

Controllo di un robot mobile

1 Introduzione

Il controllo di un robot mobile presenta difficoltà di tipo diverso rispetto al controllo di un manipolatore a causa della presenza di *vincoli non oloноми*, vincoli che limitano lo spazio delle direzioni di moto ammissibili (e legati, nel caso dell'uniciclo, al fatto che una ruota può rotolare ma non traslare lateralmente nella direzione del suo asse di rotazione) ma non lo spazio delle posizioni che il robot può assumere (o per lo meno possono ridurlo ma in misura minore rispetto a quello delle velocità). A tutti è nota la difficoltà di manovra nel parcheggiare un'automobile: il veicolo non può traslare lateralmente ma può comunque raggiungere una posizione laterale (vedasi Fig. 1) con opportune manovre. Si noti come la traiettoria riportata in Fig. 1 possa essere effettuata anche a livello infinitesimo e ripetuta più volte: in questo modo si ottiene approssimativamente una vera e propria traslazione laterale del veicolo.

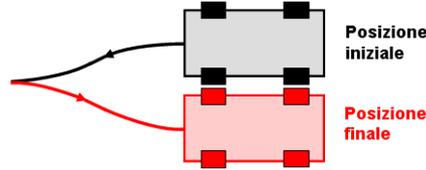


Figure 1: Spostamento laterale di un'automobile

Nel nostro caso, l'uniciclo visto in precedenza, con equazioni:

$$\begin{cases} \dot{x} = v_1 \cos\theta \\ \dot{y} = v_1 \sin\theta \\ \dot{\theta} = v_2 \end{cases} \quad (1)$$

è soggetto a un vincolo sulle velocità del seguente tipo:

$$\dot{x} \sin\theta - \dot{y} \cos\theta = 0 \quad (2)$$

che è lineare sulle velocità, cioè del tipo $A(\vec{q}) \dot{\vec{q}} = 0$, dove $\vec{q} \equiv x_r = [x, y, \theta]^T$ sono le coordinate del robot e $A(\vec{q}) = [\sin\theta, -\cos\theta, 0]$. Avendo uno spazio di dimensione 3 (infatti $\vec{q} = [x, y, \theta]^T$), un vincolo riduce lo spazio delle velocità $\dot{\vec{q}}$ ammissibili a uno spazio di dimensione 2, la cui base (il nucleo di $A(\vec{q})$) è data da (per esempio):

$$\left\{ \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \quad (3)$$

La (1) costituisce in effetti una soluzione $\dot{\vec{q}}$ generica della (2) con v_1 e v_2 i coefficienti della combinazione lineare degli elementi della base dello spazio soluzione riportati in (3).

Opportuni strumenti matematici (in particolare il Teorema di Frobenius che non vediamo in questo corso) permettono di mostrare che il vincolo (2) non è integrabile, cioè le posizioni che il robot può assumere non sono solo quelle generate dai vettori in (3), ma esiste una direzione in più (che si ottiene con un'operazione matematica - non studiata in questo corso - effettuata sulle due velocità ammissibili, nota come *parentesi di Lie*). Poiché $2 + 1 = 3$, risulta che il veicolo può in effetti raggiungere qualsiasi posizione con qualsiasi orientamento nel piano, come confermato anche dall'esperienza. Si dice che il robot è completamente *controllabile*. Vincoli che non sono integrabili come quello considerato vengono detti *non*

olonomi. Si intuisce che, avendo dei vincoli sulle velocità che non restringono lo spazio delle configurazioni possibili del robot, la pianificazione del moto risulta alquanto più complessa, come risulta evidente dalla manovra riportata in Fig. 1 dove, anziché spostarci semplicemente verso sinistra (verso il basso in figura), siamo costretti a fare tutta una sequenza di movimenti.

Osservazione 1. Il vincolo (2), e quindi lo spazio soluzione corrispondente (3), dipendono da θ e quindi hanno una validità *locale*, cambiando lungo il moto del robot.

Osservazione 2. Non tutti i vincoli sulle velocità sono necessariamente non olonomi: alcuni possono infatti essere integrati e ricondotti a vincoli sulle posizioni. Si consideri a titolo di esempio il moto di una pallina in una guida posta su un piano orizzontale. Sia $y = 0.5x + 1$ l'equazione della guida nel piano (Fig. 2). È evidente come la biglia presenti un vincolo sulle velocità del tipo

$$\dot{y} = 0.5\dot{x} \quad \rightarrow \quad \dot{x} - 2\dot{y} = 0$$

Tuttavia questo vincolo sulle velocità è integrabile (è cioè olonomo) e conduce al vincolo di posizione $y = 0.5x + 1$. Pertanto il problema in questo caso non presenta difficoltà: una volta ristretto lo spazio delle posizioni della pallina alla guida (in base al vincolo sulle posizioni), la pallina non presenta più in questo spazio nessun vincolo sulle velocità e il suo moto può essere pianificato senza difficoltà.

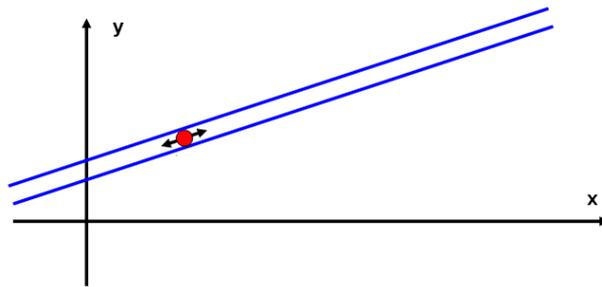


Figure 2: Moto di una biglia in una guida sul piano

Osservazione 3. In generale, l'integrabilità di un vincolo sulle velocità conduce come si è detto a un vincolo sulle posizioni del robot. Un vincolo sulle posizioni è generalmente più semplice da trattare nella pianificazione del moto: basta restringere il problema al sottospazio delle posizioni ammissibili e pianificare il moto in questo sottospazio, a questo punto senza più vincoli sulle velocità. È questo in effetti il caso di un manipolatore planare a n link: a causa dei vincoli imposti dai giunti, i $6 \cdot n$ gradi di libertà posseduti dagli n link del manipolatore si riducono a n (le n variabili di giunto). La pianificazione del moto in tale spazio risulta semplice in quanto nello spazio delle posizioni ammissibili (\mathcal{R}^n) le velocità in linea di principio non sono più vincolate e qualunque movimento è possibile (chiaramente compatibilmente con la presenza di ostacoli, con i fine corsa dei giunti e con le coppie massime erogabili dai motori dei giunti).

Come detto sopra, in questo corso non ci soffermiamo sullo studio delle metodologie per valutare la non olonomia di vincoli di velocità né sul problema di valutare la completa controllabilità di un robot soggetto a vincoli non olonomi. L'unica cosa che studieremo è il problema di individuare una legge di controllo (a catena chiusa) che porti il veicolo (l'uniciclo) in una posizione desiderata (x_d, y_d) , con orientamento finale non specificato (una volta che il robot si è fermato è comunque banale ruotarlo da fermo per raggiungere un qualsiasi orientamento finale assegnato, anche se, volendo raggiungere una posizione desiderata con orientamento anche assegnato, potrebbe essere più efficiente applicare altre leggi di controllo).

Risolveremo il problema sotto l'ipotesi semplificativa che sia possibile assegnare istantaneamente le velocità v_1 e v_2 del veicolo: proporremo una legge proporzionale che si dimostra essere convergente (nel senso che porta a zero l'errore di posizionamento e_p definito come la distanza tra posizione effettiva e posizione desiderata).

2 Controllo di posizione di un unicycle (regolazione)

Si consideri il problema seguente: dato un robot mobile del tipo *unicycle*, quindi con equazioni cinematiche espresse in (1), progettare un controllo in retroazione dallo stato (x, y, θ) (stato che si assume noto) in grado di condurre l'unicycle in una posizione desiderata (x_d, y_d) del piano con orientamento non specificato. Questo tipo di problema è detto di *regolazione parziale*, dove il termine *parziale* si riferisce al fatto che non viene specificato l'orientamento ma solo la posizione finale desiderata.

2.1 Legge proporzionale nelle velocità

Si presenta una soluzione al problema considerato assumendo come variabili di controllo le velocità longitudinale e rotazionale dell'unicycle. Questo è giustificato dal fatto che in molti casi il software di controllo ad alto livello dell'unicycle richiede l'assegnazione del valore desiderato di queste due velocità, trasferendo ad anelli di controllo a più basso livello il problema della scelta delle coppie motrici da assegnare alle ruote per ottenere tali velocità.

L'idea alla base del controllo è quella di assegnare una velocità longitudinale v_1 proporzionale alla distanza che manca all'obiettivo proiettandola sulla direzione del veicolo stesso e, contemporaneamente, assegnare a v_2 un valore che tenda a ruotare il veicolo verso l'obiettivo (x_d, y_d) . In formule:

$$v_1 = K_{v_1} e_p \cdot \cos\phi \quad (4)$$

in cui K_{v_1} è una costante positiva, $e_p = \sqrt{(x_d - x)^2 + (y_d - y)^2}$ è la distanza dall'obiettivo e $\phi = \theta - \theta_d$ l'angolo tra la direzione θ del veicolo e quella θ_d che punta verso l'obiettivo (si veda la Fig. 3). Chiaramente

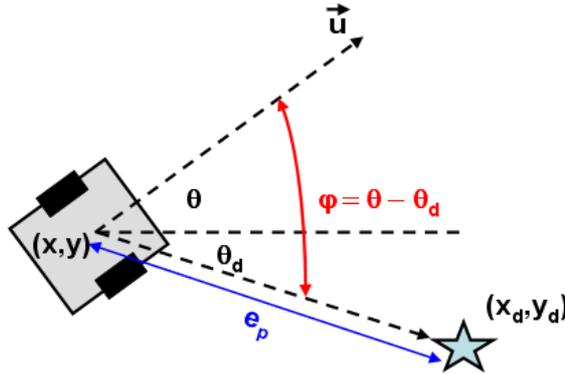


Figure 3: Grandezze utilizzate dalla legge proporzionale nelle velocità

sarà $\theta_d = \text{atan2}(y_d - y, x_d - x)$. Volendo orientare il veicolo verso l'obiettivo, si utilizza la seguente legge:

$$v_2 = K_{v_2}(\theta_d - \theta) \quad (5)$$

dove K_{v_2} è un'altra costante positiva. Si noti che $e_p = e_p(x, y)$ e $\phi = \phi(x, y, \theta)$ (essendo anche $\theta_d = \theta_d(x, y)$) per cui in effetti $v_1 = v_1(x, y, \theta)$ e $v_2 = v_2(x, y, \theta)$ è un controllo in retroazione dallo stato (x, y, θ) dell'unicycle. Volendo scrivere v_1 direttamente in funzione dello stato e delle coordinate del punto desiderato si procede nel seguente modo. Si consideri il prodotto scalare tra il versore $\vec{u} = [\cos\theta, \sin\theta]^T$ che individua la direzione di avanzamento del veicolo e quello, \vec{e}_p , che punta verso l'obiettivo e che è dato da $\vec{e}_p = (x_d - x, y_d - y)/e_p$ (essendo e_p la distanza tra robot e obiettivo). Si ha:

$$\vec{u} \cdot \vec{e}_p = \frac{(x_d - x)\cos\theta + (y_d - y)\sin\theta}{e_p}$$

Sappiamo che il prodotto scalare tra due versori è pari al coseno dell'angolo ϕ tra essi compreso. Pertanto:

$$\cos\phi = \frac{(x_d - x)\cos\theta + (y_d - y)\sin\theta}{e_p}$$

che, sostituito nella (4), fornisce:

$$v_1 = K_{v_1} [(x_d - x)\cos\theta + (y_d - y)\sin\theta] \quad (6)$$

In definitiva, la legge di controllo che si propone in questa sezione è data dalle equazioni (5) e (6). Si può dimostrare che, con questo controllo, $e_p \rightarrow 0$ al tendere di t all'infinito, cioè che l'errore di posizionamento tende asintoticamente a 0. La dimostrazione di questo fatto, essendo il problema non lineare, si fa scegliendo una funzione di tipo Lyapunov opportuna. Nello specifico, si considera

$$V(x, y) = \frac{1}{2} [(x - x_d)^2 + (y - y_d)^2] = \frac{1}{2} e_p^2 \quad (7)$$

la quale è limitata inferiormente ($V \geq 0$) e cresce allontanandosi dall'obiettivo, dove vale 0 (cioè $V(x_d, y_d) = 0$). Questa V non dipende da θ . Quando il veicolo si muove sotto l'azione di (5) e (6), la V non può aumentare. Infatti, se calcoliamo $\dot{V} = dV/dt$ lungo le traiettorie dell'uniciclo si trova:

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x} \dot{x} + \frac{\partial V}{\partial y} \dot{y} = (x - x_d)\dot{x} + (y - y_d)\dot{y} = (x - x_d)v_1\cos\theta + (y - y_d)v_1\sin\theta = \\ &= [(x - x_d)\cos\theta + (y - y_d)\sin\theta] \cdot K_{v_1} [(x_d - x)\cos\theta + (y_d - y)\sin\theta] = \\ &\quad -K_{v_1} [(x_d - x)\cos\theta + (y_d - y)\sin\theta]^2 \end{aligned}$$

che è una quantità non positiva, cioè $\dot{V} \leq 0$ per ogni x, y e θ . Questo già permette di concludere che e_p è limitato per ogni t e che non può aumentare. Per dimostrare che converge a 0, si utilizza un risultato dell'analisi matematica (noto col nome di Lemma di Barbalat), il quale dice che se una funzione V è limitata inferiormente, $\dot{V} \leq 0$ e \dot{V} è uniformemente continua (cosa che si può verificare controllando che \ddot{V} sia limitata), allora $\lim_{t \rightarrow \infty} \dot{V} = 0$.

Ora, nel nostro caso, è facile verificare che \ddot{V} è limitata, per cui

$$\lim_{t \rightarrow \infty} (x_d - x)\cos\theta + (y_d - y)\sin\theta = 0. \quad (8)$$

La precedente dice che la proiezione dell'errore e_p sull'asse del veicolo \vec{u} tende a 0 per $t \rightarrow \infty$. Ma questo non può avvenire per $(x, y) \neq (x_d, y_d)$, cioè per $e_p \neq 0$. Se infatti per assurdo ciò avvenisse, sarebbe $v_1 = 0$ e la (5) farebbe ruotare il veicolo facendo tornare diversa da 0 la proiezione dell'errore e_p sull'asse \vec{u} . Pertanto la (8) può annullarsi solo per $(x, y) = (x_d, y_d)$, il che dimostra, come, asintoticamente, il veicolo raggiunge la posizione desiderata. \square

La convergenza a 0 dell'errore di posizionamento e_p appena dimostrata vale per qualsiasi condizione iniziale (x_0, y_0, θ_0) e per qualsiasi valore positivo delle costanti K_{v_1} e K_{v_2} . Tuttavia, implementando tale legge (cioè la (5) e la (6)) su Processing, si nota un comportamento oscillatorio spiacevole (che peraltro non compromette la convergenza) quando θ_d è prossimo ai 180° . Vediamo di capire cosa accade e come si può porvi rimedio. Innanzitutto, l'implementazione su Processing del veicolo controllato mediante (5) e (6) è necessariamente un'implementazione a tempo discreto¹ con tempo di campionamento dt pari a $1/60$ di secondo (per default infatti, Processing esegue la funzione `draw()` 60 volte al secondo). Nel caso discretizzato abbiamo:

$$\theta_{k+1} = \theta_k + v_{2,k} \cdot dt$$

e

$$v_{2,k} = K_{v_2}(\theta_{d,k} - \theta_k).$$

Se scegliamo $K_{v_2} = 1/dt$, chiaramente $\theta_{k+1} = \theta_k + K_{v_2}(\theta_{d,k} - \theta_k) \cdot dt = \theta_{d,k}$ e il comportamento oscillatorio sopra menzionato non si osserva. Tuttavia è ragionevole supporre che il veicolo non sia in grado di ruotare nella direzione desiderata istantaneamente (cioè nel nostro caso in $1/60$ di secondo) ma passi per un certo numero di orientamenti intermedi (compresi tra θ_k e $\theta_{d,k}$): questo si ottiene se si considerano scelte di valori più contenuti (e più realistici) di K_{v_2} . Con queste scelte si osserva un fenomeno oscillatorio dovuto al fatto che la funzione `atan2` restituisce angoli compresi tra -180° e $+180^\circ$ per cui 181° viene restituito come -179° . Questo causa il fenomeno rappresentato in Fig. 4: essendo $\theta_d = \text{atan2}(y_d - y, x_d - x)$, quando il robot supera la linea tratteggiata mostrata in figura, l'angolo θ_d passa istantaneamente da $+180^\circ$ a

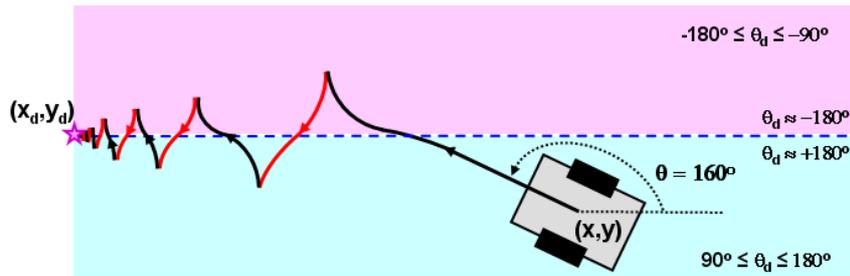


Figure 4: Movimento oscillatorio dell'uniciclo: i tratti in rosso sono percorsi in retromarcia

-180° (o viceversa) e si assiste a un moto oscillatorio come quello mostrato in figura. Per ovviare a questo inconveniente, che tra l'altro rallenta di molto il moto verso l'obiettivo, si può usare il seguente espediente²:

```

 $\theta_d = \text{atan2}(y_d - y, x_d - x) + n_g \cdot 2\pi$ 
if ( $|\theta_d + 2\pi - \theta| < |\theta_d - \theta|$ )
{
     $\theta_d = \theta_d + 2\pi;$ 
     $n_g ++;$ 
}
else
{
    if ( $|\theta_d - 2\pi - \theta| < |\theta_d - \theta|$ )
    {
         $\theta_d = \theta_d - 2\pi;$ 
         $n_g --;$ 
    }
}

```

Queste linee di codice permettono di scegliere come soluzione dell'*atan2* quella più vicina all'angolo θ attuale. Se, per esempio, $\theta = 179^\circ$ e l'*atan2* restituisce $\theta_d = -178^\circ$, il codice sopra si accorge che $\theta_d + 360^\circ = 182^\circ$ è molto più vicino a 179° (solo 3° di differenza) di quanto non sia -178° (ben 357° di differenza!). Sceglierà quindi $\theta_d = 182^\circ$ e si ricorderà con n_g (inizializzata a 0) che da adesso in poi θ_d non è data semplicemente dall'*atan2* ma dall'*atan2* + 2π (fino al prossimo aggiornamento di n_g).

¹Nella versione discretizzata, a differenza del caso a tempo continuo, si ottiene un comportamento stabile se e solo se le costanti K_{v1} e K_{v2} non superano un certo valore, come è anche possibile verificare in simulazione su Processing.

²Si tratta dello stesso espediente introdotto nello sketch *seguiCerchio.pde* in cui un manipolatore planare doveva inseguire un obiettivo che si muoveva lungo un cerchio