RoboQuaderno

Schema generale di un sistema di controllo
Parte I : Proporzionale

Gabriele e Rodolfo Zunino
giugno 2015

Sommario

Il quaderno illustra la struttura generale di un sistema di controllo, molto popolare nelle applicazioni robotiche per usare i sensori in modo da ottenere un comportamento stabile. La scheda mostra lo schema generale di un sistema di controllo, nella versione proporzionale che è la più semplice.
Applicazioni: Inseguitore di linee (*Line follower*), guida su rotta costante, guida a distanza costante, inseguitore di target IR, altri.

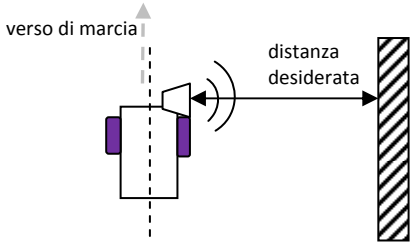
Descrizione generale

Il principio generale è di fare in modo che il robot mantenga un qualche comportamento desiderato (es, inseguire una linea al suolo, mantenere una direzione costante, etc) anche in presenza di errori, disturbi, etc. Per ottenere questo dovremo innanzitutto:

1. definire il **comportamento** desiderato;
2. scegliere il **sensore** che vogliamo usare;
3. impostare un **riferimento**, ovvero il *valore ideale* che vorremo leggere costantemente sul nostro sensore;
4. decidere la **Potenza nominale** da applicare ai motori, cioè la potenza che vorremmo dare ai motori del nostro robot nel caso ideale.

Un caso pratico

1) Vogliamo costruire un robot che avanzi mantenendo una distanza costante da una parete laterale.
2) Per questo scegliamo di usare un sensore ultrasonico (misuratore di distanza) e lo puntiamo verso la parete sul lato del robot
3) Impostiamo il riferimento (ovvero decidiamo quanto vale la distanza desiderata rispetto alla parete)
4) Fissiamo una potenza motori ideale, nel nostro caso Potenza = 20



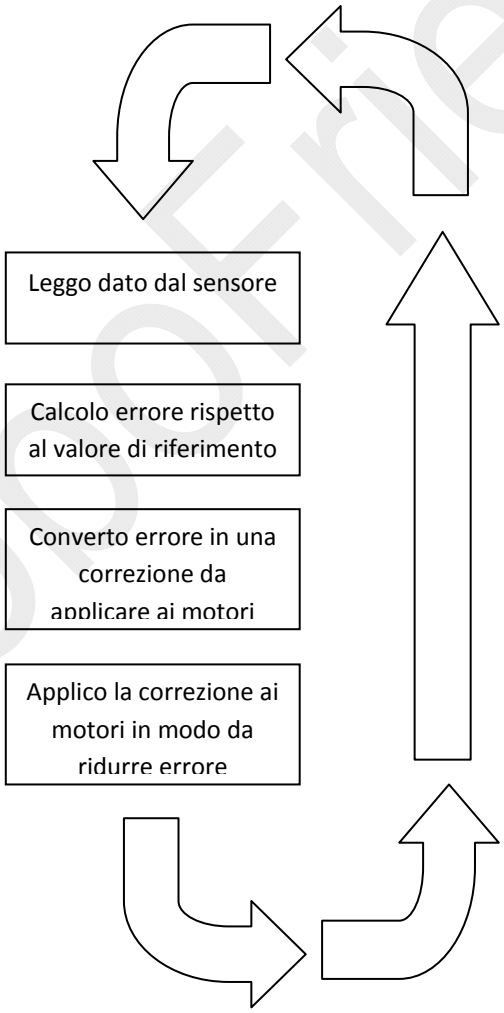
Ora possiamo iniziare a costruire il nostro sistema di controllo. L'idea di massima consiste nel leggere il valore del nostro sensore e confrontarlo con il riferimento; se sono uguali, siamo a posto! Ma siccome nella pratica non lo sono, misuriamo la discrepanza ("errore") fra valore reale (sensore) e ideale (riferimento). Dopodiché applichiamo al robot una correzione che lo farà svoltare nella direzione opportuna affinché l'errore diminuisca.

Ricordiamo che un modo tipico per fare svoltare un robot è aumentare la velocità su una ruota e contemporaneamente diminuire la velocità sull'altra ruota.

Più grande è l'errore misurato, maggiore sarà la correzione apportata: questo è il principio fondamentale di un controllore **Proporzionale (P)**. In nome tecnico di questo principio è "*Controllo con retroazione*" e si applica in tantissimi campi dell'automazione, anche nei robot più elementari.

Schema generale di un sistema di controllo

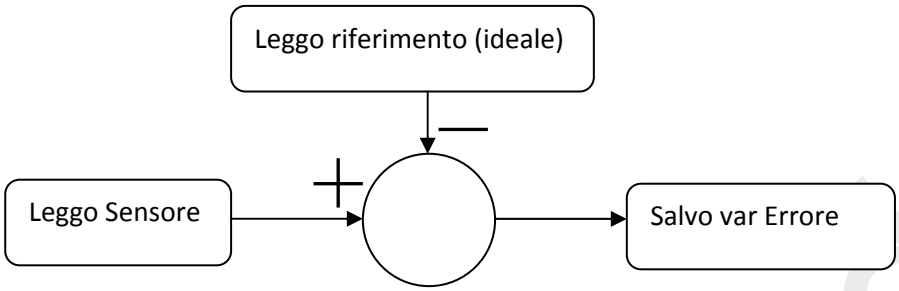
Ecco lo schema di massima (ha la forma ciclica di una sequenza che si ripete continuamente), che vale per ogni controllo di questo tipo:



Non ci resta che mettere in pratica (implementare) le poche fasi di questo ciclo di controllo.

1. Calcoliamo l'errore

La prima cosa da fare è leggere il sensore e confrontarlo con il riferimento, con una semplice sottrazione, poi salviamo il risultato in una variabile "Errore":

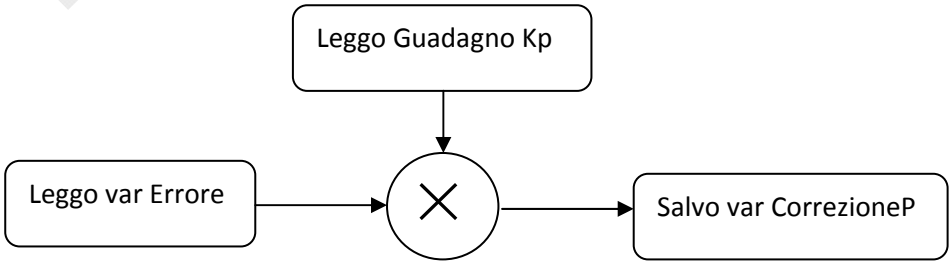


In pratica

1. Leggo valore di distanza dal sensore UltraSonico (US)
2. Leggo variabile con distanza ideale (riferimento)
3. Calcolo differenza: (sensoreUS) - (rif)
4. Salvo differenza nella var. Errore = (sensoreUS) - (rif)

2. Convertiamo l'errore in correzione per i motori

Ora dobbiamo convertire l'errore ottenuto in una qualche correzione ai motori del robot. Siccome il valore di Errore è misurato in gradi, non possiamo applicare questo numero alla potenza dei motori direttamente. Pertanto useremo una costante di conversione che *moltiplica* l'errore e lo traduce in correzione di potenza motore. Questa costante (molto importante nel controllore P) ha il nome tecnico di **guadagno proporzionale** e si indica di solito con il simbolo Kp. Pertanto continuiamo:



In pratica

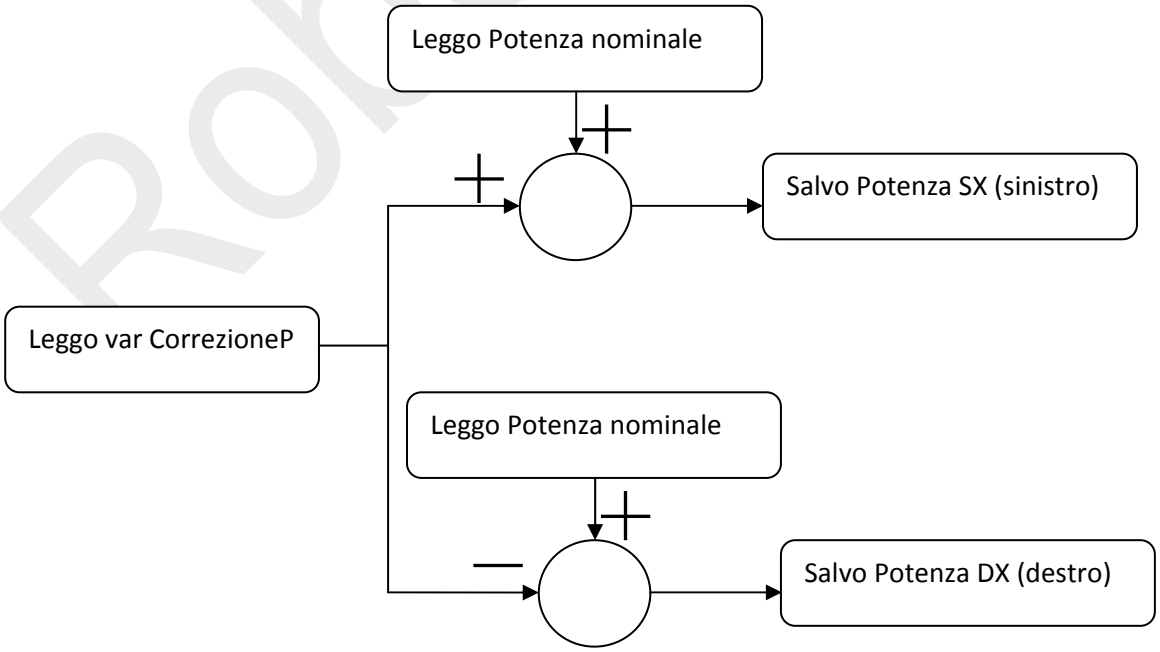
5. Leggo variabile Errore
6. Leggo variabile Kp
7. Moltiplico Kp * Errore
8. Salvo risultato in var CorrezioneP = Kp * Errore

3. Applichiamo la correzione

A questo punto non ci resta che applicare la correzione ai motori, bisogna solo capire con quale segno. Per decidere questo aspetto dobbiamo conoscere il criterio di misura del nostro sensore; per esempio, nel caso di un giroscopio, dobbiamo sapere se misura angoli positivi per rotazioni orarie o antiorarie; nel caso di un sensore ottico, se rileva valori maggiori per luminosità più chiare o scure; la cosa è più intuitiva nel caso di un sensore di distanza, visto che ci darà sempre valori positivi. Questo si deve rilevare caso per caso, provando in pratica (ad esempio mediante la funzione "Port View" del nostro robot).

Per fare in modo che il robot "viri" per mantenere il riferimento costante, dovremo sommare e sottrarre la correzione di potenza appena calcolata al motore sinistro/destro, rispettivamente. A quale motore sommare e a quale motore sottrarre dipende dal caso specifico e da come abbiamo costruito il nostro robot. Nel nostro caso assumiamo per semplicità che sommiamo sul motore SX e sottraiamo sul motore DX.

Siccome non esiste una regola generale, dovrete impostare le cose volta per volta; l'esempio pratico può aiutare a chiarire questo passo non difficile, e comunque molto di questo di impara ... provando!



In pratica

Nel nostro caso, supponiamo di avere la parete di riferimento alla destra del robot (come nella figura a pag.1) . Ci sono due casi possibili:

*Quando il robot si allontana dalla parete (deriva verso sinistra) avremo che la "distanza" è più grande del "riferimento" --> [errore > 0] e [correzione > 0]. Siccome vogliamo dirigere il robot verso destra, **sommiamo** la correzione sul motore **SX**, e **sottraiamo** la correzione sul motore **DX**, in modo che il robot ruoti verso la parete.*

*Quando il robot si avvicina alla parete (deriva verso destra) avremo che la "distanza" è più piccola del "riferimento" --> [errore < 0] e [correzione < 0]. Quindi dovremo far tornare il robot verso sinistra. La cosa bella di questo schema è che **anche in questo caso sommiamo** la correzione sul motore **SX**, e **sottraiamo** la correzione sul motore **DX**! Infatti sommiamo e sottraiamo correzioni negative, quindi abbiamo l'effetto contrario al caso precedente.*

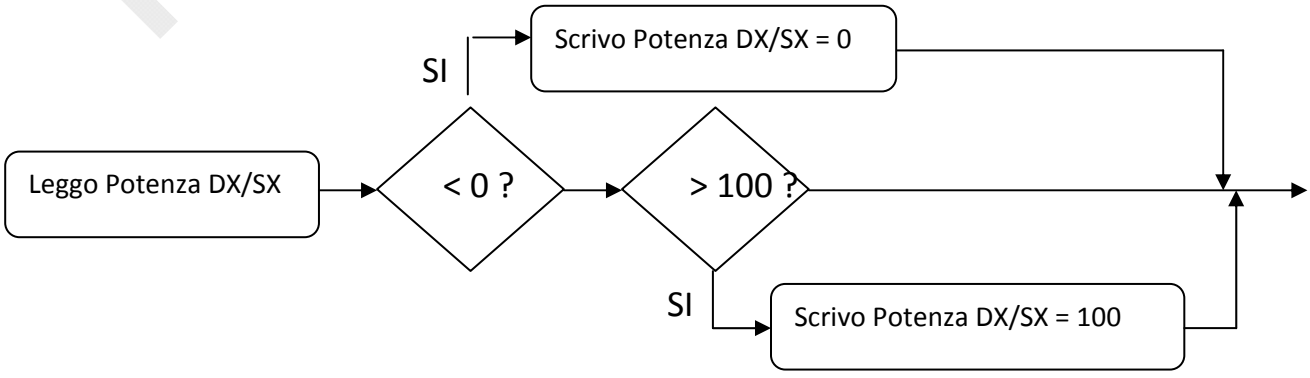
Lo schema somma/sottrai funziona sia per correzioni positive sia per correzioni negative:

9. Leggo variabile Correzione
10. Leggo variabile PotenzaNominale
11. Sommo PotenzaNominale + CorrezioneP
12. Salvo risultato in var Potenza SX
13. Sottraggo PotenzaNominale - CorrezioneP
14. Salvo risultato in var Potenza DX

4. Ultimo sforzo: rimettiamo a posto le potenze

In teoria potremmo avere finito, ma c'è un aspetto tecnico che caratterizza la meccanica dei robot (LEGO e non). La potenza applicabile a un motore non può essere qualsiasi, ma deve essere compresa in un intervallo di valori [0 ... 100]. Purtroppo non c'è nessuna garanzia che i calcoli di correzione fatti in precedenza portino le potenze ai motori a numeri negativi o maggiori di 100. Questo potrebbe danneggiare i motori o comportare malfunzionamenti nel sistema di controllo. L'ultimo passo del nostro controllore consiste nel riportare eventuali valori "fuori scala" entro il range ammissibile. Il termine tecnico per questa operazione è **normalizzazione**.

Semplicemente controlliamo che il valore sia nel range, altrimenti lo rimettiamo a fondo scala. Naturalmente lo facciamo sia per il motore DX sia per il motore DX.



In pratica

15. Leggo variabile Potenza DX
16. SE (Potenza DX < 0)
        Salvo Potenza DX = 0
17. SE (Potenza DX > 100)
        Salvo Potenza DX = 100

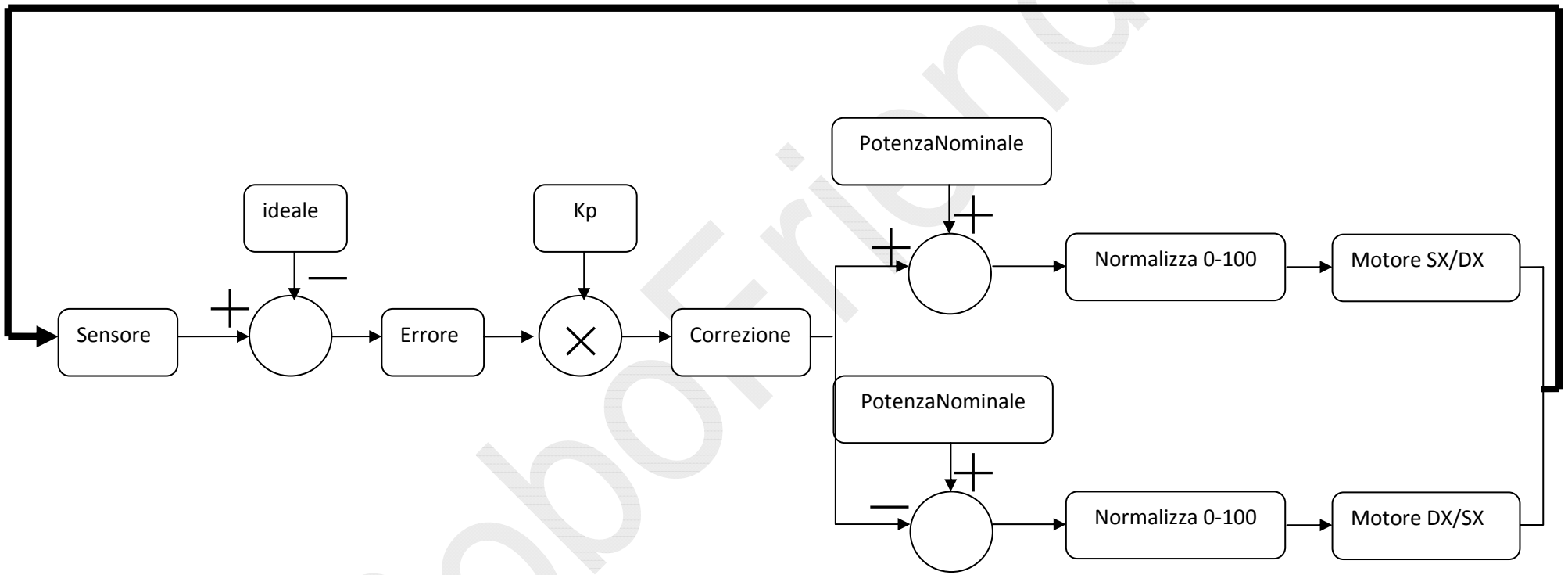
18. Leggo variabile Potenza SX
19. SE (Potenza SX < 0)
        Salvo Potenza SX = 0
20. SE (Potenza SX > 100)
        Salvo Potenza SX = 100

5. Combiniamo tutto in un loop

Ora è tutto a posto, possiamo mettere insieme tutti i vari momenti del nostro controllore e chiudere tutto in un **Loop di controllo** che continua all'infinito, o almeno fin quando decidiamo di interromperlo dall'esterno.

Per semplicità, nella figura seguente, riassumiamo lo schema senza includere i blocchi dettagliati di normalizzazione (basta replicare due volte il diagramma della pagina precedente). Per i più pratici di programmazione, questo si può fare comodamente definendo una funzione apposiuta o "MyBlock" nell'ambiente di programmazione LEGO.

Schema generale di controllore P



Dettagli tecnici

La normalizzazione è molto importante specialmente nei robot LEGO, perché l'immissione di potenze negative o fuori scala potrebbe dare comportamenti imprevedibili (specialmente nei robot NXT).

Il classico problema di tutti i controllori P è: "**A quale valore devo impostare il guadagno Kp**?"

Cominciamo subito a dire che non esiste un criterio unico e decisivo per questo problema; le scienze ingegneristiche offrono qualche idea (si chiama Ziegler-Nichols) ma non sempre si adattano bene ai casi di robotica educativa. Possiamo però dare qualche linea guida di sicuro valore generale.

1) In generale, esiste una relazione inversa fra guadagno e velocità del robot: quanto più il nostro robot va veloce, tanto più dovremo ridurre il guadagno. La spiegazione è in fondo semplice se pensiamo ad una automobile con il guidatore: quanto più l'auto va veloce, tanto meno si deve girare il volante per cambiare direzione.

2) In generale, il valore del guadagno cambia anche se sappiamo che dobbiamo affrontare grandi o piccole variazioni. Esempio: pensiamo ad un inseguitore di linea: se la linea da seguire è dritta, potremo anche usare un guadagno basso e alta velocità perché ci si aspetta che le correzioni siano ridotte. Se invece dobbiamo affrontare un percorso con una linea di curve anche strette, dovremo apportare correzioni più importanti quindi avremo bisogno di un guadagno più elevato (e quindi dovremo andare più piano!). Se questo sembra difficile, basta pensare a cosa dovete fare quando in auto affrontate una curva in sicurezza ... di solito rallentate, no?

3) Gli esperti usano una procedura empirica per impostare il guadagno:

a) si decide a quale velocità si desidera procedere (si imposta la PotenzaNominale)
b) si imposta un guadagno molto basso e si osserva il comportamento
c) se il robot oscilla, abbasso ancora il guadagno, se invece non oscilla aumento il guadagno
d) ripeto c) fino a trovare il "valore limite" del guadagno per cui il robot mantiene il comportamento desiderato e non oscilla.
e) NOTA: se non riesco a trovare un valore di Kp adeguato, riduco la PotenzaNominale (avevo scelto un valore troppo alto) e ricomincio da b)

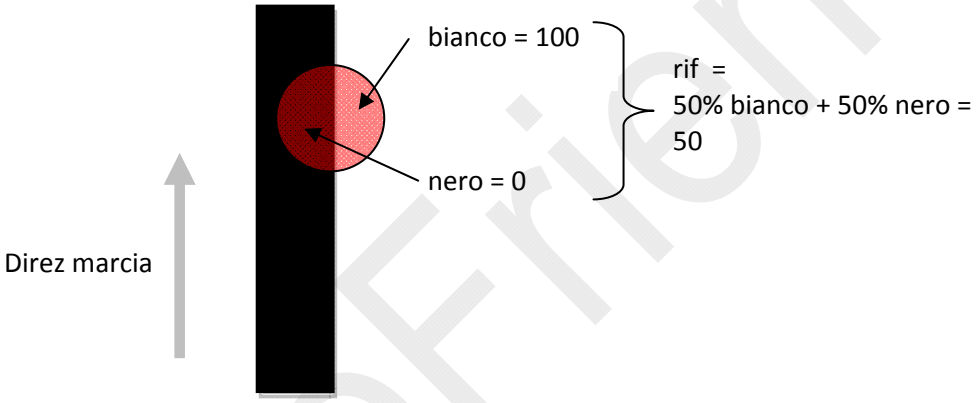
Applicazioni

Riprendiamo le impostazioni descritte a pagina 1 nei diversi casi pratici.

Inseguitore di linea

Scelta del sensore: In questo caso il sensore è ottico ("luminosità riflessa" / "Reflected Light Intensity" --> *non* sensore di colore!) e punta verso il tappeto. Esso proietta un cerchio di luce (rosso) che descrive il suo campo visivo. Il sensore dà un valore proporzionale al livello di "bianco" misurato nel campo visivo, ovvero nell'intervallo 0==NERO e 100==BIANCO.

Riferimento: Per una linea ideale nera su fondo ideale bianco, il nostro sensore dovrebbe misurare **RIF = 50**, per seguire il contorno della linea.



→ Il fatto è che non avremo mai un fondo bianco e una linea nera. Questo richiede di **misurare sul campo** (es con PortView) i valori reali di bianco e nero intorno alla linea da seguire. Questa operazione è *indispensabile* e si chiama **calibrazione**. Il riferimento sarà dato da una semplice media aritmetica:

$$RIF = \frac{BiancoMisurato + NeroMisurato}{2}$$

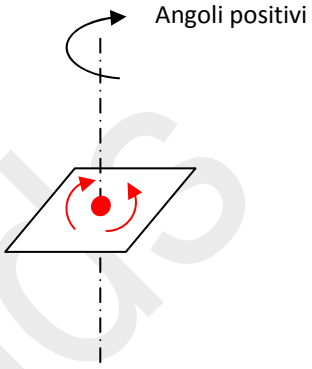
→ Il comportamento del sensore può dipendere in modo critico dalla luminosità ambiente, è necessario che un sensore ottico per line follower sia **schermato**, cioè protetto da un contorno che blocchi la visibilità di luce esterna da parte del sensore.

Potenza: da decidere, usiamo ad esempio P =20

→ [Sommare al motore DX/sottrarre al motore SX] **oppure** [Sommare al motore SX/sottrarre al motore DX] la correzione di potenza farà sì che il robot segua il contorno destro **oppure** il contorno sinistro della linea, rispettivamente. Nella figura sopra si vuole inseguire il contorno destro (cioè il robot avrà la linea nera alla sinistra del sensore), quindi sommeremo al motore DX e sottrarremo al motore SX.

Pilota Automatico su rotta costante

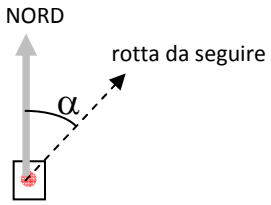
Scelta del sensore: In questo caso il sensore è un giroscopio. Esso in generale misura la velocità di rotazione intorno ad un asse. Nel caso LEGO (più comodo da usare), il sensore giroscopico fornisce anche una misura dell'angolo del sensore rispetto ad un riferimento. È sempre bene montare il sensore in asse longitudinale con il robot (cioè nella linea di mezzo fra destra e sinistra). L'asse è indicato da un puntino rosso e due archi che suggeriscono le rotazioni misurate.



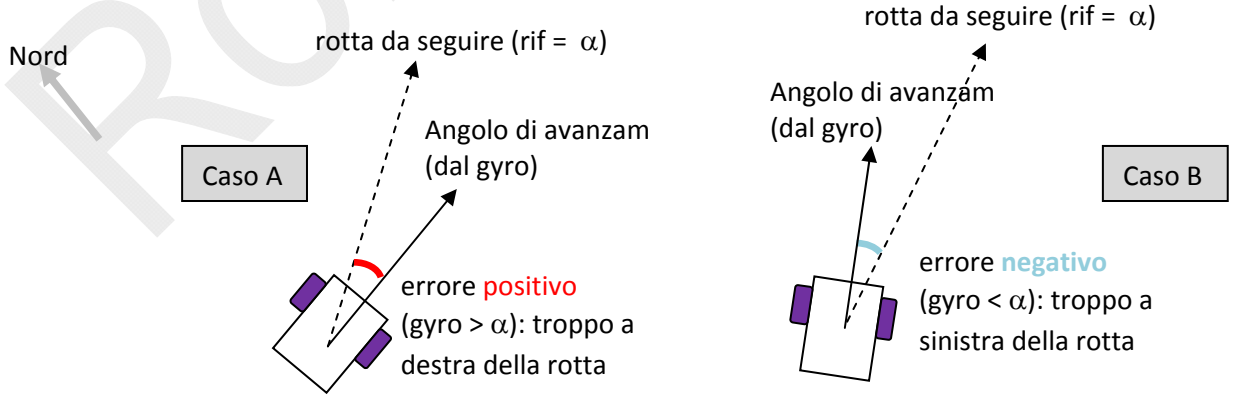
Nel nostro caso assumeremo che le rotazioni orarie diano misure di angoli positivi, mentre rotazioni antiorarie daranno angoli negativi. Questa caratteristica dipenderà solo da come avremo montato il sensore sul robot (con il pallino verso l'alto o verso il basso).

Riferimento: l'angolo di un giroscopio è misurato rispetto a una direzione di riferimento, che chiameremo per convenzione "Nord" (ma non ha niente a che vedere con quello della bussola!). Per impostare il Nord si deve applicare il comando di **Reset** al sensore giroscopico, che automaticamente assumerà il valore 0. Questo equivale a calibrare il sensore. Il Nord può essere la orientazione del campo di gara o la direzione corrente del robot prima di impostare la rotta desiderata, o una direzione che scegliamo e impostiamo di volta in volta.

Il nostro riferimento sarà dato, in ogni caso, dall'angolo di rotta desiderata rispetto al Nord; indicheremo con la lettera greca α 'alfa' questo angolo rispetto al Nord. Nella figura a lato si vede che il Nord è allineato (dopo Reset) con il fronte del gyro.



Se il gyro misura angoli in senso orario, abbiamo un errore **positivo** quando il robot deriva sulla destra della rotta voluta (A), mentre l'errore è **negativo** quando il robot deriva verso **sinistra** (B). La correzione di potenza avrà gli stessi segni essendo moltiplicata per Kp.



In questo caso, dovremo sommare la correzione alla potenza sul motore DX, e sottrarre la correzione alla potenza sul motore SX (in questo modo, con errori positivi il robot virerà verso sinistra, e con errori negativi virerà verso destra, riallineandosi sulla rotta di riferimento).

Nella pratica, per un autopilota su rotta costante dovremo:

(Caso con giroscopio che misura angoli positivi in senso orario)

1. Reset del gyro per impostare il Nord
2. Decidere la direzione di riferimento α
3. Decidere la Potenza Nominale di marcia
4. Scrivere il solito loop di controllo come a pagina 7, dove il dato del sensore è l'angolo misurato dal giroscopio
5. Impostare il valore di Kp con il solito criterio

Se il giroscopio misura angoli in segno opposto, il principio non cambia ma bisogna prestare attenzione ai segni, errori positivi indicano deriva a sinistra della rotta, e dovremo impostare la correzione della marcia affinché il robot viri verso destra; errori negativi indicano deriva a destra della rotta, e dovremo impostare la correzione della marcia affinché il robot viri verso sinistra.