δ is PP)	**AND**	(d is L)	**THEN**	jog is PP	$speed$ is PP
10	(**IF** δ is PG)	**AND**	(d is L)	**THEN**	jog is PG	$speed$ is PP

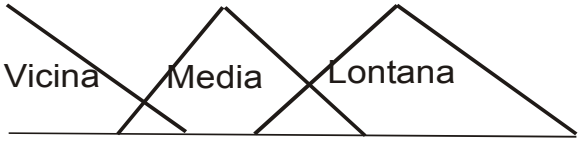
Evitare le collisioni



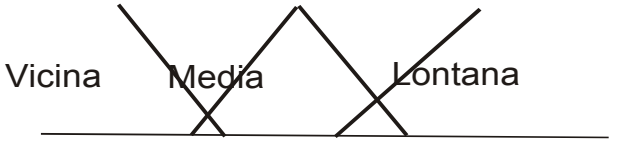
Direzione relativa α_{ij}



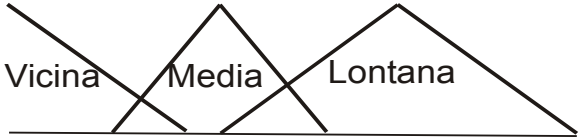
Distanza d_{ij}



Δ jog



Δ speed



Evitare le collisioni tra robot

- Ingressi:
  - d_{ij} (distanza tra R_i e R_j) divisa in Vicina, Media e Lontana.
  - α_{ij} (angolo tra la direzione R_i e R_j) divisa in Posteriore Destra, Destra, Anteriore Destra e Sinistra.
- Uscite: Δjog e Δspeed rispetto a jog e speed della navigazione punto a punto
- Attivato quando la distanza < soglia
- Regole fuzzy: diritto di precedenza a destra
  - Se R_j a sinistra di R_i ➔ $\Delta = 0$
  - Se a destra, R_i modifica la propria traiettoria in maniera più o meno brusca a seconda della distanza d_{ij}

Evitare le collisioni tra robot

- 12 regole

1	(**IF** d_{ij} is *Vicina*)	**AND**	(α_{ij} is *Destra*)	**THEN**	Δjog is Z, Δ *speed* is N
2	(**IF** d_{ij} is *Media*)	**AND**	(α_{ij} is *Destra*)	**THEN**	Δjog is N, Δ *speed* is N
3	(**IF** d_{ij} is *Lontana*)	**AND**	(α_{ij} is *Destra*)	**THEN**	Δjog is Z, $\Delta speed$ is Z
4	(**IF** d_{ij} is *Vicina*)	**AND**	(α_{ij} is *Sinistra*)	**THEN**	Δjog is Z, Δ *speed* is Z
5	(**IF** d_{ij} is *Media*)	**AND**	(α_{ij} is *Sinistra*)	**THEN**	Δ *jog* is Z, $\Delta speed$ is Z
6	(**IF** d_{ij} is *Lontana*)	**AND**	(α_{ij} is *Sinistra*)	**THEN**	Δ *jog* is Z, $\Delta speed$ is Z
7	(**IF** d_{ij} is *Vicina*)	**AND**	(α_{ij} is *Dx Ante*)	**THEN**	Δ *jog* is Z, $\Delta speed$ is N
8	(**IF** d_{ij} is *Media*)	**AND**	(α_{ij} is *Dx Ante*)	**THEN**	Δ *jog* is N, *speed* is N
9	(**IF** d_{ij} is *Lontana*)	**AND**	(α_{ij} is *Dx Ante*)	**THEN**	Δjog is Z, $\Delta speed$ is Z
10	(**IF** d_{ij} is *Vicina*)	**AND**	(α_{ij} is *Dx Post*)	**THEN**	Δjog is Z, $\Delta speed$ is P
11	(**IF** d_{ij} is *Media*)	**AND**	(α_{ij} is *Dx Post*)	**THEN**	Δjog is N, $\Delta speed$ is P
12	(**IF** d_{ij} is *Lontana*)	**AND**	(α_{ij} is *Dx Post*)	**THEN**	Δjog is Z $\Delta speed$ is Z