

ROBOTICS

Lecture notes

Simone Carletti

A.Y. 2023/2024

Indice

1 Catene Cinematiche	3
1.1 Definizioni	3
1.2 Tipi di robot	5
1.2.1 Possibili shoulder configurations	5
1.2.2 Wrist	7
2 Cinematica delle posizioni	8
2.1 Cinematica Diretta	8
2.2 Convenzioni di Denavit-Hartenberg	10
2.2.1 Sistemi di riferimento	10
2.2.2 Trasformazione delle coordinate	13
2.3 Operational space e Joint space	13
2.4 Workspace (spazio di lavoro)	14
2.5 Accuracy e Repeatability	15
2.6 Cinematica inversa	15
2.6.1 Manipolatore con polso sferico	16
3 Cinematica delle velocità	17
3.1 Jacobiano geometrico	18
3.1.1 Costruzione dello Jacobiano geometrico	19
3.2 Jacobiano analitico	22
3.3 Singolarità cinematiche	22
3.3.1 Manipolatori con polso sferico	23
3.4 Ridondanza cinematica	24
3.5 Cinematica inversa delle velocità	25
3.5.1 Metodologie avanzate per il caso rettangolare/ridondante	25
3.5.2 Raggiungimento di obiettivi secondari	27
3.5.3 Inversione dello Jacobiano nel caso di singolarità	27
3.6 Cinematica inversa delle posizioni	28
3.6.1 section:invkineinvjac	28
3.6.2 Cinematica inversa tramite Jacobiano trasposto	29
4 Statica	31
4.1 Relazione fra F e τ	32
4.2 Dualità cineto-statica	33
4.3 Ellissoidi di manipolabilità	34
4.3.1 Velocità	34
4.3.2 Forza	35
5 Dinamica	37

5.1	Lagrangiana ed equazioni di Eulero-Lagrange	37
5.1.1	Energia cinetica	37
5.1.2	Energia potenziale	40
5.2	Equazioni dinamiche del manipolatore	40
5.2.1	Interpretazione fisica dei termini	41
5.2.2	Forze non conservative	42
6	Traiettorie	43
6.1	Traiettorie nello spazio dei giunti	44
6.1.1	Point-to-Point (PTP) motion	44
6.1.2	Profilo di velocità trapezoidale (2-1-2)	46
6.1.3	Movimento attraverso una sequenza di punti	50
6.2	Traiettorie nello spazio operazionale	51
6.2.1	Primitive	51
6.2.2	Primitiva per un segmento	52
6.2.3	Primitiva per arco di circonferenza	52
6.2.4	Generazione traiettorie	53
7	Controllo	55
7.1	Attuatori elettrici	56
7.1.1	Derivazione modello	57
7.1.2	Amplificatore di potenza	58
7.1.3	Velocity generator	58
7.1.4	Torque generator	59
7.1.5	Torque generator <i>vs.</i> Velocity generator	59
7.2	Trasmissions <i>aka</i> motoriduttori	60
7.3	Controllo decentralizzato a giunti indipendenti	61
7.3.1	Controllo del singolo giunto	62
7.3.2	Struttura del controllo	64
7.3.3	Attuatori saturanti	69
7.3.4	Feedforward compensation	70
7.3.5	Computed torque feedforward control	70
7.4	Controllo centralizzato ai giunti	71
7.4.1	Controllo a dinamica inversa	72
7.4.2	Controllo PD con compensazione della gravità	75
7.5	Controllo nello spazio operazionale	77
7.5.1	Controllo a Jacobiano inverso	78
7.5.2	Controllo a Jacobiano trasposto	78
7.5.3	Controllo PD con compensazione della gravità	79
7.5.4	Controllo a dinamica inversa	80
7.5.5	Considerazioni generali	81
7.6	Controllo dell'interazione	81
7.6.1	Cedevolezza passiva ed attiva	81
7.6.2	Strategie di controllo dell'interazione	82
7.6.3	Controllo di rigidezza (<i>compliance control</i>)	82
8	Mobile Robotics	87
A	Control	115
A.1	Cose dette da Rizzo che non ho capito che c'entrano	115
A.2	Derivazione del valore di steady-state per il motore velocity-generator	116

Capitolo 1

Catene Cinematiche

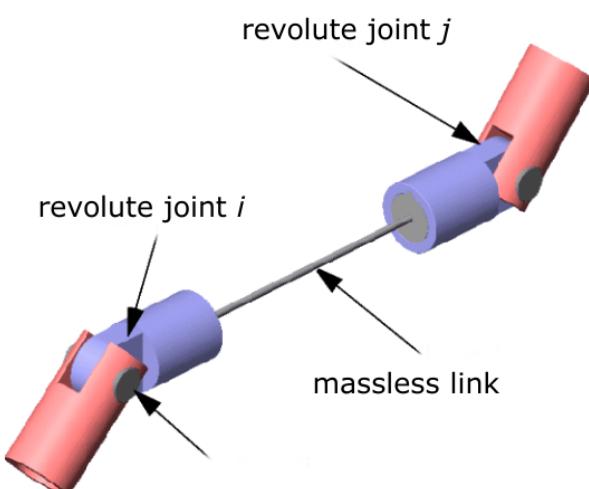
Una **kinematic chain** (KC) è composta da un numero variabile di **links** e **joints** (entrambi rigidi ed ideali) **connessi fra loro**. È importante sottolineare che una KC è un **puro oggetto geometrico** (no massa, no frizioni, etc.).

Per analizzare una KC viene posto un sistema di riferimento (RF, *reference frame*) su ogni link. In particolare esiste una convenzione per posizionare questi RF, chiamata **convenzione di Denavit–Hartenberg**.

1.1 Definizioni

- **Link:** barra geometrica idealizzata che connette 2 o più giunti
- **Joint:** componente fisico ideale che permette il movimento relativo fra 2 links (un giunto permette un singolo **DoF**: translazione o rotazione). Ne esistono di 2 tipi:
 - **Revolute** (rotational) joint
 - **Prismatic** (translational) joint

In generale i giunti sono mossi da attuatori (motori, etc.). Se questo non accade il giunto viene chiamato *passivo*.



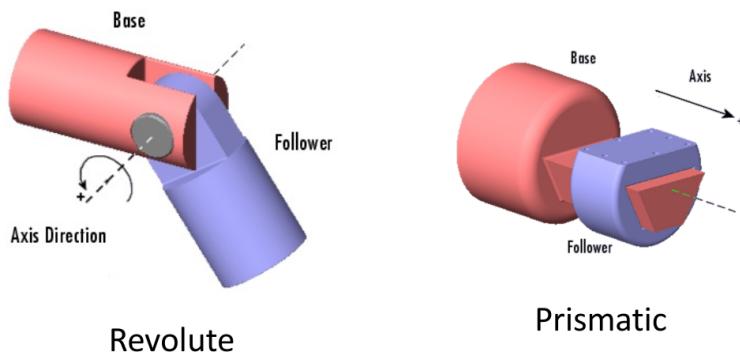


Figura 1.1: Tipi di giunti

Noi analizzeremo solo **open chains** (in contrasto alle *closed chains*): i.e. catene dove c'è solo 1 link fra due giunti qualunque. In questo caso la KC ha una struttura ad albero. Nel caso delle **closed chain**, invece, ci sono in generale più di un link fra due giunti e la struttura è ciclica.

I vari giunti sono schematizzati come mostrato in fig. 1.2.

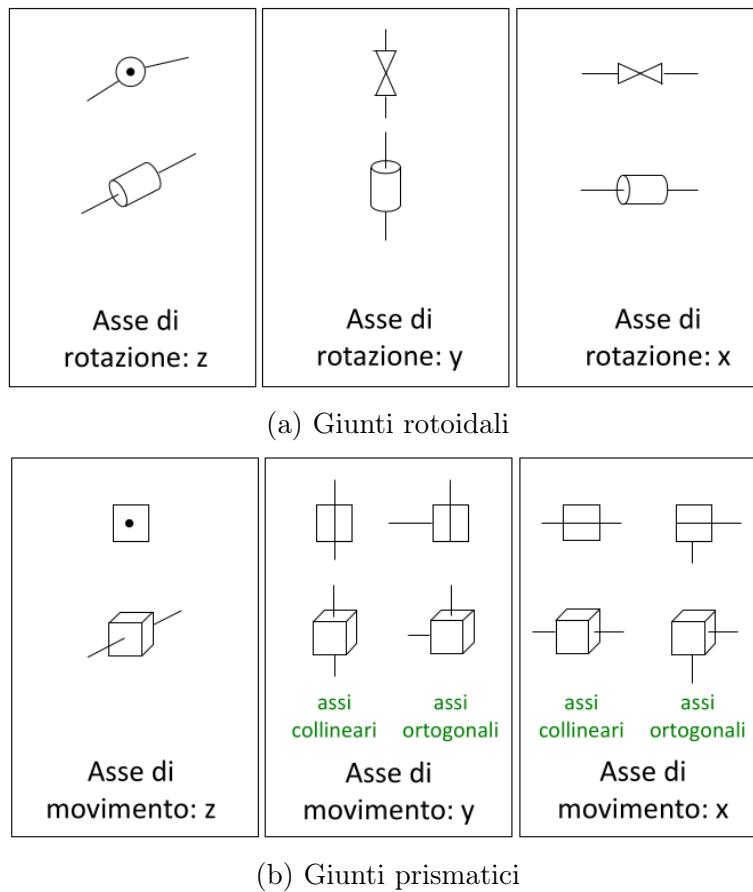


Figura 1.2

End Effector

Oltre al braccio, un robot solitamente ha un *tool* posto all'estremità (ultimo link). Questo è chiamato **end effector**, **gripper** o semplicemente **end tool**.

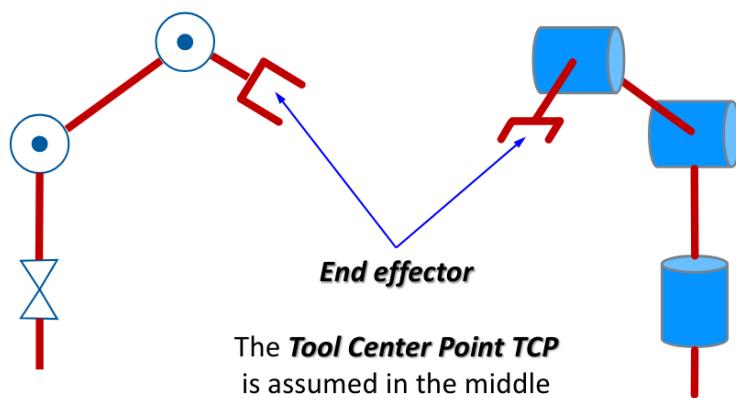


Figura 1.3: Braccio robotico schematizzato

Il **TCP** (*Tool Center Point*) è quel punto ideale sul'end-effector che i software muoveranno nello spazio. Come i vari giunti/link anche questo ha un proprio RF.

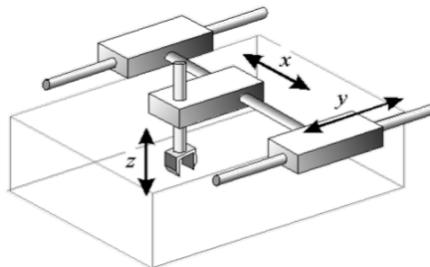
1.2 Tipi di robot

I robot industriali sono solitamente composti da una **shoulder** e un **wrist** (spalla e polso). Ponendo per notazione: P = "prismatic joint", R = "revolute joint". Di seguito i vari tipi, con le configurazioni delle relative shoulders.

1.2.1 Possibili shoulder configurations

Cartesian

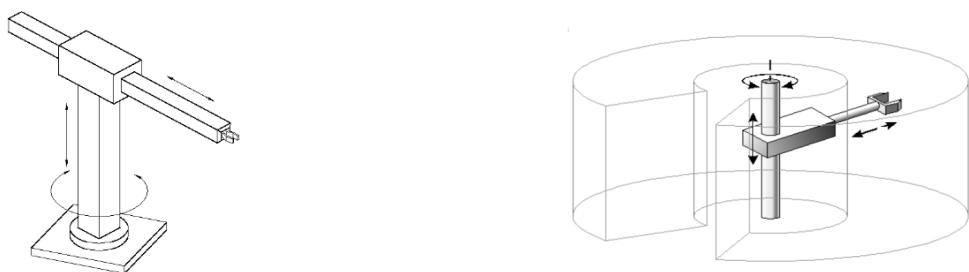
3P = P-P-P: abbiamo 3 DoF, corrispondenti alle 3 coordinate cartesiane. Il *task space* è un parallelepipedo. Posizionamento accurato ma destierità limitata.



Cylindrical

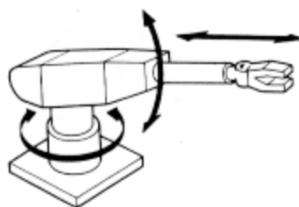
1R-2P = R-R-P. Abbiamo 3 DoF, corrispondenti alle 3 coordinate cilindriche. Il *task space* è una sezione di cilindro.

La struttura essendo meno rigida rispetto al precedente ci da meno accuratezza (che va a diminuire con l'elongazione del braccio).



Polar or spherical

2R-1P = R-R-P. Abbiamo 3 DoF, corrispondenti alle 3 coordinate polari. Il *task space* è una sezione di una sfera. È ancora meno rigido rispetto alle precedenti (e come nel caso cilindrico l'accuratezza diminuisce con l'elongazione).



SCARA

2R-1P = R-R-P. SCARA = Selective Compliance Assembly Robot Arm). Utili per manipolare piccoli componenti (e.g. μ C). Abbiamo una corrispondenza fra DoM (degree of motion) e coordinata cartesiana solo per la componente verticale. L'effetto della gravità è compensato dalla struttura stessa, che è rigida verticalmente ma *compliant* orizzontalmente.



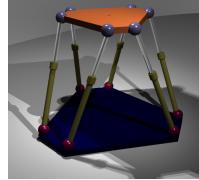
Articulated or Anthropomorphic

3R = R-R-R. Simile ad un braccio umano. Non c'è corrispondenza fra giunti e coordinate cartesiane. Il *task space* è una specie di sezione di una sfera. L'accuratezza non è costante in tutto il task-space. **È la tipologia più comune** visto che ci da la migliore dexterità.



Parallel or closed chains

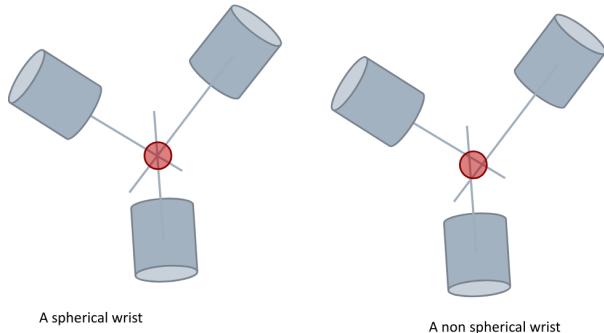
Utili per manipolare payload pesanti, visto che questo tipo di struttura è molto rigida.



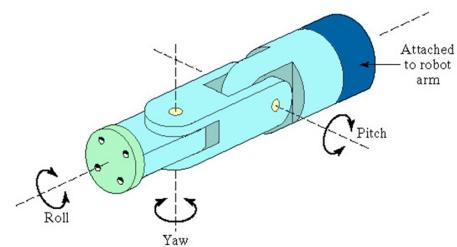
1.2.2 Wrist

Lo scopo principale del polso è quello di orientare il TCP. Possiamo dire che la shoulder setta l'origine del TCP mentre il polso la sua orientazione.

La tipologia più comune è quello dello **spherical wrist**. Comunemente un wrist (sferico e non) è composto da 3 rotational joints.



(a) Spherical vs non-spherical

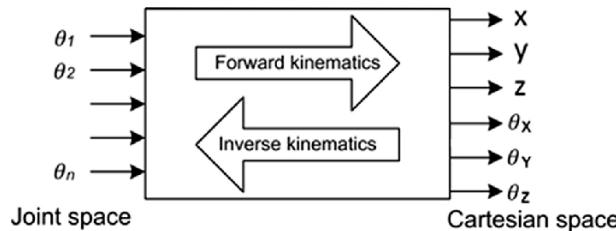


(b) Esempio

Capitolo 2

Cinematica delle posizioni

In questo capitolo studieremo il metodo geometrico per ottenere la posizione dell'end-effector a partire dalle joint variables (angoli o posizioni). Notare che qua non consideriamo forze e coppie.



2.1 Cinematica Diretta

Sia \mathcal{R}_b il **base frame** solidale con la base del robot, e sia \mathcal{R}_e un sistema di riferimento mobile solidale con la punta operativa (end-effector) del robot. Di conseguenza, la posa dell'end-effector sarà data dalla posizione ed orientamento di \mathcal{R}_e rispetto a \mathcal{R}_b .

La cinematica diretta ha come scopo il calcolo della posa della punta operativa in funzione delle variabili giunto q_i , quindi dobbiamo esprimere la relazione fra i 2 RF in modo analitico. Possiamo osservare che \mathcal{R}_e può essere rappresentato nel base frame \mathcal{R}_b per mezzo della matrice di trasformazione:

$${}^b\mathbf{T}_e(\mathbf{q}) = \begin{bmatrix} {}^b\mathbf{R}_e(\mathbf{q}) & {}^b\mathbf{p}_e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^b\mathbf{n}_e(\mathbf{q}) & {}^b\mathbf{s}_e(\mathbf{q}) & {}^b\mathbf{a}_e(\mathbf{q}) & {}^b\mathbf{t}_e(\mathbf{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

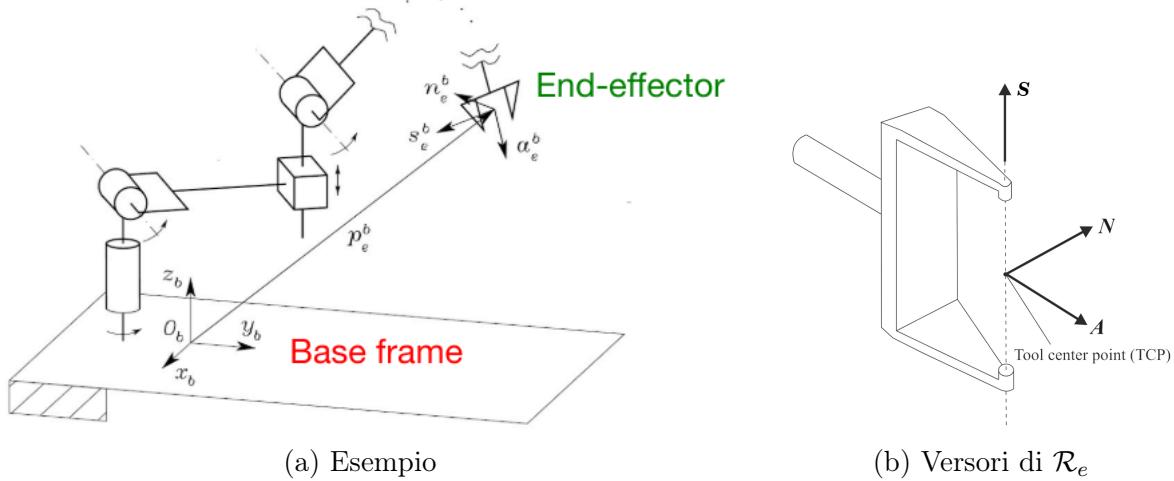
dove:

- ${}^b\mathbf{p}_e(\mathbf{q})$ è l'origine di \mathcal{R}_e in \mathcal{R}_b
- ${}^b\mathbf{R}_e(\mathbf{q})$ è la matrice di rotazione di \mathcal{R}_e rispetto a \mathcal{R}_b
- ${}^b\mathbf{n}_e(\mathbf{q}), {}^b\mathbf{s}_e(\mathbf{q}), {}^b\mathbf{a}_e(\mathbf{q})$ sono i 3 versori del sistema di riferimento dell'end-effector \mathcal{R}_e (rispettivamente **approach**, **sliding** e **normal**).

Il nome dei 3 versori deriva dal fatto che:

- approach: posizionato nella direzione di approccio dell'utensile al pezzo

- sliding: posizionato nel piano di scorrimento (sliding) delle ganasce
- normal: ottenuto come prodotto vettoriale fra i primi due (e quindi normale a loro).



Visto che

$$\mathbf{x} = (x, y, z, \phi, \theta, \psi) = {}^b\mathbf{T}_e(\mathbf{q})$$

risolvere la cinematica diretta significa determinare trovare la matrice \mathbf{T} , che è specifica per ogni particolare robot (essendo dipendente dalla sua struttura fisica).

Soltanto nel caso di manipolatori estremamente semplici, con pochi gradi di libertà, è possibile risolvere il problema della cinematica diretta delle posizioni per via geometrica (i.e a manina). In generale, per un manipolatore con $n + 1$ links, collegati da n giunti, è necessario ricorrere ad una procedura sistematica, basata su:

- Definizione di $n + 1$ sistemi di riferimento $\mathcal{R}_0 \dots \mathcal{R}_n$ ciascuno solidale con un braccio del robot dalla base (braccio/link 0, $\mathcal{R}_0 \equiv \mathcal{R}_b$) fino all'ultimo braccio (n)
- Calcolo della matrice di trasformazione come prodotto (composizione) delle varie matrici di trasformazione fra i singoli sistemi di riferimento:

$${}^b\mathbf{T}_e(\mathbf{q}) \equiv {}^0\mathbf{T}_n(\mathbf{q}) = {}^b\mathbf{T}_1(q_1){}^1\mathbf{T}_2(q_2) \cdots {}^{n-1}\mathbf{T}_n(q_n)$$

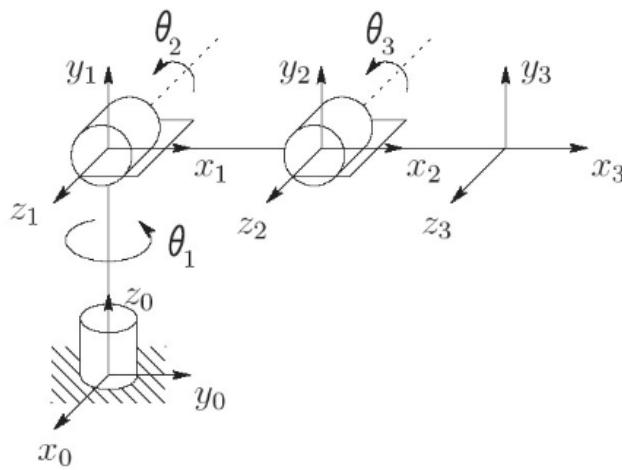


Figura 2.2: Sistemi di riferimento $\mathcal{R}_0 \dots \mathcal{R}_n$

Nota: la matrice $\mathbf{T}(\mathbf{q})$ non è costante, ma dipende dalla configurazione corrente del robot!

Inoltre, se per qualche motivo la base e l'end effector non coincidono con i frame 0 e n ci basta aggiungere le 2 rispettive trasformazioni

$${}^b\mathbf{T}_e(\mathbf{q}) = {}^b\mathbf{T}_0 {}^0\mathbf{T}_n(\mathbf{q}) {}^n\mathbf{T}_e$$

che in generale sono costanti e non dipendono da \mathbf{q} .

2.2 Convenzioni di Denavit-Hartenberg

Queste convenzioni stabiliscono delle regole generali e sistematiche per la definizione dei sistemi di riferimento di ogni link. In particolare ci permettono di definire in maniera sistematica la matrice ${}^{i-1}\mathbf{T}_i(q_i)$.

Se seguiamo le regole della convenzione (per il posizionamento dei RF), ogni matrice di quel tipo sarà funzione di soli 4 parametri:

- a_i : lunghezza del link i
- α_i : twist del link i
- d_i : offset del link i
- θ_i : angolo del giunto i

Inoltre, visto che la matrice finale ${}^{i-1}\mathbf{T}_i(q_i)$ è dipendente solo da una variabile (q_i), solo 1 fra quei 4 parametri potrà essere variabile, mentre gli altri saranno costanti. In particolare quelli variabili sono:

- Se giunto rotoidale: $\theta_i = q_i$
- Se giunto prismatico: $d_i = q_i$

2.2.1 Sistemi di riferimento

Introduciamo ora il metodo che dovremmo utilizzare per definire i vari RF.

In particolare vanno definiti $n + 1$ sistemi di riferimento \mathcal{R}_i , ognuno solidale con uno specifico

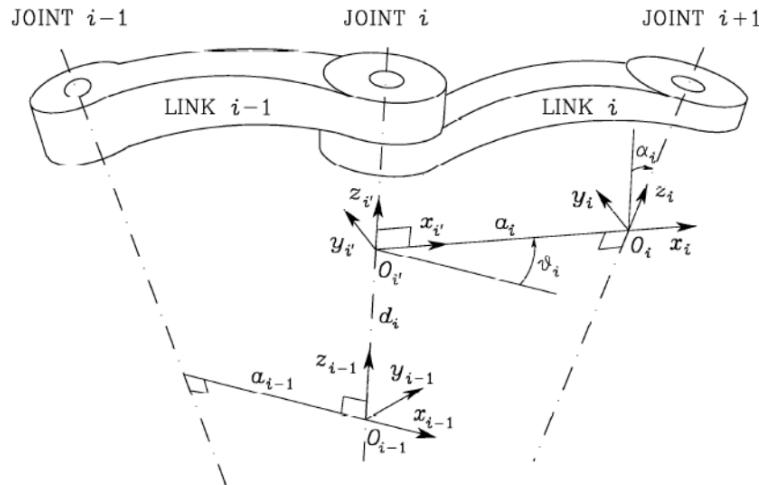


Figura 2.3: Convenzione di Denavit-Hartenberg

braccio/link i (importante notare che questo RF quindi si muoverà con quel particolare braccio, che a sua volta è attuato dal joint i)

Un ulteriore sistema di riferimento solidale con la punta operativa può essere introdotto, orientandone gli assi in maniera consona alla definizione del compito da svolgere.

Il sistema di riferimento \mathcal{R}_i solidale con $LINK_i$ viene definito secondo le seguenti regole¹:

① **Asse z_i e origine O_i**

- **L'asse z_i** è posto lungo l'asse di movimento di g_{i+1} (asse di rotazione o di traslazione a seconda del tipo di giunto)
- **L'origine O_i** è posta all'intersezione di z_i con la normale comune (*common normal*) fra gli assi z_{i-1} e z_i . La normale comune è quella retta perpendicolare ad entrambi gli assi (nota entrambi angoli retti nella figura)
- **Casi particolari:**
 - \mathcal{R}_0 : qua è univocamente definita solo la direzione di z_0 , data dall'asse di movimento di g_1 ; l'origine O_0 e x_0 possono essere fissati a piacimento
 - \mathcal{R}_n : visto che non esiste il giunto g_{n+1} , z_n e O_n non sono univocamente definiti. Per consuetudine si fissa l'origine nel centro della pinza. E visto che tipicamente l'ultimo giunto è rotoidale, z_n è presa coincidente a z_{n-1} (così da semplificare la matrice di rotazione, visto che otterremo elementi nulli)

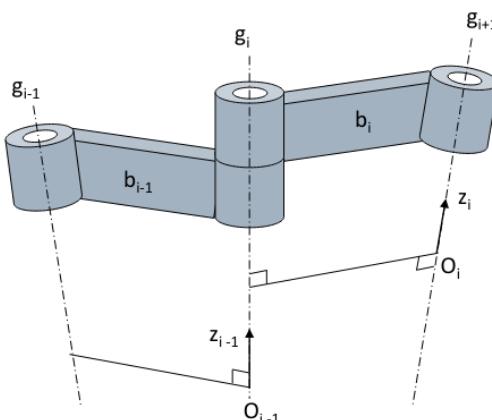


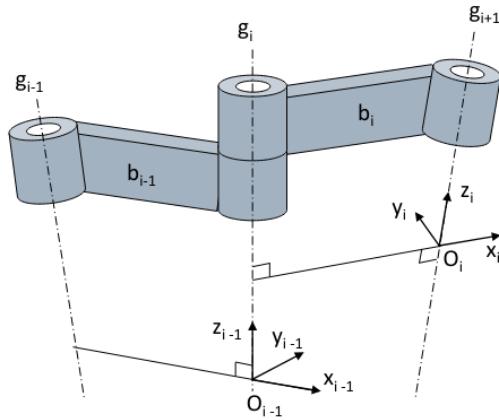
Figura 2.4: Assi z e origini

② **Asse x_i e y_i**

- **L'asse x_i** è fissato lungo la normale comune fra gli assi z_{i-1} e z_i
 - Se z_{i-1} e z_i si intersecano, l'origine di \mathcal{R}_i coincide con il loro punto di intersezione e la direzione di x_i (ortogonale a z_i) è arbitraria
 - se z_{i-1} e z_i sono paralleli, l'origine può essere posta in un punto a scelta e x_i appartiene al piano normale a z_{i-1} e z_i con direzione e verso arbitrari

¹Notazioni slide italiano: $b_i \equiv LINK_i$ $g_i \equiv JOINT_i$

- L'asse y_i completa la terna destrorsa

Figura 2.5: Assi x e y

(3) Parametri di DH

Per comprendere meglio i parametri di DH possiamo introdurre un sistema di riferimento "intermedio" $\mathcal{R}_{i'}$ (vedi fig. 2.6b):

- $z_{i'}$ diretto lungo z_{i-1}
- $O_{i'}$ posta all'intersezione di z_{i-1} con la normale comune fra z_{i-1} e z_i
- $x_{i'}$ diretto lungo la normale comune fra z_{i-1} e z_i (come x_i)

La posizione e l'orientamento di \mathcal{R}_i sono completamente specificati rispetto a \mathcal{R}_{i-1} dai seguenti 4 parametri:

- $d_i \rightarrow \text{link offset}$: coordinata di $O_{i'}$ lungo z_{i-1}
- $\theta_i \rightarrow \text{joint angle}$: angolo di rotazione da x_{i-1} a x_i attorno all'asse $z_{i'}$ (positivo quando la rotazione è anti-oraria)
- $a_i \rightarrow \text{link length}$: distanza (con segno) fra O_i e $O_{i'}$
- $\alpha_i \rightarrow \text{link twist}$: angolo di rotazione da z_{i-1} a z_i attorno all'asse x_i (positivo quando la rotazione è anti-oraria)

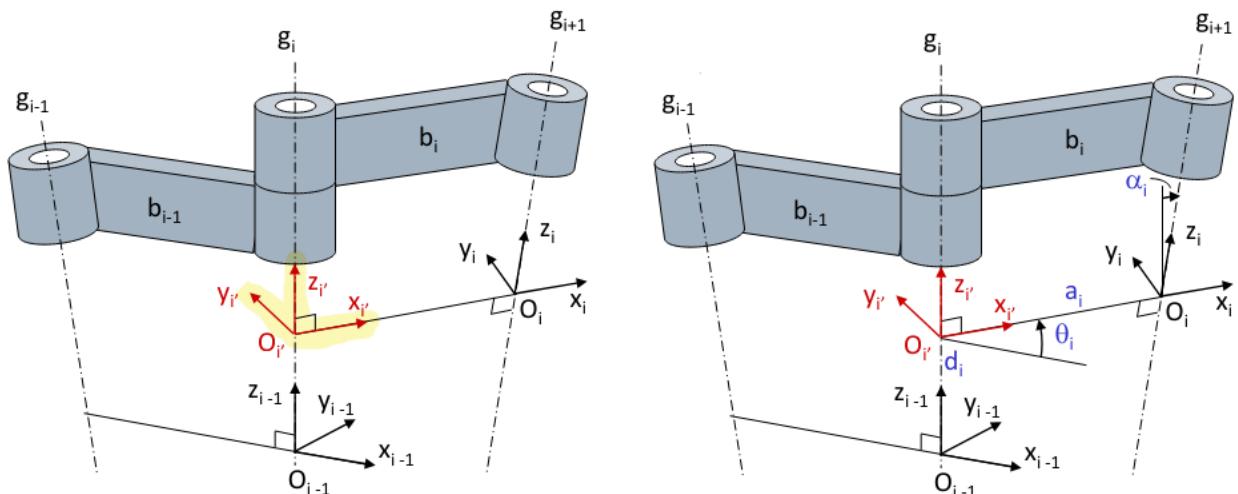


Figura 2.6

2.2.2 Trasformazione delle coordinate

Ora, come possiamo usare questi parametri per ottenere $\mathbf{T}(\mathbf{q})$?

① $\mathcal{R}_{i-1} \rightarrow \mathcal{R}_{i'}$

$\mathcal{R}_{i'}^i$ (partendo da \mathcal{R}_{i-1}) è dato da una translazione d_i lungo z_{i-1} seguito da una rotazione di θ_i attorno a z_i , quindi:

$${}^{i-1}\mathbf{T}_{i'} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

② $\mathcal{R}_{i'} \rightarrow \mathcal{R}_i$

\mathcal{R}_i (partendo da $\mathcal{R}_{i'}$) è dato da una translazione a_i lungo $x_{i'}$ seguito da una rotazione di α_i attorno a x_i , quindi:

$${}^{i'}\mathbf{T}_i = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

③ $\mathcal{R}_{i-1} \rightarrow \mathcal{R}_i$

Infine, componendo le due trasformazioni, otteniamo:

$${}^{i-1}\mathbf{T}_i(q_i) = {}^{i-1}\mathbf{T}_{i'} {}^{i'}\mathbf{T}_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Quindi, per ottenere ${}^0\mathbf{T}_n(\mathbf{q})$ non ci basterà che comporre le varie matrici ottenute sopra, le quali possono essere facilmente ricavate inserendo i giusti valori dei 4 parametri.

2.3 Operational space e Joint space

L'**operational space (task space, o cartesian space)** è quello spazio dove è definito il vettore di posizioni e orientamento dell'end effector, ovvero:

$$\mathbf{x}_e = (\mathbf{p}_e, \boldsymbol{\phi}_e) = (x, y, z, \phi, \theta, \psi)$$

In contrasto, il **joint space (o configuration space)** è quello spazio dove è definito:

$$\mathbf{q} = [q_1 \dots q_n] \in \mathbb{R}^n$$

q_i è la singola variabile di un giunto (ricordiamo che a ciascun giunto corrisponde un grado di libertà). In particolare:

- Giunto rotoidale: $q_i = \theta_i$ (rotazione)
- Giunto prismatico: $q_i = d_i$ (traslazione)

La postura della catena cinematica è quindi determinata dalla posa di tutti i corpi rigidi (i bracci) che la costituiscono ed è una funzione di \mathbf{q} :

$$\mathbf{x}_e = \begin{bmatrix} x(q_1, \dots, q_n) \\ y(q_1, \dots, q_n) \\ z(q_1, \dots, q_n) \\ \phi(q_1, \dots, q_n) \\ \theta(q_1, \dots, q_n) \\ \psi(q_1, \dots, q_n) \end{bmatrix} = \mathbf{k}(\mathbf{q})$$

dove $\mathbf{k}(\mathbf{q})$ è la **funzione di cinematica diretta**.

Una soluzione di questa funzione è data dalla matrice ${}^0\mathbf{T}_n(\mathbf{q})$ definita prima. Notiamo che però la soluzione trovata utilizza una rappresentazione non minimale dell'assetto (matrice di rotazione), mentre nella definizione di $k(q)$ è minimale (angoli di eulero).

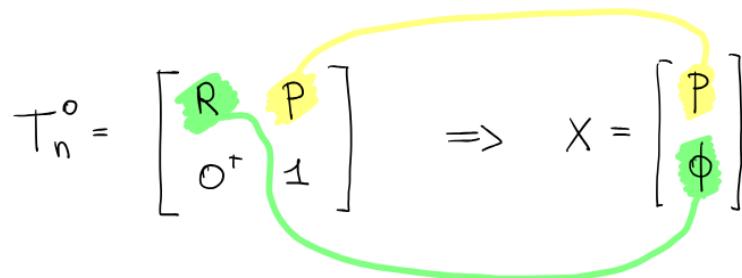


Figura 2.7

2.4 Workspace (spazio di lavoro)

Con riferimento allo spazio operazionale lo **spazio di lavoro** del robot è definito da tutti quei punti di \mathbf{x}_e ottenuti eseguendo tutte le possibili configurazioni dei giunti.

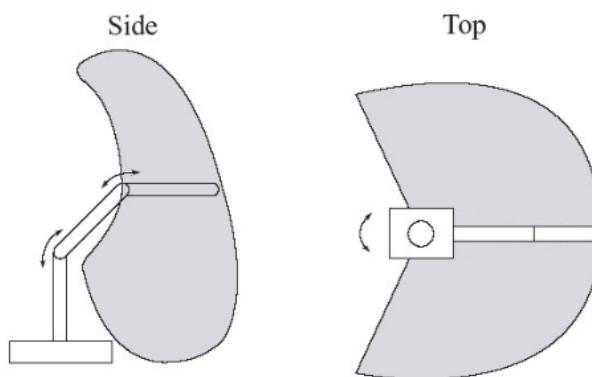


Figura 2.8: Esempio di spazio di lavoro

All'interno di questo spazio possiamo specificarne due:

- **Reachable workspace**: è lo spazio di lavoro che l'end-effector può raggiungere con almeno un orientamento
- **Dexterous workspace**: è lo spazio di lavoro che l'end-effector può raggiungere con più di un orientamento. Ovviamente *dexterous workspace* \subset *reachable workspace*

Ad esempio, la punta di un manipolatore con meno di 6 gradi di libertà non può certamente raggiungere qualunque posa nello spazio 3D.

2.5 Accuracy e Repeatability

Per via di imprecisioni meccaniche, esisteranno sempre delle discrepanze fra i parametri DH nominali e quelli reali. Questa discrepanza prende il nome di **accuratezza**. Un requisito necessario per averla elevata è avere una struttura **rigida**.

Nei manipolatori moderni è $< 1\text{mm}$ e solitamente varia con la posizione dell'end-effector.

La **repetibilità** invece è la capacità del manipolatore di ritornare ad una posizione precedente con un certo grado di precisione (questa dipende dalla struttura meccanica ma anche dalle strategie di controllo utilizzate e dalla precisione dei trasduttori).

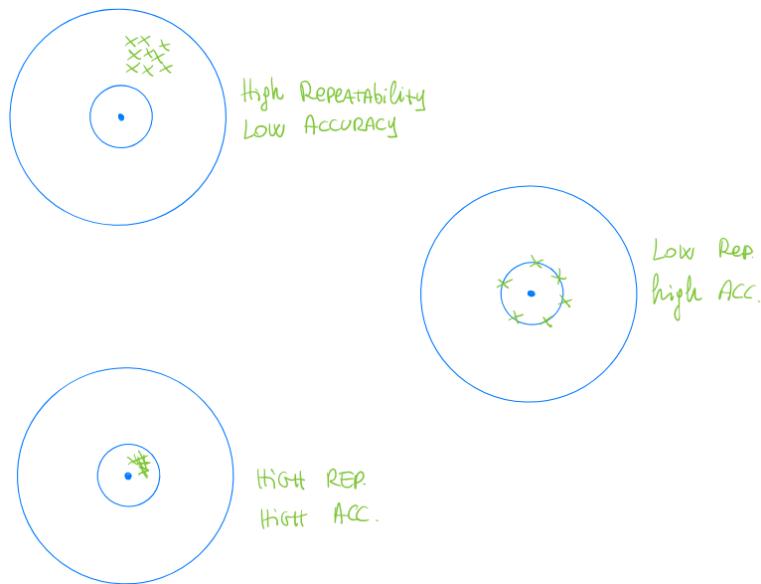


Figura 2.9: Esempi accuracy e repeatability

2.6 Cinematica inversa

Il problema della cinematica inversa delle posizioni consiste nella determinazione delle variabili giunto \mathbf{q} corrispondenti ad un dato \mathbf{x}_e o una data ${}^b\mathbf{T}_e(\mathbf{q})$.

L'esistenza della soluzione a tale problema è garantita solo se la posizione e l'orientamento della punta appartengono allo spazio di lavoro (di destrezza) del manipolatore.

Il **problema** della cinematica inversa delle posizioni è **molto più complesso** di quello della cinematica diretta:

- Le **equazioni** da risolvere sono in generale **non lineari** (a causa di \sin , \cos , ...) e di conseguenza non è sempre possibile trovare una soluzione in forma chiusa
- **Possono esistere molteplici soluzioni** (cioè più posture del manipolatore corrispondenti alla stessa posizione ed assetto della punta). Addirittura possono risultare infinite nel caso di un manipolatore ridondante

- **Possono anche non esistere proprio soluzioni:** vincoli cinematici nella struttura reale del robot potrebbero ridurre o azzerare il numero di soluzioni "ammissibili"

Puttropo **non esistono procedure** per il calcolo della cinematica inversa di un robot: la determinazione della soluzione in forma chiusa è fortemente basata su **intuizioni algebriche e geometriche**.

Nei casi in cui sia impossibile (o estremamente difficoltoso) giungere alla soluzione in forma chiusa si ricorre a **tecniche numeriche di soluzione**, ad esempio:

$$\min_q = \|\mathbf{x}_{e\ ref} - \mathbf{k}(\mathbf{q})\|$$

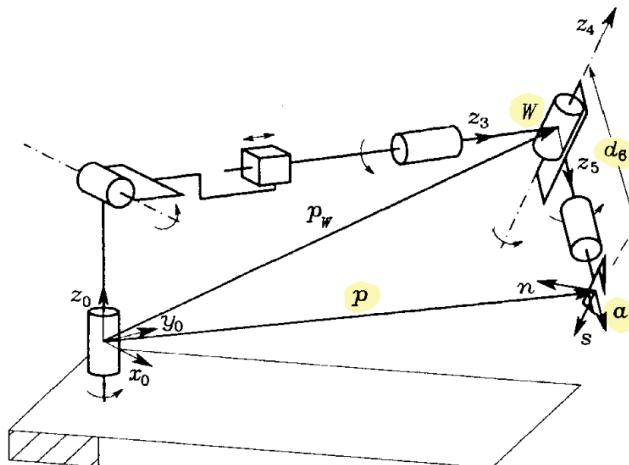
Per quanto riguarda quest'ultime abbiamo come vantaggio che sono applicabili a qualunque struttura cinematica, ma che però non forniscono (in generale) tutte le possibili soluzioni.

Alternativamente è possibile utilizzare la **Jacobiana** (che vedremo più avanti), invertendo le velocità ed integrando per ottenere la posizione.

2.6.1 Manipolatore con polso sferico

Nel caso di un manipolatore a 6 DOF con polso sferico, è **possibile disaccoppiare** il problema della cinematica inversa in due sottoproblemi, relativi uno alla determinazione della posizione (del polso) e l'altro dell'assetto (dell'end-effector).

Indichiamo il centro del polso (inteso come la posizione in cui si intersecano gli assi dei suoi giunti) con \mathbf{W} .



Se \mathbf{p} e $(\mathbf{n}, \mathbf{s}, \mathbf{a})$ (posizione e orientamento del polso) sono noti:

$$\mathbf{W} = \mathbf{p} - d_6 \mathbf{a}$$

poichè \mathbf{W} , rappresentato dal vettore p_W , è ottenibile come sottrazione vettoriale fra \mathbf{p} e il vettore lungo d_6 orientato nel verso di \mathbf{a} .

Grazie a questa osservazione, possiamo ora suddividere il problema della cinematica inversa in 2 parti:

1. Dalla posizione \mathbf{p} della punta è possibile ricavare \mathbf{W} e quindi risolvere la cinematica inversa per determinare le tre variabili giunto "della spalla": calcola la posizione del polso $\mathbf{W}(q_1, q_2, q_3)$ tramite la formula sopra, e poi risolvi la cinematica inversa per (q_1, q_2, q_3)
2. Tenendo conto della cinematica ormai nota della spalla, si passa al calcolo dei restanti q_i a partire dall'assetto della punta: calcola ${}^0R_3(q_1, q_2, q_3) \Rightarrow {}^3R_6(\theta_1, \theta_2, \theta_3) = ({}^0R_3)^T {}^0R_e$ e quindi risolvi la cinematica inversa per $(\theta_1, \theta_2, \theta_3)$ tramite 3R_6

Capitolo 3

Cinematica delle velocità

Differential Kinematics

La cinematica delle velocità (o cinematica differenziale) di un manipolatore è rappresentata dall'insieme di **relazioni che legano le velocità dei giunti alla velocità della punta operativa**, ovvero le relazioni fra la velocità nello spazio dei giunti e la velocità nello spazio operazionale.

Anche in questo caso abbiamo la relazione di cinematica diretta ($\dot{\mathbf{q}} \rightarrow \dot{\mathbf{x}}_e$) e di cinematica inversa ($\dot{\mathbf{q}} \leftarrow \dot{\mathbf{x}}_e$).

Mentre non ci sono ambiguità nella definizione della velocità nello spazio dei giunti, rappresentata dal vettore $\dot{\mathbf{q}}$, è necessario chiarire che cosa si intende per velocità nello spazio operazionale:

- **Velocità in senso geometrico:** si considera in tal caso il vettore $\mathbf{v} = (\dot{\mathbf{p}}, \boldsymbol{\omega}) \in \mathbb{R}^6$ delle velocità geometriche, formato dalla velocità lineare $\dot{\mathbf{p}} \in \mathbb{R}^3$ e dalla velocità angolare $\boldsymbol{\omega} \in \mathbb{R}^3$ della punta operativa
- **Velocità in senso analitico:** si definisce in tal caso come velocità nello spazio operazionale il vettore $\dot{\mathbf{x}} \in \mathbb{R}^6$ ottenuto derivando rispetto al tempo il vettore $\mathbf{x} = \mathbf{k}(\mathbf{q}) = (\mathbf{p}, \boldsymbol{\phi}) \in \mathbb{R}^6$ delle coordinate operazionali (determinate dalla cinematica diretta delle posizioni)

Notiamo che in entrambi i casi la parte lineare è identica (esprime la velocità lineare della punta operativa lungo le direzioni degli assi del sistema di riferimento \mathcal{R}_0). **La parte relativa all'orientamento**, invece, **differisce** (in generale $\boldsymbol{\omega} \neq \dot{\boldsymbol{\phi}}$):

- $\boldsymbol{\omega}$ corrisponde alla velocità angolare della punta (secondo le tre componenti relative agli assi di \mathcal{R}_0). È un "vero" vettore: $\boldsymbol{\omega}(t) = \boldsymbol{\omega}_1(t) + \dots + \boldsymbol{\omega}_n(t)$
- $\dot{\boldsymbol{\phi}}$ rappresenta la velocità con cui varia l'orientamento della punta, espressa come derivata rispetto al tempo dei tre angoli (ϕ, θ, ψ) (Eulero o RPY) della rappresentazione minima dell'assetto. A differenza di $\boldsymbol{\omega}$ questo non è un "vero" vettore, visto che $\dot{\boldsymbol{\phi}}(t) \neq \dot{\boldsymbol{\phi}}_1(t) + \dots + \dot{\boldsymbol{\phi}}_n(t)$ ma (al contrario del precedente, essendo questa una derivata) $\int \dot{\boldsymbol{\phi}}(\tau) d\tau = \boldsymbol{\phi}(\tau)$

In entrambi i casi i due vettori delle sono **funzioni lineari** delle velocità ai giunti, quindi:

$$\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (3.1)$$

$$\dot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}} \quad (3.2)$$

dove le due matrici sono rispettivamente chiamate **Jacobian geometrico** e **Jacobian analitico**. Inoltre, per entrambe, posto $n = \text{numero giunti}$:

$$\mathbf{J}, \mathbf{J}_A \in \mathbb{R}^{6 \times n}$$

Quindi, risolvere il problema della cinematica diretta delle velocità significa determinare \mathbf{J} o \mathbf{J}_A .

Importante notare anche che entrambi gli Jacobiani dipendono dalla configurazione corrente del robot, visto che sono funzioni di \mathbf{q} .

3.1 Jacobiano geometrico

Iniziamo studiando un modo per ricavare lo Jacobiano geometrico. Da (3.1) richiamiamo:

$$\mathbf{v} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \mathbf{J}_o(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} \quad (3.3)$$

dove abbiamo indicato la scomposizione dello Jacobiano nelle 2 sottomatrici $3 \times n$ relative alla sola posizione e orientamento.

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_p & \mathbf{J}_o \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \vdots \\ \dot{\mathbf{q}}_n \end{bmatrix}$$

Figura 3.1: Jacobiano geometrico

In primis possiamo iniziare osservando i contributi di ogni riga e ogni colonna:

- **Righe:** ogni riga specifica l'effetto di tutti i giunti sulla particolare velocità
- **Colonne:** ogni colonna, invece, identifica l'effetto del singolo giunto su tutte le velocità

Esempio:

Se la colonna i è completamente nulla, avremmo che il giunto i non avrà alcun effetto nella produzione di velocità dell'end-effector.

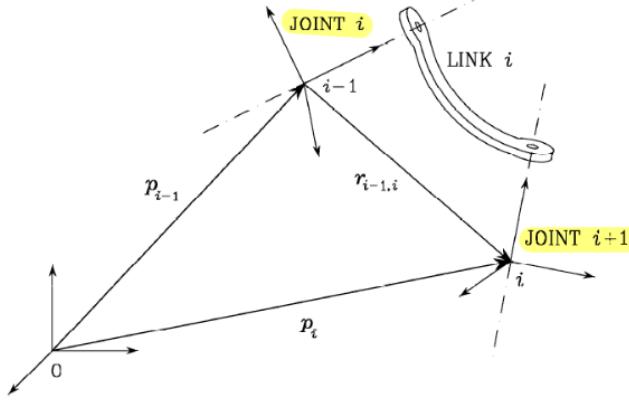
Invece se abbiamo una colonna nulla, e.g. la 1^a, avremmo che il nostro robot non produrrà alcuna velocità (e quindi nessun movimento) lungo x .

Anche dall'esempio possiamo intuire la grande utilità dello Jacobiano: **tramite questa matrice possiamo facilmente identificare quali velocità dei giunti influenzano maggiormente velocità operazionali**, oltre ad esempio, a rendere più facile l'inversione cinematica ($v = J(\mathbf{q})\dot{\mathbf{q}} \Rightarrow \dot{\mathbf{q}} = J^{-1}(\mathbf{q})v \Rightarrow \mathbf{q} = \int \dot{\mathbf{q}} = \int J^{-1}(\mathbf{q})v$), che vedremo più avanti.

3.1.1 Costruzione dello Jacobiano geometrico

Per comprendere come “costruire” lo Jacobiano geometrico, è utile determinare l'espressione della velocità (lineare ed angolare) di un singolo braccio.

Si consideri il generico braccio i , facente parte di una catena cinematica aperta, per la quale sono stati definiti i sistemi di riferimento secondo le convenzioni di DH.



Con riferimento alla figura sopra, possiamo affermare:

$$\begin{aligned}\dot{\mathbf{p}}_i &= \dot{\mathbf{p}}_{i-1} + \mathbf{v}_{i-1,i} + \boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1,i} \\ \boldsymbol{\omega}_i &= \boldsymbol{\omega}_{i-1} + \boldsymbol{\omega}_{i-1,i}\end{aligned}$$

dove $\mathbf{v}_{i-1,i} \triangleq \dot{\mathbf{r}}_{i-1,i}$ (per dim. vedi slide in italiano).

Nel caso di un **giunto prismatico** abbiamo:

$$\begin{cases} \boldsymbol{\omega}_{i-1,i} = 0 \\ \mathbf{v}_{i-1,i} = \dot{\mathbf{d}}_i \mathbf{z}_{i-1} \end{cases} \implies \begin{cases} \dot{\mathbf{p}}_i = \dot{\mathbf{p}}_{i-1} + \dot{\mathbf{d}}_i \mathbf{z}_{i-1} + \boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1,i} \\ \boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} \end{cases}$$

mentre per un **giunto rotoidale**:

$$\begin{cases} \boldsymbol{\omega}_{i-1,i} = \dot{\theta}_i \mathbf{z}_{i-1} \\ \mathbf{v}_{i-1,i} = \boldsymbol{\omega}_{i-1,i} \times \mathbf{r}_{i-1,i} \end{cases} \implies \begin{cases} \dot{\mathbf{p}}_i = \dot{\mathbf{p}}_{i-1} + \boldsymbol{\omega}_i \times \mathbf{r}_{i-1,i} \\ \boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\theta}_i \mathbf{z}_{i-1} \end{cases}$$

Richiamando la fig. 3.1 e (3.3), possiamo scomporre la matrice \mathbf{J} come segue:

$$\mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \mathbf{J}_o(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{p,1} & \mathbf{J}_{p,2} & \cdots & \mathbf{J}_{p,n} \\ \mathbf{J}_{o,1} & \mathbf{J}_{o,1} & \cdots & \mathbf{J}_{o,n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

dove $\mathbf{J}_{p,i}, \mathbf{J}_{o,i} \in \mathbb{R}^3$ sono le colonne di \mathbf{J}_p e \mathbf{J}_o (per semplicità di notazione abbiamo omesso la dipendenza da q). Da qui poi possiamo scrivere:

$$\dot{\mathbf{p}} = \sum_{i=1}^n \mathbf{J}_{p,i} \dot{q}_i \quad , \quad \boldsymbol{\omega} = \sum_{i=1}^n \boldsymbol{\omega}_{i-1,i} = \sum_{i=1}^n \mathbf{J}_{o,i} \dot{q}_i$$

e quindi:

- Nel caso di un **giunto prismatico** ($q_i = d_i$):

$$\begin{cases} \mathbf{J}_{p,i}\dot{q}_i = \mathbf{v}_{i-1,i} = \dot{d}_i \mathbf{z}_{i-1} \\ \boldsymbol{\omega}_{i-1,i} = 0 \end{cases} \implies \begin{cases} \mathbf{J}_{p,i} = \mathbf{z}_{i-1} \\ \mathbf{J}_{o,i} = \mathbf{0} \end{cases}$$

- Mentre per un **giunto rotoidale** ($q_i = \theta_i$):

$$\begin{cases} \mathbf{J}_{p,i}\dot{q}_i = \mathbf{v}_{i-1,i} = \boldsymbol{\omega}_{i-1,i} \times \mathbf{r}_{i-1,e} = \dot{\theta}_i \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ \boldsymbol{\omega}_{i-1,i} = \dot{\theta}_i \mathbf{z}_{i-1} \end{cases} \implies \begin{cases} \mathbf{J}_{p,i} = \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ \mathbf{J}_{o,i} = \mathbf{z}_{i-1} \end{cases}$$

Riassumendo:

$$\text{i-th column of } \mathbf{J} = \begin{bmatrix} \mathbf{J}_{p,i} \\ \mathbf{J}_{o,i} \end{bmatrix} = \begin{cases} \begin{bmatrix} \mathbf{z}_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{for a prismatic joint} \\ \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} & \text{for a revolute joint} \end{cases}$$

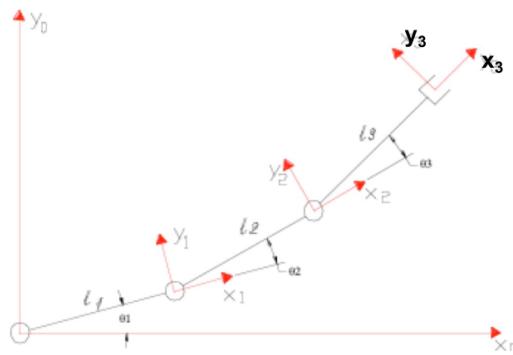
Costruzione pratica

Nella pratica, per trovare i vettori della formula di cui sopra, possiamo fare delle osservazioni.

- $\mathbf{z}_{i-1} = 3^a$ colonna di ${}^0\mathbf{R}_{i-1} = {}^0\mathbf{R}_1 \cdots {}^{i-2}\mathbf{R}_{i-1}$.
- $\mathbf{p}_e = \text{primi 3 elementi dell'ultima colonna di } {}^0\mathbf{T}_n = {}^0\mathbf{T}_1 \cdots {}^{n-1}\mathbf{T}$
- $\mathbf{p}_{i-1} = \text{primi 3 elementi dell'ultima colonna di } {}^0\mathbf{T}_{i-1} = {}^0\mathbf{T}_1 \cdots {}^{i-2}\mathbf{T}_{i-1}$

Esempio

Per semplicità non useremo il grassetto per i vettori e $p \equiv p_e$. Come si vede dalla figura abbiamo tutti giunti rotoidali.



$$J(q) = \begin{bmatrix} z_0 \times (p - p_0) & z_1 \times (p - p_1) & z_2 \times (p - p_2) \\ z_0 & z_1 & z_2 \end{bmatrix}$$

e quindi:

$${}^0T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, p_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & l_1 c_1 \\ s_1 & c_1 & 0 & l_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, p_1 = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \\ 0 \end{bmatrix}$$

$${}^0T_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies z_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, p_2 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \\ 0 \\ 0 \end{bmatrix}$$

$${}^0T_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies p = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\ 0 \\ 0 \end{bmatrix}$$

Ora dobbiamo solo calcolarci le differenze ed i prodotti vettoriali per trovare $J(q)$:

$$z_0 \wedge (p - p_0) = \begin{bmatrix} -\ell_1 s_1 - \ell_2 s_{12} - \ell_3 s_{123} \\ \ell_1 c_1 + \ell_2 c_{12} + \ell_3 c_{123} \\ 0 \end{bmatrix} \quad z_1 \wedge (p - p_1) = \begin{bmatrix} -\ell_2 s_{12} - \ell_3 s_{123} \\ \ell_2 c_{12} + \ell_3 c_{123} \\ 0 \end{bmatrix}$$

$$z_2 \wedge (p - p_2) = \begin{bmatrix} -\ell_3 s_{123} \\ \ell_3 c_{123} \\ 0 \end{bmatrix}$$

↓

$$J(q) = \begin{bmatrix} -\ell_1 s_1 - \ell_2 s_{12} - \ell_3 s_{123} & -\ell_2 s_{12} - \ell_3 s_{123} & -\ell_3 s_{123} \\ \ell_1 c_1 + \ell_2 c_{12} + \ell_3 c_{123} & \ell_2 c_{12} + \ell_3 c_{123} & \ell_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

In questo esempio possiamo già vedere l'esempio di un robot **sottoattuato**: a causa di tutti quegli 0 nella Jacobiana non possiamo controllare in alcun modo $\dot{z}, \omega_y, \omega_z$. In realtà in questo caso il risultato era atteso visto che abbiamo analizzato un manipolatore planare (2D) in 3D. In questo caso è possibile utilizzare una rappresentazione ridotta di $J(q)$, levando le righe nulle (ovviamente bisogna ricordarci che è la versione ridotta).

3.2 Jacobiano analitico

Lo Jacobiano analitico viene calcolato dall'applicazione della sua stessa definizione, derivando rispetto al tempo il vettore $\dot{\mathbf{x}}$ ottenuto dalla cinematica diretta delle posizioni:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \underbrace{\frac{\partial \mathbf{x}}{\partial \mathbf{q}}}_{\mathbf{J}_A(\mathbf{q})} \underbrace{\frac{d\mathbf{q}}{dt}}_{\dot{\mathbf{q}}} = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}$$

E $\partial \mathbf{x} / \partial \mathbf{q}$ (che è proprio lo Jacobiano analitico) non è altro che lo Jacobiano (nel senso inteso ad analisi):

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Conversione $\mathbf{J}(\mathbf{q}) \leftrightarrow \mathbf{J}_A(\mathbf{q})$

Data la rappresentazione minima prescelta per l'assetto (Eulero, RPY) è possibile esprimere $\boldsymbol{\omega}$ in funzione di $\dot{\boldsymbol{\phi}}$ come:

$$\boldsymbol{\omega} = \mathbf{T}(\boldsymbol{\phi})\dot{\boldsymbol{\phi}}$$

e quindi $\mathbf{J}(\mathbf{q})$ in funzione di $\mathbf{J}_A(\mathbf{q})$:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\phi}) \end{bmatrix} = \mathbf{J}_A(\mathbf{q})$$

(la forma di $\mathbf{T}(\boldsymbol{\phi})$ dipende dalla rappresentazione scelta per l'assetto).

I valori di $\boldsymbol{\phi}$ per cui il determinante della matrice $\mathbf{T}(\boldsymbol{\phi})$ si annulla nei vari casi corrispondono a singolarità della rappresentazione dell'assetto usata (**gimbal lock**). Questo significa che ci sono velocità angolari che non possono essere espresse per mezzo di $\dot{\boldsymbol{\phi}}$.

Esempio

Se prendiamo gli angoli di Eulero ZYZ:

$$\boldsymbol{\omega} = \begin{bmatrix} 0 & -s_\phi & c_\phi s_\theta \\ 0 & c_\phi & s_\phi s_\theta \\ 1 & 0 & c_\theta \end{bmatrix} \dot{\boldsymbol{\phi}} = \mathbf{T}(\boldsymbol{\phi})\dot{\boldsymbol{\phi}}$$

3.3 Singolarità cinematiche

Le configurazioni del manipolatore per cui lo Jacobiano (geometrico) non è di rango pieno sono dette singolarità cinematiche o configurazioni singolari. Ovvero:

$$\det(\mathbf{J}(\mathbf{q}_s)) = 0 \implies \mathbf{q}_s \text{ è una configurazione singolare}$$

Le singolarità corrispondono a configurazioni in cui la mobilità del manipolatore è ridotta:

- Esistono velocità cartesiane che non possono essere ottenute da velocità ai giunti
- Velocità non nulle ai giunti generano velocità cartesiane nulle

Inoltre:

- Quando il manipolatore è in configurazione singolare, il problema cinematico inverso può presentare infinite soluzioni
- In un intorno della singolarità, a piccole velocità della punta possono corrispondere grandi velocità dei giunti (poiché $v = J\dot{q} \implies \dot{q} = J^{-1}v \implies J^{-1} = \frac{1}{\det(J)}\text{adj}(J)$ e quindi se $\det(J) \rightarrow 0$ abbiamo $J^{-1} \rightarrow \infty$)

Le singolarità cinematiche possono essere:

- **"di confine"**: i bracci del manipolatore sono completamente stesi o retratti, in una configurazione limite ("di confine") dello spazio di lavoro
- **interne**: si trovano all'interno dello spazio di lavoro e corrispondono tipicamente all'allineamento di due o più assi di movimento dei giunti oppure a particolari configurazioni

Le configurazioni singolari interne sono le più critiche dal punto di vista pratico perché possono essere raggiunte ovunque all'interno dello spazio di lavoro durante l'esecuzione di una traiettoria pianificata nello spazio operazionale.

3.3.1 Manipolatori con polso sferico

Per i manipolatori con polso sferico le singolarità possono essere disaccoppiate in spalla e polso:

- Per il **polso**, abbiamo singolarità quando 2 dei 3 assi si allineano, in particolare (con riferimento alla figura) z_3 e z_5 . In generale, qualsiasi allineamento di due o più giunti rotanti porta ad una singolarità, poiché due rotazioni opposte di uguale entità non producono movimento nell'end-effector

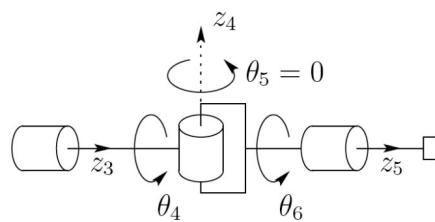


Figura 3.2: Singolarità per $\theta_5 = 0, \pi$

- Per la **spalla**, invece, possiamo analizzare le singolarità guardando il determinante di \mathbf{J}_p , che risulta nullo per $\theta_3 = 0, \pi$ (singolarità "di confine"):

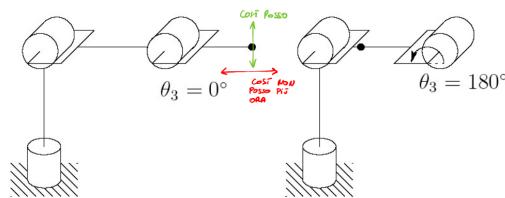


Figura 3.3: Singolarità per $\theta_3 = 0, \pi$

oppure per $a_2c_2 + a_3c_{23} = 0$ (singolarità interna):

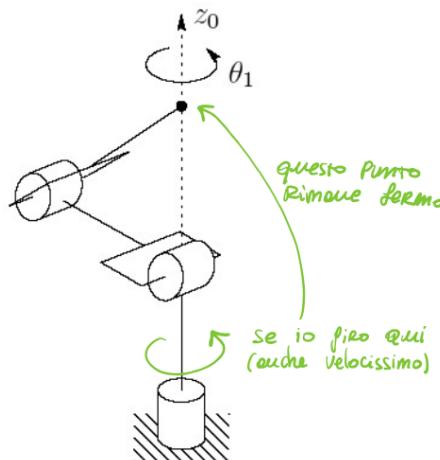


Figura 3.4: Singolarità per $a_2c_2 + a_3c_{23} = 0$

3.4 Ridondanza cinematica

Un manipolatore è ridondante dal punto di vista cinematico se possiede un **numero di gradi di libertà maggiore del numero di variabili necessarie** per descrivere il compito che deve svolgere.

Un manipolatore è **intrinsecamente ridondante** se la dimensione m del suo spazio operazionale è inferiore alla dimensione n del suo spazio dei giunti. In generale, però, la ridondanza è un concetto relativo al compito assegnato al manipolatore (*e.g. anche nel caso $m = n$ un manipolatore può risultare ridondante, come nel caso se per operare gli bastasse solo x e y .*).

Visto che $v = J(q)\dot{q}$ è un sistema lineare ($y = Ax$), possiamo analizzarlo utilizzando gli strumenti noti dell'algebra. Ricordiamo i seguenti concetti:

- $\text{Im}(A) \equiv \text{span}(A) \equiv \mathcal{R}(A)$
- $\text{Ker}(A) \equiv \mathcal{N}(A) \triangleq \{x : Ax = 0\}$
- A square and $\det(A) \neq 0 \iff \text{Ker}(A) = \{0\}$
- A rectangular $\implies \text{Ker}(A) \neq \{0\}$

Se un manipolatore è ridondante ($m > n$) abbiamo una matrice J rettangolare:

$$\mathbf{J}(\mathbf{q}) = \left[\quad \right] = \text{rettangolare} \implies \text{Ker}(A) \neq \{0\}$$

quindi, visto che il kernel non è banale, da definizione avremo:

$$\exists \mathbf{q} : \mathbf{J}(\mathbf{q}) = 0$$

ovvero **esistono velocità dei giunti non nulle che non generano velocità nell'end-effector.**

Da questa osservazione possiamo dire che:

- Il **range space** $\mathcal{R}(\mathbf{J}(\mathbf{q})) \in \mathbb{R}^m$ è quel sottospazio delle velocità dell'end-effector che possono essere generate da velocità ai giunti (per la postura corrente \mathbf{q}).
- Il **null space** $\mathcal{N}(\mathbf{J}(\mathbf{q})) \in \mathbb{R}^n$ è quel sottospazio delle velocità ai giunti che non producono alcuna velocità nell'end effector (per la postura corrente \mathbf{q}).

Altre cose dette da Rizzo

Supponiamo $\dot{\mathbf{q}}^* \in \mathcal{N}(\mathbf{J}(\mathbf{q})) \implies \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}^* = 0$. Allora, se poniamo $\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ abbiamo:

$$\mathbf{v} = \mathbf{J}(\mathbf{q})(\dot{\mathbf{q}} + \dot{\mathbf{q}}^*) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}^* = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

Ora poniamo $\langle \dot{q}_1, \dot{q}_2, \dot{q}_3 \rangle$ base di $\mathcal{N}(\mathbf{J}(\mathbf{q}))$. Grazie a questa terna, richiamando che ovviamente ora qualsiasi vettore ($\in \mathcal{N}(\mathbf{J}(\mathbf{q}))$) può essere scritto come $\alpha\dot{q}_1 + \beta\dot{q}_2 + \gamma\dot{q}_3$, possiamo definire:

$$P \triangleq [\dot{q}_1, \dot{q}_2, \dot{q}_3] = \text{matrice generatrice del null space}$$

e quindi, qualsiasi $\dot{\mathbf{q}}^* \in \mathcal{N}(\mathbf{J}(\mathbf{q}))$ può essere scritto come:

$$\dot{\mathbf{q}}^* = P\dot{q}_a \quad , \quad \dot{q}_a \text{ arbitrario}$$

3.5 Cinematica inversa delle velocità

Il problema della cinematica inversa delle velocità consiste nel calcolo di $\dot{\mathbf{q}}$ partire dalla conoscenza di \mathbf{v} .

Se lo Jacobiano $\mathbf{J}(\mathbf{q})$ è **quadrato** e **full-rank**, il problema viene risolto semplicemente dall'equazione:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\mathbf{v} \tag{3.4}$$

Se, invece, siamo nel caso **rettangolare** e **non full-rank** (come nel caso di un manipolatore **ridondante**), dobbiamo trovare altri modi, ad esempio tramite la **right pseudo-inverse**:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q})\mathbf{v} \quad \text{dove} \quad \mathbf{J}^\dagger \triangleq \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1} \tag{3.5}$$

3.5.1 Metodologie avanzate per il caso rettangolare/ridondante

Soluzioni diverse possono essere ottenute ottimizzando la ricerca di $\dot{\mathbf{q}}$ con l'introduzione di opportuni funzionali di costo e di vincoli secondari, tali da permettere di sfruttare la ridondanza del manipolatore nell'esecuzione del compito.

Ad esempio:

$$\begin{aligned} & \text{minimize} \quad g(\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \\ & \text{subject to} \quad \mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \end{aligned}$$

dove $\mathbf{W} \in \mathbb{R}^{n \times n}$ è una matrice di pesi, simmetrica e definita positiva.

Tramite il metodo dei moltiplicatori di Lagrange (che ci permettono di includere il "subject

to" dentro la funzione da minimizzare) possiamo ottenere una soluzione in forma chiusa per il problema dato:

$$g(\dot{\mathbf{q}}, \lambda) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} + \lambda^T (\mathbf{v} - \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}})$$

$$\begin{aligned} \left(\frac{\partial g(\dot{\mathbf{q}}, \lambda)}{\partial \dot{\mathbf{q}}} \right)^T &= 0 \Rightarrow \dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T \lambda && \text{isolate } \lambda \\ \left(\frac{\partial g(\dot{\mathbf{q}}, \lambda)}{\partial \lambda} \right)^T &= 0 \Rightarrow \mathbf{v} = \mathbf{J} \dot{\mathbf{q}} && \\ \left\{ \begin{array}{l} \dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T \lambda \\ \mathbf{v} = \mathbf{J} \dot{\mathbf{q}} \end{array} \right. \Rightarrow \mathbf{v} = \mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \lambda && \longrightarrow && \boxed{\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \mathbf{v}} \\ && && \text{Plugging (1) in } \dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T \lambda \\ && && \text{Solution} \end{aligned}$$

Dove possiamo notare che ponendo $\mathbf{W} = \mathbf{I}$ questa soluzione è equivalente a quella con la pseudo-inversa destra.

Visto che lo Jacobiano rettangolare è relativo anche al caso di **manipolatori ridondanti**, nella nostra soluzione dovremmo tenere in considerazione le velocità non nulle dei giunti che non producono velocità dell'end-effector: riprendendo la soluzione (3.5), possiamo aggiungervi un qualsiasi vettore del null-space e rimarrà comunque valida:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q}) \mathbf{v} + \dot{\mathbf{q}}_n \quad \text{dove} \quad \dot{\mathbf{q}}_n \in \mathcal{N}(\mathbf{J}(\mathbf{q})) \quad (*)$$

visto che l'ultimo termine è t.c. $\mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}_n = 0$:

$$\mathbf{J}(\mathbf{q}) = \mathbf{J}(\mathbf{q})(\mathbf{J}(\mathbf{q})^\dagger \mathbf{v} + \dot{\mathbf{q}}_n) = \cancel{(\mathbf{J}(\mathbf{q})(\mathbf{J}(\mathbf{q})^\dagger)}^I \mathbf{v} + \cancel{\mathbf{J}(\mathbf{q})} \dot{\mathbf{q}}_n^0 = \mathbf{v}$$

come scegliamo il vettore $\dot{\mathbf{q}}_n$? Possiamo usare un "trucco" accennato in precedenza: il *proiettore* nel null space (prima chiamato matrice generatrice):

$$\mathbf{J}^* = \mathbf{I} - \mathbf{J}^\dagger \mathbf{J}$$

questa matrice "proietta" un qualsiasi vettore nel kernel, visto che $\mathbf{J}(\mathbf{J}^* \dot{\mathbf{q}}) = 0$, di conseguenza possiamo riscrivere (*) come:

$$\boxed{\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q}) \mathbf{v} + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_d} \quad (3.6)$$

che corrisponde alla soluzione in forma chiusa del problema:

$$\begin{aligned} \text{minimize} \quad g'(\dot{\mathbf{q}}) &= \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_d)^T (\dot{\mathbf{q}} - \dot{\mathbf{q}}_d) \\ \text{subject to} \quad \mathbf{v} &= \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \end{aligned}$$

dove $\dot{\mathbf{q}}_d$ è un vettore di **velocità desiderate**: ovvero vogliamo $\dot{\mathbf{q}} \rightarrow \dot{\mathbf{q}}_d$.

Da quest'ultima soluzione possiamo un'importante osservazione: **sfruttando i gradi di libertà in più, forniti dal manipolatore ridondante, riusciamo ad imporre velocità desiderate a dei giunti (e.g. se voglio che il base joint vada molto lento, grazie alla ridondanza posso farlo).**

3.5.2 Raggiungimento di obiettivi secondari

Invece di impostare delle velocità desiderate per alcuni giunti, possiamo utilizzare i DoF "extra" per raggiungere altri obiettivi secondari.

Poniamo:

$$\dot{\mathbf{H}} = \frac{d\mathbf{H}}{dt} = \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \mathbf{J}^\dagger(\mathbf{q}) \mathbf{v} + \frac{\partial \mathbf{H}}{\partial \mathbf{q}} (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_d$$

Ora vogliamo minimizzare \mathbf{H} , quindi dobbiamo far sì che $\dot{\mathbf{H}} < 0$. Scegliamo:

$$\dot{\mathbf{q}}_d = -K \left(\frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right)^T, \quad K > 0$$

così che:

$$\dot{\mathbf{H}} = \underbrace{\frac{\partial \mathbf{H}}{\partial \mathbf{q}} \mathbf{J}^\dagger(\mathbf{q}) \mathbf{v}}_{\text{non si sa}} + \underbrace{-K \frac{\partial \mathbf{H}}{\partial \mathbf{q}} (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \left(\frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right)^T}_{<0}$$

Variando \mathbf{H} possiamo scegliere il nostro obiettivo secondario, di seguito alcune scelte tipiche:

- **Massimizza distanza da ostacoli:**

- $H = \min_{p,o} \|p(q) - o\| \quad \rightarrow \text{massimizza } H$
- p : generico punto del manipolatore
- o : generico punto di un ostacolo

- **Massimizza "distanza" dai limiti dei giunti:**

- $H(q) = -\frac{1}{2n} \sum_{i=1}^n \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \quad \rightarrow \text{minimizza } H$
- \bar{q}_i : midpoint of the joint variable
- q_{iM}, q_{im} : max and min excursion of the joint variable

- **Massimizza "distanza" dalle singolarità:**

- $H(q) = \sqrt{\det(J(q)J^T(q))} \quad \rightarrow \text{massimizza } H$
- In questo caso H è chiamata misura di *manipolabilità*

3.5.3 Inversione dello Jacobiano nel caso di singolarità

Nel caso di singolarità abbiamo \mathbf{J} non full-rank, di conseguenza invertire la matrice diventa più complesso. Ad esempio, in questo caso neanche la formula con la pseudo-inversa di (3.5) è applicabile, visto che anche $J J^T$ è singolare e quindi non si può invertire ($\nexists (J J^T)^{-1}$). Dobbiamo quindi trovare un modo alternativo per invertire lo Jacobiano.

Invece di utilizzare la pseudo-inversa, possiamo optare per un trade-off fra accuratezza e relizzabilità attraverso l'utilizzo del **damped-least-square**:

$$\mathbf{J}^* = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T + k^2 \mathbf{I})^{-1} \tag{3.7}$$

Dim.

Partiamo dalla normale pseudo-inversa destra $J^\dagger = J^T(JJ^T)^{-1}$, quest'ultima può essere vista calcolata anche tramite la SVD di J :

$$J^\dagger = V\Sigma^+U^T$$

dove

$$\Sigma^+ = \begin{bmatrix} \sigma_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \sigma_m^{-1} \end{bmatrix}^T$$

Il problema è che, nel caso di singolarità $1/q_i \rightarrow \infty$ (visto che $q_i \rightarrow 0$). Ora invece passiamo alla forma (3.7), e calcoliamo anche questa tramite la SVD: l'unica differenza è nella matrice

$$\Sigma^+ = \begin{bmatrix} \frac{1}{\sigma_1+k^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_m+k^2} \end{bmatrix}^T$$

l'aggiunta del coefficiente k^2 permette di avere valori $\neq \infty$ anche quando $q_i \rightarrow 0$.

□

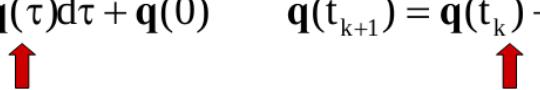
3.6 Cinematica inversa delle posizioni

3.6.1 Cinematica inversa tramite Jacobiano inverso

Se la postura iniziale $\mathbf{q}(0)$ del robot è nota, la cinematica inversa delle velocità può essere utilizzata per risolvere indirettamente il problema della cinematica inversa delle posizioni, calcolando $\mathbf{q}(t)$ per integrazione

$$v = J(\mathbf{q})\dot{\mathbf{q}} \implies \dot{\mathbf{q}} = J(\mathbf{q})^{-1}v \implies \mathbf{q} = \int \dot{\mathbf{q}} = \int J(\mathbf{q})^{-1}v$$

ovvero:

$$\mathbf{q}(t) = \int_0^t \dot{\mathbf{q}}(\tau)d\tau + \mathbf{q}(0) \quad \text{a tempo continuo} \quad \mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \dot{\mathbf{q}}(t_k)\Delta t \quad \text{a tempo discreto}$$


e inserendo la forma di \mathbf{q} ottenuta tramite lo Jacobiano (geometrico):

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \mathbf{J}^{-1}(\mathbf{q}(t_k))\mathbf{v}(t_k)\Delta t$$

Il problema con questa soluzione *naïve* è che a causa di errori numerici divergeremo dalla soluzione voluta:

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x} \tag{3.8}$$

Per risolvere il problema possiamo andare ad analizzare la dinamica d'errore:

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \quad (\star)$$

e "chiudere il loop" per avere un feedback e correggere questo drift (e quindi far sì che $\boldsymbol{e} \rightarrow 0$). Da qui, per semplicità supponendo che \boldsymbol{J}_A sia invertibile, possiamo porre:

$$\dot{\boldsymbol{q}} \triangleq \boldsymbol{J}_A^{-1}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\boldsymbol{e})$$

e, sostituendo in (\star) , otteniamo la seguente dinamica d'errore:

$$\begin{aligned} \dot{\boldsymbol{e}} &= \dot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})(\boldsymbol{J}_A^{-1}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\boldsymbol{e})) \\ &\Downarrow \\ \dot{\boldsymbol{e}} &= -\boldsymbol{K}\boldsymbol{e} \\ &\Downarrow \\ \dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} &= 0 \end{aligned}$$

dove se $\boldsymbol{K} > 0$ abbiamo stabilità asintotica dell'errore ($\implies \boldsymbol{x} \rightarrow \boldsymbol{x}_d$). Questa strategia può essere utilizzata per implementare il controllo cinematico, e funziona per basse velocità.

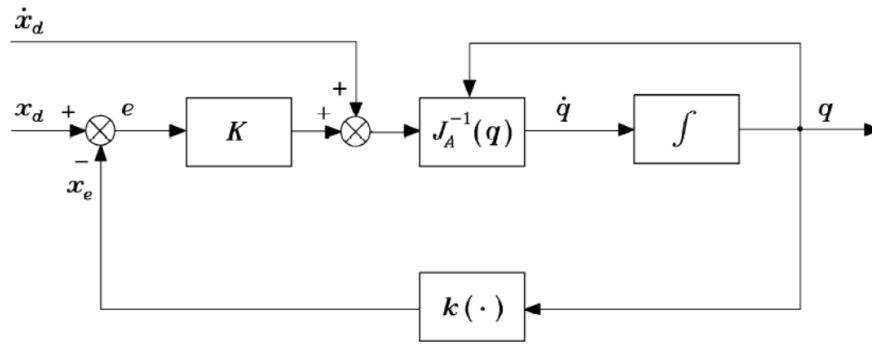


Figura 3.5: Circuito di controllo con Jacobiano inverso e feedback

3.6.2 Cinematica inversa tramite Jacobiano trasposto

Nella soluzione tramite Jacobiano inverso stiamo in pratica usando una sorta di feedback linearization per trasformare la dinamica d'errore (non-lineare) in un'equazione lineare che possiamo facilmente controllare. Lo svantaggio principale di questa tecnica, è che richiede l'inversione di $\boldsymbol{J}(\boldsymbol{q})$, che (soprattutto nel caso di singolarità o ridondanza) è computazionalmente molto pesante.

Potremmo quindi pensare di progettare un "controllore" per l'errore direttamente nel caso non-lineare, ad esempio tramite il **metodo diretto di Lyapunov**. Sceglieremo

$$V(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}^T \boldsymbol{K} \boldsymbol{e} > 0 \quad \forall \boldsymbol{e} \neq \mathbf{0} \quad , \quad V(\mathbf{0}) = 0$$

con $\boldsymbol{K} > 0$. Derivando e richiamando (3.8), otteniamo:

$$\dot{V} = \boldsymbol{e}^T \boldsymbol{K} \dot{\boldsymbol{x}}_d - \boldsymbol{e}^T \boldsymbol{K} \dot{\boldsymbol{x}}_e = \boldsymbol{e}^T \boldsymbol{K} \dot{\boldsymbol{x}}_d - \boldsymbol{e}^T \boldsymbol{K} \underbrace{\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}}_{\dot{\boldsymbol{x}}}$$

e ponendo

$$\dot{\boldsymbol{q}} \triangleq \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e}$$

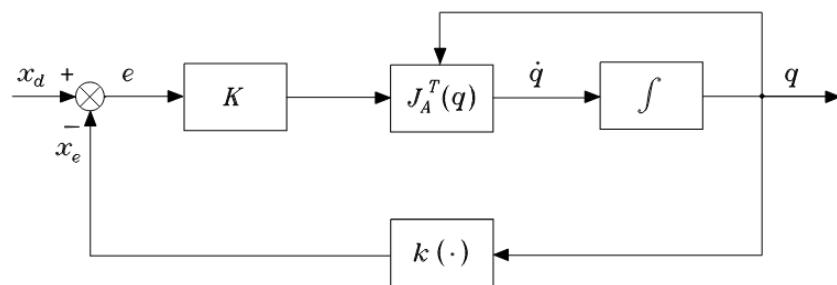


Figura 3.6: Cinematica inversa con Jacobiano trasposto (riferimento costante)

otteniamo $\dot{V} < 0$ nel caso di riferimenti costanti ($\dot{\mathbf{x}}_d = \text{const}$ i.e. $\ddot{\mathbf{x}}_d = 0$).

Per il caso di \mathbf{x}_d tempo-variante ($\dot{\mathbf{x}}_d \neq 0$), non riusciamo a garantire la convergenza dell'errore, che possiamo dire essere solo norm-bounded e inversamente proporzionale a \mathbf{K} . Per avere convergenza di tale errore dovremmo nuovamente scegliere $\dot{\mathbf{q}}$ dipendente dallo (pseudo-)inverso dello Jacobiano (quindi ritornando al caso di sez. 3.6.1)

Capitolo 4

Statica

Obiettivo della statica è la **determinazione delle relazioni fra le forze e le coppie applicate sulla punta operativa e le forze e le coppie applicate ai giunti**, quando il manipolatore si trova in una **configurazione di equilibrio** (i.e. quando tutto è fermo).

Chiamiamo **forze generalizzate** il vettore composto di forze e coppie (torques) assieme (tecnicamente questo non sarebbe un vettore perché gli elementi hanno unità di misure diverse: forze in N , coppie in Nm).

Indicheremo con

$$\mathbf{F} \triangleq \begin{bmatrix} \mathbf{f}(t) \\ \mathbf{N}(t) \end{bmatrix} \quad \mathbf{f}, \mathbf{N} \in \mathbb{R}^3$$

(f forze, N momenti) il vettore delle **forze generalizzate cartesiane** applicate sulla punta operativa (task space generalized forces (**TSGF**)), mentre con

$$\boldsymbol{\tau}(t) \triangleq \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix}$$

il vettore delle **forze generalizzate ai giunti** (joint generalized forces (**JGF**)); visto che i giunti maggiormente impiegati sono di tipo rotoidale, $\boldsymbol{\tau}(t)$ è spesso indicato come vettore delle **coppie generalizzate**.

- Per **giunti prismatici**:

$$\tau_i = \mathbf{k}_{i-1}^T \mathbf{f}_{i-1,i}$$

- Per **giunti rotoidali**:

$$\tau_i = \mathbf{k}_{i-1}^T \mathbf{N}_{i-1,i}$$

Le TSGF sono generate dalle interazioni della punta con l'ambiente (e.g. quando il TCP preme contro una superficie), mentre le JGF sono generate dalla potenza sviluppata dai motori.

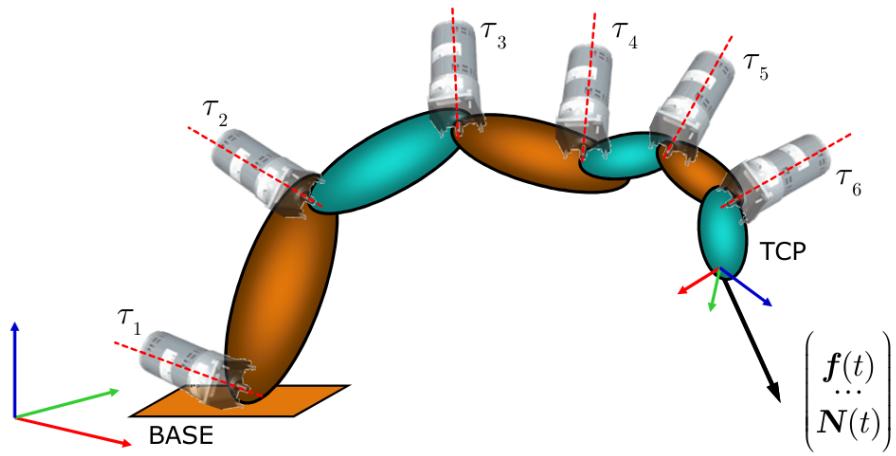


Figura 4.1: Statics

4.1 Relazione fra F e τ

Per trovare questa relazione possiamo usare il **principio dei lavori virtuali**:

- Il **lavoro** è una grandezza reale data dal prodotto di una forza per uno spostamento lineare oppure dal prodotto di un momento per uno spostamento angolare
- Il **lavoro virtuale** è quello associato agli spostamenti virtuali, dati da tutti i possibili spostamenti geometrici compatibili con gli eventuali vincoli sul sistema

Il lavoro virtuale nasce dall'applicazione del **principio di minima azione** allo studio delle forze e del movimento di un sistema meccanico. Il lavoro di una forza che agisce su una particella mentre si muove lungo uno spostamento è diverso a seconda degli spostamenti. Tra tutti i possibili spostamenti che una particella può seguire, detti **spostamenti virtuali**, quello ne minimizzerà l'azione. Questo spostamento è quindi lo spostamento seguito dalla particella secondo il principio di minima azione. Il lavoro di una forza su una particella lungo uno spostamento virtuale è noto come lavoro virtuale.

Sia

$$\delta W_{TCP} = \mathbf{F}^T \delta \mathbf{p}$$

il lavoro virtuale associato alle forze generalizzate cartesiane, dove $\delta \mathbf{p}$ è lo spostamento virtuale cartesiano (formato dagli spostamenti virtuali lineare ed angolare).

$$\rightarrow WORK = \mathbf{F} \cdot \Delta \mathbf{P}$$

Sia anche

$$\delta W_g = \boldsymbol{\tau}^T \delta \mathbf{q}$$

il lavoro virtuale associato alle forze generalizzate ai giunti, ove $\delta \mathbf{q}$ è lo spostamento virtuale ai giunti. Secondo il **principio dei lavori virtuali**, il manipolatore è in condizioni di equilibrio statico se e solo se i lavori virtuali delle forze generalizzate cartesiane e ai giunti sono uguali, qualunque sia la configurazione del robot:

$$\delta W_g = \delta W_{TCP} \iff \boldsymbol{\tau}^T \delta \mathbf{q} = \mathbf{F}^T \delta \mathbf{p} \quad \forall \mathbf{q}(t)$$

Ora, visto che il nostro manipolatore è rappresentato da un sistema tempo-invariante (la configurazione dipende da $\mathbf{q}(t)$, ma non esplicitamente dal tempo), possiamo dire che gli spostamenti virtuali coincidono con gli spostamenti elementari:

$$\delta\mathbf{p} = d\mathbf{p}, \quad \delta\mathbf{q} = d\mathbf{q}$$

Richiamando $v = J(\mathbf{q})\dot{\mathbf{q}}$ e che $v = dp/dt$, abbiamo:

$$\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \iff \frac{d\mathbf{p}}{dt} = \mathbf{J}(\mathbf{q})\frac{d\mathbf{q}}{dt} \iff d\mathbf{p} = \mathbf{J}(\mathbf{q})d\mathbf{q}$$

dal principio dei lavori virtuali però abbiamo che $\mathbf{F}^T d\mathbf{p} = \boldsymbol{\tau}^T d\mathbf{q}$, e quindi:

$$d\mathbf{p} = \mathbf{J}(\mathbf{q})d\mathbf{q} \iff \boldsymbol{\tau} d\mathbf{q} = \mathbf{F}^T \mathbf{J}(\mathbf{q})d\mathbf{q} \iff \boldsymbol{\tau}^T = \mathbf{F}^T \mathbf{J} \iff \boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}$$

Quindi, le forze ai giunti necessarie per bilanciare le forze sul TCP sono date da:

$$\boxed{\boldsymbol{\tau} = -\mathbf{J}^T(\mathbf{q})\mathbf{F}} \quad (4.1)$$

che è proprio la relazione statica cercata.

4.2 Dualità cineto-statica

In algebra lineare,

$$y = A^T x \quad \text{è il problema duale di} \quad y = Ax$$

quindi, dalla seguente osservazione:

$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{F}$ $\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{v}$	
INSIDE of the Robot	OUTSIDE of the Robot

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{F}$$

$$\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

NOTICE the similarity

notiamo che possiamo vedere la relazione statica come il duale del problema della cinematica diretta delle velocità.

Prendiamo

$$\mathbf{F}^* \in \mathcal{N}(\mathbf{J}^T) \quad s.t. \quad \mathbf{J}^T \mathbf{F}^* = 0$$

dalla relazione statica possiamo dire che \mathbf{F}^* è un vettore di forze (applicate al TCP) che non richiede alcuna compensazione dai motori: un esempio può essere visto in fig. 4.2, dove abbiamo il braccio di un manipolatore a "penzoloni"; in questo caso abbiamo la forza di gravità che "tira" in giù, ma non sono necessarie alcune compensazioni da parte dei motori.

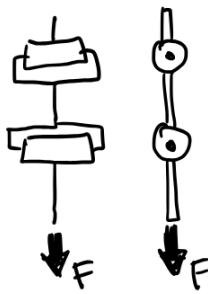


Figura 4.2: Esempio

Detto questo possiamo ora parlare più in dettaglio della relazione matematica che esiste fra τ e v . La dualità può essere caratterizzata dal range ed il kernel di J e J^T :

- $\mathcal{R}(J)$: velocità del TCP che possono essere raggiunte per una data posa
- $\mathcal{N}(J)$: velocità dei giunti che non producono alcuna velocità nel TCP (per la data posa)
- $\mathcal{R}(J^T)$: JFG (τ) che riscono a bilanciare le forze generalizzate applicate al TCP (F), per la posa corrente
- $\mathcal{N}(J^T)$: forze generalizzate applicate al TCP (F) che non richiedono compensazione di JFG/coppie (τ) (per la data posa)

Da algebra lineare, ricordiamo che:

$$\mathcal{N}(J) \equiv \mathcal{R}^\perp(J^T) \quad \mathcal{R}(J) \equiv \mathcal{N}^\perp(J^T)$$

Da qui segue che, **quando il manipolatore è in configurazione singolare**:

- Esistono velocità non nulle ai giunti $\dot{q} \in \mathcal{N}(J)$ che producono velocità nulle della punta e **coppie generalizzate ai giunti** $\tau \in \mathcal{N}(J) = \mathcal{R}^\perp(J^T)$ che non possono essere compensate da alcuna forza generalizzata sulla punta
- **Esistono forze generalizzate** $F \in \mathcal{N}(J^T)$ che non richiedono alcuna coppia ai giunti per essere bilanciata e velocità della punta operativa $v = \mathcal{N}(J^T) = \mathcal{R}^\perp(J)$ che non possono essere ottenute da velocità ai giunti

4.3 Ellissoidi di manipolabilità

4.3.1 Velocità

Spesso, per sistemi lineari SISO, se ne studia il comportamento tramite la cosiddetta risposta all'impulso, che essenzialmente caratterizza il modo in cui il sistema risponde a un'unità di ingresso. Nel caso MIMO, invece (come per un manipolatore), il concetto analogo è quello di caratterizzare l'output in termini di un input con norma unitaria. Consideriamo quindi l'insieme di tutte le velocità dei giunti del robot \dot{q} tali che:

$$\|\dot{q}\|^2 = \dot{q}_1^2 + \cdots + \dot{q}_n^2 \leq 1 \iff \dot{q}^T \dot{q} = 1$$

usando $\dot{q} = J^\dagger v$, si ottiene:

$$\dot{q}^T \dot{q} = (J^\dagger v)^T (J^\dagger v) = v^T (J J^T)^{-1} v$$

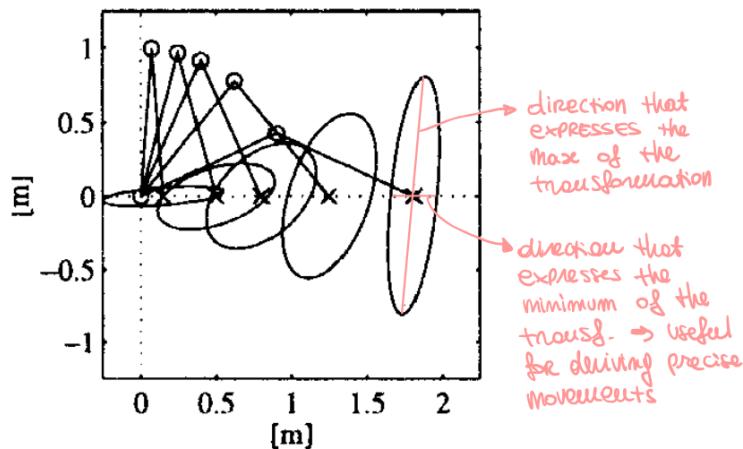


Figura 4.3: Ellissoidi di manipolabilità al variare della postura per un manipolatore con 2 giunti

possiamo quindi definire il **velocity manipulability ellipsoid** come quei punti che soddisfano:

$$E_v = \{\boldsymbol{v} : \boldsymbol{v}^T (\boldsymbol{J}\boldsymbol{J}^T)^{-1} \boldsymbol{v} = 1\}$$

notare che varia con il variare di \boldsymbol{q} . Da qui possiamo anche definire un'altro valore utile, chiamato **manipulability measure**:

$$w(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)} = |\lambda_1 \lambda_2 \cdots \lambda_n| = |\det(\boldsymbol{J})|$$

che non è altro che il volume dell'ellissoide definito prima (λ_i sono gli autovalori di \boldsymbol{J}).

Un'interessante osservazione può essere fatta per le **singolarità**: quando il manipolatore approssima una postura singolare abbiamo che $\lambda_i \rightarrow 0$ e quindi anche $w(\boldsymbol{q}) \rightarrow 0$ (non abbiamo manipolabilità!). Per questo motivo $w(\boldsymbol{q})$ è spesso usata come misura di "distanza" dalle configurazioni singolari.

4.3.2 Forza

Dalla dualità cineto-statica possiamo derivare un ellissoide di manipolabilità anche per le forze. Questa volta invece delle velocità dei giunti andiamo ad utilizzare le coppie generalizzate ai giunti:

$$\boldsymbol{\tau}^T \boldsymbol{\tau} = 1$$

allora, il **force manipulability ellipsoid** è definito come:

$$E_F = \{\boldsymbol{F} : \boldsymbol{F}^T (\boldsymbol{J}\boldsymbol{J}^T) \boldsymbol{F} = 1\}$$

dove abbiamo utilizzato $\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{F}$.

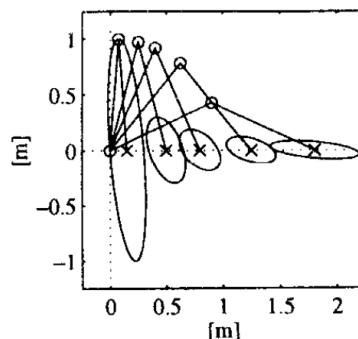


Figura 4.4: Stesso esempio di prima ma per la forza

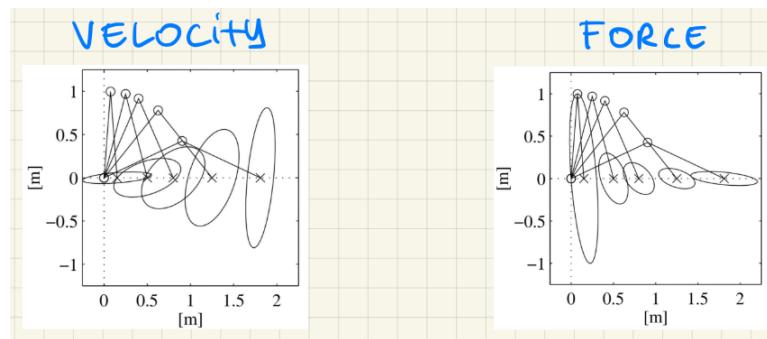


Figura 4.5: Comparazione: notare l'"inversione" fra i due

Comparando la definizione di E_F con quella di E_v possiamo notare che in questo caso il contenuto fra le parentesi (JJ^T) non è invertito (mentre prima si): da qui consegue l'importante fatto che gli assi principali dell'ellissoide di manipolabilità della forza coincidono con quelli della velocità, ma con valori "inversi".

Da questo consegue un risultato anche intuitivo: **una direzione dove abbiamo una buona manipolabilità in velocità avrà cattiva manipolabilità di forza, e viceversa → veloce e debole oppure lento e forte** (*pensa al cambio di una bici/auto: in salita vado lento ma faccio tanta forza, in piano faccio meno forza ma vado più veloce*).

Capitolo 5

Dinamica

Passiamo ora a studiare la dinamica, ovvero le relazioni fra forze e movimenti.

Il modello dinamico del manipolatore può essere determinato seguendo due diversi approcci: equazioni di Lagrange oppure equazioni di Newton-Eulero. Dal punto di vista della derivazione le prime sono più semplici, mentre per quanto riguarda la computazione le seconde permettono algoritmi ricorsivi efficienti. Noi vedremo solo la definizione Lagrangiana.

5.1 Lagrangiana ed equazioni di Eulero-Lagrange

La **Lagrangiana** \mathcal{L} di un sistema è data dalla differenza tra la sua energia cinetica totale \mathcal{T} e la sua energia potenziale totale \mathcal{U} :

$$\mathcal{L} = \mathcal{T} - \mathcal{U}$$

essa è funzione delle coordinate generalizzate usate per descrivere la posizione del sistema, nonché delle loro derivate rispetto al tempo. Nel caso di un manipolatore le variabili giunto qui possono essere assunte come coordinate generalizzate.

Le equazioni di Eulero-Lagrange, che descrivono il comportamento dinamico del sistema, sono date da:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_i} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = \mathcal{F}_i \quad i = 1, \dots, n \quad (5.1)$$

dove \mathcal{F}_i è la i-esima forza generalizzata, non conservativa (associata a \mathbf{q}_i).

5.1.1 Energia cinetica

L'energia cinetica totale di un manipolatore con n giunti (e $n + 1$ bracci rigidi, ma il braccio di base b_0 non dà contributo, essendo fisso) è data dalla somma delle energie cinetiche \mathcal{T}_{l_i} di ciascun braccio a cui vanno sommati i contributi \mathcal{T}_{m_i} associati agli n attuatori (motori) che muovono i giunti:

$$\mathcal{T} = \sum_{i=1}^n \mathcal{T}_{l_i} + \mathcal{T}_{m_i} \quad (5.2)$$

Energia cinetica dei bracci

Partiamo dalla seguente formulazione:

$$\mathcal{T}_{l_i} = \frac{1}{2} m_{l_i} \dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i + \frac{1}{2} \boldsymbol{\omega}_i^{Tb} \mathbf{R}_i^i \mathbf{I}_{l_i} ({ }^b \mathbf{R}_i)^T \boldsymbol{\omega}_i$$

dove le velocità (traslazionali e rotazionali) sono indicate rispetto al centro di massa del braccio, e \mathbf{I}_{l_i} rappresenta il tensore di inerzia del braccio i .

Dim.

Dalla definizione fisica:

$$\mathcal{T}_{l_i} = \frac{1}{2} \int_{V_{l_i}} \underbrace{\dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i}_{\text{velocity}^2} \underbrace{\rho dV}_{\text{mass}}$$

usando la formula della derivata di un vettore

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}}_{\ell_i} + \boldsymbol{\omega}_i \times \mathbf{r}_i = \dot{\mathbf{p}}_{\ell_i} + S(\boldsymbol{\omega}) \mathbf{r}_i$$

sostituendo alla precedente otteniamo (skippando i calcoli, vedi slide):

$$\mathcal{T}_{l_i} = \frac{1}{2} m_{l_i} \dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_{l_i} \boldsymbol{\omega}_i$$

Ora dobbiamo però notare che il tensore di inerzia di un corpo rigido, rispetto al suo centro di massa, è costante solo se espresso in un sistema di riferimento locale solidale con il corpo ed avente origine con il centro di massa stesso.

Nel caso sopra mostrato abbiamo espresso \mathbf{I}_{l_i} nel base-frame, e quindi non costante. Possiamo però utilizzare il tensore relativo al link-frame, trasformandolo le appropriate matrici:

$$\mathbf{I}_{l_i} = { }^b R_i^i \mathbf{I}_{l_i} ({ }^b R_i)^T$$

dove ${ }^b R_i$ è la matrice di rotazione del link-frame w.r.t base-frame.

Effettuando questa sostituzione abbiamo la tesi.

□

Ora, richiamando (3.3), possiamo seguire lo stesso concetto e scrivere:

$$\begin{aligned} \dot{\mathbf{p}}_{l_i} &= \mathbf{J}_{p,1}^{(l_i)} \dot{\mathbf{q}}_1 + \mathbf{J}_{p,2}^{(l_i)} \dot{\mathbf{q}}_2 + \cdots + \mathbf{J}_{p,i}^{(l_i)} \dot{\mathbf{q}}_i = \mathbf{J}_p^{(l_i)} \dot{\mathbf{q}}_i \\ \boldsymbol{\omega}_{l_i} &= \mathbf{J}_{o,1}^{(l_i)} \dot{\mathbf{q}}_1 + \mathbf{J}_{o,2}^{(l_i)} \dot{\mathbf{q}}_2 + \cdots + \mathbf{J}_{o,i}^{(l_i)} \dot{\mathbf{q}}_i = \mathbf{J}_o^{(l_i)} \dot{\mathbf{q}}_i \end{aligned}$$

dove $\mathbf{J}_{u,j}^{(l_i)}$ è la j -esima colonna di $\mathbf{J}_u^{(l_i)}$.

Queste due nuove matrici $\mathbf{J}_p^{(l_i)}, \mathbf{J}_o^{(l_i)}$ seguono un concetto analogo a quello delle colonne dello Jacobiano geometrico, però invece di far riferimento all'intero manipolatore, riguardano solo il link i :

- Se il giunto è prismatico:

$$\begin{aligned} \mathbf{J}_{p,j}^{(l_i)} &= \mathbf{k}_{j-1} \\ \mathbf{J}_{o,j}^{(l_i)} &= \mathbf{0} \end{aligned}$$

- Se il giunto è rotoidale:

$$\begin{aligned}\mathbf{J}_{p,j}^{(l_i)} &= \mathbf{k}_{j-1} \times (\mathbf{p}_{l_j} - \mathbf{p}_{j-1}) \\ \mathbf{J}_{o,j}^{(l_i)} &= \mathbf{k}_{j-1}\end{aligned}$$

per entrambi le formule valgono $\forall j = 1, \dots, i$.

Quindi, usando quest'ultime formulazioni possiamo riscrivere la forma iniziale come:

$$\mathcal{T}_{l_i} = \frac{1}{2} m_{l_i} (\dot{\mathbf{q}}^T \mathbf{J}_p^{(l_i)T}) (\mathbf{J}_p^{(l_i)} \dot{\mathbf{q}}) + \frac{1}{2} (\dot{\mathbf{q}}^T \mathbf{J}_o^{(l_i)T}) ({}^b \mathbf{R}_i {}^i \mathbf{I}_{l_i} ({}^b \mathbf{R}_i)^T) (\mathbf{J}_o^{(l_i)} \dot{\mathbf{q}})$$

Alla fin fine, però questa formulona non è altro che

$$\frac{1}{2} \dot{\mathbf{q}}^T \text{roba}(\mathbf{q}) \dot{\mathbf{q}} = \frac{1}{2} m v^2$$

ovvero l'energia cinetica.

Energia cinetica dei motori

Un ragionamento analogo può essere fatto per i motori. Inanzitutto supponiamo che il motore i -esimo che attua il giunto i sia localizzato sul braccio $i - 1$, come mostrato in fig. 5.1.

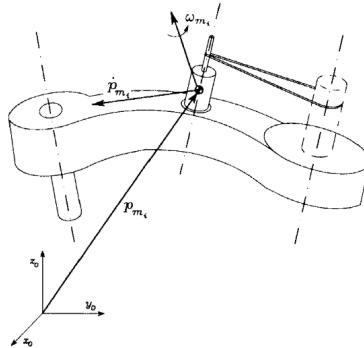


Figura 5.1: Posizionamento motore

Inoltre, per semplicità supponiamo che il contributo delle trasmissioni (gears) possa essere incluso nel motore stesso e che non ci siano movimenti indotti (i.e. il giunto i non muove altri giunti).

Quindi, analogamente ai bracci:

$$\mathcal{T}_{m_i} = \frac{1}{2} m_{m_i} \dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i + \frac{1}{2} \boldsymbol{\omega}_i^T {}^b \mathbf{R}_i {}^i \mathbf{I}_{m_i} ({}^b \mathbf{R}_i)^T \boldsymbol{\omega}_i$$

dove nuovamente le velocità sono relative al centro di massa e il tensore di inerzia è relativo al base frame. Facendo le solite sostituzioni:

$$\mathcal{T}_{m_i} = \frac{1}{2} m_{m_i} (\dot{\mathbf{q}}^T \mathbf{J}_p^{(m_i)T}) (\mathbf{J}_p^{(m_i)} \dot{\mathbf{q}}) + \frac{1}{2} (\dot{\mathbf{q}}^T \mathbf{J}_o^{(m_i)T}) ({}^b \mathbf{R}_{m_i} {}^m_i \mathbf{I}_{m_i} ({}^b \mathbf{R}_{m_i})^T) (\mathbf{J}_o^{(m_i)} \dot{\mathbf{q}})$$

dove l'unica differenza è che qui il tensore di inerzia è relativo ad un sistema di riferimento locale, solidale con il rotore (avente l'asse z diretto lungo il suo asse di rotazione), e ${}^b \mathbf{R}_{m_i}$ è la relativa matrice di rotazione.

Per la definizione di $\mathbf{J}_u^{(m_i)}$ abbiamo una differenza solo per la parte di orientamento (oltre alla ovvia sostituzione $\mathbf{p}_{l_j} \rightarrow \mathbf{p}_{m_j}$), che diventa:

$$\begin{cases} \mathbf{J}_{o,j}^{(m_i)} = \mathbf{J}_{o,j}^{(l_i)} & j = 1, \dots, i-1 \\ \mathbf{J}_{o,j}^{(m_i)} = k_{ri} \mathbf{z}_{mi} & j = i \end{cases}$$

per entrambi i tipi di giunti.

Energia cinetica totale

Ritornando all'energia cinetica totale, possiamo inserire i nostri risultati dentro (5.2) ed ottenere:

$$\mathcal{T} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij}(\mathbf{q}) \dot{\mathbf{q}}_i \dot{\mathbf{q}}_j = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} \quad (5.3)$$

dove $\mathbf{B}(\mathbf{q})$ è la matrice d'inerzia (data dalla somma dei vari termini visti prima), simmetrica, definita positiva e configuration-dependent. Infine notiamo che \mathcal{T} è funzione sia delle posizioni sia delle velocità dei giunti.

5.1.2 Energia potenziale

In maniera analoga all'energia cinetica, definiamo:

$$\mathcal{U} = \sum_{i=1}^n \mathcal{U}_{l_i} + \mathcal{U}_{m_i} \quad (5.4)$$

L'ipotesi che il manipolatore sia costituito da bracci rigidi implica l'assenza di elementi nella sua struttura che possano immagazzinare energia potenziale, di conseguenza essa è dovuta solo alla presenza della forza di gravità:

$$\begin{aligned} \mathcal{U}_{l_i} &= -m_{l_i} \mathbf{g}_0^T \mathbf{p}_{l_i} \\ \mathcal{U}_{m_i} &= -m_{m_i} \mathbf{g}_0^T \mathbf{p}_{m_i} \\ &\Downarrow \end{aligned}$$

$$\mathcal{U} = - \sum_{i=1}^n m_{l_i} \mathbf{g}_0^T \mathbf{p}_{l_i} + m_{m_i} \mathbf{g}_0^T \mathbf{p}_{m_i}$$

dove \mathbf{g}_0^T è il vettore di accelerazione gravitazionale nel base-frame (*e.g.* $\mathbf{g}_0^T = (0, 0, -g)$ se l'asse z è diretto verso l'alto).

5.2 Equazioni dinamiche del manipolatore

Essendo ormai note le espressioni di \mathcal{T} e \mathcal{U} , è possibile calcolare la funzione Lagrangiana \mathcal{L} come:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q})$$

e quindi ricavare le equazioni dinamiche del manipolatore dalle equazioni di Eulero-Lagrange (5.1). Eseguendo le derivate richieste si ottiene:

$$\sum_{j=1}^n \mathbf{B}_{ij}(\mathbf{q}) \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk}(\mathbf{q}) \dot{q}_k \dot{q}_j + g_i(\mathbf{q}) = \mathcal{F}_i \quad i = 1, \dots, n \quad (5.5)$$

dove

$$h_{ijk} = \frac{\partial B_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial B_{jk}}{\partial q_i}$$

e il termine $g_i(q)$ dovuto alla gravità risulta pari a:

$$g_i(\mathbf{q}) = \frac{\partial \mathcal{U}}{\partial q_i} = - \sum_{j=1}^n m_{l_j} \mathbf{g}_0^T \mathbf{J}_{p_i}^{(l_j)}(\mathbf{q}) + m_{m_j} \mathbf{g}_0^T \mathbf{J}_{p_i}^{(m_j)}(\mathbf{q})$$

La stessa formula può essere riscritta in forma compatta come

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{F} \quad (5.6)$$

oppure

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{q} + \mathbf{g}(\mathbf{q}) = \mathcal{F} \quad (5.7)$$

in modo che risulti nella forma di una *comoda* EDO di 2° ordine. I due nuovi termini sono definiti come:

$$\sum_{j=1}^n b_{ij}(\mathbf{q}) \ddot{q}_j + \underbrace{\sum_{j=1}^n \sum_{k=1}^n h_{ijk}(\mathbf{q}) \dot{q}_k \dot{q}_j + g_i(\mathbf{q})}_{\triangleq C(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{q}} = \xi_i$$

$\text{h}(\mathbf{q}, \dot{\mathbf{q}})$

5.2.1 Interpretazione fisica dei termini

- Termini "di accelerazione": $\sum_{j=1}^n \mathbf{B}_{ij}(\mathbf{q}) \ddot{q}_j$
 - $\mathbf{B}_{ii}(\mathbf{q})$ rappresenta il momento di inerzia equivalente sull'asse del giunto i -esimo nella configurazione \mathbf{q} , quando tutti gli altri giunti sono fermi
 - $\mathbf{B}_{ii}(\mathbf{q})$ con $i \neq j$, tiene conto degli effetti delle accelerazioni del giunto j sul giunto i
- Termini "quadratici in velocità": $\sum_{j=1}^n \sum_{k=1}^n h_{ijk}(\mathbf{q}) \dot{q}_k \dot{q}_j$
 - $h_{ijj}(\mathbf{q}) \dot{q}_j^2$ tiene conto degli effetti sul giunto i della **forza centrifuga** dovuta al moto del giunto j
 - $h_{ijk}(\mathbf{q}) \dot{q}_j \dot{q}_k$ tiene conto degli **effetti di Coriolis** indotti sul giunto i dalle velocità dei giunti j e k
- Termini "dipendenti dalla configurazione": $g_i(\mathbf{q})$
 - Rappresenta il momento generato sul giunto i dalla forza di gravità, nella configurazione \mathbf{q}

5.2.2 Forze non conservative

Per concludere ci manca solo da valorizzare il vettore delle forze generalizzate \mathcal{F} . In generale possiamo dire che:

$$\mathcal{F} = \underbrace{\boldsymbol{\tau}}_{\substack{\text{coppie attive} \\ \text{di comando}}} - \underbrace{\boldsymbol{F}_v \dot{\boldsymbol{q}}}_{\substack{\text{coppie di} \\ \text{attrito viscoso}}} - \underbrace{\boldsymbol{F}_s \text{sgn}(\dot{\boldsymbol{q}})}_{\substack{\text{coppie di} \\ \text{attrito statico}}} - \underbrace{\boldsymbol{J}^T(\boldsymbol{q}) \boldsymbol{h}}_{\substack{\text{coppie generate per bilanciare} \\ \text{forze di contatto esterne}}}$$

dove \boldsymbol{h} è il vettore delle forze generalizzate applicate dalla punta operativa sull'ambiente esterno. Sostituendo a (5.7) e riordinando i termini otteniamo:

$$\boxed{\boldsymbol{B}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \boldsymbol{q} + \boldsymbol{F}_v \dot{\boldsymbol{q}} + \boldsymbol{F}_s \text{sgn}(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q}) \boldsymbol{h}} \quad (5.8)$$

dove nuovamente abbiamo la forma di una EDO di 2° ordine: $\alpha \ddot{\boldsymbol{q}} + \beta \dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$.

Capitolo 6

Traiettorie

L'obiettivo del **trajectory planning** è quello di creare l'input di riferimento al sistema di controllo del movimento del robot.

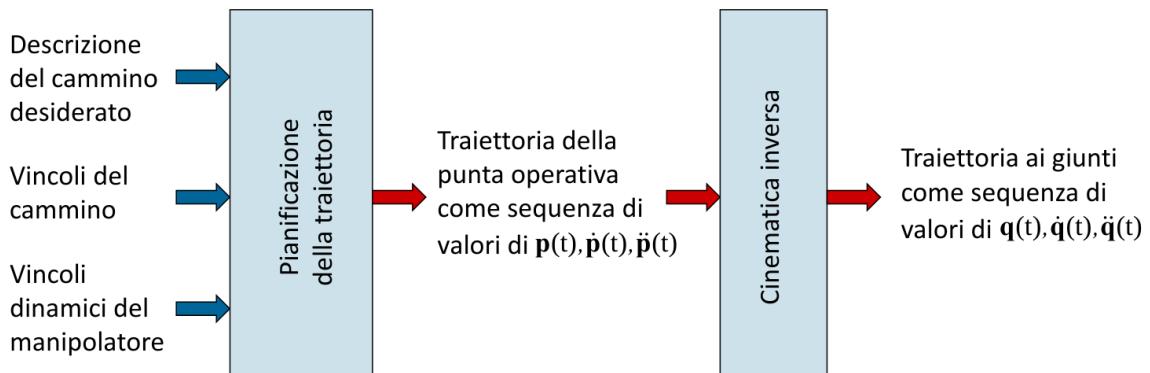
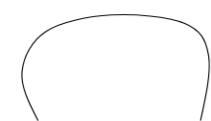


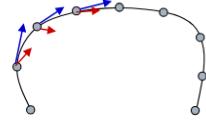
Figura 6.1: Flowpath

Path vs Trajectory:

- Il **cammino (path)** è il luogo dei punti nello spazio dei giunti o nello spazio operazionale, che il manipolatore deve seguire durante l'esecuzione del movimento assegnato
- La **traiettoria (trajectory)** è un cammino sul quale è stata specificata una **legge temporale**, per esempio in termini di velocità ed accelerazioni in ogni suo punto



Cammino, descritto da
 $f(p, \alpha) = 0$ o da $g(q) = 0$



Traiettoria, descritta da
 $f(p(t), \alpha(t)) = 0$ o da $g(q(t)) = 0$

Figura 6.2: Path vs Trajectory

Il cammino desiderato viene specificato dall'utente per mezzo di un numero ridotto di parametri, quali punti estremi del cammino, eventuali punti intermedi, primitive geometriche di interpo-

lazione. I requisiti sulla legge temporale sono principalmente legati a velocità ed accelerazioni massime.

Traiettorie nello spazio dei giunti o nello spazio operazionale?

- La pianificazione nello **spazio operazionale** è indicata ogni volta in cui sia necessario far seguire alla punta operativa un particolare percorso e/o sia necessario **evitare ostacoli** o zone “proibite” dello spazio di lavoro
- La pianificazione nello **spazio dei giunti** non consente di assegnare un particolare percorso alla punta, ma è utile per **evitare singolarità cinematiche** e/o per **sfruttare la ridondanza** del manipolatore, se presente

6.1 Traiettorie nello spazio dei giunti

In generale, devono comportare un basso sforzo computazionale e almeno posizioni e velocità devono essere funzioni continue del tempo (può essere interessante anche la continuità nell’accelerazione). Effetti indesiderati, come traiettorie non regolari per interpolare un dato insieme di punti, devono essere evitate.

6.1.1 Point-to-Point (PTP) motion

La traiettoria più semplice è quella corrispondente ad un **movimento lineare da un punto ad un altro** (PTP - Point-To-Point) nello spazio dei giunti (**attenzione: il movimento cartesiano risultante è in generale NON lineare!**).

Il manipolatore deve muoversi dalla configurazione iniziale $\mathbf{q}(t_0) = \mathbf{q}_0$ alla configurazione finale desiderata $\mathbf{q}(t_f) = \mathbf{q}_f$:

$$\mathbf{q}(t_0) = \mathbf{q}_0 \quad \longrightarrow \quad \mathbf{q}(t_f) = \mathbf{q}_f$$

rispettando i vincoli di velocità ed accelerazioni massime determinati dalle caratteristiche tecnologiche degli attuatori.

Eventuali vincoli sul *jerk* (derivata dell’accelerazione rispetto al tempo) possono essere presenti per evitare sollecitazioni meccaniche eccessive. Si parla di movimento coordinato quando gli istanti di inizio e fine del moto coincidono per tutti i giunti.

Ovviamente esistono infinite soluzioni al problema sopra mostrato, possiamo però provare ad identificare quella ad energia minima. Immagina un manipolatore con giunti rotoidali, di conseguenza i movimenti saranno eseguiti attraverso delle coppie $\boldsymbol{\tau}$ fornite dai motori: possiamo provare a **minimizzare l’energia di tali motori**: imponendo $\dot{\mathbf{q}} = \boldsymbol{\omega}$ abbiamo ($F = ma \rightsquigarrow \boldsymbol{\tau} = I\ddot{\boldsymbol{\omega}}$):

$$\boldsymbol{\tau} = \mathbf{I}\ddot{\boldsymbol{\omega}}$$

dove \mathbf{I} è il tensore d’inerzia del manipolatore. Quindi, volendo minimizzare l’energia:

$$\begin{aligned} & \text{minimize} && \int_0^{t_f} \boldsymbol{\tau}^2(t) dt \\ & \text{subject to} && \int_0^{t_f} \boldsymbol{\omega}(t) dt = \mathbf{q}_f - \mathbf{q}_i \end{aligned}$$

dove ricordiamo che $\int \mathbf{f}^2 = \text{energia}$.

Le soluzioni al problema sopra definito sono nella di un polinomio di 2° grado:

$$\dot{\mathbf{q}}(t) = \boldsymbol{\omega}(t) = \alpha t^2 + \beta t + \gamma$$

Dalla forma di $\dot{\mathbf{q}}$ possiamo ricavare le forme di \mathbf{q} e $\ddot{\mathbf{q}}$, che saranno rispettivamente polinomi di 3° grado e di 1° grado¹. Per comodità di notazione indichiamo analiticamente i coefficienti di \mathbf{q} , e poi deriviamo per ottenere gli altri:

$$\begin{aligned}\mathbf{q}(t) &= a_3 t^3 + a_2 t^2 + a_1 t + a_0 \\ \dot{\mathbf{q}}(t) &= 3a_3 t^2 + 2a_2 t + a_1 \\ \ddot{\mathbf{q}}(t) &= 6a_3 t + 2a_2\end{aligned}\tag{6.1}$$

(per capire meglio, rispetto alla notazione di prima abbiamo $\alpha = 3a_3$, $\beta = 2a_2$, $\gamma = a_1$).

Di seguito vediamo un esempio di traiettoria:

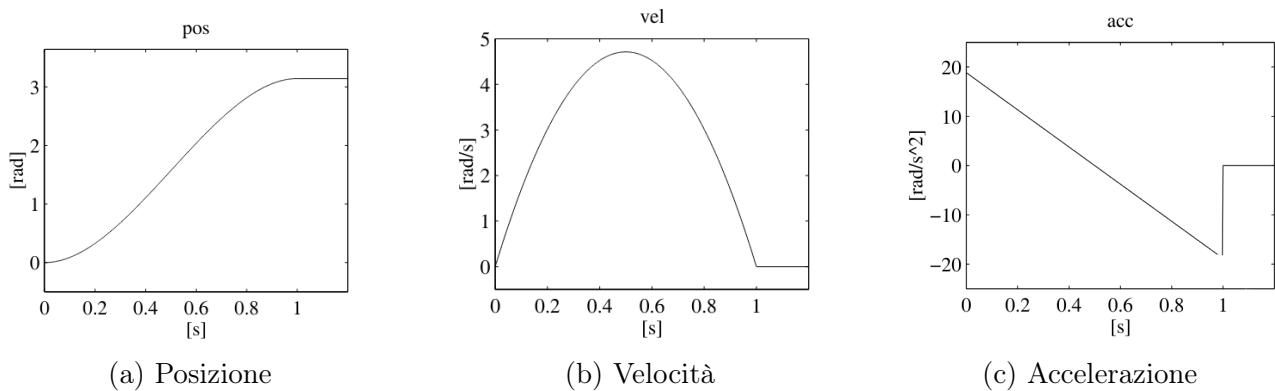


Figura 6.3: Esempio

notare come l'accelerazione non è continua per $t = 1$: fortunatamente a quell'istante di tempo $v = 0$ e quindi non ci sono problemi.

Soluzione generale

Quindi, come impostiamo i nostri requisiti? Visto che abbiamo 4 coefficienti (a_4, \dots, a_0) possiamo imporre 4 grandezze: nel caso seguente imporremo posizione iniziale e finale, e velocità iniziale e finale (che solitamente sono imposte a 0):

$$\left\{ \begin{array}{l} \mathbf{q}(t_i) = a_3 t_i^3 + a_2 t_i^2 + a_1 t_i + a_0 \\ \mathbf{q}(t_f) = a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 \\ \dot{\mathbf{q}}(t_i) = 3a_3 t_i^2 + 2a_2 t_i + a_1 \\ \dot{\mathbf{q}}(t_f) = 3a_3 t_f^2 + 2a_2 t_f + a_1 \\ \mathbf{q}(t_i) = \mathbf{q}_i \\ \mathbf{q}(t_f) = \mathbf{q}_f \\ \dot{\mathbf{q}}(t_f) = \dot{\mathbf{q}}_i \\ \dot{\mathbf{q}}(t_f) = \dot{\mathbf{q}}_f \end{array} \right.$$

¹se $\dot{\mathbf{q}}$ è un polinomio di 2° ordine, derivando per andare a $\ddot{\mathbf{q}}$ otteniamo un poly di 1° ordine, mentre integrando per andare a \mathbf{q} otteniamo un poly di 3° ordine

Visto che solitamente $t_i = 0$, otteniamo:

$$\begin{cases} \mathbf{q}_i = a_0 \\ \mathbf{q}_f = a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 \\ \dot{\mathbf{q}}_i = a_1 \\ \dot{\mathbf{q}}_f = 3a_3 t_f^2 + 2a_2 t_f + a_1 \end{cases}$$

dove, imponendo i nostri requisiti di $q_i, q_f, \dot{q}_i, \dot{q}_f$ otteniamo i coefficienti per i 3 polinomi.

Alternativa per assegnare l'accelerazione

Abbiamo visto che una soluzione della forma (6.1) ci garantisce la minimizzazione dell'energia dei motori, purtroppo però abbiamo a disposizione solo 4 coefficienti, quindi non c'è modo (ad esempio) di impostare l'accelerazione. Se vogliamo assegnare anche \ddot{q}_f, \ddot{q}_i ci servono 4 coefficienti, ovvero:

$$\mathbf{q}(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

N.B. una soluzione di questo tipo non minimizza l'energia.

6.1.2 Profilo di velocità trapezoidale (2-1-2)

Anche chiamata LSPB (*linear segment paraboloid blend*), questa è una traiettoria molto frequente nelle applicazioni industriali, vista la sua semplicità di implementazione. Inoltre permette di verificare che le velocità e le accelerazioni siano compatibili con le caratteristiche fisiche del manipolatore (visto che se ne possono impostare minimi/massimi facilmente).

L'idea alla base è di imporre un profilo **trapezoidale** alla velocità, ovvero accelerazione costante all'inizio e alla fine e nulla nel mezzo (i.e. velocità lineare all'inizio/fine e costante nel mezzo).

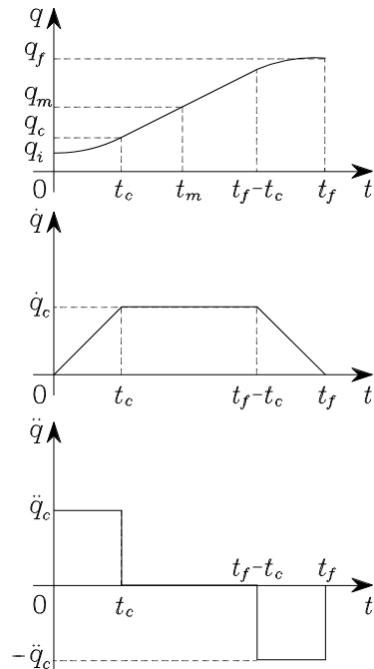


Figura 6.4: Esempio

Visto che (in generale) supponiamo velocità iniziale/finale nulle e durata dei periodi di accelerazione uguale, otteniamo traiettorie simmetriche rispetto ad un punto di mezzo, che è dato da:

$$t_m = \frac{t_f}{2} \quad , \quad \mathbf{q}_m = \frac{\mathbf{q}_f + \mathbf{q}_i}{2}$$

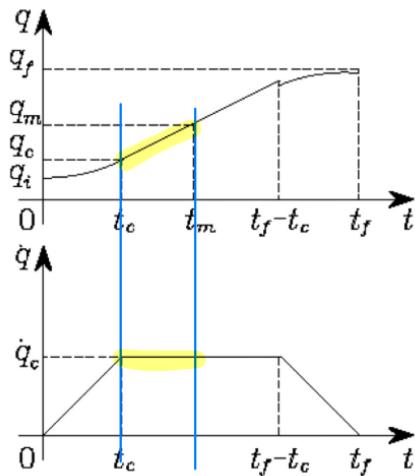
Calcolo dei valori di velocità e accelerazione

Solitamente, per questo tipo di profilo viene dato il tempo finale t_f (ovvero, visto che in generale $t_i = 0$, viene dato il tempo desiderato per eseguire la traiettoria). Da questo vincolo però dobbiamo trovarci i corretti valori di velocità che ci assicurano un passaggio da \mathbf{q}_i a \mathbf{q}_f nel tempo dato t_f .

In primis possiamo vedere che la velocità di crociera (i.e. quella costante) $\dot{\mathbf{q}}_c$ è data da:

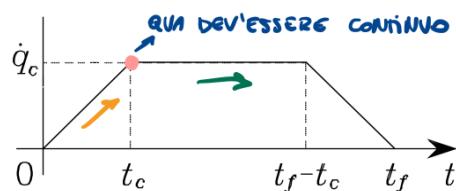
$$\dot{\mathbf{q}}_c = \frac{\mathbf{q}_m - \mathbf{q}_i}{t_m - t_c} = \frac{\text{rise}}{\text{run}}$$

dove \mathbf{q}_c e t_c sono rispettivamente l'angolo e il tempo in cui inizia la fase di crociera. La formula può essere facilmente capita anche guardando l'immagine di seguito.



Ora, per quanto riguarda l'accelerazione, abbiamo il seguente vincolo (richiamando $v = at$):

$$\ddot{\mathbf{q}}t_c = \dot{\mathbf{q}} = \frac{\mathbf{q}_m - \mathbf{q}_i}{t_m - t_c}$$



Inoltre, ricordando fisica 1, possiamo esprimere il valore di q_c tramite la legge di moto rettilineo uniformemente accelerato:

$$x = x_0 + \frac{1}{2}at^2 \implies \mathbf{q}_c = \mathbf{q}_i + \frac{1}{2}\ddot{\mathbf{q}}_c t_c^2$$

Concludendo, combinando le ultime due equazioni abbiamo:

$$\ddot{\mathbf{q}}_c t_c^2 + \ddot{\mathbf{q}}_c t_f t_c + \mathbf{q}_f - \mathbf{q}_i = 0$$

Questa equazione può ora essere risolta per ottenere t_c . In particolare di solito per risolverla si assegna il seguente constraint:

$$\operatorname{sgn}(\ddot{\mathbf{q}}) = \operatorname{sgn}(\mathbf{q}_f - \mathbf{q}_i)$$

Quindi, dati t_f , \mathbf{q}_i , \mathbf{q}_f troviamo:

$$t_c = \frac{t_f}{2} - \frac{1}{2} \sqrt{\frac{t_f^2 \ddot{\mathbf{q}} - 4(\mathbf{q}_f - \mathbf{q}_i)}{\ddot{\mathbf{q}}_c}}$$

(6.2)

vista la forma della soluzione, abbiamo ulteriori vincoli (dati dal segno del fattore sotto radice):

$$|\ddot{\mathbf{q}}| \geq \frac{4|\mathbf{q}_f - \mathbf{q}_i|}{t_f^2} \quad (6.3)$$

Conoscere il valore di t_c ci permette di disegnare il controllo per la traiettoria terminando la fase di accelerazione ed iniziando quella di decelerazione ai tempi ora noti.

Bang-Bang controller

Nel caso particolare in cui

$$t_c = \frac{t_f}{2}$$

(solitamente abbiamo \leq invece di $=$), otteniamo un controllore "bang-bang", ovvero con **profilo triangolare di velocità**. Il nome deriva dal fatto che i motori sono attuati o al massimo o sono spenti

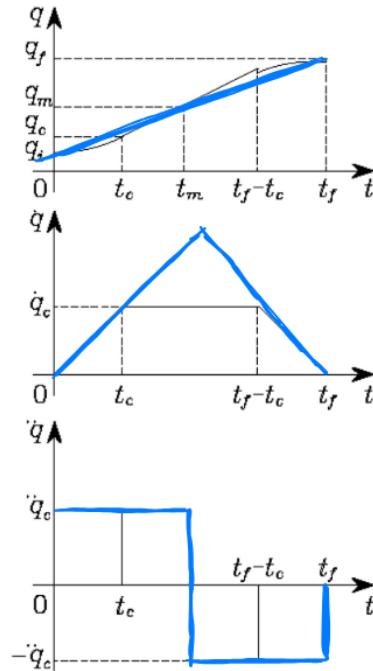


Figura 6.5: Esempio bang-bang

Soluzione profilo trapezoidale

Per concludere, vediamo come si ottengono valori di $\mathbf{q}(t)$ noti t_f , q_i e q_f . Calcolando t_c con (6.2) prima e scegliendo un \ddot{q} che rispetta il vincolo (6.3) abbiamo:

$$\boxed{\mathbf{q}(t) = \begin{cases} \mathbf{q}_i + \frac{1}{2}\ddot{q}_c t^2 & 0 \leq t \leq t_c \\ \mathbf{q}_i + \ddot{q}_c t_c (t - t_c/2) & t_c < t \leq t_f - t_c \\ \mathbf{q}_i - \frac{1}{2}\ddot{q}_c (t_f - t)^2 & t_f - t_c < t \leq t_f \end{cases}}$$

Esempio

Dati

$$(q_i, q_f, t_i, t_f, \ddot{q}_c) = (0, \pi, 0, 1, 6\pi)$$

visto che

$$\frac{4|q_f - q_i|}{t_f^2} = 4\pi$$

la scelta di $\ddot{q} = 6\pi$ è corretta. Ora usando le formule viste:

$$t_c = \frac{1}{2} - \frac{1}{2}\sqrt{\frac{\pi}{3}} = 0.12$$

e quindi:

$$q(t) = \begin{cases} 3\pi t^2 & 0 \leq t \leq 0.12 \\ 0.72\pi(t - 0.06) & 0.12 < t \leq 0.88 \\ \pi - 3\pi(1-t)^2 & 0.88 < t \leq 1 \end{cases}$$

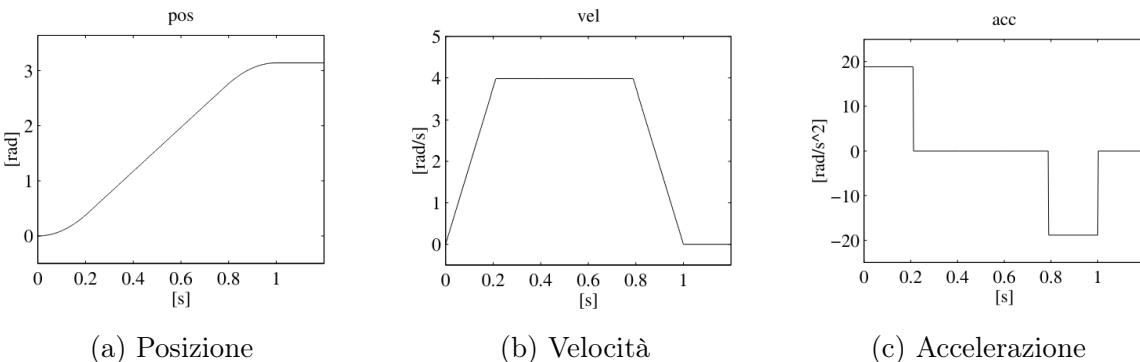


Figura 6.6: Esempio

◇

Assegnazione della velocità di crociera

Sfruttando le formule note invece di assegnare l'accelerazione, possiamo ottenere il valore di \dot{q}_c da \ddot{q}_c , di fatto assegnando la velocità invece dell'accelerazione:

Assigning

$$\frac{|q_f - q_i|}{t_f} < |\dot{q}_c| \leq 2 \frac{|q_f - q_i|}{t_f}$$

From

$$\dot{q}_c = \ddot{q}_c t_c \quad \ddot{q}_c t_c^2 - \ddot{q}_c t_f t_c + q_f - q_i = 0$$

We derive

$$t_c = \frac{q_i - q_f + \dot{q}_c t_f}{\dot{q}_c} \quad \ddot{q}_c = \frac{\dot{q}_c^2}{q_i - q_f + \dot{q}_c t_f}$$

And we generate

$$q(t) = \begin{cases} q_i + \frac{1}{2} \ddot{q}_c t^2 & 0 \leq t \leq t_c \\ q_i + \ddot{q}_c t_c \left(t - \frac{t_c}{2} \right) & t_c < t \leq t_f - t_c \\ q_f - \frac{1}{2} \ddot{q}_c (t_f - t)^2 & t_f - t_c < t \leq t_f \end{cases}$$

Comparazione con PTP

Visto che questa soluzione non minimizza l'energia, abbiamo ovviamente una soluzione peggiore da quel punto di vista, anche se non di troppo (circa 12.5% peggio). Come già accennato questo profilo è molto usato perché semplice da implementare e leggero dal punto di vista computazionale.

6.1.3 Movimento attraverso una sequenza di punti

In caso di compiti complessi, la traiettoria è specificata da una sequenza di punti (configurazioni q_k) che il manipolatore deve raggiungere in determinati istanti di tempo.

In questo caso si utilizzano polinomi interpolatori di grado opportuno, tali da garantire la continuità di $q(t)$ ed almeno della sua derivata prima (la velocità ai giunti). Nel caso in cui venga garantita anche la continuità dell'accelerazione si parla di splines, sequenze di polinomi di terzo grado, calcolabili per via ricorsiva.

Svantaggi

- Non è possibile assegnare velocità iniziali e finali
- I polinomi di alto ordine hanno una tendenza oscillatoria. Questo può portare a traiettorie innaturali
- La precisione numerica peggiora con l'aumentare del grado del polinomio
- Risolvere il sistema di equazioni ha un alto costo computazionale;
- I coefficienti del polinomio dipendono da tutti i punti assegnati. Questo implica che, anche un cambiamento di un singolo punto del percorso impone il ricalcolo di tutti i coefficienti

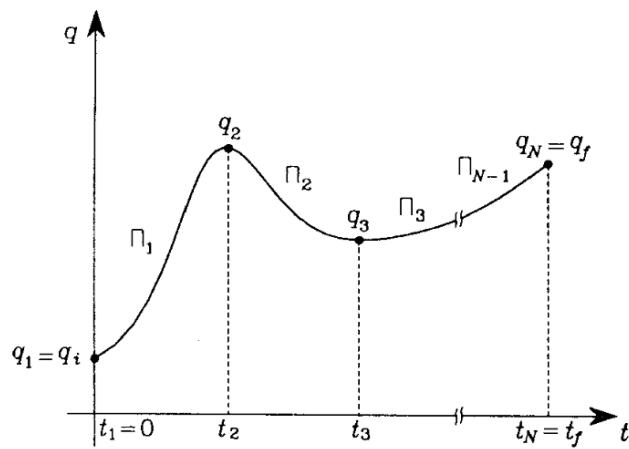


Figura 6.7

Invece di usare polinomi di grado N potremmo usare alternative, come una serie di polinomi di 3° grado, sequenze di polinomi 2-1-2, etc.

6.2 Traiettorie nello spazio operazionale

La pianificazione della traiettoria nello spazio operazionale permette di definire un **particolare percorso** (ad esempio un tratto rettilineo o un arco di circonferenza) che deve **essere seguito dalla punta operativa** per lo svolgimento del compito prefissato e/o per evitare ostacoli o zone “proibite” dello spazio di lavoro.

A partire dalla traiettoria cartesiana pianificata $\mathbf{x}(t)$ (e sue derivate) vengono determinati online i corrispondenti campioni di posizione, velocità ed accelerazione ai giunti per mezzo della cinematica inversa del manipolatore.

In generale un movimento continuo per le coordinate cartesiane di posizione $\mathbf{x}(t)$ equivale ad un cammino da \mathbf{x}_0 a \mathbf{x}_f , pianificato assegnando una **specifica curva parametrica**. I cammini cartesiani più comunemente pianificati corrispondono a: **segmenti di retta** e **archi di circonferenza** (il primo caso si riconduce al movimento lineare PTP nello spazio).

6.2.1 Primitive

Identifichiamo una curva Γ tramite la sua rappresentazione parametrica alla lunghezza d’arco s :

$$\mathbf{f} = \mathbf{p}(s)$$

Eccetto casi particolari, \mathbf{p} permette la definizione di 3 versori che lo caratterizzano (vedi fig. 6.8):

Versore tangente	$\mathbf{t} = \frac{d\mathbf{p}}{ds}$
Versore normale	$\mathbf{n} = \frac{\frac{d^2\mathbf{p}}{ds^2}}{\left\ \frac{d^2\mathbf{p}}{ds^2} \right\ }$
Versore binormale	$\mathbf{b} = \mathbf{t} \times \mathbf{n}$

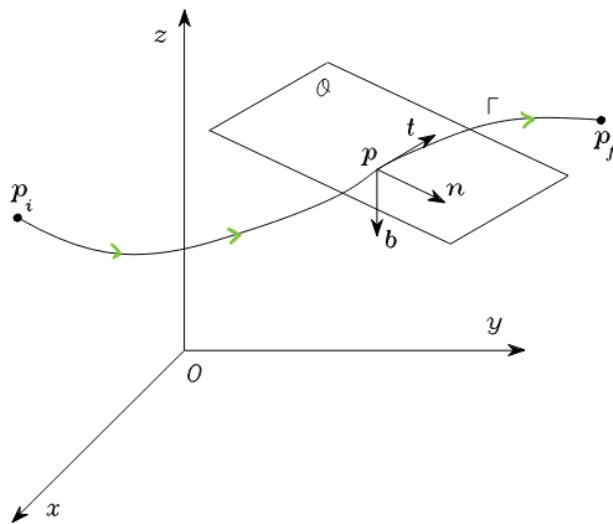


Figura 6.8

6.2.2 Primitiva per un segmento

La rappresentazione parametrica di un segmento che connette \$p_i\$ a \$p_f\$ è:

$$p(s) = p_i + \frac{s(p_f - p_i)}{\|p_f - p_i\|}$$

i versori:

$$t = \frac{(p_f - p_i)}{\|p_f - p_i\|}$$

mentre \$n\$ non è ben definito.

6.2.3 Primitiva per arco di circonferenza

Per quanto riguarda una circonferenza, essa può essere univocamente definita tramite il versore \$r\$ dell'asse centrale, un vettore \$d\$ di un punto nell'asse e un vettore \$p_i\$ di un punto sulla circonferenza.

Sia \$\delta < p_i - d\$ e \$|\delta r| < \|\delta\|\$ per assicurare che \$p_i\$ non sia sull'asse centrale. Il centro della circonferenza è dato da \$c = d + (\delta^T r)r\$, mentre il raggio \$\rho = \|p_i - c\|\$.

Prendiamo ora un sistema di riferimento \$\mathcal{O}' - x'y'z'\$ avente l'origine coincide con il centro della circonferenza \$c\$, l'asse \$x\$ orientato nel verso di \$p_i - c\$ e l'asse \$z\$ è orientato nel verso di \$r\$.

Quindi, nel r.f. appena definito, i punti della circonferenza sono rappresentati da:

$$p'(s) = \begin{bmatrix} \rho \cos\left(\frac{s}{\rho}\right) \\ \rho \sin\left(\frac{s}{\rho}\right) \\ 0 \end{bmatrix}$$

di conseguenza, se definiamo un'apposita matrice di rotazione \${}^{\mathcal{O}}\mathbf{R}_{\mathcal{O}'}\$ possiamo indicare la circonferenza nell'r.f. originale \$\mathcal{O}\$ come:

$$p(s) = c + {}^{\mathcal{O}}\mathbf{R}_{\mathcal{O}'} p'(s)$$

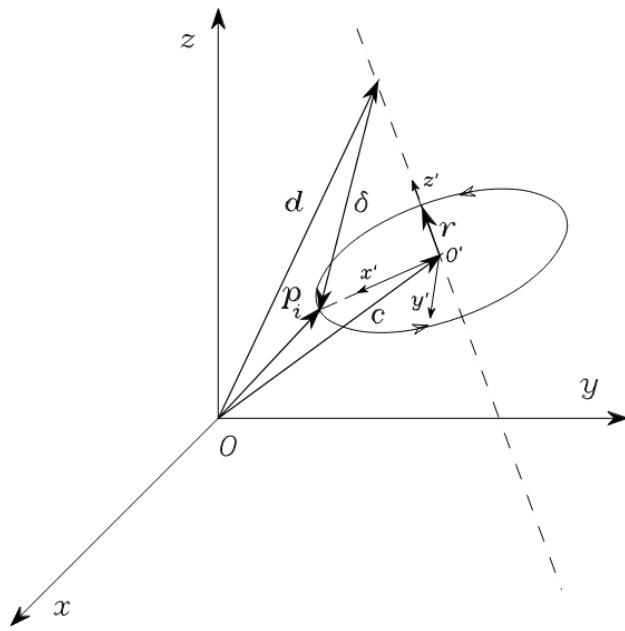


Figura 6.9

6.2.4 Generazione traiettorie

Abbiamo definito le primitive: ci manca ora da utilizzarle per pianificare delle traiettorie. Vogliamo generare una traiettoria nello spazio operazionale del tipo:

$$x_{\text{traj}}(t) = \begin{bmatrix} p(t) \\ \phi(t) \end{bmatrix}$$

Data la primitiva $p = f(s)$ possiamo utilizzare essa e le sue derivate per generare la relativa parte della traiettoria tramite una legge d'interpolazione (e.g. cubico o 2-1-2):

Segment

$$p(s) = p_i + \frac{s(p_f - p_i)}{\|p_f - p_i\|} \quad \dot{p} = \frac{\dot{s}(p_f - p_i)}{\|p_f - p_i\|} = \dot{s}t \quad \ddot{p} = \frac{\ddot{s}(p_f - p_i)}{\|p_f - p_i\|} = \ddot{s}t$$

Circumference

$$p(s) = c + R \begin{bmatrix} \rho \cos\left(\frac{s}{\rho}\right) \\ \rho \sin\left(\frac{s}{\rho}\right) \\ 0 \end{bmatrix} \quad \dot{p} = \mathbf{R} \begin{bmatrix} -\dot{s} \sin\left(\frac{s}{\rho}\right) \\ \dot{s} \cos\left(\frac{s}{\rho}\right) \\ 0 \end{bmatrix} \quad \ddot{p} = \mathbf{R} \begin{bmatrix} -\dot{s}^2 \frac{1}{\rho} \cos\left(\frac{s}{\rho}\right) - \ddot{s} \sin\left(\frac{s}{\rho}\right) \\ -\dot{s}^2 \frac{1}{\rho} \sin\left(\frac{s}{\rho}\right) + \ddot{s} \cos\left(\frac{s}{\rho}\right) \\ 0 \end{bmatrix}$$

Per quanto riguarda l'orientamento, invece, andiamo ad utilizzare gli angoli di Eulero (così da evitare problemi di vincoli di ortogonalità che hanno le matrici di rotazione):

We have to assign a timing law to the Euler angles

$$\phi = (\varphi, \theta, \psi)$$

$$\begin{aligned}\phi &= \phi_i + \frac{s(\phi_f - \phi_i)}{\|\phi_f - \phi_i\|} \\ \dot{\phi} &= \frac{\dot{s}(\phi_f - \phi_i)}{\|\phi_f - \phi_i\|} \\ \ddot{\phi} &= \frac{\ddot{s}(\phi_f - \phi_i)}{\|\phi_f - \phi_i\|}\end{aligned}$$

Una volta specificati, tramite $p(t)$ e $\phi(t)$, un percorso ed una traiettoria nello spazio operativo, l'applicazione della cinematica inversa ci permetterà di ricavare le corrispondenti traiettorie $q(t)$ nello spazio dei giunti.

Capitolo 7

Controllo

Il problema del controllo del moto di un manipolatore consiste nella **determinazione** dell'andamento temporale delle coppie generalizzate (**generalized torque**), che gli attuatori devono applicare ai giunti **affinché venga eseguito il compito assegnato**, rispettando specifiche sul transitorio e sul comportamento in regime permanente.

Lo schema di controllo può essere sviluppato:

1. Nello **spazio dei giunti**
2. Nello **spazio operazionale**

tenendo conto che la descrizione del compito è solitamente elaborata nello spazio operazionale, mentre le azioni di controllo esercitate dagli attuatori sono definite nello spazio dei giunti.

Joint space control

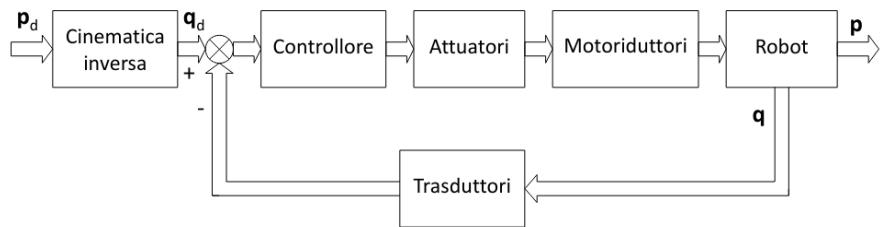


Figura 7.1: Schema generale di *joint space control*

L'azione di controllo fa sì che $q(t)$ inseguiva il vettore $q_d(t)$ della traiettoria desiderata ai giunti, ricavato dalla cinematica inversa (che è comodamente disaccoppiata dal controllo).

Svantaggio: Non si ha né un feedback, né un controllo diretto sulla variabile dello spazio operazionale $\mathbf{p} \implies$ dobbiamo avere una cinematica inversa perfetta (ma tolleranze di costruzione, elasticità ai giunti, giochi nei motoriduttori potrebbero causare imprecisioni).

Operational space control

Qui cerchiamo di fare inseguire un vettore \mathbf{p}_d definito nello spazio operazionale. Con questa tecnica introduciamo una **complessità maggiore**, visto che la cinematica inversa è ora inclusa

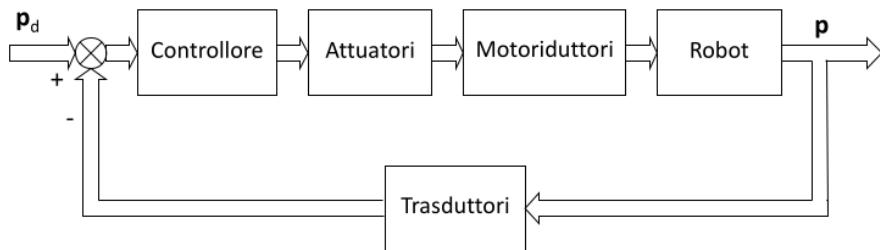


Figura 7.2: Schema generale di *operational space control*

nel controllore. Il **vantaggio** però è quello di poter operare direttamente nello spazio operazionale.

Anche se in primo luogo potrebbe non sembrare un vantaggio (la “misura” di \mathbf{p} è spesso ricavata indirettamente da misure ai giunti per mezzo della cinematica diretta \Rightarrow sembra lo stesso svantaggio di imprecisione della cinematica inversa che c’era nel controllo sui giunti): questo controllo è però utile (ad esempio) come base per il controllo dell’interazione con l’ambiente.

Cosa vedremo

Vedremo che la presenza di **motoriduttori** con elevato rapporto di trasformazione tende a **linearizzare la dinamica** del manipolatore e quindi a disaccoppiare i giunti, grazie alla riduzione degli effetti non lineari \Rightarrow si giustifica in tal caso l’adozione di una strategia di **controllo decentralizzato**, a giunti indipendenti.

Questo però introduce uno svantaggio: i motoriduttori introducono possibili fenomeni di elasticità, gioco ed attriti non lineari, talora più “fastidiosi” degli effetti di accoppiamento fra i giunti. L’utilizzo di **motori ad azione diretta** elimina questi problemi ma mantiene rilevanti gli effetti non lineari e di accoppiamento fra i giunti, che non possono essere trascurati o considerati come disturbi \Rightarrow diventa opportuno in tal caso utilizzare una strategia di **controllo centralizzato**, che tenga conto della dinamica non lineare del manipolatore. **Svantaggio:** la legge di controllo è necessariamente non lineare e computazionalmente pesante.

7.1 Attuatori elettrici

Prima di introdurre i sistemi di controllo, iniziamo parlando di come poter modellare gli attuatori elettrici che andremmo ad utilizzare.

Gli attuatori maggiormente impiegati sono motori DC o brushless. Fortunatamente dal punto di vista matematico, un motore DC e un brushless dotato degli appositi sensori, possono essere modellati allo stesso modo. A seconda delle caratteristiche del regolatore inserito, il comportamento dell’attuatore può essere assimilato:

- Ad un generatore di velocità controllata (**velocity controlled**)
- Ad un generatore di coppia controllata (**torque controlled**)

In fig. 7.3 possiamo vedere il modello del motore che utilizzeremo. Notazione usata:

- ω_m e θ_m sono la velocità e la posizione angolare dell’albero motore
- V_a e I_a sono la tensione e la corrente del circuito di armatura

- τ_m è la coppia motrice, mentre τ_r è la coppia dovuta al carico

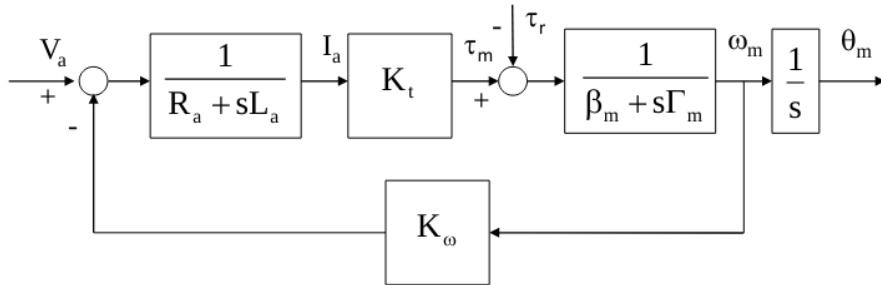


Figura 7.3: Modello del motorore comandato in armatura

7.1.1 Derivazione modello

Partendo dal circuito illustrato in fig. 7.4 (il quale modella un motore DC), possiamo derivare il circuito mostrato in fig. 7.3.

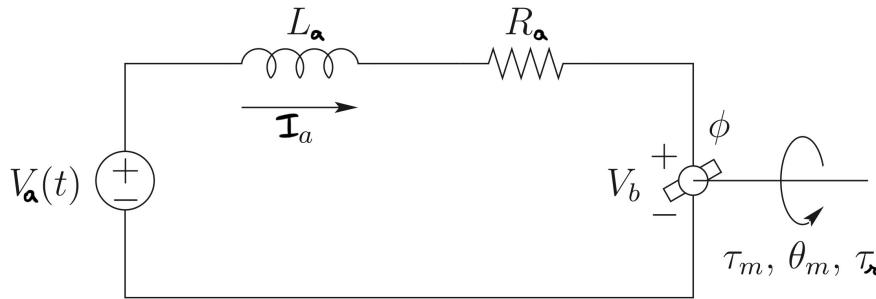


Figura 7.4: Modello circuitale del motore

Il bilancio elettrico dell'armatura è definito da:

$$L \frac{dI_a}{dt} + RI_a = V_a - V_b \quad \xrightarrow{\mathcal{L}} \quad V_a = (R_a + sL_a)I_a + V_g \quad (7.1)$$

$$V_g = K_\omega \omega_m$$

Dove V_g è la **back-emf**, ovvero quella tensione generata dalla dal movimento delle coils del motore nel campo magnetico (e questa tensione è proporzionale alla velocità del conduttore nel campo, ovvero a $\omega_m = \dot{\theta}_m$). Questa tensione, tenderà ad opporsi al flusso di corrente nel conduttore.

Il bilancio meccanico è invece definito da:

$$\tau_m = (s\Gamma_m + \beta_m)\omega_m + \tau_r \quad (7.2)$$

$$\tau_m = K_i \Gamma_a \quad (7.3)$$

Dove Γ_m e β_m sono rispettivamente il momento d'inerzia e il coefficiente di attrito viscoso dell'albero motore. Queste formule derivano da:

1. Ricordando da fisica 1: $F = ma \rightsquigarrow \tau = I\alpha$ dove I , α sono il momento di inerzia e l'accellerazione angolare
2. Allora, facendo riferimento al motore: $\tau = I\alpha \rightsquigarrow \tau_m = \Gamma_m \dot{\omega}_m$
3. Ora "aggiungiamo" l'attrito viscoso (che come ricordiamo è proporzionale alla velocità): $\tau_m - \beta_m \omega_m = \Gamma_m \dot{\omega}_m$
4. Infine, passando a Laplace: $\tau_m = \beta_m \omega_m + \Gamma_m s \omega_m \implies \tau_m = (s\Gamma_m + \beta_m)\omega_m \quad \square$

7.1.2 Amplificatore di potenza

Visto che solitamente V_a ha valori elevati, si tende ad utilizzare un **amplificatore di potenza** per poter comandare il motore con una tensione $V_c \ll V_a$: il nuovo circuito è illustrato in fig. 7.5.

G_v è il guadagno di tensione, mentre T_v è una costante di tempo trascurabile (poiché nell'ordine di 10-100 kHz), di conseguenza il blocco dell'amplificatore di potenza può essere assimilato al solo guadagno G_v .

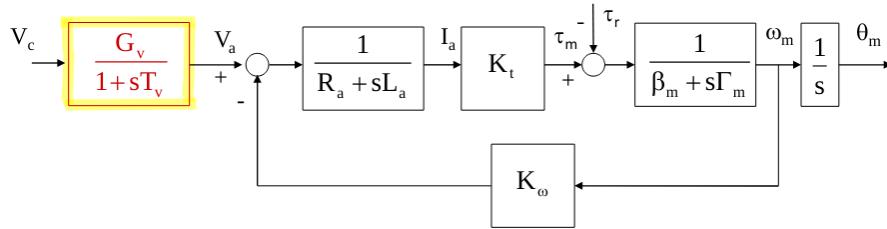


Figura 7.5: Modello motore con amplificatore di potenza

7.1.3 Velocity generator

Possiamo ora completare il circuito aggiungendo una retroazione sulla corrente I_a e un compensatore che utilizzi tale feedback.

Prima di vedere l'effetto della retroazione, però, iniziamo dal caso più semplice, dove $K_i = 0$: in questa modalità riusciamo ad impostare a piacimento la velocità del motore, giocando sulla tensione di controllo V'_c .

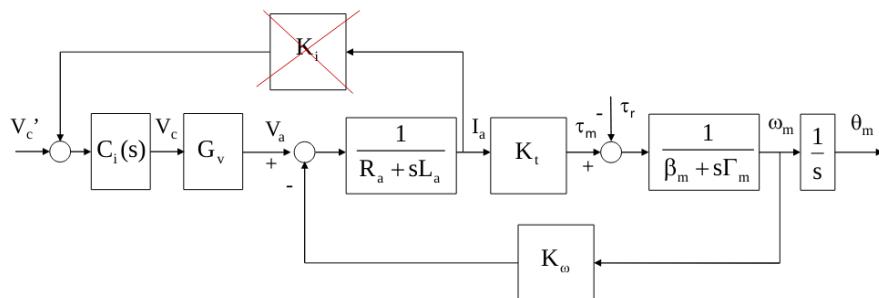


Figura 7.6: Circuito motore in modalità velocity generator

Per dimostrare quanto affermato, analizziamo lo steady-state del circuito in fig. 7.6. Ricordandoci che il coefficiente di attrito viscoso meccanico β_m è trascurabile rispetto al coefficiente di attrito viscoso elettrico (*mi son dimenticato come si ricava ma avevo chiesto al prof e aveva senso*):

$$\beta_m \ll \frac{K_\omega K_t}{R_a}$$

e supponendo $C_i(s) = 1$, $\tau_r = 0$ (per semplicità di calcolo), otteniamo:

$$\omega_m \approx \frac{G_v}{K_\omega} V'_c$$

Ovvero, in questo caso, possiamo impostare la velocità angolare a piacere (per la derivazione del valore di steady-state vedi appendice A.2).

7.1.4 Torque generator

Passiamo ora al caso in cui $K_i \neq 0$: circuito in fig. 7.7. In questa modalità, vedremo invece che è possibile controllare a piacere la coppia (al posto della velocità come nel caso precedente).

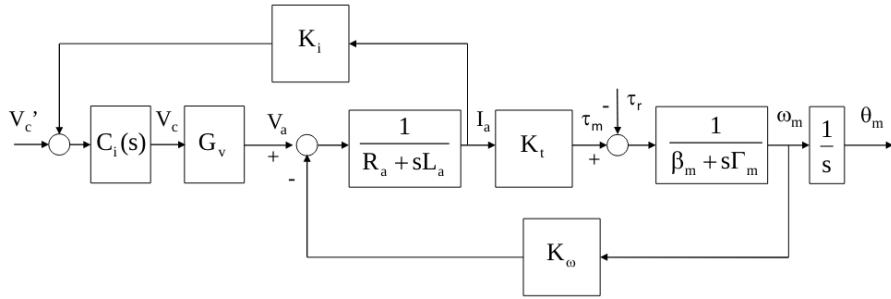


Figura 7.7: Circuito motore in modalità torque generator

Adesso, come accennato, il motore si comporta come un *torque-controlled generator*. Questo perchè, con $K_i \neq 0$, se si sceglie $K_i \gg R_a$ otteniamo il seguente steady-state (la derivazione segue lo stesso ragionamento usato per il caso velocity-controlled):

$$\tau_m \approx \frac{K_t}{K_i} \left(V_c' - \frac{K_\omega}{G_v} \omega_m \right) \approx \frac{K_t}{K_i} V_c'$$

(supponendo $G_v \gg 1 \implies \frac{K_\omega}{G_v} \rightarrow 0$). Questo significa che possiamo impostare il torque del motore giocando sul voltaggio di controllo V_c' .

7.1.5 Torque generator vs. Velocity generator

Quando ci conviene utilizzare uno invece che l'altro? In seguito vedremo che in tutte le applicazioni in cui è necessario ottenere un'elevata reiezione di coppie di disturbo, come nel caso del **controllo decentralizzato a giunti indipendenti**, è opportuno NON inserire l'anello di controllo in corrente ed utilizzare i motori come **generatori di velocità**.

Al contrario, in caso di strategie di **controllo centralizzate**, è consigliabile introdurre l'anello di retroazione in corrente ed utilizzare i motori come **generatori di coppie** (opportuni accorgimenti vengono adottati in entrambi i casi per limitare la corrente ed evitare danni ai dispositivi).

E' possibile vedere questa differenza calcolando le relazioni input/output fra ω_m e V_c' , τ_r . Da queste equazioni si può vedere che senza feedback di corrente (i.e. velocity-controlled generator) si ha una reiezione migliore della coppia di disturbo τ_r : i coefficienti legati a τ_r in questo caso sono molto minori del caso torque-controlled (\implies riduciamo di più gli effetti dei disturbi).

Semplificando:

$$\omega_m^{\text{velocity}} = \alpha_{\text{velocity}} V_c' + \lambda_{\text{velocity}} \tau_r \quad (7.4)$$

$$\omega_m^{\text{torque}} = \alpha_{\text{torque}} V_c' + \lambda_{\text{torque}} \tau_r \quad (7.5)$$

se il primo è per il caso velocity-controlled, mentre il secondo per il caso torque-controlled abbiamo che $\lambda_{\text{velocity}} \ll \lambda_{\text{torque}}$.

7.2 Trasmissions *aka* motoriduttori

Vediamo ora in dettaglio gli effetti che i motoriduttori utilizzati nel motore hanno sul sistema.

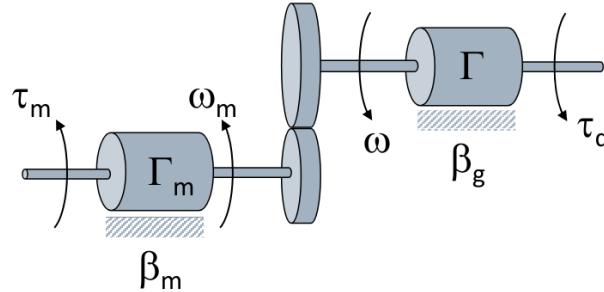


Figura 7.8: Motoriduttore

In generale, se indichiamo con k_r il rapporto di trasformazione (*gear-ratio*) di un motoriduttore ideale, abbiamo che:

$$\theta_m = k_r \theta$$

ove θ_m è la posizione angolare dell'albero motore e θ è la posizione angolare sul secondario (giunto). k_r è invece definito come il rapporto tra i raggi dei due *gears*:

$$k_r = \frac{r}{r_m} = \frac{\theta_m}{\theta} = \frac{\omega_m}{\omega}$$

Considerando attriti viscosi e coppie, possiamo scrivere le equazioni di equilibrio:

$$\begin{aligned}\tau_m &= \Gamma_m \dot{\omega}_m + \beta_m \omega_m + f r_m \\ f r &= \Gamma \dot{\omega} + \beta_g \omega + \tau_d\end{aligned}$$

(La forza f scambiata tra i due ingranaggi genera una coppia di reazione $f r_m$ per il moto all'asse del motore ed una coppia motrice $f r$ per il moto di rotazione del carico).

Dopo alcuni passaggi si ottiene:

$$\tau_m = \left(\Gamma_m + \frac{\Gamma}{k_r^2} \right) \dot{\omega}_n + \left(\beta_m + \frac{\beta_g}{k_r^2} \right) \omega_m + \frac{\tau_d}{k_r}$$

Notiamo che **qualunque coppia applicata sul secondario è riportata sul primario ridotta del fattore k_r (e viceversa)**.

Poiché k_r è elevato, l'effetto di coppie di **disturbo** agenti sul secondario risulta **fortemente ridotto** sul primario. Se tali coppie dipendono non-linealmente da θ , allora la presenza di un alto fattore di riduzione tende a **linearizzare il sistema**.

Nel caso vettoriale (molteplici giunti), possiamo riscrivere alcune utili relazioni:

$$\mathbf{q}_m = \mathbf{K}_r \mathbf{q} \quad (7.6)$$

$$\boldsymbol{\tau}_m = \mathbf{K}_r^{-1} \boldsymbol{\tau} \quad (7.7)$$

Dove quest'ultima deriva (intuitivamente) dal fatto che con l'aumentare del *gear-ratio* si diminuisce la velocità angolare ma si aumenta il torque (e viceversa).

7.3 Controllo decentralizzato a giunti indipendenti

Decentralized Joint Space Control

Nel controllo decentralizzato a giunti indipendenti **ogni** singolo **giunto** attuato è considerato come un **sottosistema SISO** disaccoppiato e indipendente, descritto da un modello dinamico approssimato. Gli **effetti di accoppiamento** non-lineari presenti nella dinamica propria del robot sono **considerati come disturbi**.

Lo schema di controllo complessivo è formato da n controllori (uno per ogni giunto), basati su reti di compensazione classiche, ciascuno agente in modo indipendente dagli altri.

Ricordiamo che il nostro obiettivo è quello di far eseguire una sequenza $\mathbf{q}(t)$ in modo che $\mathbf{q}(t) \rightarrow \mathbf{q}_d(t)$.

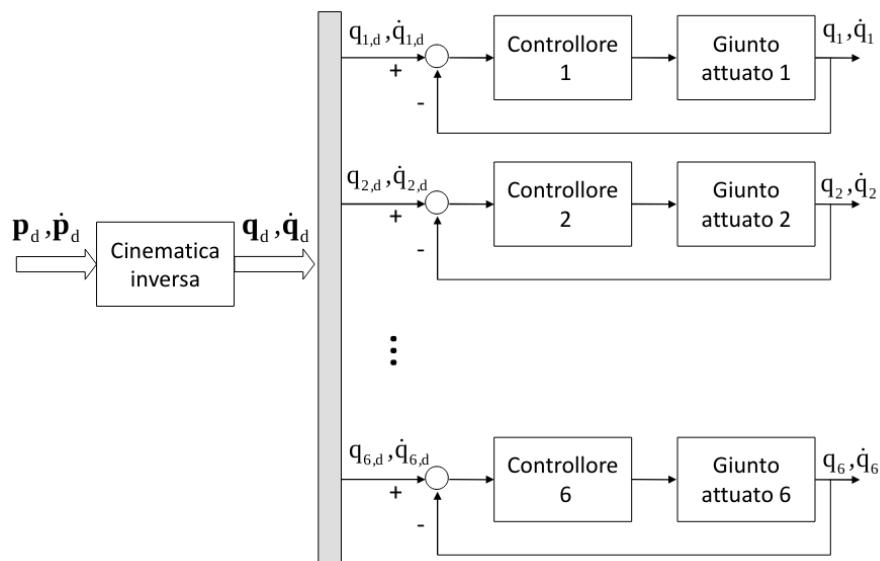


Figura 7.9: Schema generale del controllo a giunti indipendenti

Come accennato in precedenza, in questo caso andremo a controllare gli attuatori in velocità (questo esce in "automatico" dalle equazioni: vedi sezione 7.3.1).

Ora qua Rizzo ha detto roba che non ho capito. Lo stesso concetto si deriva comunque seguendo i prossimi discorsi. Per vedere quello che ha detto Rizzo vedi appendice A.1

Richiamando i capitoli precedenti, ricordiamo che il modello dinamico del manipolatore (in assenza di forze scambiate con l'ambiente esterno) è espresso da:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} = \mathbf{K}_r \boldsymbol{\tau}_m \quad (7.8)$$

L'ultima uguaglianza deriva dall'utilizzo dei motoriduttori (vedi (7.7)).

Per elaborare uno schema di controllo decentralizzato ai giunti è opportuno riportare le equazioni dinamiche a monte del motoriduttore (sul primario, i.e. lato motore):

$$\mathbf{K}_r^{-1}\mathbf{B}(\mathbf{q})\mathbf{K}_r^{-1}\ddot{\mathbf{q}}_m + \mathbf{K}_r^{-1}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_r^{-1}\dot{\mathbf{q}}_m + \mathbf{K}_r^{-1}\mathbf{F}_v\mathbf{K}_r^{-1}\dot{\mathbf{q}} + \mathbf{K}_r^{-1}\mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_m$$

(Questa equazione è ottenuta semplicemente applicando (7.6) e (7.7) a (7.8)).

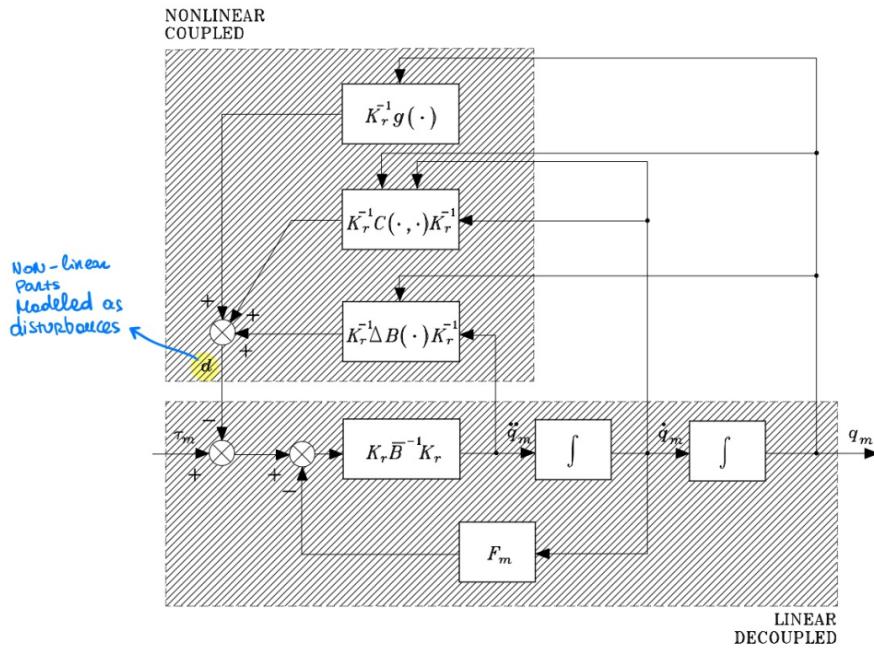


Figura 7.10: Circuito relativo a (7.10)

È possibile notare che $\mathbf{B}(q)$ può essere decomposto in $\mathbf{B}(q) = \bar{\mathbf{B}} + \Delta\mathbf{B}(q)$, ove $\bar{\mathbf{B}}$ è una matrice diagonale costante e $\Delta\mathbf{B}(q)$ è una matrice configuration-dependent. Possiamo quindi riscrivere l'ultima equazione come:

$$\begin{aligned} (\bar{\mathbf{B}} + \Delta\mathbf{B}(q))\mathbf{K}_r^{-1}\ddot{q} + \mathbf{C}(q, \dot{q})\mathbf{K}_r^{-1}\dot{q} + \mathbf{F}_v\mathbf{K}_r^{-1}\dot{q} + \mathbf{g}(q) &= \mathbf{K}_r\tau_m \\ \Downarrow \\ \mathbf{K}_r^{-1}\bar{\mathbf{B}}\mathbf{K}_r^{-1}\ddot{q}_m + d + \mathbf{K}_r^{-1}\mathbf{F}_v\mathbf{K}_r^{-1}\dot{q}_m &= \tau_m \end{aligned}$$

dove possiamo considerare gli effetti degli altri giunti su quello corrente come **disturbi**:

$$d \triangleq \mathbf{K}_r^{-1}\Delta\mathbf{B}(q)\mathbf{K}_r^{-1}\ddot{q}_m + \mathbf{K}_r^{-1}\mathbf{C}(q, \dot{q})\mathbf{K}_r^{-1}\dot{q}_m + \mathbf{K}_r^{-1}\mathbf{g}(q) \quad (7.9)$$

Inoltre, visto che questi termini sono moltiplicati per \mathbf{K}_r^{-1} , **più alto è il gear-ratio e meno questi termini influenzano il nostro sistema** ($d \downarrow$ quando $\mathbf{K}_r \uparrow$), portandoci così ad un sistema sempre più lineare e disaccoppiato (ovviamente c'è un limite, visto che gear-ratio altissimi non produrrebbero praticamente alcune velocità).

Riassumendo:

$$\mathbf{K}_r^{-1}\bar{\mathbf{B}}\mathbf{K}_r^{-1}\ddot{q}_m + \mathbf{F}_m\dot{q}_m + d = \tau_m \quad (7.10)$$

dove $\mathbf{F}_m \triangleq \mathbf{K}_r^{-1}\mathbf{F}_v\mathbf{K}_r^{-1}$.

Detto questo, per la parte lineare possiamo ora far riferimento alla nota teoria del controllo LTI per sistemi SISO.

7.3.1 Controllo del singolo giunto

Partendo da (7.10) possiamo estrarre la relazione per un singolo giunto:

$$\Gamma\ddot{q}_m + \beta\dot{q}_m + d = \tau_m \quad (7.11)$$

dove Γ e β sono rispettivamente il momento di inerzia totale equivalente ed il coefficiente di attrito viscoso totale equivalente, definiti come segue¹:

$$\Gamma = \frac{1}{\mathbf{K}_{r_i}^2} \bar{\mathbf{B}}_{ii} \quad , \quad \beta = \frac{1}{\mathbf{K}_{r_i}^2} \mathbf{F}_{mi},$$

Passando a Laplace (con condizioni iniziali nulle) otteniamo:

$$(s\Gamma + \beta)\omega_m(s) = \tau_m(s) - d(s) \quad (7.12)$$

considerando $\omega_n(s)$ come uscita, otteniamo il circuito di fig. 7.11: notare che questo **coincide col modello del motore in modalità velocity-generator** (fig. 7.6) con $C_i(s) = 1 \implies$ usiamo attuatore in modalità velocity-controlled.

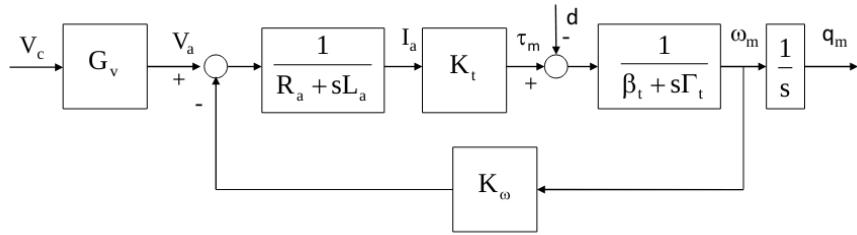


Figura 7.11: Schema a blocchi del singolo giunto

Cerchiamo ora di semplificare un po' questo modello. Possiamo iniziare supponendo L_a trascurabile, dato che le perdite ad essa associate sono solitamente molto piccole. Dall'equazione di equilibrio elettrico (7.1) possiamo quindi rimuovere L_a , e ottenere:

$$V_a - R_a I_a = K_\omega \omega_m \quad (7.13)$$

Allora, ricordando che $I_a = \tau_m K_t^{-1}$ e sostituendo τ_m con l'espressione (7.12) otteniamo

$$V_a - R_a K_t^{-1} \tau_m = K_\omega \omega_m \implies V_a - R_a K_t^{-1} ((s\Gamma + \beta)\omega_m + d) = K_\omega \omega_m$$

Riorganizzando i termini, e ignorando il termine relativo a β (visto che è trascurabile rispetto a ω_m), otteniamo:

$$\left(\frac{R_a \Gamma}{K_t K_\omega} s + 1 \right) \omega_m = \frac{1}{K_\omega} V_a - \frac{R_a}{K_t K_\omega} d$$

Da qui possiamo identificare 2 funzioni di trasferimento (a seconda di cosa consideriamo ingresso):

$$G_\omega(s) \triangleq \frac{\omega_m(s)}{V_a(s)} = \frac{1}{K_\omega(1+sT)} \quad (7.14)$$

$$G_d(s) \triangleq \frac{\omega_m(s)}{d(s)} = -\frac{T}{\Gamma(1+sT)} = -K_d G_\omega(s) \quad (7.15)$$

dove:

$$T = \frac{R_a \Gamma}{K_t K_\omega} \quad , \quad K_d = \frac{R_a}{K_t}$$

Da queste equazioni possiamo quindi passare allo schema a blocchi di fig. 7.12.

¹notando che $\mathbf{K}_{r_i}^{-1} \alpha \mathbf{K}_{r_i}^{-1} = (1/\mathbf{K}_{r_i}^2) \alpha$

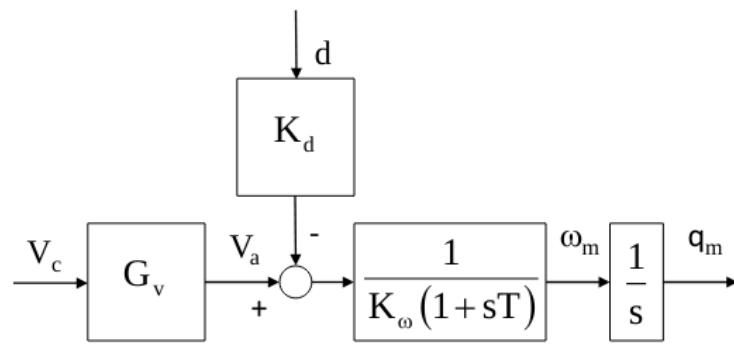
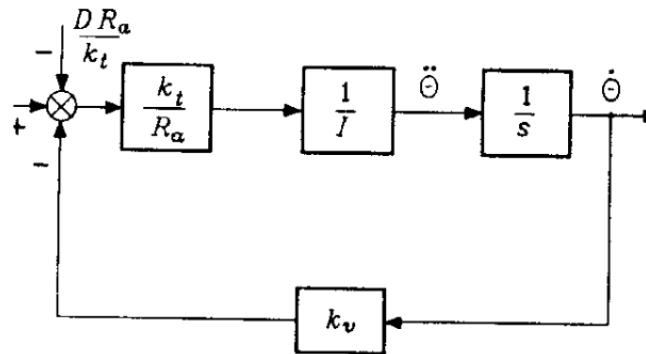


Figura 7.12: Schema a blocchi semplificato

Comparazione con le slide di Rizzo

Nelle slide è presente il seguente circuito. Anche se a prima vista potrebbe sembrare diverso è in realtà identico a quello di figura 7.12.



Ricordando la differente notazione: $I \equiv \Gamma$, $k_v \equiv K_\omega$, possiamo unire i 3 blocchi in uno unico. Partiamo dal forward path:

$$F(s) = \frac{k_t}{R_a s I}$$

Poi, incorporando il feedback, otteniamo

$$\frac{F(s)}{1 + F(s)k_v} = \frac{\frac{k_t}{R_a s I}}{1 + \frac{k_v k_t}{R_a s I}} = \frac{1}{k_v(1 + sT)}$$

Ovvero la stessa forma di quanto mostrato in fig. 7.12.

7.3.2 Struttura del controllo

In fig. 7.13 possiamo vedere lo schema di controllo generale, con retroazione su posizione, velocità e accelerazione (nota: per semplicità G_v è stato incluso nell'anello di controllo più interno).

In generale, il controllore viene progettato modo che si abbia un guadagno elevato nel blocco a monte del punto di ingresso del disturbo (per ottenere un elevato fattore di attenuazione) e in modo che ci sia un'azione integrale, affinché gli effetti della coppia di gravità vengano cancellati in regime permanente.

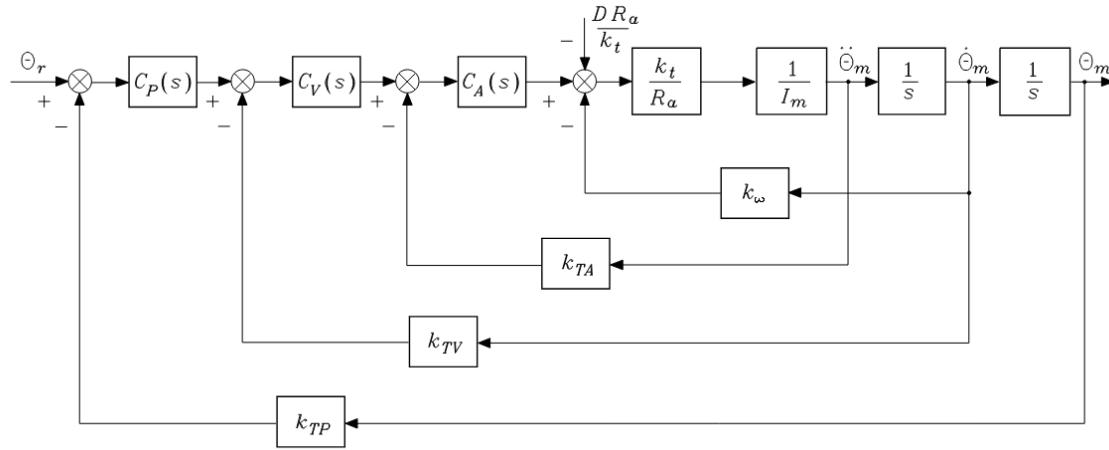


Figura 7.13: Schema generale del controllo

Queste richieste portano alla scelta di un **controllore PI (proporzionale-integrale)** della forma:

$$C(s) = K_p \frac{1 + sT_p}{s} \quad (7.16)$$

Retroazione sulla posizione

Possiamo partire dal caso più semplice, ovvero introducendo un controllo con solo la retroazione k_{TP} su θ_m . Poniamo $C_v(s) = C_A(s) = 1$ e $k_{TV} = k_{TA} = 0$, mentre mettiamo $C_p(s)$ nella forma (7.16).

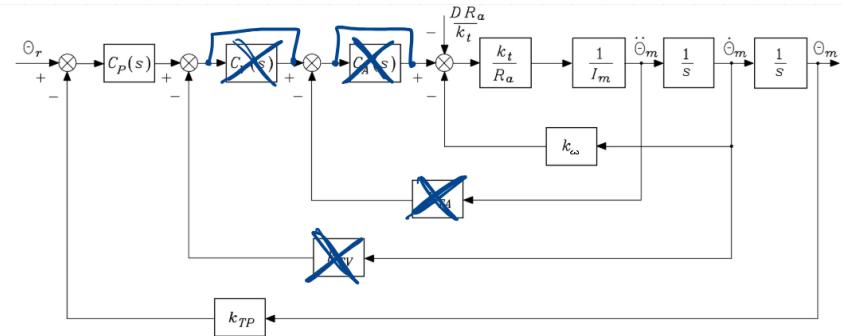


Figura 7.14: Solo position feedback

La funzione di trasferimento del ramo diretto risulta (richiamando (7.16) e (7.14)):

$$G(s) = \underbrace{K_p \frac{1 + sT_p}{s}}_{C_p(s)=(7.16)} \underbrace{\frac{1}{K_\omega(1 + sT)}}_{G_\omega(s)=(7.14)} \frac{1}{s} = \frac{K_p(1 + sT_p)}{s^2 K_\omega(1 + sT)}$$

da cui si ricava la funzione di trasferimento ad anello chiuso:

$$W(s) = \frac{K_p(1 + sT_p)}{s^2 K_\omega(1 + sT) + K_p K_{TP}(1 + sT_P)}$$

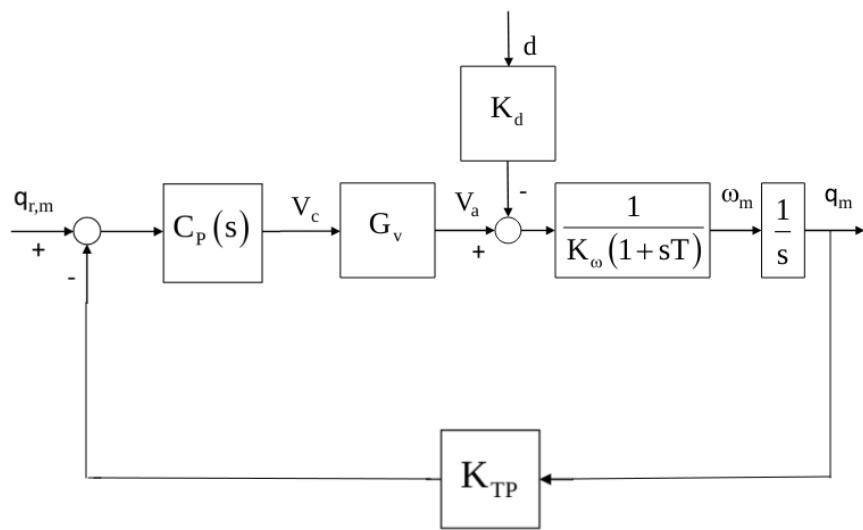


Figura 7.15: Circuito di fig. 7.14 con blocco motore accorpato ($q_{r,m} \equiv \theta_r$, $q_m \equiv \theta_m$)

Da qui è possibile analizzarne la stabilità, ad esempio tramite il luogo delle radici (vedi fig. 7.16).

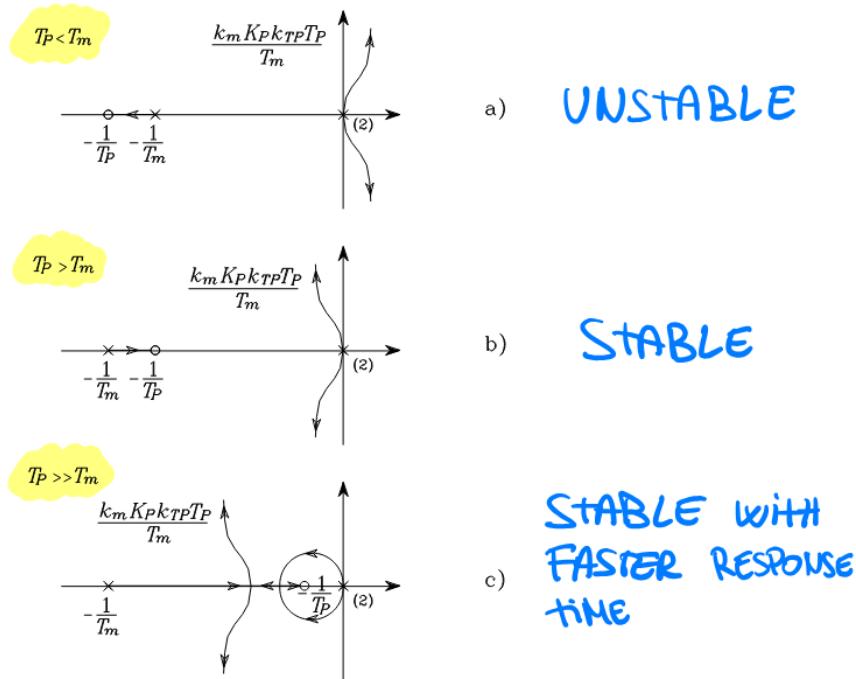


Figura 7.16: Root locus per $W(s)$ con solo retroazione sulla posizione

I parametri K_P e T_P devono essere scelti in modo da:

- Garantire l'asintotica stabilità del sistema in catena chiusa
- Evitare oscillazioni significative nella sua risposta
- Garantire un elevato fattore di attenuazione del disturbo

Per calcolare meglio i requirements è possibile esprimere $W(s)$ in funzione di ω_n e ζ :

$$W(s) = \frac{\frac{1}{K_{TP}}(1 + sT_P)}{\left(1 + \frac{2\zeta}{\omega_n} + \frac{s^2}{\omega_n^2}\right)(1 + \tau s)}$$

Come possiamo vedere anche dal root locus:

- Deve essere $T_P > T$ affinché si abbia asintotica stabilità
- Se $T_P \gg T$ si velocizza la risposta del sistema (diventano dominanti i poli complessi coniugati). Inoltre per elevati valori del guadagno K_P la pulsazione del polo reale di $W(s)$ tende a quella dello zero ($\tau \approx T_P$) e quindi il sistema risulta rappresentato in prima approssimazione dalla sola dinamica del II° ordine associata ai poli complessi coniugati
- La parte reale dei poli dominanti non può comunque essere inferiore a $-1/(2T)$

Applicando le regole dell'algebra dei blocchi, è possibile anche ricavare la funzione di trasferimento fra il disturbo D e l'uscita θ_m del sistema:

$$\frac{\Theta_m(s)}{D(s)} = -K_d \frac{\frac{s}{K_p K_{TP}(1+sT_p)}}{1 + \frac{s^2 K_\omega(1+sT)}{K_p K_{TP}(1+sT_p)}}$$

Da questa t.f. è possibile vedere che **il fattore di attenuazione del disturbo vale $K_p K_{TP}$** . Valori troppo elevati di K_p possono però portare ad avere oscillazioni inaccettabili sull'uscita (lo smorzamento ζ risulta troppo piccolo).

Il tempo necessario per avere un'attenuazione significativa del disturbo è approssimabile come $T_R = \max\{T_p, \frac{1}{\zeta \omega_n}\}$

Retroazione di posizione e velocità

Per questo secondo caso consideriamo il seguente setup (qua $C_A(s) = 1$, $k_{TA} = 0$), rappresentato in fig. 7.17:

$$C_P(s) = K_p \quad C_V(s) = K_V \frac{1 + sT_V}{s}$$

Ovvero un'azione solo proporzionale per la posizione, mentre un'azione PI per la velocità.

La costante di tempo T_V può essere scelta in modo da cancellare il polo meccanico del giunto attuato:

$$T_V = T$$

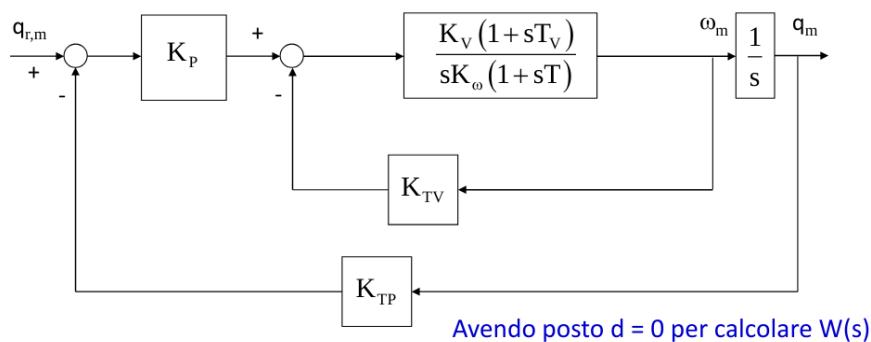


Figura 7.17: Circuito con retroazione su posizione e velocità

Come per il caso precedente, possiamo analizzare la stabilità del sistema tramite il root-locus della funzione di trasferimento in ciclo chiuso $W(s)$:

$$W(s) = \frac{K_p K_V}{K_\omega s^2 + K_V K_{TV} s + K_{TP} K_p K_V}$$

e gli effetti dei disturbi con la funzione di trasferimento $\Theta_m(s)/D(s)$ (saltando la definizione della forma esplicita): in questo caso il **fattore di attenuazione del disturbo** è dato dal prodotto $\mathbf{K}_P \mathbf{K}_{TV} \mathbf{K}_V$, ormai completamente definito, avendo imposto K_P e K_V per imporre i poli desiderati.

Aumentando il guadagno del feedback di posizione K_P , è possibile confinare i poli dell'anello chiuso in una regione del piano complesso con grandi valori assoluti della parte reale. Poi, la posizione effettiva può essere stabilita mediante una scelta adeguata di K_V .

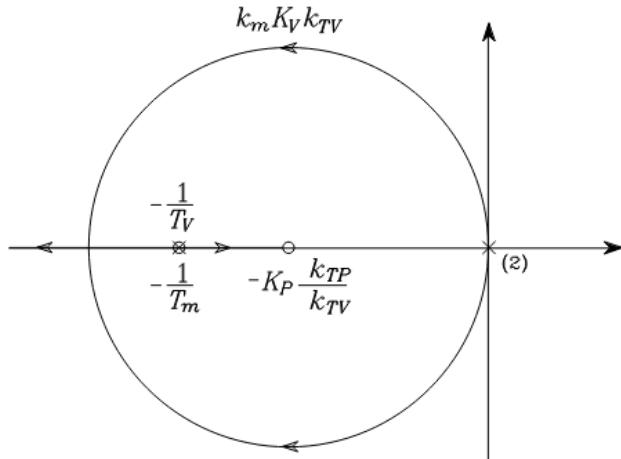


Figura 7.18: Root-locus con retroazione di velocità e posizione

Retroazione di posizione, velocità ed accelerazione

Per concludere vediamo il controllo con tutti e 3 i feedback:

$$C_P(s) = K_P \quad C_V(s) = K_V \quad C_A(s) = K_A \frac{1 + sT_A}{s}$$

Avendo a disposizione un parametro libero in più, sarebbe possibile in questo caso:

- Assegnare la dinamica desiderata al sistema ad anello chiuso
- Imporre il fattore di attenuazione del disturbo

Problema: La misura dell'**accelerazione però non è solitamente disponibile**. Per implementare uno schema di controllo comprendente la retroazione dell'accelerazione risulterebbe necessario ricavarla indirettamente dalle misure disponibili.

Avendo a disposizione la misura diretta della velocità, è possibile **stimare l'accelerazione** per mezzo di un filtro del primo ordine (vedi fig. 7.20), purché avente banda sufficientemente ampia. Scegliendo questa larghezza di banda sufficientemente ampia, gli effetti dovuti ai ritardi di misurazione non sono un problema, e quindi è possibile prendere l'uscita del filtro di accelerazione come quantità per il feedback. Problemi possono comunque incorrere a causa del rumore presente sul segnale di accelerazione così ottenuto.

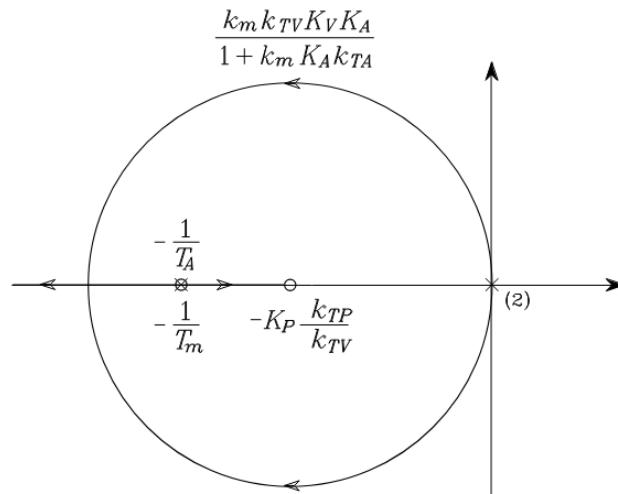


Figura 7.19: Root-locus con retroazione di accelerazione, velocità e posizione

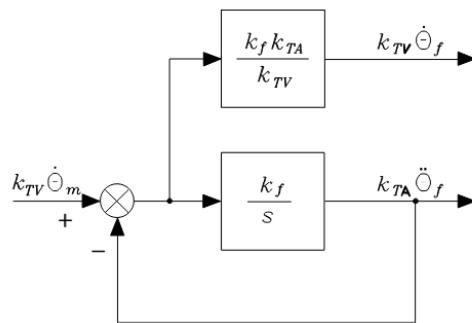


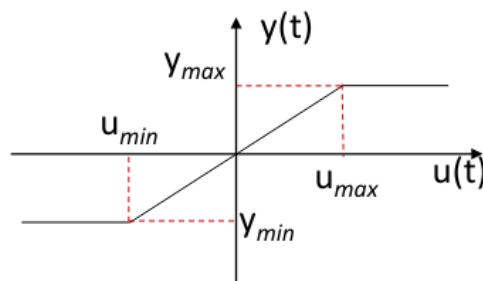
Figura 7.20: Filtro del I° ordine per la stima dell'accelerazione

7.3.3 Attuatori saturanti

In un'applicazione reale, il comportamento del sistema può allontanarsi da quello del suo modello teorico a causa di dinamiche “parassite” o non lineari, non incluse nella descrizione considerata, con evidenti conseguenze sulle prestazioni del controllo. Una di queste è ad esempio la presenza di attuatori saturanti.

Quest'ultima però può essere modellata semplicemente aggiungendo sul ramo diretto dell'anello di controllo un blocco che rappresenta la seguente relazione:

$$\begin{cases} y_{max} & , u(t) > u_{max} \\ ku(t) & , u_{min} \leq u(t) \leq u_{max} \\ y_{min} & , u(t) < u_{min} \end{cases}$$



L'inserimento di blocchi di **saturazione** è solitamente legato a **necessità di sicurezza e di salvaguardia** del sistema.

Quando le grandezze coinvolte (correnti, tensioni) raggiungono il valore massimo consentito ed entra in azione il blocco di saturazione, l'inseguimento della traiettoria non è più realizzato con le caratteristiche e l'accuratezza prevista. In caso di movimenti PTP non è inusuale che vengano raggiunte situazioni di saturazione: in tal caso infatti è solitamente prioritario raggiungere il più rapidamente possibile la configurazione finale desiderata, anziché rispettare la traiettoria prefissata in ogni istante.

7.3.4 Feedforward compensation

Quando è richiesto l'inseguimento di traiettorie con elevati valori di velocità e di accelerazione, è possibile ridurre l'errore di inseguimento utilizzando i valori del riferimento in velocità (ed in accelerazione) per calcolare termini di compensazione in avanti (feedforward compensation), da sommare all'azione del controllore posto sul ramo diretto.

Si noti che, in generale, calcolare derivate non è possibile (perchè dovremmo "predire il futuro"). In questo caso però essendo la traiettoria nota a priori, non ci sono problemi.

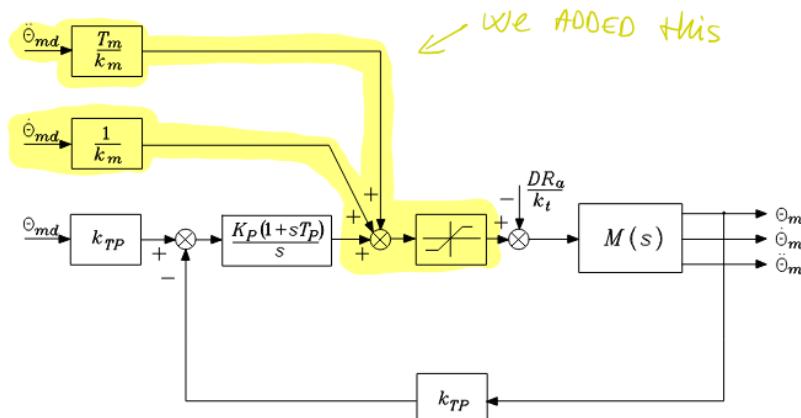


Figura 7.21: Schema a blocchi del controllo con feedback di posizione e feedforward compensation decentralizzato

7.3.5 Computed torque feedforward control

È possibile aggiungere un altro blocco di feedforward. Richiamando (7.10):

$$\mathbf{K}_r^{-1} \bar{\mathbf{B}} \mathbf{K}_r^{-1} \ddot{\mathbf{q}}_m + \mathbf{F}_m \dot{\mathbf{q}}_m + \mathbf{d} = \boldsymbol{\tau}_m$$

se vogliamo reiettare perfettamente il disturbo d potremmo provare a calcolarci in anticipo il valore di d , per poi sommarlo nel lato destro dell'equazione.

$$\mathbf{K}_r^{-1} \bar{\mathbf{B}} \mathbf{K}_r^{-1} \ddot{\mathbf{q}}_m + \mathbf{F}_m \dot{\mathbf{q}}_m + \mathbf{d} = \boldsymbol{\tau}_m + \mathbf{d}$$

Ponendo $q \equiv q_d$ ci basta usare (7.9) per calcolarci d (calcolo fatto offline).

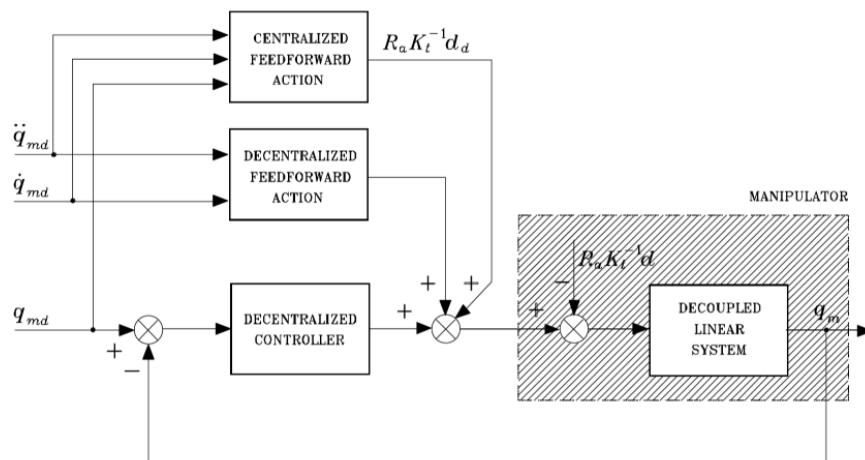


Figura 7.22: Circuit with computed torque feedforward control

7.4 Controllo centralizzato ai giunti

Centralized Joint Space Control

In alcuni casi il **controllo decentralizzato** a giunti indipendenti **può risultare inadeguato**. Questo accade, ad esempio, quando:

- sono **richieste velocità operative elevate** \Rightarrow le coppie di disturbo strutturato dipendenti dalla velocità (di Coriolis, centrifughe) influiscono pesantemente sul comportamento dei giunti
- i **motori sono ad azione diretta** \Rightarrow a causa dell'assenza dei motoriduttori ($\mathbf{K}_r = \mathbf{I}$) non si beneficia di alcuna riduzione degli effetti non lineari e di accoppiamento fra i giunti

In tali casi gli effetti delle coppie di disturbo \mathbf{d} possono determinare errori troppo elevati di inseguimento della traiettoria.

Poiché in questi casi non è possibile ridurre sufficientemente gli effetti indotti dalle coppie di disturbo \mathbf{d} , diventa conveniente cercare di eliminare direttamente tali coppie, ricorrendo ad una **strategia di controllo contenente termini non lineari di compensazione**.

Si parla di **controllo centralizzato** perché la coppia applicata a ciascun giunto risulta funzione anche delle variabili di posizione e velocità degli altri giunti, a differenza di quanto accade nel controllo a giunti indipendenti.

Negli schemi di controllo centralizzato il manipolatore è considerato come un **unico sistema MIMO**, con n ingressi (le coppie applicate ai giunti) e n uscite (le posizioni dei giunti) che interagiscono fra loro secondo relazioni non lineari. La **legge di controllo** centralizzato dovrà tenere conto del modello dinamico del manipolatore ed essere **non-lineare** (visto che il modello è non lineare).

Ora usiamo attuatori torque-controlled

Come accennato all'inizio in questo caso useremo motori in modalità generatori di coppia. Intuitivamente questo segue dal fatto che gli attuatori dovranno fornire coppie tali da compensare “esattamente” (o quanto meglio possibile) i disturbi strutturati: da questo si conclude che è opportuno utilizzare i motori come generatori di coppia anziché di velocità.

Richiamando la forma semplificata del bilancio elettrico di armatura (7.13), possiamo scrivere:

$$I_a = R_a^{-1}(V_a - K_\omega \omega_m) = R_a^{-1}(G_v V_c - K_\omega \omega_m)$$

Dove l'ultima ugualanza deriva dall'amplificatore di potenza: $V_a = G_v V_c$. Di conseguenza, ricordando (sempre al modello del motore: vedi fig. 7.3), che $\tau_m = K_t I_a$, possiamo scrivere:

$$\tau = K_r \tau_m = K_r K_t I_a = K_r K_t R_a^{-1}(G_v V_c - K_\omega \omega_m)$$

che si può notare essere uguale a (A.1.1).

Visto che però vogliamo avere un motore torque-controlled è necessario inserire la corrente in retroazione. Per magimonia (forse vedi sezione 7.1.4) diciamo che il termine con K_ω diventa trascurabile e quindi:

$$I_a = R_a^{-1}(G_v V_c - \cancel{K_\omega \omega_m}) = G_i V_c$$

Di conseguenza

$$\tau = \mathbf{K}_r \mathbf{K}_t \mathbf{G}_i \mathbf{V}_c = \mathbf{u}$$

ove \mathbf{u} è il vettore di comandi disponibili per il controllo.

7.4.1 Controllo a dinamica inversa

Il principale approccio di controllo centralizzato è detto a **dinamica inversa**, ed è basato sull'idea di ottenere una **linearizzazione della dinamica del sistema per mezzo di una retroazione non lineare** degli stati tale da condurre ad un sistema lineare e disaccoppiato rispetto ad un nuovo vettore di accelerazioni di comando da progettare.

Detto in altre parole, applichiamo la tecnica nota come **feedback linearization**.

Prima di cominciare, per comodità definiamo il modello della dinamica come:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \tau = \mathbf{u} \quad (7.17)$$

dove:

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \triangleq \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

Supponendo di poter calcolare esattamente $\mathbf{B}(\mathbf{q})$ e $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$, definiamo il nostro comando di controllo come

$$\mathbf{u} = \mathbf{B}(\mathbf{q})\mathbf{y} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$$

questo perchè, così facendo, riusciamo ad ottenere una **linearizzazione esatta del sistema**:

$$\begin{aligned} \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \cancel{\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})} &= \tau = \mathbf{u} = \mathbf{B}(\mathbf{q})\mathbf{y} + \cancel{\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})} \\ &\Updownarrow \\ \cancel{\mathbf{B}(\mathbf{q})}\ddot{\mathbf{q}} &= \cancel{\mathbf{B}(\mathbf{q})}\mathbf{y} \\ &\Updownarrow \\ \ddot{\mathbf{q}} &= \mathbf{y} \end{aligned}$$

Il sistema risultante è costituito da n doppi integratori: la i-esima componente \mathbf{y}_i del nuovo comando influenza solo il comportamento della i-esima coordinata giunto \mathbf{q}_i , che risulta indipendente dal moto degli altri giunti.

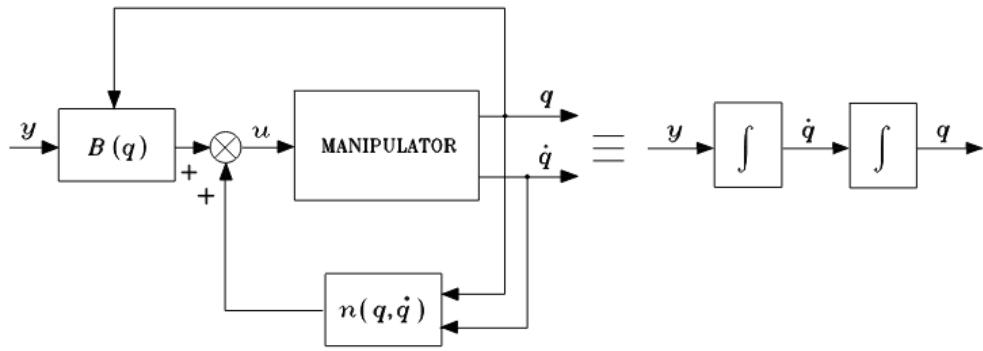


Figura 7.23: Feedback linearization

La scelta più semplice per il nuovo comando y è data da una legge di **controllo di tipo PD**:

$$y = -K_P q - K_D \dot{q} + r$$

che quindi, sostituendo a $\ddot{q} = y$, otteniamo il seguente sistema di equazioni del secondo ordine:

$$\ddot{q} + K_D \dot{q} + K_P q = r \quad (7.18)$$

asintoticamente stabile se le matrici K_P e K_D sono definite positive. Scegliendo in particolare K_P e K_D diagonali, il sistema rimane disaccoppiato e ad ogni variabile giunto viene assegnata la dinamica corrispondente ai guadagni imposti con:

$$K_P = \text{diag}\{\omega_{n1}^2, \dots, \omega_{nn}^2\} \quad K_D = \text{diag}\{2\zeta\omega_{n1}, \dots, 2\zeta\omega_{nn}^2\}$$

Il vettore di riferimento r è definito a partire dalla traiettoria desiderata, **pianificata ai giunti**, come:

$$r \triangleq \ddot{q}_d + K_D \dot{q}_d + K_P q_d$$

Possiamo ora analizzare la dinamica dell'errore di inseguimento, sostituendo la definizione di r a (7.18):

$$(\ddot{q}_d - \ddot{q}) + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) = 0 \implies \ddot{e} + K_D \dot{e} + K_P e = 0$$

se $e(0) = 0$ e $\dot{e}(0) = 0$, l'errore converge a 0 con velocità e caratteristiche determinate da K_P e K_D .

Il sistema risultante è mostrato in fig. 7.24, dove notiamo due anelli: quello più interno è l'anello di linearizzazione, mentre quello esterno è il controllo (ora lineare) del nostro sistema.

Problemi

L'ipotesi di linearizzazione esatta presuppone la capacità di calcolare *online* le matrici $B(q)$ e $n(q, \dot{q})$ in modo esatto. Nella pratica però questo non è sempre possibile. Per compenmare a questi errori è necessario utilizzare altre tecniche, ad esempio algoritmi di controllo robusto.

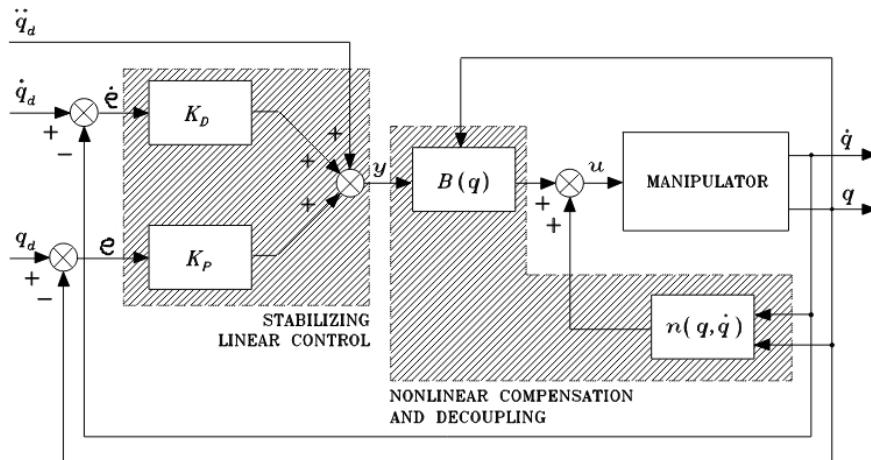


Figura 7.24: Schema a blocchi del controllo a dinamica inversa nello spazio dei giunti

Supponiamo di avere solo delle approssimazioni delle matrici, che chiameremo $\hat{B}(q)$ e $\hat{n}(q, \dot{q})$:

$$\begin{aligned}
 B(q)\ddot{q} + n(q, \dot{q}) &= \hat{B}(q)y + \hat{n}(q, \dot{q}) \\
 &\Updownarrow \\
 \ddot{q} &= B^{-1}\hat{B}y + B^{-1}(\hat{n} - n) \\
 &\Updownarrow \\
 \ddot{q} &= y + (B^{-1}\hat{B} - I)y + B^{-1}(\hat{n} - n) \\
 &\Updownarrow \\
 \ddot{q} &= y - \eta(q, \dot{q}, y) \\
 &\Downarrow \\
 \ddot{e} + K_D\dot{e} + K_Pe &= \eta(q, \dot{q}, y)
 \end{aligned}$$

dove

$$\eta(q, \dot{q}, y) \triangleq (I - B^{-1}\hat{B})y - B^{-1}(\hat{n} - n)$$

Il sistema ottenuto non è più lineare e disaccoppiato.

Per fortuna, **nella pratica**, il metodo della dinamica inversa **risulta abbastanza robusto** anche nel caso di linearizzazione approssimata, per cui viene usato sia senza ulteriori modifiche, sia con l'aggiunta di un termine robustificante.

Vari schemi visti come dinamica inversa

Tra gli schemi a dinamica inversa con linearizzazione approssimata, che non prevedono l'inserimento di termini robustificanti, possiamo includere:

- **Controllo a giunti indipendenti:** questo approccio precedentemente analizzato può essere considerato come un caso particolare di controllo a dinamica inversa con linearizzazione approssimata, ottenuto ponendo:

$$\hat{B}(q) = \bar{B} \quad , \quad \hat{n}(q, \dot{q}) = 0$$

- **Metodo della coppia calcolata** (o controllo a dinamica inversa in feedforward): visto in sezione 7.3.5

- **Controllo PD con compensazione della gravità**, che vediamo nella prossima sezione; questo controllo può essere visto come dinamica inversa ponendo:

$$\hat{\mathbf{B}}(\mathbf{q}) = \mathbf{I} \quad , \quad \hat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q})$$

7.4.2 Controllo PD con compensazione della gravità

Nel caso in cui si voglia assegnare al manipolatore una **postura di equilibrio costante** (anziché una traiettoria completa), è possibile realizzare un controllore tale da garantire la stabilità asintotica globale di tale postura.

Facendo riferimento a (7.17): $\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}$, definiamo il controllo come:

$$\mathbf{u} \triangleq \mathbf{g}(\mathbf{q}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_D\dot{\mathbf{q}}$$

Così facendo, sostituendo \mathbf{u} a (7.17), otteniamo:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{g}(\mathbf{q}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_D\dot{\mathbf{q}} \quad (7.19)$$

Ovvero si riescono a cancellare gli effetti della gravità.

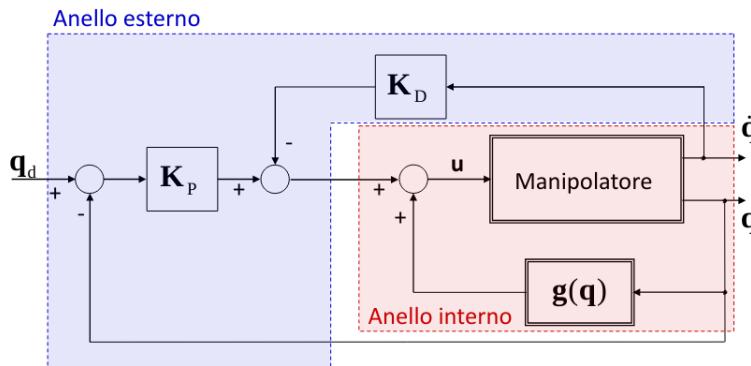


Figura 7.25: Controllo PD con compensazione della gravità

(notare che nel circuito non è presente $\dot{\mathbf{q}}_d$: questo poiché essendo \mathbf{q}_d costante la sua derivata è nulla)

Analizzando il sistema risultante da (7.19), possiamo vedere che l'unico punto di equilibrio è dato da $\mathbf{q} \equiv \mathbf{q}_d$:

$$\begin{aligned} x \text{ punto di equilibrio} &\iff \dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{pmatrix} = 0 \\ \text{sostituendo a (7.19)} &\iff \mathbf{B}(\mathbf{q})\overset{0}{\cancel{\dot{\mathbf{q}}}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\overset{0}{\cancel{\dot{\mathbf{q}}}} + \mathbf{F}_v\overset{0}{\cancel{\dot{\mathbf{q}}}} = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_D\overset{0}{\cancel{\dot{\mathbf{q}}}} \\ &\iff 0 = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) \\ \text{essendo } K_P > 0 &\iff q = q_d \end{aligned} \quad (7.20)$$

Ovvero proprio quello che volevamo: stabilità in una postura costante \mathbf{q}_d .

Per fare un esempio, immagina il robot in figura 7.26: il motori dovranno applicare delle coppie (i quali valori dipendono dal valore di \mathbf{u} mostrato prima) per far rimanere il braccio in questa posizione statica.

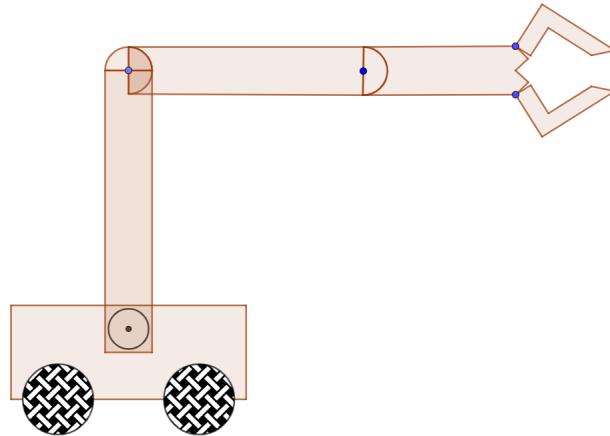


Figura 7.26: Esempio di compensazione gravità

Dimostrazione stabilità

Per dimostrare le caratteristiche di stabilità del punto di equilibrio corrispondente alla postura desiderata, è possibile applicare il **metodo diretto di Lyapunov**. Definiamo $e \triangleq \mathbf{q}_d - \mathbf{q}$ e consideriamo come funzione di Lyapunov:

$$V(\dot{\mathbf{q}}, e) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2} e^T \mathbf{K}_P e > 0 \quad \forall \mathbf{q}, e \neq 0$$

Dove possiamo interpretare il primo termine come l'energia cinetica del sistema ed il secondo come l'energia potenziale immagazzinata grazie alle rigidezze equivalenti date dai guadagni delle retroazioni delle posizioni ai giunti.

In questa funzione è stato scelto come stato del sistema il vettore $(e, \dot{\mathbf{q}})^T$ che, dal punto di vista dell'equilibrio è equivalente a quanto specificato in (7.20):

$$\gamma \triangleq \begin{pmatrix} e \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{q} - \mathbf{q}_d \\ \dot{\mathbf{q}} \end{pmatrix} \implies \dot{\gamma} = \begin{pmatrix} \dot{e} \\ \ddot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} - \dot{\mathbf{q}}_d^0 \\ \ddot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{pmatrix}$$

Ora passiamo allo studio del segno di \dot{V} lungo le traiettorie del sistema (come è richiesto dal metodo di Lyapunov). Derivando (ricordando che $\dot{\mathbf{q}}_d = \mathbf{0}$, essendo \mathbf{q}_d costante), otteniamo:

$$\dot{V} = \dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}} \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{K}_P e$$

procediamo sostituendo a $\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}}$ la forma di esso fornita da (7.19) (ricordando la definizione di $e \triangleq \mathbf{q}_d - \mathbf{q}$):

$$\begin{aligned} \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} &= \mathbf{K}_P e - \mathbf{K}_D \dot{\mathbf{q}} \\ \Updownarrow \\ \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} &= -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{K}_P e - \mathbf{K}_D \dot{\mathbf{q}} \\ \Downarrow \\ \dot{V} &= \dot{\mathbf{q}}^T (-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{K}_P e - \mathbf{K}_D \dot{\mathbf{q}}) + \frac{1}{2} \dot{\mathbf{q}} \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{K}_P e \end{aligned}$$

Raccogliendo i termini rimasti si ha

$$\dot{V} = \frac{1}{2} \dot{q}^T (\cancel{\dot{B}(q)} - \cancel{2C(q, \dot{q})}) \dot{q} - \dot{q}^T (F_v + K_D) \dot{q}$$

dove la cancellazione a 0 è fatta grazie alla *null-property* di quella differenza (si può dimostrare dalle equazioni della dinamica).

Allora:

$$\dot{V} = -\dot{q}^T (F_v + K_D) \dot{q}$$

che risulta essere **semi-definita negativa**. È solo semi-definita poichè è presente solo \dot{q} : $\dot{V} = 0$ per $\dot{q} = 0$ ma per qualsiasi e .

Possiamo però ora osservare che comunque $\dot{V} = 0 \iff \dot{q} = 0$, e di conseguenza anche $\ddot{q} = 0$. Ricordando da (7.20) che $(\dot{q}, \ddot{q}) = (0, 0)$ è la condizione di equilibrio, possiamo dire che \dot{V} si annulla solo per l'unico punto di equilibrio del sistema. Dal teorema di *La Salle-Krasowski* risulta pertanto globalmente asintoticamente stabile, come desiderato.

Thm. *La Salle-Krasowski*

Supponi che un sistema dinamico sia rappresentato da $\dot{\mathbf{x}} = f(\mathbf{x})$, dove \mathbf{x} è il vettore dello stato e $f(0) = 0$.

Se prendiamo una funzione V tale che:

$$\begin{cases} V(\mathbf{x}) > 0 \\ \dot{V}(\mathbf{x}) \leq 0 \end{cases}$$

(i.e. V positiva e \dot{V} semi-definita negativa), **ma**

$$\dot{V}(\mathbf{x}) = 0 \iff \mathbf{x} = 0$$

allora abbiamo che l'origine $\mathbf{x} = 0$ è **asintoticamente stabile**. ◊

7.5 Controllo nello spazio operazionale

La **traiettoria** desiderata da applicare al manipolatore è **spesso pianificata nello spazio operazionale**, mentre gli schemi di controllo analizzati finora sono tutti definiti nello spazio dei giunti. In tutti questi casi, il riferimento di posizione ai giunti deve essere determinato a partire da quello cartesiano per mezzo della **cinematica inversa** (i riferimenti di velocità ed accelerazione sono spesso determinati per differenziazione numerica, dato che la cinematica inversa delle velocità e soprattutto delle accelerazioni può risultare troppo gravosa).

Una diversa soluzione al problema può essere trovata definendo lo **schema di controllo direttamente nello spazio operazionale** anziché in quello dei giunti.

Poiché le variabili disponibili (misurabili) sono quelle ai giunti (posizione ed eventualmente la velocità), è necessario in questo caso utilizzare la **cinematica diretta** per determinare le variabili cartesiane da confrontare con il riferimento desiderato. Si definiscono così anelli di controllo in retroazione in cui l'**inversione cinematica della traiettoria è sostituita dalla trasformazione delle coordinate**, racchiusa nell'anello di controllo stesso.

Tutti gli **schemi di controllo realizzati nello spazio operazionale sono “pesanti”** dal punto di vista computazionale: questo può portare a serie limitazioni sulla scelta del passo di campionamento e conseguentemente a degradate prestazioni del controllo. Nonostante questo, a differenza degli schemi di controllo nello spazio dei giunti (che sono solitamente idonei ad ottenere un buon controllo del moto), gli schemi di controllo nello spazio operazionale sono basilari nel caso in cui il manipolatore sia in **contatto con l’ambiente esterno** ed il compito da svolgere richieda (anche) il **controllo delle forze scambiate**.

Gli schemi di controllo nello spazio operazionale sono tipicamente riconducibili a due schemi generali:

- **Controllo a Jacobiano inverso**
- **Controllo a Jacobiano trasposto**

In entrambi i casi nel ramo in retroazione viene inserita la cinematica diretta per la determinazione di \mathbf{p} a partire da \mathbf{q} ; il vettore \mathbf{p} viene poi confrontato con il vettore \mathbf{p}_d della traiettoria desiderata pianificata, costruendo così il vettore dell’errore (o deviazione) nello spazio operazionale: $\Delta\mathbf{p} \triangleq \mathbf{p}_d - \mathbf{p}$ (dal punto di vista della notazione il Siciliano usa \mathbf{x} : $\mathbf{p} \equiv \mathbf{x}$).

7.5.1 Controllo a Jacobiano inverso

Negli schemi a Jacobiano inverso si presuppone che la deviazione $\Delta\mathbf{p}$ sia sufficientemente piccola e che quindi la corrispondente deviazione nello spazio dei giunti possa essere calcolata attraverso lo Jacobiano inverso come:

$$\Delta\mathbf{q} = J_a^{-1}(\mathbf{q})\Delta\mathbf{p}$$

Il sistema risultante può essere visto come un sistema meccanico con una molla generalizzata a n dimensioni nello spazio dei giunti, la cui rigidezza è data dalla matrice dei guadagni. Il ruolo di tale sistema è portare la deviazione $\Delta\mathbf{q}$ a zero. Se la matrice dei guadagni è diagonale, la molla generalizzata corrisponde a n elementi elastici indipendenti, uno per ogni giunto.

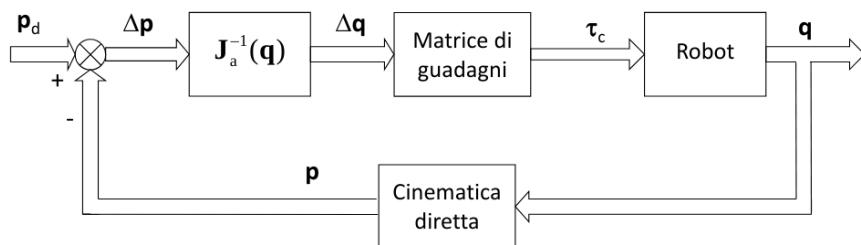


Figura 7.27: Schema a Jacobiano inverso

7.5.2 Controllo a Jacobiano trasposto

Negli schemi a Jacobiano trasposto l’errore $\Delta\mathbf{p}$ nello spazio operazionale viene direttamente moltiplicato per una matrice di guadagni. L’uscita di tale blocco può essere considerata come la forza elastica generata da una molla generalizzata, la cui funzione è di ridurre o cancellare la deviazione $\Delta\mathbf{p}$. In altre parole la forza risultante guida la punta operativa lungo una direzione tale ridurre $\Delta\mathbf{p}$.

La forza elastica definita nello spazio operazionale viene quindi trasformata nella corrispondente coppia da applicare ai giunti per mezzo dello Jacobiano trasposto (secondo le relazioni della statica viste).

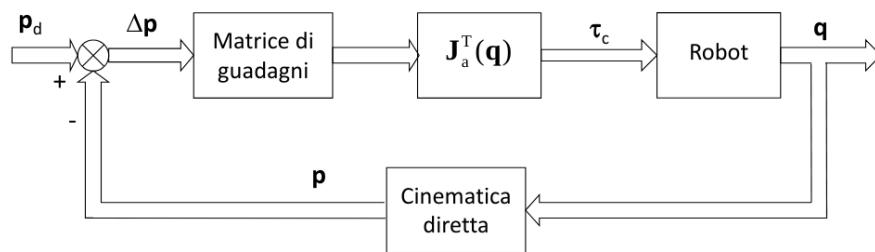


Figura 7.28: Schema a Jacobiano trasposto

Per questi schemi generali (sia questo che quello a Jacobiano inverso) **non è né garantita la stabilità asintotica** né l'accuratezza di inseguimento della traiettoria, che dovranno essere assicurate dall'impiego di specifiche leggi di controllo, inserite in schemi dell'uno o dell'altro tipo.

7.5.3 Controllo PD con compensazione della gravità

Vediamo nuovamente una tecnica di controllo PD con compensazione della gravità, questa volta però nello spazio operazionale.

Sia p_d la posa costante desiderata per la punta operativa; **obiettivo** del controllo è **garantire** che **asintoticamente** si abbia $p_d - p = 0$.

È possibile dimostrare che tale richiesta è soddisfatta dalla legge di controllo:

$$u = g(q) + J_A^T(q)(K_P(p_d - p) - K_D \dot{J}_A(q)\dot{q}) \quad (7.21)$$

costituita da una compensazione non lineare delle coppie di gravità, definita nello spazio dei giunti, e da una legge lineare di tipo PD definita nello spazio operazionale.

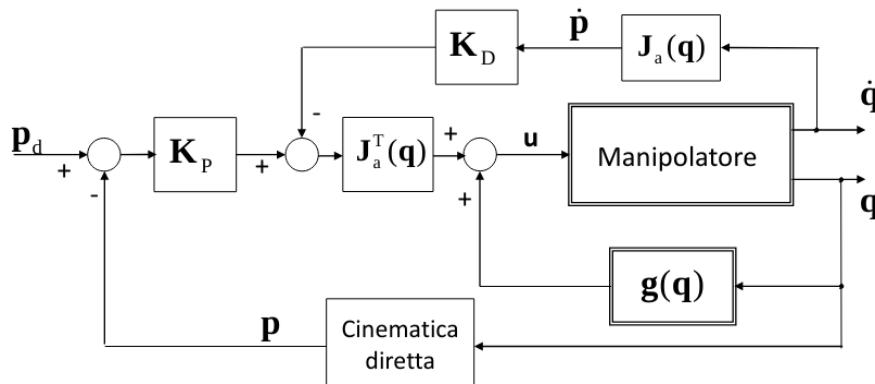


Figura 7.29: Schema circuitale controllo PD con compensazione della gravità

Dimostrazione stabilità

Con passaggi simili a quelli seguiti nello spazio dei giunti, si può dimostrare che il sistema raggiunge una postura di equilibrio asintoticamente stabile. Partiamo nuovamente selezionando una funzione di Lyapunov (di nuovo $e \triangleq p_d - p$):

$$V(\dot{q}, e) = \frac{1}{2} \dot{q}^T B(q) \dot{q} + \frac{1}{2} e^T K_P e > 0 \quad \forall \dot{q}, e \neq 0$$

e analizziamone il segno:

$$\dot{V} = \dot{q}^T B(q) \ddot{q} + \frac{1}{2} \dot{B}(q) \dot{q} + \dot{e} K_P e = \dot{q}^T B(q) \ddot{q} + \frac{1}{2} \dot{B}(q) \dot{q} - \dot{q}^T J_A^T(q) K_P e$$

dove abbiamo usato il fatto che $\dot{x}_d = 0$, $\dot{x} = J_A(q)\dot{q} \implies \dot{e} = \dot{x}_d - \dot{x} = -J_A(q)\dot{q}$. Raccogliendo, richiamando di nuovo la *null-property* di $\dot{B}(q) - 2C(q, \dot{q})$ e sostituendo a $B(q)\ddot{q}$ la forma di esso fornita da (7.19), otteniamo:

$$\dot{V} = -\dot{q}^T F \dot{q} + \dot{q}^T (u - g(q) - J_A^T(q) K_P e)$$

dove, se sostituiamo ad u la forma del controllo di (7.21), otteniamo:

$$\dot{V} = -\dot{q}^T F \dot{q} - \dot{q}^T J_A^T(q) K_D J_A(q) \dot{q}$$

e possiamo vedere che, per qualsiasi traiettoria del sistema, $\dot{V} \leq 0$ e $\dot{V} = 0 \iff \dot{q} = 0$ (allora, per La Salle-Krasowski abbiamo stabilità asintotica). Seguendo lo stesso ragionamento per la stabilità visto in sezione 7.5.3, possiamo dire che il sistema raggiunge la postura di equilibrio asintoticamente stabile corrispondente a:

$$J_A^T(q) K_P e = 0$$

e, sotto l'assunzione che lo Jacobiano abbia *full-rank*, risulta

$$e = 0$$

ovvero ciò che volevamo.

Possiamo infine notare che questo schema (analogo a quello visto nello spazio dei giunti), è un esempio di controllo nello spazio operazionale a Jacobiano trasposto.

7.5.4 Controllo a dinamica inversa

In sezione 7.4.1 avevamo visto il controllo a dinamica inversa nello spazio dei giunti, dove avevamo selezionato il controllo $u = B(q)y + n(q, \dot{q})$, che sotto l'ipotesi di linearizzazione esatta portava ad un sistema di n doppi integratori $\ddot{q} = y$ (Il comando dell'anello esterno y era stato quindi progettato per garantire l'inseguimento di una traiettoria q_d nello spazio dei giunti).

È possibile ora **riprogettare tale comando y direttamente nello spazio operazionale** per ottenere l'inseguimento di una traiettoria cartesiana p_d . Derivando la relazione della cinematica diretta delle velocità $\dot{p} = J_A(q)\dot{q}$ si ottiene l'espressione dell'accelerazione cartesiana:

$$\ddot{p} = J_A(q)\ddot{q} + \dot{J}_A(q, \dot{q})\dot{q}$$

che "suggerisce" di definire y come (ricordando che $e \triangleq p_d - p$):

$$y \triangleq J_A^{-1}(q)(\ddot{p}_d + K_D \dot{e} + K_P e - \dot{J}_A(q, \dot{q})\dot{q})$$

(il ragionamento per la scelta di questa forma è lo stesso fatto in sezione 7.4.1).

Con questa definizione di y infatti si ottiene:

$$\ddot{e} + K_D \dot{e} + K_P e = 0$$

e scegliendo K_P e K_D (diagonali) definite positive, si assegna la dinamica all'errore di inseguimento, garantendone la convergenza a zero.

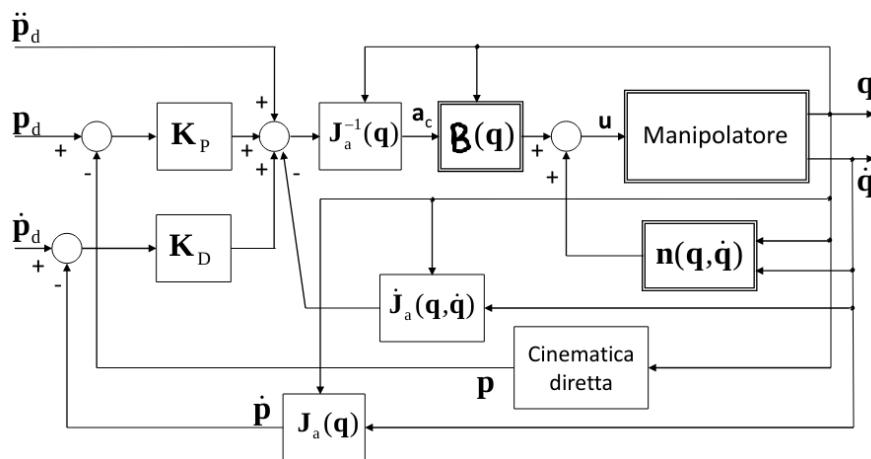


Figura 7.30: Controllo a dinamica inversa nello spazio operazionale

7.5.5 Considerazioni generali

Tutti gli schemi di **controllo** sviluppati **nello spazio operazionale richiedono** il calcolo dello **Jacobiano**, che è influenzato dalla **presenza di singolarità e di ridondanza**.

Ad esempio, in caso di passaggio per una configurazione singolare gli algoritmi di controllo potrebbero "bloccarsi" e "bloccare" il manipolatore in una configurazione diversa da quella desiderata (poiché $J_A^T(q)K_P e = 0$ ma con $e \neq 0$).

Inoltre, lo sviluppo degli schemi considerati presuppone l'utilizzo di una rappresentazione minima dell'assetto (data dalla terna di angoli, che compare nella definizione di p); in caso contrario si dovrebbe fare ricorso allo Jacobiano geometrico con un aumento della complessità dell'analisi di stabilità.

7.6 Controllo dell'interazione

Molti compiti prevedono il contatto della punta operativa con superfici esterne o la manipolazione di oggetti. Per compiere questo tipo di operazioni non è sufficiente controllare il moto del manipolatore, ma è necessario saper gestire la sua interazione con l'ambiente esterno.

La grandezza che descrive in modo efficace la condizione di interazione, e che deve essere controllata in modo diretto o indiretto, è quindi la **forza di contatto** sulla punta operativa. L'idea base per il controllo della forza è quella di agire sulla cedevolezza (**compliance**) del sistema robotambiente, in modo passivo o attivo.

7.6.1 Cedevolezza passiva ed attiva

Il sistema formato dal manipolatore e dall'ambiente esterno con cui è in contatto è rappresentato da una **molla generalizzata equivalente**, di cui si impone il coefficiente (matrice nel caso a più dimensioni) di rigidezza (ovvero il suo reciproco, cioè la cedevolezza).

Il **controllo** della cedevolezza può avvenire in maniera **passiva**, mediante l'inserimento di appositi dispositivi installati al polso del robot che reagiscono in modo opportuno a forze esterne.

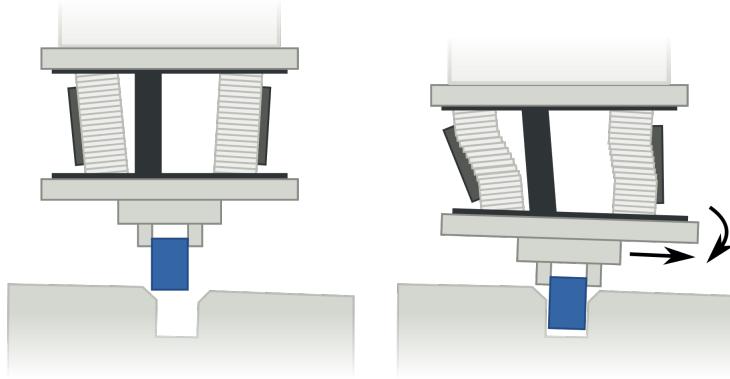


Figura 7.31: Esempio di *Remote Center of Compliance*

Ad esempio, Un **RCC** (*Remote Center of Compliance*) inserisce una cedevolezza passiva fra robot e superficie esterna, “disaccoppiando” le forze e le coppie di contatto: le forze generano solo traslazioni, le coppie solo rotazioni.

Tali dispositivi sono però poco versatili, di limitato utilizzo e non sono adatti ad applicazioni più evolute, che richiedano il controllo dei movimenti del robot mentre sono applicate forze controllate su oggetti ed elementi dell’ambiente esterno.

Si passa quindi ad un **controllo attivo** della cedevolezza, cioè ad un controllo attivo dell’interazione (il prezzo da pagare è ovviamente costituito da un aumento della complessità del controllo e necessità di conoscere (in modo più o meno accurato) le caratteristiche degli elementi con cui il robot entra in contatto).

7.6.2 Strategie di controllo dell’interazione

Le strategie di controllo dell’interazione possono essere distinte in due categorie principali:

- **Controllo indiretto della forza:** ottenuto attraverso uno schema di controllo del moto, senza la chiusura esplicita di un anello di forza. Tra le principali tecniche troviamo controllo della rigidezza (o della cedevolezza) e controllo di impedenza.
- **Controllo diretto della forza:** prevedono la chiusura di un anello di forza, in aggiunta ad un anello interno di posizione o di velocità; possono includere il contemporaneo controllo di posizione e forza lungo direzioni appropriate (controllo ibrido).

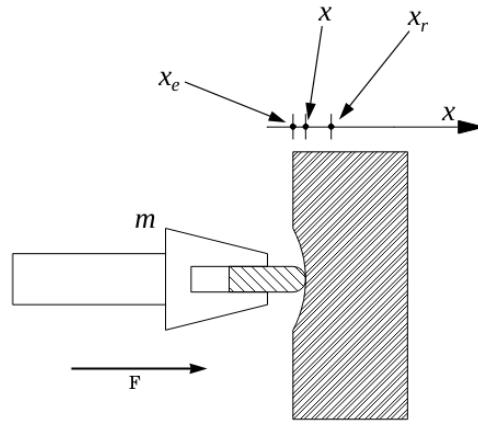
7.6.3 Controllo di rigidezza (*compliance control*)

Prima di vedere il caso generale, introduciamo un caso semplificato per aiutarci a capire.

Caso semplificato

Il concetto alla base del controllo di rigidezza può essere ben rappresentato nel caso semplificato ad 1 grado di libertà, ovvero sotto l’ipotesi che la forza applicata dalla punta del manipolatore sia diretta lungo la sola direzione perpendicolare alla superficie di contatto.

Partiamo supponendo che la rigidezza della superficie non sia infinita. A causa di tale cedevolezza la punta provoca una deformazione, anche se estremamente piccola, della superficie stessa:



Sia x la posizione della punta lungo la direzione di applicazione della forza, x_e la posizione della superficie “a riposo” e x_r la posizione di riferimento (interna alla superficie stessa e quindi non raggiungibile nella pratica). La forza applicata dal manipolatore è data da:

$$f_e = K_e(x - x_e)$$

ove K_e è il coefficiente di rigidezza dell’oggetto con cui il robot è in contatto.

L’equazione dinamica del sistema robot + ambiente esterno può essere scritta come (ovvero $F = ma = m\ddot{x}$ + legge di Hooke):

$$m\ddot{x} + K_e(x - x_e) = f_c$$

ove m è la massa equivalente del manipolatore nella direzione x e f_c è la forza di comando applicata (si suppone che la gravità non eserciti alcuna azione lungo la direzione x).

Si consideri quindi una legge di controllo di tipo puramente PD (per l’ipotesi introdotta non è richiesta compensazione della gravità lungo x)

$$f_c = K_P(x_r - x) - K_D\dot{x}$$

con x_r costante.

Sostituendo l’espressione di f_c nell’equazione dinamica del sistema si ottiene:

$$m\ddot{x} + K_e(x - x_e) = K_P(x_r - x) - K_D\dot{x} \implies m\ddot{x} + K_D\dot{x} + (K_e + K_P)x = K_e x_e + K_P x_r$$

Scegliendo K_D e K_P positivi, il sistema risulta asintoticamente stabile. In condizione di equilibrio $(\dot{x}, \ddot{x}) = (0, 0)$ si ha:

$$(K_e + K_P)x = K_e x_e + K_P x_r$$

da cui si ricava la **posizione** raggiunta dalla punta in condizioni di **equilibrio**:

$$\bar{x} = \frac{K_e x_e + K_P x_r}{K_e + K_P}$$

e la forza applicata in questo stato di equilibrio risulta quindi:

$$\bar{f}_e = K_e(\bar{x} - x_e)$$

Attraverso K_P e x_r è possibile assegnare il valore desiderato per la forza da applicare all’equilibrio. Si ottiene così un **controllo indiretto della forza** mediante le sole retroazioni di posizione e di velocità.

Oss.

Il guadagno K_D non ha alcuna influenza sul comportamento del sistema in regime permanente, ma solo sulla stabilità del sistema e sulla velocità con cui viene raggiunta la condizione di stazionarietà. Il valore della forza all'equilibrio dipende sia dalla posizione x_e dell'oggetto con cui il manipolatore è in contatto, sia dal suo coefficiente di rigidezza K_e . **Solo conoscendo il valore di questi parametri è possibile assegnare esattamente la forza all'equilibrio.**

Caso generale

Nel caso completo (a n gradi di libertà), possiamo rappresentare il legame fra le forze generalizzate applicate dal manipolatore \mathbf{F}_e e lo spostamento elementare come

$$\mathbf{F}_e = \begin{bmatrix} \mathbf{f} \\ \mathbf{n} \end{bmatrix} = \mathbf{K} \begin{bmatrix} d\mathbf{x} \\ \boldsymbol{\omega} dt \end{bmatrix}$$

ove \mathbf{K} è la matrice di rigidezza dell'ambiente, simmetrica e semi-definita positiva.

In assenza di accoppiamenti fra le componenti di traslazione e di rotazione, la matrice risulta diagonale, inoltre, è solo semi-definita positiva, perché l'ambiente potrebbe non resistere a forze lungo tutte le direzioni.

Il vettore dello spostamento elementare può essere trasformato nelle coordinate operazionali ($d\mathbf{p} = (d\mathbf{x}, d\boldsymbol{\alpha})^T$) ricordando il legame esistente fra lo Jacobiano geometrico e quello analitico, dato da:

$$\mathbf{J}(\mathbf{q}) = \mathbf{T}_A(\mathbf{p}) \mathbf{J}_A(\mathbf{q}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\alpha}) \end{bmatrix} \mathbf{J}_A(\mathbf{q}) \quad (7.22)$$

ove l'espressione di $T(\boldsymbol{\alpha})$ dipende dalla particolare rappresentazione scelta per l'assetto della punta. Possiamo quindi esprimere quindi \mathbf{F}_e come:

$$\mathbf{F}_e = \mathbf{K} \mathbf{T}_A(\mathbf{p}) d\mathbf{p} = \mathbf{K} \mathbf{T}_A(\mathbf{p}) (\mathbf{p} - \mathbf{p}_e) \quad (7.23)$$

ove \mathbf{p}_e è il vettore della posa dell'ambiente a riposo.

Le coppie generalizzate risultanti (presenti nel modello dinamico del manipolatore) sono quindi date da:

$$\tau_e = \mathbf{J}^T(\mathbf{q}) \mathbf{F}_e = \underbrace{\mathbf{J}_A^T(\mathbf{q}) \mathbf{T}_A^T(\mathbf{p})}_{\mathbf{J}^T(\mathbf{q})} \underbrace{\mathbf{K} \mathbf{T}_A(\mathbf{p}) (\mathbf{p} - \mathbf{p}_e)}_{\mathbf{F}_e}$$

Questa relazione può essere vista anche come:

$$\tau_e = \mathbf{J}_A^T(\mathbf{q}) \mathbf{K}_A(\mathbf{p}) (\mathbf{p} - \mathbf{p}_e)$$

dove

$$\mathbf{K}_A(\mathbf{p}) \triangleq \mathbf{T}_A^T(\mathbf{p}) \mathbf{K} \mathbf{T}_A(\mathbf{p}) \quad (7.24)$$

è la matrice di rigidezza (*rigidity*) dell'ambiente, espressa nelle coordinate operazionali (funzione delle coordinate stesse!). Se esiste, \mathbf{K}_A^{-1} è la matrice di cedevolezza (*compliance*).

Controllo

Consideriamo allora l'equazione dinamica del manipolatore, comprensiva del contributo delle forze scambiate con l'ambiente esterno (ponendo $e \triangleq p_d - p$):

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v\dot{q} + g(q) = u - J^T(q)F_e$$

e l'algoritmo di controllo PD con compensazione della gravità, sviluppato nello spazio operazionale:

$$u = g(q) + J_A^T(q)(K_P e - K_D J_A(q)\dot{q})$$

Unendo le due equazioni, e ponendoci in condizione di calcolo dell'equilibrio $(\dot{q}, \ddot{q}) = (0, 0)$, otteniamo:

$$\begin{aligned} \cancel{B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v\dot{q}} + g(q) &= g(q) + J_A^T(q) \left(K_P e - \cancel{K_D J_A(q)\dot{q}} \right) - J^T(q)F_e \\ &\Downarrow \\ J_A^T(q)K_P e &= J^T(q)F_e \\ &\Downarrow \\ e &= K_P^{-1}(J_A^T(q))^{-1}J^T(q)F_e \\ &\Downarrow \\ e &= K_P^{-1}T_A^T(p)F_e \end{aligned}$$

dove l'ultima implicazione è stata fatta richiamando (7.22): $J(q) = T_A(q)J_A(q) \implies T_A(q) = J(q)J_A^{-1}(q) \implies T_A^T(q) = (J_A^{-1}(q))^T J^T(q)$.

Poi, ricordando la definizione di F_e da (7.23) e quella di $K_A(p)$ da (7.24) possiamo scrivere:

$$e = K_P^{-1}T_A^T(p)F_e = K_P^{-1}T_A^T(p)K_T A(p)(p - p_e) = K_P^{-1}K_A(p)(p - p_e)$$

e quindi all'equilibrio:

$$\begin{aligned} e &= K_P^{-1}K_A(p)(p - p_e) \\ &\Downarrow \text{espandi } e \\ p_d - p &= K_P^{-1}K_A(p)(p - p_e) \\ &\Downarrow \text{sposta } p_d \text{ a destra} \\ p &= p_d - K_P^{-1}K_A(p)(p - p_e) \\ &\Downarrow \text{espandi la molt. con la parentesi a dx} \\ p &= p_d - K_P^{-1}K_A(p)p + K_P^{-1}K_A(p)p_e \\ &\Downarrow \text{isola } p \\ p + K_P^{-1}K_A(p)p &= p_d + K_P^{-1}K_A(p)p_e \\ &\Downarrow \\ (I + K_P^{-1}K_A(p))p &= p_d + K_P^{-1}K_A(p)p_e \end{aligned}$$

$$\Downarrow \\ \mathbf{p} = (\mathbf{I} + \mathbf{K}_P^{-1} \mathbf{K}_A(\mathbf{p}))^{-1} (\mathbf{p}_d + \mathbf{K}_P^{-1} \mathbf{K}_A(\mathbf{p}) \mathbf{p}_e)$$

da cui è possibile anche ricavare il vettore delle forze generalizzate $\bar{\mathbf{F}}_e$ applicate all'equilibrio, applicando il valore di \mathbf{p} trovato a (7.23). Quest'ultimo ($\bar{\mathbf{F}}_e$) è funzione di \mathbf{K}_P^{-1} , che può essere considerata come la cedevolezza attiva (*active compliance*) del robot, e della posa di riferimento \mathbf{p}_d , oltre che delle caratteristiche fisiche e geometriche dell'ambiente (\mathbf{K} e \mathbf{p}_e).

È possibile controllare la rigidezza \mathbf{K}_P del robot (o la sua cedevolezza \mathbf{K}_P^{-1}), in modo che "il robot prevalga sull'ambiente" o viceversa, lungo le diverse direzioni:

- valori elevati degli elementi diagonali di \mathbf{K}_P rendono rigido il robot lungo le direzioni in cui si vuole controllare la posizione
- mentre valori bassi degli elementi diagonali di \mathbf{K}_P rendono "cedevole" il robot lungo le direzioni in cui si vuole controllare la forza

Nelle slide in italiano ora qua fa il controllo di impedenza, controllo diretto della forza e moto vincolato, che però noi skippiamo.

Capitolo 8

Mobile Robotics

Vedi slide

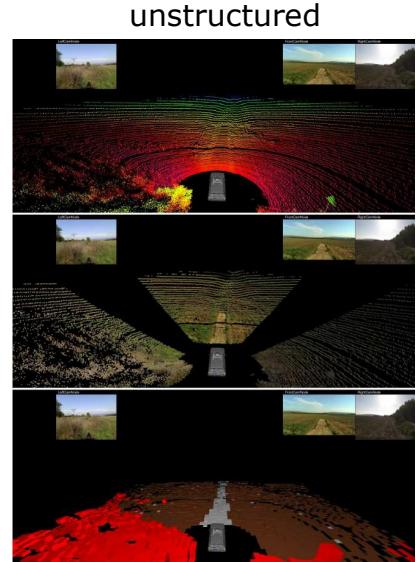
Definitions

- A mobile robot is a structure capable to move and act (autonomously or remotely operated) in terrestrial, underwater or aerial environments
- Environments can be assumed to be
 - totally structured, partially structured or unstructured
 - totally known, partially known or unknown
- Structured environment = one knows the type and the geometric characteristics of the environment
 - office space: corridors, doors, chairs, tables, etc.
 - obstacles: static or dynamics or both
 - constant or slow-varying in time

Autonomy

- Autonomy is the ability to move independently from a human supervisor
- It requires
 - Intelligence, i.e.,
 - CPU, algorithms, database
 - onboard or “in the cloud”
 - Sensors (for perceiving the environment)
 - Actuators (for motion and manipulation, etc.)
 - Energy source:
 - onboard generated
 - or supplied by an “umbilical cord”

Examples



Fundamental problems in mobile robotics

- **Locomotion**: how the robot moves in the environment
- **Perception**: how the robot perceives the environment
- **Representation**: how the robot organizes the knowledge about the environment
- **Mapping**: how to build the map of the environment
- **Localization**: where is the robot in the map
- **Path planning/action planning**: what the robot shall do to go from here to there; what are the actions to be performed to complete a specified task
- **Supervision and control**: how are the command to actuators generated to perform simple or complex tasks. How to generate tasks

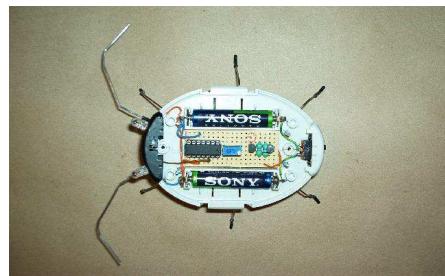
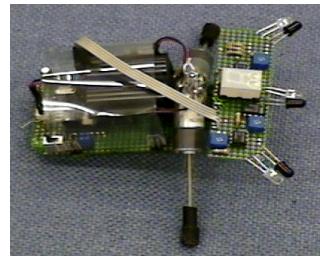
Locomotion

- Terrestrial robots
 - Wheeled
 - Legged
 - Mixed wheels/legs
 - Biped (humanoids)
 - Others (biomimetics = imitation of natural locomotion)
- Underwater robots
 - Propellers
 - Water jets
- Aerial robots
 - Fixed wings
 - Rotating wings (helicopters and quadcopters)
 - Airships and dirigibles
 - Flapping wings

Terrestrial Robots – wheeled



Terrestrial Robots – legged



Terrestrial Robots – “humanoids”



Underwater Robots

- To move they use
 - Propellers
 - Water jets
 - Fins or entire body motion



Aerial robots (UAV)

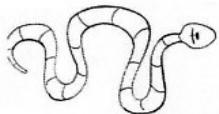
- Airship (lighter than air) or aircrafts (heavier than air)
- They use
 - Propellers
 - Rotating wings
 - Flapping wings



Natural Locomotion



Longitudinal waves



Transversal waves



Running



Jumping



Step

Biomimetic systems

Nature imitation



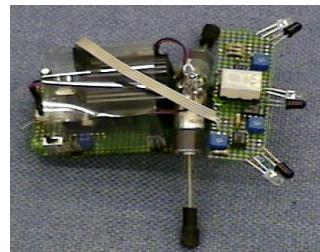
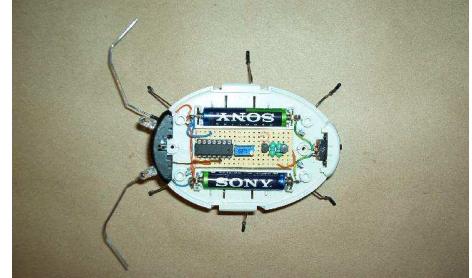
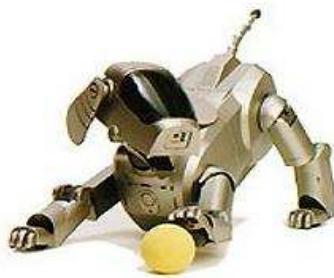
Longitudinal waves



Transversal waves

Biomimesis

Copy from nature



Mobile Robots: topics treated

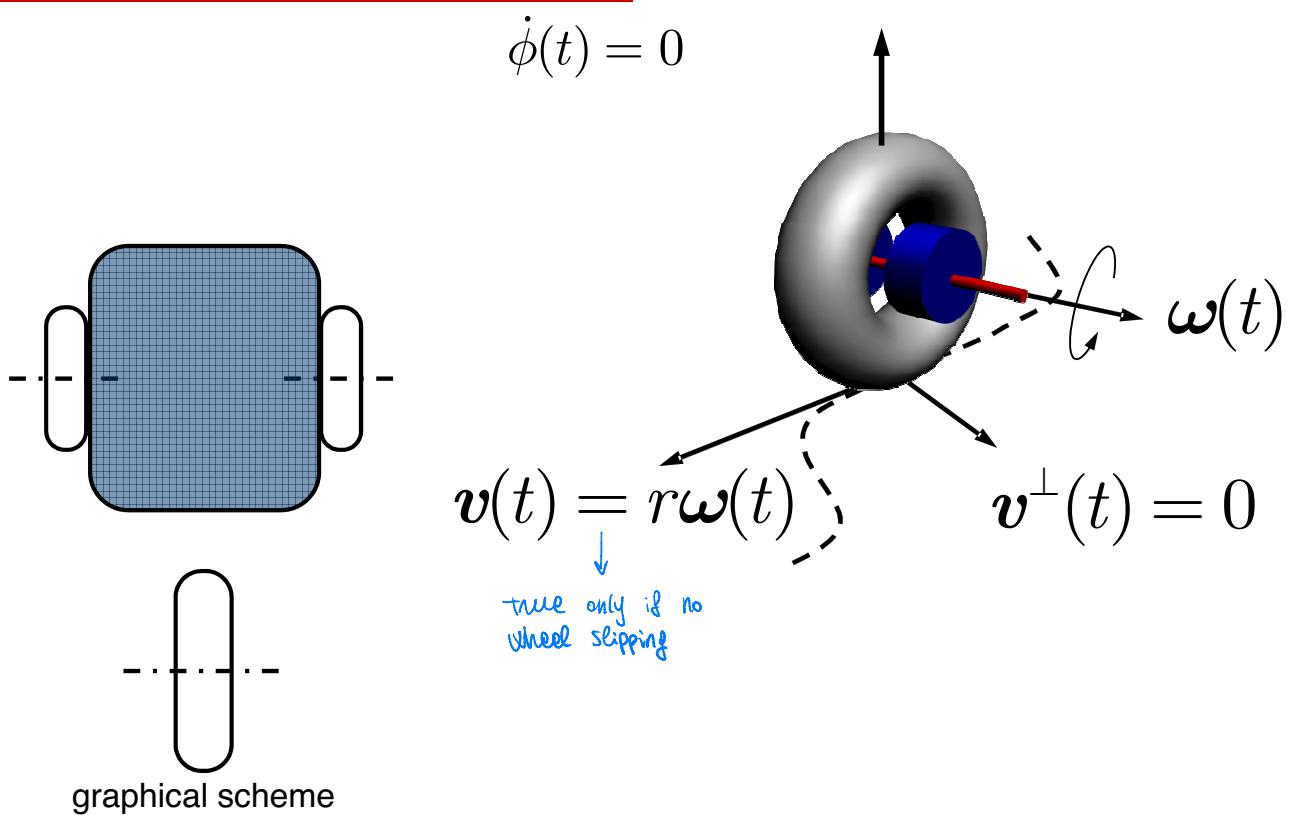
- Wheeled robots
- Kinematics
- Sensors
 - Absolute and relative position (odometry)
 - Speed
 - Proximity and distance
 - Active ranging
 - Vision
- Intelligence

Wheel Types

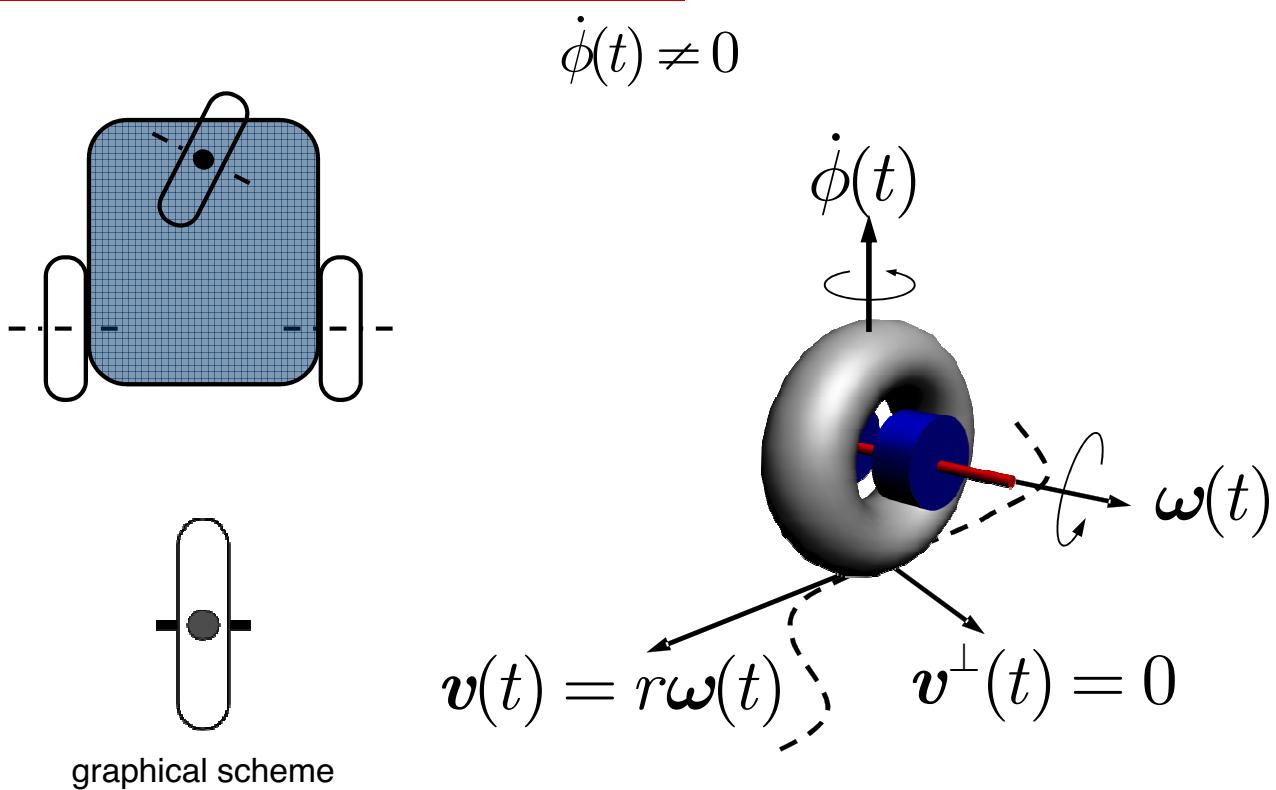
- Simple non steering wheels
- Simple steering wheels
- Castor wheel
- Omniwheel (omnidirectional wheel) or Swedish wheel
- Spherical omniwheel

- Wheel may be active or passive

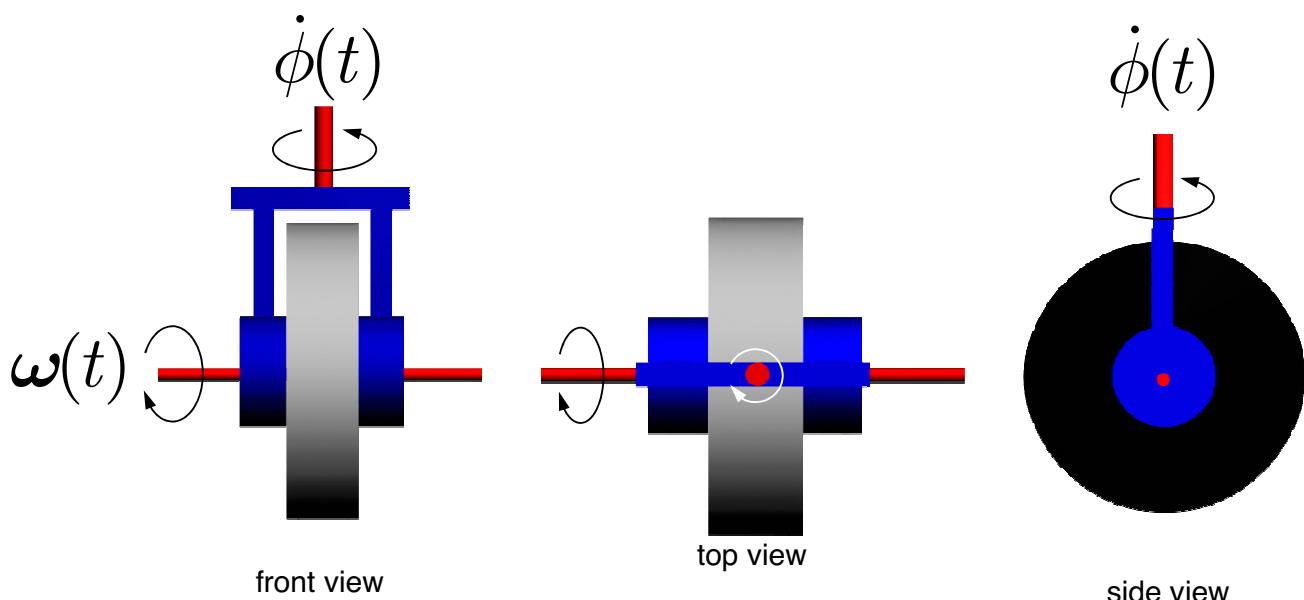
Simple non steering wheel



Simple steering wheel



Simple steering wheel



Castor Wheel



Omniwheel



Omniwheel



Industrial platforms
with omniwheels

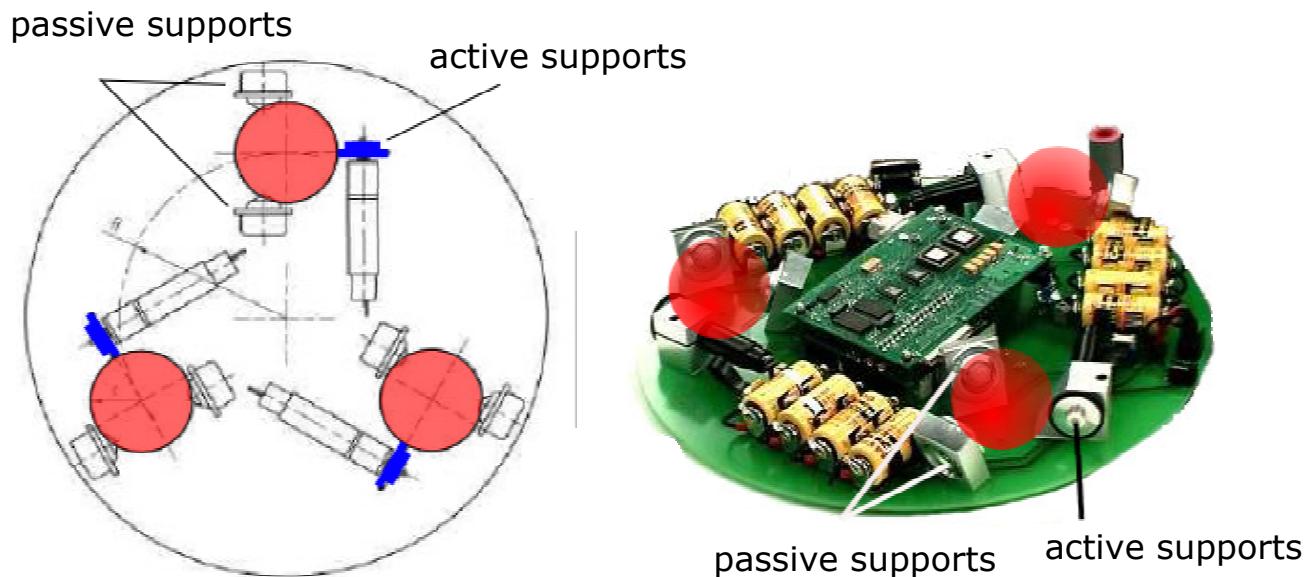


Omniwheel

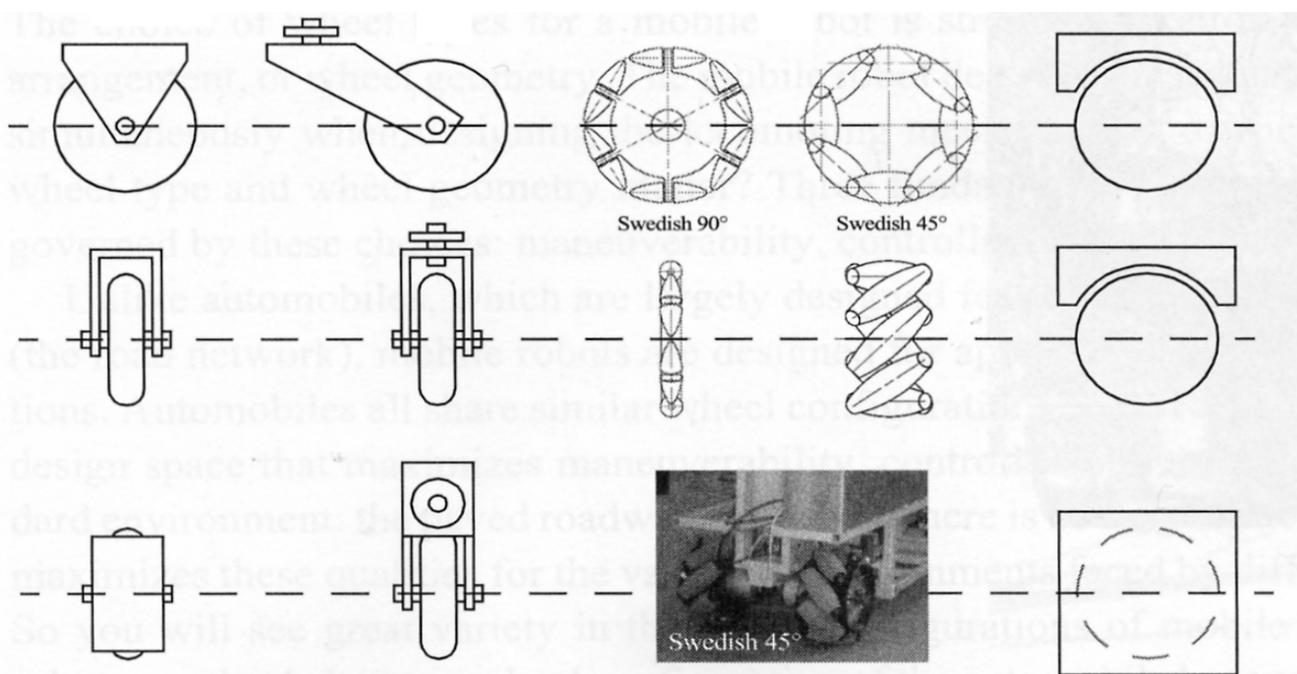


Omniwheel can also be used as a support in a differential drive robot

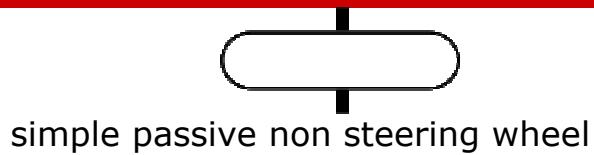
Omnidirectional Spherical Wheel



Wheels Symbols



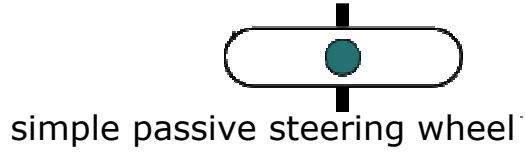
Wheels Symbols



simple passive non steering wheel



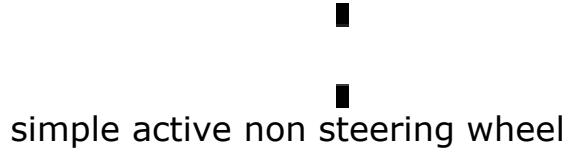
passive castor wheel



simple passive steering wheel



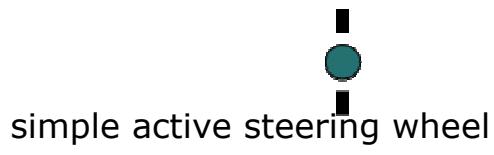
active castor wheel



simple active non steering wheel



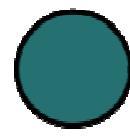
omnidirectional wheel



simple active steering wheel



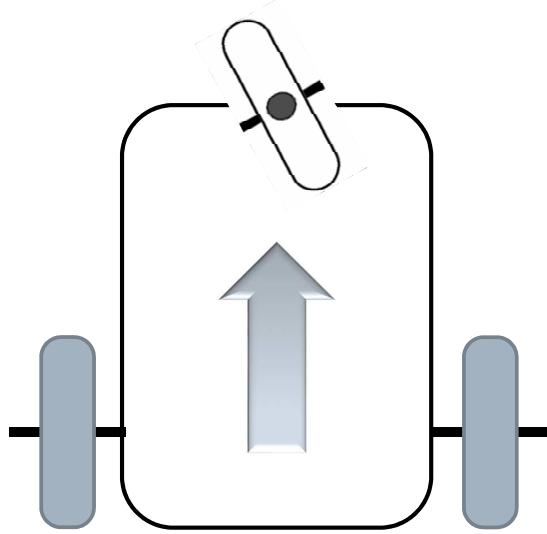
passive spherical wheel



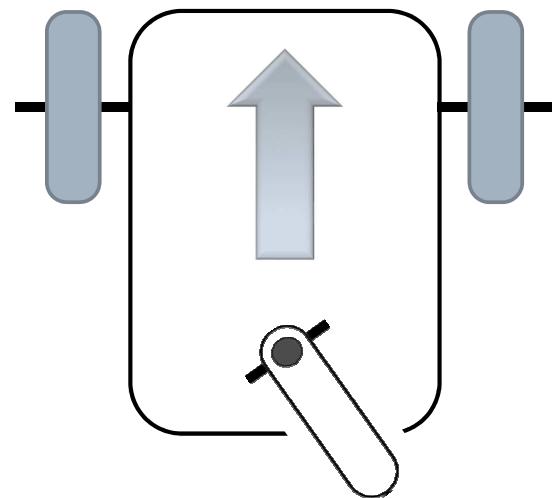
active spherical wheel

Typical structures

Active fixed wheels + steering wheel

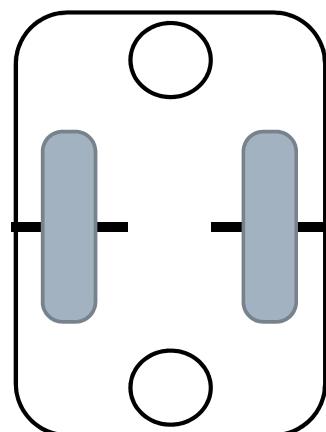
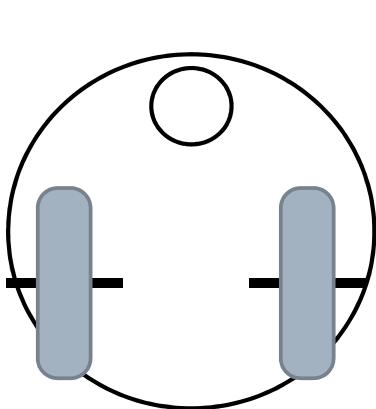


Active fixed wheels + castor passive wheel

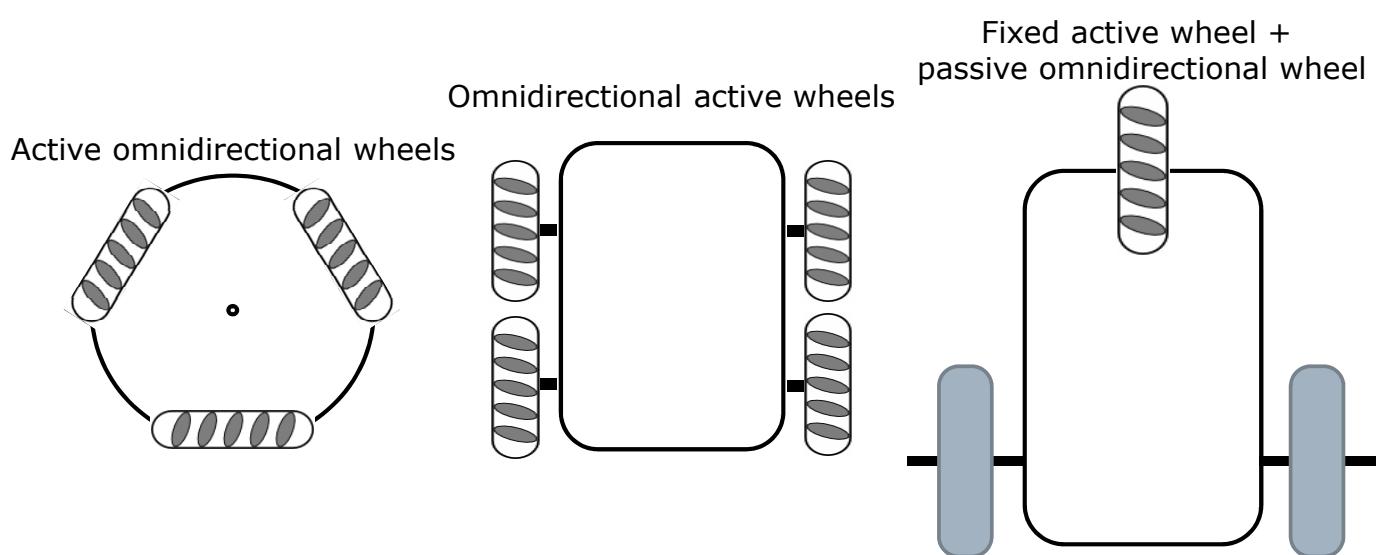


Typical structures

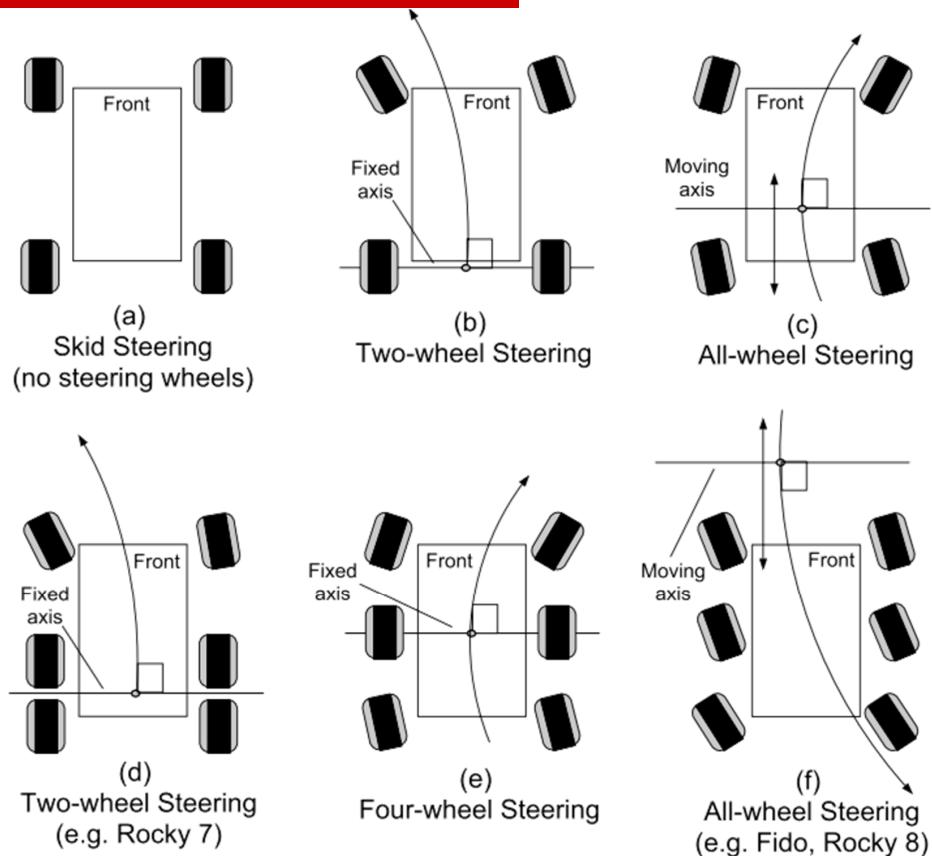
Differential wheels + passive spherical wheels



Typical structures



Steering



Steering

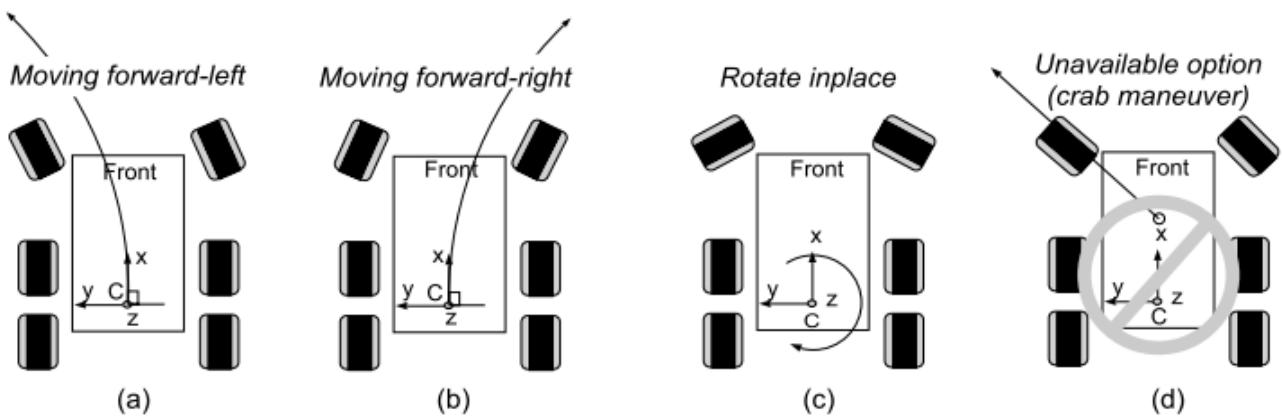


Fig. 2. The mobility of a six-wheel rover with two-wheel front steering (e.g. *Rocky 7*)

Fundamental problems in mobile robotics

- **Locomotion**: how the robot moves in the environment
- **Perception**: how the robot perceives the environment
- **Mapping**: how to build the map of the environment
- **Localization**: where is the robot wrt the map
- **Representation**: how the robot organizes the knowledge about the environment
- **Path planning/action planning**: what the robot shall do to go from here to there; what are the actions to be performed to complete a specified task
- **Supervision and control**: how are the command to actuators generated to perform simple or complex tasks. How to generate tasks



Robotics

Mobile robots

Part 1: Constraints and Kinematic Models

Kinematics and Constraints

- Wheeled vehicles are subject to **constraints** that limit the local mobility, without preventing (in general) the possibility of attaining arbitrary poses

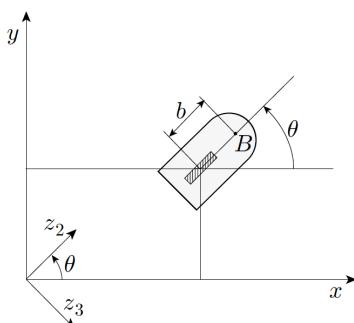
■ *Think about parking your car: you can't move sideways, but you can do parallel parking with a suitable sequence of maneuvers!*



2

Constraints

- Let $\mathbf{q} \in \mathbb{R}^n$ the vector **generalized coordinates** that describes the **configuration** of the robot and assume for the moment that the **configuration space \mathcal{C}** (the set of all possible robot configurations) coincides with \mathbb{R}^n
 - For example in the case of a **unicycle** (single steered wheel) we have $\mathbf{q} = [x \ y \ \theta]^T$, where (x, y) are the coordinate of the contact point of the wheel and θ the wheel angle w.r.t. an axis
- The robot motion is described by $\mathbf{q}(t)$ and can be subject to constraints



3

Constraint Classification

e.g.

$$x(t) - 2\theta(t) = 0 \rightarrow \text{bilateral and scleronomic}$$

$$x(t) - 2\theta(t) + 3t = 0 \rightarrow \text{bil. and Reconomic}$$

i.e. time-Variant vs time-invariant parameters

- Constraints can be expressed by

- equalities \rightarrow **bilateral** constraints =
- inequalities \rightarrow **unilateral** constraints <

they can explicitly depend on time (**reconomic** constraints)
or being independent from it (**scleronomic** constraints)

→ We will only deal with **bilateral and scleronomic constraints**, that is, represented by equalities in which the time does not explicitly appear

- We call **holonomic** (or **integrable**) those constraints that can be written as

$$h_i(\mathbf{q}) = 0, \quad i = 1, \dots, k < n$$

4

Holonomic Constraints

e.g. if $x - 2\theta = 0$ (Holonomic) and
 $\mathbf{q} = (x, y, \theta) \Rightarrow q(x, y) = q(\theta, y) \Rightarrow$ one dim.
 is removed from \mathbf{q} thanks to the constraint

- A system with all holonomic constraints is called a **holonomic system**
- The effect of holonomic constraints is to reduce the space of the accessible configurations to a subspace \mathcal{C} with dimension $n - k$
 - The existence of k holonomic constraints leads to the elimination of k generalized coordinates, which can be expressed as functions of the other $n - k$ (implicit function theorem) \Rightarrow implicit function thm.
 - This procedure may lead to singularities. Alternatively, we can define a new, reduced set of $n - k$ coordinates, directly defined on the accessible subspace

they PRODUCE limits ON the CONFIGURATIONS (e.g. RAILS of a TRAIN)

5

Kinematic constraints

e.g. $x + 2\dot{x} = 0 \Rightarrow \frac{dx}{dt} = -\frac{x}{2}$

limits the motion, not the config.
(holonomic constr. limit the configuration)

- The constraints that involve both the generalized coordinates and their derivatives (velocities) are called **kinematic constraints** and, in general, they are expressed as

$$a_i(\mathbf{q}, \dot{\mathbf{q}}) = 0, \quad i = 1, \dots, k < n$$

- Kinematic constraints limit the admissible instantaneous motion of the system, constraining the set of generalized velocities that can be obtained in each configuration
- Kinematic constraints can be often written in **Pfaffian form**, that is, they are **linear in the generalized velocities**

(in matrix form)

$$\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}$$

↓ e.g.
[3x + y] \dot{x} = 0

this limits ONLY the motion (like a car that CANNOT move sideways)⁶

Holonomic and nonholonomic constraints

- The existence of k holonomic constraints implies the existence of k kinematic constraints → Derivi un Holonomic Constr. e ottieni un KINEM. Constr.

$$\frac{dh_i(\mathbf{q})}{dt} = \frac{dh_i(\mathbf{q})}{d\mathbf{q}} \dot{\mathbf{q}} = 0, \quad i = 1, \dots, k$$

Abbiamo una
derivate, vogliamo
"TORNARE INDIETRO"

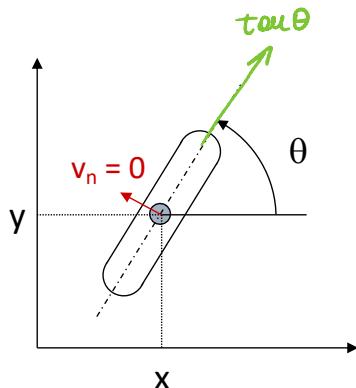
- The contrary is not true in general, since kinematic constraints can be reduced to holonomic constraints only if they are integrable. If not, they are called **nonholonomic** or **non-integrable**

- A system with at least a nonholonomic constraint is called **nonholonomic system**
- Nonholonomic constraints do not imply any loss of accessibility in \mathcal{C} (the number of generalized coordinates cannot be reduced), while velocities are constrained over a subspace of dimension $n - k$

Example of nonholonomic constraint

→ Q HOLOnomic CONSTRA. → CAN be in ANY configuration

- A unicycle rolls without slipping on a horizontal plane



$$\mathbf{q} = [x \ y \ \theta]^T$$

(x, y) are the coordinates of the point of contact with the plane, θ is the wheel angle w.r.t. the x axis, the wheel rotation angle ϕ is not influent for our analysis

- The **pure rolling constraint** implies that the velocity of the contact point cannot have nonzero components along the normal direction to the wheel plane

$$\frac{dy}{dx} = \tan \theta$$

i.e. it CAN ONLY move along the direction $\tan \theta$ (because of the orientation of the wheel)

Example of nonholonomic constraint

CONT.

- Such a constraint can be expressed in Pfaffian form through:

chain rule

$$\frac{dy}{dx} = \frac{dy}{dt} \cdot \frac{dt}{dx} = \frac{\dot{y}}{\dot{x}} = \frac{\sin \theta}{\cos \theta} \Rightarrow \dot{y} \cos \theta = \dot{x} \sin \theta$$

$$\Rightarrow \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \Rightarrow [\sin \theta \ -\cos \theta \ 0] \dot{\mathbf{q}} = 0$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

NON INTEGRABLE

- The constraint is **nonholonomic**, since it does not imply any loss of accessibility in the wheel configuration space
⇒ It is possible to displace the wheel from any initial configuration \mathbf{q}_i to any final configuration \mathbf{q}_f through a suitable sequence of motions that do not violate the pure rolling constraint

Kinematic model

e.g. $\underbrace{[\cos\theta \sin\theta 0]}_{A^T} \dot{q} = 0$

- Form the matrix formulation of the kinematic constraints

$$A^T(q)\dot{q} = 0$$

it is evident that the $(n - k)$ admissible generalized velocities belong to the null space of $A^T(q)$ ↪ i.e. All velocities \dot{q} that satisfy our constraint $A^T(q)\dot{q} = 0$

$$\dot{q} \in N(A^T(q))$$

- Denoting with $\{g_1(q), \dots, g_{n-k}(q)\}$ a base of $N(A^T(q))$, it is possible to define the admissible trajectories as the solution of the nonlinear dynamical system

$$\dot{q} = \sum_{j=1}^m g_j(q) u_j = G(q)u, \quad m = n - k$$

that represents the kinematic model of the constrained system

14

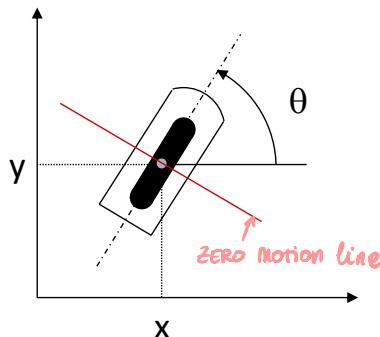
Kinematic model

- Vector q represents the **state vector** of the system, while $u = [u_1 \dots u_m]^T$ is the **input vector**
- The obtained system is said to be **driftless**, since a null input causes $\dot{q} = 0$
- The choice of the **vectorial input fields** $g_j(q)$, is not unique and, consequently, also the choice of input u is not unique
 - The components of u can either have a physical interpretation and/or be related to the available control input, or not

15

Kinematic model of a unicycle

- Let us consider a **unicycle**, a vehicle with a single steerable wheel



seen in Prev. example

The configuration is described by

$$\mathbf{q} = [x \ y \ \theta]^T$$

and the pure rolling constraint is
 $\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \Rightarrow [\sin \theta \ -\cos \theta \ 0] \dot{\mathbf{q}} = 0$

- The line passing by the contact point of the wheel and orthogonal to the direction of motion is called **zero motion line**

- A base of the null space of the constraint matrix is given by:

$$\mathbf{g}_1(\mathbf{q}) = [\cos \theta \ \sin \theta \ 0]^T, \quad \mathbf{g}_2(\mathbf{q}) = [0 \ 0 \ 1]^T$$

16

Kinematic model of a unicycle

CONT.

- Thus, we have:

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}$$

and, correspondingly, the **unicycle kinematic model** is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

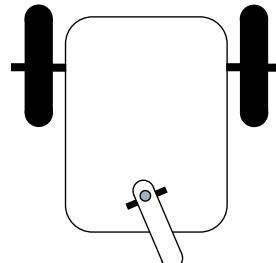
In this case the inputs have a physical meaning and are related to straightforward control inputs: v is the **driving velocity** (modulus of the velocity of the contact point) and ω is the **steering velocity**

17

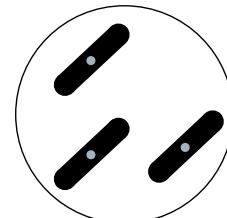
Equivalence between the unicycle and other structures

- A unicycle would not be practically useful, due to its balancing problems. From a kinematic point of view, however, it is equivalent to more stable structures, such as

- Differential drive vehicle



- Synchro drive vehicle



18

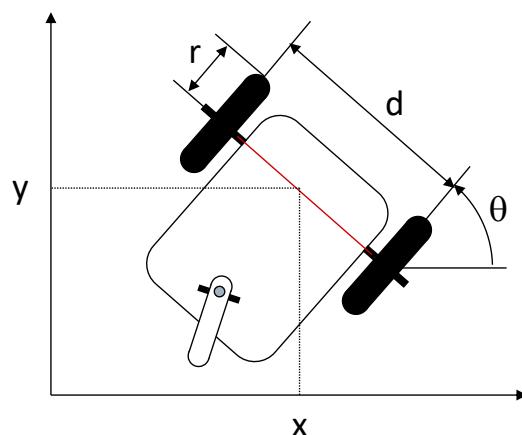
Equivalence with the differential drive

- The «physical» inputs are the rotation velocities of the two wheels, ω_R and ω_L
- There exists a direct correspondence between such input and the ones of the unicycle (drive velocity v and steering velocity ω)

$$v = \frac{r(\omega_R + \omega_L)}{2}$$

$$\omega = \frac{r(\omega_R - \omega_L)}{d}$$

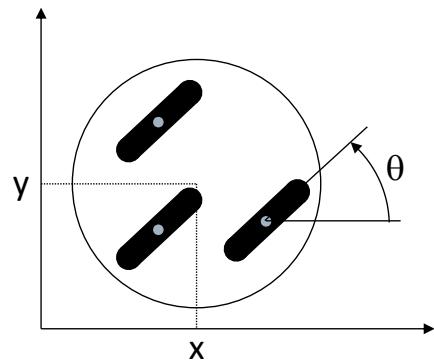
(x, y) are the coordinates of the middle point of the wheel axis, θ the vehicle heading



19

Equivalence with a synchro drive vehicle

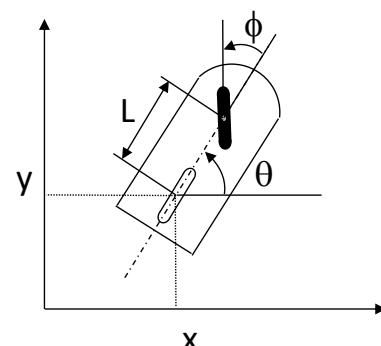
- The equivalence is straightforward, since both drive and steer velocity are common to the three wheels
 - Coordinates (x, y) can represent here any point of the chassis, while θ is the wheels heading (the chassis has constant heading)



20

Kinematic model of a bicycle

- Let us consider a **bicycle**, that is a vehicle with a steered wheel and a fixed one, mounted at a distance L
 - Possible choice for the generalized coordinates:
 - $(x, y) \rightarrow$ coordinates of the contact point of the rear wheel
 - $\theta \rightarrow$ heading of the vehicle w.r.t. x
 - $\phi \rightarrow$ steering angle of the front wheel



$$\mathbf{q} = [x \quad y \quad \theta \quad \phi]^T$$

21

Kinematic model of a bicycle

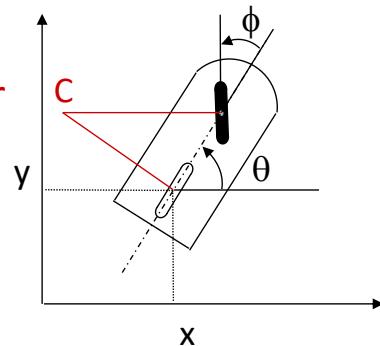
- There exist **two pure rolling constraints**, one for each wheel

front wheel $\rightarrow \dot{x}_f \sin(\theta + \phi) - \dot{y}_f \cos(\theta + \phi) = 0$

rear wheel $\rightarrow \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \rightarrow$ *same as unicycle*

where (x_f, y_f) are the coordinates of the center of the rear wheel

- The two motion lines intersect in a point C called **instantaneous center of rotation**
 - The position of C depends only on \mathbf{q}
 - Every point of the chassis moves instantaneously along a circumference arc centered in C



22

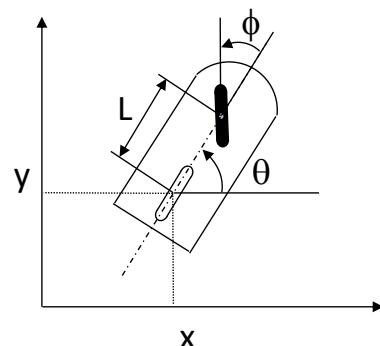
Kinematic model of a bicycle

- The coordinates (x_f, y_f) of the center of the front wheel can be expressed as a function of those of the rear wheel, using the rigid body constraint

| $x_f = x + L \cos \theta$
| $y_f = y + L \sin \theta$

- After some algebra, the first pure rolling constraint can be expressed as:

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - L \dot{\theta} \cos \phi = 0$$



23

Kinematic model of a bicycle

- Matrix form of the constraint $\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}$

$$\mathbf{A}^T(\mathbf{q}) = \begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 \\ \sin(\theta + \phi) & -\cos(\theta + \phi) & -L \cos \phi & 0 \end{bmatrix}$$

- The rank of \mathbf{A}^T is 2 and, consequently, its null space has dimension $4 - 2 = 2$. A possible base for $N(\mathbf{A}^T(\mathbf{q}))$ is formed by the columns of

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \cos \theta \cos \phi & 0 \\ \sin \theta \cos \phi & 0 \\ \frac{\sin \phi}{L} & 0 \\ 0 & 1 \end{bmatrix}$$

24

Kinematic model of a bicycle

- With this choice the kinematic model is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi \\ \sin \theta \cos \phi \\ \frac{\sin \phi}{L} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2$$

where input u_2 is the steering velocity ω (provided by the front wheel)

- The expression of u_1 depends on the drive (front or back) of the vehicle
 - If the bicycle has front drive, we have directly $u_1 = v$, where v is the driving velocity of the front wheel

25

Kinematic model of a bicycle

- If the vehicle is back drive, we can derive u_1 observing that the first two equations must coincide with those of the unicycle. This leads to set

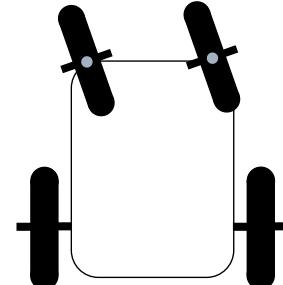
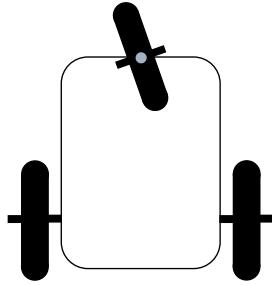
$$u_1 = \frac{v}{\cos \phi}$$

where v is the **drive velocity of the back wheel**, thus yielding the kinematic model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \phi \\ \frac{L}{v} \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

Equivalence of the bicycle with other structures

- Also the bicycle presents balance problems in practice, but also in this case we have **kinematically equivalent structures**
 - tricycle**
 - automobile**



In both cases we may have rear or front drive. The point (x, y) is the midpoint of the back wheel axis, θ is the vehicle heading, and ϕ is the steering angle

Appendice A

Control

A.1 Cose dette da Rizzo che non ho capito che c'entra-no

Come accennato in precedenza, in questo caso andremo a controllare gli attuatori in velocità. È possibile dimostrare (vedi dopo) la seguente assunzione:

$$\mathbf{G}_v \mathbf{v}_c \approx \mathbf{K}_\omega \mathbf{K}_r \dot{\mathbf{q}}$$

L'importante di questa espressione è la proporzionalità fra \mathbf{G}_v e $\dot{\mathbf{q}}$ (velocità), che notiamo essere indipendente dai parametri del manipolatore. Inoltre questa proporzione è tanto più valida quanto velocità/accelerazioni sono piccole (per questa cosa viene in aiuto anche il gear-reduction ratio K_r).

Dim.

Partendo dal modello dinamico $\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$ introduciamo in esso la frizione viscosa elettrica. Ovvvero poniamo $\mathbf{F}_v = \mathbf{F}_{v.\text{mech.}} + \mathbf{F}_{v.\text{electr.}} = \mathbf{F}_{v.\text{mech.}} + \mathbf{K}_r \mathbf{K}_t \mathbf{R}_a^{-1} \mathbf{K}_\omega \mathbf{K}_r$.

Inoltre dal modello del motore: $\boldsymbol{\tau}_m = \mathbf{K}_r^{-1} \boldsymbol{\tau} = \mathbf{K}_t \mathbf{I}_a \implies \boldsymbol{\tau} = \mathbf{K}_r \mathbf{K}_t \mathbf{I}_a$. Sappiamo anche che $V = Ri \implies i = \frac{V}{R}$ e quindi, ricordando che $\mathbf{V}_c = \mathbf{G}_v \mathbf{V}'_c$, otteniamo $\mathbf{I}_a = \mathbf{R}_a^{-1} \mathbf{G}_v \mathbf{V}_c$. Unendo il tutto otteniamo $\boldsymbol{\tau} = \mathbf{K}_r \mathbf{K}_t \mathbf{R}_a^{-1} \mathbf{G}_v \mathbf{V}_c$. Inserendo tutto nella formula del modello dinamico otteniamo:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_{v.\text{mech.}} \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{K}_r \mathbf{K}_t \mathbf{R}_a^{-1} \mathbf{G}_v \mathbf{V}_c - \mathbf{K}_r \mathbf{K}_t \mathbf{R}_a^{-1} \mathbf{K}_\omega \mathbf{K}_r$$

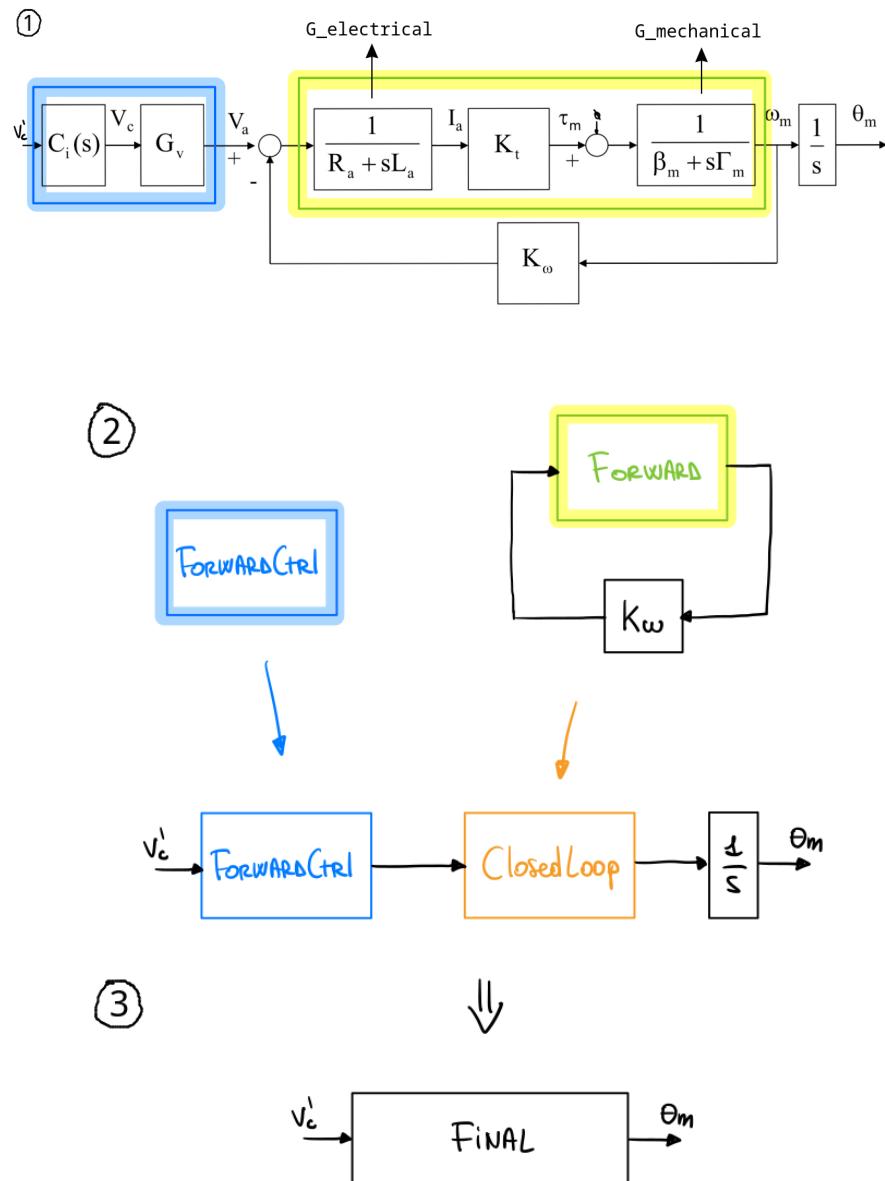
Ovvvero, raccogliendo i termini a destra (ricordando che originalmente quello era $\boldsymbol{\tau}$):

$$\boldsymbol{\tau} = \mathbf{K}_r \mathbf{K}_t \mathbf{R}_a^{-1} (\mathbf{G}_v \mathbf{V}_c - \mathbf{K}_\omega \mathbf{K}_r \dot{\mathbf{q}}) \quad (\text{A.1.1})$$

Quello che otteniamo fra le parentesi è l'espressione ipotizzata inizialmente. (la quasi uguaglianza viene dal fatto che $\mathbf{K}_r \gg 1$, \mathbf{R}_a molto piccolo, $\boldsymbol{\tau}$ non troppo grosso).

□

A.2 Derivazione del valore di steady-state per il motore velocity-generator



Per calcolare il valore di steady-state utilizziamo il tool simbolico di Matlab.

- ① Iniziamo definendo i vari simboli e i vari blocchi:

```

syms s R L beta Gamma K_omega K_i G_v C_i K_t

% Electrical and Mechanical balance t.f.
G_electrical = 1/(R + s*L);
G_mechanical = 1/(beta + s*Gamma);

```

- ② Proseguiamo accorpando i percorsi di *forward* e di *feedback*:

```

% Simplify controller (remember that here K_i = 0)
ForwardCtrl = C_i * G_v;

```

```
% Calculate the closed-loop t.f. (supposing tau_r = 0)
Forward = G_electrical * K_t * G_mechanical;
ClosedLoop = Forward/(1 + Forward*K_omega);
```

- (3) Infine uniamo tutto e calcoliamo il valore in *steady-state* tramite il teorema del valore finale $y_\infty = \lim_{s \rightarrow 0} sF(s)$:

```
% Put everything together
Final = simplify(ForwardCtrl*ClosedLoop*(1/s));

% Calculate steady-state (lim_{s->0} sF(s))
steady_state = subs(s*Final, s, 0);
% Remove beta from the assumption that beta << (K_omega*K_t)/R
steady_state = subs(steady_state, beta, 0);
% Set C_i = 1
steady_state = subs(steady_state, C_i, 1);

pretty(simplify(steady_state))
```

dove, l'ultima espressione ritorna proprio:

$$\frac{G_v}{K_\omega}$$

□