



Controllo del moto e robotica industriale

Pianificazione di traiettorie

Prof. Paolo Rocco (paolo.rocco@polimi.it)

Pianificazione della traiettoria



Con la **pianificazione della traiettoria** si intende stabilire la modalità con cui si vuole che evolva il movimento del manipolatore, da una postura iniziale ad una postura finale. Si tratta di definire sia il percorso geometrico sia la legge di moto da realizzare (ossia la dipendenza temporale di posizioni, velocità ed accelerazioni) .

La corretta pianificazione della traiettoria ha estrema importanza, perché comporta che la traiettoria possa essere eseguita da parte del sistema di controllo del moto in anello chiuso, senza incorrere nei limiti di saturazione degli attuatori o provocare dannose sollecitazioni meccaniche alla struttura.

In generale ci si pone l'obiettivo di pianificare traiettorie a curvatura quanto più possibile regolare.

N.B. È importante non confondere i concetti di *percorso* (luogo geometrico dei punti che il manipolatore deve descrivere) e *traiettoria* (percorso su cui sia stata specificata una legge oraria).

Ingressi e uscite dell'algoritmo



Per l'algoritmo di pianificazione della traiettoria potremmo definire i seguenti:

“Ingressi”

- Definizione del percorso
- Vincoli al percorso
- Vincoli dovuti alla dinamica del manipolatore

“Uscite”

- Traiettorie (ai giunti o all'organo terminale) espresse come sequenza dei valori assunti da posizione, velocità, accelerazione

In generale l'utente specifica un numero ristretto di parametri:

- Per il *percorso*: punti estremi, eventuali punti intermedi, primitive geometriche
- Per la *legge di moto*: tempo complessivo, velocità e/o accelerazioni massime, velocità e/o accelerazioni in determinati punti

Spazio operativo e spazio dei giunti



Traiettorie nello spazio operativo: si definisce il percorso dell'organo terminale del manipolatore nel comune spazio cartesiano.

- La descrizione del compito è naturale
- Sono facilmente descrivibili i vincoli sul percorso
- Punti di singolarità o gradi di mobilità ridondanti generano problemi
- Occorre in ogni caso uno stadio di inversione cinematica per passare allo spazio di attuazione

Traiettorie nello spazio di giunto: si specifica direttamente l'andamento desiderato per la posizione, la velocità e l'accelerazione dei singoli giunti.

- Consente di risolvere direttamente problemi relativi a singolarità cinematiche
- È una modalità di interesse quando i giunti non danno luogo ad un moto coordinato oppure quando interessa solo che gli assi si portino da una posizione iniziale ad una finale (e non ha interesse il movimento risultante nello spazio operativo)
- Per i punti estremi può essere necessaria un'inversione cinematica delle posture assegnate dall'utente



Parte I

Moto punto-punto nello spazio dei giunti

Traiettorie nello spazio dei giunti



Con la pianificazione della traiettoria nello spazio dei giunti si desidera generare una funzione $q(t)$ che interpola i valori assegnati per le variabili di giunto, rispettando i vincoli imposti.

Alcuni criteri per la scelta della traiettoria possono essere:

- Bassa complessità computazionale
- Continuità di posizioni e velocità (ed eventualmente di accelerazioni)
- Minimizzazione di effetti indesiderati (curvatura non regolare)

Operiamo una distinzione:

Moto punto-punto:

si specificano solo i punti estremi ed il tempo di transizione

Moto su percorso assegnato:

si specificano anche dei punti intermedi

N.B. Nel seguito faremo riferimento alla pianificazione della traiettoria per una singola variabile di giunto.

Traiettorie polinomiali



Il caso più semplice di pianificazione della traiettoria per moto punto-punto si ha quando sono specificate alcune condizioni iniziali e finali sulla posizione, velocità ed eventualmente anche su accelerazione e jerk (derivata dell'accelerazione) ed il tempo di percorrenza.

Si possono prendere in considerazione funzioni polinomiali del tipo:

$$q(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n$$

Più alto è il grado n del polinomio, più condizioni al contorno si possono soddisfare e più *smooth* (dolce) sarà la traiettoria.

Traiettorie cubiche



Si supponga che siano specificate le seguenti condizioni al contorno:

- un istante iniziale ed un istante finale t_i e t_f
- posizione e velocità iniziale q_i e $\dot{q}_i$
- posizione e velocità finale q_f e $\dot{q}_f$

Si hanno quindi quattro condizioni al contorno, per poter rispettare le quali occorre utilizzare un polinomio di grado almeno pari a tre (cubica):

$$q(t) = a_0 + a_1(t - t_i) + a_2(t - t_i)^2 + a_3(t - t_i)^3$$

Imponiamo le condizioni al contorno:

$$q(t_i) = q_i$$

$$\dot{q}(t_i) = \dot{q}_i$$

$$q(t_f) = q_f$$

$$\dot{q}(t_f) = \dot{q}_f$$

si ottiene:

$$a_0 = q_i$$

$$a_1 = \dot{q}_i$$

$$a_2 = \frac{-3(q_i - q_f) - (2\dot{q}_i + \dot{q}_f)T}{T^2}$$

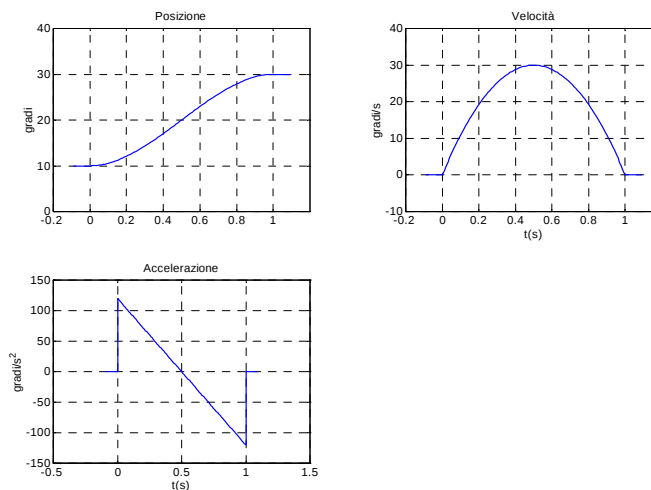
$$a_3 = \frac{2(q_i - q_f) + (\dot{q}_i + \dot{q}_f)T}{T^3}$$

$$T = t_f - t_i$$

Traiettorie cubiche: esempio



$$t_i = 0, t_f = 1s, q_i = 10^\circ, q_f = 30^\circ, \dot{q}_i = \dot{q}_f = 0^\circ/s$$



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [9]

Polinomi di grado cinque



Per imporre condizioni anche sulle accelerazioni, occorre passare a polinomi di grado 5:

$$q(t) = a_0 + a_1(t - t_i) + a_2(t - t_i)^2 + a_3(t - t_i)^3 + a_4(t - t_i)^4 + a_5(t - t_i)^5$$

Imponendo le condizioni al contorno:

$$q(t_i) = q_i \quad q(t_f) = q_f$$

$$\dot{q}(t_i) = \dot{q}_i \quad \dot{q}(t_f) = \dot{q}_f$$

$$\ddot{q}(t_i) = \ddot{q}_i \quad \ddot{q}(t_f) = \ddot{q}_f$$

si ottiene:

$$a_0 = q_i$$

$$a_1 = \dot{q}_i$$

$$a_2 = \frac{1}{2} \ddot{q}_i$$

$$a_3 = \frac{20(q_f - q_i) - (8\dot{q}_f + 12\dot{q}_i)T - (3\ddot{q}_f - \ddot{q}_i)T^2}{2T^3}$$

$$a_4 = \frac{30(q_i - q_f) + (14\dot{q}_f + 16\dot{q}_i)T + (3\ddot{q}_f - 2\ddot{q}_i)T^2}{2T^4}$$

$$a_5 = \frac{12(q_f - q_i) - 6(\dot{q}_f + \dot{q}_i)T - (\ddot{q}_f - \ddot{q}_i)T^2}{2T^5}$$

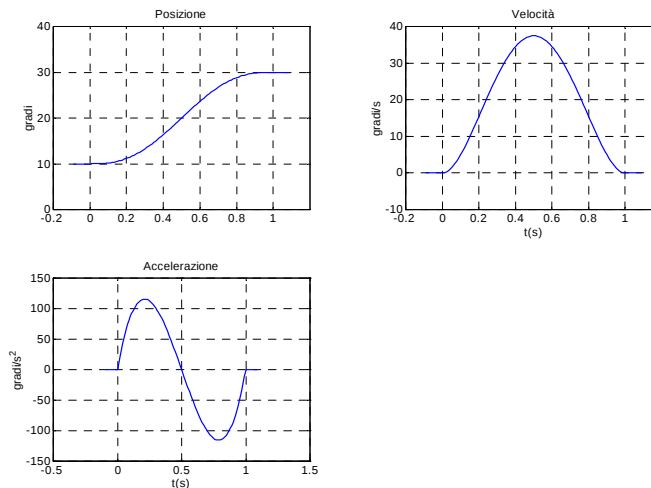
$$T = t_f - t_i$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [10]

Polinomio di grado cinque: esempio



$$t_i = 0, t_f = 1s, q_i = 10^\circ, q_f = 30^\circ, \dot{q}_i = \dot{q}_f = 0^\circ/s, \ddot{q}_i = \ddot{q}_f = 0^\circ/s^2$$



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [11]

Traiettoria armonica



La traiettoria armonica generalizza l'equazione di un moto armonico, nel quale l'accelerazione è proporzionale allo spostamento, ma di segno opposto. Una traiettoria armonica presenta derivate continue in tutti i punti interni all'intervallo temporale della traiettoria, per qualsiasi ordine di derivazione.

Le equazioni sono le seguenti:

$$q(t) = \frac{q_f - q_i}{2} \left(1 - \cos \left(\frac{\pi(t - t_i)}{t_f - t_i} \right) \right) + q_i$$

$$\dot{q}(t) = \frac{\pi(q_f - q_i)}{2(t_f - t_i)} \sin \left(\frac{\pi(t - t_i)}{t_f - t_i} \right)$$

$$\ddot{q}(t) = \frac{\pi^2(q_f - q_i)}{2(t_f - t_i)^2} \cos \left(\frac{\pi(t - t_i)}{t_f - t_i} \right)$$

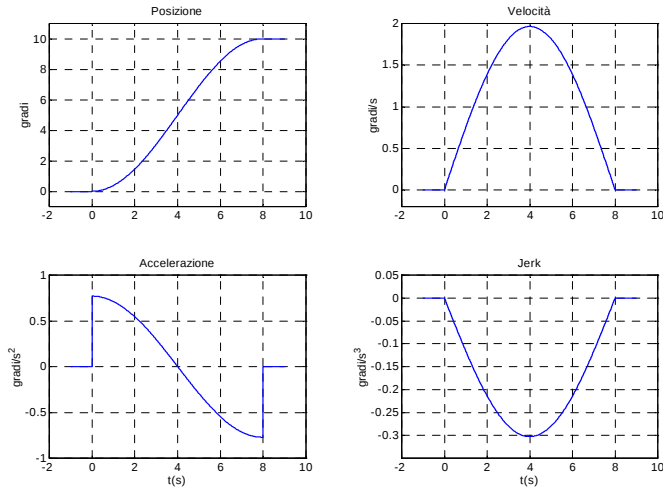
$$j(t) = -\frac{\pi^3(q_f - q_i)}{2(t_f - t_i)^3} \sin \left(\frac{\pi(t - t_i)}{t_f - t_i} \right) \quad (\text{jerk})$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [12]

Traiettoria armonica (esempio)



$$t_i = 0, t_f = 8s, q_i = 0^\circ, q_f = 10^\circ$$



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [13]

Traiettoria cicloidale



La traiettoria armonica presenta discontinuità nell'accelerazione negli istanti iniziale e finale, e quindi valori non definiti (o infiniti) di jerk. Un'alternativa è la traiettoria cicloidale, che presenta continuità anche nell'accelerazione. Le formule sono:

$$q(t) = \left(q_f - q_i \right) \left(\frac{t - t_i}{t_f - t_i} - \frac{1}{2\pi} \sin \left(\frac{2\pi(t - t_i)}{t_f - t_i} \right) \right) + q_i$$

$$\dot{q}(t) = \frac{q_f - q_i}{t_f - t_i} \left(1 - \cos \left(\frac{2\pi(t - t_i)}{t_f - t_i} \right) \right)$$

$$\ddot{q}(t) = \frac{2\pi(q_f - q_i)}{(t_f - t_i)^2} \sin \left(\frac{2\pi(t - t_i)}{t_f - t_i} \right)$$

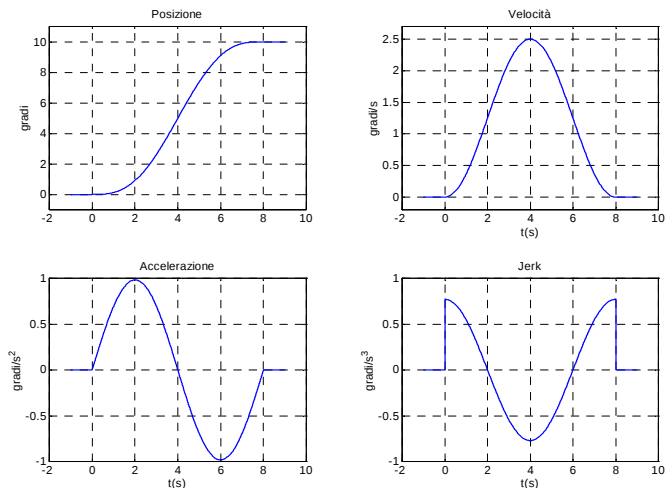
$$j(t) = \frac{4\pi^2(q_f - q_i)}{(t_f - t_i)^3} \cos \left(\frac{2\pi(t - t_i)}{t_f - t_i} \right)$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [14]

Traiettoria cicloidale (esempio)



$$t_i = 0, t_f = 8s, q_i = 0^\circ, q_f = 10^\circ$$

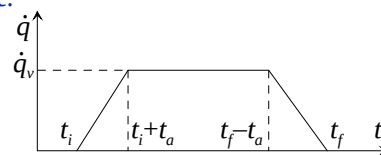


Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [15]

Profilo di velocità trapezoidale (P.v.t.)



Un procedimento di largo utilizzo nella pratica industriale per generare la traiettoria consiste nel pianificare un profilo di posizione lineare raccordato all'inizio e alla fine della traiettoria con tratti parabolici. Il **profilo di velocità** che ne risulta ha il tipico **andamento trapezoidale**.



La traiettoria è quindi divisa in **tre parti**:

1. Accelerazione costante, velocità a rampa, posizione a parabola;
2. Accelerazione nulla, velocità costante, posizione lineare;
3. Decelerazione costante, velocità a rampa, posizione a parabola.

Spesso la durata t_a della fase di accelerazione (fase 1) è posta uguale alla durata della fase di decelerazione (fase 3): si ottiene in questo modo una traiettoria simmetrica rispetto all'istante medio tra quelli iniziale e finale. Naturalmente deve essere $t_a \leq (t_f - t_i)/2$.

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [16]

P.v.t.: equazioni traiettoria



Fase di accelerazione

$$t \in [t_i, t_i + t_a]$$

$$\begin{cases} \ddot{q}(t) = \frac{\dot{q}_v}{t_a} \\ \dot{q}(t) = \frac{\dot{q}_v}{t_a} (t - t_i) \\ q(t) = q_i + \frac{\dot{q}_v}{2t_a} (t - t_i)^2 \end{cases}$$

Fase di velocità costante

$$t \in [t_i + t_a, t_f - t_a]$$

$$\begin{cases} \ddot{q}(t) = 0 \\ \dot{q}(t) = \dot{q}_v \\ q(t) = q_i + \dot{q}_v \left(t - t_i - \frac{t_a}{2} \right) \end{cases}$$

Fase di decelerazione

$$t \in [t_f - t_a, t_f]$$

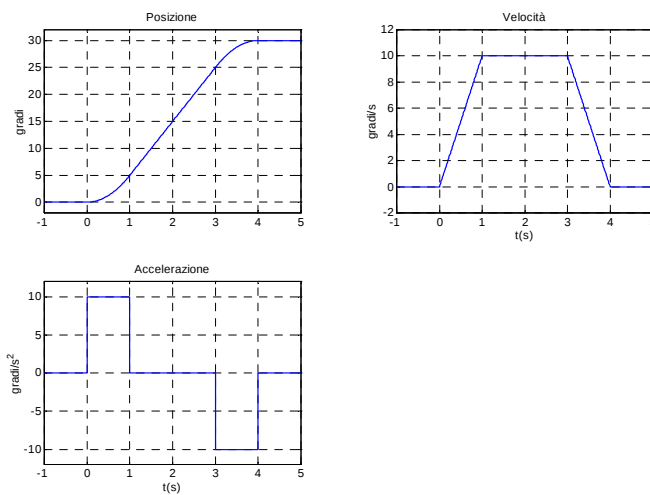
$$\begin{cases} \ddot{q}(t) = -\frac{\dot{q}_v}{t_a} \\ \dot{q}(t) = \frac{\dot{q}_v}{t_a} (t_f - t) \\ q(t) = q_f - \frac{\dot{q}_v}{2t_a} (t_f - t)^2 \end{cases}$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [17]

P.v.t.: esempio



$$t_i = 0, t_f = 4s, t_a = 1s, q_i = 0^\circ, q_f = 30^\circ, \dot{q}_v = 10^\circ/s$$



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [18]

P.v.t.: vincoli



Nella scelta dei parametri della traiettoria devono essere soddisfatti dei vincoli, che assicurano il posizionamento corretto nel tempo prestabilito. Si osservi che la velocità all'istante t_i+t_a si può ottenere come:

$$\ddot{q}t_a = \frac{q_m - q_a}{t_m - t_a} \quad \text{dove} \quad \begin{cases} q_a = q(t_i + t_a) \\ q_m = (q_i + q_f)/2 \\ t_m = (t_f - t_i)/2 \end{cases}$$

Inoltre risulta:

$$q_a = q_i + \frac{1}{2} \ddot{q}t_a^2$$

Eliminando dalle precedenti equazioni le variabili q_m , q_a , t_m , si ricava il vincolo:

$$\ddot{q}t_a^2 - \ddot{q}(t_f - t_i)t_a + (q_f - q_i) = 0$$

Un vincolo sulla velocità si ottiene dalla formula:

$$\dot{q}_v = \frac{q_f - q_i}{t_f - t_i - t_a}$$

P.v.t.: possibili specifiche



Dati:

- la distanza da percorrere $h = q_f - q_i$
- il tempo di percorrenza $T = t_f - t_i$

Se imponiamo $t_a \leq T/2$:

$$t_a \Rightarrow \begin{cases} \dot{q}_v = \frac{h}{T - t_a} \\ \ddot{q} = \frac{\dot{q}_v}{t_a} \end{cases}$$

Se imponiamo la velocità:

$$\dot{q}_v \Rightarrow \begin{cases} t_a = \frac{T\dot{q}_v - h}{\dot{q}_v} \\ \ddot{q} = \frac{\dot{q}_v}{t_a} \end{cases}$$

Se imponiamo l'accelerazione:

$$\ddot{q} \Rightarrow \begin{cases} t_a = \frac{\ddot{q}T - \sqrt{\ddot{q}^2 T^2 - 4\ddot{q}h}}{2\ddot{q}} \\ \dot{q}_v = \ddot{q}t_a \end{cases}$$

(deve essere: $|\ddot{q}| \geq \frac{4|h|}{T^2}$)

P.v.t.: accelerazione e velocità max



Se si vogliono imporre i **valori massimi di velocità ed accelerazione** consentiti dall'attuatore, si pone:

$$\begin{cases} t_a = \frac{\dot{q}_{\max}}{\ddot{q}_{\max}} & \text{tempo di accelerazione} \\ h = \dot{q}_{\max}(T - t_a) & \text{distanza percorsa} \end{cases}$$

La durata del moto (che non è imposta) vale quindi:

$$T = \frac{h}{\dot{q}_{\max}} + \frac{\dot{q}_{\max}}{\ddot{q}_{\max}}$$

da cui:

$$q(t) = \begin{cases} q_i + \frac{1}{2}\ddot{q}_{\max}(t - t_i)^2 & t_i \leq t < t_i + t_a \\ q_i + \dot{q}_{\max}t_a \left(t - t_i - \frac{t_a}{2} \right) & t_i + t_a \leq t < t_f - t_a \\ q_f - \frac{1}{2}\ddot{q}_{\max}(t_f - t)^2 & t_f - t_a \leq t \leq t_f \end{cases}$$

Attenzione: il tratto lineare esiste se e solo se: $h \geq \frac{\dot{q}_{\max}^2}{\ddot{q}_{\max}}$

P.v.t.: velocità max non raggiunta



Se: $h < \frac{\dot{q}_{\max}^2}{\ddot{q}_{\max}}$

si impone:
$$\begin{cases} t_a = \sqrt{\frac{h}{\ddot{q}_{\max}}} \\ T = 2t_a \end{cases}$$

da cui:

$$q(t) = \begin{cases} q_i + \frac{1}{2}\ddot{q}_{\max}(t - t_i)^2 & t_i \leq t < t_i + t_a \\ q_f - \frac{1}{2}\ddot{q}_{\max}(t_f - t)^2 & t_f - t_a \leq t \leq t_f \end{cases}$$

La velocità massima raggiunta in questo caso è quindi:

$$\dot{q}_v = \ddot{q}_{\max}t_a = \frac{h}{t_a} = 2\frac{h}{T}$$

P.v.t.: tempo di posizionamento



Imponendo la massima accelerazione e/o la massima velocità, nella pianificazione della traiettoria non si impone la durata del posizionamento. Esso risulta dalle seguenti formule.

1) Se non si raggiunge la velocità massima:

$$T = 2t_a = 2\sqrt{\frac{h}{\ddot{q}_{\max}}}$$

2) Se si raggiunge la velocità massima:

$$T = \frac{h}{\dot{q}_{\max}} + \frac{\dot{q}_{\max}}{\ddot{q}_{\max}}$$

P.v.t.: moto coordinato di più giunti



Se si devono coordinare più giunti con gli stessi vincoli sulle accelerazioni e velocità massime, per quello che deve effettuare il movimento maggiore si imposta il massimo valore di accelerazione $\ddot{q}_{\max}$ e si calcolano i valori t_a e T corrispondenti.

Per gli altri giunti si pianifica il movimento sulla base della durata T e del tempo di accelerazione t_a , secondo le formule viste precedentemente.

Limitazione del jerk

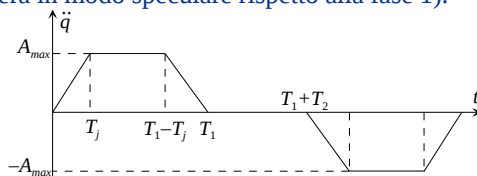


Nella traiettoria a profilo di velocità trapezoidale l'accelerazione ha un andamento discontinuo. Ne consegue che il jerk (derivata dell'accelerazione) assume valori infiniti. Questo provoca forti sollecitazioni sul sistema meccanico, che possono indurre vibrazioni.

Per ovviare a questo problema, si può modificare la pianificazione della traiettoria, dando continuità al profilo di accelerazione, che assume andamento trapezoidale nelle due fasi di accelerazione iniziale e decelerazione finale.

La traiettoria è quindi divisa in tre parti principali:

1. **Fase di accelerazione** (l'accelerazione viene portata linearmente al valore massimo, mantenuta al valore massimo e riportata linearmente a zero);
2. **Fase di crociera** (velocità costante);
3. **Fase di decelerazione** (si decelera in modo speculare rispetto alla fase 1).



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [25]

Messa in scala di traiettorie



Una volta pianificata una traiettoria è spesso necessario scalarla per soddisfare i vincoli del sistema di attuazione, che si manifestano in termini di saturazioni. Si parla in particolare di:

1. Messa in scala **cinematica**: si deve fare in modo che la traiettoria rispetti vincoli sulla massima velocità e sulla massima accelerazione;
2. Messa in scala **dinamica**: si deve fare in modo che agli attuatori del sistema meccanico non vengano richieste coppie superiori alle coppie massime erogabili.

Il problema della messa in scala cinematica ha rilevanza per quei profili di traiettoria (cubica, armonica, ecc.) per i quali tali valori non sono imposti nella pianificazione.

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [26]

Normalizzazione della traiettoria



Al fine di scalare la traiettoria, è opportuno esprimerla in **forma parametrica** in funzione di un parametro $\sigma=\sigma(t)$, opportunamente normalizzato.

Data la traiettoria $q(t)$, definita tra i punti q_i e q_f e di durata $T=t_f-t_i$, la sua espressione in **forma normalizzata** è la seguente:

$$q(t) = q_i + h\sigma(\tau)$$

con $h = q_f - q_i$ e:

$$0 \leq \sigma(\tau) \leq 1, \quad \tau = \frac{t-t_i}{T}, \quad 0 \leq \tau \leq 1$$

Ne consegue:

$$\begin{cases} \frac{dq(t)}{dt} = \frac{h}{T} \sigma'(\tau) \\ \frac{d^2q(t)}{dt^2} = \frac{h}{T^2} \sigma''(\tau) \\ \vdots \\ \frac{d^n q(t)}{dt^n} = \frac{h}{T^n} \sigma^{(n)}(\tau) \end{cases}$$

I valori massimi di velocità, accelerazione, ecc., si ottengono in corrispondenza dei valori massimi delle funzioni $\sigma^{(i)}(\tau)$: modificando la durata T della traiettoria è possibile soddisfare i vincoli sulle saturazioni cinematiche.

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [27]

Traiettoria polinomiale di grado 3



La traiettoria può essere parametrizzata con il polinomio:

$$\sigma(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3$$

Imponendo le condizioni al contorno $\sigma'(0)=0$, $\sigma'(1)=0$ (oltre a $\sigma(0)=0$, $\sigma(1)=1$):

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 3, \quad a_3 = -2$$

da cui:

$$\begin{aligned} \sigma(\tau) &= 3\tau^2 - 2\tau^3 & \sigma''(\tau) &= 6 - 12\tau \\ \sigma'(\tau) &= 6\tau - 6\tau^2 & \sigma'''(\tau) &= -12 \end{aligned}$$

I valori massimi di velocità ed accelerazione sono quindi:

$$\begin{aligned} \sigma'_{\max} = \sigma'(0.5) &= \frac{3}{2} \Rightarrow \dot{q}_{\max} = \frac{3h}{2T} \\ \sigma''_{\max} = \sigma''(0) &= 6 \Rightarrow \ddot{q}_{\max} = \frac{6h}{T^2} \end{aligned}$$

Operando sulla durata T della traiettoria possiamo rispettare i vincoli su velocità ed accelerazione.

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [28]

Traiettoria polinomiale di grado 5



La traiettoria può essere parametrizzata con il polinomio:

$$\sigma(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5$$

Imponendo le condizioni al contorno $\sigma(0)=0$, $\sigma(1)=1$, $\sigma'(0)=0$, $\sigma'(1)=0$, $\sigma''(0)=0$, $\sigma''(1)=0$:

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = 10, \quad a_4 = -15, \quad a_5 = 6$$

da cui:

$$\begin{aligned}\sigma(\tau) &= 10\tau^3 - 15\tau^4 + 6\tau^5 & \sigma''(\tau) &= 60\tau - 180\tau^2 + 120\tau^3 \\ \sigma'(\tau) &= 30\tau^2 - 60\tau^3 + 30\tau^4 & \sigma'''(\tau) &= 60 - 360\tau + 360\tau^2\end{aligned}$$

I valori massimi di velocità, accelerazione e jerk sono quindi:

$$\begin{aligned}\sigma'_{\max} = \sigma'(0.5) &= \frac{15}{8} & \Rightarrow \quad \dot{q}_{\max} &= \frac{15h}{8T} \\ \sigma''_{\max} = \sigma''(0.2123) &= \frac{10\sqrt{3}}{3} & \Rightarrow \quad \ddot{q}_{\max} &= \frac{10\sqrt{3}h}{3T^2} \\ \sigma'''_{\max} = \sigma'''(0) &= 60 & \Rightarrow \quad j_{\max} &= 60 \frac{h}{T^3}\end{aligned}$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [29]

Traiettoria armonica



La traiettoria può essere parametrizzata con la funzione:

$$\sigma(\tau) = \frac{1}{2}(1 - \cos \pi\tau)$$

da cui: $\sigma'(\tau) = \frac{\pi}{2} \sin \pi\tau$

$$\sigma''(\tau) = \frac{\pi^2}{2} \cos \pi\tau$$

$$\sigma'''(\tau) = \frac{\pi^3}{2} \sin \pi\tau$$

I valori massimi di velocità, accelerazione e jerk sono quindi:

$$\begin{aligned}\sigma'_{\max} = \sigma'(0.5) &= \frac{\pi}{2} & \Rightarrow \quad \dot{q}_{\max} &= \frac{\pi h}{2T} \\ \sigma''_{\max} = \sigma''(0) &= \frac{\pi^2}{2} & \Rightarrow \quad \ddot{q}_{\max} &= \frac{\pi^2 h}{2T^2} \\ \sigma'''_{\max} = \sigma'''(0.5) &= \frac{\pi^3}{2} & \Rightarrow \quad j_{\max} &= \frac{\pi^3 h}{2T^3}\end{aligned}$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [30]

Traiettoria cicloidale



La traiettoria può essere parametrizzata con la funzione:

$$\sigma(\tau) = \tau - \frac{1}{2\pi} \sin 2\pi\tau$$

da cui:

$$\sigma'(\tau) = 1 - \cos 2\pi\tau$$

$$\sigma''(\tau) = 2\pi \sin 2\pi\tau$$

$$\sigma'''(\tau) = 4\pi^2 \cos 2\pi\tau$$

I valori massimi di velocità, accelerazione e jerk sono quindi:

$$\sigma'_{\max} = \sigma'(0.5) = 2 \quad \Rightarrow \quad \dot{q}_{\max} = 2 \frac{h}{T}$$

$$\sigma''_{\max} = \sigma''(0.25) = 2\pi \quad \Rightarrow \quad \ddot{q}_{\max} = 2\pi \frac{h}{T^2}$$

$$\sigma'''_{\max} = \sigma'''(0) = 4\pi^2 \quad \Rightarrow \quad j_{\max} = 4\pi^2 \frac{h}{T^3}$$

Messa in scala cinematica: esempio



Si vuole realizzare una traiettoria con $q_i = 10^\circ$, $q_f = 50^\circ$, per un attuatore caratterizzato da: $\dot{q}_{\max} = 30$, $\ddot{q}_{\max} = 80$

Si hanno i seguenti risultati ($h = 40^\circ$):

Traiettoria	Formule	Vincoli	$T_{\min}$
Polin. grado 3	$\begin{cases} \dot{q}_{\max} = \frac{3h}{2T} \\ \ddot{q}_{\max} = \frac{6h}{T^2} \end{cases}$	$\begin{cases} T = \frac{3h}{60} = 2 \\ T = \sqrt{\frac{6h}{80}} = 1.732 \end{cases}$	2
Polin. grado 5	$\begin{cases} \dot{q}_{\max} = \frac{15h}{8T} \\ \ddot{q}_{\max} = \frac{10\sqrt{3}h}{3T^2} \end{cases}$	$\begin{cases} T = \frac{15h}{240} = 2.5 \\ T = \sqrt{\frac{10\sqrt{3}h}{240}} = 1.699 \end{cases}$	2.5
Armonica	$\begin{cases} \dot{q}_{\max} = \frac{\pi h}{2T} \\ \ddot{q}_{\max} = \frac{\pi^2 h}{2T^2} \end{cases}$	$\begin{cases} T = \frac{\pi h}{60} = 2.094 \\ T = \sqrt{\frac{\pi^2 h}{160}} = 1.571 \end{cases}$	2.094
Cicloidale	$\begin{cases} \dot{q}_{\max} = 2 \frac{h}{T} \\ \ddot{q}_{\max} = 2\pi \frac{h}{T^2} \end{cases}$	$\begin{cases} T = \frac{2h}{30} = 2.667 \\ T = \sqrt{\frac{2\pi h}{80}} = 1.772 \end{cases}$	2.667

Messa in scala dinamica



Discuteremo la tecnica della scalatura dinamica della traiettoria facendo direttamente riferimento al modello dinamico del manipolatore robotico, la cui equazione (in assenza di attrito e di forze di interazione con l'ambiente) è la seguente:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau$$

Per ogni giunto vale quindi un'equazione del tipo:

$$B_i^T(q)\ddot{q} + \frac{1}{2}\dot{q}^T C_i(q)\dot{q} + g_i(q) = \tau_i \quad i = 1, \dots, n$$

con $C_i(q)$ opportuna matrice.

Consideriamo una parametrizzazione della traiettoria:

$$q = q(\sigma), \quad \sigma = \sigma(t)$$

da cui:

$$\dot{q} = \frac{dq}{d\sigma} \dot{\sigma}, \quad \ddot{q} = \frac{d^2q}{d\sigma^2} \dot{\sigma}^2 + \frac{dq}{d\sigma} \ddot{\sigma}$$

Messa in scala dinamica



Sostituendo nel modello dinamico relativo al singolo giunto si ha:

$$\left[B_i^T(q(\sigma)) \frac{dq}{d\sigma} \right] \ddot{\sigma} + \left[B_i^T(q(\sigma)) \frac{d^2q}{d\sigma^2} + \frac{1}{2} \frac{dq^T}{d\sigma} C_i(q(\sigma)) \frac{dq}{d\sigma} \right] \dot{\sigma}^2 + g_i(q(\sigma)) = \tau_i$$

ovvero un'equazione nella forma:

$$\alpha_i(\sigma)\ddot{\sigma} + \beta_i(\sigma)\dot{\sigma}^2 + \gamma_i(\sigma) = \tau_i$$

Si osservi che γ_i dipende solo dalla posizione (non dalla velocità).

Le coppie necessarie per eseguire il moto richiesto sono quindi:

$$\tau_i(t) = \alpha_i(\sigma(t))\ddot{\sigma}(t) + \beta_i(\sigma(t))\dot{\sigma}^2(t) + \gamma_i(\sigma(t)), \quad i = 1, \dots, n, \quad t \in [0, T]$$

Messa in scala dinamica



Per ottenere una diversa parametrizzazione della traiettoria consideriamo ora una scalatura dell'asse dei tempi, per esempio lineare:

$$t \rightarrow \theta = kt \quad \theta \in [0, kT]$$

Si ottiene:

$$\sigma(t) = \hat{\sigma}(\theta), \quad \dot{\sigma}(t) = k\hat{\sigma}'(\theta), \quad \ddot{\sigma}(t) = k^2\hat{\sigma}''(\theta)$$

dove l'apice indica la derivata rispetto a θ .

Se $k > 1$ la traiettoria è più lenta.

Se $k < 1$ la traiettoria è più veloce.

Messa in scala dinamica



$$\tau_i(t) = \alpha_i(\sigma(t))\ddot{\sigma}(t) + \beta_i(\sigma(t))\dot{\sigma}^2(t) + \gamma_i(\sigma(t)), \quad i = 1, \dots, n, \quad t \in [0, T]$$

Con la nuova parametrizzazione le coppie diventano:

$$\begin{aligned} \tau_i(\theta) &= \alpha_i(\hat{\sigma}(\theta))\hat{\sigma}''(\theta) + \beta_i(\hat{\sigma}(\theta))\hat{\sigma}'^2(\theta) + \gamma_i(\hat{\sigma}(\theta)) = \\ &= \alpha_i(\sigma(t))\frac{\ddot{\sigma}(t)}{k^2} + \beta_i(\sigma(t))\frac{\dot{\sigma}^2(t)}{k^2} + \gamma_i(\sigma(t)) = \\ &= \frac{1}{k^2} [\alpha_i(\sigma(t))\ddot{\sigma}(t) + \beta_i(\sigma(t))\dot{\sigma}^2(t)] + \gamma_i(\sigma(t)) = \\ &= \frac{1}{k^2} [\tau_i(t) - g_i(\sigma(t))] + g_i(\sigma(t)) \end{aligned}$$

Pertanto:

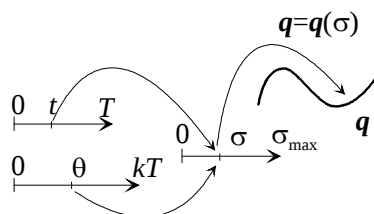
$$\tau_i(\theta) - g_i(\theta) = \frac{1}{k^2} [\tau_i(t) - g_i(t)]$$

Messa in scala dinamica



$$\tau_i(\theta) - g_i(\theta) = \frac{1}{k^2} [\tau_i(t) - g_i(t)]$$

- Riparametrizzando la traiettoria non è necessario ricalcolare la dinamica del sistema;
- Le nuove coppie si ottengono, senza considerare il termine gravitazionale (che non dipende dalla parametrizzazione), moltiplicando per il fattore $1/k^2$ le coppie ottenute con la traiettoria originale;
- La nuova traiettoria ha durata kT .

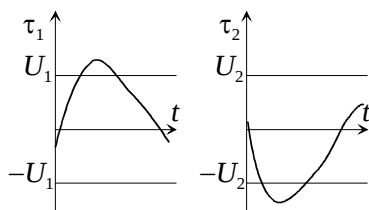


Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [37]

Messa in scala dinamica



Si abbia un manipolatore a due gradi di libertà sottoposto ad una traiettoria che genera le seguenti coppie:



Per mettere in scala la traiettoria si calcolerà il valore:

$$k^2 = \max \left\{ 1, \frac{|\tau_1|}{U_1}, \frac{|\tau_2|}{U_2} \right\} \geq 1$$

Le nuove coppie saranno realizzabili ($\tau(\theta) = \tau(t)/k^2$) ed almeno una saturerà in un punto.

Mettere in scala una traiettoria per evitare che la coppia ecceda il valore massimo in un dato tratto può rallentare eccessivamente l'esecuzione: si ricorre allora ad una messa in scala variabile (cioè applicata solo nei tratti in cui si ha saturazione di coppia).

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [38]



Parte II

Moto su percorso assegnato nello spazio dei giunti

Interpolazione mediante polinomi



Il problema di determinare una traiettoria che passi per n punti può essere risolto in modo univoco adottando una funzione polinomiale di grado $n-1$, del tipo:

$$q(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_{n-1} t^{n-1}$$

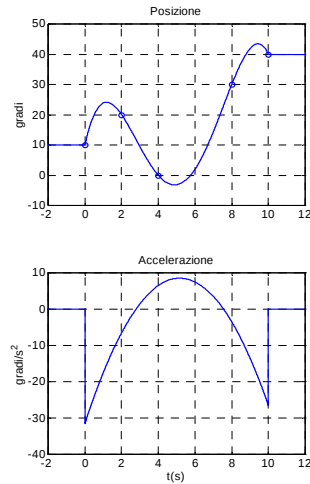
Dati i valori $t_i, q_i, i=1, \dots, n$ si costruiscono i vettori $\mathbf{q}, \mathbf{a}$ e la matrice $\mathbf{T}$ (di Vandermonde) come:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{n-1} \\ q_n \end{bmatrix} = \begin{bmatrix} 1 & t_1 & \dots & t_1^{n-1} \\ 1 & t_2 & \dots & t_2^{n-1} \\ & \vdots & & \vdots \\ 1 & t_{n-1} & \dots & t_{n-1}^{n-1} \\ 1 & t_n & \dots & t_n^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-2} \\ a_{n-1} \end{bmatrix} = \mathbf{T} \mathbf{a}$$

Ne consegue:

$$\mathbf{a} = \mathbf{T}^{-1} \mathbf{q} \quad (\text{la matrice } \mathbf{T} \text{ è sempre invertibile se } t_i > t_{i-1}, i=1, \dots, n)$$

Interpolazione mediante polinomi: esempio



$$\begin{array}{ccccc} t_1 = 0 & t_2 = 2 & t_3 = 4 & t_4 = 8 & t_5 = 10 \\ q_1 = 10^\circ & q_2 = 20^\circ & q_3 = 0^\circ & q_4 = 30^\circ & q_5 = 40^\circ \end{array}$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [41]

Interpolazione mediante polinomi



Un vantaggio evidente dell'interpolazione polinomiale è che la funzione $q(t)$ ha derivate continue di ordine qualsiasi all'interno dell'intervallo $[t_1, t_n]$. Tuttavia il metodo non è efficiente dal punto di vista numerico: all'aumentare del numero n di punti aumenta il numero condizionante k (rapporto tra il massimo ed il minimo valor singolare) della matrice T di Vandermonde, rendendo il problema della sua inversione mal condizionato numericamente.

Se, per esempio, $t_i = i/n$, $i=1, \dots, n$:

n	3	4	5	6	10	15	20
k	15.1	98.87	686.43	4924.37	$1.519 \cdot 10^7$	$4.032 \cdot 10^{11}$	$1.139 \cdot 10^{16}$

Esistono anche altri metodi, più efficienti, per calcolare i coefficienti del polinomio, ma le difficoltà numeriche permangono per valori elevati di n .

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [42]

Interpolazione mediante polinomi



Anche prescindendo dalle difficoltà numeriche, l'interpolazione di n punti mediante un unico polinomio di grado $n-1$ presenta degli svantaggi:

1. il grado del polinomio dipende da n e, per elevati valori di n , la quantità di calcoli da eseguire può essere notevole;
2. la variazione di un solo punto (t_i, q_i) implica il ricalcolo dell'intero polinomio;
3. l'aggiunta di un punto finale (t_{n+1}, q_{n+1}) implica l'utilizzo di un polinomio di grado maggiore ed il ricalcolo di tutti i coefficienti
4. la soluzione che si ottiene presenta in generale oscillazioni indesiderate

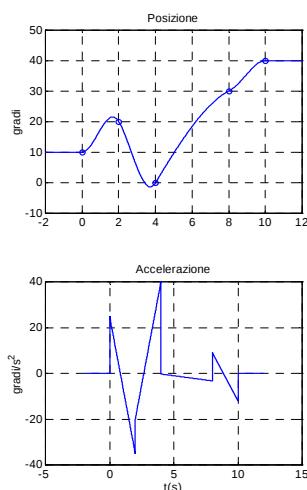
Un'alternativa è, anziché considerare un unico polinomio di grado $n-1$, utilizzare $n-1$ polinomi di grado p (tipicamente inferiore), ognuno dei quali definito in un tratto della traiettoria.

Il grado p dei polinomi è normalmente preso uguale a 3 (tratti di traiettoria **cubica**).

Una primo, ovvio, modo di procedere consiste nell'assegnare posizioni e velocità in tutti i punti e calcolare i coefficienti delle cubiche tra due punti consecutivi.

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [43]

Interpolazione mediante cubiche



$t_1 = 0$	$t_2 = 2$	$t_3 = 4$	$t_4 = 8$	$t_5 = 10$
$q_1 = 10^\circ$	$q_2 = 20^\circ$	$q_3 = 0^\circ$	$q_4 = 30^\circ$	$q_5 = 40^\circ$
$\dot{q}_1 = 0^\circ/s$	$\dot{q}_2 = -10^\circ/s$	$\dot{q}_3 = 10^\circ/s$	$\dot{q}_4 = 3^\circ/s$	$\dot{q}_5 = 0^\circ/s$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [44]

Interpolazione mediante cubiche



Se vengono solo specificati i punti di passaggio senza specificare le velocità intermedie, queste si possono calcolare approssimativamente con regole del tipo:

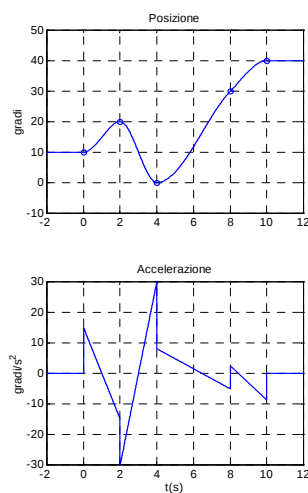
$$\begin{aligned} \dot{q}_1 &= 0 \\ \dot{q}_k &= \begin{cases} 0 & \text{sign}(v_k) \neq \text{sign}(v_{k+1}) \\ \frac{v_k + v_{k+1}}{2} & \text{sign}(v_k) = \text{sign}(v_{k+1}) \end{cases} \\ \dot{q}_n &= 0 \end{aligned}$$

essendo:

$$v_k = \frac{q_k - q_{k-1}}{t_k - t_{k-1}}$$

la pendenza (rapporto incrementale) nel tratto $[t_{k-1}, t_k]$.

Interpolazione mediante cubiche



$$\begin{array}{ccccc} t_1 = 0 & t_2 = 2 & t_3 = 4 & t_4 = 8 & t_5 = 10 \\ q_1 = 10^\circ & q_2 = 20^\circ & q_3 = 0^\circ & q_4 = 30^\circ & q_5 = 40^\circ \end{array}$$

Spline



L'interpolazione mediante cubiche eseguita come visto precedentemente genera una traiettoria che presenta accelerazione discontinua nei punti di passaggio. Per ovviare a questo problema, sempre mantenendo interpolanti cubiche, si deve rinunciare ad imporre specifici valori di velocità nei punti intermedi, limitandosi ad imporre la continuità in due tratti contigui di posizione, velocità ed accelerazione.

La traiettoria che si ottiene con questo procedimento prende il nome di **spline** (smooth path line).

Si può dimostrare che la spline è la funzione interpolante **a curvatura minima**, a parità di condizioni di continuità sulle derivate.

Spline: condizioni da imporre



Poiché con n punti si hanno $n-1$ polinomi del tipo:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

ciascuno dei quali ha 4 coefficienti, il numero totale di coefficienti da calcolare è $4(n-1)$. Le condizioni da imporre sono:

- $2(n-1)$ condizioni di passaggio per punti (ogni cubica deve interpolare i punti alle sue estremità);
- $n-2$ condizioni sulla continuità delle velocità nei punti intermedi
- $n-2$ condizioni sulla continuità delle accelerazioni nei punti intermedi

Si hanno quindi:

$$4(n-1) - 2(n-1) - 2(n-2) = 2$$

gradi di libertà residui.

Una modalità (non unica) di utilizzare questi 2 gradi di libertà consiste nell'assegnare opportune condizioni iniziali e finali sulla velocità.

Spline: posizione analitica del problema



Si desidera determinare una funzione:

$$q(t) = \{q_k(t), \quad t \in [t_k, t_{k+1}], \quad k = 1, \dots, n-1\}$$
$$q_k(\tau) = a_{k0} + a_{k1}\tau + a_{k2}\tau^2 + a_{k3}\tau^3, \quad \tau \in [0, T_k] \quad (\tau = t - t_k, \quad T_k = t_{k+1} - t_k)$$

con le condizioni:

$$\begin{aligned} q_k(0) &= q_k, \quad q_k(T_k) = q_{k+1} & k &= 1, \dots, n-1 \\ \dot{q}_k(T_k) &= \dot{q}_{k+1}(0) = v_{k+1} & k &= 1, \dots, n-2 \\ \ddot{q}_k(T_k) &= \ddot{q}_{k+1}(0) & k &= 1, \dots, n-2 \\ \dot{q}_1(0) &= v_1, \quad \dot{q}_{n-1}(T_{n-1}) = v_n \end{aligned}$$

dove le quantità v_k , $k=2, \dots, n-1$ non sono specificate.

Il problema consiste nel ricavare i coefficienti a_{ki} .

Spline: algoritmo



Si assumano inizialmente note le velocità v_k , $k=2, \dots, n-1$ nei punti intermedi. In questo modo, per ogni polinomio cubico si hanno quattro condizioni al contorno su posizione e velocità, che danno origine al sistema:

$$\begin{cases} q_k(0) = a_{k0} = q_k \\ \dot{q}_k(0) = a_{k1} = v_k \\ q_k(T_k) = a_{k0} + a_{k1}T_k + a_{k2}T_k^2 + a_{k3}T_k^3 = q_{k+1} \\ \dot{q}_k(T_k) = a_{k1} + 2a_{k2}T_k + 3a_{k3}T_k^2 = v_{k+1} \end{cases}$$

che risolto dà:

$$\begin{cases} a_{k0} = q_k \\ a_{k1} = v_k \\ a_{k2} = \frac{1}{T_k} \left[\frac{3(q_{k+1} - q_k)}{T_k} - 2v_k - v_{k+1} \right] \\ a_{k3} = \frac{1}{T_k^2} \left[\frac{2(q_k - q_{k+1})}{T_k} + v_k + v_{k+1} \right] \end{cases}$$

Spline: algoritmo



Naturalmente le velocità v_k , $k=2, \dots, n-1$ vanno calcolate. Imponiamo la continuità dell'accelerazione nei punti intermedi:

$$\ddot{q}_k(T_k) = 2a_{k2} + 6a_{k3}T_k = 2a_{k+1,2} = \ddot{q}_{k+1}(0), \quad k = 1, \dots, n-2$$

Sostituendo le espressioni per i termini a_{k2} , a_{k3} , $a_{k+1,2}$ e moltiplicando per $(T_k T_{k+1})/2$ si ottiene:

$$T_{k+1}v_k + 2(T_{k+1} + T_k)v_{k+1} + T_k v_{k+2} = \frac{3}{T_k T_{k+1}} [T_k^2(q_{k+2} - q_{k+1}) + T_{k+1}^2(q_{k+1} - q_k)]$$

In forma matriciale:

$$\begin{bmatrix} T_2 & 2(T_1 + T_2) & T_1 & & & \\ 0 & T_3 & 2(T_2 + T_3) & T_2 & & \\ & & & & \ddots & \\ & & & T_{n-2} & 2(T_{n-3} + T_{n-2}) & T_{n-3} \\ & & & & T_{n-1} & 2(T_{n-2} + T_{n-1}) \\ & & & & & T_{n-2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-3} \\ c_{n-2} \end{bmatrix}$$

dove le costanti c_k dipendono solo dalle posizioni intermedie e dalle durate dei segmenti, grandezze note.

Spline: algoritmo



Poiché le velocità v_1 e v_n sono note (specificate come dati iniziali), eliminando le relative colonne si ha:

$$\begin{bmatrix} 2(T_1 + T_2) & T_1 & & & & \\ T_3 & 2(T_2 + T_3) & T_2 & & & \\ & & & \ddots & & \\ & & & T_{n-2} & 2(T_{n-3} + T_{n-2}) & T_{n-3} \\ & & & & T_{n-1} & 2(T_{n-2} + T_{n-1}) \end{bmatrix} \begin{bmatrix} v_2 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{3}{T_1 T_2} [T_1^2(q_3 - q_2) + T_2^2(q_2 - q_1)] - T_2 v_1 \\ \frac{3}{T_2 T_3} [T_2^2(q_4 - q_3) + T_3^2(q_3 - q_2)] \\ \vdots \\ \frac{3}{T_{n-3} T_{n-2}} [T_{n-3}^2(q_{n-1} - q_{n-2}) + T_{n-2}^2(q_{n-2} - q_{n-3})] \\ \frac{3}{T_{n-2} T_{n-1}} [T_{n-2}^2(q_n - q_{n-1}) + T_{n-1}^2(q_{n-1} - q_{n-2})] - T_{n-2} v_n \end{bmatrix}$$

ovvero un'equazione del tipo $Av=c$

Spline: algoritmo



- La matrice A è a struttura dominante diagonale e risulta sempre invertibile per $T_k > 0$.
- Inoltre la matrice A è a struttura tridiagonale, per cui esistono tecniche numeriche efficienti (metodo di Gauss-Jordan) per la sua inversione.
- Una volta nota l'inversa di A si possono calcolare le velocità $v_2, \dots, v_{n-1}$ come:

$$\mathbf{v} = \mathbf{A}^{-1} \mathbf{c}$$

il che risolve completamente il problema.

È possibile anche determinare le spline con un algoritmo alternativo (ma del tutto equivalente) che ricava invece delle velocità le accelerazioni nei punti intermedi.

Spline: tempo di percorrenza



Il tempo totale di percorrenza della spline è dato da:

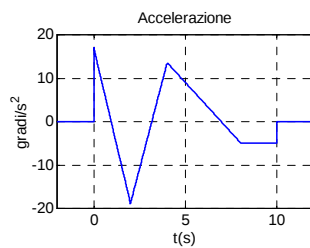
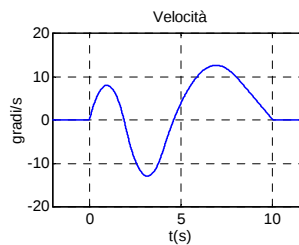
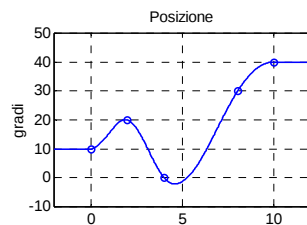
$$T = \sum_{k=1}^{n-1} T_k = t_n - t_1$$

È possibile impostare un problema di ottimo che minimizza il tempo totale di percorrenza. Il problema si pone nei termini di determinare i valori T_k in modo da minimizzare T , con i vincoli sulle massime velocità ed accelerazioni di giunto. Formalmente:

$$\begin{cases} \min_{T_k} T = \sum_{k=1}^{n-1} T_k \\ \text{tale che} & \begin{cases} |\dot{q}(\tau, T_k)| < v_{\max} & \tau \in [0, T] \\ |\ddot{q}(\tau, T_k)| < a_{\max} & \tau \in [0, T] \end{cases} \end{cases}$$

Si tratta quindi di un problema di ottimo non lineare con funzione obiettivo lineare, risolvibile con le tecniche della ricerca operativa.

Spline: esempio



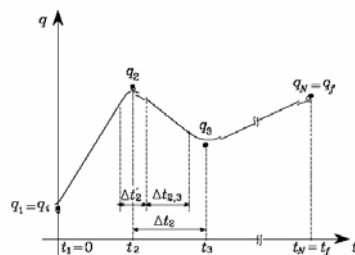
$$\begin{array}{cccccc} t_1 = 0 & t_2 = 2 & t_3 = 4 & t_4 = 8 & t_5 = 10 \\ q_1 = 10^\circ & q_2 = 20^\circ & q_3 = 0^\circ & q_4 = 30^\circ & q_5 = 40^\circ \end{array}$$

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [55]

Interpolazione con tratti rettilinei



Un modo alternativo, molto semplice, per gestire il problema dell'interpolazione di punti consiste nel collegare i punti di percorso con funzioni lineari. Per evitare discontinuità di velocità, i tratti lineari possono essere raccordati da tratti parabolici nell'intorno dei punti di percorso.



La traiettoria risultante $q(t)$ non tocca nessuno dei punti di percorso, anche se vi passa vicino. In questo caso i punti di percorso prendono il nome di **punti di via** (via points).

Il disegno è tratto dal testo:
L.Sciavicco, B.Siciliano
Robotica industriale – Modellistica e controllo di robot
manipolatori (2a ed.)
Mc Graw-Hill, 2000

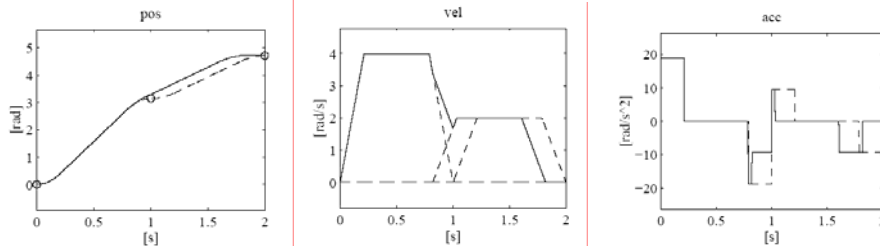
Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [56]

P.v.t. ed interpolazione di punti



Se invece si volessero utilizzare in sequenza tratti di traiettoria con profilo di velocità trapezoidale per interpolare punti intermedi si otterrebbe un movimento che passa per i punti intermedi a velocità nulla (cioè fermandosi).

Un modo per ovviare al problema consiste nell'iniziare la pianificazione di un tratto di traiettoria in anticipo rispetto alla conclusione del tratto precedente:



Il disegno è tratto dal testo:
L.Sciavicco, B.Siciliano
Robotica industriale – Modellistica e controllo di robot
manipolatori (2a ed.)
Mc Graw-Hill, 2000

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [57]

Pianificazione di traiettorie



Parte III

Traiettorie nello spazio operativo

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [58]

Traiettorie nello spazio operativo



La pianificazione della traiettoria nello spazio dei giunti comporta movimenti imprevedibili dell'organo terminale. Quando si desidera che il moto si sviluppi lungo un percorso definito nello spazio operativo, è necessario pianificare direttamente la traiettoria in questo spazio.

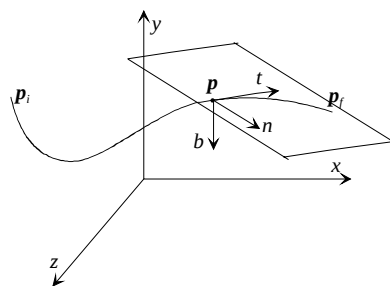
In generale il percorso può essere caratterizzato dando N punti (di via) $\mathbf{x}(t_k)$, $k=1, \dots, N$. Le componenti $x_i(t_k)$ possono essere singolarmente raccordate con una delle tecniche viste precedentemente.

Se però il moto dell'organo terminale deve rispettare una traiettoria con precisione, la traiettoria stessa deve poter essere espressa analiticamente.

Primitive di percorso



Consideriamo una rappresentazione parametrica di una curva nello spazio. La parametrizzazione può essere effettuata rispetto all'*ascissa curvilinea* (lunghezza dell'arco di traiettoria): $\mathbf{p} = \mathbf{p}(s)$



Possiamo definire i versori tangente, normale e binormale

$$\begin{aligned} \mathbf{t} &= \frac{d\mathbf{p}(s)}{ds} \\ \mathbf{n} &= \frac{d^2\mathbf{p}(s)/ds^2}{\|d^2\mathbf{p}(s)/ds^2\|} \\ \mathbf{b} &= \mathbf{t} \times \mathbf{n} \end{aligned}$$

Segmento nello spazio



Come esempio di primitiva di percorso possiamo considerare un segmento nello spazio:

$$\mathbf{p}(s) = \mathbf{p}_i + \frac{s}{\|\mathbf{p}_f - \mathbf{p}_i\|} (\mathbf{p}_f - \mathbf{p}_i)$$

$$\frac{d\mathbf{p}}{ds} = \frac{1}{\|\mathbf{p}_f - \mathbf{p}_i\|} (\mathbf{p}_f - \mathbf{p}_i)$$

$$\frac{d^2\mathbf{p}}{ds^2} = \mathbf{0}$$

In questo caso non è possibile definire la terna $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ in modo univoco.

Arco di circonferenza



Una rappresentazione parametrica di una circonferenza di raggio ρ giacente in un piano $x'y'$ ed avente centro nell'origine di tale piano è la seguente:

$$\mathbf{p}'(s) = \begin{bmatrix} \rho \cos(s/\rho) \\ \rho \sin(s/\rho) \\ 0 \end{bmatrix}$$

Detti:

- $\mathbf{c}$ il vettore che identifica il centro della circonferenza nella terna base
- $\mathbf{R}$ la matrice di rotazione dalla terna base alla terna $x'y'z'$

la rappresentazione parametrica di un generico arco di circonferenza nello spazio è:

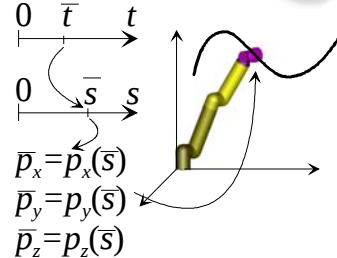
$$\mathbf{p}(s) = \mathbf{c} + \mathbf{R}\mathbf{p}'(s)$$
$$\frac{d\mathbf{p}}{ds} = \mathbf{R} \begin{bmatrix} -\sin(s/\rho) \\ \cos(s/\rho) \\ 0 \end{bmatrix} \quad \frac{d^2\mathbf{p}}{ds^2} = \mathbf{R} \begin{bmatrix} -\cos(s/\rho)/\rho \\ -\sin(s/\rho)/\rho \\ 0 \end{bmatrix}$$

Traiettorie di posizione



Per la pianificazione della posizione, tenendo conto della parametrizzazione rispetto all'ascissa curvilinea $\mathbf{p} = \mathbf{p}(s)$, si può attribuire la legge oraria alla variabile $s(t)$.

Per la determinazione di $s(t)$ vale quanto detto precedentemente circa la pianificazione della traiettoria per variabili scalari.



$\dot{\mathbf{p}} = \dot{s} \frac{d\mathbf{p}}{ds} = \dot{s} \mathbf{t}$ $\dot{s}$ è quindi il modulo, con segno positivo o negativo, della velocità

Per il segmento:

$$\dot{\mathbf{p}} = \frac{\dot{s}}{\|\mathbf{p}_f - \mathbf{p}_i\|} (\mathbf{p}_f - \mathbf{p}_i) = \dot{s} \mathbf{t}$$

$$\ddot{\mathbf{p}} = \frac{\ddot{s}}{\|\mathbf{p}_f - \mathbf{p}_i\|} (\mathbf{p}_f - \mathbf{p}_i) = \ddot{s} \mathbf{t}$$

Traiettorie di orientamento



Per la pianificazione dell'orientamento si potrebbe pensare di interpolare (per esempio linearmente) le componenti dei versori $\mathbf{n}(t)$, $\mathbf{s}(t)$ e $\mathbf{a}(t)$.

Si tratta però di un procedimento sconsigliabile, perché non si può garantire ad ogni istante l'ortonormalità dei versori.

Un'alternativa consiste nell'interpolare una terna di angoli di Eulero, utilizzando le relazioni:

$$\phi(s) = \phi_i + \frac{s}{\|\phi_f - \phi_i\|} (\phi_f - \phi_i) \quad \text{Per la legge oraria } s(t) \text{ si usano profili polinomiali o a velocità trapezoidale.}$$

$$\dot{\phi} = \frac{\dot{s}}{\|\phi_f - \phi_i\|} (\phi_f - \phi_i)$$

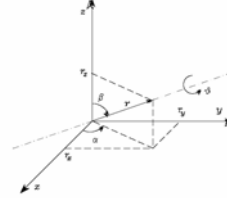
La velocità angolare ω , legata linearmente a $\dot{\phi}$, ha una variazione continua del modulo.

$$\ddot{\phi} = \frac{\ddot{s}}{\|\phi_f - \phi_i\|} (\phi_f - \phi_i)$$

Traiettorie di orientamento



L'orientamento può anche essere pianificato ricorrendo alla rappresentazione asse/angolo: assegnate due terne con origini coincidenti e orientamenti differenti, è sempre possibile determinare un vettore $\mathbf{r}$ tale che la seconda sia ottenibile dalla prima tramite una rotazione di un angolo ϑ_f intorno all'asse di tale vettore.



Siano $\mathbf{R}_i$ e $\mathbf{R}_f$ le matrici di rotazione, rispetto alla terna base, della terna iniziale e della terna finale.

La matrice di rotazione tra le due terne, con la relativa rappresentazione in termini di asse/angolo, è quindi:

$$\mathbf{R}_f^i = \mathbf{R}_i^T \mathbf{R}_f = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \vartheta_f = \cos^{-1} \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right)$$

$$\mathbf{r} = \frac{1}{2 \sin \vartheta_f} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Traiettorie di orientamento



Definiamo $\mathbf{R}^i(t)$ la matrice che descrive la transizione tra $\mathbf{R}_i$ e $\mathbf{R}_f$. Risulta:

$$\mathbf{R}^i(0) = \mathbf{I}, \quad \mathbf{R}^i(t_f) = \mathbf{R}_f^i$$

La matrice $\mathbf{R}^i(t)$ può essere interpretata come $\mathbf{R}^i(\vartheta(t), \mathbf{r})$, dove:

- $\mathbf{r}$ è costante e può essere calcolato con la formula vista prima dagli elementi di $\mathbf{R}_f^i$
- $\vartheta(t)$ può essere fatto variare con un'opportuna legge oraria, con $\vartheta(0)=0$, $\vartheta(t_f)=\vartheta_f$

Per caratterizzare l'orientamento in terna base è poi sufficiente calcolare:

$$\mathbf{R}(t) = \mathbf{R}_i \mathbf{R}^i(t)$$



Parte IV

Programmazione del moto

Programmazione per insegnamento



Una prima modalità di programmazione del moto per un robot è la programmazione cosiddetta **teaching-by-showing**.

L'operatore con il teach pendant muove il manipolatore lungo il percorso desiderato. I trasduttori di posizione memorizzano le posizioni che il robot deve raggiungere, che saranno poi raccordate da un software di generazione della traiettoria. Il robot sarà quindi in grado di ripetere autonomamente il movimento insegnato con apprendimento sul campo.



COMAU SpA

Non vengono richieste particolari capacità di programmazione all'operatore, che tipicamente è un tecnico d'officina.

D'altra parte il metodo presenta limitazioni, in quanto la messa a punto del programma richiede che l'operatore abbia a disposizione il robot (che quindi non è operativo), non è prevista la possibilità di eseguire funzioni di condizionamento logico o cicli di attesa ed in generale non si possono programmare attività complesse.

Ambienti di programmazione



La programmazione di un sistema robotico richiede quindi la presenza di un **ambiente di programmazione**, corredato da opportuni linguaggi, attraverso il quale l'operatore possa impartire i comandi alla macchina per mezzo di programmi appositamente scritti.

L'ambiente di programmazione è parte della interfaccia uomo-macchina (MMI), cui è affidato anche il compito di verificare la corretta esecuzione dei compiti da parte del robot e comunicare con l'operatore, in particolare in presenza di anomalie o errori.

Un **linguaggio di programmazione** è un linguaggio ad alto livello (simile al PASCAL) con il quale l'operatore può programmare il movimento del robot ed operazioni complesse in cui il robot, inserito in una cella di lavoro, interagisce con altre macchine e dispositivi. Rispetto ad un linguaggio di programmazione general purpose, il linguaggio prevede specifiche funzionalità orientate al robot.

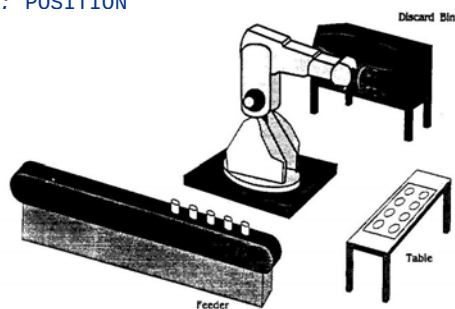
Nel seguito si farà cenno al linguaggio di programmazione **PDL2** della **COMAU**.

Esempio di programma



Il seguente programma trasferisce i pezzi da un trasportatore ad una tavola di lavoro o ad un contenitore di scarto, sulla base di segnali digitali in ingresso:

```
PROGRAM pezzo
VAR
  riposo, trasp, tavola, scarto: POSITION
BEGIN CYCLE
  MOVE TO riposo
  OPEN HAND 1
  WAIT FOR $DIN[1]=ON
  MOVE TO trasp
  CLOSE HAND 1
  IF $DIN[2] = OFF THEN
    MOVE TO tavola
  ELSE
    MOVE TO scarico
  ENDIF
  OPEN HAND
END pezzo
```



I disegni sono tratti dal manuale:
Linguaggio di programmazione PDL2
Versione 3.0
COMAU S.p.A.

Classi di dati



Oltre alle abituali classi di dati di un qualsiasi linguaggio di programmazione (integer, real, boolean, string, array), nel PDL2 sono definite alcune classi specifiche per applicazioni robotiche. Tra queste:

VECTOR: rappresentazione di un vettore mediante le 3 componenti

POSITION: tre componenti di posizione cartesiana, tre componenti di orientamento (angoli di Eulero) ed una stringa di configurazione (che indica se la configurazione è di spalla/gomito/polso alto o basso)

JOINTPOS: posizioni dei giunti del braccio, misurate in gradi

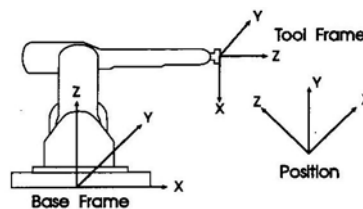
PATH: sequenza di nodi (tratti di movimento) che devono essere interpretati in successione.

È inoltre possibile definire **semafori**, utili per la sincronizzazione nel caso in cui vi siano programmi concorrenti che condividono le stesse risorse.

Sistemi di riferimento



Nel PDL2 per ogni manipolatore viene predefinita una terna di riferimento universale. L'operatore può ridefinire la terna di riferimento di base (**\$BASE**) relativamente alla terna universale. Questo è utile qualora si debba riposizionare il robot nell'area di lavoro, perché evita di ricalcolare tutte le posizioni. Inoltre il programmatore può definire una terna (**\$TOOL**) relativamente alla terna utensile del manipolatore, utile quando viene cambiato l'attrezzo montato sulla flangia.



L'istruzione MOVE



Con l'istruzione **MOVE** vengono impartiti i comandi per la movimentazione dei bracci. La sintassi dell'istruzione è la seguente:

`MOVE <ARM[n]> <trajectory> dest_clause <opt_clauses> <and_clause>`

(si osservi che un unico controllore può gestire più bracci (ARM)).

La clausola **trajectory** può assumere uno dei seguenti valori:

LINEAR	(movimento lineare nello spazio cartesiano)
CIRCULAR	(movimento circolare nello spazio cartesiano)
JOINT	(movimento nello spazio dei giunti)

Di default viene eseguito un movimento nello spazio dei giunti.

MOVE: clauseole di destinazione



Esistono varie clauseole di destinazione per l'istruzione MOVE. Le principali sono:

MOVE TO

Sposta il braccio verso la destinazione specificata, che può essere una variabile di tipo POSITION o JOINTPOS. Per esempio:

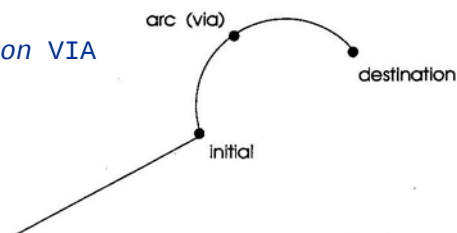
`MOVE LINEAR TO POS(x,y,z,e1,e2,e3,config)`

`MOVE TO home`

La clausola opzionale **VIA** può essere utilizzata per specificare un punto in cui il braccio dovrà effettuare un passaggio al volo. Per esempio:

`MOVE TO initial`

`MOVE CIRCULAR TO destination VIA
arc`



MOVE: clausole di destinazione



MOVE NEAR

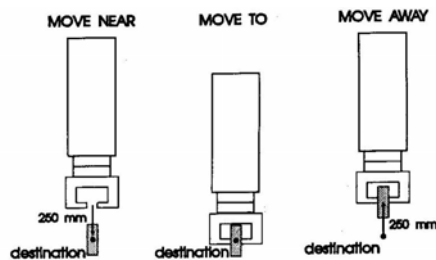
Con questa clausola si può specificare una destinazione posta lungo il vettore di approach dell'utensile, entro una certa distanza (espressa in mm) da una determinata posizione. Esempio:

MOVE NEAR *destination* BY 250.0

MOVE AWAY

Si può specificare una destinazione posta lungo il vettore di approach dell'utensile, ad una distanza specificata dalla posizione attuale. Esempio:

MOVE AWAY 250.0



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [75]

MOVE: clausole di destinazione

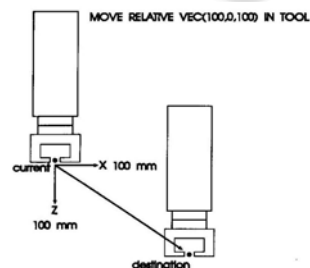


MOVE RELATIVE

Si può specificare una destinazione in relazione alla posizione attuale del braccio. Esempio:

MOVE RELATIVE VEC(100,0,100) IN *valore_terna*

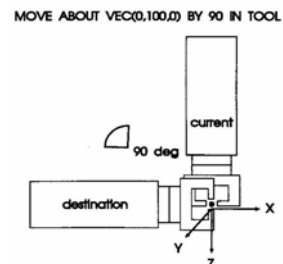
valore_terna può essere TOOL o BASE



MOVE ABOUT

Definisce la destinazione che deve raggiungere l'utensile dopo una rotazione attorno al vettore specificato rispetto alla posizione attuale. Esempio:

MOVE ABOUT VEC(0,100,0) BY 90 IN *valore_terna*



Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [76]

Velocità di percorrenza



Con la clausola opzionale **WITH** è possibile assegnare valori a delle variabili temporanee predefinite. In particolare è possibile operare sulle seguenti variabili:

\$PROG_SPD_OVR

È un valore percentuale con il quale si può modificare il valore di default di velocità con cui il braccio esegue i movimenti nello spazio dei giunti.

\$PROG_ACC_OVR, \$PROG_DEC_OVR

Sono valori percentuali con i quali si possono modificare i valori di default di accelerazione e decelerazione con cui il braccio esegue i movimenti nello spazio dei giunti.

\$LIN_SPD

È il valore della velocità lineare per un movimento cartesiano, espresso in metri al secondo.

Esempi: MOVE TO *p1* WITH \$PROG_SPD_OVR=50
MOVE LINEAR TO *p2* WITH \$LIN_SPD=0.6

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [77]

Movimento continuo (MOVEFLY)

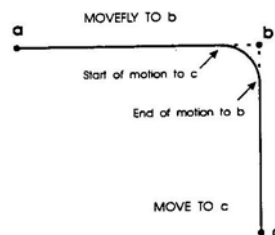


Se si utilizza l'istruzione **MOVEFLY** ed il movimento è seguito da un altro movimento, il braccio non si arresterà alla prima destinazione, ma si sposterà dal punto di partenza del primo movimento fino al punto finale del secondo, senza fermarsi sul punto comune ai due movimenti.

Esempio:

```
MOVE TO a
MOVEFLY TO b ADVANCE
MOVE TO c
```

(la clausola **ADVANCE** consente l'interpretazione della successiva istruzione **MOVE** non appena ha inizio il primo movimento).



L'istruzione **MOVEALONG** consente invece di eseguire un movimento composto da singoli segmenti di movimento.

Controllo del moto e robotica industriale - Pianificazione di traiettorie - P. Rocco [78]