BIBBIA DI ROBOTICA

Autore: Andrea Mazzera

Indice

1. Cinematica 2
1.1 Caratteristiche Cinematiche dei Robot 2
1.2 Rappresentazione Non Minima dell'Orientamento 4
1.2.1 Matrice di Rotazione 4
1.2.2 Rappresentazione Asse-Angolo 8
1.2.3 Quaternione Unitario 10
1.3 Rappresentazione Minima dell'Orientamento 11
1.3.1 Angoli di Eulero 11
1.4 Trasformazione Omogenea 13
1.5 Convenzione di Denativ-Hartenberg 14
1.6 Cinematica Diretta 16
1.6.1 Cinematica Diretta per Catene Aperte 16
1.6.2 Cinematica Diretta per Catene Chiuse 22
1.7 Spazio dei Giunti, Spazio Operativo e Spazio di Lavoro 25
1.8 Calibrazione Cinematica 27
1.9 Cinematica Inversa 29
2. Cinematica Differenziale e Statica 31
2.1 Caratterizzazione della Cinematica Differenziale 31
2.2 Singolarità Cinematiche 40
2.2.1 Caratterizzazione delle Singolarità 40
2.2.2 Decomposizione delle Singolarità 42
2.3 Analisi della Ridondanza 44
2.4 Inversione Cinematica Differenziale 46
Caratterizzazione dell'Inversione Cinematica Differenziale 46
Inversione per Manipolatori Ridondanti 46
Inversione con Singolarità Cinematiche 50
2.5 Jacobiano Analitico 52
2.6 Algoritmi di Inversione Cinematica 54
2.6.1 Caratteristiche degli Algoritmi CLIK 54
2.6.2 Algoritmi del Primo Ordine 55
2.6.3 Algoritmi del Secondo Ordine 59
2.6.4 Errore di Orientamento 61

2.7 Statica 65
2.8 Dualità Cineto-Statica 67
2.8.1 Trasformazione di Velocità e Forze 69
2.8.2 Dualità per Catene Chiuse 70
2.9 Ellissoidi di Manipolabilità 71
3. Pianificazione delle Traiettorie 74
3.1 Definizione di Pianificazione e Caratteristiche 74
3.2 Traiettorie nello Spazio dei Giunti 75
3.3 Traiettorie nello Spazio Operativo 81
3.3.1 Definizione e Approccio 81
3.3.2 Calcolo di Posizione e Orientamento 84
4. Attuatori e Sensori 88
4.1 Generalità sugli Attuatori e Sensori 88
4.1.1 Azionamenti 93
4.1.2 Riduttore 98
4.1.3 Controllo di Posizione 100
4.2 Sensori Propriocettivi 102
4.2.1 Trasduttori di Posizione 102
4.2.2 Trasduttori di Velocità 105
4.3 Sensori Esterocettivi 106
4.3.1 Sensore di Forza 106
4.3.2 Sensori di Distanza 108
4.3.3 Sensore di Visione 112
5. Unità di Governo 115
5.1 Architettura Funzionale 115
5.2 Architettura Funzionale per Robot Industriali 117
5.3 Ambiente di Programmazione 118
5.4 Architettura Hardware 120
6. Dinamica 121
6.1 Caratterizzazione della Dinamica 121
6.2 Formulazione di Lagrange 122
6.2.1 Definizione 122
6.2.2 Dimostrazione del Modello Matematico 124
6.2.3 Proprietà del Modello Dinamico 129

6.3 Formulazione di Newton-Eulero 135
6.4 Identificazione dei Parametri Dinamici 138
6.5 Scalatura Dinamica della Traiettoria 141
6.6 Modello Dinamico nello Spazio Operativo 143
6.7 Ellisoide di Manipolabilità Dinamica 146
7. Controllo del Moto 148
7.1 Problema del Controllo del Moto 148
7.2 Controllo del Moto nello Spazio dei Giunti 149
7.3 Controllo Decentralizzato 151
7.3.1 Modello Decentralizzato 151
7.3.2 Controllo Indipendente ai Giunti 153
7.3.3 Compensazione in Avanti Decentralizzata 160
7.3.4 Compensazione in Avanti a Coppia Precalcolata 163
7.4 Controllo Centralizzato 164
7.4.1 Definizione di Controllo Centralizzato 164
7.4.2 Controllo a Dinamica Inversa 165
7.4.3 Controllo PD+ 168
7.4.4 Controllo Robusto 171
7.4.5 Controllo Adattativo 175
7.6 Controllo del Moto nello Spazio Operativo 179
7.6.1 Linee Guida per il Controllo nello Spazio Operativo 179
7.6.2 Controllo PD con Compensazione di Gravità 180
7.6.3 Controllo a Dinamica Inversa 181
8. Force Control 182
8.1 Characterization of Interaction Control 182
8.2 Compliance Control 183
8.2.1 Compliance Behaviour 183
8.2.2 Definition of Compliance Control 185
8.3 Impedence Control 188
8.3.1 Definition of Impedence Control 188
8.3.2 Admittance Control 190
8.4 Force Control 191
8.5 Motion Constraint 193
8.6 Natural Constraints and Ariticial Constraints 195

8.7 Hybrid Force/Motion Control 197
9. Visual Servoing 198
9.1 Characterization of Visual Servoing 198
9.2 Visual System 200
9.3 Image Processing 201
9.4 Pose Estimation 203
9.5 Camera Calibration 206
9.6 Stereo Vision 207
9.7 Visual Servoing in the Operational Space 208
9.8 Visual Servoing in the Image Space 211
9.9 Hybrid Visual Control 214
9.10 Visual and Force Control 215
10. Robots with Flexible Elements 216
10.1 Characterization 216
10.2 Dynamic Modelling 217
10.3 Inverse Dynamics 220
10.4 Regulation Control 223
10.5 Trajectory Tracking 225
11. Cooperative Manipulation 227
11.1 Dual Manipulator System 227
11.2 Object Manipulation 229
11.3 Control 231
11.4 Grasping 233
11.4.1 Multifinger Hand 233
11.4.2 Grasp Classification 236
11.4.3 Control Properties 238

1. Cinematica

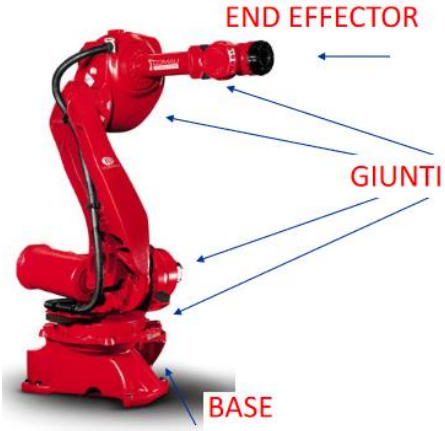
1.1 Caratteristiche Cinematiche dei Robot

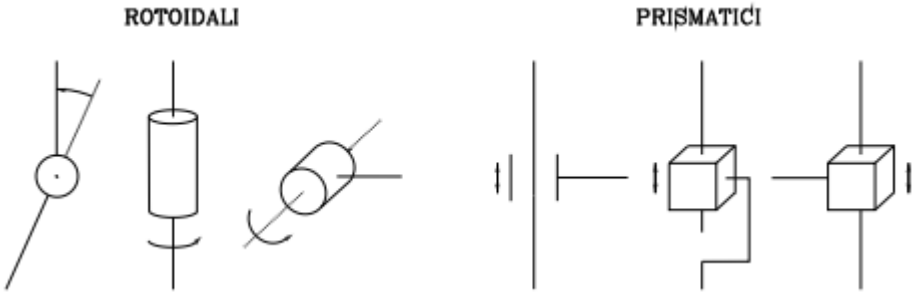
Un robot è un sistema complesso costituito da diversi sotto-sistemi:

- **Sistema Meccanico**:
  - Organi di locomozione (ruote, cingoli, gambe meccaniche)
  - Organi di manipolazione (braccia meccaniche, utensili, mani artificiali).
- **Sistema di Attuazione**:
  - Anima le componenti meccaniche del robot;
  - Controllo del moto (servomotori, azionamenti e organi di trasmissione).
- **Sistema Sensoriale**:
  - Sensori propriocettivi (stato interno del robot);
  - Sensori esterocettivi (stato esterno dell'ambiente).
- **Sistema di Governo**: si realizza a partire dal modello matematico di tutti i componenti del robot.
  - Esecuzione dell'azione nel rispetto dalla pianificazione del compito

Tuttavia, dal punto di vista topologico, un robot è una *catena cinematica* costituita da:

- **Link**: sono le parti rigide e passive del sistema;
- **Giunti**: sono gli elementi di collegamenti dei link e sono invece la parte attiva del sistema che sarà soggetta al controllo. I giunti di fatto rappresentano i gradi di libertà e le variabili di stato del sistema stesso. Ciascun giunto ha *un solo grado di libertà* e può essere:
  - **A. Rotoidale**: concede il grado di libertà sulla rotazione;
  - **B. Prismatico**: concede il grado di libertà sulla traslazione.

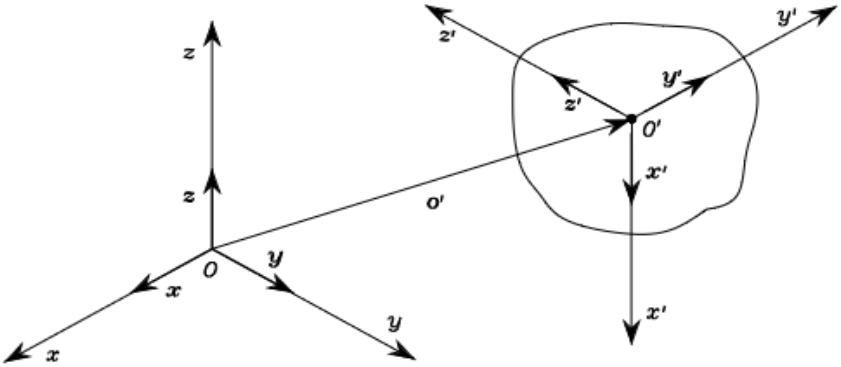




Essendo il robot una catena cinematica possiamo introdurre il seguente studio. La **Cinematica** *è lo studio delle relazioni tra posizioni dei giunti e la posa dell'end effector* dove:

- La posizione dei giunti è intesa come la postura del robot, ovvero la posa dei diversi elementi rigidi che lo compogono e che dipendono direttamente dalle variabili di giunto;
- La posa dell'end-effector è la combinazione di posizione e orientamento;

Essendo il robot una catena cinematica si capisce come il suo moto complessivo può essere visto come un insieme di moti elementari (rotoidali o traslatori) di un giunto rispetto al precedente. Allora se dobbiamo studiare posizioni o orientamenti di oggetti rispetto ad altri, è necessario associare a ciascuno di essi delle terne di tre assi. Facciamo un semplice esempio per capire questo aspetto. Immaginiamo per esempio si avere un sistema di riferimento inerziale rappresentato da una terna fissa $Oxyz$ e un corpo di forma generica con la propria terna $O'x'y'z'$ di riferimento solidale ad esso. Con il termine solidale vogliamo dire che la terna si muove insieme al corpo ad esso associato.



Immaginiamo di voler descrivere, rispetto alla terna di riferimento inerziale fisso:

- La posizione o' dell'origine O';
- L'orientamento della terna $x'y'z'$.

La definizione della posizione di una terna o in generale di un qualsiasi oggetto non è quasi mai problematica, infatti in questo caso la posizione della terna $O'x'y'z'$ è semplicemente un vettore colonna o':

$$o' = [o'_x \quad o'_y \quad o'_z]^T$$

Al contrario, invece, il discorso sull'orientamento risulta essere più complesso e necessità maggiore attenzione. Esistono diversi modi per descrivere l'orientamento nello spazio tridimensionale e queste diverse metodologie si differenziano in due grandi categorie:

A. **Rappresentazioni Non Minime dell'Orientamento**: l'orientamento è descritto attraverso più di tre parametri. Alcune di queste rappresentazioni che tratteremo sono la matrice di rotazione, la rappresentazione asse-angolo e il quaternione unitario;
B. **Rappresentazioni Minime dell'Orientamento**: l'orientamento è descritto attraverso al massimo tre parametri. Per questo tipo di rappresentazione studieremo gli angoli di Eulero.

1.2 Rappresentazione Non Minima dell'Orientamento

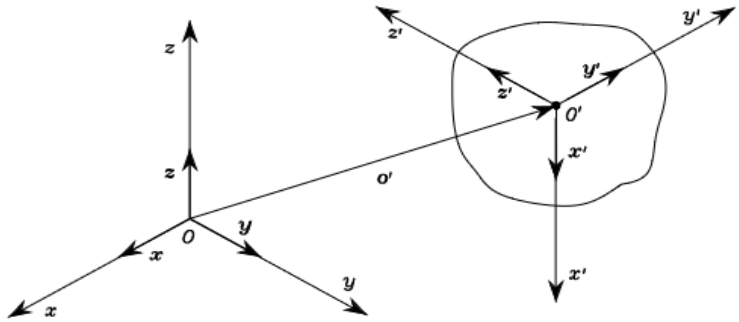
1.2.1 Matrice di Rotazione

La **Matrice di Rotazione R** è una matrice $3x3$ che ha tre fondamentali interpretazioni (funzioni):

✓ **Intepretazione 1**:

Fornire l'orientamente di una terna di coordinate rispetto ad un'altra. Motivo per cui, i vettori colonna della matrice di rotazione saranno i coseni direttore degli assi della terna ruotata rispetto alla terna di partenza.

Per studiare questa interpretazione consideriamo di avere un sistema di riferimento inerziale rappresentato da una terna fissa $Oxyz$ e un corpo di forma generica con la propria terna $O'x'y'z'$ di riferimento solidale ad esso.



Vogliamo descrivere, rispetto alla terna di riferimento $Oxyz$:

- La posizione o' dell'origine O', la quale è definita dal seguente vettore colonna i cui elementi sono proprio le coordinate di O' rispetto alla terna $Oxyz$.
- L'orientamento della terna $Ox'y'z'$, il quale può essere descritto mediante i coseni direttori x', y', z' degli assi della terna 1 rispetto agli assi della terna inerziale.

$$o' = [o'_x \quad o'_y \quad o'_z]^T$$

$$\begin{cases} x' = x'_x x + x'_y y + x'_z z \\ y' = y'_x x + y'_y y + y'_z z \\ z' = z'_x x + z'_y y + z'_z z \end{cases}$$

Dunque possiamo avvarlere di questi tre versori per descrivere l'orientamento della terna, o per meglio dire la rotazione, della terna solidale al corpo rispetto alla terna di riferimento fissa definendo così la matrice di rotazione:

$$\boldsymbol{R} = [\boldsymbol{x}' \quad \boldsymbol{y}' \quad \boldsymbol{z}'] = \begin{bmatrix} \boldsymbol{x}'^T\boldsymbol{x} & \boldsymbol{y}'^T\boldsymbol{x} & \boldsymbol{z}'^T\boldsymbol{x} \\ \boldsymbol{x}'^T\boldsymbol{y} & \boldsymbol{y}'^T\boldsymbol{y} & \boldsymbol{z}'^T\boldsymbol{y} \\ \boldsymbol{x}'^T\boldsymbol{z} & \boldsymbol{y}'^T\boldsymbol{z} & \boldsymbol{z}'^T\boldsymbol{z} \end{bmatrix}$$

La quale tra l'altro presenta due proprietà molto utili dove la seconda discende dalla prima:

- **Proprietà 1 - Ortogonalità**: $R^T R = I$
- **Proprietà 2**: $R^{-1} = R^T$
- **Proprietà 3**: la $R_k(-\theta)$, ovvero la matrice di rotazione intorno ad un asse k e di un angolo $-\theta$ equivale alla matrice di rotazione trasposta intorno ad un asse k ma con angolo cambiato di segno: $R_k(-\theta) = R_k^T(\theta)$

La matrice di rotazione è una rappresentazione non minima dell'orientamento nello spazio perché descrive quest'ultimo mediante nove parametri. Questa sovrabbondanza è dovuta alla presenza di vincoli di ortogonalità tra i vettori colonna della matrice R. Questo comporta che non possiamo specificare tutti i valori della matrice di rotazione visto che alcuni di essi saranno ridondanti.

Poiché descrivere la rotazione intorno ad asse qualunque può essere più o meno complicata, ci conviene definire delle rotazioni elementari intorno agli assi della terna ortgonale:

- **Rotazione di α intorno a z**:

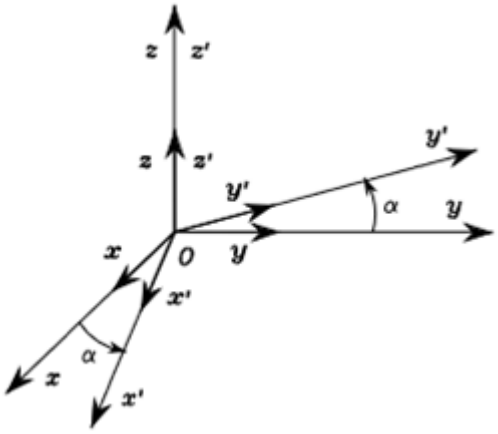
$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Rotazione di β intorno a y**:

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

- **Rotazione di γ intorno a x**:

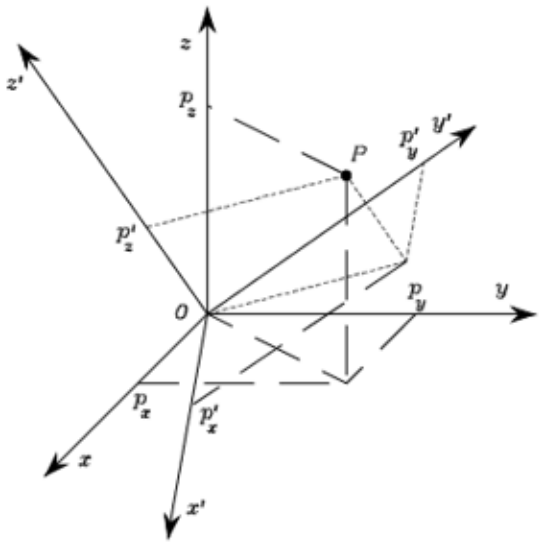
$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$



Nota: il vettore colonna e riga con tutti zeri ed un solo uno lo troviamo solo sulla componente relativa alla rotazione e tra l'altro l'1 può trovarsi solo sulla diagonale principale.

✓ **Intepretazione 2**:

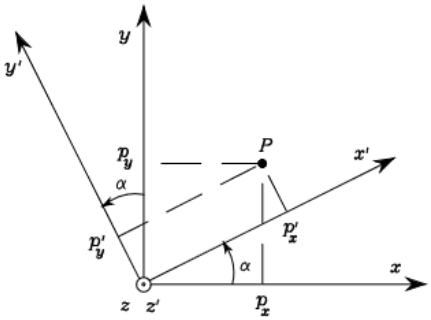
Rappresenta una trasformazione di coordinate che mette in relazione le coordinate di uno stesso punto in due terne differenti. Supponiamo di avere sempre due terne, una fissa (x, y, z) e una mobile (x', y', z'), le quali hanno le origini coincidenti e la seconda è ruotata rispetto alla prima. Consideriamo un punto P di cui conosciamo il vettore colonna p', ovvero il vettore le cui componenti ci dicono la posizione di tale punto nel riferimento mobile. Il nostro scopo è conoscere il vettore colonna p che ci dice la posizione di P nella terna fissa. Per risolvere questo problema impieghiamo proprio la matrice di rotazione che descrive la rotazione della terna mobile rispetto a quella fissa: $p = [x' \;\; y' \;\; z']p' = Rp'$



E' da notare che se volessimo considerare il problema inverso, allora dovremmo effettivamente fare la formula inversa della precedente: $p' = R^T p$

Esempio:

$$p_x = p'_x \cos\alpha - p'_y \sin\alpha$$
$$p_y = p'_x \sin\alpha + p'_y \cos\alpha$$
$$p_z = p'_z$$

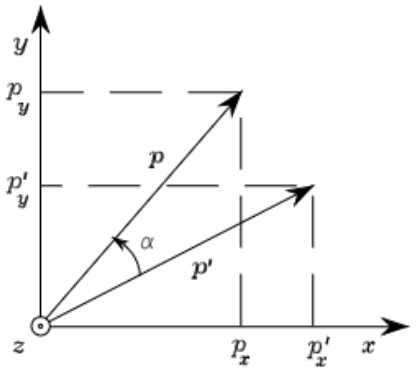


✓ **Interpretazione 3**:

Fare da operatore che consente di ruotare un vettore in una stessa terna di coordinate. Tale interpretazione è strettamente legata alla precedente e consiste nella possibilità di trasformare un vettore da un riferimento ad un altro ruotati tra loro. In tal caso sfruttiamo nuovamente la relazione: $p = Rp'$

Esempio:

$$p_x = p'_x \cos\alpha - p'_y \sin\alpha$$
$$p_y = p'_x \sin\alpha + p'_y \cos\alpha$$
$$p_z = p'_z$$
$$\boldsymbol{p} = \boldsymbol{R}_z(\alpha)\boldsymbol{p}'$$



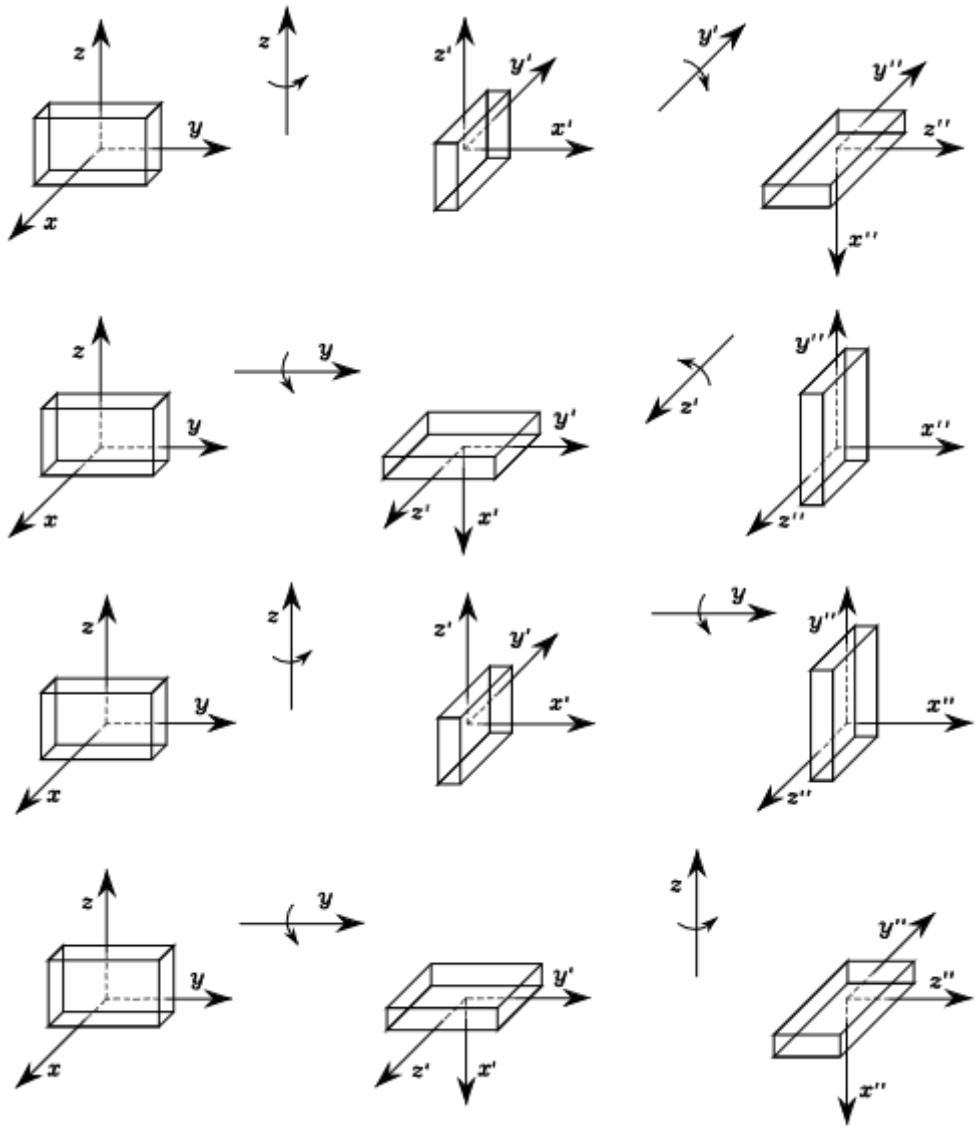
Nella realtà chiaremente è piuttosto raro avere delle semplici rotazioni elementari intorno agli assi, mentre invece è molto più comune avere un insieme di rotazioni visto che saranno riferite anche ad assi generici. A tal proposito, possiamo vedere una rotazione composta come una successione di rotazioni elementari. Per capire questa affermazione, supponiamo di avere tre sistemi di riferimento 0,1,2 dove tra la prima e la seconda abbiamo una rotazione, così come tra la seconda e la terza, mentre invece abbiamo una rotazione composta tra la prima e la terza. Poi prendiamo un generico punto P di cui conosciamo la posizione rispetto alla terna 2. Ora per trovare la coordinate di P nel riferimento 1 usiamo la matrice di rotazione della terna 2 rispetto alla terna 1 e quindi la relazione è: $p^1 = R_2^1 p^2$

Ora per conoscere le coordinate del punto P nel riferimento 0 dovremo effettuare una serie di rotazioni: $p^0 = R_1^0 p^1 = R_1^0[R_2^1 p^2] = R_1^0 R_2^1 p^2 = R_2^0 p^2$

Questo semplice esempio ci da la possibilità di definire due tipologie di rotazioni:

- **Rotazione in Terna Corrente**: quindi ruotiamo una certa terna rispetto a quella che attualmente stiamo considerando. Ciò implica dal punto di vista pratico che dovremo postmoltiplicare le diverse matrici di rotazione, infatti nell'esempio partiamo da R_1^0 e poi postmolitplichiamo per R_2^1;
- **Rotazione in Terna Fissa**: è la rotazione duale della precedente in cui si vanno a considerare le rotazioni dei frame rispetto ad una terna fissa. Ciò implica dal punto di vista pratico che dovremo premoltiplicare le diverse matrici di rotazione, dunque se volessimo prendere come riferimento l'esempio precedente, supponendo R_0 al terna fissa, allora dovremmo scrivere: $R_2^0 = R_2^1 R_1^0$

$$R_i^j = \left(R_j^i\right)^{-1} = \left(R_j^i\right)^T$$

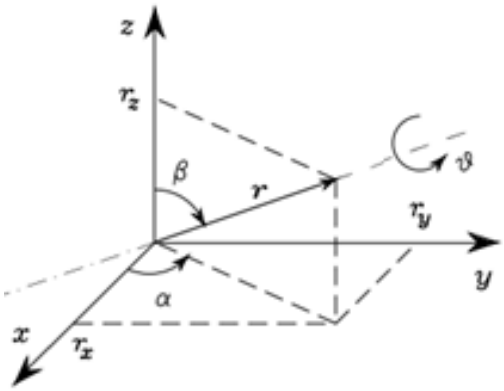

z
y
x
z
z'
y'
x'
y'
y''
z''
x''
z
y
x
y
y'
z'
x'
z'
y''
x''
z''
z
y
x
z
z'
y'
x'
y
y''
x''
z''
z
y
x
y
y'
z'
x'
z
y''
z''
x''

1.2.2 Rappresentazione Asse-Angolo

La **Rappresentazione Asse-Angolo** *si tratta di una rappresentazione non minima dell'orientamento che esprime la rotazione di un dato angolo attorno ad un asse nello spazio mediante l'utilizzo di quattro parametri*:

- Tre sono le componenti del versore $\boldsymbol{r} = [\boldsymbol{r_x} \;\; \boldsymbol{r_y} \;\; \boldsymbol{r_z}]^{\boldsymbol{T}}$ appartenente all'asse di rotazione attorno al quale essa viene realizzata;
- Il quarto è l'angolo $\boldsymbol{\theta}$ di cui ruotiamo attorno ad r.

A partire dalla conoscenza di θ ed r è possibile ricavare la matrice di rotazione $R(\theta, r)$ corrispondente appunto alla rotazione di un angolo θ attorno al versore r. Per fare ciò supponiamo di avere una terna fissa $Oxyz$ come in figura e il versore r di un certo asse di rotazione.



Usando la convenzione per cui l'angolo è positivo se la rotazione avviene in senso antiorario, la matrice $R(\theta, r)$ si costruisce mediante le seguenti rotazioni elementari:

1. Allineare r con z, che si ottiene come successione di una rotazione per $-\alpha$ su z e una rotazione di $-\beta$ su y;
2. Ruotare di un angolo θ attorno a z;
3. Riallineare con la direzione iniziale di r, che si ottiene come sequenza di una rotazione di attorno a y di un angolo β e di una rotazione di a attorno a z.

$$R(\theta, r) = R_z(a)R_y(\beta)R_z(\theta)R_y(-\beta)R_z(-a)$$

4. Osserviamo che in prima battuta nella matrice $R(\theta, r)$ comparire funzioni seno e coseno dipendenti dagli angoli a e β che tuttavia sono solo angoli ausiliari che non riguardano la nostra rotazione. Allora per liberare la matrice di rotazione di questi angoli sarà necessario esprimere questi ultimi in funzione di r_x, r_y, r_z (come di seguito).

$$\sin a = \frac{r_y}{\sqrt{r_x^2 + r_y^2}} \qquad \cos a = \frac{r_x}{\sqrt{r_x^2 + r_y^2}} \qquad \sin \beta = \sqrt{r_x^2 + r_y^2} \qquad \cos \beta = r_z$$

5. Così facendo la matrice risultante dipenderà sono dai parametri desiderati θ, r_x, r_y, r_z. Troviamo così la **Matrice di Rotazione della Rappresentazione Asse-Angolo**:

$$\boldsymbol{R(\theta, r)} = \begin{bmatrix} \boldsymbol{r_x^2(1 - c_\theta) + c_\theta} & \boldsymbol{r_x r_y(1 - c_\theta) - r_z s_\theta} & \boldsymbol{r_x r_y(1 - c_\theta) + r_y s_\theta} \\ \boldsymbol{r_x r_y(1 - c_\theta) + r_z s_\theta} & \boldsymbol{r_y^2(1 - c_\theta) + c_\theta} & \boldsymbol{r_y r_z(1 - c_\theta) - r_x s_\theta} \\ \boldsymbol{r_x r_z(1 - c_\theta) - r_y s_\theta} & \boldsymbol{r_y r_z(1 - c_\theta) + r_x s_\theta} & \boldsymbol{r_z^2(1 - c_\theta) + c_\theta} \end{bmatrix}$$

E' possibile considerare anche il *problema inverso*, ovvero a partire dalla matrice di rotazione R (di fianco) vogliamo ricavare θ ed r. Ciò può essere fatto mediante le seguenti relazioni e il vincolo a cui sono soggette:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\boldsymbol{\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)} \qquad \boldsymbol{r = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}} \qquad \boldsymbol{Vincolo}: \; r_x^2 + r_y^2 + r_z^2 = 1$$

Osservando tali relazioni scopriamo che la rappresentazione asse-angolo presenta una singolarità di rappresentazione quando $\sin\theta = 0$ (niente rotazione o rotazione di 180°) e ciò avviene per $\theta = 0, \pi$. Tuttavia, tale condizione non è particolarmente problematica in quanto avviene al confine dello spazio di lavoro del manipolatore. Motivo per cui la rappresentazione asse-angolo è particolarmente vantaggiosa nel problema della pianificazione della traiettoria di un manipolatore. In particolare, si presta in maniera adeguata per la parte di orientamento dell'end-effector in quanto tale rappresentazione esplica la rotazione rispetto all'asse di simmetria attorno a cui essa avviene. Tuttavia, con l'adozione della rappresentazione asse/angolo, esiste il problema per cui la rotazione di un angolo θ intorno ad r è indistinguibile dalla rotazione di un angolo $-\theta$ intorno ad r, cioè tale rappresentazione non è unica e infatti genera la medesima matrice di rotazione: $R(\theta, r) = R(-\theta, -r)$

1.2.3 Quaternione Unitario

Il **Quaternione Unitario** $\boldsymbol{Q = \{\eta, \varepsilon\}}$ è uno strumento impiegato per realizzare una rappresentazione non minima dell'orientamento di un corpo nello spazio costituito da:

- **Parte Scalare** $\boldsymbol{\eta}$: $\boldsymbol{\eta = \cos(\theta/2)}$
- **Parte Vettoriale** $\boldsymbol{\varepsilon}$: $\boldsymbol{\varepsilon = \sin(\theta/2) \cdot r = [\varepsilon_x \;\; \varepsilon_y \;\; \varepsilon_z]^T}$

Tali quantità sono soggette al **vincolo**: $\boldsymbol{\eta^2 + \varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2 = 1}$

Questa rappresentazione presenta diverse analogie con la rappresentazione asse angolo come la presenza di una parte scalare e vettoriale. Inoltre, anche per il quaternione unitario con ragionamenti analoghi possiamo estrapolare una **Matrice di Rotazione del Quaternione** $\boldsymbol{R(\eta, \varepsilon)}$:

$$\boldsymbol{R(\eta,\varepsilon) = \begin{bmatrix} 2(\eta^2+\varepsilon_x^2)-1 & 2(\varepsilon_x\varepsilon_y-\eta\varepsilon_z) & 2(\varepsilon_x\varepsilon_z+\eta\varepsilon_y) \\ 2(\varepsilon_x\varepsilon_y+\eta\varepsilon_z) & 2(\eta^2+\varepsilon_y^2)-1 & 2(\varepsilon_y\varepsilon_z-\eta\varepsilon_x) \\ 2(\varepsilon_x\varepsilon_z-\eta\varepsilon_y) & 2(\varepsilon_y\varepsilon_z+\eta\varepsilon_x) & 2(\eta^2+\varepsilon_z^2)-1 \end{bmatrix}}$$

Grazie al quaternione unitario siamo in grado di superare i limiti della rappresentazione asse/angolo in cui avevamo il problema di non essere in grado di discriminare la rotazione di un angolo θ intorno ad r e la rotazione di un angolo $-\theta$ intorno ad $-r$. Questo perché entrambe le rotazioni generano lo stesso quaternione e quindi garantiscono l'unicità di rappresentazione.

Risolviamo ora il problema inversa supponendo di conoscere la matrice di rotazione di fianco. La parte scalare e la parte vettoriale del quaternione possono essere ricavati mediante le seguenti relazioni:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\eta = \frac{1}{2}\sqrt{r_{11}+r_{22}+r_{33}+1} \qquad \varepsilon = \frac{1}{2}\begin{bmatrix} sgn(r_{32}-r_{23})\sqrt{r_{11}-r_{22}-r_{33}+1} \\ sgn(r_{13}-r_{31})\sqrt{r_{22}-r_{33}-r_{11}+1} \\ sgn(r_{21}-r_{12})\sqrt{r_{33}-r_{11}-r_{22}+1} \end{bmatrix}$$

In queste relazioni si è assunto che la parte scalare è positiva, quindi $\eta = \cos(\theta/2) \geq 0$. Ciò implica che l'argomento del coseno è compreso da $\pm\pi/2$ e quindi $\theta \in [-\pi, \pi]$. Così facendo, il quaternione unitario non è soggetto alla singolarità di rappresentazione che colpisce la rappresentazione asse-angolo. Infine, citiamo due proprietà del quaternione unitario:

- **Proprietà 1 – Inversa del Quaternione**: grazie al fatto che $R^{-1} = R^T$, possiamo dire che l'inversa di un quaternione è: $Q^{-1} = \{\eta, -\varepsilon\}$
  Dunque, se io associo un quaternione all'orientamento in una terna rispetto all'altra, cioè vado da una terna ad un'altra con η ed ε, posso ritornare indietro semplicemente cambiando segno al secondo parametro.
- **Proprietà 2 – Prodotto di Quaternioni**: dati due quaternioni $Q_1 = \{\eta_1, \varepsilon_1\}$ e $Q_2 = \{\eta_2, \varepsilon_2\}$, corrispondenti alle matrici di rotazione R_1 ed R_2. Il quaternione corrispondente al prodotto R_1R_2 è dato da: $Q_1 * Q_2 = \{\eta_1\eta_2 - \varepsilon_1^T\varepsilon_2, \eta_1\varepsilon_2 + \eta_2\varepsilon_1 + \varepsilon_1 \times \varepsilon_2\}$
  Inoltre è facile dimostrare che, $Q_2 = Q_1^{-1}$: $Q_1 * Q_2 = Q_1 * Q_1^{-1} = \{1,0\}$

1.3 Rappresentazione Minima dell'Orientamento

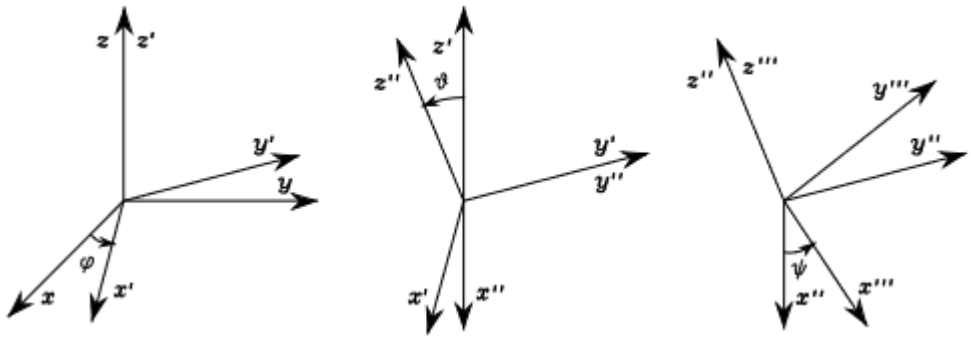
1.3.1 Angoli di Eulero

Gli **Angoli di Eulero $\boldsymbol{\phi} = [\boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\psi}]^{\boldsymbol{T}}$** sono una rappresentazione minima dell'orientamento ottenuta *formalmente* dalla combinazione in cascata di tre rotazione elementari intorno a tre assi. Mediante gli angoli di Eulero, poiché abbiamo a disposizione tre angoli e tre assi, possiamo esprime $3^3 = 27$ possibili rotazioni, alcune delle quali saranno rotazioni intorno ad assi paralleli. Allora di queste 27 solo 12 terne descrivono rotazioni successive intorno ad assi non paralleli tra loro e quindi non rappresentati la stessa informazione. Citiamo due di queste possibili 12 combinazioni:

- **Terna ZYZ** descrive le seguenti rotazioni:
  1. Rotazione della terna di riferimento intorno all'asse z di un angolo φ. Questa rotazione è descritta dalla matrice $R_z(\varphi)$;
  2. Rotazione della terna corrente intorno all'asse y' di un angolo θ. Questa rotazione è descritta dalla matrice $R_{y'}(\theta)$;
  3. Rotazione della terna corrente intorno all'asse z'' di un angolo ψ. Questa rotazione è descritta dalla matrice $R_{z''}(\psi)$.

  Poiché si tratta di rotazioni *in terna corrente* la matrice di rotazione risultate è la seguente:

$$\boldsymbol{R(\phi) = R_z(\varphi) R_{y'}(\theta) R_{z''}(\psi)} = \begin{bmatrix} \boldsymbol{c_\varphi c_\theta c_\psi - s_\varphi s_\psi} & \boldsymbol{-c_\varphi c_\theta s_\psi - s_\varphi c_\psi} & \boldsymbol{c_\varphi s_\theta} \\ \boldsymbol{s_\varphi c_\theta c_\psi + c_\varphi s_\psi} & \boldsymbol{-s_\varphi c_\theta s_\psi + c_\varphi c_\psi} & \boldsymbol{s_\varphi s_\theta} \\ \boldsymbol{-s_\theta c_\psi} & \boldsymbol{s_\theta s_\psi} & \boldsymbol{c_\theta} \end{bmatrix}$$



Risolviamo ora il problema inverso in cui lo scopo è ricavare gli angoli di Eulero corrispondenti ad una data matrice di rotazione R. Comparando la R assegnata e la $R(\phi)$ poc'anzi definitiva, sotto l'assunzione che $r_{13} \neq 0$ e $r_{23} \neq 0$, otteniamo la seguenti relazioni dipendendi dal range di θ:

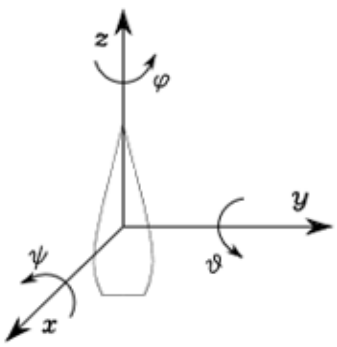
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$Se\ \theta \in (0, \pi)$$
$$\varphi = Atan2(r_{23}, r_{13})$$
$$\theta = Atan2\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right)$$
$$\psi = Atan2(r_{32}, -r_{31})$$

$$Se\ \theta \in (-\pi, 0)$$
$$\varphi = Atan2(-r_{23}, -r_{13})$$
$$\theta = Atan2\left(-\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right)$$
$$\psi = Atan2(-r_{32}, r_{31})$$

I casi $\theta = 0$ e $\theta = \pi$, che corrispondono a rotazioni successive di φ e ψ intorno ad assi paralleli, portano a delle **Singolarità di Rappresentazione**, ovvero siamo in grado al massimo di ricavare $\varphi \pm \psi$.

- **Angoli rpy**: si tratta di un'altra terna di angoli di Eulero che viene tipicamente impiegata in ambito aeronautico. Essi, infatti, denotano i cambiamenti di assetto di un aereo e in particolari di rotazioni rispetto ad un telaio fisso vincolato al centro di massa di quest'ultimo. Questa terna è descritta dalle seguenti rotazioni:



1. **Imbardata** (**y**): rotazione della terna di riferimento intorno a x di un angolo ψ. Questa rotazione è descritta dalla matrice $R_x(\psi)$;
2. **Beccheggio** (**p**): rotazione della terna fissa intorno ad y di un angolo θ. Questa rotazione è descritta dalla matrice $R_y(\theta)$;
3. **Rollio** (**r**): rotazione della terna fissa intorno a z di un angolo φ. Questa rotazione è descritta dalla matrice $R_z(\varphi)$.

Poiché si tratta di rotazioni *in terna fissa* la matrice di rotazione risultate è la seguente:

$$\boldsymbol{R(\phi) = R_z(\varphi)R_y(\theta)R_x(\psi) = \begin{bmatrix} c_\varphi c_\theta & c_\varphi s_\theta s_\psi - s_\varphi c_\psi & c_\varphi s_\theta c_\psi + s_\varphi s_\psi \\ s_\varphi c_\theta & s_\varphi c_\theta s_\psi + c_\varphi c_\psi & s_\varphi s_\theta c_\psi - c_\varphi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}}$$

E' da notare che essendoci le funzioni seno e coseno tutti i nove elementi della matrice di rotazione saranno sempre minori di uno.

Risolviamo ora il problema inverso in cui lo scopo è ricavare gli angoli di Eulero corrispondenti ad una data matrice di rotazione R. Comparando la R assegnata e la $R(\phi)$ poc'anzi definitiva, otteniamo la seguenti relazioni dipendendi dal range di θ:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$Se\ \theta \in (-\pi/2, \pi/2)$$
$$\varphi = Atan2(r_{21}, r_{11})$$
$$\theta = Atan2\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right)$$
$$\psi = Atan2(r_{32}, r_{33})$$

$$Se\ \theta \in (\pi/2, 3\pi/2)$$
$$\varphi = Atan2(-r_{21}, -r_{11})$$
$$\theta = Atan2\left(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}\right)$$
$$\psi = Atan2(-r_{32}, -r_{33})$$

I casi $\theta = \pm\pi/2$, che corrispondono a rotazioni successive di φ e ψ intorno ad assi paralleli, portano a delle **Singolarità di Rappresentazione**, ovvero siamo in grado al massimo di ricavare $\varphi \pm \psi$.

Per concludere, osserviamo che delle dodici terne di angoli di Eulero la metà sono soggette alle singolarità per $\theta = 0, \pi$, mentre l'altra metà alle singolarità per $\theta = \pm\pi/2$.

1.4 Trasformazione Omogenea

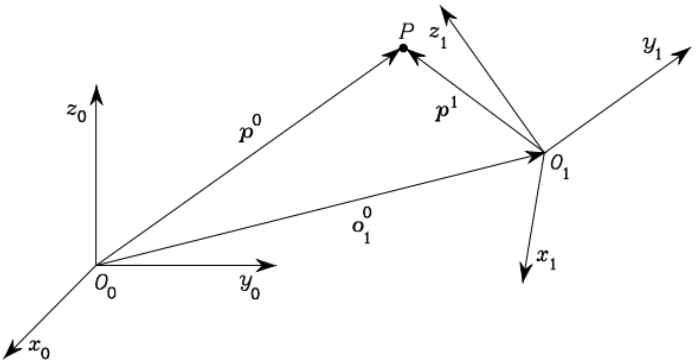
Una **Trasformazione Omogenea** *è una rappresentazione compatta della posa (posizione e orientamento) di un corpo rigido nello spazio*. Tale trasformazione è ottenuta mediante una **Matrice di Trasformazione Omogenea** $\boldsymbol{A_j^i}$, nella quale l'apice indica la terna a cui ci referiamo mentre con il pedice la terna corrente. Inoltre, i vettori soggetti a questa trasformazione sono scritti in **Forma Omogenea**, ovvero oltre alle tre classiche componenti ne presentano una quarta unitaria.

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \rightarrow \tilde{p} = \begin{bmatrix} p \\ 1 \end{bmatrix}$$

Per comprendere come utilizzare la trasformazione omogenea partiamo ricordando che la posizione di un corpo nello spazio è individuata in termini di:

- Posizione di un apportuno punto solidale con il corpo rigido (traslazione);
- Orientamento, espresso in base alle componenti dei versori degli assi di una terna solidale al corpo stesso (rotazione).

Ora, consideriamo una terna 0 di riferimento e una terna 1, l'idea è definire uno strumento che metta insieme la parte di posizione e orientamento. La posizione della terna 1 rispetto alla terna 0 ci viene data dal vettore o_1^0. Consideriamo ora un punto P generico di cui conosciamo p^1, ovvero la posizione rispetto alla terna 1, mentre vogliamo ricavare p^0, ovvero la posizione rispetto alla terna 0. Ora ci chiediamo quale è la relazione che lega i vettori p^0 e p^1. Possiamo sfruttare la regola del parallelogramma, nella quale però, affichè possiamo sommare in maniera consistente o_1^0 che è definito rispetto alla terna 0, dobbiamo moltiplicare quest'ultimo per la matrice di rotazione R_1^0:



$$p^0 = o_1^0 + R_1^0 p^1$$

La trasformazione omogenea, che quindi tiene conto sia della traslazione che della rotazione, è data dalla seguente relazione in cui $\tilde{p}^0$ e $\tilde{p}^1$ sono la forma omogena dei vettori p^0 e p^1:

$$\boldsymbol{\tilde{p}^0} = \begin{bmatrix} \boldsymbol{p^0} \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{3x3} & | & \boldsymbol{3x1} \\ - - & + & - \\ \boldsymbol{1x3} & | & \boldsymbol{1x1} \end{bmatrix} \begin{bmatrix} \boldsymbol{p^1} \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_1^0} & \boldsymbol{o_1^0} \\ \boldsymbol{0^T} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{p^1} \\ \boldsymbol{1} \end{bmatrix} = \boldsymbol{A_1^0 \tilde{p}^1}$$

Eventualmente possiamo effettuare anche la trasformazione inversa: $p^1 = -R_0^1 o_1^0 + R_0^1 p^0$

A questo punto potremmo chiederci in quale ambito potremmo impiegare lo strumento della trasfromazione omogena. Ebbene, in seguito, quando verrà trattata la cinematica diretta di un manipolatore verranno impiegati vettori scritti proprio in forma omogena. Per concludere, è importante osservare che, come la matrice R, anche la matrice di trasformazione omogenea può essere ottenuta come composizione di trasformazioni successive: $\tilde{p}^0 = A_1^0 A_1^0 \dots A_n^{n-1} \tilde{p}^n$

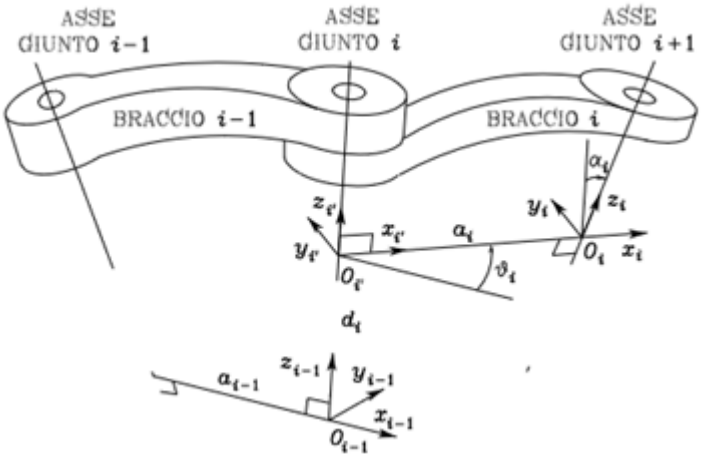
Tuttavia, contrariamente alla matrice di rotazione, la matrice di trasformazione omogenea *non presenta* la proprietà di ortogonalità: $A^{-1} \neq A^T$

$$\tilde{p}^1 = A_0^1 \tilde{p}^0 = (A_1^0)^{-1} \tilde{p}^0 = \begin{bmatrix} R_0^1 & -R_0^1 o_1^0 \\ 0^T & 1 \end{bmatrix} \tilde{p}^0$$

1.5 Convenzione di Denativ-Hartenberg

La **Convenzione di Denavit-Hartenberg** (**D-H**) consiste nel posizionare le terne dei link costituenti il robot secondo le seguenti regole:

- ✓ Si sceglie l'asse z_i giacente lungo l'asse del giunto $i+1$;
- ✓ Si individua O_i all'intersezione dell'asse z_i con la normale comune agli assi z_{i-1} e z_i, e con O' si indica l'intersezione della normale comune con z_{i-1};
- ✓ Si assume l'asse x_i diretto lungo la normale comune agli assi z_{i-1} e z_i con verso positivo dal giunto i al giunto $i+1$;
- ✓ Si sceglie l'asse y_i in modo da completare la terna levogira.



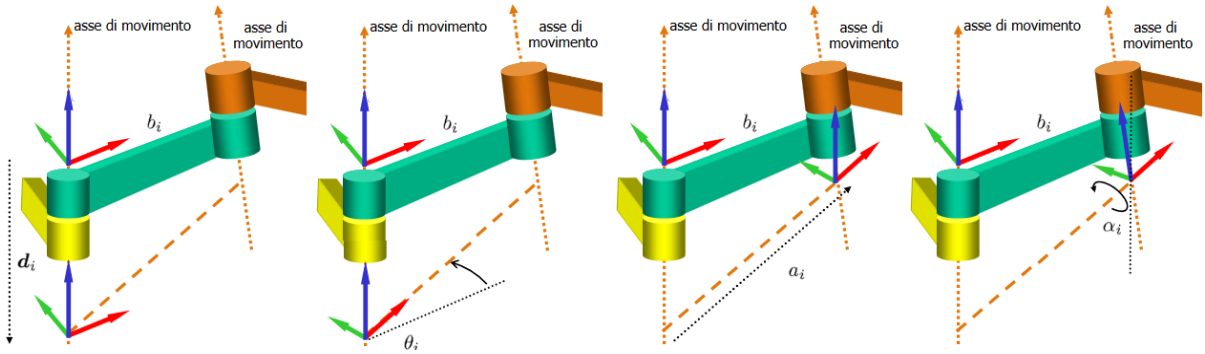
La convenzione di D-H fornisce, per ogni riferimento, un numero di gradi di libertà rispetto al riferimento precedente caratterizzato da quattro parametri:

❖ **Parametri Traslazionali**:
- a_i = distanza di O_i da O_i';
- d_i = coordinata su z_{i-1} di O_i';

❖ **Parametro Rotoidali**:
- α_i = angolo intorno all'asse x_i tra l'asse z_{i-1} e l'asse z_i valutato positivo in senso antiorario;
- θ_i = angolo intorno all'asse z_{i-1} tra l'asse x_{i-1} e l'asse x_i valutato positivo in senso antiorario.

I quattro parametri di fatto è come se rappresentassero un insieme di traslazioni e rotazioni per portare la terna del braccio precedente sulla terna del braccio successivo secondo quanto mostrato in figura:

1. Traslazione lungo z_{i-1} di una quantità d_i per portare il riferimento $i-1$ alla quota della normale comune;
2. Rotazione intorno z_{i-1} di un angolo θ_i per allineare x_i con il segmento di minina distanza;
3. Traslazione lungo x_i di a_i per allineare sullo stesso asse le origini delle due terne consecutive;
4. Rotazione intorno x_i di un angolo α_i per allineare z_{i-1} e z_i completando così la sovrapposizione delle terne.



Tre di questi quattro parametri dipendono esclusivamente dalla geometria del robot e dunque sono costanti. La **Variabile di Giunto** è il quarto parametro che invece sarà dipendente dal tempo:

- ✓ La variabile di giunto è θ_i nel caso di un giunto rotoidale;
- ✓ La variabile di giunto è d_i nel caso di un giunto prismatico.

A questo punto però, osservando le regole della convenzione, capiamo che nascono dei problemi circa il link 0 e link n visto che il primo non ha un giunto a monte mentre il secondo a valle:

- ❖ **Link 0**: visto che è presente il giunto 1 possiamo posizionare z_0 coincidente con l'asse di tale giunto. Invece, la posizione dell'orgine della terna O_0 e la direzione di x_0 sono arbitrarie. L'asse y_0 ovviamente deve completare la terna levogira;
- ❖ **Link $\boldsymbol{n}$**: essendo l'ultimo link non esiste un giunto $i+1$ e quindi avremmo il problema di posizionare z_n. Motivo per cui quello che possiamo fare è scegliere z_n parallelo z_{n-1}, mentre l'orgine e x_n saranno arbitrari. E' da notare che, se il giunto n è *rotoidale* possiamo posizionare O_n alla stessa quota di O_{n-1} in modo da annullare d_n (visto che è costante nei giunti rotoidale). Nel caso di un giunto prismatico allora z_n è determinato mentre x_n è arbitrario.

Ora che abbiamo posizionato le terne secondo la convenzione di Denavit-Hartenberg, vediamo come si realizza la trasformazione omogenea che ci da la posa della terna i rispetto alla terna $i-1$ (facciamo riferimento all'immagine di inizio paragrafo):

$$A_i^{i-1}(q_i) = A_{i'}^{i-1}A_i^{i'} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ -s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dove:

- $A_{i'}^{i-1}$ = matrice di trasformazione omogenea che rototrasla la terna $i-1$ sulla terna i'. In essa compare la matrice di rotazione elementare che esprime la rotazione intorno all'asse z_i di un angolo θ_i;
- $A_i^{i'}$ = matrice di trasformazione omogenea che rototrasla la terna i' sulla terna i. . In essa compare la matrice di rotazione elementare che esprime la rotazione intorno all'asse x_i di un angolo α_i;

1.6 Cinematica Diretta

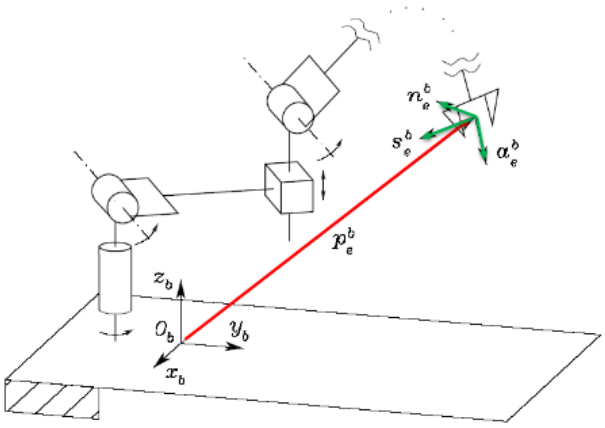
1.6.1 Cinematica Diretta per Catene Aperte

La **Cinematica Diretta** *si pone l'obiettivo di determinare la posa, quindi posizione e orientamento, dell'end effector del robot a partire dai valori delle variabili di giunto.* Analiticamente, l'obiettivo è costruire una matrice di trasformazione omogena $T_e^b(q)$ rispetto ad un sistema di riferimento base b che dipende dal vettore delle variabili di giunto $q = [q_1 \quad q_2 \quad \dots \quad q_n]^T$ (le varie q_i sono o metri oppure radianti a seconda della natura del giunto).

$$T_e^b(q) = \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) & p_e^b(q) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Supponiamo di avere un manipolare costituiti da diversi link (linee) e giunti (rotoidali e prismatici). In questo schema definiamo due sistemi di riferimento:

- Terna base b soldiale dal piano principale;
- Terna utensile e solidale all'end effector in cui i tre versori sono:
  - a_e (approccio): è il versore che punta della direzione di approccio dell'end effector rispetto all'oggetto da manipolare;
  - s_e (scorrimento): è il versore che punta nella direzione di scorrimento degli elementi dell'end effector, in questo caso nella direzione di apertura/chiusura delle ganasce della pinza;
  - n_e (normale): è il versore che deve essere normale agli altri due per completare la terna levogira.



Ovviamente dovremo mettere degli apici b perché stiamo ragionando il tutto rispetto alla terna base.

Tipicamente, la $T_e^b(q)$ può essere ricavata nel seguente modo:

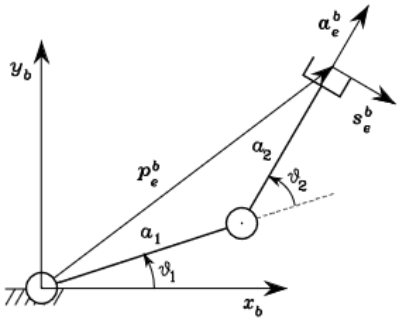
$$\boldsymbol{T_e^b(q) = T_0^b T_n^0(q) T_e^n = T_0^b\left[A_1^0(q_1) \cdot A_1^0(q_1) \cdot \dots A_n^{n-1}(q_n)\right] T_e^n}$$

Dove:

- ✓ T_0^b = matrice di trasformazione omogena *nota* che ci restituisce la posa del link 0 rispetto alla terna base;
- ✓ T_e^n = matrice di trasformazione omogena *nota* che ci restituisce la posa dell'end effector rispetto alla terna n dell'n −esimo link;
- ✓ $T_n^0(q)$ = matrice di trasformazione omogena che ci restituisce la posa del link n rispetto alla terna 0 del link 0. Tale matrice si calcola semplicemente post-moltiplicando la matrice $A_i^{i-1}(q_i)$ che legano i diversi link $i-1$ ed i consecutivi legati tra loro mediante *un solo giunto* che concede un solo grado di libertà. Infatti, ciascuna di queste matrici A sarà funzione solo di una variabile di giunto q_i.

Esempio: supponiamo di avere un robot planare a due bracci di cui vogliamo calcolare la posa dell'end effector o, per essere più precisi, di un punto soldiale all'end effector che tipicamente si trova al centro dalla pinza). In questo esempio supponiamo che la terna b e 0 coincidono così come per la terna e e la terna n. Attraverso delle semplici considerazione di carattere geometrico sui possibili triangoli rettandoli che possiamo considerare riusciamo a calcolare la prima riga della matrice $T_e^b(q)$.

$$T_e^b(q) = \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) & p_e^b(q) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & s_{12} & c_{12} & a_1c_1 + a_2c_{12} \\ 0 & -c_{12} & s_{12} & a_1s_1 + a_2s_{12} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Seppur queste considerazioni di carattere geometrico risultano essere piuttosto intuitive per le semplici catene analizzate, la complessità di tale approccio cresce all'aumentare della complessità della catena cineamtica oggetto di studio. Per cui come vedremo ci servirà una *procedura sistematica* ad hoc per calcolare la cinematica diretta. La risoluzione delle cinematica diretta risulta essere agevolata dall'applicazione della convenzione di Denavit-Hartenberg, dalla quale perveniamo ad una procedura applicazione a tutte le possibili catene cinematiche aperte. Andiamo ad analizzare tale procedura:

1. Individuare e numerare consecutivamente gli assi dei giunti. Assegnare, rispettivamente le direzioni agli assi $z_0, \dots, z_{n-1}$;
2. Fissare la terna base posizionandone l'origine sullasse z_0. Gli assi x_0 e y_0 sono scelti in maniera tale da ottenere una terna levogira;
3. Individuare l'origine O_i all'intersezione di z_i con la normale comune agli assi z_{i-1} e z_i. Ora:
   a. Se tali assi sono paralleli e il giunto i è rotoidale, posizionare O_i in modo da annullare d_i;
   b. Se il giunto è prismatico, scegliere O_i in corrispondenza di una posizione di riferimento per la corsa del giunto (per esempio fine corsa).
4. Fissare l'asse $x_i$diretto lungo la normale comune agli assiz_{i-1} e $z_i$con verso positivo dal giunto i al giunto $i+1$;
5. Fissare l'asse y_i in modo da ottenere una terna levogira;
6. Ripetere i passi 3,4,5 per $i = 1, \dots, n-1$;
7. Fissare la terna n, allineando z_n lungo la direzione di z_{n-1} se il giunto n è rotoidale, ovvero scegliendo z_n in maniera arbitraria se il giunto n è prismatico; fissare l'asse Xn in accordo al punto 4;
8. Costruire per $i = 1, \dots, n$ la tabella dei parametri $a_i, d_i, \alpha_i, \theta_i$;
9. Calcolare, sulla base dei parametri del punto precedente, le matrici di trasformazione omogena $A_i^{i-1}(q_i)$ per $i = 1, \dots, n$;
10. Calcolare $T_n^0(q) = A_1^0 \dots A_n^{n-1}$ che fornisce la posizione e orientamento della terna n rispetto alla terna 0;
11. Assegnate T_0^b e T_e^n, calcolare la funzione cinematica diretta $T_e^b(q) = T_0^b T_n^0 T_e^n$ che fornisce posizione e orientamento della terna utensile rispetto alla terna base.

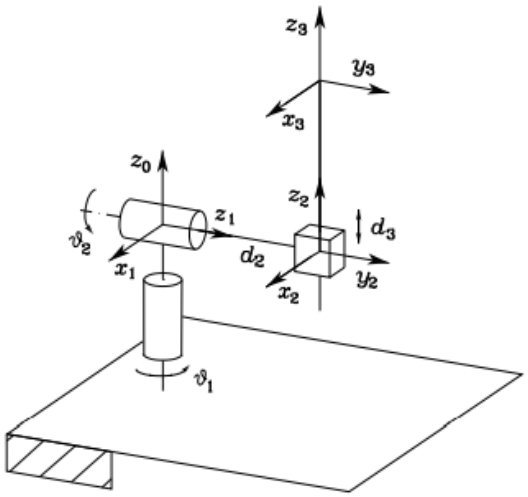
I tre aspetti però che di fatto ci fanno apprezzare la potenzialità della convenzione di denavit-hartenberg sono i seguenti:

- ✓ La tabella di parametri ci consente di calcolare facilmente le matrici $A_i^{i-1}(q_i)$ e di conseguenza la T_n^0 da impiegare per la cinematica diretta;
- ✓ La cinematica ci consente di descrivere il manipolatore in termini puramente geometrici. Lo step successivo è la cinematica differenziale che vi aggiunge il fattore tempo. Uno strumento fondamentale in questo ambito è la matrice Jacobiana, cioè la matrice i cui elementi sono le derivate parziali prime della funzione. Il calcolo di tali elementi e quindi delle derivate ha chiaramente un costo computazionale. Ebbene grazie alla tabella di parametri ottenuta dalla convenzione di Denavit-Hartenberg saremo in grado di calcolare le colonne della Jacobiana senza effettuare alcun calcolo;
- ✓ Uno step ancora successivo sarà la dinamica dei robot in cui saranno inclusi anche la velocità e l'accellerazione. In questo ambito, sempre grazie alla tabella di parametri saremo in grado di scrivere i termini della matrice di inerzia, il vettore di gravità, i termini di forza centrifuga e di forza di coriolis.

Ultima osservazione a cui bisogna porre attenzione riguarda la distinguibilità tra una terna di angoli dovuti alla convenzione di Denavit-Hartenberg e una terna di angoli di Eulero. Ricordiamo che nella terna di angoli di Eulero le matrici di rotazione dipendono solo ed esclusivamente dai suddetti angoli. Ora, seppur le matrici di rotazione complessiva ottenuta con ambi gli strumenti è identica, la matrice A_i^{i-1} definita dalla convenzione dipende in realtà da coppie di angoli θ_i ed a_i. Proprio a causa di ciò possiamo affermare con certezza che una terna di angoli $\theta_i, \theta_j, \theta_k$ dovuti alla convenzione non sono degli angoli di Eulero.

Esempio – Manipolatore Sferico:

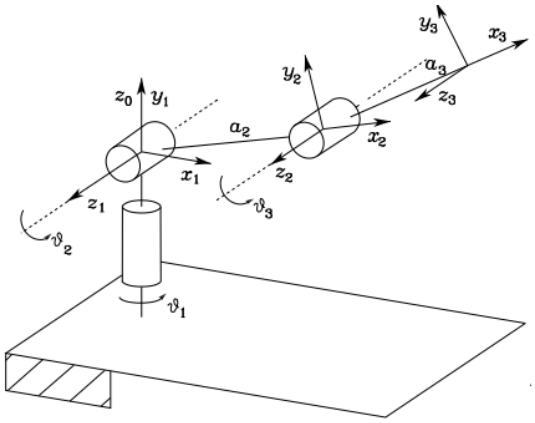
Braccio	a_i	α_i	d_i	ϑ_i
1	0	$-\pi/2$	0	ϑ_1
2	0	$\pi/2$	d_2	ϑ_2
3	0	0	d_3	0



$$
\begin{aligned}
\boldsymbol{T}_3^0 &= \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \boldsymbol{A}_3^2 \\
&= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & c_1 s_2 d_3 - s_1 d_2 \\ s_1 c_2 & c_1 & s_1 s_2 & s_1 s_2 d_3 + c_1 d_2 \\ -s_2 & 0 & c_2 & c_2 d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Esempio – Manipolatore Antropomorfo:

Braccio	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3



$$\boldsymbol{A}_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

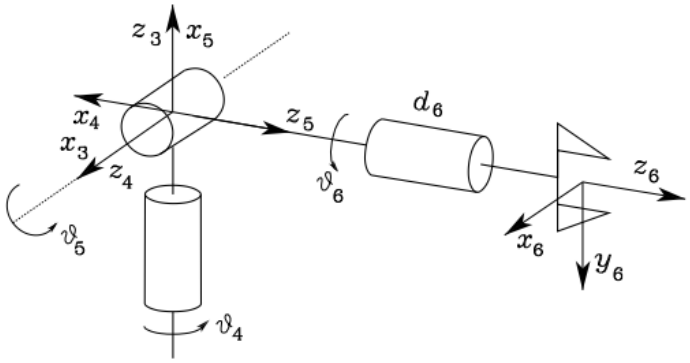
$$\boldsymbol{A}_i^{i-1} = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad i = 2, 3$$

$$\boldsymbol{T}_3^0 = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \boldsymbol{A}_3^2$$

$$= \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & c_1(a_2 c_2 + a_3 c_{23}) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & s_1(a_2 c_2 + a_3 c_{23}) \\ s_{23} & c_{23} & 0 & a_2 s_2 + a_3 s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esempio – Polso Sferico:

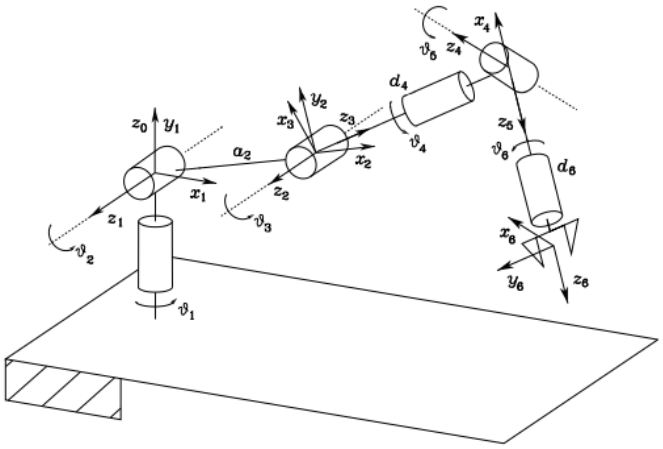
Braccio	a_i	α_i	d_i	ϑ_i
4	0	$-\pi/2$	0	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6



$$
\begin{aligned}
\boldsymbol{T}_6^3 &= \boldsymbol{A}_4^3 \boldsymbol{A}_5^4 \boldsymbol{A}_6^5 \\
&= \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Esempio – Manipolatore Antropomorfo con Polso Sferico:

Braccio	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	0	$\pi/2$	0	ϑ_3
4	0	$-\pi/2$	d_4	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6



$$\boldsymbol{T}_6^0 = \begin{bmatrix} \boldsymbol{n}^0 & \boldsymbol{s}^0 & \boldsymbol{a}^0 & \boldsymbol{p}^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{p}^0 = \begin{bmatrix} a_2 c_1 c_2 + d_4 c_1 s_{23} + d_6\big(c_1(c_{23}c_4 s_5 + s_{23}c_5) + s_1 s_4 s_5\big) \\ a_2 s_1 c_2 + d_4 s_1 s_{23} + d_6\big(s_1(c_{23}c_4 s_5 + s_{23}c_5) - c_1 s_4 s_5\big) \\ a_2 s_2 - d_4 c_{23} + d_6(s_{23}c_4 s_5 - c_{23}c_5) \end{bmatrix}$$

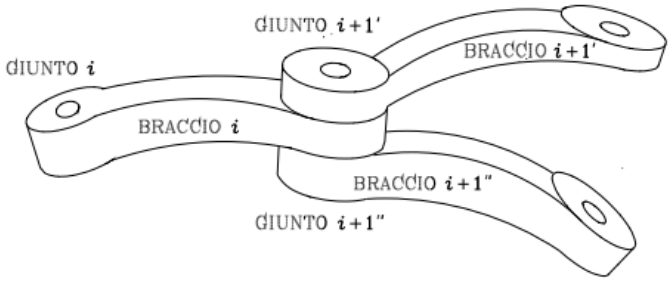
$$\boldsymbol{n}^0 = \begin{bmatrix} c_1\big(c_{23}(c_4 c_5 c_6 - s_4 s_6) - s_{23}s_5 c_6\big) + s_1(s_4 c_5 c_6 + c_4 s_6) \\ s_1\big(c_{23}(c_4 c_5 c_6 - s_4 s_6) - s_{23}s_5 c_6\big) - c_1(s_4 c_5 c_6 + c_4 s_6) \\ s_{23}(c_4 c_5 c_6 - s_4 s_6) + c_{23}s_5 c_6 \end{bmatrix}$$

$$\boldsymbol{s}^0 = \begin{bmatrix} c_1\big(-c_{23}(c_4 c_5 s_6 + s_4 c_6) + s_{23}s_5 s_6\big) + s_1(-s_4 c_5 s_6 + c_4 c_6) \\ s_1\big(-c_{23}(c_4 c_5 s_6 + s_4 c_6) + s_{23}s_5 s_6\big) - c_1(-s_4 c_5 s_6 + c_4 c_6) \\ -s_{23}(c_4 c_5 s_6 + s_4 c_6) - c_{23}s_5 s_6 \end{bmatrix}$$

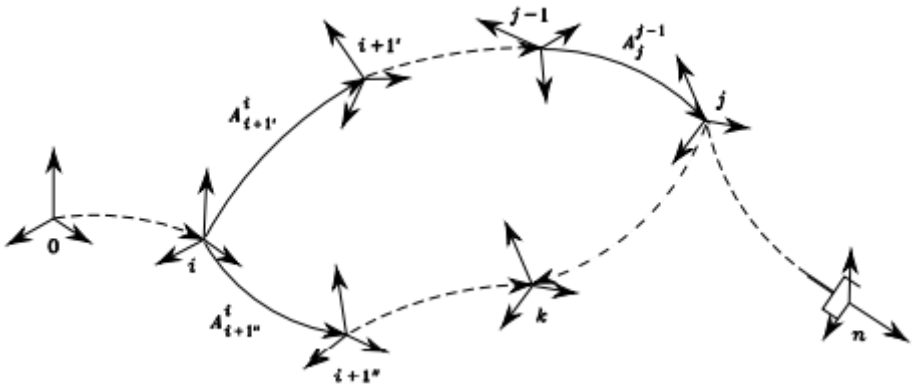
$$\boldsymbol{a}^0 = \begin{bmatrix} c_1(c_{23}c_4 s_5 + s_{23}c_5) + s_1 s_4 s_5 \\ s_1(c_{23}c_4 s_5 + s_{23}c_5) - c_1 s_4 s_5 \\ s_{23}c_4 s_5 - c_{23}c_5 \end{bmatrix}$$

1.6.2 Cinematica Diretta per Catene Chiuse

In realtà la procedura analizzata nel sotto-paragrafo precedente, a fronte di una modifica che analizzeremo, può essere applicata anche al caso delle catene cinematiche chiuse. Cominciamo questo studio immaginando di avere un manipolatore a catena chiusa costituito da $n+1$ link. In particolare, dal link 0 al link i, quest'ultimo azionato dal giunto i, costituiscono una catena cinematica aperta. A rendere il manipolatore una catena chiusa è la presenza di un anello chiuso di cui in figura è mostrato il tratto iniziale in cui abbiamo che il link i è collegato ai link $i+1'$ e $i+1'$ mediante due giunti che supponiamo essere allineati. I due rami dell'anello si ricongiungono mediante i link j e k, collegati aloro volta mediante il giunto $j+1$.



Tipicamente in una catena cinematica chiusa sono presenti dei giunti non attuati (o passivi), cioè non sono controllati attivamente da alcun attuatore. L'idea per procedere in questo scenario è definire un **Giunto di Taglio** come quel giunto non attuato in corrispondenza del quale apro la catena cinematica. Perciò apriamo la catena per esempio tra j e k, creando così due catene cinematiche aperte.



Ricordiamo che la parte difficile della cinematica diretta è calcolare $T_n^0(q)$, la quale abbiamo visto essere il prodotto delle matrici di trasformazioni omogenee $A_i^{i-1}(q_i)$. In questo scenario possiamo scrivere $T_n^0(q)$ come: $T_n^0(q) = A_i^0 A_j^i A_n^j$

Dove:

- A_i^0 = matrice di trasformazione della parte iniziale aperta della catena cinematica;
- A_j^i = matrice di trasformazione di un ramo dell'anello chiuso;
- A_n^j = matrice di trasformazione che va dall'estremo di una dei rami della catena all'ultimo braccio. Dunque potrà essere solo una delle due seguenti matrici di rotazione:

$$A_j^i(q') = A_{i+1'}^i(q_{i+1'}) \dots A_j^{j-1}(q_j)$$
$$A_k^i(q'') = A_{i+1''}^i(q_{i+1''}) \dots A_k^{k-1}(q_k)$$

A questo punto però potremmo chiederci quale complessità appone la presenza di una catena cinematica chiusa.

Ebbene nel giunto in cui ho aperto la catena, il quale nella realtà è un giunto di chiusura, dobbiamo porre dei vincoli di posizione e rotazione sul giunto $j+1$:

- ✓ **Vincolo 1**: immaginiamo che il giunto j e il giunto k sono allineati. Per la convenzione D-H i loro assi z devono *coincidere*: $z_j^i(q') = z_k^i(q'')$
- ✓ **Vincolo 2** (Giunto Prismatico): se il giunto di taglio è prismatico allora il θ relativo tra i due giunti non varia. In altre parole, gli assi x delle due terne sovrapposte nel giunto di taglio devono avere orientamento costante.

$$x_j^{iT}(q')x_k^i(q'') = \cos\theta_{jk} = costante$$

- ✓ **Vincolo 3** (Giunto Rotoidale): la posizione relativa tra le origini delle due terne j e k, riproiettate in terna j, sia 0 sulle componenti x ed y mentre sia costante lungo z:

$$R_i^j(q')\left(p_j^i(q') - p_k^i(q'')\right) = [0 \quad 0 \quad d_{jk}]^T$$

- ✓ **Vincolo 4** (Giunto Prismatico): dal precedente vincolo, visto che ho un giunto prismatico, estraggo solo due componenti:

$$\begin{bmatrix} x_j^{iT}(q') \\ y_j^{iT}(q') \end{bmatrix}\left(p_j^i(q') - p_k^i(q'')\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

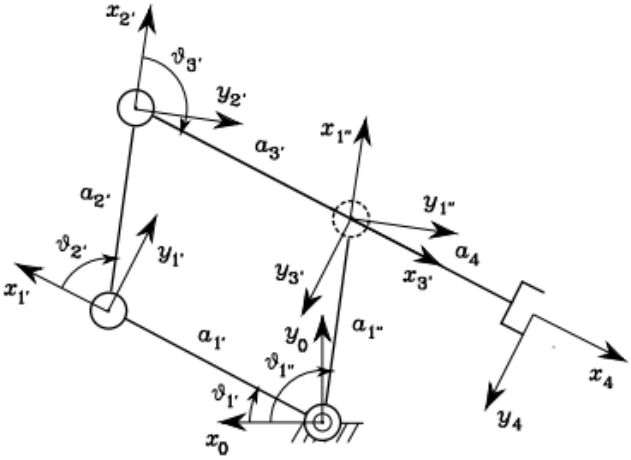
In definitiva, per ambo le tipologie di giunti avremo sempre un sistemi di vincoli con sei equazione. A questo punto però nasce la problema della risoluzione di queste equazioni in forma analitica. Ebbene ci sono delle strutture che ci facilitano la risoluzione mentre altre risulta essere particolarmente complesso e saranno perciò necessari metodi numerici. Questi vincoli e la cinematica diretta non vanno scritti per tutti i giunti della catena chiusa (quinti tutti i q' e tutti i q'') ma bensì solo per un *numero minimo di giunti* e che in particolare sono solo quelli attuati.

Per concludere, vale la pena descrivere in maniera generica la procedura per calcolare la funzione cinematica diretta per un manipolatore a catena chiusa usando la convenzione di Denavit-Hartenberg:

1. Selezionare un giunto non attuato della catena chiusa. Ipotizzare di aprire tale giunto in modo da ottenere una catena aperta con struttura ad albero;
2. Calcolare le trasformazioni omogenee secondo la convenzione di Denavit-Hartenberg;
3. Trovare i vincoli di uguaglianza per le due terne connesse dal giunto di taglio;
4. Risolvere i vincoli in termini di un numero ridotto di variabili di giunto;
5. Esprimere le trasformazioni omogenee in funzione di tali variabili di giunto e calcolare la funzione cinematica diretta per composizione della varie trasformazioni dalla terna base alla terna utensile.

Esempio – Robot a Parallelogramma: supponiamo di avere la seguente catena cinematica chiusa nella quale il giunto 0 è attuato mentre i tre superiori non sono attuati. Inoltre, scegliamo il giunto 3 (quello tratteggiato) come giunto di taglio. Applichiamo la convenzione D-H per ambo le catene aperte appena ottenute e definiamo la relativa tabella.

Braccio	a_i	α_i	d_i	ϑ_i
$1'$	$a_{1'}$	0	0	$\vartheta_{1'}$
$2'$	$a_{2'}$	0	0	$\vartheta_{2'}$
$3'$	$a_{3'}$	0	0	$\vartheta_{3'}$
$1''$	$a_{1''}$	0	0	$\vartheta_{1''}$
4	a_4	0	0	0



Prendiamo come riferimento per la cinematica diretta il ramo superiore della catena cinematica e quindi calcoliamo la matrice di trasformazione omogenea tra la terna 0 e la terna $3'$:

$$\boldsymbol{A}_{3'}^{0}(\boldsymbol{q}') = \boldsymbol{A}_{1'}^{0}\boldsymbol{A}_{2'}^{1'}\boldsymbol{A}_{3'}^{2'} = \begin{bmatrix} c_{1'2'3'} & -s_{1'2'3'} & 0 & a_{1'}c_{1'} + a_{2'}c_{1'2'} + a_{3'}c_{1'2'3'} \\ s_{1'2'3'} & c_{1'2'3'} & 0 & a_{1'}s_{1'} + a_{2'}s_{1'2'} + a_{3'}s_{1'2'3'} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In realtà ci converebbe calcolare anche la matrice $A_{1''}^{0}$ in quanto non dobbiamo dimenticare di costruire l'insieme dei vincoli che rappresentano il collegamento fisico dei giunti sovrapposti. I vincoli sono:

- **Orientamento**: è soddisfatta perché gli assi z delle terna $3'$ e $1''$ sono già allineati nel giunto di taglio visto che il robot è planare: $z_{3'}(q') = z_{1''}(q'')$
- **Posizione**: essendo tutti giunti rotoidali dobbiamo porre il seguente vincoli circa le posizioni relative delle terne.

$$R_0^{3'}(q')\left(p_{3'}^{0}(q') - p_{1''}^{0}(q'')\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{array}{l} x:\ a_{1'}(c_{1'} + c_{1'2'3'}) + a_{1''}(c_{1'2'} - c_{1''}) = 0 \\ y:\ a_{1'}(s_{1'} + s_{1'2'3'}) + a_{1''}(s_{1'2'} - s_{1''}) = 0 \end{array}$$

Come si può osservare d_{jk} è costante visto che è nulla, per cui il vincolo è soddisfatto.

Possiamo dire che: $\theta_{2'} = \theta_{1''} - \theta_{1'}$ $\qquad$ $\theta_{3'} = \pi - \theta_{2'} = \pi - \theta_{1''} + \theta_{1'}$

Per concludere, la cinematica diretta è:

$$T_4^0(q) = A_{3'}^{0}(q)A_4^{3'} = \begin{bmatrix} -c_1' & s_1' & 0 & a_{1''}c_{1''} - a_4c_{1'} \\ -s_{1'} & -c_{1'} & 0 & a_{1''}c_{1''} - a_4c_{1'} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.7 Spazio dei Giunti, Spazio Operativo e Spazio di Lavoro

Lo **Spazio dei Giunti** è lo spazio in cui è definito il vettore $n \times 1$ delle n variabili di giunto q_i. Esse risulteranno essere pari a θ_i nel caso di un giunto rotoidale oppure pari a d_i per uno prismatico. E' precisare che il vettore dei giunti è definito nel *senso dell'algebra*, quindi sarebbe più corretto chiamarlo array colonna in quanto le dimensioni delle variabili *sono differenti* (per giunti prismatici sono i metri mentre per giunti rotoidali sono i radianti).

$$q = \begin{bmatrix} q_1 \\ \cdots \\ q_n \end{bmatrix} \qquad dove: \quad \begin{matrix} q_i = \theta_i & se\ giunto\ rotoidale \\ q_i = d_i & se\ giunto\ prismatico \end{matrix}$$

Abbiamo visto che la cinematica diretta ci consente di esprimere la posizione e l'orientamento della terna dell'end-effector rispetto alla terna base in funzione della variabili di giunto. Ora se dobbiamo specificare un task per il nostro robot, sarà necessario assegnare la posizione e l'orientamento dell'end effector magari anche in funzione del tempo per realizzare una traiettoria. Per quanto riguarda la posizione bastano le solite tre coordinate, mentre per quanto riguarda l'orientamento abbiamo visto in prima battuta la matrice di rotazione. Tale rappresentazione però è ridondante in quanto impiega nove parametri ed è quindi abbastanza complicato impiegarla per definire l'orientamento dell'end-effector. A tal proposito sono nate la rappresentazioni con un numero inferiore di parametri. In definitiva, possiamo pensare di esprimere la posa dell'end effector mediante un vettore colonna $(m \times 1)$ con $m \leq n$ affinchè il robot possa eseguire il task. Per esempio potremmo avere un compito per il quale servono un certo numero m di parametri sufficienti a descrivere la posa dell'end-effector oppure un altro numero m di parametri per aggiungere anche l'esigenza di estendere la manipolazione al resto del robot. Lo **Spazio Operativo** è un spazio strettamente legato al task (infatti in inglese si chiama Operation Space o Task Space) in cui è definito il vettore x_e che esprime la posa dell'end-effector con un numero $m \leq n$ di parametri scelti a seconda del task da eseguire.

$$x_e = \begin{bmatrix} p \\ \phi \end{bmatrix} = \begin{bmatrix} posizione \\ orientamento \end{bmatrix}$$

Poiché esiste una dipendenza tra la posa dell'end-effector e le variabili di giunto, possiamo definire l'**Equazione Cinematica Diretta**: $\boldsymbol{x = k(q)}$

La **Ridondanza Cinematica** *è una condizione per la quale, indicando con n il numero di giunti del robot ed m la dimensione dello spazio del compito, si ha che:* $\boldsymbol{n > m}$
In pratica, il robot ha più gradi di libertà di quanti non siano realmente necesssari per svolgere il compito. Capiamo dunque che, il concetto di ridondanza è strettamente legato al compito che il robot deve svolgere. Vediamo alcuni esempi di task con relative dimensioni:

- ✓ Posizionamento nel piano → $m = 2$;
- ✓ Posizionamento nello spazio→ $m = 3$;
- ✓ Orientamento nel piano → $m = 1$;
- ✓ Puntamento nello piano → $m = 2$;
- ✓ Posizionamento e orientamento nello spazio → $m = 6$;

Per esempio, un robot planare con $n = 3$ giunti è ridondante rispetto al compito del posizionamento nel piano in cui $m = 2$, ma non è ridondante rispetto al compito posizionamento e orientamento nel piano in cui $m = 3$.

Una caso particolare della ridondanza è la **Ridondanza Funzionale**, ovvero quando $m = n$ però sono un numero $r < m$ componenti dello spazio operativo interessano il compito specifico. Un esempio è il tipico robot industriale a sei DOF per il quale $m = n = 6$ che è funzionalmente ridondante nei confronti del task tagli laser che coinvolge solo $r = 5$ variabili di giunto (la direzione di avvicinamento è irrilevante). Andiamo ora ad riassumere vantaggi e svantaggi della ridondanza. I principali vantaggi sono:

- ✓ Migliore capacità di evitare gli ostacoli nello spazio cartesiano o le singolarità cinematiche nello spazio dei giunti;
- ✓ Aumentare la manipolabilità in determinate azioni;
- ✓ Distribuire più uniformemente le velocità ai singoli giunti;
- ✓ Aumentare l'affidabilità rispetto ai guasti.

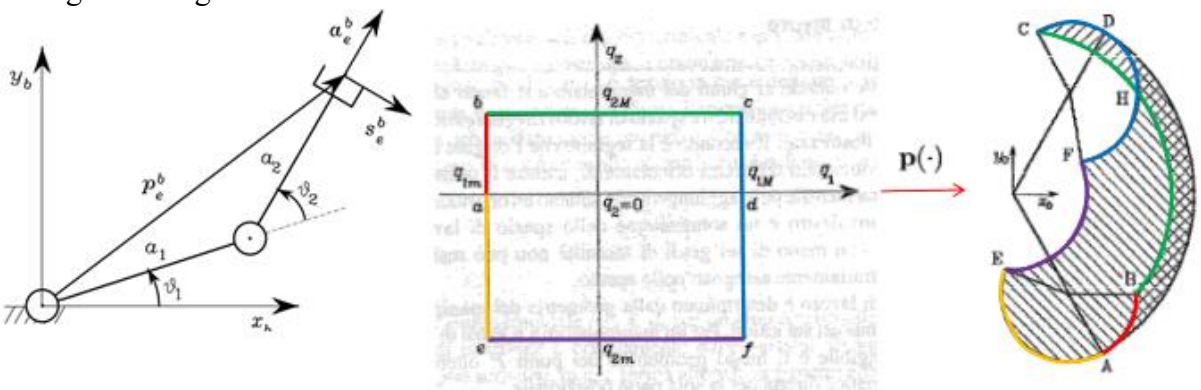
Gli svantaggi invece sono:

- o Maggiore complessità strutturale in termini di meccanica e attuazione;
- o Algoritmi di cinematica inversa e di controllo più complessi.

Lo **Spazio di Lavoro Raggiungibile** è definito come lo spazio di lavoro che possiamo descrivere con l'organo terminale facendo variare tutte le variabili di giunto dai rispettivi valori minimi ai valori massimi: $p = p(q) \qquad q_{im} \leq q_i \leq q_{iM} \qquad i = 1, \dots, n$

Dal punto di vista pratico lo spazio di lavoro raggiungibile sarà quindi un volume la cui superfice esterna, in virtù del fatto che abbiamo solo giunti prismatici o rotoidali, sarà una superfice planare, sferica, cilindrica oppure toroidale. Si definisce invece **Spazio di Lavoro Destro** (nel senso di Destrezza) quella porzione dello spazio di lavoro raggiungibile in cui il manipolatore può assumere orientamenti diversi.

Esempio: verifichiamo quale sia lo spazio dei giunti e lo spazio di lavoro del manipolatore planare a due bracci. E' da notare che lo spazio dei giunti è un rettangolo in cui la parte sopra all'asse delle ordinate sono tutte le configurazione gomito basso, contrariamente abbiamo le configurazioni gomito alto.



Alle lettere in minuscolo dello spazio dei giunti corrispondono le lettere grandi dello spazio di lavoro, il quale è una corona circolare.

1.8 Calibrazione Cinematica

La **Calibrazione Cinematica** *è una procedura il cui scopo è correggere i quattro parametri della convenzione D-H al fine di migliorare l'accuratezza del manipolatore*. In particolare, si tratta di una procedura che deve essere eseguita perché le lavorazioni con cui sono realizzati i componenti del robot hanno precisione finita (per esempio un braccio che nel datasheep del robot ha una misura nominale di $50\ cm$ avrà una misura reale che oscilla intorno al valore nominale) ma soprattuto deve essere ripetuta periodicamente a causa dell'usura in seguito all'impiego del robot. Infine, la calibrazione non è da confondere con la procedura di Homing che avviene all'avvio del sistema, in cui quest'ultimo viene messo in una posizione di partenza da cui poi posizionare le terne.

Fatte queste premesse sulla calibrazione cinematica, lo scopo ora è andare a caratterizzare tale procedura. Elemento fondamentale a cui fa riferimento l'intera procedura sono le cosidette **Posizioni Franche**, ovvero configurazioni del robot in cui la posa dell'organo terminale è definita con una elevatissima precisione così come i parametri della convenzione D-H. In pratica sono delle posizioni di riferimento rispetto alle quali effettueremo la calibrazione. A questo punto, per iniziare lo studio della procedura di calibrazione dinamica, riprendiamo l'equazione cinematica diretta: $x = k(q)$

A seconda del giunto, la q mi definisce la variabile di quest'ultimo (θ per un giunto rotoidale e d per un giunto prismatico) mentre gli altri tre parametri della convenzione di D-H sono fissi. Ebbene, con la calibrazione cinematica, è come se dovessimo mettere in discussione tutte e quattro le variabile, cioè dal punto di vista operativo supponiamo che siano tutte variabili. Dunque, l'equazione cinematica diretta diventa: $x = k(a, \alpha, d, \theta)$

Riassumendo i risultati della calibrazione cinematica saranno:

- ✓ Correzioni sui valori delle tre parametri costanti della convezione D-H affetti da inaccuratezza meccanica;
- ✓ Correzione sul valore della variabile di giunto. La correzione di una quantità variabile è altrettanto importante perché essa viene ottenuta mediante appositi sensori, encoder, ecc che per loro natura presentano incertezza di misura.

Ora, per ottenere queste correzioni, quindi i valori veri dei quattro parametri, sarebbe comodo impiegare dei metodi di identificazione analitici. Il problema però di questi metodi è sono impiegabili sono per sistemi lineari contrariamente al caso di un sistema robotico. Quanto detto ci suggerisce che la strada da intraprendere sarà quella della linearizzazione. Allora indichiamo con:

- x_m = posa misurata, cioè quella che misuriamo mediante strumenti di misura di natura ottica o meccanica in modo da garantire la più alta precisione possibile;
- x_n = posa nominale cioè la posa ricavata dai parametri della tabella D-H e dalla lettura degli encoder (parametri fissi + variabili di giunto).

Allora per poter impiegare i metodi di identificazione dei sistemi lineari dobbiamo linearizzare l'equazione cinematica diretta nell'intorno del valore nominale dei parametri di D-H:

$$\Delta x = x_m - x_n = \frac{\partial k}{\partial a}\Delta a + \frac{\partial k}{\partial \alpha}\Delta \alpha + \frac{\partial k}{\partial d}\Delta d + \frac{\partial k}{\partial \theta}\Delta \theta = \phi(\xi_n)\Delta\xi = \phi(\xi_n)(\xi_m - \xi_n)$$

Dove:
- Δx è un vettore colonna di dimensione $m \times 1$;
- $\phi(\xi_n)$ è una matrice bassa di dimensione $m \times 4n$;
- $\Delta\xi$ è un vettore colonna di dimensione $4n \times 1$.

Quello che ci interessa nell'ultima relazione è $\Delta\xi$, ovvero lo scostamento tra x_m ed x_n, che potremmo calcolare andando a risolvere il sistema di m equazioni in $4n$ incognite rappresentato dalla matrice $\phi(\xi_n)$. Essendo una matrice basse, il sistema avrà più incognite che equazioni e dunque avremo un numero infinito di soluzioni. Dopodichè, sommare a ξ_n la $\Delta\xi$ appena ricavate in modo da avere il valore vero dei parametri di D-H. Tuttavia, possiamo prendere una strada differente ragionando sul semplice risultato pratico per il quale una misura è tanto più precisa quanto maggiore sono il numero di misurazioni. Allora possiamo supporre di fare l misurazioni tale che $l \cdot m \gg 4n$, da cui possiamo scrivere:

$$\Delta\bar{x} = \begin{bmatrix} \Delta x_1 \\ \dots \\ \Delta x_l \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \dots \\ \phi_l \end{bmatrix} \Delta\xi = \bar{\phi}\Delta\xi$$

Questa volta però, la matrice $\bar{\phi}$ risulta essere alta di dimensione $lm \times 4n$ anziché bassa, per cui non possiamo ricavare una soluzione esatta da un sistema che ha più equazioni che incognite. Bisogna però ricordare che la nostra esigenza primaria è avere una maggiore accuratezza, la quale si ottiene da un numero crescente di misurazioni e il problema appena citato può essere risolto adottando la matrice pseudoinversa di $\bar{\phi}$ (sinistra visto che moltiplicata da sinistra per $\bar{\phi}$ ci da la matrice identità): $\Delta\xi = (\bar{\phi}^T\bar{\phi})^{-1}\bar{\phi}^T\Delta\bar{x}$

Proprio poiché il sistema ha più equazioni che incognite, la soluzione appena trovata è una *soluzione a norma minima*, ovvero essa minimizza istantaneamente lo scarto tra $\Delta\bar{x}$ e $\bar{\phi}\Delta\xi$.

Ottenuto $\Delta\xi$ possiamo calcolare il nuovo set di parametri nominali: $\xi' = \xi_n + \Delta\xi$

A questo punto rifacciamo la stessa procedura descritta fino ad ora partendo dal nuovo set di parametri ξ' e da una delle posizioni franche definite inizialmente. In questo modo arriveremo a calcolare un altro set di parametri che indicheremo con ξ''. Da ciò possiamo concludere che la calibrazione cinematica è una *procedura iterativa*, la quale si ripeterà fin quando il modulo di $\Delta\xi$ non sarà minore di una certa soglia: $|\Delta\xi| < \varepsilon$

E' da notare che la medesima procedura verrà applicata anche nella dinamica in cui, seppur il modello dinamico è molto più complesso di quello cinematico (vista la presenza di masse, accellerazioni, gravità, ecc.) è un modello lineare rispetto ai parametri di D-H. Quindi dalla calibrazione dinamica (identificazione dei parametri dinamici) non otterremo le correzioni dei parametri ma proprio questi ultimi.

Nota: in futuro uno strumento fondamentale che useremo è la matrice Jacobiana, ovvero la matrice delle derivate parziali (rispetto alla componenti vettoriali) di una funzione vettoriale. La matrice ϕ citata precedentemente è a tutti gli effetti uno Jacobiano.

1.9 Cinematica Inversa

La **Cinematica Inversa** *si pone l'obiettivo di determinare i valori delle variabili di giunto a partire dalla posa, quindi posizione e orientamento, dell'end effector del robot*. Dunque, nella cinematica inversa è assegnata la matrice di trasformazione omogenea T della terna dell'effector rispetto alla terna base oppure la x (dell'equazione cinematica diretta) e vogliamo determinare il vettore delle variabili di giunto q. Capiamo dunque che il problema cinematico inverso è il problema duale della cinematica diretta dove, al contrario, è assegnato q e dobbiamo ricavare T o x. La cinematica inversa risulta essere per sua natura un problema più complesso rispetto alla cinema diretta in quanto:

- *Non sempre* esiste una soluzione in forma chiusa visto che le equazioni da risolvere sono non lineari. Motivo per cui spesso saranno necessari l'adozione di tecniche numeriche, le quali hanno il vantaggio di essere applicabili a qualsiasi struttura cinematica;
- Possono esistere soluzioni multiple, quindi più di una oppure addirittura infinite se stiamo parlando di un manipolatore ridondante. La loro esistenza è garantita solo se la posa dell'organo terminale appartiene allo spazio di lavoro del manipolatore. E' da specificare che, qualora fossero presenti dei limiti di giunzione meccanica è possibile ridurre il numero di soluzioni ammissibili per la struttura reale;
- Non sempre esistono soluzioni ammissibili.

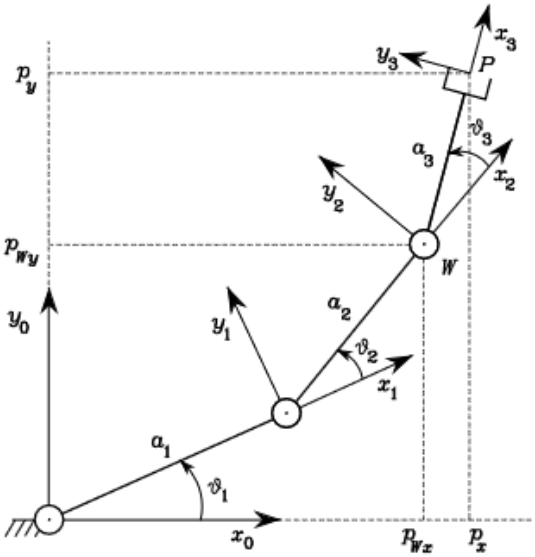
Nel seguito vedremo alcuni esempi di risoluzione delle cinematica inversa sfruttando intuizioni algebriche e geometriche. Tuttavia, dobbiamo tener presente che tali intuizioni non risultano essere delle soluzioni robuste perché non impedescono di avere complicazioni in corrispondenza delle configurazioni singolari del robot e nel loro intorno. Invece, l'obiettivo centrale nel seguito sarà ricercare uno strumento robusto con cui risolvere la cinematica inversa.

Esempio: consideriamo il manipolatore planare a tre bracci in figura. La risoluzione della cinematica inversa in questo caso consiste nel trovare le espressioni della variabili di giunto $\theta_1, \theta_2, \theta_3$ per una data posa dell'organo terminale. La posa dell'end-effector la specifichiamo come di seguito:

- **Posizione**: coordinate p_x e p_y;
- **Orientamento**: angolo ϕ con l'asse x_0, il quale risulterà essere: $\phi = \theta_1 + \theta_2 + \theta_3$

Da ciò possiamo scrivere l'equazione cinematica diretta in cui conosciamo p_x, p_y, ϕ:

$$x(k) = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$



Si può dimostrare che la posizione $[p_{Wx}, p_{Wy}]$ della terna 2, ovvero quella precedente all'end-effector, è data dalle espressioni di fianco. Quadrando e sommando queste due quantità:

$$p_{W_x} = p_x - a_3 c_\phi = a_1 c_1 + a_2 c_{12}$$
$$p_{W_y} = p_y - a_3 s_\phi = a_1 s_1 + a_2 s_{12}$$

$$p_{W_x}^2 + p_{W_y}^2 = a_1^2 + a_2^2 + 2a_1 a_2 c_2$$

Da quest'ultima relazione possiamo ricavare l'espressione di c_2, la quale a sua volta ci consente di calcolare anche l'espressione di s_2 grazie alla relazione $c_2^2 + s_2^2 = 1$:

$$s_2 = \pm\sqrt{1 - c_2^2}$$

In particolare, il segno positivo lo abbiamo nelle posture del tipo elbow-down, mentre il segno negativo per le posture elbow-up. Perciò possiamo affermare che: $\theta_2 = Atan2(s_2, c_2)$

Sostituiamo θ_2 nelle espressioni di p_{Wx}, p_{Wy} in modo da ricavare con alcuni passaggi algebrici θ_1, il quale avrà come espressione: $\theta_1 = Atan2(s_1, c_1)$

Per differenza possiamo così trovare anche l'espressione dell'ultima variabile di giunto θ_3 in modo da poter risolvere la cinematica inversa: $\theta_3 = \phi - \theta_1 - \theta_2$

Se l'orientamento ϕ non fosse assegnato e il manipolatore fosse ridondante, la cinematica inversa avrebbe infinite soluzioni.

---

2. Cinematica Differenziale e Statica

2.1 Caratterizzazione della Cinematica Differenziale

La **Cinematica Differenziale** *studia la relazione tra velocità dei giunti $\dot{q}$ e velocità dell'organo terminale (velocità lineari $\dot{p}_e$ e angolari ω_e).* In altre parole, vogliamo esprimere la velocità (lineare $\dot{p}_e$ e angolare ω_e) dell'end-effector in funzione delle velocità ai giunti $\dot{q}$ attraverso una *relazione lineare*. L'**Equazione Cinematica Differenziale** è la relazione lineare *nelle velocità* che lega v_e e $\dot{q}$:

$$\dot{q} \rightarrow v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} \quad \Leftrightarrow \quad v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} J_P(q)\dot{q} \\ J_O(q)\dot{q} \end{bmatrix} = J(q)\dot{q}$$

Lo **Jacobiano Geometrico $J(q)$** è la matrice $6 \times n$ che caratterizza in cui:

$$J = \begin{bmatrix} J_{P1} & & J_{Pn} \\ & \dots & \\ J_{O1} & & J_{On} \end{bmatrix}$$

- **Righe**: abbiamo sempre sei righe perché abbiamo tre componenti che riguardano il contributo del giunto alla velocità lineare e gli ultimi tre sono il contributo alla velocità angolare;
- **Colonne**: ciascuna colonna è associata ad un giunto, quindi per n giunti avremo n colonne ciascuna con sei righe.

Le colonne dello Jacobiano possono essere calcolate a partire dalle relazioni già fornite nella cinematica diretta mediante la convenzione di D-H e risultano essere le seguenti a seconda del tipo di giunto:

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \textit{se } i \text{ è } \textit{un giunto prismatico} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \textit{se } i \text{ è } \textit{un giunto rotoidale} \end{cases}$$

Dove:

- $\boldsymbol{z_{i-1}}$: l'asse z della terna $i-1$ ed è la terza colonna della matrice di rotazione R_{i-1}^0 (la quale è data dal prodotto di matrici $R_1^0 \dots R_{i-1}^{i-2}$), la quale è ricavabile moltiplicando quest'ultima per il vettore $z_0 = [0 \;\; 0 \;\; 1]^T$: $\boldsymbol{z_{i-1} = R_1^0 \dots R_{i-1}^{i-2} \cdot z_0}$
- $\boldsymbol{p_e}$: la posizione dell'organo terminale rispetto alla terna 0 ed è la quarta colonna della matrice di trasformazione omogena $A_1^0(q_1) \dots A_n^{n-1}$, la quale è ricavabile moltiplicando quest'ultima per il $p_0 = [0 \;\; 0 \;\; 0 \;\; 1]$: $\boldsymbol{p_e = A_1^0(q_1) \dots A_n^{n-1}(q_n) \cdot p_0}$
- $\boldsymbol{p_{i-1}}$: la posizione della terna $i-1$ rispetto alla terna 0 ed è la quarta colonna della matrice di trasformazione omogena $A_1^0(q_1) \dots A_{i-1}^{i-2}(q_{i-1})$, la quale è ricavabile moltiplicando quest'ultima per il $p_0 = [0 \;\; 0 \;\; 0 \;\; 1]$: $\boldsymbol{p_{i-1} = A_1^0(q_1) \dots A_{i-1}^{i-2}(q_{i-1}) \cdot p_0}$

Osservando le definizioni date, ci accorgiamo che grazie all'impiego simultaneo dello Jacobiano e della convenzione di D-H abbiamo direttamente le relazioni che ci servono per risolvere la cinematica differenziale.

Ora che abbiamo caratterizzato completamente la cinematica differenziale e definito lo Jacobiano, andiamo a studiare come ottenere questi risultati. AnzituttoPerò noi sappiamo che le velocità lineari e angolari sono le derivate temporali della posizione, mentre grazie alle relazioni prima citate abbiamo completamente aggirato le necessità, e quindi lo sforzo computazionale, di effettuare la derivazione. Affrontando la cinematica differenziale è intuitivo, sapendo il legame tra posizione e velocità, cercare di ricorrere alla derivata. Tale approccio è semplice quando parliamo p_e e $\dot{p}_e$, discorso ben diverso per quanto riguarda $R_e(q)$ e ω_e. Visto che ω_e è un vettore 3×1, esso non può essere la derivata $R_e(q)$ che è una matrice 3×3. Allora per capire che legame tra ω_e ed $R_e(q)$ dobbiamo fare una serie di considerazioni atte a definire la *derivata di una matrice di rotazione*. Prendiamo la relazione che definisce una matrice ortonormale, ovvero che il prodotto di una matrice di questo tipo per la sua trasposta è pari alla matrice identità. Se introduciamo il parametro tempo questa relazione deve valere per ogni istante di tempo: $RR^T = I \quad \rightarrow \quad R(t)R^T(t) = I \quad \forall t$

Deriviamo poi rispetto al tempo ambo i membri e osserviamo che il secondo termine è il trasposto del primo. Perciò indicheremo il primo con S e il secondo con S^T. Se $S = -S^T$ allora S sarà una matrice antisimmetrica e dunque avrà una struttura come mostrata di seguito:

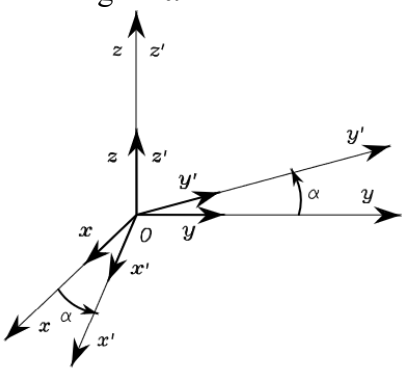
$$\left[\dot{R}R^T\right] + \left[R\dot{R}^T\right] = 0 \quad \rightarrow \quad S + S^T = 0 \quad \rightarrow \quad S = \begin{bmatrix} 0 & s_1 & s_2 \\ -s_1 & 0 & s_3 \\ -s_2 & -s_3 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

I tre componenti s_1, s_2, s_3 sono proprio gli elementi che caratterizzano la velocità angolare $\omega_e = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, per cui sarebbe più corretto usare la notazione $S\big(\omega(t)\big)$ vista la dipendenza dal tempo.

Facciamo un esempio per dimostrare quanto detto circa la relazione tra ω_e e R.

Esempio: consideriamo il caso di una rotazione elementare di un angolo α attorno all'asse z di cui conosciamo la matrice di rotazione.

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Poiché l'esempio è molto semplice noi già sappiamo che $\omega = [0 \;\; 0 \;\; \dot{a}]^T$, perciò vediamo se otteniamo lo stesso risultato usando la matrice S. Calcolando le matrici costituendi S e poi facende il prodotto otteniamo quest'ultima. I tre termini sopra la diagonale principale sono le componenti della velocità angolare:

$$S\big(\omega(t)\big) = \dot{R}(t)R^T(t) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{a} & 0 \\ \dot{a} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \rightarrow \quad \omega = \begin{bmatrix} 0 \\ 0 \\ \dot{a} \end{bmatrix}$$

Avendo posto che: $S(\omega(t)) = \dot{R}(t)R^T(t)$,

Scopriamo che la **Derivata della Matrice di Rotazione** si può calcolare come il prodotto tra l'operatore antisimmetrico S e la matrice di rotazione stessa: $\boldsymbol{\dot{R}(t) = S\big(\omega(t)\big)R(t)}$

Ora che abbiamo dimostrato che la velocità angolare ω_e non è la derivata di R dobbiamo capire come legare questi due elementi. Immaginiamo di avere un vettore p' costante e un secondo vettore p ottenuto come rotazione in funzione del tempo: $p(t) = R(t)p'$

La derivata rispetto al tempo di $\dot{p}(t)$ può essere scritta in uno dei seguenti modi, dove il secondo descrive la velocità del punto descritto dal vettore p:

a. $\dot{p}(t) = \dot{R}(t)p' = S\big(\omega(t)\big)R(t)p'$
b. $\dot{p}(t) = \omega(t) \times p(t) = \omega(t) \times R(t)p'$

Confrontando queste due relazioni perviamo a questo fondamentale risultato, ovvero che l'operatore antisimmetrico descrive il prodotto vettoriale con il vettore velocità angolare, perciò possiamo scrivare che: $\omega(t) \times = S\big(\omega(t)\big)$

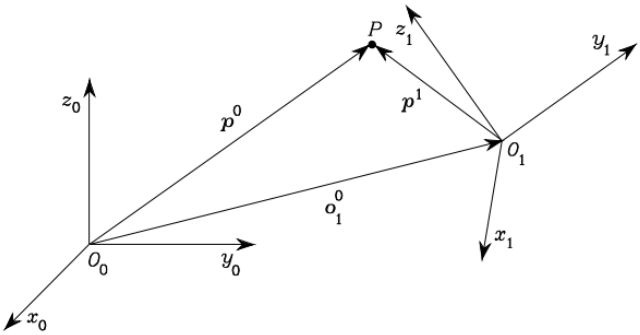
Questo risultato ha anche una interpretazione fisica che possiamo dimostrare ricordando come si svolge il prodotto vettoriale tra due generici vettori a e b. Dalla teoria dell'algebra lineare il risultato del prodotto vettoriale è il vettore il blu. Tuttavia, esso può essere espresso anche come il prodotto tra matrice verde e il vettore b.

$$a \times b = \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

La matrice in questione è proprio l'operatore antisimmetrico S, il quale rappresenta il prodotto vettoriale $a \times$ con il vettore di interesse b: $a \times b = S(a) \cdot b = -S(b) \cdot a$

Il passo successivo è studiare come scrivere le relazioni che descrivono le velocità lineari e angolari. Supponiamo di avere una terna fissa 0 e una terna 1 con un certo orientamento rispetto alla prima definita dalla matrice di rotazione R_1^0. Consideriamo un punto P qualsiasi di cui conosciamo:

- La posizione p^1 rispetto alla terna 1;
- La posizione o_1^0 dell'origine della terna 1 rispetto alla terna 0.



Per calcolare la posizione del punto P rispetto alla terna 0 possiamo sfruttare i dati a nostra disposizione, dove chiaramente p^1 dovrà essere moltiplicato per la matrice di rotazione R_1^0 visto che stiamo parlando di terna differenti: $p^0 = o_1^0 + R_1^0 p^1$

Se successivamente andiamo a derivare passiamo alle velocità e scopriamo che la velocità di un punto rispetto a terra (terna 0) possiamo scriverla nella seguente forma:

$$\dot{p}^0 = \dot{o}_1 + R_1^0 \dot{p}^1 + \dot{R}_1^0 p^1 = \dot{o}_1 + R_1^0 \dot{p}^1 + S(\omega_1^0) R_1^0 p^1 = \dot{o}_1 + R_1^0 \dot{p}^1 + \omega_1^0 \times r_1^0$$

In pratica, stiamo dicendo che la suddetta velocità è data da tre contributi:

- Velocità con cui il braccio si muove rispetto a terra;
- Contributo di traslazione;
- Contributo di rotazione.

Nota: da questo punto in poi se una posizione o matrice di rotazione non presenta nessun apice, automaticamente ci riferiamo alla terna 0. Inoltre, indicheremo con r il segmento che congiunge due terne.

Raffiniamo la scrittura della velocità adattandola al tipo di velocità di cui ci stiamo interessando, se è lineare oppure angolare:

A. Velocità Lineare: facendo riferimento all'immagine seguente, di fatto i ragionamenti sono analoghi a quelli appena fatti solo che rivisti in modo più generico sulle terne del manipolatore. La posizione dell'origine della terna i rispetto alla terna 0 è:

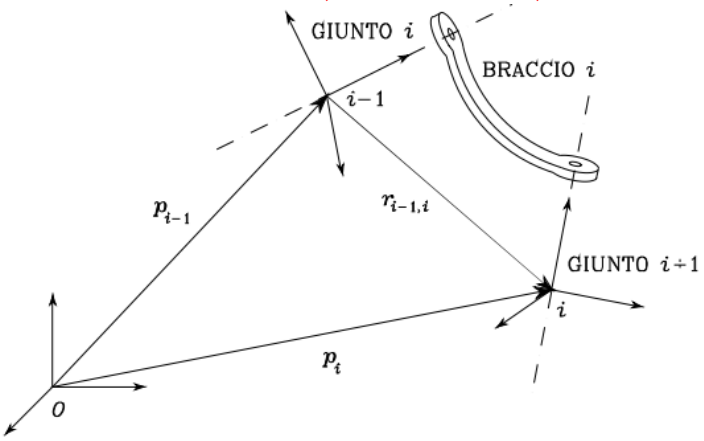
$$p_i = p_{i-1} + R_{i-1} r_{i-1,i}^{i-1}$$

Dove:

- p_{i-1} = posizione della terna $i-1$ rispetto alla terna 0;
- $R_{i-1} r_{i-1,i}^{i-1}$ = posizione relative delle terne $i-1$ e i riportata rispetto alla terna 0.

Otteniamo la velocità derivando rispetto al tempo:

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{R}_{i-1} \dot{\boldsymbol{r}}_{i-1,i}^{i-1} + \boldsymbol{\omega}_{i-1} \times \boldsymbol{R}_{i-1} \boldsymbol{r}_{i-1,i}^{i-1}$$
$$= \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{v}_{i-1,i} + \boldsymbol{\omega}_{i-1} \times \boldsymbol{r}_{i-1,i}$$



B. Velocità Angolare: per quanto riguarda la velocità angolare, se conosciamo la velocità angolare della terna $i-1$ e la velocità relativa tra quest'ultima e la terna i, visto che stiamo parlando di vettori liberi essa sarà semplicemente la somma:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \boldsymbol{R}_{i-1} \boldsymbol{\omega}_{i-1,i}^{i} = \boldsymbol{\omega}_{i-1} + \boldsymbol{\omega}_{i-1,i}$$

Ciò si può dimostrare partentendo dalla scrittura della matrice di rotazione $R_i = R_{i-1} R_i^{i-1}$ (che si ottiene semplicemente dalla convenzione di D-H), dopodichè deriviamo tale relazione:

$$\begin{aligned}
\dot{R}_i &= S(\omega_i) R_i = \dot{R}_{i-1} R_i^{i-1} + R_{i-1} \dot{R}_i^{i-1} \\
&= [S(\omega_{i-1}) R_{i-1}] R_i^{i-1} + R_{i-1} \left[S\left(\omega_{i-1,i}^{i-1}\right) R_i^{i-1} \right] \\
&= S(\omega_{i-1}) R_i + R_{i-1} S\left(\omega_{i-1,i}^{i-1}\right) I R_i^{i-1} \\
&= S(\omega_{i-1}) R_i + R_{i-1} S\left(\omega_{i-1,i}^{i-1}\right) R_{i-1}^T R_{i-1} R_i^{i-1} \\
&= S(\omega_{i-1}) R_i + R_{i-1} S\left(\omega_{i-1,i}^{i-1}\right) R_{i-1}^T I \\
&\downarrow \textbf{Proprietà di } \boldsymbol{S}\textbf{:} \quad \boldsymbol{R S(\omega) R^T = S(R\omega)} \downarrow \\
&= S(\omega_{i-1}) R_i + S\left(R_{i-1} \omega_{i-1,i}^{i-1}\right) R_i^{i-1}
\end{aligned}$$

Osservando solo i termini i verde, essi sono rispettivamente $\omega_i, \omega_{i-1}, \omega_{i-1,i}$ e dunque siamo arrivati alla relazione vista inizialmente.

Ora, due terne consecutive ruotano l'una rispetto all'altra perché è presente un giunto prismatico o rotoidale. Da ciò capiamo che dobbiamo *specializzare* la scrittura della velocità a seconda del tipo di giunto. Riprendiamo le due relazioni in rosso della velocità appena ricavate:

- $\boldsymbol{\dot{p}_i = \dot{p}_{i-1} + v_{i-1,i} + \omega_{i-1} \times r_{i-1,i}}$
- $\boldsymbol{\omega_i = \omega_{i-1} + \omega_{i-1,i}}$

E' da notare che nella velocità lineare, se il braccio trasla a variare è il termine $v_{i-1,i}$, mentre nella velocità angolare se il braccio ruota a variare è la velocità mutua $\omega_{i-1,i}$. Allora per un:

A. Giunto Prismatico: se il giunto è prismatico chiaramente la velocità mutua tra le due terne è nulla, mentre a variare è la velocità lineare $v_{i-1,i}$, che sarebbe la velocità $\dot{d}_\iota$ tra le origini della terna i e la terna $i-1$ lungo la direzione dell'asse z_{i-1}. La $\dot{d}_\iota$ è nota in quanto è la derivata del parametro d_i definito nella convenzione di D-H, il quale ricordiamo essere la distanza tra le origine la terna i e $i-1$ lungo la direzione dell'asse z_{i-1}. Riassumendo:

- ✓ $\omega_{i-1,i} = 0$
- ✓ $v_{i-1,i} = \dot{d}_\iota z_{i-1}$
- ✓ $\omega_i = \omega_{i-1}$
- ✓ $\dot{p}_i = \dot{p}_{i-1} + v_{i-1,i} + \omega_{i-1} \times r_{i-1,i} = \dot{p}_{i-1} + \dot{d}_\iota z_{i-1} + \omega_{i-1} \times r_{i-1,i}$

Possiamo perciò affermare che:

- **Velocità Lineare**: $\dot{q}_i J_{Pi} = v_{i-1,i} = \dot{d}_i z_{i-1} \quad \rightarrow \quad J_{Pi} = z_{i-1}$
- **Velocità Angolare**: $\dot{q}_i J_{Oi} = \omega_{i-1,i} = 0 \quad \rightarrow \quad J_{Oi} = 0$

B. Giunto Rotoidale: anzitutto abbiamo una rotazione di $\dot{\theta}_i$ intorno a z_{i-1}, quindi $\omega_{i-1,i} = \dot{\theta}_i z_{i-1}$. Inoltre, avremo un termine di trasporto $v_{i-1,i} = \omega_{i-1,i} \times r_{i-1,i}$. Riassumendo:

- ✓ $\omega_{i-1,i} = \dot{\theta}_i z_{i-1}$
- ✓ $v_{i-1,i} = \omega_{i-1,i} \times r_{i-1,i}$
- ✓ $\omega_i = \omega_{i-1} + \dot{\theta}_i z_{i-1}$
- ✓ $\dot{p}_i = \dot{p}_{i-1} + \omega_i \times r_{i-1,i}$

Possiamo perciò affermare che:

- **Velocità Lineare**:

$$\dot{q}_i J_{Pi} = \omega_{i-1,i} \times r_{i-1,n} = \dot{\theta}_\iota z_{i-1} \times (p_e - p_{i-1}) \quad \rightarrow \quad J_{Pi} = z_{i-1} \times (p_e - p_{i-1})$$

- **Velocità Angolare**: $\dot{q}_i J_{Oi} = \omega_{i-1,i} = \dot{\theta}_i z_{i-1} \quad \rightarrow \quad J_{Oi} = z_{i-1}$

Siamo così giunti al risultato che volevamo dimostrare:

$$\begin{bmatrix} \boldsymbol{J_{Pi}} \\ \boldsymbol{J_{Oi}} \end{bmatrix} = \begin{cases} \begin{bmatrix} \boldsymbol{z_{i-1}} \\ \boldsymbol{0} \end{bmatrix} & \textbf{\textit{se i} è \textit{un giunto prismatico}} \\ \begin{bmatrix} \boldsymbol{z_{i-1} \times (p_e - p_{i-1})} \\ \boldsymbol{z_{i-1}} \end{bmatrix} & \textbf{\textit{se i} è \textit{un giunto rotoidale}} \end{cases}$$

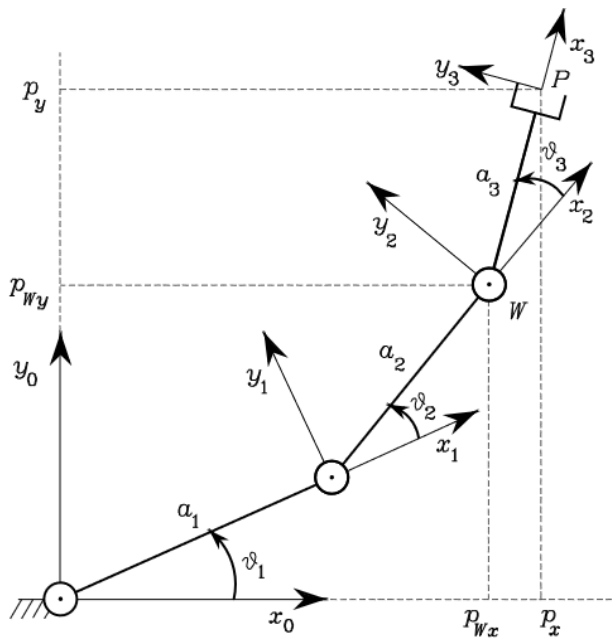
Per concludere, osserviamo che fino ad ora abbiamo parlato dello Jacobiano nella terna 0, tuttavia grazie alla matrice di rotazione è possibile rappresentarlo anche in una terna generica t. Infatti, per otternere J^t basta premoltiplicare J per una matrice a blocchi dove sulla diagonale troviamo le matrici di rotazione dalla terna t alla terna corrente (ecco perché non ci sono i pedici). In questo modo possiamo anche ricavare le velocità (lineari e angolari) della terna t.

$$J^t = \begin{bmatrix} R^t & 0 \\ 0 & R^t \end{bmatrix} J$$

$$\begin{bmatrix} \dot{p}^t \\ \omega^t \end{bmatrix} = J^t \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} R^t & 0 \\ 0 & R^t \end{bmatrix} J \dot{q}$$

Partiamo dallo studio dello Jacobiano geometrico per evidenziare il fatto che, nel momento in cui consideriamo un sistema meccanico in movimento, dove se trasla parleremo di velocità lineare mentre se ruota di velocità angolare, la prima è la derivata temporale del vettore *applicato* di posizione che definisce la posizone del corpo rigido la seconda non solo è un vettore *libero* ma non è la derivata degli angoli di eulero (nonostante presenti tre componenti). Dunque, se dobbiamo studiare le velocità di questo sistema con la cinemati differenziale dobbiamo far riferimento a grandezze che hanno un significato fisico.

Esempio: consideriamo il manipolatore a tre bracci di fianco.



Poiché sono presenti solo giunti rotoidali, le colonne dello Jacobiano saranno scritte nella forma:

$$J_{O_i} = \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix}$$

Allora lo Jacobiano avrà la seguente espressione, di cui dovremo andare ad identificare tutti i parametri in esso contenuti.

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{p}_3 - \boldsymbol{p}_0) & \boldsymbol{z}_1 \times (\boldsymbol{p}_3 - \boldsymbol{p}_1) & \boldsymbol{z}_2 \times (\boldsymbol{p}_3 - \boldsymbol{p}_2) \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix}$$

$$\boldsymbol{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_2 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix}$$

$$\boldsymbol{p}_3 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix} \qquad \boldsymbol{z}_0 = \boldsymbol{z}_1 = \boldsymbol{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

In defintiva, lo Jacobiano avrà la seguente espressione:

$$\boldsymbol{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Osserviamo che lo Jacobiano geometrico in questo caso è una matrice $6x3$ (matrice alta), perciò al più esso potrà avere rango colonna pari tre. In questo caso è ancora più semplice il calcolo del rango perché abbiamo tre righe nulle, ovvero la riga relativa alla velocità lineare lungo l'asse z, le righe relative alla velocità angolare lungo x ed y. Dunque, le uniche componenti di velocità rilevanti per questa geometria sono le velocità lineari lungo x ed y insieme alla velocità angolare intorno a z.

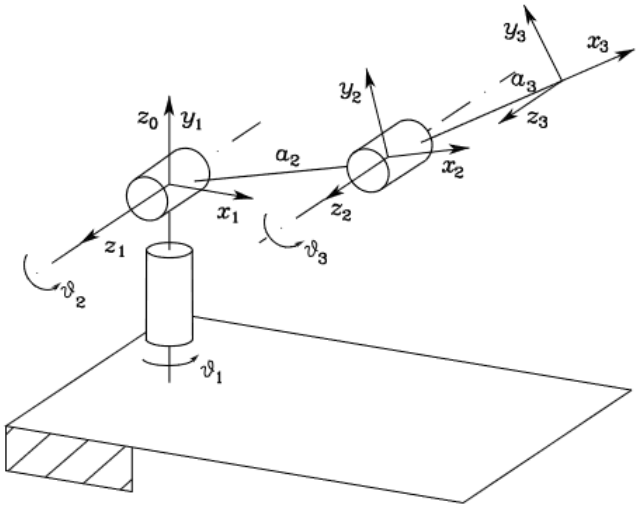
E' notare che, visto la semplicità dell'esempio, avremmo potuto anche ragionare in maniera completamente differente. Anzitutto potevamo definire il vettore x della cinematica diretta costituito da:

- Le componenti di posizione dell'organo terminale che di fatto corrispondono alle prime due righe del vettore p_3 che sono rispettivamente p_x e p_y;
- La terza componente era l'orientamento ϕ che, in questo esempio, è semplicemente la somma delle tre variabili di giunto: $\phi = \theta_1 + \theta_2 + \theta_3$

Ricordando l'equazione cinematica diretta: $x = k(q)$

Per calcolare lo Jacobiano di questa funzione vuol dire calcolare $\partial k / \partial q$, il risultato sarebbero state le tre righe diverse da zero dello Jacobiano calcolato precedentemente. Abbiamo così scoperto che, la selezione delle righe diverse da zero dello Jacobiano geometrico porta alla definizione di quello che chiameremo Jacobiano analitico, il quale caratterizza le componenti di velocità significative per questa geometria. Come specificato inizialmente, questo approccio è adottabile in questo esempio mentre non lo è per qualsiasi manipolatore. Questo perché di fatto noi non scriviamo mai una $\phi(q)$ e per ricavarla dovremmo calcolare la $R(q)$ che ci da l'orientamento, dopodichè mediante formule invese estrapolare gli angoli di Eulero. Tali formule però sarebbe strutturate come $if - then - else$ a causa della loro differente formulazione in funzione dell'angolo θ. Pensare di fare la derivata in questo scenario è particolarmente complesso. Inoltre, problema ancora maggiore è che derivando ϕ per ottenere $\dot{\phi}$, quest'ultima non ha il significato di velocità angolare necessaria per la cinematica differenziale.

Esempio: supponiamo di avere un manipolatore antropomorfo.



Poiché sono presenti solo giunti rotoidali, le colonne dello Jacobiano saranno scritte nella forma:

$$J_{O_i} = \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix}$$

Allora lo Jacobiano avrà la seguente espressione, di cui dovremo andare ad identificare tutti i parametri in esso contenuti.

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{p}_3 - \boldsymbol{p}_0) & \boldsymbol{z}_1 \times (\boldsymbol{p}_3 - \boldsymbol{p}_1) & \boldsymbol{z}_2 \times (\boldsymbol{p}_3 - \boldsymbol{p}_2) \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix}$$

$$\boldsymbol{p}_0 = \boldsymbol{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_2 = \begin{bmatrix} a_2 c_1 c_2 \\ a_2 s_1 c_2 \\ a_2 s_2 \end{bmatrix} \quad \boldsymbol{p}_3 = \begin{bmatrix} c_1(a_2 c_2 + a_3 c_{23}) \\ s_1(a_2 c_2 + a_3 c_{23}) \\ a_2 s_2 + a_3 s_{23} \end{bmatrix}$$

$$\boldsymbol{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \boldsymbol{z}_1 = \boldsymbol{z}_2 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

In defintiva, lo Jacobiano avrà la seguente espressione:

$$\boldsymbol{J} = \begin{bmatrix} -s_1(a_2 c_2 + a_3 c_{23}) & -c_1(a_2 s_2 + a_3 s_{23}) & -a_3 c_1 s_{23} \\ c_1(a_2 c_2 + a_3 c_{23}) & -s_1(a_2 s_2 + a_3 s_{23}) & -a_3 s_1 s_{23} \\ 0 & a_2 c_2 + a_3 c_{23} & a_3 c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}$$

2.2 Singolarità Cinematiche

2.2.1 Caratterizzazione delle Singolarità

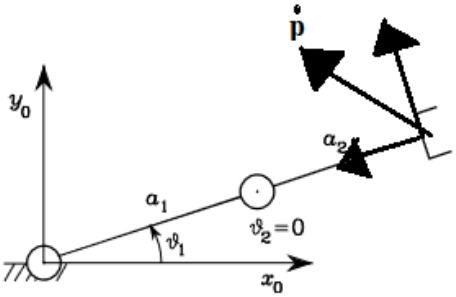
Una **Singolarità Cinematica** *è un punto (o un insieme di punti) all'interno dello spazio di lavoro del robot in cui la matrice Jacobiana di quest'ultimo, che ricordiamo lega la velocità dei giunti alla velocità dell'end-effector, perde rango*. Le singolarità si possono classificare in due categorie:

A. Singolarità ai confini dello spazio di lavoro raggiundibile;
B. Singolarità all'interno dello spazio di lavoro raggiundibile.

Come si può intuire la prima categoria non è particolarmente problematica in quanto, ai confini dello spazio di lavoro raggiungibile, il robot già di per sé non ha una buona manipolazione e quindi già a priori sono zone da evitare. Perciò, la seconda categoria di singolarità è quella certamente più problematica in quanto riguardano quella parte dello spazio di lavoro in cui il manipolatore è più destro. In definitiva, le principali problematiche dovute alla singolarità sono:

- **Perdita di mobilità**: l'incapacità del manipolatore di muovere, indipendentemente dal movimento dei giunti, l'end-effector in una direzione. Per capire questo aspetto consideriamo il manipolatore planare a due bracci in figura con la relativa matrice Jacobiana.

$$J = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \end{bmatrix}$$



Ora per avere una singolarità, dalla definzione capiamo che tale matrice di rango 2 deve passare a rango 1 e ciò avviene quando il determinante di J è pari a zero. Perciò, calcoliamo il determinante di J: $\det(J) = a_1 a_2 s_2$

Tale determinante si annulla quando $s_2 = 0$ e ciò avviene per $\theta_2 = 0$ oppure per $\theta_2 = \pi$, ovvero quando il secondo braccio è allineanto esternamente o internamente al primo braccio. Allora $\forall\ \theta_1$ il manipolatore planare a due bracci ha una singolarità in $\theta_2 = 0$ e $\theta_2 = \pi$. Soffermiamoci sulla singolarità $\theta_2 = 0$, quindi abbiamo il bracci completamente steso. Ora se noi vogliamo imprimere all'organo terminale una certa $\dot{p}$ (essendo a due bracci posso solo lavorare sulle posizioni p e velocità lineari $\dot{p}$), quest'ultima potremo scomporla in una componente radiale $\dot{p}_r$ e una tangenziale $\dot{p}_t$. Nello scenario della singolarità che stiamo studiando è percepibile solo la componente tangenziale e quindi se proviamo a far variare θ_2 al fine di uscire dalla configurazione singolare, il robot continuerà a muoversi solo tenendo conto di tale componente e di fatto non riuscirà ad uscire da tale configurazione. In effetti diremo che se in una qualunque configurazione il range $\mathcal{R}(J)$ è l'intero spazio dei giunti, nella configurazioni singolari la componente radiale finisce nel nullo $\mathcal{N}(J)$. Tuttavia, c'è da dire che queste singolarità sono di confine allo spazio di lavoro e come detto precendentemente non sono un grande problema.

- **Infinite soluzioni al problema cinematico inverso**: se la J scende di rango e il suo determinante è nullo allora non risulterà essere invertibile. Da ciò ne deriva che non potremo calcolare la sola soluzione della cinematica inversa, cioè quella che a partire dalla posa dell'end-effector ci fa ottenere la posa dei giunti.

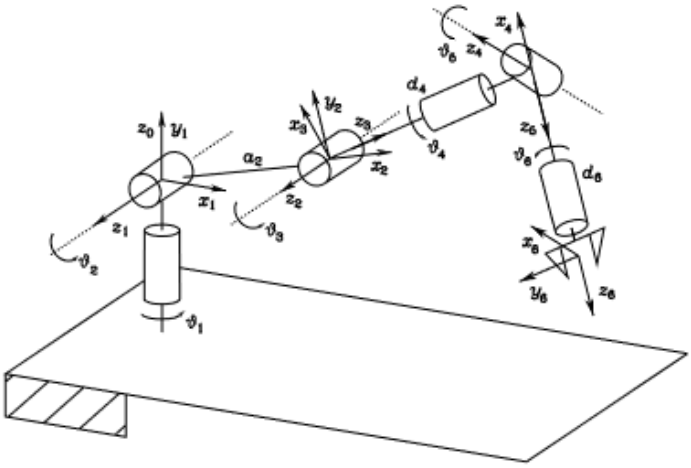
- **Velocità elevato nello spazio dei giunti**: problematiche non sono solo le coonfigurazioni in corrispondenza delle singolarità ma anche il trovarsi nel loro intorno. Ricordiamo che J^{-1} è calcolata come prodotto tra l'inverso del determinante di J per la matrice aggiunta. Se il determinante di J è molto piccolo, visto che stiamo in prossimità della singolarità, il termine $1/\det(J)$ risulterà essere molto grande. Ciò si traduce, nell'intorno della singolarità, ad avere delle velocità molto elevate che possono usurare eccessivamente i giunti.

2.2.2 Decomposizione delle Singolarità

La **Decomposizione delle Singolarità** *è una tecnica in cui il calcolo delle singolarità di una struttura cinematica viene scomposta in due sotto-problemi:*

- **Calcolo delle Singolarità del Braccio**;
- **Calcolo delle Singolarità di Polso**.

Si tratta di una tecnica che viene impiegata per strutture cinematiche complesse in cui il calcolo del determinante dello Jacobiano non risulta essere agevole. Andiamo a studiare questa tecnica considerando di avere un manipolatore con polso sferico come in figura. Inoltre, supponiamo che la sua matrice Jacobiana può essere scomposta a blocchi:



$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

Dove:

✓ $J_P = [J_{11} \quad J_{12}]$, mentre J_{11} e J_{12} sono i contributi dei diversi giunti alla velocità lineare;

✓ $J_O = [J_{21} \quad J_{22}]$, mentre J_{21} e J_{22} sono i contributi dei diversi giunti alla velocità angolare.

Se il polso è sferico, allora gli ultimi giunti sono tutti rotoidali, quindi le matrici J_{12} e J_{22} saranno le seguenti:

$$J_{12} = [z_3 \times (p_e - p_3) \quad z_4 \times (p_e - p_4) \quad z_5 \times (p_e - p_5)]$$
$$J_{22} = [z_3 \quad z_4 \quad z_5]$$

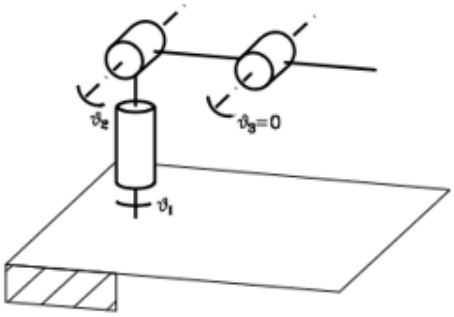
Tuttavia, se $p_e = p_W$ o comunque giace sulla medesima direzione, allora $p_W - p_i$ sono paralleli a z_i con $i = 3,4,5$. Come sappiamo il prodotto vettoriale è nullo se i vettori in questione sono paralleli, proprio come in questo caso da cui possiamo affermare che la matrice J_{12} presenta tutti elementi nulli: $J_{12} = [0 \quad 0 \quad 0]$

Lo Jacobiano perciò avrà una struttura come matrice triangolare bassa, il cui determinante è dato dal prodotto dei determinanti relativi alle matrici presenti sulla diagonale principale, quindi J_{11} e J_{22}: $\det(J) = \det(J_{11})\det(J_{22})$

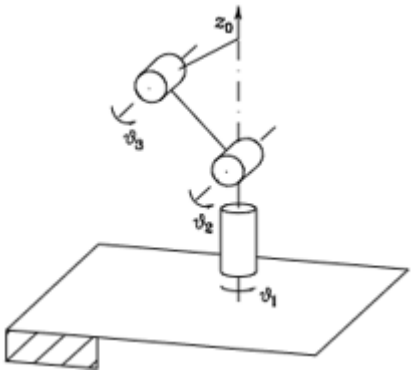
La matrice J_{11} la definiamo andando ad effettuare i seguenti passaggi:

- **Calcolo delle Singolarità del Braccio**: $\det(J_{11}) = -a_2 a_3 s_3 (a_2 c_2 + a_3 c_{23})$
  Osservando tale determinante ci accorgiamo essere presenti due singolarità:

- **Singolarità di Gomito**: avviene quando gli ultimi due bracci sono *allineati*, quindi distesi oppure ritratti. Nel nostro caso, il determinante si annulla quando $s_3 = 0$ e ciò avviene per $\theta_3 = 0, \pi$.



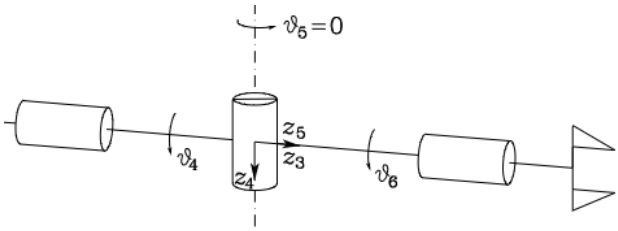
- **Singolarità di Spalla**: avviene quando l'origine della terna dell'end-effector giace sulla z_0 della terna base. Nel nostro caso, tale singolarità è definita dall'annullamento del termine fra parentesi tonde del determinante che stiamo studiante. Poiché in essa sono presenti funzioni trascendenti coseno e seno, non è banale capire in quali condizioni essa si annulla. A tal proposito, riprendiamo la matrice di trasformazione T_3^0 ricavata per il manipolatore antropomorfo dalla cinematica diretta:



$$\boldsymbol{T}_3^0 = \boldsymbol{A}_1^0\boldsymbol{A}_2^1\boldsymbol{A}_3^2 = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & c_1(a_2c_2+a_3c_{23}) \\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1(a_2c_2+a_3c_{23}) \\ s_{23} & c_{23} & 0 & a_2s_2+a_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Osserviamo che la parentesi che stiamo studiando compare nei primi due termini dell'ultima colonna, i quali sono rispettivamente p_x e p_y. Se la parentesi $(a_2c_2 + a_3c_{23}) = 0$, allora $p_x = p_y = 0$. Perciò, l'origine della terna solidale all'end-effector giace sull'asse z_0 della terna di base. Ebbene, tutte le posture che giaccino lungo z_0 sono posture singolari. Infatti, in tali punti la matrice J_{11} presenta la prima colonna identicamente nulla con un decremento di rango da 3 a 2. In definitiva, per qualunque $\dot{\theta}_1$, la velocità dell'organo terminale non cambia. Infine, è importante notare che, a differenza delle singolarità del polso che vedremo a breve, le singolarità del braccio sono ben identificate nello spazio operativo, e quindi possono essere opportunamente evitate nella fase di pianificazione della traiettoria dell'end-effector.

- **Calcolo delle Singolarità di Polso Sferico**: il polso sferico è analogo (non uguale) ad una terna di angoli di Eulero ZYZ, la quale soffre di una singolarità di rappresentazione quando il secondo angolo $\theta = 0, \pi$. Allora il polso sferico presentare una singolarità di polso per $\theta_5 = 0$ e $\theta_5 = \pi$ e quindi la rotazione degli altri due angoli non porterà a rotazioni dell'organo terminale. Tra l'altro, osservando come abbiamo scritto la matrice J_{22}:



$$J_{22} = [z_3 \quad z_4 \quad z_5]$$

I valori $\theta_5 = 0, \pi$ corrispondono ad una condizione di allineamento tra z_3 e z_5, ovvero di dipendenza tra le colonne dello Jacobiano da cui ne consegue una perdita di rango. Questa singolarità è piuttosto pericolosa in quando può essere riscontrata all'interno dello spazio di lavoro e quindi bisogna prestare attenzione alla programmazione del movimento dell'organo terminale.

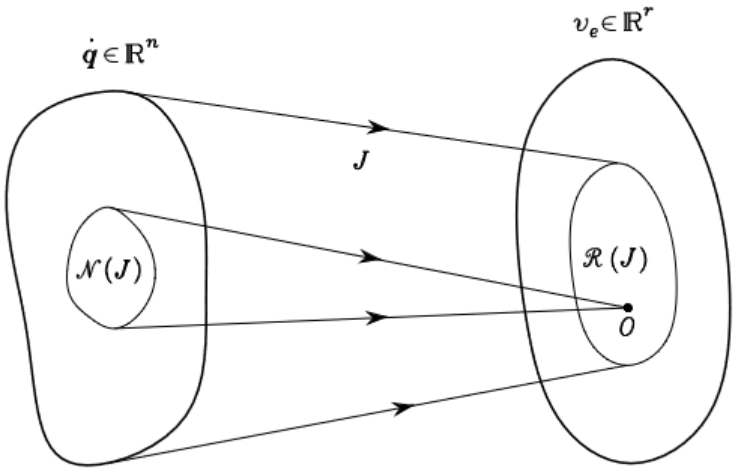
2.3 Analisi della Ridondanza

In realtà, oltre alla presenza delle singolarità cinematiche, a dare problemi nella cinematica differenziale è anche il concetto di ridonanza. Ricordiamo che un manipolatore è ridondante quando esso è dotato di un numero di gradi di libertà superiori a quelli necessari per realizzare un certo task. Anticipiamo che questo problema verrà affrontato in maniera grafica ragionando sugli insiemi. Perciò, anzitutto riprendiamo l'equazione cinematica diretta: $v_e = J(q)\dot{q}$
Della quale è doveroso precisare due fondamentali forzature:

- **Forzatura 1**: quello che facciamo è "congelare" la postura q del robot al fine di studiare la sua attitudide, dalla posizione corrente, a muoversi in un'altra posizione.
- **Forzatura 2**: il vettore v_e presenta al suo interno grandezze con unità di misura differenti (m/s per le velocità lineari e rad/s per le velocità angolari). Perciò dovremmo studiare la relazione $v_e = J(q)\dot{q}$ separatamente per le velocità lineari e angolari.

In ogni caso però, tale trasformazione che ci porta dallo spazio di partenza delle velocità dei giunti, tipicamente in $\mathcal{R}^n$, allo spazio di arrivo delle velocità dell'organo terminale, tipicamente in $\mathcal{R}^r$. L'operatore che ci porta dallo spazio di partenza a quello di arrivo è la matrice Jacobiana (nel disegno non è scritto $J(q)$ perché abbiamo congelato una particolare postura del robot). Parlando di trasformazioni lineari, ricordiamo che esse sono caratterizzati da due fondamentali insiemi:

- **Range $\mathcal{R}$**: lo spazio di partenza delle velocità dei giunti $\dot{q}$ è contenuto in R^n, tuttavia, a causa della presenza dei fine corsa su questi ultimi non sarà possibile esprimere tutte le v_e possibili. Motivo per cui lo spazio di arrivo corrisponde al range dello Jacobiano $\mathcal{R}(J)$, ovvero il sottospazio di R^r delle velocità v_e che possono essere generate dalle velocità $\dot{q}$ nella data postura del manipolatore: $\mathcal{R}(J) = \{\dot{q} \ : \ J(q)\dot{q} = v_e\}$
- **Nullo $\mathcal{N}$**: nello spazio di partenza, non tutte le $\dot{q}$ generano moto dell'organo terminale. Il nullo dello Jacobiano $\mathcal{N}(J)$ è l'insieme di $\dot{q}$, diverse dalla soluzione banale nulla, che presentano tale proprietà: $\mathcal{N}(J) = \{\dot{q}^* \neq 0 : J(q^*)\dot{q}^* = 0\}$
  E' da notare che queste *non sono le singolarità del manipolatore* ma semplicemente delle configurazioni che non generano il moto dell'organo terminale.



Se lo Jacobiano è di rango pieno, allora: $\dim(\mathcal{R}(J)) = r \qquad \dim(\mathcal{N}(J)) = n - r$
Inoltre, se lo Jacobiano perde di rango a causa di una singolarità, allora il suo range perde di rango mentre quello del suo nullo aumenta a causa della proprietà:

$$\dim(\mathcal{R}(J)) + \dim(\mathcal{N}(J)) = n$$

In definitiva, se il manipolatore è ridondante, allora $\mathcal{N}(J) \neq \emptyset$ e saranno quindi necessarie delle tecniche per gestire i gradi di libertà ridondanti, i quali saranno in numero pari a $n - r$. Un approccio impiegabile potrebbe essere il seguente. Immaginiamo di avere una soluzione $\dot{q}^*$ dell'equazione cinematica diretta $v_e = J(q)\dot{q}$. Da questa soluzione $\dot{q}^*$ possiamo definire una nuova soluzione $\dot{q}$ ottenuta sommando due termini: $\dot{q} = \dot{q}^* + P\dot{q}_0$

I quali sono:

- La soluzione $\dot{q}^*$ ricavata in precedenza;
- Un termine costituito una matrice $P \in R^{n \times n}$ tale $\mathcal{R}(P) = \mathcal{N}(J)$ il cui compito è "filtrare" una $\dot{q}_0$, dove quest'ultima va a caratterizzare i **Moti Interni alla Struttura**. In altre parole, essa non da alcun contributo alla posa dell'end-effector ma solo alla configurazione del braccio.

Grazie a questo approccio, l'effetto di $\dot{q}_0$ non porta ad una cambiamento di posa dell'organo terminale ma bensì ad una *riconfigurazione* del manipolatore in posizioni più abili per l'esecuzione di un determinato compito.

2.4 Inversione Cinematica Differenziale

Caratterizzazione dell'Inversione Cinematica Differenziale

L'**Inversione Cinematica Differenziale** *consiste nel ricavare, a partire dalla velocità v_e dell'organo terminale con le relative condizioni iniziali, la velocità $\dot{q}$ dei giunti*. Questo studio lo affronteremo studiando diversi scenari, ciascuno con la propria tecnica risolutiva per calcolare $\dot{q}$. Tuttavia, essi presenteranno in comune lo step finale in cui dovremo integrare la $\dot{q}$ al fine di determinare $q(t)$. In realtà, poiché nella pratica lavoriamo su sistemi tempo discreto, sarebbe più corretto parlare di integrazione numerica, per esempio con la regola di Eulero (indicata a destra).

$$q(t) = \int_0^t \dot{q}(\xi)d\xi + q(0) \quad \rightarrow \quad q(t_{k+1}) = q(t_k) + \dot{q}(t_k)\Delta t$$

Tuttavia, è da osservare un aspetto interessante, ovvero che un sistema meccanico come un manipolatore presenza frequenze dell'ordine di qualche decina di Hertz, mentre le tipiche frequenze di campionamento sono dell'ordine del KHz. Ciò equivale ad avere periodi di campionamento Δt piuttosto ampi, quindi di fatto dal confronto degli ordini di grandezza è come se stiamo eseguendo un controllo in tempo continuo.

Prima di affrontare i casi più complessi e problematici dell'inversione cinematica differenziale diamo un accenno al caso più semplice, ovvero quando abbiamo un manipolatore non ridondante ($n = r = \dim(spazio\ operativo)$) e non singolare. In tal caso, lo Jacobiano è quadrato e di rango pieno per cui è sempre invertibile. Allora l'inversione cinematica può essere attuata mediante l'impiego dell'inverso dello Jacobiano: $\boldsymbol{\dot{q} = J^{-1}(q)v_e}$

A questo punto affrontiamo i casi più problematici in cui lo Jacobiano non è invertibile, ovvero nel caso in cui il manipolatore è ridondante e successivamente quando è in una configurazione singolare. E' da specificare che non è casuale l'ordine di trattazione di questi casi, infatti le singolarità possono riguardare manipolatori ridondanti e non, per cui conviene prima trattare il caso ridondanza senza singolarità e successivamente introdurre quest'ultima. Andiamo ora a studiare questi due casi:

A. **Ridondanza**: lo Jacobiano non è quadratro;
B. **Singolarità**: lo Jacobiano perde di rango.

Inversione per Manipolatori Ridondanti

La presenza della ridondanza cinematica, quindi $r < n$, comporta che lo Jacobiano è rettangolare basso (più colonne che righe). Ciò implica che la relazione $v_e = J\dot{q}$ presenta infinite soluzioni. E' notare che non abbiamo scritto $J(q)$ perché altrimenti l'equazione sarebbe non lineare. Allora possiamo pensare di procedere impostando un problema di ottimizzazione vincolato. In particolare, data v_e e J per una data configurazione, vogliamo trovare una $\dot{q}$ soluzione del seguente problema di minimizzazione vincolato, dove $W \in R^{n\times n}$ è una matrice di peso simmetrica definita positiva. Trattandosi di un problema vincolato utilizziamo il **metodo dei moltiplicatori di Lagrange** in cui definiamo la nuova funzione obiettivo in cui il vettore dei moltiplicatori $\lambda \in R^{r\times 1}$ tiene conto del vincolo:

$$\begin{aligned} &\boldsymbol{f.o.}\ \min g(\dot{q}) = \frac{1}{2}\dot{q}^T W \dot{q} \\ &\boldsymbol{Vincolo}:\quad v_e = J\dot{q} \end{aligned} \quad \rightarrow \quad \boldsymbol{f.o.}\ g(\dot{q},\lambda) = \frac{1}{2}\dot{q}^T W \dot{q} + \lambda^T(v_e - J\dot{q})$$

Per un problema di ottimalità esistono delle *condizioni necessarie ma non sufficienti* per minimizzare o massimizzare un funzionale di costo, le quali ci dicono che i gradienti di quest'ultimo rispetto a $\dot{q}$ e λ devono essere nulli (essendo i gradienti dei vettori riga bisogna trasporli):

$$\left(\frac{\partial g}{\partial \dot{q}}\right)^T = W\dot{q} - J^T\lambda = 0 \qquad \left(\frac{\partial g}{\partial \lambda}\right)^T = v_e - J\dot{q} = 0$$

Dato che non abbiamo alcuna certezza sull'invertivilità di J, ci conviene ricavare $\dot{q}$ dall'equazione di sinistra visto che $W = W^T > 0$: $\quad \dot{q} = W^{-1}J^T\lambda \quad \rightarrow \quad v_e = JW^{-1}J^T\lambda$
Ora se $J \in R^{r\times n}$, $W \in R^{n\times n}$ e $J^T \in R^{n\times r}$ allora il prodotto di queste tre matrici appartiene a $R^{r\times r}$. Inoltre, se J ha rango r la matrice $JW^{-1}J^T$ avrà rango pieno (visto che appartiene a $R^{r\times r}$) e quindi quest'ultima risulterà essere invertivile. Il vettore $\lambda = (JW^{-1}J^T)^{-1}v_e$ e dunque possiamo concludere che la soluzione ottima del problema è:

$$\dot{q} = W^{-1}J^T(JW^{-1}J^T)^{-1}v_e \xrightarrow{Se\ scelgo\ W=I} \boldsymbol{\dot{q} = J^+v_e = J^T(JJ^T)^{-1}v_e}$$

Dove J^+ è la pseudo-inversa destra (di Moore-Perrous) di J. Sapendo che $J \in R^{r\times n}$, quindi $JJ^T \in J \in R^{r\times r}$, se moltiplichiamo una matrice $r \times r$ per $J^T \in R^{n\times r}$ otteniamo sempre una matrice pseudo-inversa $J^+ \in R^{n\times r}$.

Nell'analisi sulla ridondanza abbiamo visto ad una certa soluzione $\dot{q}^*$ possiamo sommare una $\dot{q}_0$, opportunamente filtrata dal **Proiettore *P* nello Spazio Nullo**, che va a caratterizzare i moti interni alla struttura e ci consente di sfruttare al meglio i gradi ridondanti: $\quad \dot{q} = \dot{q}^* + P\dot{q}_0$
Nel problema appena visto, se $v_e = 0$ allora $\dot{q} = 0$. Se però a questa $\dot{q}$ sommiamo un $\dot{q}_0$ e quest'ultimo non deve dare contributo al moto dell'organo terminale, dobbiamo fare in modo che permanga la condizione $v_e = 0$. Per fare ciò, matemticamente, dobbiamo portare $\dot{q}$ a $\dot{q}_0$ anziché a zero, ovvero dobbiamo minimizzare la norma del vettore $\dot{q} - \dot{q}_0$, la quale sarà di fatto la nostra nuova funzione obiettivo del seguente problema di minimizzazione vincolato:

$$g(\dot{q}) = \frac{1}{2}(\dot{q}^T - \dot{q}_0^T)(\dot{q} - \dot{q}_0) \qquad Vincolo:\ \ v_e = J\dot{q}$$

Analogamente a quanto fatto in precedenza, utilizziamo il metodo dei moltiplicatori di Lagrange:

$$g(\dot{q},\lambda) = \frac{1}{2}(\dot{q} - \dot{q}_0)^T(\dot{q} - \dot{q}_0) + \lambda^T(v_e - J\dot{q})$$

Dalla prima condizione necessaria circa il gradiente di g rispetto a $\dot{q}$, ricaviamo l'espressione di quest'ultimo. Successivamente sostituiamo $\dot{q}$ nel vincolo $v_e = J\dot{q}$ per ricavare l'espressione di λ:

$$\left(\frac{\partial g}{\partial \dot{q}}\right)^T = (\dot{q} - \dot{q}_0)^T - \lambda^T J = 0 \qquad \rightarrow \qquad \begin{aligned} v_e &= J\dot{q} \\ v_e &= JJ^T\lambda + \dot{q}_0 \\ \lambda &= (JJ^T)^{-1}(v_e - J\dot{q}_0) \end{aligned}$$

$$\dot{q} - \dot{q}_0 = J^T\lambda$$
$$\dot{q} = J^T\lambda + \dot{q}_0$$

Sostituendo λ in $\dot{q}$ troviamo la nuova soluzione: $\qquad \boldsymbol{\dot{q} = J^+v_e + (I - J^+J)\dot{q}_0}$
Dove:

- **Primo Termine**: è relativo all'obiettivo di avere norma minima per la velocità dei giunti;

- **Secondo Termine**: è la soluzione omogenea che soddisfa il vincolo addizionale specificato da $\dot{q}_0$.

L'aspetto più interessante è che il proiettore in $\mathcal{N}(J)$ è: $\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})$

Perciò ci siamo ritrovati esattamente con la formula di $\dot{q}$ ottenuta alla fine dell'analisi di ridondanza fatta in precedenza: $\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\boldsymbol{v}_{\boldsymbol{e}} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\boldsymbol{0}} = \dot{\boldsymbol{q}}^{*} + \boldsymbol{P}\dot{\boldsymbol{q}}_{\boldsymbol{0}}$

Arrivati a questo punto però, ci chiediamo come scegliere la $\dot{q}_0$. Possiamo pensare di sceglierla come definita di fianco, ovvero come prodotto tra un certo guadagno $k_0 > 0$ e il vettore gradiente di un indice $w(q)$ (dovrà essere trasposto perché il gradiente è un vettore riga ma per le velocità serve un vettore colonna). Il problema ora si sposta sulla scelta dell'indice $w(q)$ e abbiamo tre possibili strade:

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q}\right)^T$$

- **Misura di Manipolatibilità**: è una metrica che ci dice quando il determinante dello Jacobiano va a zero. Poiché però stiamo considerando il caso di un manipolatore ridondante, lo Jacobiano è rettangolare e quindi tale indice dovrà lavorare su JJ^T che sicuramente è una matrice quadrata: $\boldsymbol{w}(\boldsymbol{q}) = \sqrt{\mathbf{det}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q}))}$

La $\dot{q}_0$ scelta con questo indice massimizza la misura di manipolabilità, perché più w è elevato e più siamo lontani dalla singolarità. Chiaramente questa $\dot{q}_0$ non sarà quella che massimizza in assoluto $w(q)$ per due motivi:

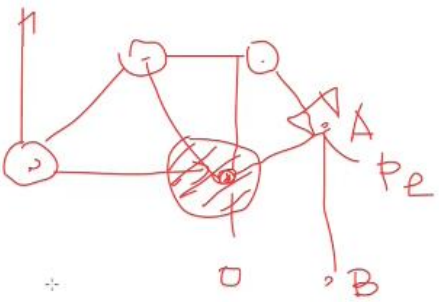
- ❖ Ha una validità locale postura per postura visto che la J cambia con q;
- ❖ E' un tentativo di muovere la struttura cercando di fargli assumere delle posture che, compatibilimente con il vincolo primario, presenti una manipolabilità maggiore.

- **Distanza dai Fine-Corsa dei Giunti**: nella realtà i giunti di un robot presentano sempre dei fine-corsa che vanno a definire l'intervallo di variazione $[q_{im}, q_{iM}]$ della posizione q_i del giunto $i-$esimo. L'indica $w(q)$ viene scelto secondo la seguente formula:

$$\boldsymbol{w}(\boldsymbol{q}) = -\frac{\mathbf{1}}{\mathbf{2}\boldsymbol{n}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}}\left(\frac{\boldsymbol{q}_{\boldsymbol{i}} - \bar{\boldsymbol{q}}_{\boldsymbol{i}}}{\boldsymbol{q}_{\boldsymbol{iM}} - \boldsymbol{q}_{\boldsymbol{im}}}\right)^{\mathbf{2}}$$

Il termine fra parantesi, cioè il rapporto tra la differenza della posizione q_i del giunto i e del punto medio della corsa del giunto $\bar{q}_i = (q_{iM} + q_{im})/2$ e la differenza dei fine-corsa che quindi stabilisce il range di variazione del giunto, è di fatto la distanza normalizzata dal centro dell'intervallo (la normalizzazione è ottenuto grazie al denominatore e quindi tutta la parentesi è in modulo < 1). Poiché vogliamo essere sicuri che la paretensi sia una quantità positiva mettiamo un quadrato e quindi per normalizzare dobbiamo moltiplicare il tutto per $1/2$. Questo ragionamento deve essere applicato a tutti i giunti e quindi dobbiamo passare alla sommatoria, dalla quale per normalizzare dobbiamo moltiplicare il tutto per $1/n$. Tutta questa quantità vogliamo minimizzarla e quindi è necessario un segno meno davanti.

A. **Distanza di un Ostacolo**: l'indice in questione lavorerà appunto sulla distanza da un ostacolo. Per capire in che modo ciò avviene, supponiamo di avere un manipolatore planare a tre bracci che deve solo posizionare l'organo terminale dal punto A al punto B. Dunque, tale manipolatore è ridondante per questo task e supponiamo inoltre che sia presente un ostacolo lungo il suo cammino di centro o. L'indice che verrà scelto è dato dalla seguente formula, la quale ci dice di calcolare tutti i segmenti di minima distanza tra o e i bracci del robot, dopodichè dobbiamo prendere quello minimo. Lo scopo è massimizzare questo indice (ecco perché ha segno positivo):



$$\boldsymbol{w(q) = \min_{p,o} \|p(q) - o\|}$$

Sulla scia di quest'ultimo approccio è doverso citare un caso molto particolare dell'esempio appena visto, ovvero quello per cui l'ostacolo si trova esattamente al centro della retta che congiunge direttamente A e B. Dunque il robot per poter assolvere al suo compito dovrà presentare una traiettoria come quella descritta nella figura di fianco. In questo scenario, abbiamo la presenza di due task mutuamente esclusivi fra loro da eseguire:

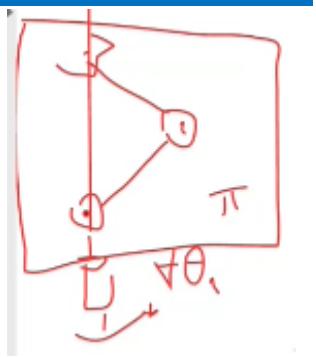
- ✓ **Task 1**: ovvero il compito di portare l'organo terminale da A a B, il quale possiamo esprimere con $x_1 = k_1(q)$ e da cui posso ricavare $\dot{x}_1 = J_1^{-1}\dot{q}$;
- ✓ **Task 2**: ovvero il compito di evitare l'ostacolo, il quale possiamo esprimere con $x_2 = k_2(q)$ e da cui posso ricavare $\dot{x}_2 = J_2^{-1}\dot{q}$.

Riprendendo la formulazione della soluzione dell'inversione cinematica vista in precedenza, possiamo definire una nuova versione della $\dot{q}$ che soddisfa entrambi i task. In particolare, il primo termine soddisfa il task 1, mentre il secondo soddisfa il task 2 ed inoltre il primo task ha priorità maggiore del secondo:

$$\dot{q} = J^+ v_e + (I - J^+ J)\dot{q}_0 = J_1^+ \dot{x}_1 + (I - J_1^+ J_1) J_2^+ \dot{x}_2$$

Da ciò capiamo che, affinchè il robot evito l'ostacolo, dobbiamo invertire i task. Ciò vuol dire che $J_1^+ \dot{x}_1$ finisce nel nullo mentre $\dot{q}_0 = J_2^+ \dot{x}_2$ diventa il task primario. Questa tecnica, definita di **task priority**, ci consente di aggirare l'ostacolo senza con lo svantaggio di non avere la garanzia che il task di portare l'end-effector da A a B venga svolto correttamente.

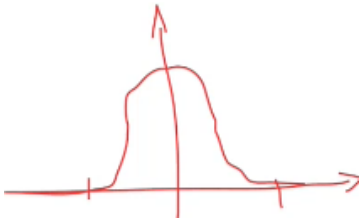
Inversione con Singolarità Cinematiche



Il problema delle soluzione in rosso trovate precedentemente è che richiedono la condizione di rango pieno per lo Jacobiana. La presenza di singolarità cinematiche può inficiare questa proprietà e quindi dobbiamo irrobustire la soluzione in vista di questa eventualità. Le singolarità ricordiamo essere problematiche non solo quando il manipolatore si trova in esse, per le quali abbiamo una perdita di mobilità, ma anche in loro prossimità a causa dei piccoli valori di det (J) che causano elevate velocità, comportando così delle velocità $\dot{q}$ elevate che possono usurare gravemente gli attuatori. Una robustezza della soluzione a fronte di singolarità cinematiche può essere ottenuta adottando l'**Inversa ai Minimi Quadrati Smorzata J^***, la quale in pratica è una pseudoinversa dotata di un termine di smorzamento k^2:

$$\boldsymbol{J^* = J^T(JJ^T + k^2 I)^{-1}}$$

L'aggiunta di una matrice diagonale su cui troviamo dei termini k^2, sommati alla JJ^T fa si che non siano presenti poi autovalori piccoli. Questa soluzione però sacrifica la corretta risoluzione della cinematica perché $JJ^* \neq I$, perciò tale soluzione presenta un errore. Il fattore di smorzamento k potrà essere caratterizzato da un andamento a campana, ovvero superata una certa soglia di posizione della singolarità, più ci avviciniamo a quest'ultima e più esso cresce. In pratica verrà attivato solo quando è necessario.



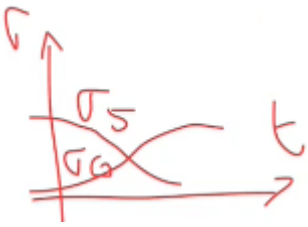
La J^* può essere ricavata analiticamente considerando questo funzionale di costo, dove il secondo termine è una forma quadratica analoga a quella scelta per il caso della ridondanza. L'unica differenza è che la matrice W è una matrice scalare con tutti i pesi uguali e quindi è presente il termine k^2. Applichiamo il metodo dei moltiplicatori di Lagrange, ottenendo così il seguente funzionale di costo.

$$g''(\dot{q}) = \frac{1}{2}k^2\dot{q}^T\dot{q} + \frac{1}{2}(v_e - J\dot{q})^T(v_e - J\dot{q})$$

Esso è costituito da due termini:

- **Primo Termine**: minimizza le velocità $\dot{q}$;
- **Secondo Termine**: minimizza lo scostamento tra $J\dot{q}$ e v_e. Questo secondo termine tiene conto che, nell'intorno della singolarità, il vincolo $v_e = J\dot{q}$ è mal posto ovvero non è totalmente soddisfatto.

Mettendo insieme questi due termini nello stesso funzionale metto in conto il vincolo $v_e = J\dot{q}$ non sia perfettamente soddisfatto. Sarà dunque il k a decretare se è più importante mantenere le velocità basse oppure soddisfare il vincolo $v_e = J\dot{q}$.



Analitacamente è possibile calcolare il fattore di smorzamento k, tuttavia un modo più semplice e veloce è impiegare la funzione Matlab SVD che calcola i valori singolari σ_i della matrice Jacobiana. Tuttavia, questa strada presenta una problematica che riguarda il minimo valor singolar e il valor singolare immediatamente superiore. Supponendo di avere una matrice $6 \times n$, quindi avremo σ_i con $i = 1 \dots, 6$, potremmo avere che ad un certo punto il minimo valor singolare σ_6 diventa maggiore di σ_5. Dall'incroncio in figura, di fatto, i due valori singolari di

invertono ($\sigma_5 \rightarrow \sigma_6$, $\sigma_6 \rightarrow \sigma_5$). A risoluzione di questa ambiguità di Matlab esiste un algoritmo intelligente che, dopo aver individuato il minimo valor singolare e il valor singoalre immediantamente superiore, sintonizza la k a seconda del tipo di singolarità (spalla, polso, ecc.).

Ultimo aspetto da osservare è che, come detto in precedenza, la soluzione con la pseudoinversa tendere a ridurre le velocità. Perciò il termine con la $w(q)$ potrebbe risultare essere inessenziale visto che ci tiene lontani dalle singolarità. In realtà, nell'intorno nelle singolarità è vero che la pseduinverso minimizza la velocità ma comuqneu rimangono valori probitivi. Risulta perciò necessario un termine aggiuntivo nella $\dot{q}$ che ci tenga lontano da esso. SI tratta di una metodologia con valenza locale che quindi non da alcuna garanzia che funzioni per ogni q, però risulta essere accetabile perché altrimenti dovremmo impiegare tecniche di ottimizzazione non lineare vincolati che non ci consentirebbero di ricavare delle soluzioni di tipo real-time. Tale proprietà risulta essere fondamentale per un robot che interagisce con l'ambiente circostante.

2.5 Jacobiano Analitico

Lo **Jacobiano Analitico $J_A(q)$** è la matrice che, come lo jacobiano gemetrico, lega le velocità dell'organo terminale alle velocità dei giunti però in questo caso le prime sono definite *nello spazio operativo*. Nella cinematica abbiamo ricavato l'equazione cinematica diretta, dalla quale attraverso la derivazione rispetto al tempo possiamo ricavare l'equazione cinematica differenziale diretta. In questa equazione la velocità dell'organo terminale, che ora indichiamo con $\dot{x}_e$, è specificata in termini di un numero minimo di parametri nello spazio operazionale che sono $\dot{p}_e$ e $\dot{\phi}_e$ (esattamente come nella cinematica diretta).

$$\dot{x}_e = \begin{bmatrix} \dot{p}_e \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} J_P(q) \\ J_\phi(q) \end{bmatrix} = J_A(q)\dot{q}$$

Questa riduzione del numero di parametri si ripercuote anche sulle dimensioni dello Jacobiano analitico. Infatti, lo Jacobiano geometrico è una matrice $6 \times n$, mentre lo Jacobiano analitico è una matrice $m \times n$. A questo punto l'obiettivo è caratterizzare completamento lo Jacobiano analitico in modo da poter scrivere l'equazione cinematica differenziale diretta. La scrittura della suddetta equazione è però possibile solo se possiamo scrivere p_e e ϕ_e in funzione di q (relazioni di fianco), dopodichè derivarle e infine scrivere l'equazione sopra citata. Tale considerazione è immediato nel caso della posizione:

$$\begin{cases} p_e = p_e(q) \\ \phi_e = \phi_e(q) \end{cases}$$

$$p_e = p_e(q) \quad \rightarrow \quad \dot{p}_e = \frac{\partial p_e}{\partial q}\dot{q} = J_P(q)\dot{q}$$

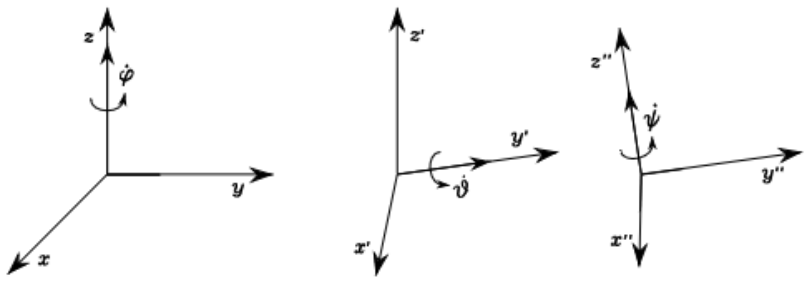
Contrariamente risulta essere più problematico $\phi_e = \phi_e(q)$, in quanto la relazione tra ϕ_e e q non sempre è semplice ed immediata. Perciò il seguente passaggio per ora non è realizzabile perché non sappiamo definire $J_\phi(q)$:

$$\phi_e = \phi_e(q) \quad \rightarrow \quad \dot{\phi}_e = \frac{\partial \phi_e}{\partial q}\dot{q} = J_\phi(q)\dot{q}$$

Potremmo pensare di calcolare $R_e(q)$, a partire da q, successivamente mediante delle formule di inversione estrarre la terna di angoli di Eulero $\phi_e(q)$. Quest'ultimo passaggio però ci porta ad un risultato $\phi_e(q)$ del tipo if-then-else, la cui scelta arbitraria non sarebbe precisa e soprattutto la $\dot{\phi}_e$ calcolata a partire da ϕ_e non corrisponde alla velocità angolare ω_e: $\dot{\phi}_e \neq \omega_e$
In realtà, le due grandezze sono legate tra loro mediante la matrice $T(\phi_e)$ che effettua una operazione di mapping: $\omega_e = T(\phi_e)\dot{\phi}_e$
Allora l'obiettivo a questo punto è caratterizzare questo legame:

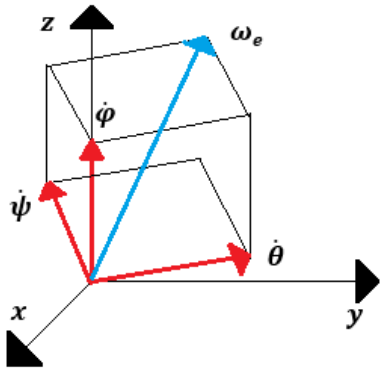
$$\dot{\phi} = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \rightarrow \quad \omega_e = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Prendiamo come riferimento per gli angoli di Eulero la terna ZYZ:



- Per effetto di $\dot{\varphi}$: $[\omega_x \quad \omega_y \quad \omega_z]^T = \dot{\varphi}[0 \quad 0 \quad 1]^T$
- Per effetto di $\dot{\theta}$: $[\omega_x \quad \omega_y \quad \omega_z]^T = \dot{\theta}[-s_\varphi \quad c_\varphi \quad 0]^T$
- Per effetto di $\dot{\psi}$: $[\omega_x \quad \omega_y \quad \omega_z]^T = \dot{\psi}[c_\varphi s_\varphi \quad s_\varphi s_\theta \quad c_\theta]^T$

Allora la situazione attuale nella terna corrente è mostrata in figura, dalla quale possiamo ricavare la relazione che lega ω_e e $\dot{\phi}_e$. In particolare la matrice T appena ricavata dipenderà *solo esclusivamente* dai primi due angoli in quanto $\dot{\psi}$ riguarda l'ultima rotazione che ancora dovremo svolgere



$$\omega_e = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\theta \\ 0 & c_\varphi & s_\varphi s_\theta \\ 1 & 0 & c_\theta \end{bmatrix} \dot{\phi}_e = T(\phi_e)\dot{\phi}_e$$

Per dimostrare la correttezza dei passaggi fatti, calcoliamo il determinante della matrice $T(\phi_e)$: $\det(T) = -s_\varphi^2 s_\theta - c_\varphi^2 s_\theta = -s_\theta$

I valori di θ per cui il determinante si annulla sono $\theta = 0, \pi$ e di fatto, come visto nel paragrafo circa gli angoli di Eulero, queste sono le singolarità di rappresentazione. In pratica, ci siamo ritrovati nella stessa situazione problematica trattata quando abbiamo parlato degli angoli di Eulero, infatti per questi valori di θ di fatto ci ritroviamo a ruotare due volte intorno allo stesso asse.

Ora abbiamo tutti gli strumenti per caratterizzare la relazione tra la velocità v_e dell'organo terminale e la velocità dei giunti $\dot{q}$:

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} \dot{p}_e \\ T(\phi_e)\dot{\phi}_e \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & T(\phi_e) \end{bmatrix} \begin{bmatrix} \dot{p}_e \\ \dot{\phi}_e \end{bmatrix} = T_A \dot{x}_e$$

La velocità dell'organo terminale v_e è pari a: $v_e = J(q)\dot{q}$

Uguagliando le due espressioni della v_e ricaviamo:

$$J(q)\dot{q} = T_A \dot{x}_e \quad \rightarrow \quad \dot{x}_e = T_A^{-1} J(q)\dot{q} = J_A(q)\dot{q}$$

La matrice $J_A(q)$ è detto Jacobiano Aanalitico, il quale oltre che soffrire delle stesse singolarità dello Jacobiano geometrico soffrirà anche delle singolarità di rappresentazione. Queste ultime dipendono essenzialmente dalla matrice T_A e in particolare dagli elementi sulla diagonale. Poiché I è certamente non singolare, a dare problema è solo la $T(\phi_e)$. E' da notare che una scelta differente di angoli di Eulero sarebbe stata ininfluente in quanto sei terne soffrono delle singolarità per $\theta = 0, \pi$ mentre le altre sei per $\theta = \pm\pi/2$.

In conclusione, se vogliamo lavorare con tutte grandezze che abbiano un significato fisico impieghiamo lo Jacobiano geometrico, se invece vogliamo lavorare con grandezze facenti parte lo spazio operativo impieghiamo lo Jacobiano analitico.

2.6 Algoritmi di Inversione Cinematica

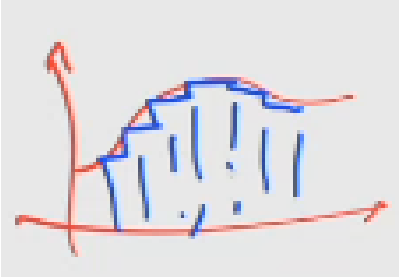
2.6.1 Caratteristiche degli Algoritmi CLIK

Un **Algoritmo CLIK** (**Close Loop Inverse Kinematics**) *è una particolare tipologia di algoritmi di risoluzione della cinematica inversa differenziale il cui obiettivo è far tendere a zero l'errore di spazio operativo* e*, il quale è dato dalla differenza tra la posa desiderata dell'organo terminale e quella effettiva:* $\boldsymbol{e = (x_d - x_e) \to 0}$

Data questa definizione ci potremmo chiedere perché utilizzare degli algoritmi che risolvono la cinematica differenziale inversa. Per capire ciò riprendiamo la soluzione alla cinematica inversa che si può ottenere dall'equazione cinematica differenziale:

$$v_e = J(q)\dot{q} \quad \to \quad \dot{q} = J^{-1}(q)v_e$$

Poiché lavoriamo su sistemi tempo discreto dobbiamo eseguire una integrazione numerica secondo per esempio la regola di Eulero di cui in figura è mostrato il risultato (andamento blu):



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

Questo vuol dire che le $\dot{q}$ calcolate non soddisfano la relazione $\dot{q} = J^{-1}(q)v_e$ nel tempo continuo. Risultato di ciò la posa dell'organo terminale effettiva non coincide con quella desiderata. La **Deriva della Soluzione** è quel quel fenomeno di cumulazione dell'errore dovute alle approssimazioni dell'integrazione numerica. Tali errori, anche se di pochi mm, possono causare per esempio il mancato afferraggio dell'oggetto che il robot deve manipolare. Capiamo dunque l'importanza di impiegare algoritmi (e quindi degli schemi) che risolvano la cinematica inversa con l'intento di rendere *asintoticamente stabile l'errore del sistema*, il quale è dato dalla differenza tra la posa desiderata e quella effettiva: $\boldsymbol{e = x_d - x_e \to 0}$

Impiegando la funzione cinematica diretta differenziale leghiamo le velocità $\dot{q}$ all'errore e in modo da avere una relazione che descriva l'evoluzione dell'errore nel tempo:

$$\dot{e} = \dot{x}_d - J_A(q)\dot{q}$$

Nel seguito vedremo alcuno algoritmi CLIK in cui verrà scelta una determina $\dot{q}$ in funzione di e:

- **CLIK con Inversa dello Jacobiano**;
- **CLIK con Pseudo-Inversa dello Jacobiano**;
- **CLIK con Trasposta dello Jacobiano**;
- **CLIK del Secondo Ordine**.

2.6.2 Algoritmi del Primo Ordine

Gli **Algoritmi di CLIK del Primo Ordine** *consentono l'inversione cinematica specificando posizione e velocità.* Gli algortmi CLIK del primo ordine che trattiamo sono i seguenti:

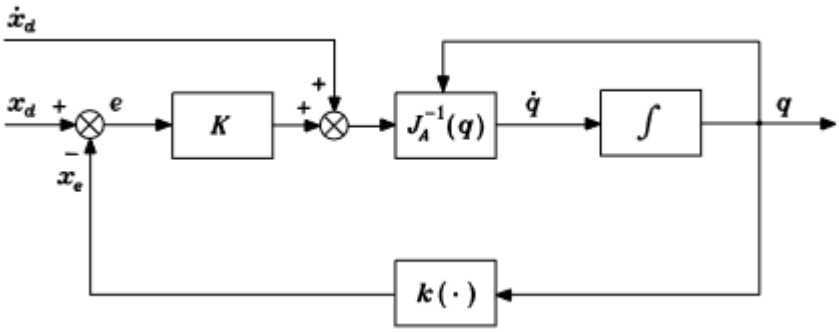
A. Inversa o Pseudo-Inversa dello Jacobiano:

a. Se il manipolatore non è ridondante possiamo risolvere la cinematica inversa con l'inversa dello Jacobiano: $\dot{q} = J_A^{-1}(q)(\dot{x}_d + Ke)$

b. Se il manipolatore è ridondante possiamo risolvere la cinematica inversa con la pseudo-inversa dello Jacobiano: $\dot{q} = J_A^{+}(q)(\dot{x}_d + Ke) + (I - J_A^{+}J_A)\dot{q}_0$

Entrambi gli algoritmi CLIK possono essere implementati con il seguente schema a blocchi:

- Il feedforward fornita x_d garantisce il mantenimento di un errore nullo lungo la traiettoria indipendentemente da $x_d(t)$;
- $k(\cdot)$ è il blocco non lineare che indica la funzione cinematica diretta con cui viene calcola la x_e e successivamente il segnale errore e;
- $J_A^{-1}(q)$ o J_A^{+} è introdotto per compensare $J_A(q)$ e rendere il sistema lineare.
- Gli integratori garantiscono errore a regime nullo;



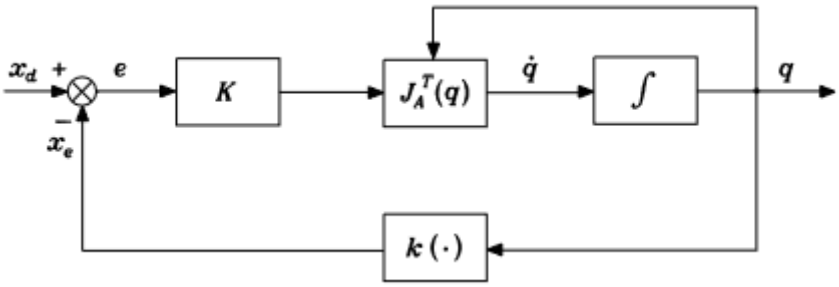
E' da notare che eventualmente potremmo applicare nuovamente l'algoritmo click in riferimento a $\dot{q}_0$ e impiegare la task priority nel caso di più task come fatte nei paragrafi precedenti.

Gli algoritmi CLIK citati ci consentono di definire il sistema errore: $\dot{e} + Ke = 0$

In cui, se scegliamo la matrice $K > 0$ e diagonale, tale sistema risulterà essere globalmente asintoticamente stabile e dunque l'errore e a regime tende a zero. In particolare, il velocità di convergenza a zero dipende dagli autovalori della matrice K, più essi saranno maggiori e tanto maggiore sarà tale velocità. Poiché però lavoriamo su sistemi tempo discreto, esiste un limite superiore su tali autovalori (in particolare per quello più grande) legata al tempo di campionamento del sistema. Allora se scegliamo una matrice K troppo grande non è possibile garantire l'asintotica stabilità. Una possibile scelta della matrice K può essere quella di porla come matrice diagonale costante i cui termini k sono circa pari l'inverso del tempo di campionamento ΔT: $k \approx 1/\Delta T$

B. Trasposta dello Jacobiano: $\dot{q} = J_A^T(q)Ke$

Lo schema a blocchi che implementa la soluzione della trasposta dello Jacobiano è il seguente:



Supponiamo però questa volta di non voler linearizzare il sistema e quindi eliminare il termine non lineare, ovvero cancellare $J_A(q)$ con l'inversa. La strada da impiegare nel caso vogliamo mantenere il sistema non lineare è il **Metodo di Lyapunov**, ovvero una condizione sufficiente mediante la quale possiamo calcolare la dipendenza $\dot{q}(e)$ tale che $e = 0$ e che quindi assicuri l'asintotica stabilità del sistema errore. Il metodo di Lyapunov afferma che il nostro sistema è GAS (Globalmente Asintoticamente Stabile) se troviamo una funzione V per le quali valgono le seguenti condizioni:

- ✓ **Condizione 1**: $V(x) > 0 \quad \forall x \neq 0 \quad V(0) = 0 \ per\ x = 0$
- ✓ **Condizione 2**: la funzione V deve decrescere nell'intorno del punto di equilibrio e quindi ciò si traduce nell'affermare che: $\dot{V}(x) < 0 \quad \forall x$

Scegliamo allora la funzione V come forma quadratica:

$$V = \frac{1}{2} e^T K e \quad dove: matrice\ K > 0$$

Andiamo ora a dimostrare le due condizioni del teorema di Lyapunov:

- ✓ **Condizione 1**: se abbiamo che K è una matrice simmetrica definita positiva allora:

$$V(e) > 0 \ \ \forall e \neq 0 \quad e \quad V(0) = 0 \quad per\ \ e = 0$$

- ✓ **Condizione 2**: calcoliamo la derivata rispetto al tempo di V osservando che $\dot{e} = \dot{x}_d - \dot{x}_e$ e e supponendo di avere un riferimento $x_d = cost$, da cui si ha $\dot{x}_d = 0$.

$$\dot{V} = e^T K \dot{e} = e^T K(\dot{x}_d - \dot{x}_e) = e^T K \dot{x}_d - e^T K \dot{x}_e = e^T K \dot{x}_d - e^T K J_A \dot{q}$$

Per dimostrare che $\dot{V} < 0$ dobbiamo studiare i segni dei due termini che lo compogono. Per quanto riguarda $e^T K \dot{x}_d$, esso può assumere qualsiasi segno ($\geq o \leq$) tuttavia non è un problema perché al termine della traiettoria $\dot{x}_d \to 0$. Più complesso è il discorso circa $e^T K J_A \dot{q}$ il quale dovrà essere positivo in modo da garantire $\dot{V} < 0$. Andiamo prima a studiare il termine $e^T K J_A$:

$$e^T K J_A = [\ 1 \times n\ ] \begin{bmatrix} n & \\ & n \end{bmatrix} \begin{bmatrix} m & \\ & n \end{bmatrix} = [\ 1 \times n\ ]$$

Allora affinchè V (e quindi la sua derivata) risulti essere uno scalare bisogna moltiplicare il vettore riga $[1 \times n]$ $e^T K J_A$ per un vettore $\dot{q}$ che sia un vettore colonna $[n \times 1]$. Banalmente per trovarci V scalare scegliamo il trasposto di $e^T K J_A$: $\dot{q} = J_A^T K e$

Allora se chiamiamo $z = e^T K J_A$ e $z^T = J_A^T K e$, il termine di seguito è di fatto la norma di z, la quale ovviamente risulta essere una quantità *sempre positiva o nulla*:

$$e^T K J_A J_A^T K e = [e^T K J_A] \cdot [J_A^T K e] = z^T z = \|z\|$$

In definitiva, abbiamo dimostrato che il termine $e^T K J_A J_A^T K e$ presente in $\dot{V}$ è sempre ≥ 0 e quindi siamo sicuri che al netto: $\dot{V} = e^T K \dot{x}_d - e^T K J_A J_A^T K e = -e^T K J_A J_A^T K e < 0$

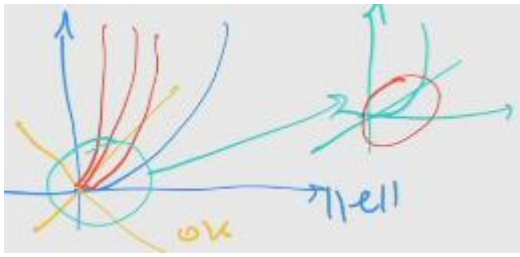
Essendo verificate entrambe le condizioni del teorema di Lyapunov, il sistema risulta essere asintoticamente stabile e dunque le traiettorie del sistema a regime convergono all'origine $e = 0$. Tale condizione può essere visualizzata anche graficamente. Supponendo per semplicità che l'errore abbia due componenti e_1 ed e_2 in modo da ragionare per semplicità in due dimensioni, andiamo a tracciare le curve di livello di V (ellissi viola) nel piano (e_1, e_2) come in figura. Allora se scelgo $K > 0$ e se partiamo da un certo errore (punto azzurro), essendo il sistema asintoticamente stabile, la traiettoria convengerà appunto nell'origine $e = 0$.

Fino ad ora abbiamo parlato della condizione per cui l'errore e a regime tende a zero. E' interessante però chiederci come si comporta l'algoritmo CLIK con trasposta dello jacobiano in transitorio. Allora riprendendo l'espressione della $\dot{V}$: $\dot{V} = e^T K \dot{x}_d - e^T K J_A J_A^T K e$

Abbiamo che:

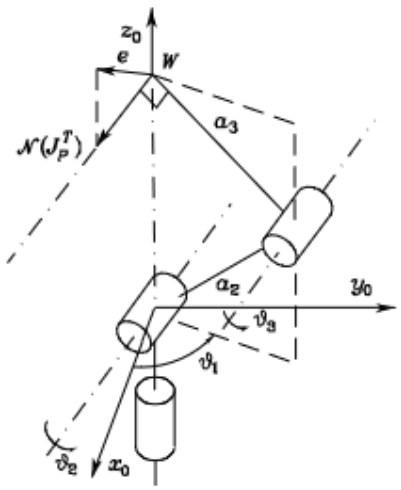
- Il termine $e^T K \dot{x}_d$, come detto precedentemente, è di segno incerto. A regime esso non dava alcun problema perché $\dot{x}_d \to 0$. Invece in transitorio $\dot{x}_d \neq 0$ e grazie alla matrice K l'intero termine di segno incerto è lineare nell'errore e;
- Il termine $e^T K J_A J_A^T K e$ di segno ≥ 0 che garantiva la condizione $\dot{V} < 0$ a regime, anche in transitorio continua a mantenere il medesimo segno. Tale termine, grazie alla matrice K^2, è quadratico nell'errore.

Tracciamo un grafico di questi due termini in funzione della norma dell'errore. Quindi sull'asse delle ascisse avremo $\|e\|$ mentre sull'asse delle ordinate i due termini prima citati. E' da notare che non impieghiamo direttamente e in quanto esso presenta più componenti, perciò è più facile ragionare sulla sua norma. Allora il termine $e^T K J_A J_A^T K e$ essendo quadratico nell'errore sarà una parabola rivolta verso l'alto, mentre il primo termine essendo lineare nell'errore sarà una retta verso il basso o verso l'alto:



a) Nel caso il termine $e^T K \dot{x}_d$ è la retta che va verso il basso non ci sono problemi perché il termine di segno positivo prevale sempre su di esso. Assicurandoci così che anche in transitorio $\dot{V}$ è sempre negativo.
b) Nel caso il termine $e^T K \dot{x}_d$ è la retta che va verso l'alto, si avrà un certo tratto che prevale sulla parabola (e quindi il termine buono). Allora potremmo pensare di aumentare K quel tanto che basta da non perturbare la stabilità del sistema e far prevalere la parabola sulla retta. Tuttavia, nell'intorno dell'origine avremo sempre la medesima condizione prima citata.

Risultato di ciò, in transitorio l'algoritmo CLIK con trasposta dello Jacobiano non ha un buon comportamento.



Ultimo aspetto da valutare prima di tirare le somme circa i pro e i contro di questo algoritmo è il caso problematico in cui $Ke \in \mathcal{N}(J_A^T)$ e quindi $\mathcal{N}(J_A^T) \neq \emptyset$. In tal caso, infatti, la $\dot{V}$ è solo semi-definita negativo (≤ 0) e ciò vuol dire che l'algoritmo CLIK può bloccarsi a $\dot{q} = 0$ con un errore $e \neq 0$. Tuttavia, questa situazione si verifica solo se la posizione dell'end-effector assegnata *non è effettivamente raggiungibile* dalla configurazione corrente. Per ragionare su questo aspetto consideriamo il caso di un manipolatore antropomorfo in singolarità di spalla come in figura. Il $\mathcal{R}(J_A)$ in questa situazione non è l'intero spazio euclideo ma solo il piano π della sotto-struttura braccio/avrambraccio. In algebra, c'è una relazione notevole per cui $\mathcal{R}(J_A) = \mathcal{N}^{\perp}(J_A^T)$, dove il $\mathcal{N}(J_A^T)$ è la retta normale al piano π. Allora noi stiamo chiedendo al manipolatore che si trova nella posizione x_e giacente sull'asse di base di muoverci in x_d tale che $x_d - x_e$ è ortogonalmente al piano, il quale però è un moto impossibile. Ciò però non è da intendersi come una falla nell'algoritmo, anzi testimonio la sua *forte robustezza nell'intorno delle singolarità in quanto non tenta di invertire lo Jacobiano*, il cui determinante sarebbe molto piccolo e causando il problema delle eccessive velocità. Tra l'altro avere una direzione lungo il $\mathcal{N}(J_A^T)$ è un vantaggio anche in vista della discretizzazione, perché con le diverse approssimazioni numeriche se c'è una piccola componente fuori dal nullo avremo una via di fuga dall'asse singolare.

In conclusione, l'algoritmo CLIK con trasposta dello Jacobiano ha lo svantaggio di non avere un buon comportamento in transitorio come il CLIK con inversa, però ha i pregi di essere:

- ✓ Robusto nei confronti della ridondanza (nel senso che la condizione di ridondanza non precluse l'uso di J_A^T);
- ✓ Robusto nei confronti delle singolarità;
- ✓ Scarso sforzo computazionale;
- ✓ Errore a regime nullo.

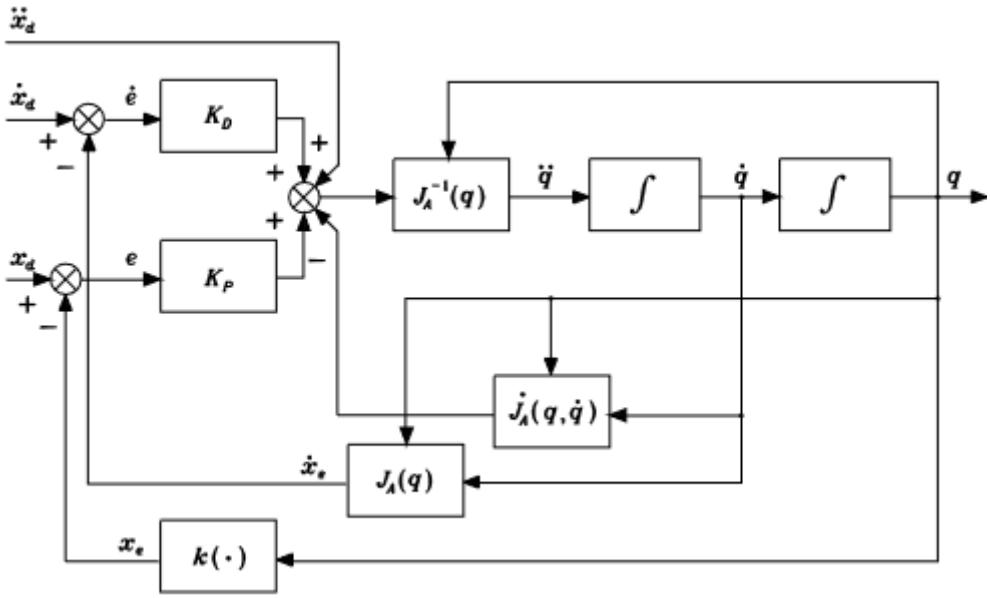
2.6.3 Algoritmi del Secondo Ordine

Gli **Algoritmi CLIK del Secondo Ordine** *consentono l'inversione cinematica specificando posizione velocità e accellerazione*. Allora il CLIK del secondo ordine, date il set completo di $x_d, \dot{x}_d, \ddot{x}_d$, cerca di ricavare le $q, \dot{q}, \ddot{q}$ tale che x_e tende a x_d e $\dot{x}_e$ tende a $\dot{x}_d$:

$$\begin{matrix} x_d \\ \dot{x}_d \\ \ddot{x}_d \end{matrix} \rightarrow \begin{matrix} q \\ \dot{q} \\ \ddot{q} \end{matrix} : \begin{matrix} x_e \rightarrow x_d \\ \dot{x}_e \rightarrow \dot{x}_d \end{matrix}$$

La soluzione dell'algoritmo CLIK del secondo ordine è dato dalle seguenti soluzioni, le quali sono implementata dallo schema a blocchi in figura:

a. Se il manipolatore non è ridondante: $\ddot{q} = J_A^{-1}(q)\left(\ddot{x}_d + K_D\dot{e} + K_P e - \dot{J}_A\dot{q}\right)$
b. Se il manipolatore è ridondante: $\ddot{q} = J_A^{+}(q)\left(\ddot{x}_d + K_D\dot{e} + K_P e - \dot{J}_A\dot{q}\right) + (I - J_A^{+}J_A)\ddot{q}_0$



Tutti i precedenti algoritmi di inversione cinematica citati fino ad ora era del primo ordine in quanto specificafano solo posizione e velocità, tuttavia ,come vedremo in seguito, il modello dinamico del manipolatore è del secondo ordine e dunque algoritmi di questa categoria risultano utili per fini di controllo e in particolar modo per l'inversione di una traiettoria di moto. Studiamo questa classe di algoritmi ragionando per semplicità su un manipolatore non ridondante. Riprendiamo anzitutto l'equazione cinematica differenziale (al posto di v_e abbiamo $\dot{x}_e$): $\dot{x}_e = J_A(q)\dot{q}$

Derivando quest'ultima, otteniamo la relazione che lega le accelerazioni dello spazio dei giunti e le accelerazioni dello spazio operativo: $\ddot{x}_e = J_A(q)\ddot{q} + \dot{J}_A(q,\dot{q})\dot{q}$

Nell'ipotesi di uno Jacobiano quadrato e non singolare, la soluzione prevede l'impiego dell'inversa J_A^{-1}: $\ddot{q} = J_A^{-1}(q)\left(\ddot{x}_e - \dot{J}_A(q,\dot{q})\dot{q}\right)$

Inoltre, teniamo in conto del fenomeno della deriva della soluzione a causa dell'integrazione numerica secondo la regola di Eulero di $\ddot{q}$. Per cui l'errore e la sua derivata seconda sono:

$$e = x_d - x_e$$
$$\ddot{e} = \ddot{x}_d - \ddot{x}_e = \ddot{x}_d - J_A(q)\ddot{q} - \dot{J}_A(q,\dot{q})\dot{q}$$

A questo punto scegliamo l'accelerazione ai giunti come:

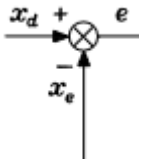
$$\boldsymbol{\ddot{q}} = J_A^{-1}(q)\left(\ddot{x}_e - \dot{J}_A(q,\dot{q})\dot{q}\right) = \boldsymbol{J_A^{-1}(q)}\left(\boldsymbol{\ddot{x}_d} + \boldsymbol{K_D\dot{e}} + \boldsymbol{K_P e} - \boldsymbol{\dot{J}_A\dot{q}}\right) \quad Dove: \begin{matrix} K_P > 0 \\ K_D > 0 \end{matrix}$$

Sostiuendo $\ddot{q}$ in $\ddot{e}$ otteniamo la seguente equazione differenziale omogenea, della quale è possibile dimostrare l'asintotica stabilità:

$$\begin{aligned} \ddot{e} &= \ddot{x}_d - J_A(q)\ddot{q} - \dot{J}_A(q,\dot{q})\dot{q} \\ &= \ddot{x}_d - J_A(q)\left[J_A^{-1}(q)\left(\ddot{x}_d + K_D\dot{e} + K_P e - \dot{J}_A(q,\dot{q})\dot{q}\right)\right] - \dot{J}_A(q,\dot{q})\dot{q} \\ &= \ddot{x}_d + K_D\dot{e} + K_P e = 0 \end{aligned}$$

Inoltre, l'errore tende a zero lungo la traiettoria con una velocità di convergenza che dipende dall'opportuna scelta delle matrici K_P e K_D. Per concludere, è necessario specificare che questo algoritmo deve necessariamente lavorare con *errori molto piccoli* in quanto il set di variabili $q, \dot{q}, \ddot{q}$ che calcolo saranno i riferimenti per il sistema di controllo nello spazio dei giunti. Allora, nel primo stadio inverto la cinematica ottenendo $q, \dot{q}, \ddot{q}$, dopodichè queste ultime saranno le grandezze desiderate che andranno a controllare gli attuatori dei giunti affinchè l'end-effector esegua il moto desiderato.

2.6.4 Errore di Orientamento



In tutti gli algoritmi CLIK visti fino ad ora e nei relativi schemi implementativi il ruolo chiave è l'errore di posa e, il quale abbiamo detto essere la differenza tra la posa desidera e quella effettiva dell'end-effector: $e = x_d - x_e$
Ricordiamo che la posa include sia la parte di posizione che orientamento dell'organo terminale, per cui è doveroso caratterizzare più nel dettaglio l'errore tenendo conto di questo aspetto.

Circa la parte di posizione semplicemente abbiamo che l'errore di posizione e_p è data dalla differenza tra la posizione desiderata e quella effettiva, da cui possiamo derivando rispetto al tempo otteniamo anche l'evoluzione di tale errore.

$$e_P = p_d - p_e(q)$$
$$\downarrow$$
$$\dot{e}_p = \dot{p}_d - \dot{p}_e(q) = \dot{p}_d - J_P(q)\dot{q}$$

Per quanto riguarda l'orientamento potremmo pensare di scrivere qualcosa di analogo alla posizione come definito di fianco. Pertanto, supponendo che non si verifichino singolarità cinematiche o di rappresentazione, la soluzione con inversa dello Jacobiano per un manipolatore non ridondante è:

$$e_O = \phi_d - \phi_e(q)$$
$$\downarrow$$
$$\dot{e}_O = \dot{\phi}_d - \dot{\phi}_e(q)$$

$$\dot{q} = J_A^{-1}(\dot{x}_d + Ke) = J_A^{-1}\begin{bmatrix}\dot{p}_d + K_P e_P \\ \dot{\phi}_d + K_o e_O\end{bmatrix} \qquad dove: \begin{matrix} K_P > 0 \\ K_O > 0 \end{matrix}$$

Il problema però è che l'espressione dell'errore così definita presenta diverse criticità:

- Gli angoli di Eulero $\phi_e(q)$ relativi alla terna dell'end-effector non sempre sono funzione diretta di q nel caso generale, invece è vero nel caso di manipolatori planari;
- La differenza $\phi_d - \phi_e(q)$ può avere un senso nell'ottica del controllo ma dal punto di vista geometrico non ha alcun significato (il risultato sarebbe un vettore i cui elementi sono dei numeri che non indicano un orientamento);
- Gli angoli di Eulero ϕ_d e ϕ_e si estraggono dalle rispettive matrici di rotazione e ciò comporta un certo carico computazionale.

Capiamo perciò che dobbiamo trovare un nuovo modo per definire l'errore di orientamento e a sua volta una corrispondente soluzione alla cinematica differenziale inversa. Lo studio che verrà di seguito verrà realizzato secondo queste ipotesi:

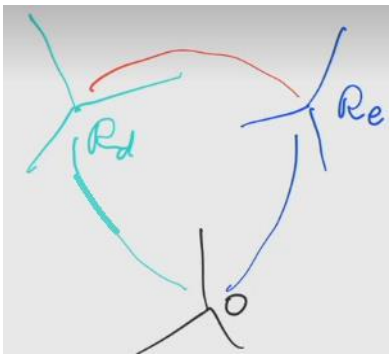
- ✓ Supponiamo di impiegare sempre l'algoritmo CLIK con inversa dello Jacobiano (eventualmente tutti i concetti che vedremo si può estendere anche all'algoritmo con J_A^T);
- ✓ Il manipolatore è dotato di polso sferico.

La scelta dell'ipotesi circa il polso sferico non è motivato solo nel conferire generalità allo studio di seguito ma anche perché grazie ad esso impiegare il disaccoppiamento cinematico per risolvere il problema della cinematica inversa per la parte di posizione separatamente da quello per la parte di orientamento. In particolare, si potrebbe pensare di adottare una risoluzione in due stadi:

- **Stadio 1 – Risoluzione della Parte di Posizione**: grazie al disaccoppiamento cinematico possiamo scegliere come punto terminale il centro polso anziché l'origine della terna dell'end-effector. Così facendo risolviamo la cinematica inversa relativa alla struttura portante. L'utente specifica p_d e $R_d = [n_d \;\; s_d \;\; a_d]$ (quindi sono noti) e da questo si ricava la posizione desiderata del centro polso p_{wd} come: $p_{wd} = p_d - d_6 a_d$

Successivamente, su utilizza l'algoritmo CLIK $\dot{q}_P = J_P^{-1}(\dot{p}_d + K_P e_P)$ per risolvere la cinematica inversa della struttura portante in modo da calcolare $\dot{q}_P$ e q_P. In vista del successivo stadio, ci conviene ricavare da $\dot{q}_P$ e q_P la matrice R_w, la quale è la matrice di rotazione dovuta ai primi tre giunti del manipolatore che individua la terna base per il polso sferico.

- **Stadio 2 – Risoluzione della Parte di Orientamento**: grazie alla conoscenza di R_d e il calcolo di R_w, possiamo ricavare la matrice di rotazione che rappresenta la specifica di moto riferita alla terna base del polso sferico (anziché dell'intera struttura): $R_d^w = R_w^T R_d$
  Dal calcolo di R_d^w è possibile ricavare q_O e $\dot{q}_O$ in due modi:
  a) Sfruttando le formule inverse degli angoli di Eulero;
  b) Imposto un algoritmo che utilizza di angoli di Eulero visto che $q_O = [\theta_4 \;\; \theta_5 \;\; \theta_6]$ in questa specifica casistica del polso sferico sono una terna di angoli di Eulero. E' da notare che, l'approccio appena descritto risulta essere impiegabile anche se la struttura portante, invece che avere tre giunti, ne presentasse per esempio quattro.



Andiamo a studiare più nel dettaglio il problema dell'orientamento supponendo di avere le seguenti terne definite rispetto ad una terna 0 (con la matrice di rotazione pari alla matrice identità):

- $R_d = [n_d \quad s_d \quad a_d]$ = matrice di rotazione desiderata per l'end-effector;
- $R_e = [n_e \quad s_e \quad a_e]$ = matrice di rotazione dell'end-effector calcolata dalla variabili di giunto.

Il nostro obiettivo è fare in modo che la terna effettiva abbia lo stesso orientamento di quella desiderata, ovvero: $R_e = R_d$

Definiamo allora una matrice di rotazione mutua R_e^d tra la terna desiderata e quella effettiva, la quale possiamo ricavare come segue: $R_e = R_d R_e^d \quad \rightarrow \quad R_e^d = R_d^T R_e$

Avendo la matrice di rotazione mutua R_e^d posso procedere con tre differenti strade:

A. Angolari di Eulero

Da R_e^d estraggo gli angoli di Eulero ϕ_{de}, i quali non coincidono $\phi_d - \phi_e$, i quali posso impiegare come errore di orientamento da portare a zero mediante un algoritmo. Infatti, se $\phi_{de} = 0$ la matrice di rotazione mutua $R_e^d = I$ e dunque risulterà essere soddisfatto l'obiettivo di far coincidere le due terne visto che: $R_e = R_d$

Tra l'altro questo approccio è computazionalmente leggero perché non dobbiamo estrarre separatamente ϕ_d e ϕ_e, dopodichè farne la differenza.

B. Approccio con Rappresentazione Asse/Angolo

Da R_e^d estraggo l'asse r e l'angolo θ in modo tale da definire l'errore di orientamento tra le due terne come: $\boldsymbol{e_O = r \sin(\theta)}$

E' da notare che l'espressione dell'errore è stata scelta appositamente per fare in modo di definire l'errore di orientamento mediante tre componenti (come per gli angoli di Eulero) anziché quattro come ci porterebbe a fare la rappresentazione asse/angolo.

Facciamo alcune osservazioni riguardo l'angolo θ:

- Esso rappresenta l'entità dell'errore di orientamento, infatti se $\theta = 0$ l'errore si annulla, $e_O = 0$. Il caso $\theta = \pi$ non viene considerato perché l'errore è nullo ma le terne sono in posizione opposta l'una rispetto all'altra;
- L'espressione dell'errore che abbiamo definito vale solo se $\theta \in [-\pi/2, \pi/2]$. Ciò ci porterebbe a pensare che questo è uno svantaggio. Questo in realtà non è un problema perché se l'angolo tra le due terne in modulo è maggiore di 90° vuol dire che esse sono letteralmente distanti fra loro. Nella realtà facciamo eseguire al robot una traeittoria a partire dalla condizioni iniziale di aggancio, ovvero le due terne inizialmente sono coindicidenti e quindi $R_e = R_d$. Da cui avremo che l'errore risulterà essere diverso da zero a causa della discretizzazione, dall'uso della pseudo-inversa, ecc. però alla fine dei conti θ successivamente sarà compreso nel range descritto.

A questo punto si può dimostrare che l'errore di orientamento può essere scritto nella seguente forma. Inoltre, la condizione $\theta \in [-\pi/2, \pi/2]$ si trasforma in una serie di condizioni che affermano la positività del prodotto scalare tra i versori degli assi delle terne:

$$\boldsymbol{e_O} = \frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{n_e} \times \boldsymbol{n_d} + \boldsymbol{s_e} \times \boldsymbol{s_d} + \boldsymbol{a_e} \times \boldsymbol{a_d}) \qquad se \qquad \begin{matrix} n_e^T n_d \geq 0 \\ n_e^T n_d \geq 0 \\ n_e^T n_d \geq 0 \end{matrix}$$

La derivata dell'errore di orientamento rispetto al tempo è:

$$\dot{e}_O = L^T \omega_d - L\omega_e \qquad dove: \;\; L = -\frac{1}{2}\big(S(n_d)S(n_e) + S(s_d)S(s_e) + S(a_d)S(a_e)\big)$$

Allora in definitiva l'errore complessivo è:

$$\dot{e} = \begin{bmatrix} \dot{e}_P \\ \dot{e}_O \end{bmatrix} = \begin{bmatrix} \dot{p}_d - p_e(t) \\ L^T \omega_d - LJ_O \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{p}_d - J_P \dot{q} \\ L^T \omega_d - LJ_O \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{p}_d \\ L^T \omega_d \end{bmatrix} - \begin{bmatrix} I & 0 \\ 0 & L \end{bmatrix} J\dot{q}$$

Da cui possiamo derivare l'espressione di $\dot{q}$ che, come si può osservare, ricorre all'uso dello Jacobiano geometrico anziché di quello analitico come nel caso predecente con gli angoli di Eulero, evitando così il verificarsi di singolarità di rappresentazione.

$$\boldsymbol{\dot{q}} = \boldsymbol{J^{-1}} \begin{bmatrix} \boldsymbol{\dot{p}_d + K_P e_P} \\ \boldsymbol{L^{-1}(L^T \omega_d + K_O e_O)} \end{bmatrix}$$

Se volessimo convalidare la corretta scelta dell'espressione dell'errore di orientamento basta sostituire $\omega_e = J_O^{-1}\big(L^{-1}(L^T \omega_d + K_O e_O)\big)$ nella $\dot{e}_O = L^T \omega_d - L\omega_e$. Se dopo tale sostituzione si verifica che $\dot{e}_O + K_O e_O = 0$, allora abbiamo ottenuto come risultato che il sistema errore di orientamento è globalmente asintoticamente stabile.

C. Approccio con il Quaternione

Indicando con:

- $Q_d = \{\eta_d, \epsilon_d\}$ = quaternione associato ad R_d;
- $Q_e = \{\eta_e, \epsilon_e\}$ = quaternione associato ad R_e.

L'errore di orientamento descritto dalla matrice di rotazione $R_d R_e^T$ può essere espressi in termini di quaternione mutuo ΔQ definito come: $\Delta Q = \{\Delta\eta, \Delta\epsilon\} = Q_d * Q_e^{-1}$

Infatti se $\Delta Q = \{1,0\}$ allora $R_e = R_d$, quindi le due terne risulteranno essere allineate. Allora l'errore di orientamento in termini di ΔQ, per far si che abbia tre componenti, è dato dalla parte vettoriale estratta da quest'ultimo: $\boldsymbol{e_O = \Delta\epsilon = \eta_e(q)\epsilon_d - \eta_d\epsilon_e(q) - S(\epsilon_d)\epsilon_e(q)}$

L'unico aspetto negativo di questa espressione dell'errore è che non è possibile calcolare direttamente η_e e ϵ_e dalle variabili di giunto ma sarà necessario il calcolo intermedio di R_e. La soluzione che sfrutta l'inversa dello Jacobiano (geometrico) è la seguente:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\begin{bmatrix}\dot{\boldsymbol{p}}_d + \boldsymbol{K_P e_P} \\ \boldsymbol{\omega_d} + \boldsymbol{K_O e_O}\end{bmatrix}$$

Anche in questo caso verifichiamo la giusta scelta dell'errore di orientamento andando a sostituire $\omega_e = J^{-1}(\omega_d + K_O e_O)$ in $\dot{e}_O$ e verificando che il sistema è globalmente asintoticamente stabile. Contrariamente al caso della rappresentazione asse/angolo, l'equazione dell'errore di orientamento è non lineare in quanto contiene l'errore di velocità angolare $(\omega_d - \omega_e)$ anziché $\dot{e}_O$: $\omega_d - \omega_e + K_O e_O = 0$

Perciò non posso affermare immediatamente che il sistema è globalmente asintotitcamente stabile. Allora per dimostrare ciò facciamo uso delle **Equazioni di Propagazione del Quaternione** che legano le derivate temporali di quest'ultimo alla velocità angolare ω. Calcoliamo queste equazioni sia per Q_d che per Q_e:

$$\boldsymbol{Funzioni\ Scalari}: \quad \dot{\boldsymbol{\eta}}_e = -\frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{\epsilon}_e^T\boldsymbol{\omega}_e \qquad \dot{\boldsymbol{\eta}}_d = -\frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{\epsilon}_d^T\boldsymbol{\omega}_d$$

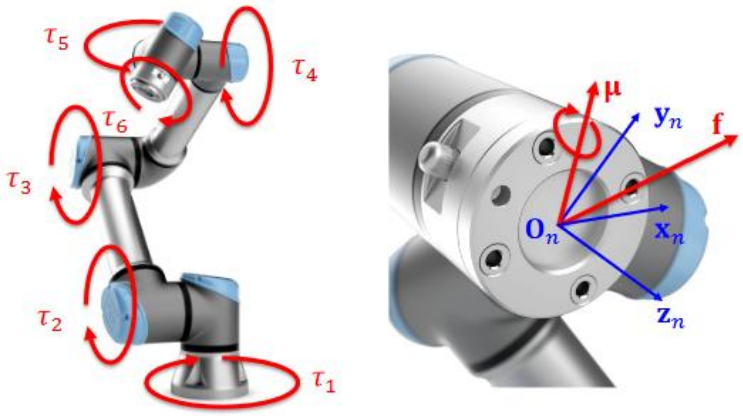
$$\boldsymbol{Funzioni\ Vettoriali}: \quad \dot{\boldsymbol{\epsilon}}_e = \frac{\boldsymbol{1}}{\boldsymbol{2}}\big(\boldsymbol{\eta}_e\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\epsilon}_e)\big)\boldsymbol{\omega}_e \qquad \dot{\boldsymbol{\epsilon}}_d = \frac{\boldsymbol{1}}{\boldsymbol{2}}\big(\boldsymbol{\eta}_d\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\epsilon}_d)\big)\boldsymbol{\omega}_d$$

Studiamo la stabilità del sistema errore di orientamento $\omega_d - \omega_e + K_O e_O = 0$ ricorrendo a Lyapunov. A tal proposito adottiamo la funzione di Lyapunov candidata V sotto indicata. Differenziandola rispetto al tempo e tenendo conto delle equazioni di propagazione riusciamo a dimostrare che la $\dot{V}$ è definita negativa confermando così che il sistema è globalmente asintoticamente stabile:

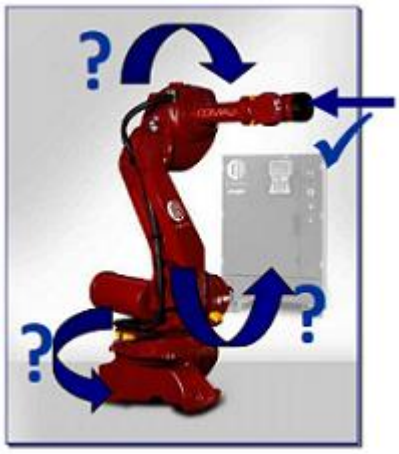
$$V = (\eta_d - \eta_e)^2 + (\epsilon_d - \epsilon_e)^T(\epsilon_d - \epsilon_e) \quad \rightarrow \quad \dot{V} = -e_O^T K_O e_O$$

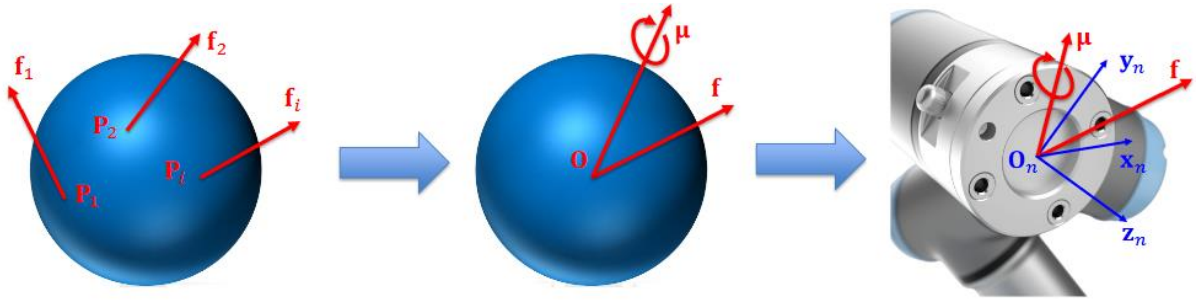
2.7 Statica

La **Statica** *è lo studio del legame tra wranch (forze generalizzate intese come forze lineari e momenti) applicate ai giunti, che indicheremo con* $\tau \in R^{n\times 1}$*, e le wranch applicate all'end-effector, che indicheremo con* $\gamma_e \in R^{r\times 1}$ (r è la dimensione dello spazio operativo). *L'* **Equazione Fondamentale della Statica** definisce questo legame grazie all'impiego della trasposta dello Jacobiano: $\boldsymbol{\tau = J^T(q)\gamma_e}$



Il motivo che ci spinge a studiare la statica di un manipolatore è la necessità di comprendere come una forza applicata dall'esterno all'organo terminale possa essere *equilibrata staticamente* dalle coppie applicate ai giunti. Nella realtà, sull'organo terminale agiscono tipicamente una sistema di forza in diversi punti. Per cui è necessaria una rappresentazione sintetica di queste ultime. L'end-effector può essere visto come un corpo rigido sul quale agisce un'unica wranch risultante somma di tutte le forze e momenti agenti su di esso in uno specifico punto, il quale sarà l'origine della terna solidale all'end-effector stesso. La forza lineare risultante f_e e il momento risultante μ_e li inglobiamo in un unico vettore γ_e, il quale abbiamo indicato come wranch agenti sull'organo terminale.





A questo punto ci chiediamo come facciamo a caratterizzare la relazione fondamentale della statica e dunque il legame tra γ_e e le coppie ai giunti τ. Per farlo calcoliamo il lavoro compiuto dalle due forze in questione:

- **Lavoro Elementare** $\boldsymbol{dW_q}$: è il lavoro elementare compiuto *dal singolo giunto* dW_{q_i} ed è pari al prodotto tra la coppia del giunto τ_{q_i} per lo spostamento del giunto dq_i. Poiché dobbiamo considerare il lavoro elementare complessivo scriviamo: $\boldsymbol{dW_q = \tau^T dq}$

- **Lavoro Elementare $\boldsymbol{dW_\gamma}$**: è il lavoro compito dalle forze ageti sull'end-effector. In teoria noi vorremmo una scrittura del tipo $dW_\gamma = \gamma^T dx_e$, tuttavia se la parte di posizione la riusciamo a scrivere agevolmente, lo stesso non si può dire per la parte di orientamento. Questa passaggio è necessario in quanto stiamo parlando di grandezze differenti. Lo spostamento elementare lineare possiamo scriverlo come dp_e, mentre lo spostamento angolare come $\omega_e dt$. Ricordando che $v_e = [\dot{p}_e \quad \omega_e]^T = J(q)\dot{q} = J(q)dq/dt$, portando a primo membro il dt otteniamo il termine in evidenza.

$$\boldsymbol{dW_\gamma} = f_e^T dp_e + \mu_e^T \omega_e dt = f_e^T [J_P(q)dq] + \mu_e^T [J_O(q)dq] = \boldsymbol{\gamma_e^T J(q) dq}$$

Osserviamo che entrambi i lavori elementari li abbiamo espressi in funzione degli spostamenti elementari dq. Poiché il manipolatore è considerato un sistema tempo-invariante in cui la configurazione dipende da q ma non esplicitamente dal tempo, si può affermare che gli spostamenti elementari coincidono con gli spostamenti virtuali: $dq = \delta q$

Allora: $\delta W_\tau = \tau^T \delta q \qquad \delta W_\gamma = \gamma_e^T J(q) \delta q$

Possiamo ora definire il seguente principio. Il **Principio dei Lavori Virtuali** ci dice che il manipolatore è in equilibrio statico *se e solo se*:

$$\boldsymbol{\delta W_\tau = \delta W_\gamma \quad \forall\, \delta q = spostamento\ virtuale}$$

Sostituendo l'espressioni di dW_q e dW_γ ricavate in precedenza nella relazione dei lavori virtuali arriviamo a definire l'**Equazione Fondamentale della Statica**: $\boldsymbol{\tau = J^T(q)\gamma_e}$

Infine, è da notare che se nella cinematica differenziale leggevamo lo Jacobiano sulle righe, ora che impieghiamo il suo trasposto lo leggeremo sulle colonne.

2.8 Dualità Cineto-Statica

Indicando con il termine:

- **Twist**: le velocità generalizzate, le quali tengono conto di velocità lineare e angolare;
- **Wranch**: le forze generalizzate, le quali tengono conto di forze lineari e momemnti.

La **Dualità Cineto-Statica** *afferma che le grandezze che, nel rispetto dei lavori virtuali, sono duali alle twist sono proprio le wranch. Inoltre, lo Jacobiano caratterizza questo legame essendo presente sia nella cinematica differenziale che nella statica.*

$$\boldsymbol{v_e = J\dot{q}}$$
$$\boldsymbol{\tau = J^T \gamma_e}$$

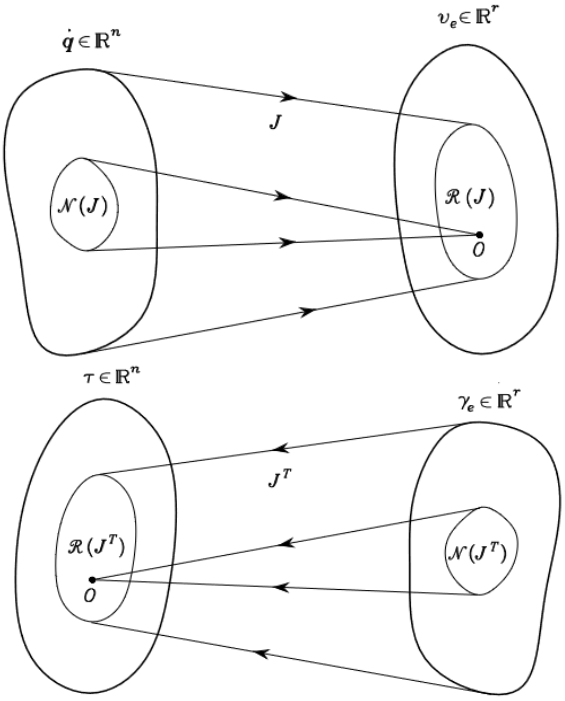
Osservando il principio della dualità cineto-statica potrebbe sembrare ambiguo collegare due studi che sembrano essere contrastanti in quando:

- La cinematica differenziale studia il moto di una struttura cinematica in assenza di forze;
- La statica studia una struttura che è in equilibrio e quindi immobile in presenza di forze.

In realtà, a fare da collante tra i due studi è proprio il principio dei lavori virtuali, il quale ricordiamo afferma che il manipolatore è in equilibrio se e solo se il lavoro compiuto da tutte le forze agenti sul sistema è nulla, ovvero si bilanciano, e ciò per ogni possibile spostamento (spostamento virtuale). Se introduciamo il tempo in questa relazione passiamo alle velocità e quindi non parliamo più di lavoro ma bensì di *potenza*. Allora se pensiamo al manipolatore come ad un interfaccia tra lo spostamento dei giunti e gli spostamenti dell'organo terminale, quando esso è ferma vuol dire che la potenza dei giunti è uguale alla potenza dell'organo terminale. In conclusione, grazie alla dualità cineto-statica arriviamo alla conclusione che il manipolatore è a tutti gli effetti **trasformatore meccanico di potenza**.

Questo legame tra cinematica differenziale e statica può essere osservato anche da un punto di vista matematico ragionando sul range e sul nullo di J e J^T.

- **Cinematica Differenziale**: abbiamo la trasformazione in figura in cui, grazie a J, passiamo dall'insieme $\dot{q}$ al sotto-insieme $\mathcal{R}(J)$ contenuto nell'insieme della v_e. Inoltre, nell'insieme delle $\dot{q}$ è presente il sotto insieme nullo dello Jacobiano $\mathcal{N}(J)$ che è quel sotto-insieme di $\dot{q}$ associate a velocità nulla dell'organo terminale.
- **Statica**: abbiamo la trasformazione in figura in cui, grazie a J^T, passiamo dall'insieme delle forze γ_e al $\mathcal{R}(J^T)$ contenuto nell'insieme delle τ. Inoltre, nell'insieme delle γ è presente il sotto insieme $\mathcal{N}(J^T)$ che è quel sotto-insieme di γ_e associate a coppie nulle ai giunti.

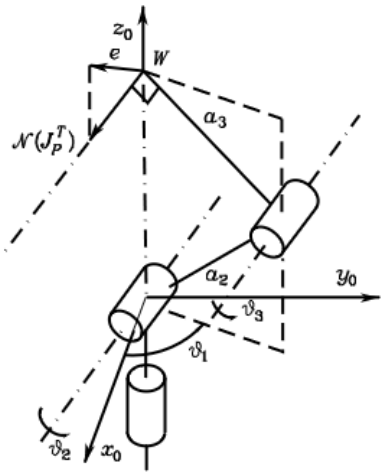


In algebra, esiste una teorema che lega il range di una matrice A al nullo della sua trasposta: $\mathcal{R}(A) = \mathcal{N}^{\perp}(A^T)$ $\quad$ $\mathcal{N}(A) = \mathcal{R}^{\perp}(A^T)$

Nel nostro caso con la dualità cineto-statica esattamente lo stesso risultato:

$$\boldsymbol{\mathcal{R}(J) = \mathcal{N}^{\perp}(J^T)} \qquad \boldsymbol{\mathcal{N}(J) = \mathcal{R}^{\perp}(J^T)}$$

E' da notare che le $\gamma_e \in \mathcal{N}(J^T)$ sono tutte quelle forze all'organo terminale *assorbite* dalla struttura meccanica nella struttura corrente. Allora quelle forze, se sono bilanciate dalla struttura, non richiedono l'applicazione di alcuna coppia. Per studiare nel dettaglio questo aspetto consideriamo un manipolatore non ridondante (la ridondanza crea un $\mathcal{N}(J)$ ma è inifluente per J^T). Inoltre, supponiamo manipolatore antropomorfo in singolarità di spalla. Il range di J abbiamo detto essere piano definito dalla sotto-struttura braccio-avambraccio, perciò il complemento ortogonale sarà la retta uscente dal suddetto piano. Allora se io cerco di imprimere una forza γ_e all'organo terminale che appartiene al $\mathcal{N}(J^T)$, non riuscirò a muovere il manipolatore in quanto essa sarà completamente assorbita dalla struttura.

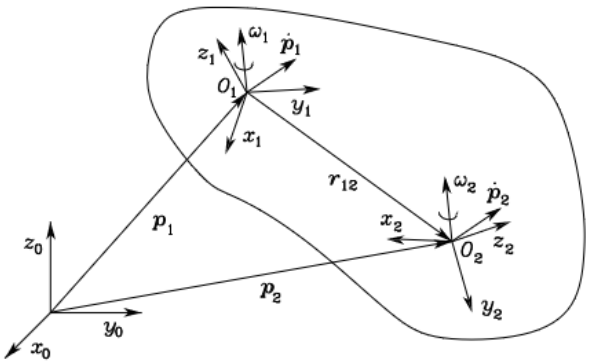


Quando abbiamo visto gli algoritmi Clik si basa su J o su J^T, dove in quest'ultimo la legge era $\dot{q} = J^T K e$. Poc'anzi abbiamo detto che $\tau = J^T \gamma_e$. Confrontando queste equazione ke gioca il ruolo della forza γ_e: $\gamma_e = Ke$

Capiamo perciò che quest'ultima è una forza elastica di rigidità K e intensità Ke. Allora fare un CLik con la trasposta significa trovarsi in x_e e voler andare in x_d, si crea un errore. Fisicamente è come se andassi a tirare il manipolatore con una forza elastica ke per tirarlo dalla posizione corrente a quella desiderata. Se ragionaniamo su $\dot{q}$ e τ invece, il manipolatore è come se fosse un sistema ideale in cui le coppie sono delle coppie di attrito viscose unitarie.

2.8.1 Trasformazione di Velocità e Forze

Abbiamo capito ormai che lo Jacobiano può essere usato a livello generale per legare velocità e forza di due sistemi differenti. Approfondiamo tale aspetto considerando l'esempio in figura in cui abbiamo una terna di riferimento e un corpo rigido su cui defniamo due terne solidale ad esso. Di queste terne conosciamo p_1 e p_2 che sono rispettivamente i vettori di posizione delle loro origini rispetto alla terna di riferimento.



Il primo passo è stabilire il legame tra le velocità (lineari e angolari) delle due terne.

- **Velocità Lineari**: avremo un primo termine relativo alla velocità lineare di una terna rispetto a quella base e un secondo termine in cui compare la distanza r_{12} tra le due terne:

$$\dot{p}_2 = \dot{p}_1 + \omega_1 \times r_{12} = \dot{p}_1 + S(\omega_1)r_{12} = \dot{p}_1 - S(r_{12})\omega_1$$

- **Velocità Angolari**: essendo vettori liberi relativi allo stesso corpo rigido: $\omega_2 = \omega_1$

Passando alla forma matriciale, la matrice a secondo membro è a tutti gli effetti uno Jacobiano visto che le lega le velocità di due terne differenti:

$$\begin{matrix} \dot{p}_2 = \dot{p}_1 - S(r_{12})\omega_1 \\ \omega_2 = \omega_1 \end{matrix} \quad \rightarrow \quad \begin{bmatrix} \dot{p}_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} I & -S(r_{12}) \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{p}_1 \\ \omega_1 \end{bmatrix}$$

Successivamente possiamo pensare, invece di esprimere tutte queste grandezze rispetto alla terna di riferimento, le possiamo definire rispetto alle rispettive terne. In tal caso, compariranno delle matrici di rotazione:

$$r_{12} = R_1 r_{12}^1$$

$$\begin{matrix} \dot{p}_1 = R_1 \dot{p}_1^1 \\ \omega_1 = R_1 \omega_1^1 \end{matrix} \qquad \begin{matrix} \dot{p}_2 = R_2 \dot{p}_2^2 = R_1 R_2^1 \dot{p}_2^2 \\ \omega_2 = R_2 \omega_2^2 = R_1 R_2^1 \omega_2^2 \end{matrix}$$

Risciviamo la forma matriciale prima definita:

$$\begin{bmatrix} \dot{p}_2^2 \\ \omega_2^2 \end{bmatrix} = \begin{bmatrix} R_1^2 & -R_1^2 S(r_{12}^1) \\ 0 & R_1^2 \end{bmatrix} \begin{bmatrix} \dot{p}_1^1 \\ \omega_1^1 \end{bmatrix}$$

Allora la relazione che lega le velocità di due terne differenti, definite rispetto a queste ultime è la seguente: $\boldsymbol{v_2^2 = J_1^2 v_1^1}$

In virtù della dualità cineto-statica riusciamo a trovare poi la relazione che lega velocità e forze di due terne differenti. E' da notare come proprio per il principio di dualità in una relazione abbiamo a primo membro grandezze relative alla terna 2 mentre a secondo membro della terna 1, mentre è il viceversa nell'altra relazione.

$$\boldsymbol{\gamma_1^1 = J_1^{2^T} \gamma_2^2} \quad \rightarrow \quad \begin{bmatrix} f_1^1 \\ \mu_1^1 \end{bmatrix} = \begin{bmatrix} R_2^1 & 0 \\ S(r_{12}^1)R_2^1 & R_2^1 \end{bmatrix} \begin{bmatrix} f_2^2 \\ \mu_2^2 \end{bmatrix}$$

2.8.2 Dualità per Catene Chiuse

Ricordiamo che per le catena cinematiche chiuse consideravamo una catena equivalente aperta in corrispondenza del giunto di taglio, ovvero uno dei possibili giunti non attuati. Allora avremo un nuovo vettore delle variabili di giunto q_{open} costituito da un insieme di variabili di giunto indipendenti q_a e un insieme di variabili non attuate q_u.

$$q_{open} = \begin{bmatrix} q_a \\ q_u \end{bmatrix}$$

Grazie al principio di dualità siamo in grado di ricavare la relazione che lega le q_u con le q_a, ovvero:

$$q_u = q_u(q_a)$$

Risolvendo i vincoli $q_u = q_u(q_a)$ calcoliamo:

$$\dot{q}_{open} = \Upsilon \dot{q}_a = \begin{bmatrix} I \\ \dfrac{\partial q_u}{\partial q_a} \end{bmatrix} \dot{q}_a$$

Per la dualità cineto-statica ricaviamo la relazione:

$$\boldsymbol{\tau_a = \Upsilon^T \tau_{open}}$$

Esempio: applichiamo quanto detto per le catena cinematiche chiuse al caso di un manipolatore a parallelogramma. Dalla cinematica abbiamo stabilito che le variabili attuate q_a sono $\theta_{1'}$ e $\theta_{1''}$, mentre le variabili non attuate q_u sono θ_2'' e θ_3''.

$$q_a = \begin{bmatrix} \theta_{1'} \\ \theta_{1''} \end{bmatrix} \qquad q_u = \begin{bmatrix} \theta_{2'} \\ \theta_{3'} \end{bmatrix}$$

Inoltre, sempre dalla cinematica, le relazioni che legano le q_u alle q_a in questo caso sono:

- $\theta_{2'} = \theta_{1''} - \theta_{1'}$
- $\theta_{3'} = \pi - \theta_{2'} = \pi - \theta_{1''} + \theta_{1'}$

Allora riprendendo l'espressione dello Jacobiano Υ, abbiamo un primo blocco che è la matrice identità 2×2 e un secondo blocco che è la derivata di q_u rispetto a q_a:

$$\Upsilon = \begin{bmatrix} I \\ \dfrac{\partial q_u}{\partial q_a} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \rightarrow \quad \tau_a = \Upsilon^T \tau_{open} = \begin{bmatrix} \tau_{1'} - \tau_{2'} + \tau_{3'} \\ \tau_{1''} + \tau_{2'} - \tau_{3'} \end{bmatrix}$$

Per quanto riguarda invece τ_u, essendo riferita a giunti non attuati sarà $\tau_u = [0 \quad 0]^T$.

2.9 Ellissoidi di Manipolabilità

L'**Ellissoide di Manipolabilità** è una superficie grazie alla quale si riesce ad avere una rappresentazione geometrica della capacità dell'end-effector del manipolatore (in generale si parla di un punto in una catena cinematica) di generare velocità e forze nello spazio. Si tratta di uno strumento che discende da tutta la teoria sulla cinematica differenziale, statica e dualità cineto-statica con il quale, in sede di progettazione, è possibile valutare la capacità del manipolatore di compiere un certo task in una determinata configurazione. Considerando per semplicità un manipolatore ridondante ($n > r$) e non singolare (J a rango pieno), ci sono due tipi di ellissoidi:

- **Ellissoide di Manipolabilità in Velocità**

E' l'insieme di tutte le possibili velocità all'organo terminale esprimibili per una data configurazione del robot.

Supponiamo che lo spazio delle velocità ai giunti sia una sfera di raggio costante unitario la cui equazione è: $\dot{q}^T\dot{q} = 1$

Sapendo che per un manipolatore ridondante la velocità ai giunti è: $\dot{q} = J^+(q)v_e$

Sostituendo nella precedente relazione, riusciamo a ricavare l'espressione dell'ellisoide di manipolabilità in velocità:

$$\dot{q}^T\dot{q} = [J^+(q)v_e]^T J^+(q)v_e \quad \rightarrow \quad \boldsymbol{v_e^T\left(J(q)J^T(q)\right)^{-1}v_e = 1}$$

A questo punto il problema è che stiamo ragionando su insiemi di grandezze di significato differente, infatti abbiamo velocità lineari in m/s e velocità angolari in rad/s. Se però prendiamo la matrice JJ^T posso eseguire una composizione in valori singolari e da ciò scopriamo che:

- Gli assi dell'ellissoide sono gli autovettori u_i di JJ^T;
- La dimensione degli assi sono i valori singolari $\sigma_i = \sqrt{\lambda_i(JJ^T)}$ di JJ^T, i quali sono la radice quadrata degli autovalori λ_i di JJ^T.

Inoltre, il volume dell'ellissoide è *proporzionale* all'indice di manipolabilità $w(q) = \sqrt{\det(J(q)J^T(q))}$. Per concludere, il significato dell'ellissoide è il seguente, la direzione dell'asse maggiore indica la direzione in cui l'organo terminale può muoversi a velocità maggiore e viceversa per l'asse minore. Da ciò ne deriva che tanto più l'ellissoide assomiglia ad una sfera e tanto meglio il manipolatore potrà muoversi velocemente in ogni direzione.

L'**Ellissoide di Manipolabilità in Forza** *è l'insieme di tutte le possibili wranch all'organo terminale equivalenti alle wranch nello spazio dei giunti.*

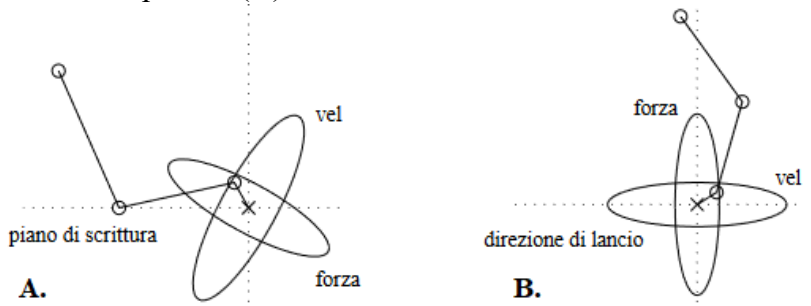
Supponiamo che lo spazio delle coppie ai giunti sia una sfera di raggio costante, la cui equazione è quindi: $\tau^T\tau = 1$

Sapendo che $\tau = J^T(q)\gamma_e$:

$$\tau = J^T(q)\gamma_e \quad \rightarrow \quad \gamma_e^T\left(J(q)J^T(q)\right)^{-1}\gamma_e = 1$$

Grazie alla dualità cineto-statica possiamo affermare che <u>la direzione lungo la quale si ha maggiore manipolabilità in velocità è anche la direzione con la minore manipolabilità in forza</u>

e viceversa. Per capire l'importanza di quanto detto supponiamo di avere un task di controllo come la scrittura su una superfice (A) e un task di attuazione come il lancio di un peso (B).



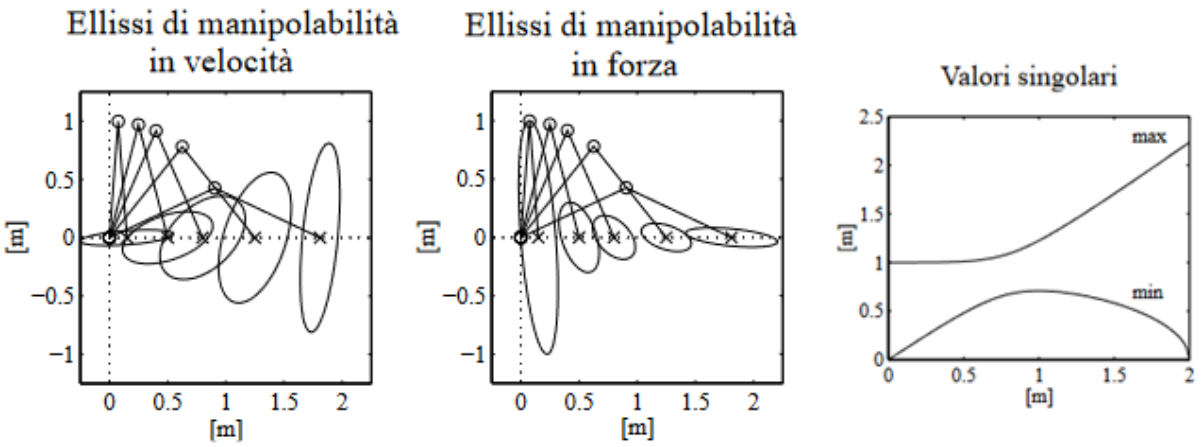
Il primo task, essendo di controllo, richiede di avere una piccola velocità e forza lungo la direzione orizzontale affinchè venga eseguito correttamente. Per cui il manipolatore dovrà posizionarsi in una configurazione tale che gli ellissoidi di velocità e forza siano disposti in modo tale che lungo al direzione orizzontale entrambe le grandezze sia molto piccole. Discorso diverso circa il secondo task, per il quale vogliamo la massima forza verticalmente e la massima velocità orizzontalmente. Per cui gli ellissoidi saranno disposti appunto come nella figura B.

Esempio: consideriamo un manipolatore planare a due bracci. In tal caso la misura di manipolabilità la prendiamo come: $w(q) = |\det(J)| = a_1 a_2 |s_2|$

Allora per questo indice di qualità abbiamo due massimi in corrispondendo di $\theta_2 = \pm\pi/2$, ovvero quando i due bracci sono a 90° fra loro, e quando $a_1 = a_2$. Come possiamo osservare nel grafico di sinistra:

a) Se il braccio è quasi tutto steso, allora l'ellissoide è verticale e dunque il robot avrà una maggiore manipolabilità lungo l'asse maggiore verticale dell'ellisse;
b) Se il braccio è a 90°, allora avremo quasi una circonferenza e dunque avremo la stessa manipolabilità lungo tutte le direzioni;
c) Se il braccio è tutto ritratto, allora l'ellissoide è orizzontale e dunque il robot avrà una maggiore manipolabilità lungo l'asse maggiore orizzontale dell'ellisse.

Come possiamo osservare, per l'ellissoide di manipolabilità in forza abbiamo praticamente un grafico speculare al precedente.



Ora, grazie alla matrice Jacobiano, al principio dei lavori virtuali e alla dualità cinetostatica scopriamo che un **manipolatore può essere visto come un trasformatore meccanico di**

<u>velocità e forza dallo spazio dei giunti allo spazio operativo</u>. Parlando di trasformatore chiaramente possiamo definire dei rapporti di trasformazione:

- **Rapporto di Trasformazione Lungo una Direzione per l'Ellisoide in Forza**:

$$\boldsymbol{a}(\boldsymbol{q}) = (\boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})\boldsymbol{u})^{-1/2}$$

- **Rapporto di Trasformazione Lungo una Direzione per l'Ellisoide in Velocità:**

$$\boldsymbol{a}(\boldsymbol{q}) = \left(\boldsymbol{u}^{\boldsymbol{T}}\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})\right)^{-1}\boldsymbol{u}\right)^{-1/2}$$

3. Pianificazione delle Traiettorie

3.1 Definizione di Pianificazione e Caratteristiche

La **Pianificazione delle Traiettorie** è la generazione degli ingressi di riferimento per il sistema di controllo del moto. Prima di analizzare del dettaglio questo argomento, è doveroso fare una precisazione sui termini "percorso" e "traiettoria", i quali spesso vengono utilizzati come sinonimi ma, nel campo della robotica, presentano una sostanziale differenza:

- **Percorso**: è il luogo delle pose dell'organo terminale che il robot deve esprimere nello spazio operativo. Da ciò si capisce che il termine percorso ha una connotazione puramente geometrica;
- **Traiettoria**: è un percorso a cui è applicata una *legge oraria del moto*, quindi vengono messi in campo velocità e/o accellerazione. Una traiettoria, perciò, ha un significato sia geometrico che fisico.

Una traiettoria può essere definita sia nello spazio dei giunti che nello spazio operativo, anche se tipicamente viene specificato in quest'ultimo per due motivi:

- ➢ Nativamente il task da eseguire è definito nello spazio operativo;
- ➢ E' più semplice imporre all'end-effector del manipolatore di evitare certe regioni dello spazio in cui possono essere presenti ostacoli, operatori umani, ecc.

Di contro, pianificare nello spazio operativo ci costringe ad adottare algoritmi di inversione cinematica che prendono i riferimento nello spazio operativo e li invertono in riferimenti nello spazio dei giunti. Viceversa i vantaggi della pianificazione nello spazio dei giunti è l'assenza della parte di cinematica inversa oltre che una migliore gestione dei gradi ridondanti e delle singolarità. Tuttavia, lo svantaggio principale di questo approccio è la difficoltà nella pianificazione (per esempio se abbiamo una traiettoria molto varia geometricamente e in termini di velocità/accellerazioni).

A prescindere dall'approccio adottato, un algoritmo di pianificazione ha:

- **Ingresso**: la definzione del percorso e dei relativi vincoli su quest'ultimo. Inoltre, vengono forniti anche i vincoli sulla dinamica del manipolatore. In generale, per la pianificazione vengono utilizzati un numero ristretto di parametri sul percorso e sulla legge del moto:
  - ❖ **Percorso**: punto iniziale, finale, punti intermedi, primitive geometriche (linee, circonfonferenze);
  - ❖ **Legge del moto**: tempo complessivo di esecuzione, velocità/accellerazioni massime, velocità/accellerazioni desiderate nei punti assegnati.
- **Uscita**: le traiettorie dei giunti o dell'organo terminale espresse come sequenza temporale dei valori di posizione, velocità e accellerazione.

Ultimo aspetto generale per la pianificazione è che essa deve presentare tre fondamenti caratteristiche:

- ✓ La pianificazione deve essere poco onerosa dal punto di vista computazionale. Vedremo inseguito l'utilizzo di polinomi interpolanti dei diversi punti della traiettoria, i quali proprio per questa proprietà non superano mai l'ordine tre;
- ✓ Almeno posizione e velocità devono essere funzioni continue del tempo;
- ✓ Minimizzazione degli effetti indesiderati come traiettorie a curvatura non regolare.

3.2 Traiettorie nello Spazio dei Giunti

La **Pianificazione delle Traiettorie nello Spazio dei Giunti** *consiste nella generazione di una funzione* $q(t)$ *che, nel rispetto dei vincoli imposti, interpola i valori assegnati per le variabili di giunto*. In questo tipo di pianificazione possiamo avere due principali tipologie di moto:

A. Moto Punto-Punto

Vengono forniti solo gli estremi della traiettoria e il tempo complessivo per passare dal primo al secondo. Supponendo per semplicità di considerare una sola variabile di giunto, il relativo moto punto-punto come detto poc'anzi richiede la specifica della posizione iniziale q_i, quella finale q_f e il tempo di transizione t_f. Il problema così formulato presenta però infinite soluzioni, visto che una traiettoria tra due estremi può esprimersi in infiniti modi. La soluzione perciò viene trovata riconducendosi ad un problema di ottimizzazione. In particolare, si tratterò di problema di minimizzazione del seguente funzionale di costo, il quale rappresenta l'energia dissipata dall'attuatore che esplica una coppia τ nella movimentazione del link. La coppia τ è data dal prodotto del momento d'inerzia I e l'accellerazione $\dot{\omega}$: $\tau = I\dot{\omega}$

Inoltre, tale problema è soggetto al vincolo indicato di seguito:

$$\min \int_0^{t_f} \tau^2(t)dt \qquad Vincolo: \quad \int_0^{t_f} \omega dt = q_f - q_i$$

La soluzione di questo problema è un polinomio quadratico, il quale esprime la traiettoria della variabile di giunto: $at^2 + bt + c = q(t)$

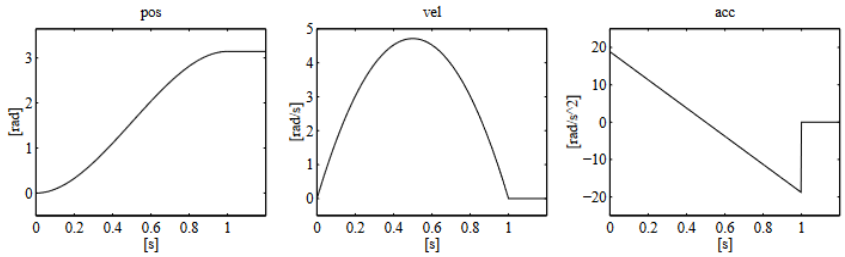
Il problema però di questo polinomio è che ci consente di agire solo sulla posizione. Perciò tipicamente si preferisce impiegare (non solo per lo spazio dei giunti ma anche quello operativo) una polinomiale cubica $q(t)$ come di fianco.

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$
$$\dot{q}(t) = 3a_3t^2 + 2a_2t + a_1$$
$$\ddot{q}(t) = 6a_3t + 2a_2$$

Calcoliamo ora i coefficienti:

- Posizione iniziale: $q_i = a_0$ se $t_i = 0$;
- Velocità iniziale $\dot{q}_i = a_1$
- Posizione finale: $q_f = a_3t_f^3 + a_2t_f^2 + a_1t_f + a_0$
- Velocità finale: $\dot{q}_f = 3a_3t_f^2 + 2a_2t_f + a_1$

Vediamo graficamente cosa accade:

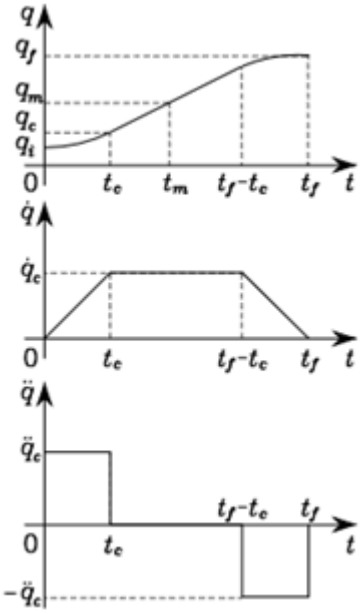


Se volessimo imporre dei vincoli anche all'accellerazione dovremmo adottare un polinomio quintico: $q(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$

Ricordando che i dati da dare in ingresso al pianificatore sono q_i, q_f, t_f:

a) Se il t_f scelto è adatto alla pianificazione, allora quest'ultima verrà eseguita senza problemi;
b) Se il t_f scelto è troppo piccolo, allora il manipolatore va in blocco.

Allora l'utilizzo di una polinomiale non ci permette di avere una verifica diretta se la velocità, ed eventualemente l'accellerazione, sono compatibili con il manipolatore fisico. Motivo per cui nella pratica industriale viene spesso adottato un approccio alternativo che consiste nell'assegnazione di un particolare profilo di velocità. Il **Profilo di Velocità Trapezoidale** è costituito da tre andamenti:



1. **Fase di Accellerazione Costante**: nell'intervallo $[0, t_c]$ abbiamo una fase di moto uniformemente accellerato caratterizzato da un valore di accellerazione costante $\ddot{q}_c$. La velocità in questa fase cresce in maniera lineare passando da 0 a $\dot{q}_c$. La posizione infine passa con andamento parabolico da q_i a q_c;
2. **Fase di Accellerazione Nulla**: nell'intervallo $[t_c, t_f - t_c]$ l'accelerazione risulta essere nulla. La velocità in questa fase è costante ed è pari a $\dot{q}_c$. La posizione infine varia in modo in modo lineare;
3. **Fase di Decellerazione Costante**: nell'intervallo $[t_f - t_c, t_f]$ abbiamo una fase di moto uniformemente decellerato caratterizzato da un valore di decellerazione costante $-\ddot{q}_c$. La velocità in questa fase decresce in maniera lineare passando da $\dot{q}_c$ a 0. La posizione infine varia con andamento parabolico fino al valore finale q_f.

L'utilizzo del profilo di velocità trapezzoidale ha due fondamentali vantaggi:

- ✓ Garantisce la continuità della velocità;
- ✓ Ci consente di capire immediatamente se la traiettoria con i tre dati forniti è fattibile.

Un altro aspetto che possiamo valutare è *l'imposizione di una accellerazione* $\ddot{q}$ che sia congruente con il profilo scelto e i dati in ingresso. Per fare ciò sfruttiamo la proprietà di continuità della velocità per quale in t_c la velocità alla fine della prima rampa è uguale alla velocità all'inizio del tratto costante (relazione di fianco). Ora, il valore di q_c all'istante t_c è dato dalla seguente equazione di moto uniformemente accellerato. Sostituendo nella precedente relazione, otteniamo l'equazione differenziale di secondo grado in t_c:

$$\ddot{q}_c t_c = \frac{q_m - q_c}{t_m - t_c}$$

$$q_c = q_i + \frac{1}{2}\ddot{q}_c t_c^2$$

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

Da questa equazione ricaviamo il valore di t_c, il quale ha senso se il radicando è ≥ 0 e da ciò otteniamo la condizione che ci interessa per l'accellerazione:

$$t_c = \frac{t_f}{2} - \frac{1}{2}\sqrt{\frac{t_f^2\ddot{q}_c - 4(q_f - q_i)}{\ddot{q}_c}} \quad \rightarrow \quad \frac{t_f^2\ddot{q}_c - 4(q_f - q_i)}{\ddot{q}_c} \geq 0 \quad \rightarrow \quad |\ddot{\boldsymbol{q}}_{\boldsymbol{c}}| \geq \frac{\boldsymbol{4}|\boldsymbol{q_f} - \boldsymbol{q_i}|}{\boldsymbol{t_f^2}}$$

In definitiva, scegliano una accellerazione $\ddot{q}_c$ che rispetta questa condizione, saremo sicuri che la legge oraria rispetta i vincoli del manipolatore e in questo caso risulterà essere:

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

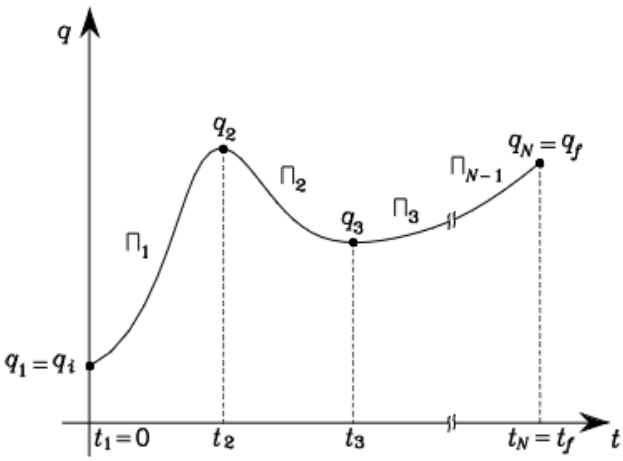
B. Moto con Sequenza di Punti

Vengono forniti, oltre agli estremi della traiettoria, anche una sequenza di punti intermedi e il tempo di transizione. In generale, se abbiamo N punti in cui il robot deve transitare, allora sarà necessario un polinomio di grado al più $N-1$. Se vogliamo dare dei vincoli alla velocità iniziale e finale, allora il polinomio sarà di grado $N-1+2$. Gli svantaggi di avere un numero grande di punti di attraversamento sono molteplici:

- Aumento della complessità computazionale;
- Aumento del carattere oscillatorio del sistema;
- Degenerazione della precisione numerica;
- I coefficienti del polinomio dipendono da tutti i punti assegnati. Perciò cambiare alcuni punti del percorso vuol dire dover ricalcolare tutto.

Per risolvere questi problemi si impiegano sequenze di polinomi di grado più basso dove ciascuno di essi impone dei vincoli di posizione e velocità per ciascun punto di attraversamento. Perciò, ora la $q(t)$ sarà un insieme di polinomi tipicamente cubici: $\boldsymbol{q(t) = \Pi_k(t_k)}$

Graficamente abbiamo la situazione mostrata in figura, dove per esempio il polinomio Π_1 collegata $q_1 = q_i$ a q_2 e così via fino a Π_{N-1} che collega q_{N-1} a $q_N = q_f$.

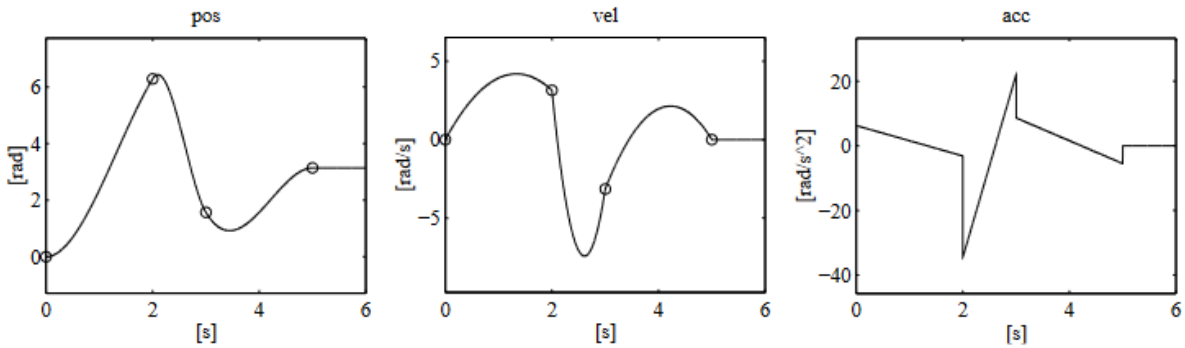


A questo punto, l'approccio con sequenza di polinomi presenta diverse strade:

a. Vincoli Imposti sulle Velocità nei Punti di Percorso: in pratica consiste nell'imporre i vincoli visti in precedenza per ciascun polinomio cubico, ottenendo così $N-1$ sistemi di quattro equazioni in quattro incognite che possono essere risolti indipendenemente:

$$\begin{array}{l} \textit{Vincoli di Posizione}: \\ \textit{Vincoli di Continuità delle Posizioni}: \\ \textit{Vincoli sulle Velocità}: \\ \textit{Vincoli di Continuità delle Velocità}: \end{array} \quad \begin{cases} \Pi_k(t_k) = q_k \\ \Pi_k(t_{k+1}) = q_{k+1} \\ \dot{\Pi}_k(t_k) = \dot{q}_k \\ \dot{\Pi}_k(t_{k+1}) = \dot{q}_{k+1} = \dot{\Pi}_{k+1}(t_{k+1}) \end{cases}$$

Allora, immaginando di avere il percorso in figura (immagine pos), grazie a questo metodo siamo sicuri di garantire la continuità della posizione e della velocità ma non chiaramente dell'accellerazione.



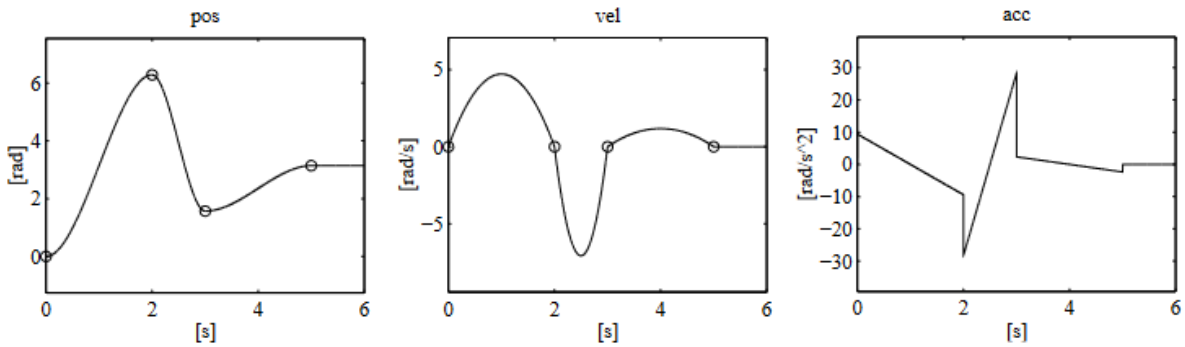
b. Velocità Calcolate nei Punti di Percorso: invece di imporre le velocità le andiamo a calcolare nei *punti di percorso*. In tal caso sarà necessario specificare delle primitive di attraversamento per tali punti, quindi se il robot va da un punto ad un altro mediante un segmento, un arco di circonferenza, ecc. Allora, supponendo di avere il percorso in figura (grafico pos), dove le primitive di collegamento tra i diversi punti del percorso sono le rette e indichiamo con $v_k = (q_k - q_{k-1})/(t_k - t_{k-1})$ la loro inclinazione. Possiamo imporre velocità iniziale $\dot{q}_1$ e finale $\dot{q}_N$ nulle, mentre nei punti intermedi $\dot{q}_k$ li calcoliamo come di seguito:

$$\dot{q}_1 = 0$$

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

$$\dot{q}_N = 0$$

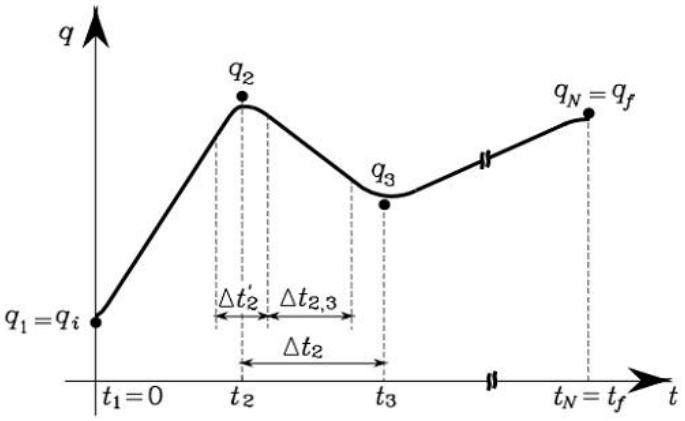
Risultato di ciò in tutti i punti intermedi la velocità risulterà essere nulla mentre fra di essi avrà il valore calcolato mediante la relazione di sopra.



c. Interpolazione di Polinomi Parabolico-Lineari: qualora il nostro obiettivo primario fosse la continuità della derivata prima negli istanti temporali t_k allora possiamo pensare di adottare sequenze di polinomi parabolico-lineari. L'idea di base è cercare di far passare la variabile di giunto solo *nell'intorno di q_k* anziché per il suddetto valore. A tal proposito definiamo:

- Δt_k = intervallo di tempo che separa il valore q_k da q_{k+1};
- $\Delta t'_k$ = intervallo tempo che stabilisce quando avere una traiettoria parabolica;
- $\Delta t_{k,k+1}$ = intervallo di tempo che stabilisce quando avere una traiettoria lineare.

Avremo perciò la situazione descritto in figura in cui tra un punto q_k e il successivo avremo dei tratti lineari, mentre nel loro intorno avremo dei tratti parabolici.



Grazie a ciò, oltre a garantire la continuità della derivata prima stiamo apportando un notevole risparmio computazionale. Qualora vogliamo che la variabile q passi molto vicino a q_k, dobbiamo ridurre l'intervallo $\Delta t'_k$ e ciò vuol dire aumentare l'accellerazione. Come sappiamo però questo comporterà avere delle maggiori forze generalizzate agenti sul giunto. E' interessante notare che, assegnati i q_k, $\Delta t'_k$, Δt_k, il profilo di velocità che si ottiene con questo approccio è di fatto trapezzoidale.

d. Spline: tale soluzione viene adottata quando si desidera garantire la continuità anche delle accellerazioni, cosa che non accade nei due casi precedenti. Ciò si ottiene imponendo i seguenti vincoli:

$$\begin{array}{r} \textit{Vincoli di Posizione}: \\ \textit{Vincoli di Continuità delle Posizioni}: \\ \textit{Vincoli di Continuità delle Velocità}: \\ \textit{Vincoli di Continuità delle Accellerazioni}: \end{array} \quad \begin{cases} \Pi_{k-1}(t_k) = q_k \\ \Pi_{k-1}(t_k) = \Pi_k(t_k) \\ \dot{\Pi}_{k-1}(t_k) = \dot{\Pi}_k(t_k) \\ \ddot{\Pi}_{k-1}(t_k) = \ddot{\Pi}_k(t_k) \end{cases}$$

Tuttavia, il sistema risultante che include anche i punti iniziale e finale non è risolvibile in quanto presenta $4N - 2$ equazioni in $4(N - 1)$ coefficienti incogniti. Allora, per rendere il sistema risolvibile, potremmo pensare in prima battuta di eliminare le due equazioni cubiche che consentono l'assegnazione arbitraria dei valori di accellerazioni iniziale e finale per sostituirle con un polinomio quarto per i punti estremi della traiettoria (proprio per poter assegnare tali accellerazioni). In realtà, è possibile impiegare il seguente artificio per poter impiegare solo ed esclusivamente polinomi cubici. Si introducono dei punti virtuali sui quali imporre vincoli di continuità su posizione,velocità e accellerazione ma senza specificare le loro posizioni effettive. Grazie all'introduzione di questi due punti aggiuntivi avremo un totale di $N + 1$ polinomi cubici. Allora considerando $N + 2$ istanti di tempo t_k dove, per convenzione, t_2 e t_{N+1} sono associati ai punti virtuali, il sistema di equazioni risultante avrà $4(N + 1)$ equazioni per la determinazione $4(N + 1)$ coefficienti incogniti degli $N + 1$ polinomi cubici. In particolare, avremo:

- $4(N - 2)$ equazioni per $N - 2$ punti di percorso interni.

$$\begin{array}{r} \textit{Vincoli di Posizione}: \\ \textit{Vincoli di Continuità delle Posizioni}: \\ \textit{Vincoli di Continuità delle Velocità}: \\ \textit{Vincoli di Continuità delle Accellerazioni}: \end{array} \quad \begin{cases} \Pi_{k-1}(t_k) = q_k \\ \Pi_{k-1}(t_k) = \Pi_k(t_k) \\ \dot{\Pi}_{k-1}(t_k) = \dot{\Pi}_k(t_k) \\ \ddot{\Pi}_{k-1}(t_k) = \ddot{\Pi}_k(t_k) \end{cases}$$

- 6 equazioni per i punti iniziale e finale.

$$\begin{array}{ll} \Pi_1(t_1) = q_i & \Pi_{N+1}(t_{N+2}) = q_f \\ \dot{\Pi}_1(t_1) = \dot{q}_i & \dot{\Pi}_{N+1}(t_{N+2}) = \dot{q}_f \\ \ddot{\Pi}_1(t_1) = \ddot{q}_i & \ddot{\Pi}_{N+1}(t_{N+2}) = \ddot{q}_f \end{array}$$

- 6 equazioni per i punti virtuali.

$$\begin{array}{r} \textit{Vincoli di Continuità delle Posizioni}: \\ \textit{Vincoli di Continuità delle Velocità}: \\ \textit{Vincoli di Continuità delle Accellerazioni}: \end{array} \quad \begin{cases} \Pi_{k-1}(t_k) = \Pi_k(t_k) \\ \dot{\Pi}_{k-1}(t_k) = \dot{\Pi}_k(t_k) \\ \ddot{\Pi}_{k-1}(t_k) = \ddot{\Pi}_k(t_k) \end{cases}$$

La soluzione del sistema, anche in questi termini, è computazionalmente impegnativa anche per bassi valori di N.

Motivo per cui è conveniente riformulare il problema sfruttando il fatto che i polinomi sono cubici e dunque che la derivata seconda $\ddot{\Pi}_k(t)$ è funzione lineare del tempo. Quest'ultima può essere scritta come segue in cui $\Delta t_k = t_{k+1} - t_k$ è l'intervallo di tempo per andare da q_k a q_{k+1}:

$$\ddot{\Pi}_k(t) = \frac{\ddot{\Pi}_k(t_k)}{\Delta t_k}(t_{k+1} - t) + \frac{\ddot{\Pi}_k(t_{k+1})}{\Delta t_k}(t - t_k)$$

Derivando tue volte ricaviamo i polinomi cubici $\Pi_k(t_k)$, i quali sono funzioni di quattro incognite: $$\Pi_k(t) = f\left(\Pi_k(t_k), \Pi_k(t_{k+1}), \ddot{\Pi}_k(t_k), \ddot{\Pi}_k(t_{k+1})\right)$$

In realtà, andando a studiare i vincoli prima citati, possiamo semplificare ulteriormente:

- N variabili $q_k \neq 2, N+1$ sono assegnate, quindi tutti i punti della traiettoria fatta eccezione per quelli virtuali sono assegnati;
- Imponiamo vincoli di continuità sia sui punti virtuali q_2 e q_{N+1}, che sui punti di cammino $q_3, \dots, q_N$;
- Le velocità iniziali $\dot{q}_i$ e finali $\dot{q}_f$ così come le accellerazioni iniziali $\ddot{q}_i$ e finali $\ddot{q}_f$ sono assegnate;
- Vincoli di continuità per le accellerazioni su $\ddot{q}_k$ per $k = 2, \dots, N+2$.

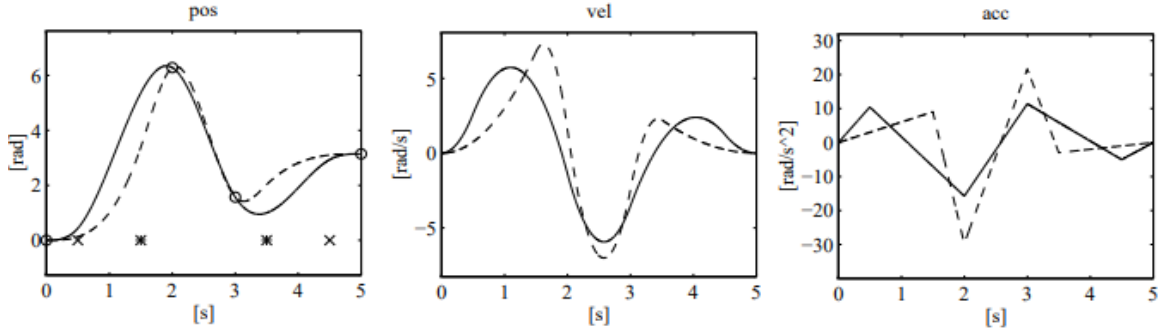
Grazie a tutte queste considerazioni abbiamo ridotto il problema ad uno con N equazioni in N incognite. Tale sistema lineare può essere espresso nella seguente forma dove la matrice A contiene i coefficienti dei polinomi che ha una struttura a fasce diagonale a blocchi, mentre il vettore b è costituito dai termini noti dei polinomi cubici:

$$\dot{\Pi}_1(t_2) = \dot{\Pi}_2(t_2)$$
$$\vdots$$
$$\dot{\Pi}_N(t_{N+1}) = \dot{\Pi}_{N+1}(t_{N+1})$$

$$A\begin{bmatrix} \ddot{\Pi}_2(t_2) \\ \vdots \\ \ddot{\Pi}_{N+1}(t_{N+1}) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{11} & \dots & 0 & 0 \\ a_{11} & a_{11} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{N-1,N-1} & a_{N-1,N} \\ 0 & 0 & \dots & a_{N,N-1} & a_{N,N} \end{bmatrix} \begin{bmatrix} \ddot{\Pi}_2(t_2) \\ \vdots \\ \ddot{\Pi}_{N+1}(t_{N+1}) \end{bmatrix} = b$$

Mediante questa struttura si può impiegare un metodo risolutivo che viene definito Avanti-all'Indietro, cioè si risolvono tutte le equazioni rispetto a solo due incognite. Per esempio,

4. Nella prima equazione si esprime $\ddot{\Pi}_2(t_2)$ in funzione di $\ddot{\Pi}_3(t_3)$;
5. Si sostituisce nella seconda equazione, la quale ora avrà come incognite solo $\ddot{\Pi}_3(t_3)$ e $\ddot{\Pi}_4(t_3)$. In maniera analoga al passo 1, scriviamo $\ddot{\Pi}_3(t_3)$ in funzione $\ddot{\Pi}_4(t_3)$. Infine, sostituiamo nella terza equazione;
6. Ripetiamo i passi 1 e 2 per tutte le equazioni fatta eccezione per l'ultima, la quale presenterà come unica incognita $\ddot{\Pi}_{N+1}(t_{N+1})$ che possiamo calcolare;
7. A questo punto si procede a ritroso fino a calcolare $\ddot{\Pi}_2(t_2)$.



3.3 Traiettorie nello Spazio Operativo

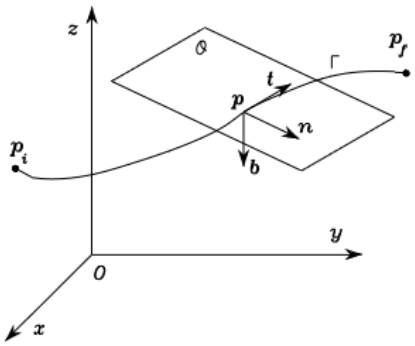
3.3.1 Definizione e Approccio

La **Pianificazione delle Traiettorie nello Spazio Operativo** *consiste nel calcolo del vettore x_e, il quale rappresenta la posa (posizione e orientamento) dell'organo terminale* e la cui dimensione è strettamente legata al tipo di task. Il vantaggio principale di questa pianifiazione è che il compito da eseguire è nativamente definito nello spazio operativo ed è quindi più intuitivo capire che traiettoria il robot deve compiere. La pianificazione dello spazio operativo può essere realizzata mediante differenti approcci, tuttavia alla base c'è una problematica dovuta alla successiva presenza del controllore del robot. Quest'ultimo, per calcolare il segnale errore necessario per il controlo, necessità di riferimenti in ingresso definito nello spazio dei giunti. Perciò, a valle della pianificazione sarà necessaria una inversione cinematica che chiaramente ha un certo costo computazionale. Risolto il problema della natura dei riferimenti in ingresso al controllore ne sovraggiunge un secondo, ovvero la loro continuità di aggiornamento con cui esse vengono forniti a quest'ultimo. Infatti, la la complessità computazionale indotta dai blocchi di pianificazione e inversione cinematica pone un limite superiore alla frequenza di campionamento per generare le sequenze nello spazio dei giunti. A risoluzione di questo problema, a valle dei due blocchi citati viene eseguita la **Microinterpolazione Lineare**, ovvero una tecnica che consente di aumentare la frequenza di campionamento e migliorare così le prestazioni dinamiche.

Descritto a livello generale cosa genera la pianificazione nello spazio operativo e le relative problematiche, andiamo ad analizzare i due approcci citati inizialmente:

A. **Polinomi Interpolanti**: interpoliamo dei punti di una traiettoria mediante dei polinomi tipicamente cubici. Si tratta quindi della stessa tecnica impiegata per lo spazio dei giunti con la differenza però che, in questo caso, i polinomi sono definiti nello spazio operativo ed inoltre essi non interpolano delle configurazioni del robot ma bensì delle pose dell'end-effector. E' un approccio utilizzabile, per esempio, quando non è di particolare interesse ciò che accade tra una posa e l'altra;

B. **Primitive di Moto**: una primitiva di moto è la legge esatta con cui l'organo terminale si muove tra una posa e la successiva. Per descrivere delle leggi di moto su primitive di percorso è necessario impiegare delle *rappresentazioni parametriche* della curva Γ nello spazio del tipo: $p = f(s)$



Dunque, al valore del parametro s che appartiene ad un certo intervallo sarà associato un punto $p \in R^{3\times1}$ (vettore colonna) della curva Γ. In particolare, s è detta ascissa curvilinea, ovvero la lunghezza dell'arco che sottende la curva dal valore iniziale della curva al valore attuale. L'utilizzo di questa parametrizzazione ci consente anche di sapere il verso di percorrenza della curva Γ . Il punto p appena calcolato sulla curva individua una terna di versori (t, n, b) dove:

- t: è il versore tangente alla curva Γ e punta nella direzione di percorrenza di quest'ultima;
- n: è il versore normale a t e giacente sul cosidetto piano osculatore;
- b: completa la terna levogira e sarà quindi ortogonale ai precedenti versori.

Questi tre versori possono essere ricavati mediante le seguenti formule:

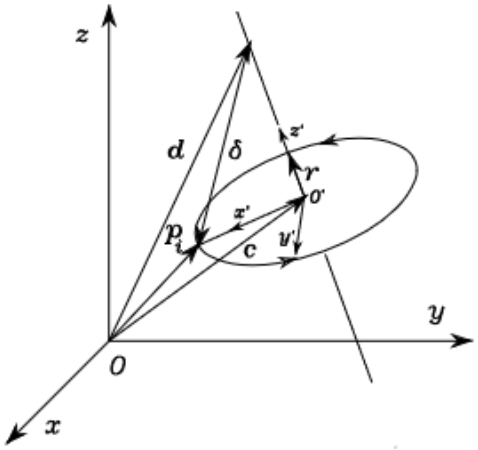
$$t = \frac{dp}{ds} \qquad n = \frac{1}{\left\|\frac{d^2p}{ds^2}\right\|}\frac{d^2p}{ds^2} \qquad b = t \times n$$

Le primitive di moto principali sono le seguenti:

a. **Segmento**: la parametrizzazione del segmento è dato dal valore iniziale, finale e l'ascisse curvilinea s che si muove su di esso. In particolare, l'ascissa curvilinea varierà seconda una legge cubica oppure parabolico-lineare analoga a quelle viste per la pianificazione delle traiettorie nello spazio dei giunti.

$$p(s) = p_i + \frac{s}{\|p_f - p_i\|}(p_f - p_i)$$
$$\frac{dp}{ps} = \frac{1}{\|p_f - p_i\|}(p_f - p_i)$$
$$\frac{d^2p}{ps^2} = 0$$

b. **Circonferenza**: prima di ricavare la rappresentazione parametrica della circonferenza è necessario definire i suoi parametri caratteristici. Supponiamo che la circonferenza sia specificata da:
   - Vettore unitario dell'asse del cerchio r (è orientato in figura come z');
   - Vettore di posizione d di un punto lungo l'asse del cerchio;
   - Vettore posizione di un punto iniziale p_i sulla circonferenza.



Ora, per evitare che la circonferenza degeneri in un punto, indicando con $\delta = p_i - d$, deve risultare che: $|\delta^T r| < \|\delta\|$

Grazie a questi parametri possiamo trovare il vettore di posizione c del centro della circonferenza. In questo caso: $c = d + (\delta^T r)r$

Si vuole ora trovare una rappresentazione parametrica del cerchio in funzione della lunghezza dell'arco. E' da specificare che questa rappresentazione risulta essere semplice se il sistema di riferimento $O'x'y'z'$ è opportunamento scelto. Infatti, se viene scelto in modo che O' coincidi con il centro della circonferenza abbiamo che la sua rappresentazioen parametrica è:

$$p'(s) = \begin{bmatrix} \rho\cos(s/\rho) \\ \rho\sin(s/\rho) \\ 0 \end{bmatrix} \qquad dove:\ raggio = \rho = \|p_i - c\|$$

Divesamente invece, avremo una $p(s)$ che è data dalla somma del vettore di posizione c della circonferenza e dal prodotto $Rp'(s)$, dove $R = [x'\ y'\ z']$ è la matrice di rotazione della terna $O'x'y'z'$ rispetto alla terna $Oxyz$:

$$p(s) = c + Rp'(s) = c + R\begin{bmatrix}\rho\cos(s/\rho)\\ \rho\sin(s/\rho)\\ 0\end{bmatrix}$$

$$\frac{dp}{ds} = R\begin{bmatrix}-\sin(s/\rho)\\ \cos(s/\rho)\\ 0\end{bmatrix}$$

$$\frac{d^2p}{ds^2} = R\begin{bmatrix}-\frac{1}{\rho}\cos(s/\rho)\\ -\frac{1}{\rho}\sin(s/\rho)\\ 0\end{bmatrix}$$

3.3.2 Calcolo di Posizione e Orientamento

Ricordiamo che lo spazio operativo è quello spazio in cui è definito il vettore x_e che esprime la posa dell'end-effector.

$$x_e = \begin{bmatrix} p \\ \phi \end{bmatrix} = \begin{bmatrix} posizione \\ orientamento \end{bmatrix}$$

In precedenza, abbiamo visto le metodologie per pianificare una traiettoria nello spazio operativo perciò ora dobbiamo studiare come generarla nel suddetto spazio. Sapendo che la posa dell'end-effector tiene conto sia la posizione che l'orientamento ed eseguire una traiettoria nello spazio operativo vuol dire portare l'end-effector da una certa posa iniziale ad una finale, dobbiamo analizzare posizione e orientamento separatamente:

- **Posizione**

Abbiamo che la traiettoria in termini di posizione dell'organo terminale è funzione dell'ascissa curvilinea s: $$p_e = f(s)$$

Ora, a seconda del tipo di traiettoria (punto-punto, serie di punti), avremo una differente espressione di p_e e di s che dovremo andare a definire:

A. Traiettoria di Posizione Punto-Punto: in questo caso indicando con:

- p_i = posizione iniziale dell'end-effector quando $s = 0$ per $t = 0$;
- p_f = posizione finale dell'end-effector quando $s = s_f$ per $t = t_f$.

L'espressione della posizione dell'end-effector in funzione dell'ascissa curvilinea è la seguente, dove quest'ultima può variare con una legge cubica oppure parabolico lineare:

Segmento:

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

Circonferenza:

$$\boldsymbol{p_e(s) = c + R\begin{bmatrix} \rho\cos(s/\rho) \\ \rho\sin(s/\rho) \\ 0 \end{bmatrix}}$$

Per completezza, andiamo anche a calcolare la velocità e l'accellerazione di p_e, le quali avranno differenti espressioni:

Segmento:

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

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

Circonferenza:

$$\dot{p}_e = R\begin{bmatrix} -\dot{s}\sin(s/\rho) \\ \dot{s}\cos(s/\rho) \\ 0 \end{bmatrix}$$

$$\ddot{p}_e = R\begin{bmatrix} -\dot{s}^2\cos(s/\rho)/\rho - \ddot{s}\sin(s/\rho) \\ -\dot{s}^2\sin(s/\rho)/\rho + \ddot{s}\cos(s/\rho) \\ 0 \end{bmatrix}$$

In realtà, entrambe le grandezze posso essere ricondotte ad una forma prodotto tra ampiezza del vettore relativa ad essa e il vettore t. Per esempio, prendendo la velocità (ma vale lo stesso discorso per l'accellerazione): $$\dot{p}_e = \dot{s}t$$

Dove $\dot{s}$ è l'ampiezza del vettore velocità nel punto p il cui segno positivo o negativo dipende dalla direzione di $\dot{p}_e$ lungo t.

B. Traiettoria di Posizione per Serie di Punti: consideriamo una traiettoria costituita da una serie di $N+1$ punti $p_0, p_1, \dots p_N$ connessi tra loro mediante N segmenti. L'espressione della posizione dell'end-effector in funzione dell'ascissa curvilinea è la seguente:

$$\boldsymbol{p_e = p_0 + \sum_{j=1}^{N} \frac{s_j}{\|p_j - p_{j-1}\|}(p_j - p_{j-1})}$$

Questa volta però, l'ascissa curvilinea varia con una legge più articolata del caso precedente. Infatti s_j, ovvero la lunghezza dell'arco relativo al $j-$esimo segmento che collega p_{j-1} e p_j, è definito come:

$$s_j(t) = \begin{cases} 0 & 0 \le t \le t_{j-1} \\ s_j'(t) & t_{j-1} < t < t_j \\ \|p_j - p_{j-1}\| & t_j \le t \le t_f \end{cases} \qquad s_j'(t) \begin{matrix} varia\ con\ legge\ cubica \\ o\ parabolico - lineare \end{matrix}$$

Se andiamo poi a calcolare velocità e accellerazione di p_e, i quali possono essere scritti con una formulazione analoga al caso precedente della traeittoria punto-punto:

$$\dot{p}_e = \sum_{j=1}^{N} \dot{s}_j t_j \qquad \ddot{p}_e = \sum_{j=1}^{N} \ddot{s}_j t_j$$

In questo scenario abbiamo però la problematica della discontinuità della derivata prima quando abbiamo segmenti non allineati. Questo si traduce in un arresto del manipolatore prima che esso possa proseguire nella direzione del segmento successivo. Per risolvere questo problema si può adottare la legge parabolico-lineare in modo da rilassare i vincoli di passaggi per i punti ed inoltre si modifica la $s_j(t)$ in modo che venga anticipata la generazione del segmento prima che il precedente venga completato:

$$s_j(t) = \begin{cases} 0 & 0 \le t \le t_{j-1} - \Delta t_j \\ s_j'(t)(t + \Delta t_j) & t_{j-1} - \Delta t_j < t < t_j - \Delta t_j \\ \|p_j - p_{j-1}\| & t_j - \Delta t_j \le t \le t_f - \Delta t_j \end{cases}$$

Dove Δt_j è l'anticipo sulla generazione del $j-$esimo segmento. Esso può essere valutato ricorsivamente come somma tra il complessivo degli anticipi di tutti i segmenti precedenti Δt_{j-1} e l'antipo δt_j sulla generazione del segmento corrente:

$$\Delta t_j = \Delta t_{j-1} + \delta t_j \quad j = 1, \dots, N \quad \Delta t_0 = 0$$

- **Orientamento**:

Sapendo che la matrice di rotazione esprime l'orientamento dell'end-effector, possiamo pensare di interpolare le seguenti matrici di rotazione per ottenere la parte di traiettoria relativa all'orientamento:

- R_i = matrice di rotazione iniziale relativa quindi ad una condizione iniziale $O_i x_i y_i z_i$ dell'orientamento della terna solidale all'end-effector;
- R_f = matrice di rotazione finale relativa quindi ad una condizione finale $O_f x_f y_f z_{ff}$ dell'orientamento della terna solidale all'end-effector.

In realtà, questa operazione non viene eseguita perché risulta essere complesso a causa della presenza dei vincoli di ortonormalità tra le colonne delle matrici di rotazione. Motivo per cui, sulla pianificazione della traiettoria per la parte di orientamento si adottano rappresentazioni alternative come:

A. Angoli di Eulero: rappresentiamo l'orientamento mediante gli angoli di Eulero in modo da definire il vettore ϕ_e:

$$\phi_e = (\rho, \theta, \psi)$$

A questo punto l'idea è interpolare ϕ_i e ϕ_f secondo una legge cubica o parabolica lineare che andremo ad inglobare nella variabile s da cui dipenderà ϕ_e (e le relative derivate). Pertanto la $\phi_e, \dot{\phi}_e, \ddot{\phi}_e$ saranno definite dai profili indicati di fianco. Inoltre, così facendo, la velocità angolare dell'end-effector sarà legata a $\dot{\phi}_e$ dalla seguente trasformazione:

$$\phi_e = \phi_i + \frac{s}{\|\phi_f - \phi_i\|}(\phi_f - \phi_i)$$

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

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

$$\omega_e = T(\phi_e)\dot{\phi}_e = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\theta \\ 0 & c_\varphi & s_\varphi s_\theta \\ 1 & 0 & c_\theta \end{bmatrix} \dot{\phi}_e$$

B. Asse Angolo: descriviamo una rotazione θ intorno ad un asse fisso r nello spazio operativo. Dunque partiamo da due terne $O_i - x_i y_i z_i$ e $O_f - x_f y_f z_f$ associate alle rispettivi matrici di rotazione R_i ed R_f. Seppur rappresentati orientamenti diversi, le due terne hanno le origini in comune e quindi non presentano una distanza relativa. Perciò la matrice di rotazione tra le due terne si può calcolare mediante la relazione di sinistra. Ricordando che $R_i^{-1} = R_i^T$ possiamo calcolare la R_f^i:

$$R_f = R_i R_f^i \quad \rightarrow \quad R_f^i = R_i^T R_f = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Ora se $R^i(t)$ definisce la transizione da R_i ad R_f, allora si deve verificare che $R^i(0) = I$ e $R^i(t_f) = R_f^i$. Perciò la matrice R_f^i possiamo esprimerla come rotazione attorno ad un asse fisso nello spazio. Il vettore unitario r^i dell'asse e l'angolo di rotazione θ_f possono essere calcolati utilizzando le relazioni viste per la rappresentazione asse-angolo:

$$\theta_f = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \qquad r^i = \frac{1}{2\sin\theta_f}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Ma allora la matrice R_f^i è di fatto la matrice $R^i(\theta(t), r^i)$ vista in tale rappresentazione.

$$\boldsymbol{R}(\vartheta, \boldsymbol{r}) = \begin{bmatrix} r_x^2(1-c_\vartheta)+c_\vartheta & r_x r_y(1-c_\vartheta)-r_z s_\vartheta & r_x r_z(1-c_\vartheta)+r_y s_\vartheta \\ r_x r_y(1-c_\vartheta)+r_z s_\vartheta & r_y^2(1-c_\vartheta)+c_\vartheta & r_y r_z(1-c_\vartheta)-r_x s_\vartheta \\ r_x r_z(1-c_\vartheta)-r_y s_\vartheta & r_y r_z(1-c_\vartheta)+r_x s_\vartheta & r_z^2(1-c_\vartheta)+c_\vartheta \end{bmatrix}$$

Quindi è sufficiente assegnare a θ una legge oraria tra quelle trattata per il singolo giunto avendo $\theta(0) = 0$ e $\theta(t_f) = \theta_f$, dopodichè si calcolano le componenti di r^i. Poiché r^i è costante, la velocità e l'accelerazione risultanti sono rispettivamente: $\omega^i = \dot{\theta} r^i$ $\quad \dot{\omega}^i = \ddot{\theta} r^i$
Infine, per caratterizzare la traiettoria di orientamento dell'end-effector rispetto alla terna base, sono necessarie le seguenti trasformazioni:

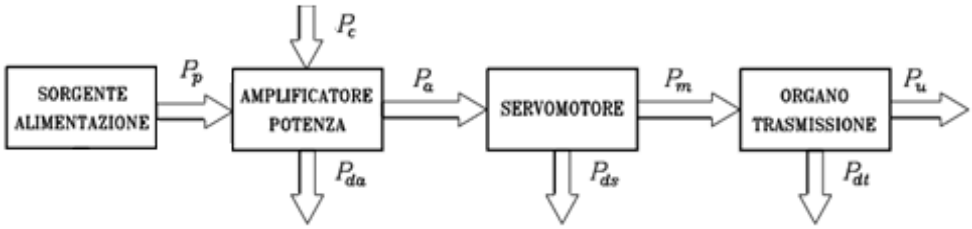
$$\boldsymbol{R_e(t) = R_i R^i(t)}$$
$$\omega_e(t) = R_i \omega^i(t) = R_i \dot{\theta} r^i$$
$$\dot{\omega}_e(t) = R_i \dot{\omega}^i(t) = R_i \ddot{\theta} r^i$$

Una volta che un percorso e una traiettoria sono stati specificati nello spazio operativo in termini di pe(t) e φe(t) o Re(t), le tecniche di cinematica inversa possono essere utilizzate per trovare le corrispondenti traiettorie nello spazio articolare q(t).

4. Attuatori e Sensori

4.1 Generalità sugli Attuatori e Sensori

Il **Sistema di Attuazione** è la parte del sistema robotico che si occupa della sua movimentazione e può essere schematizzato come segue:



- **Sorgente Alimentazione**

Fornisce la potenza primario P_p al sistema e la sua natura dipende dal tipo di attuazione:

a. **Servomotori Elettrici**: l'alimentazione è costituita da un trasformatore e da un ponte raddrizzatore tipicamente non controllato che converte la tensione alternata in continua di pportuna ampiezza per alimentare l'amplificatore di potenza;
b. **Servomotori Idraulici**: l'alimentazione è più complessa in quanto è costituita da un motore primo (tipicamente un motore asincrono trifase) che aziona una pompa ad ingranaggi o pistoni per comprimere il fluido. Inoltre, sarà necessario un serbatoio per immaganizzare l'energia idraulica al fine di ridurre le inevitabili oscillazioni di pressione causata dalla variazione di richieta di portata, quest'ultima a sua volta legata alle condizioni di funzionamento del motore. In effetti, il serbatoio svolge la medesima funzione del condensatore quando viene impiegato come filtro all'uscita di un ponte raddrizzatore. Si può infine dedurre come la presenza di complessi circuiti idraulici operanti ad alte pressioni (dell'ordine di 100 atm) provochi un apprezzabile inquinamento dell'ambiente di lavoro.

- **Amplificatore di Potenza**

Ha il compito di modulare il flusso di potenza dell'alimentazione verso l'attuatore. Perciò, l'amplificatore di potenza preleva una quota parte di P_p che è proporzionale al segnale di controllo P_c. Essi possono essere realizzati mediante:

- Servomotori elettrici e in particolare amplificatori a transistori come:
  a. Convertitori CC/CC come i chopper;
  b. Convertitori CC/AC come gli inverter.
- Servomotori idraulici come servovalvole elettroidrauliche.

Per comandare un servomotore elettrico è necessario dotarlo di a tensione o corrente di forma idonea a seconda del tipo di servomotore impiegato. La tensione (o corrente) è continua per i servomotori CC a magneti permanenti, mentre è alternata per i servomotori CC brushless. Il valore della tensione per i servomotori in c.c. a magneti permanenti oi valori di tensione e frequenza per i servomotori in c.c. brushless sono determinati dal segnale di controllo dell'amplificatore, in modo da far eseguire al motore il movimento desiderato. Per le gamme di potenza tipicamente richieste dai movimenti articolari (dell'ordine di pochi kilowatt), vengono impiegati amplificatori a transistor opportunamente commutati mediante tecniche di modulazione di larghezza di impulso (PWM). Consentono di ottenere un'efficienza di

conversione di potenza Pa/(Pp + Pc) maggiore di 0,9 e un guadagno di potenza Pa/Pc dell'ordine di 106. Gli amplificatori utilizzati per controllare i servomotori in corrente continua a magneti permanenti sono convertitori DC-DC (elicotteri), mentre quelli impiegati per controllare i servomotori CC brushless sono i convertitori CC-CA (inverter). Il controllo di un servomotore idraulico viene effettuato variando la portata del fluido compresso erogato al motore. Il compito di modulare la portata è tipicamente affidato ad un'interfaccia (servovalvola elettroidraulica). Ciò consente di stabilire una relazione tra il segnale elettrico di comando e la posizione di un distributore in grado di variare la portata del fluido trasferito dalla sorgente primaria al motore. Il segnale di comando elettrico è solitamente amplificato in corrente e alimenta un solenoide che muove (direttamente o indirettamente) il distributore, la cui posizione viene misurata da un apposito trasduttore. In questo modo si ottiene un servo di posizione sullo stelo della valvola che riduce il verificarsi di qualsiasi problema di stabilità che può sorgere sul controllo del motore. L'ampiezza del segnale di controllo determina la portata del compresso

- **Servomotore**

Un servomotore è un particolare tipo di motore che, contrariamente a quelli tradizionali, deve lavorare in condizioni di velocità e coppia variabili per asservire un certo compito. Il suo scopo perciò sarà trasformare la potenza utile P_u in ingresso in potenza meccanica P_m direttamente fornita al giunto oppure, come più spesso accade, all'organo di trasmissione antecendente al giunto. Nella robotica, tipicamente, i servomotori devono assolvere due compiti:

- ✓ Inseguire una traiettoria;
- ✓ Regolazione di posizione.

Per poter assolvere a questi compiti, i motori costituenti i servomotori devono presentare le seguenti caratteristiche:

- ➢ Bassa inerzia ed elevato valore del rapporto potenza/peso;
- ➢ Elevata possibilità di sovraccarico e sviluppo di coppie impulsive;
- ➢ Capacità di sviluppare elevate accelerazioni;
- ➢ Elevato campo di variazione di velocità (da 1 a 1000);
- ➢ Elevata precisione di posizionamento (almeno 1/1000 di giro);
- ➢ Basso fattore di ondulazione della coppia in modo da garantire rotazioni continue anche a bassa velocità.

I servomotori impiegati nella robotica possono essere di tre tipi:

A. **Servomotori Pneumatici**: sono costituiti da motori pneumatici, i quali trasformano l'energia pneumatica fornita da un compressore in energia meccanica mediante pistoni o turbine ad aria. A causa delle comprimibilità del fluido di lavoro, i termini non lineari introdotti nel modello sono particolarmente complessi. Perciò, a causa della complessità del modello e del relativo controllore, i motori pneumatici vengono utilizzati esclusivamente per azioni molto semplici come apertura e chiusura di una pinza.

B. **Servomotori Idraulici**: sono costituiti da motori idraulici che trasformano energia idraulica immagazzinata in un serbatoio di accumulazione mediante opportune pompe, in energia meccanica. Tipicamente sono del tipo coppia pistone-cilindro per realizzare moto di traslazione oppure con pistoni assiali o radiali. I vantaggi dei servomotori idraulici sono:
   - ✓ Non presentano problemi di surriscaldamento in situazioni statiche;
   - ✓ Sono autolubrificati e il fluido circolante facilita lo smaltimento del calore;
   - ✓ Sono intrinsecamente sicuri in ambienti pericolosi

- ✓ Hanno un ottimo rapporto potenza/peso;
- ✓ Coppie elevate a basse velocità.

Gli svantaggi invece sono:

- o Necessità di una centrale idraulica di alimentazione;
- o Costo elevato, ridotta gamma di prodotti, difficoltà di miniaturizzazione;
- o Basso rendimento nella conversione di potenza;
- o Necessità di manutenzione periodica;
- o Inquinamento dell'ambiente di lavoro per perdite di olio;
- o Caratteristiche dinamiche dipendenti dalla temperatura del fluido in pressione.

C. Servomotori Elettrici: energia elettrica della rete di distribuzione e la convertono in uno tipo di energia (tipicamente meccanica). I motori elettrici principalmente impiegati sono:

a. **Servomotore a Corrente Continua a Magneti Permanenti**: un circuito statorico che genera il flusso magnetico di eccitazione (induttore): questo generatore` e sempre a magneti permanenti realizzati con materiali ceramici o con terre rare (campi elevati in spazi ridotti). Un circuito di armatura (indotto) che reca un avvolgimento costituito dalle spire di un solenoide avvolto su materiale ferromagnetico che costituisce il rotore. Un collettore che consente di collegare, mediante delle spazzole, i circuiti rotorici ad un circuito esterno di alimentazione secondo una logica di commutazione attuata dal movimento del rotore.

b. **Servomotore a Commutazione Elettronica**: un generatore di flusso magnetico di eccitazione (induttore)che funge da rotore; questo generatore` e a magnetipermanenti di materiale ceramico o a terre rare⋆un circuito di armatura (indotto) costituito da unavvolgimento polifase che funge da statore⋆un circuito di commutazione statico che, a partire dai segnaliforniti da un sensore di posizione posto sull'albero motore,genera la sequenza di alimentazione delle fasi del circuito di armatura in dipendenza del moto di rotazione del rotore

c. **Motori a Passo**: caratteristiche di funzionamento dinamico dipendenti dal carico. Impiegati nei micromanipolatori.

I vantaggi sono:

- ✓ La disponibilità diffusa di sorgente di alimentazione;
- ✓ Costo contenuto e vasta gamma di prodotti;
- ✓ Buon rendimento di conversione di potenza;
- ✓ Facile manutenzione;
- ✓ Assenza di inquinamento dell'ambiente di lavoro;
- ✓ Buone caratteristiche dinamiche;
- ✓ Ottima flessibilità di controllo.

Gli svantaggi invece sono:

- o Problemi di surriscaldamento in situazioni statiche a causa dell'effetto della gravità sul manipolatore (freni di stazionamento);
- o Necessità a di particolari protezioni per impiego in ambienti infiammabili;
- o Necessità di utilizzare organi di trasmissione

- **Organo di Trasmissione**

Filtra la potenza meccanica P_m in ingresso e genera la potenza utile P_u impiegata dal giunto. Diciamo però che in generale gli organi di trasmissione hanno tre fondamentali funzioni:

- ✓ **Modificare coppia e velocità del moto**. Per esempio i motori elettrici forniscono alte velocità e basse coppia, mentre nella robotica è richiesto esattamente il contrario. Perciò, a tal proposito vengono utilizzati i riduttori per ovviare a questo problema;
- ✓ **Cambiare le caratteristiche del moto**. Nella pratica i giunti prismatici, pur muovendosi linearmente, devono il loro moto a motori elettrici che hanno moto rotatorio perciò tra di essi è necessario un interfacciamento tra trasforma quest'ultimo moto in uno rettilineo;
- ✓ **Ridurre gli effetti di termini di accoppiamento non lineare nel modello**. I motori associati a giunti che muovono link introducono termini non lineari nel modello, in quanto dipendono appunto non linearmente della variabili di giunto, risultando essere degli elementi di disturbo. I riduttori meccanici dividono questi termini per il rapporto di trasmissione, il quale tipicamente è un numero molto elevato. Risultato di ciò i valori non lineari saranno trascurabili.

Gli organi di trasmissione oltre a dei lati positivi dati dalle funzioni appena descritto, presentaranno chiaramante degli svantaggi. Perciò, potremmo chiederci perché non effettuare un accoppiamento diretto per esempio tra l'asse del motore e del giunto. Il vantaggio è l'eliminazione di elasticità e gioco, tuttavia questo comporterà la presenta di termini non lineari significativi nel modello. La presenza di tali termini comporterà una elevata complessità del controllore da realizzare. Vediamo ora alcuni tipi di organi di trasmissione:

A. Ruote Dentate: esse portano ad una variazione dell'asse di rotazione e/o traslano il punto di applicazione della forza. Vista la necessità di basse velocità e alte coppia, le ruote dentate devono presentare dei denti di sezione larga e alberi rotti al fine di avere delle deformazioni limitate;

B. Vite-Madrevite: la loro principale funzione è trasformare il movimento rotatotorio in moto traslatorio. Altre funzioni secondarie che hanno questi tipi di organi è ridurre gli attriti, il gioco e aumentare la rigidezza;

C. Cinghie Dentate: consentono di allocare il motore lontano dall'asse del giunto, evitando così che il robot debba trasportare il peso dei propri attuatori. Il problema però è che supportano solo alte velocità e basse coppie a causa della loro deformabilità;

D. Catene: consentono di allocare il motore lontano dall'asse del giunto, evitando così che il robot debba trasportare il peso dei propri attuatori. Riescono a sopportare basse velocità ed alte coppia, tuttavia a basse velocità tendono ad oscillare.

E' da notare che, negli ultimi tre blocchi dello schema del sistema di attuazione, poiché essi utilizzano la potenza fornita dalla sorgente di alimentazione a causa di diversi processi sono chiaramente presenti delle perdite indicate con P_{da}, P_{ds}, P_{dt}.

Un **Sensore** è un dispositivo in grado di rilevare una grandezza interagendo con essa. Spesso il termine sensore viene associato al termine trasduttore, anche se quest'ultimo non fa altro che convertire una grandezza in un'altra di natura diversa. Il sensore invece interagisce con la grandezza da misurare, dopodichè mediante una operazione di conversione che si basa su un a certa legge nota, fornisce in uscita una grandezza di natura differente. Il sistema di sensoristica, quindi l'insieme dei sensori di cui è costituito il robot, gli conferisce la percezioni di se stesso e del mondo circostante. A questo punto potremmo chiederci perché studiare i sensori e i sistemi sensoristici. La motivazione risiede nel loro ruolo centrale della sensoristica

nell'ottenimento di elevate prestazioni del sistema robotico. Tipicamente i sensori vengono utilizzati per retroazionare l'uscita del sistema con cui calcolare i segnali errori, i quali misurano lo scostamento tra il riferimento e l'uscita stessa, necessari alle leggi di controllo. I sensori si dividono in due grandi caterogie che andremo a trattare nello specifico in seguito:

A. **Sensori Propriocettivi**: *misurano grandezze interne al sistema come la posizione, la velocità o la coppia dei giunti*;
B. **Sensori Esterocettivi**: *misurano grandezze esterna al sistema*.

4.1.1 Azionamenti

Le sorgenti di alimentazione devono essere coerenti con la tecnologia impiegata per i servomotori, perciò gli azionamenti possibili sono:

A. Azionamenti Elettrici

Tipicamente si utilizza un trasformatore accoppiato con un ponte raddrizzatore non controllato. Anzitutto dobbiamo andare a caratterizzare il modello matematico per poter arrivare ad uno schema a blocchi dell'azionamento elettrico. A tal proposito andiamo a studiare le seguenti relazioni:

- **Equilibrio Elettrico**: da un punto di vista modellistico, un motore CC a magneti permanenti e un motore CC senza spazzole forniti con il modulo di commutazione e il sensore di posizione possono essere descritti dalle stesse equazioni differenziali. Nel dominio della variabile complessa s, l'equilibrio elettrico dell'armatura è descritto dalle equazioni della tensione di armatura V_a e della forza elettromotrice indotta V_g, quest'ultima proporzionale dalla velocità angolare mediante la costante di tensione k_v dipendente dalle caratteristiche costruttive del motore.

$$\begin{cases} V_a = (R_a + sL_a)I_a + V_g \\ V_g = k_v\Omega \end{cases}$$

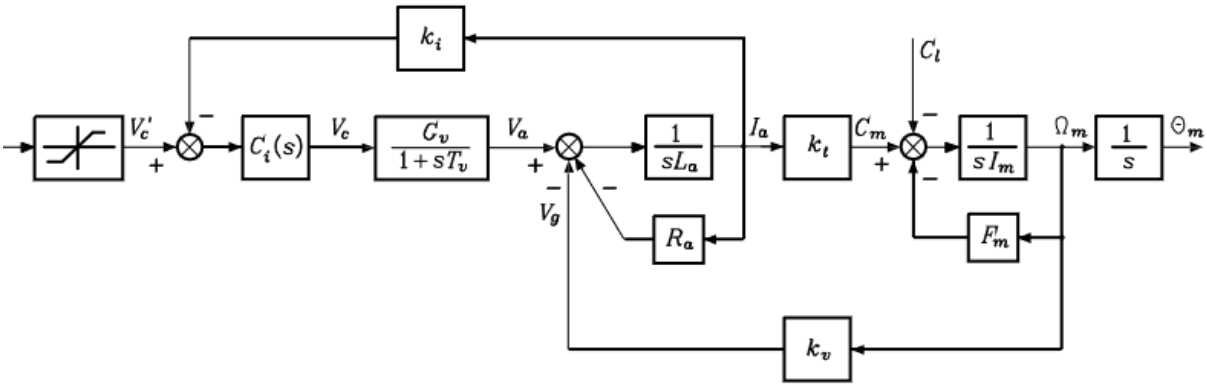
- **Equilibrio Meccanico**: è dato dalle equazioni della coppia motrice C_m e della coppia di reazione al carico C_l, le quali dipendono dal momento di inerzia I_m e dal coefficiente di attrito viscoso F_m all'albero motore. La C_l è propozionale alla corrente di armatura I_a secondo una costante k_t numericamente uguale a k_v.

$$\begin{cases} C_m = (sI_m + F_m)\Omega + C_l \\ C_l = k_t I_a \end{cases}$$

- **Amplificatore di Potenza**: indicando con G_v il guadagno in tensione dell'amplificatore, che tipicamente è un numero grande, mentre T_v è la sua costante di tempo, che tipicamente è un numero piccolo, il guadagno complessivo dell'amplificatore è dato dalla funzione di trasferimento di fianco.

$$\frac{V_a}{V_c} = \frac{G_v}{1 + sT_v}$$

Nella seguente figura è schematizzato l'azionamento elettrico, nel quale oltre ai blocchi corrispondenti alle relazioni di cui sopra, è presente un anello di retroazione della corrente di armatura. Quest'ultima è misurata da un trasduttore k_i tra l'amplificatore di potenza e l'avvolgimento di armatura del motore. Inoltre, lo schema presenta un regolatore di corrente $C_i(s)$ e un elemento con una caratteristica di saturazione non lineare.



Lo scopo di tale feedback è duplice.

- Da un lato la tensione V_c' svolge il ruolo di riferimento di corrente e quindi, mediante un'opportuna scelta del regolatore $C_i(s)$, si può ridurre lo sfasamento tra la corrente I_a e la tensione V_c' rispetto allo sfasamento tra I_a e V_c;

- D'altra parte, l'introduzione di una non linearità di saturazione consente la limitazione dell'ampiezza di V_c, e quindi funziona come un limite di corrente che garantisce la protezione dell'amplificatore di potenza ogni volta che si verificano delle condizioni di funzionamento anomale.

  La scelta del regolatore Ci(s) dell'anello di corrente consente di ottenere dall'azionamento elettrico un comportamento controllato in velocità o in coppia, a seconda dei valori raggiunti dal guadagno dell'anello k_i:

  A. Se $k_i = 0$ → Generatore Controllato in Velocità: si che il coefficiente di attrito viscoso meccanico è trascurabile rispetto al coefficiente di attrito elettrico: $F_m \ll (k_v k_t)/R_a$

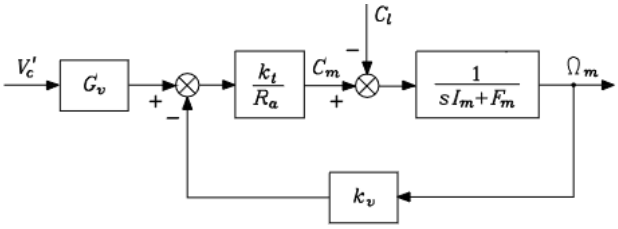
  Di fatto è come se andiamo ad eliminare la retroazione in corrente. A regime, considerando le relazioni di questa casistica, e supponendo inoltre che il guadagno del regolatore in corrente sia unitario, ovvero $C_i(0) = 1$, e che $C_l = 0$ otteniamo una *relazione diretta* tra la velocità angolare dell'asse del motore ω_m e il riferimento di corrente v_c'. Allora, l'azionamento si comporta come un generatore controllato in velocità. Per quanto riguarda il comportamento dinamico, se ipotizziamo:

$$\boldsymbol{\omega_m \approx \frac{G_v}{k_v} v_c'}$$

  - Un modello di ordine ridotto ottenuto ritentendo la costante di tempo elettrica L_a/R_a trascurabile rispetto a quella meccanica I_m/F_m;
  - Assumento $T_v = 0$;
  - Considerando un controllore puramente proporzionale.

  In questa casistica (ovvero per $k_i = 0$), ricaviamo il seguente schema grazie al quale possiamo stabilire la relazione tra ingresso V_c' e uscita Ω_m:

$$\Omega_m = \frac{\frac{1}{k_v}}{1 + s\frac{R_a I_m}{k_v k_t}} G_v V_c' - \frac{\frac{R_a}{k_v k_t}}{1 + s\frac{R_a I_m}{k_v k_t}} C_l$$



  B. Se $k_i \neq 0$ → Generatore Controllato in Coppia: si ha che, scegliendo un guadagno per l'anello in corrente di valore alto, ovvero $G_v k_i \gg R_a$, la coppia motrice a regime sarà data dalla relazione di fianco.

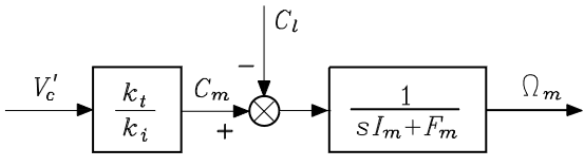
$$\boldsymbol{c_m \approx \frac{k_t}{k_i}\left(v_c' - \frac{k_v}{G_v}\omega_m\right)}$$

  Ma, se G_v ha un valore alto, allora la coppia è praticamente indipendente dalla velocità angolare ω_m. Allora, l'azionamento si comporta come un generatore controllato in coppia. Per quanto riguarda il comportamento dinamico, se ipotizziamo:

  - $G_v k_i \gg R_a$;
  - $k_v \Omega_m / G_v k_i \approx 0$.

  In questa casistica (ovvero per $k_i \neq 0$), ricaviamo il seguente schema grazie al quale possiamo stabilire la relazione tra ingresso V_c' e uscita Ω_m:

$$\Omega_m = \frac{\frac{k_t}{k_i F_m}}{1 + s\frac{I_m}{F_m}} V_c' - \frac{\frac{1}{F_m}}{1 + s\frac{I_m}{F_m}} C_l$$



La relazione tra l'ingresso di controllo e l'uscita di posizione dell'attuatore può essere espressa in modo unificato dalla funzione di trasferimento $M(s)$, dove:

$$M(s) = \frac{k_m}{s(1 + sT_m)}$$

a. Generatore Controllato in Velocità:

$$k_m = \frac{1}{k_v} \qquad T_m = \frac{R_a I_m}{k_v k_t}$$

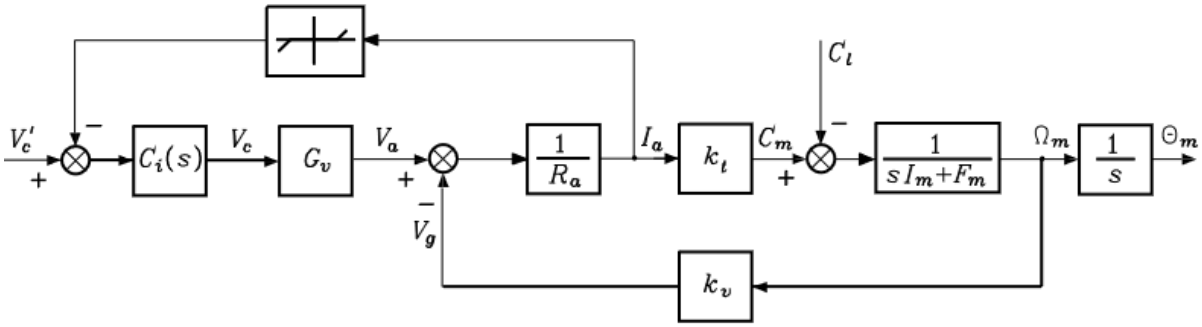
b. Generatore Controllato in Coppia:

$$k_m = \frac{k_t}{k_i F_m} \qquad T_m = \frac{I_m}{F_m}$$

Possiamo ora fare un confronto i due scenari. Osservando le relazioni ingresso/uscita, l'amplificatore di potenza contribuisce ad essa nel caso del controllo di velocità per mezzo della costante G_v. Invece, nel controllo in coppia in cui è presente l'anello di corrente, l'amplificatore si trova all'interno di quest'ultimo anello di retroazione locale presentandosi non a sé stante ma pittuosto nell'espressione di k_m come $1/k_i$. Da ciò si può concludere che, tipicamente il generatore controllato in velocità *è più conveniente* rispetto al generatore controllato in coppia perché:

- ✓ La coppia di carico C_l, la quale abbiamo detto essere un disturbo, viene pesata (moltiplicata) per una quantità costante piccola rispetto al caso del generatore controllato in coppia; $\frac{R_a}{k_v k_t} \ll \frac{1}{F_m}$
- ✓ Si ha una più bassa latenza e quindi una migliore prontezza ai disturbi in quanto vale la relazione di fianco. $\frac{R_a}{k_v k_t} \ll \frac{I_m}{F_m}$

A questo punto però potremmo porci il problema della protezione, tuttavia, esso può essere risolto agevolmente introducendo un limite di corrente che non viene eseguito da una saturazione sul segnale di comando ma sfrutta una retroazione di corrente con non linearità di zona morta sul percorso di retroazione, come nel seguete schema.



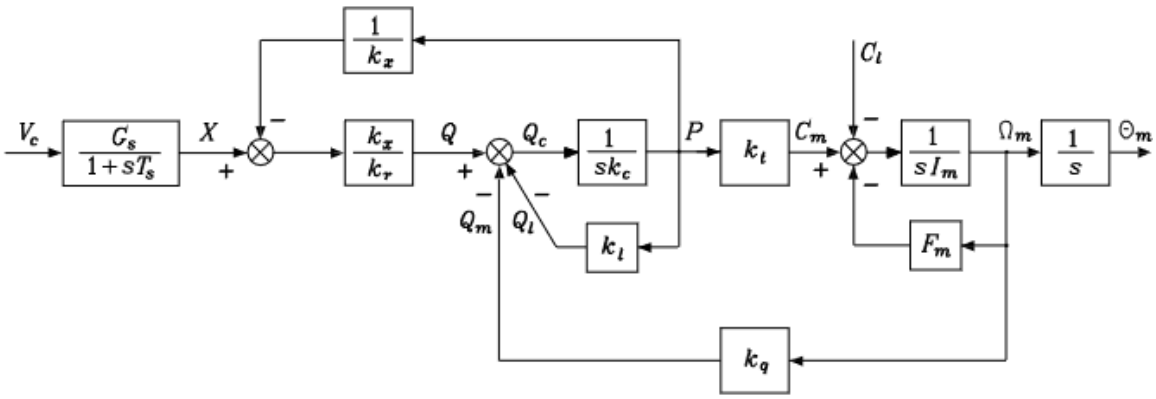
Perciò, seppur sia vantaggioso inserire una retroazione della I_a nel caso del generatore controllato in coppia per i seguenti motivi:

- ✓ Protezione dell'amplificatore;
- ✓ Migliore inseguimento del riferimento di corrente.

Tale approccio è svantaggioso quando andiamo ad impiegare un controllo indipendenti ai giunti, ovvero lavoriamo sul singolo giunto mentre tutti gli altri sono un disturbo. Se non vogliamo che questo disturbo sia troppo incisivo sull'albero di uscita del motore. In tal caso per proteggere l'amplificatore invece di utilizzare una non linearità di tipo saturazione si impiega una saturazione soglia da inserire sulla catena diretta. Se invece il controllo è centralizzato in cui controlliamo tutti i giunti allora si utilizza il generatore controllato di coppia.

B. Azionamenti Idraulici

Tipicamente si utilizza come servovalvola una pompa azionata da un motore primo, in genere asincrono trifase a velocità costante, insieme ad un serbatoio di accumulazione, filtri e valvole. Nella seguente immagine è mostrato lo schema di un motore idraulico con servovalvola e del serbatoio. I ragionamenti fatti nel caso dell'azionamento elettrico possono essere riportanti anche al caso degli azionamenti idraulici.



Le differenze tuttavia sono principalmente due:

- **Differenza 1**: il modello matematico farà riferimento alle equazioni che descrivono la relazione tra portata e pressione, la relazione tra il fluido e le parti in movimento e l'equilibrio meccanico delle parti in movimento;
- **Differenza 2**: la retroazione della pressione P (invece che della corrente come nell'altro schema) non è soggetta alla nostra arbitrarietà ma è *strutturale*. Da ciò capiamo che è possibile far svolgere a un azionamento idraulico il ruolo di un generatore controllato in velocità o coppia, come per un azionamento elettrico. Infatti, l'unico modo per modificare l'anello di retroazione della pressione è attraverso l'introduzione di opportuni trasduttori e la realizzazione della relativa circuiteria di controllo.

A questo punto, iniziamo a caratterizzare l'equazione della portata Q: $\boldsymbol{Q = Q_m + Q_l + Q_c}$

Dove:

- Q_m = portata motrice. La portata volumetrice trasferita al motore è proporzionale alla variazione di volume nelle camere per unità di tempo. Perciò, facendo d'ora in poi riferimento a servomotori rotativi, tale variazione è proporzionale alla velocità angolare Ω_m: $Q_m = k_m \Omega_m$
- Q_l = portata che rappresenta la perdita per trasfilamenti. Se P è la pressione del servomotore dovuta al carico, allora il legame tra Q_l e P è: $Q_l = k_l P$
- Q_c = portata che rappresenta la perdita dovuta all pressione elevata. Se V è il volume istantaneo del fluido e γ è il coefficiente di comprimibilità, dove quest'ultimo è uniforme, allora la relazione tra Q_c e V è: $Q_c = \gamma V$

L'equilibrio meccanico delle parti in movimento è descritto dalla seguente equazione in cui la coppia motrice è proporzionale alla pressione P, cioè $C_m = k_t P$:

$$C_m = (sI_m + F_m)\Omega_m + C_l$$

Relativamente alla servovalvola, la funzione di trasferimento tra la posizione dello stelo X e la tensione di comando V_c si può esprimere grazie all'effetto linearizzante ottenuto dal feedback di posizione.

$$\frac{X}{V_c} = \frac{G_s}{1 + sT_s}$$

Dove:

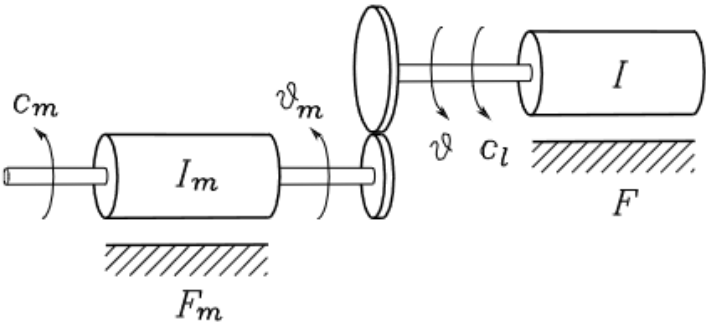
- $G_s =$ guadagno equivalente della servovalvola;
- $T_s =$ costante di tempo della servovalvola ed è dell'ordine dei ms perciò può essere trascurata rispetto alle altre costanti di tempo del sistema.

Infine, per quanto riguarda il distributore, il rapporto tra la pressione differenziale, la portata e lo spostamento dello stelo è fortemente non lineare. Perciò, la linearizzazione attorno ad un punto di lavoro porta all'equazione: $P = k_x X - k_r Q$

4.1.2 Riduttore

Un **Riduttore** è un dispositivo meccanico, tipicamente costituito da ruote dentate, la cui funzione è trasformare un moto a bassa coppia e velocità elevata dell'albero in ingresso in un moto ad alta coppia e bassa velocità sull'albero in uscita. Nel nostro caso inoltre, il riduttore ha uno scopo anche *controllistico*, ovvero rendere trascurabili i termini non lineari del modello grazie all'alto valore del rapporto di trasmissione. Andiamo a studiare nel dettaglio questo aspetto considerando la coppia cinematica in figura dove:

- **Lato motore m**: conosciamo il coefficiente di attrito viscoso F_m e il momento d'inerzia I_m;
- **Lato carico**: conosciamo il coefficiente di attrito viscoso F e il momento d'inerzia I.



Supponendo di avere un accoppiamento perfetto delle due ruote, quindi assenza di effetti dissipativi e gioghi, abbiamo l'equilibrio delle forze lato motore e lato carico lungo la superfice di contatto (relazione di sinistra). Da questa relazione possiamo ricavare il rapporto di trasmissione k_r, il quale risulta essere una quantità molto grande in quanto il raggio della ruota grande è molto maggiore del raggio della ruota piccola, ovveo $r \gg r_m$ (anche se non sembra in figura):

$$f_m = r_m\theta_m = r\theta = f \quad \rightarrow \quad k_r = \frac{r}{r_m} = \frac{\theta_m}{\theta} = \frac{\omega_m}{\omega}$$

Non resta che ricavare la coppia motrice c_m e la coppia resistente f_r:

- **Coppia Motrice c_m**: essa è data dalla somma dei seguenti termini dove il terzo rappresenta l'effetto del carico visto dal motore. Perciò il motore quando movimenta il suo asse, percepisce la presenza del carico e quindi della f, il quale viene pesato per il raggio r_m della ruota piccola: $c_m = I_m\dot{\omega}_m + F_m\omega_m + f \cdot r_m$
- **Coppia Resistente f_r**: essa è data dalla somma dei seguenti termini in cui compare la coppia di reazione al carico c_l applicata all'asse del carico, la quale è un termine non lineare funzione della configurazione q del robot: $f_r = I\dot{\omega} + F\omega + c_l(q)$

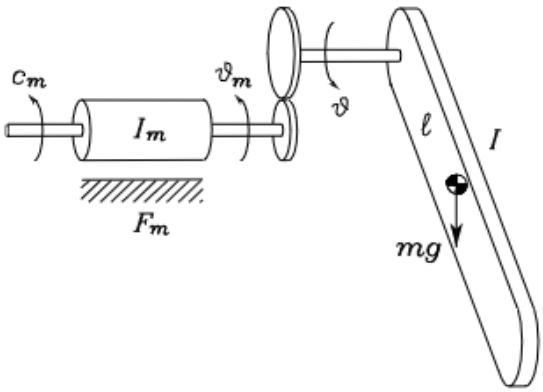
Sostituendo la f_r nella relazione di c_m otterremo una espressione di quest'ultima in cui il momento di inerzia equivalente e la forza di attrito viscoso equivalente vengono notevolmente smorzati dal termine k_r^2.

Come detto in precedenza, la coppia di carico $c_l(q)$ è funzione della q, perciò risulta essere non lineare. Tuttavia, grazie al rapporto di trasmissione k_r tale termine viene notevolmente ridotto ed eventualmente possiamo trascurarlo:

$$c_m = I_{eq}\dot{\omega}_m + F_{eq}\omega_m + \frac{c_l(q)}{k_r} = \left(I_m + \frac{I}{k_r^2}\right)\dot{\omega}_m + \left(F_m + \frac{F}{k_r^2}\right)\omega_m + \cancel{\frac{c_l(q)}{k_r}}$$

In definitiva, abbiamo così ottenuto una *linearizzazione del modello* e conseguentemente una semplificazione del controllo.

In alcuni casi possiamo immaginare che un singolo link sia equivalente ad un *pendolo rigido*, il quale abbiamo detto essere attuato da un riduttore. Anche in questo caso calcoliamo la coppia motrice e la coppia resistente



- **Coppia Motrice c_m**: essa presenta una espressione analoga all'esempio precedente: $c_m = I_m\dot{\omega}_m + F_m\omega_m + fr_m$
- **Coppia Resistente f_r**: essa è data dalla somma dei seguenti termini in cui compare la coppia di carico c_l, la quale è un termine non lineare funzione della configurazione q del robot:

$$f_r = I\dot{\omega} + F\omega + c_l = I\dot{\omega} + F\omega + mgl \cdot sin(\theta)$$

Sostituendo la f_r nella relazione di c_m otterremo una espressione di quest'ultima in cui l'effetto del carico è stato linearizzato:

$$c_m = I_{eq}\dot{\omega}_m + F_{eq}\omega_m + \left(\frac{mgl}{k_r}\right)\sin\left(\frac{\theta_m}{k_r}\right) = \left(I_m + \frac{I}{k_r^2}\right)\dot{\omega}_m + \left(F_m + \frac{F}{k_r^2}\right)\omega_m + \left(\frac{mgl}{k_r}\right)\sin\left(\frac{\theta_m}{k_r}\right)$$

Nel caso reale di n bracci, gli accoppiamenti non lineari tra i motori verrano ridotti dalla presenza di organi di trasmissione con elevati rapporti di riduzione.

4.1.3 Controllo di Posizione

Dopo aver esaminato le modalità di controllo della velocità angolare di un azionamento elettrico o idraulico, va risolto il problema del controllo del movimento di un braccio del manipolatore. Il **Controllo di Posizione** consiste nella realizzazione di uno schema di controllo che consenta il perfetto inseguimento del riferimento $\Theta_r(t)$, ovvero il moto desiderato, da parte di Θ_m, ovvero il moto effettivo del motore: $\boldsymbol{\Theta_m(t) = \Theta_r(t)}$

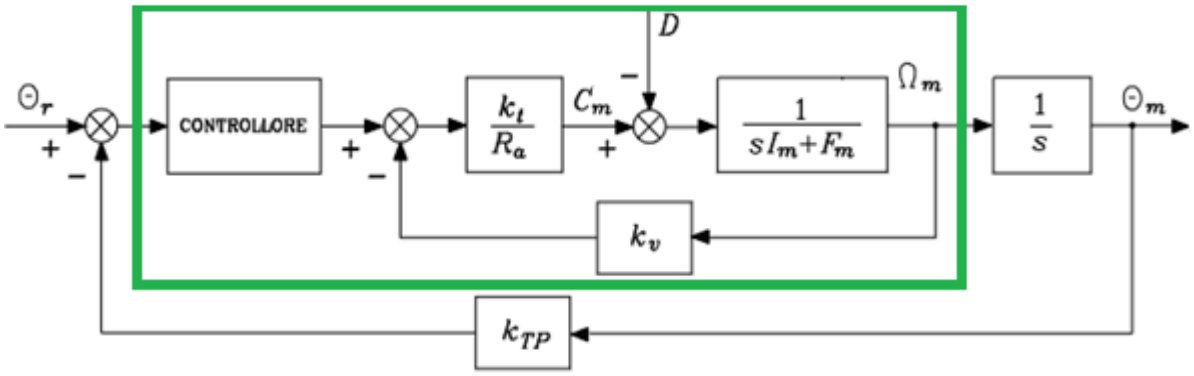
Dalla teoria del controllo sappiamo che, oltre all'inseguimento del riferimento, quando progettiamo un controllore dobbiamo garantire:

- ✓ Stabilità a ciclo chiuso;
- ✓ Buona prestazioni dinamiche, le quali si riferiscono ai transitori;
- ✓ Reiezioni ai disturbi, il quali non si riferiscono solo alla presenza del carico.

Per affrontare questo problema adottiamo delle ipotesi semplificative:

- ✓ Abbiamo già eseguito la pianificazione della traiettoria nello spazio operativo;
- ✓ Abbiamo già eseguito l'inversione cinematica (eventualmente differenziale) per ricavare i riferimenti delle variabili di giunto $\Theta_r(t)$;
- ✓ Il moto di un giunto è indipendente dal moto degli altri giunti, anzi l'interazione tra essi può essere visto come un disturbo che chiameremo D;
- ✓ Il disturbo D agisce solo a valle del controllore che dovremo realizzare.

Lo schema di controllo che ci consente di risolvere il controllo di posizione è il seguente. Anzitutto abbiamo come ingresso al sistema il riferimento Θ_r, mentre come uscita Θ_m, la quale viene retroazionata per mezzo di un trasduttore con costante di trasduzione costante k_{TP}. Il segnale errore, dato dalla differenza tra $\Theta_r - \Theta_m$ viene fornito al riquadro verde che di fatto è proprio l'azionamento elettrico con generatore controllato in velocità. La scelta di quest'ultimo, anziché controllato in coppia, è dovuta al requisito della reiezione ai disturbi. Osserviamo che il segnale generato dal controllore viene dato in ingresso all'amplificatore, la cui uscita è la coppia motrice C_m che dovrebbe andare in ingresso al blocco che rappresenta il motore e definito dal momento di inerzia I_m e coefficiente di attrito viscoso F_m. Il risultato sarà un moto dell'asse molto vicino al moto desiderato Θ_r.



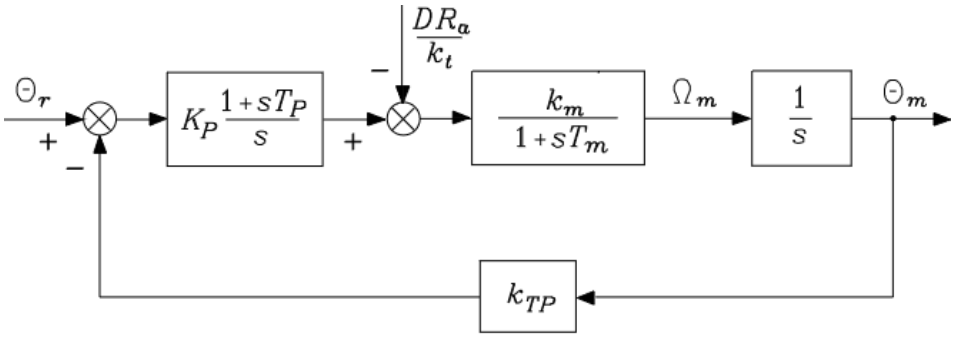
Andiamo ora semplificare questo schema. Al fine di garantire la stabilità, un buon comportamento dinamico e la reizione ai disturbi ci conviene adottare un semplice controllore PI. Il guadagno proporzionale K_p e quello integrale K_I dovranno essere scelti opportunamente sulla base degli obiettivi da conseguire. Ricordiamo che per l'azionamento abbiamo impiegato il generatore controllato di velocità, per il quale possiamo definire il guadagno tensione-velocità k_m e la costante di tempo T_m del motore come:

$$k_m = \frac{1}{k_v} \qquad T_m = \frac{R_a I_m}{k_v k_t}$$

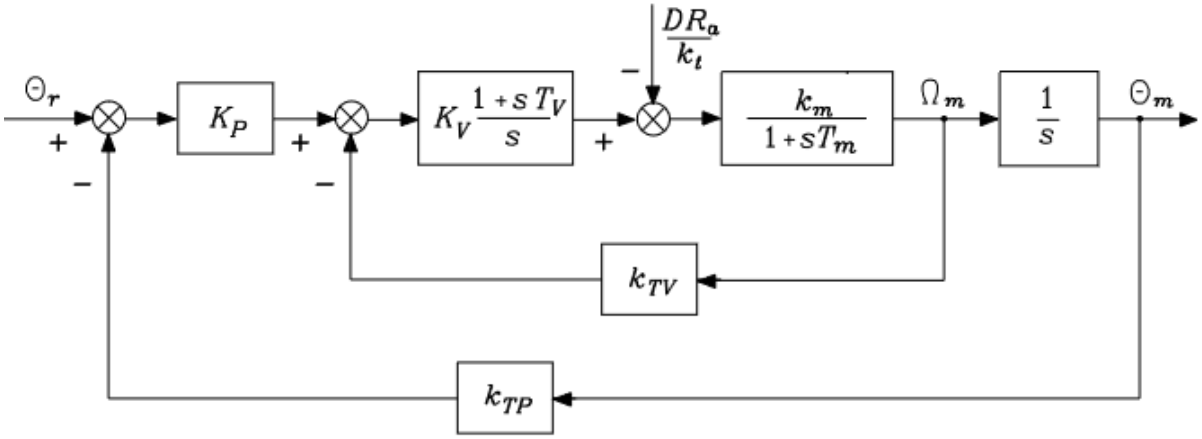
Queste ultime le possiamo integrare internamente al blocco relativo al motore.

Ultima semplificazione che possiamo adottare è dettata dal fatto che la coppia di reazione influenza la velocità dell'asse motore con un coefficiente pari a $R_a/k_v k_t$, il quale ricordiamo essere molto inferiore al peso $1/Fm$ sulla coppia reazione che avremmo con il controllo in coppia. Perciò la coppia di reazione possiamo porla come: $C_l = DR_a/k_t$

Pertanto, dallo schema precedente possiamo pervenire allo **Schema di Controllo dell'Azionamento con Feedback di Posizione**:



In realtà, possiamo una migliore reiezione ai disturbi e una banda passante maggiore, quindi una migliore reattività del sistema e quindi un migliore inseguimento del riferimento, adottando una **Retroazione Tachimetrica**. Otteniamo così lo **Schema di Controllo dell'Azionamento con Feedback di Posizione e Velocità** mostrato di seguito. In pratica è come adottare nuovamente lo schema del generatore controllato in velocità per realizzare un secondo anello di retroazione interno che appunto lavora su questa grandezza. Infatti, avremo a monte un blocco proporzionale K_P, mentre sulla linea diretta il regolatore PI con parametri K_V e T_V, invece sulla linea di retrozione un trasduttore di velocità k_{TV}.



Come ultima considerazione, lo schema appena visto può essere estese anche al caso in cui il motore sia accoppiato ad un carico tramite una riduzione ad ingranaggi. In tal caso, è sufficiente tenere conto della seguente relazione ricavata in precedenza:

$$c_m = I_{eq}\dot{\omega}_m + F_{eq}\omega_m + \frac{c_l(q)}{k_r} = \left(I_m + \frac{I}{k_r^2}\right)\dot{\omega}_m + \left(F_m + \frac{F}{k_r^2}\right)\omega_m + \frac{c_l(q)}{k_r}$$

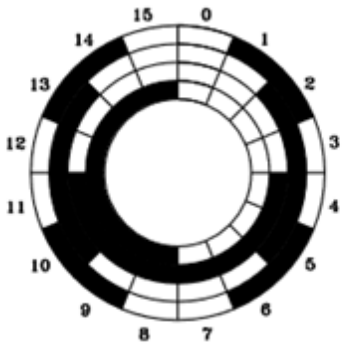
Perciò, nello schema dovremo sostituire I_m ed F_m con I_{eq} e F_{eq}, inoltre dovremo scalare il disturbo per il fattore $1/k_r$ dove k_r è il rapporto di trasmissione del riduttore.

4.2 Sensori Propriocettivi

4.2.1 Trasduttori di Posizione

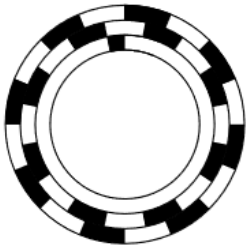
I **Trasduttori di Posizione** *forniscono un segnale elettrico proporzionale allo spostamento lineare o angolare di un apparato meccanico rispetto ad una data posizione di riferimento*. Essi vengono suddisivisi in due grandi categorie a seconda del tipo di posizione misurata, quindi parleremo di trasduttori di posizione lineari oppure angolari. Tipicamente i trasduttori di posizione lineari, come potenziometri e inductosyns,vengono impiegati per il controllo di macchine utensili. Nella robotica, invece, vengono più spesso impiegati i *trasduttori di posizione angolari*, i quali dunque misurano la posizione angolare dell'albero su cui sono calettati. Ciò è motivato dalla caratteristica dei servomotori di essere rotativi, inoltre questo tipo di trasduttori godono di elevata precisione, robustezza e affidabilità. Andiamo ad analizzare i trasduttori di posizione angolare maggiormente impiegati:

- **Encoder Assoluto**: è un disco di vetro ottico costituito da un insieme **Tracce**, ovvero di corone concentriche caratterizzate da specifiche sequenze di zone chiare e scure ottenunte mediante il deposito di un film metallico. Il disco poi viene calettato sull'albero di cui si vuole valutare la posizione angolare. Il sensore è inoltre dotato di una coppia emettitore di luce-fotodiodo posti sulle due faccie del disco in posizione fissa. Quando l'albero, e quindi il disco sono in rotazione, il primo emetterà una luce di cui solo la parte attraversante le zone chiare riuscirà a raggiungere il fotodiodo. La presenza o assenza di luce viene associata ad un segnale binario. Allora opacizzando le zone del disco secondo una certa legge di codifica sarà possibile interpretare i segnali binari per risalire alla posizione angolare. In particolare, nel caso dell'encoder in figura, viene adottato la **Codifica Grey** implementata dalla tabella di verità di fianco. Infine, maggiore sarà il numero di traccie e maggiore sarà il numero di bit a nostra disposizione, aumentando così la risoluzione dello strumento. La risoluzione tipica richiesta per il controllo congiunto vengono impiegati encoder assoluti con un numero minimo di 12 tracce (bit) (risoluzione 1/4096 per cerchio). E' da specificare però che, qualora sia presente un riduttore, un cerchio lato giunto corrisponde a più cerchi lato motore. Dunque, in tal caso si avrà la necessità di una elettronica per contare e memorizzare il numero di cerchi effettivi.



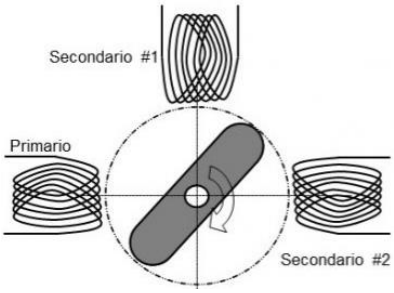
#	Codice	#	Codice
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

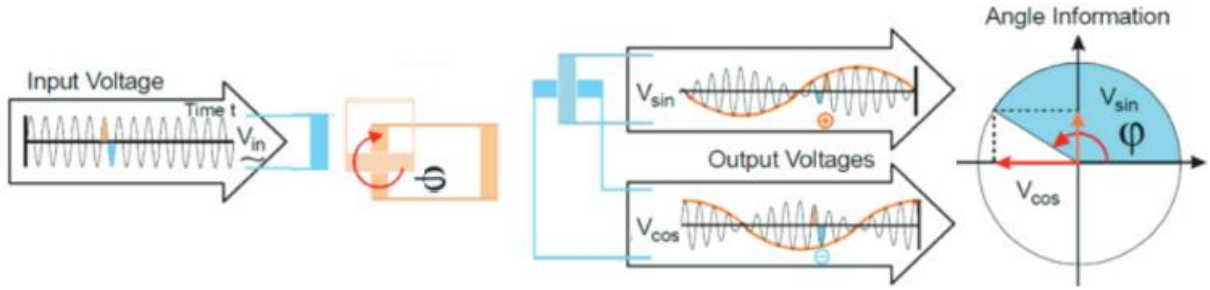
- **Encoder Incrementale** è una versione alternativa dell'encoder assoluto in grado non solo di calcolare la posizione angolare del giunto ma anche il *verso della rotazione*. Presenta una struttura di base analoga all'encoder assoluto, con la differenza però che è costituto da solo due corone in cui i settori trasparenti e opachi sono in quadratura (disposti a scacchiera). Inoltre, sarà presente una terza corona più interna che indicherà lo zero meccanico. L'encoder incrementale effettua dunque una misura di posizione dell'albero per l'appunto di tipo incrementale rispetto alla posizione precedente di quest'ultimo. Per quanto riguarda il senso di rotazione, poiché a ciascuna traccia corrisponde un segnale di uscita, basta verificare quale dei due segnali precede l'altro. Gli encoder incrementali presentano una struttura molto semplice e hanno dei costi ridotti rispetto agli encoder assoluti, tuttavia presentano un importante svantaggio. Se utilizziamo questo tipo di encoder per un sistema di azionamento articolato, è richiesta ovviamente una elettronica che effettui le valutazioni di posizione, i cui risultati sono conservati in memorie volatili. Se per qualche motivo la memoria dovesse essere cancellata (a causa di disturbi, danni, ecc.), avremmo perso la misura. Questa condizione non si verifica negli encoder assoluti in quanto l'informazione di posizione angolare è codificata direttamente sul disco ottico a fronte però di un costo maggiore del trasduttore. Infine, se viene impiegato un circuito esterno, le misurazioni della velocità possono essere ricostruite dalle misurazioni della posizione. Infatti, se ad ogni transizione viene generato un impulso, è possibile ottenere una misura di velocità in tre possibili modi:



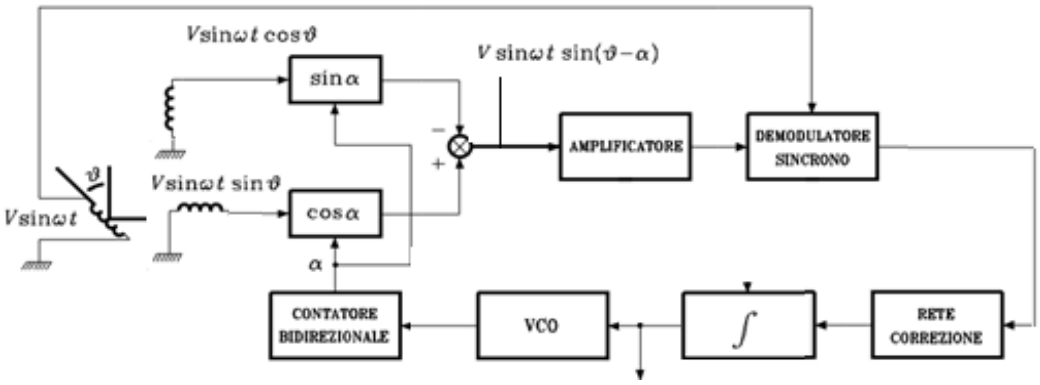
a) Convertitore tensione-frequenza con uscita analogica;
b) Misurando digitalmente la frequenza del treno di impulsi nel caso si vogliano misurare alte velocità;
c) Misurando digitalemente il tempo di campionamento del treno d'impulsi nel caso si vogliano misurare basse velocità.

- **Resolver**: è un trasduttore di posizione elettromeccanico che sfrutta il principio di mutua induzione per misurare spostamenti angolari. Il resolvere è in pratica una macchina elettrica costituita da:
  - Rotore su cui è presenta la bobina di induttanza alimentata da una tensione in ingresso sinusoidale $V\sin(\omega t)$ (con frequenze tipiche nell'intervallo da 0,4 a 10 kHz);
  - Statore su cui sono presenti due avvolgimenti a 90 gradi elettrici l'uno dall'altro e sui quali vengono indotta delle tensioni in controfase (una avreno il seno dell'angolo e l'altra il coseno) la cui ampiezza dipende dall'angolo di rotazione del rotore.





Un **RDC** (**Resolver-Digital Converter**) dovrà poi trattare le due sinusoidi per estrapolare i dati digitali corrispondenti alla posizione angolare. Lo schema a blocchi dell'RDC è mostrato nella seguente figura.



Andiamo ad analizzare la struttura dell'RDC partendo dall'uscita del resolver. Dai due avvolgimenti statorici abbiamo rispettivamente in uscita le tensioni $Vsen(\omega t)cos(\theta)$ e $Vsen(\omega t)sen(\theta)$.

2. Il primo entra in un blocco moltiplicatore che esegue il prodotto di tale segnale per $\sin(a)$, analogamente il secondo per $\cos(a)$;
3. Successivamente i due segnali in uscita ai moltiplicatori vengono sommati fornendo $Vsin(\omega t)\sin(\theta - a)$;
4. Tale segnale viene poi opportunamente amplificato, poi trattato dapprima da un demodulatore sincrono e poi da una rete correttrice per conferire al sistema stabilità e larghezza banda. Infine, il segnale viene integrato;
5. Il segnale risultante viene convertito in un treno di impulsi da un convertitore tensione-frequenza;
6. Il treno d'impulsi va in ingresso ad un contatore avanti-indietro che fornisce come feedback ai blocchi moltiplicatori un parametro a che rappresenta la misura di posizione angolare di interesse.

La presenza di due integratori (uno è rappresentato dal contatore avanti-indietro) nel loop assicura che le misurazioni di posizione (digitali) e velocità (analogiche) siano prive di errori finché il rotore ruota a velocità costante. Infatti un errore di arrotondamento su α influisce sulla misura della posizione. I vantaggi del resolver sono:

- ✓ Affidabilità e robustezza (infatti viene impiegato in scenari soggetti a vibrazioni);
- ✓ Conferisce una trasmissione continua priva di limiti meccanici della posizione angolare grazie al principio della mutua induzione;
- ✓ Precisione.

Gli svantaggi invece sono:

- L'ingombro e il peso;
- Costoso;
- Richiede una buona competenza per la sua implementazione.

4.2.2 Trasduttori di Velocità

I **Trasduttori di Velocità** *forniscono un segnale elettrico proporzionale alla velocità lineare o angolare.* Sebbene le misure di velocità possano essere ricostruite da trasduttori di posizione, spesso si preferisce ricorrere a misure dirette di velocità, mediante opportuni trasduttori. I trasduttori di velocità più tipici sono i tachimetri, i quali sono dispositivi basati sui principi di funzionamento delle macchine elettriche. I due tipi fondamentali di tachimetri sono:

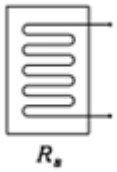
- **Tachimetro in Corrente Continua** o **Dinamico Tachimetrica**: è un generatore in CC a magneti permanenti che, grazie al suo flusso di campo costante, genera una tensione di uscita proporzionale alla velocità angolare. Due aspetti a cui porre attenzione sono:
  - ❖ La cura di costruzione in modo da ottenere un rapporto lineare ingresso/uscita e ridurre gli effetti dell'isteresi magnetica;
  - ❖ A causa della presenza di un commutatore, la tensione di uscita presenta un'ondulazione residua che non può essere eliminata con un opportuno filtraggio in quanto la sua frequenza dipende dalla velocità angolare.
- **Tachimetro in Corrente Alternata**: è una macchina elettrica che presenta due avvolgimenti statorici in quadratura e un *rotore a coppa*, la cui peculiare forma conferisce un basso momento di inerzia. Quando uno degli avvolgimenti è alimentata da una tensione sinusoidale di ampiezza costante, sull'altro avremo una tensione della medesima natura e frequenza, con ampiezza proporzionale alla velocità angolare. Inoltre, a seconda del segno della fase avremo una rotazione in un verso o nell'altro. In tal modo otteniamo il medesimo risultato del tachimetro in CC ma senza adottare una struttura a generatore, il che comporta diversi vantaggi:
  - ✓ L'ondulazione residua sulla tensione di uscita può essere eliminata con un opportuno filtro visto che la sua frequenza fondamentale è doppia rispetto alla frequenza di alimentazione;
  - ✓ Assenza di contatti striscianti;
  - ✓ Basso momento di inerzia grazie all'adozione del rotore a coppa.

  Tuttavia, a causa degli inevitabili accoppiamenti parassiti tra la bobina dello statore e la circuiteria di misura, a rotore fermo sarà presente una tensione residua che rende diversa da zero la misura di velocità.

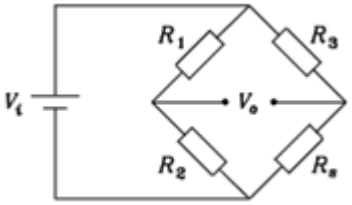
4.3 Sensori Esterocettivi

4.3.1 Sensore di Forza

Il **Sensore di Forza** *misura la forza o coppia in maniera indiretta valutando la deformazione indotta da queste ultime su ad un elemento estensibile di caratteristiche adeguate*. Tutti i sensori di forza (forza lineare e coppia) presentano tutti il medesimo componente in comune, che dunque è necessario studiare.



L'**Estensimetro** è il componente base dei sensori di forza ed è una resistenza, che chiameremo R_s, realizzata con un filo conduttore di basso coefficiente di temperatura e disposto su un supporto coibentato che viene incollato all'elemento soggetto a sollecitazione. L'estensimetro è scelto in modo tale che la resistenza R_s cambi linearmente nell'intervallo di deformazione ammissibile per l'elemento estensibile. Quando la forza (coppia) agisce sulla superfice deformandola, quest'ultima si ripercuote sull'estensimetro collocato su di essa. Le dimensioni del resistore cambiano e quindi provocano una variazione della resistenza elettrica. Per trasformare le variazioni di resistenza in un segnale elettrico, l'estensimetro viene inserito in un braccio di un ponte di Wheatstone che viene bilanciato in assenza di sollecitazioni sull'estensimetro stesso. La tensione di uscita al ponte vale:



$$V_o = \left(\frac{R_2}{R_1 + R_2} - \frac{R_s}{R_3 + R_s}\right) V_i$$

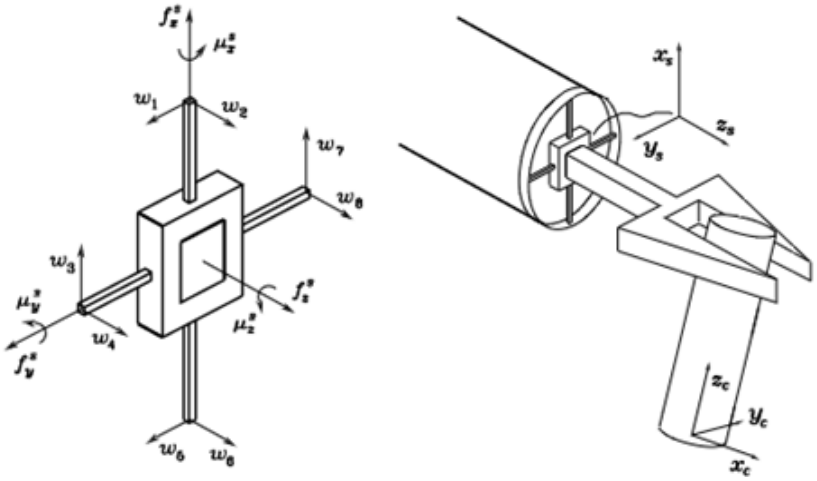
Il problema principale dell'estensimetro è che variazioni di temperatura modificano le sua dimensioni senza che si applicata alcuna sollecitazione esterna. Un modo per eliminare l'influenza della temperatura è quello connettere l'estensimetro di misura su un ramo del ponte e impiegarne un altro su un ramo adiacente a quest'ultimo tale anch'esso da non essere sottoposto ad alcuna sollecitazione ma esposto alla stessa temperatura. Per le caratteristiche del ponte di Wheatstone le deformazioni dovute alla temperatura producono la stessa variazione di resistenza in modulo ma di segno discorde così da elidersi.

Vediamo alcuni tipici utilizzi sensori di forza che impiegano gli estensimetri:

- **Sensori di Coppia all'Albero**: consentono di ottenere una misura diretta di coppia grazie all'impiegono di estensimetri montati su un apparato estensibile interposto tra il motore e il giunto, ad esempio un albero cavo. Tale apparato deve:
  - Avere bassa rigidezza torsionale;
  - Avere elevata rigidezza flessionale;
  - Misurare un rapportorto di proporzionalità tra la coppia applicata e deformazione indotta.

  A questo punto collegando gli estensimetri e i relativi ponte di Wheastone all'albero cavo sarà possibile misurare il segnale sbilanciato risultante che sarà proprozionale alla coppia applicata. La necessità di una misura diretta di coppia, anziché una indiretta valutando corrente di armatura di un servomotore CC a magneti permanenti, si ha qualora si desideri una misurazione insensibile al cambiamento dei parametri relativi alla coppia. E' da notare però che la coppia misurata è quella trasmessa dal servomotore al giunto ma non corrisponde alla C_m in quanto non porta in conto coppia di inerzia, attrito del motore e delle trasmissione a monte del punto di misura.

- **Sensore di Forza del Polso**: viene realizzato con una configurazione detta a **Croce di Malta** (figura di sinistra). Su ciascun ramo della croce sono posti una coppia di estensimetri per un totale di otto, ciascuno dei quali collegati ad un relativo ponte di Wheastone. Infine, la croce di Malta viene posta come *collegamento* tra il polso del manipolatore e l'end-effector. Così facendo, oltre a misurare le tre componenti di forza e le tre di momento, è possibile anche disaccopiarle tra loro.



La **Matrice di Calibrazione** è la matrice che mette in relazione le misurazioni della deformazione, che corrispondono alle uscite ω_i con $i = 1, \dots, 8$ degli otto ponti di wheastone, con le componenti della forza $f_c = (f_x, f_y, f_z)$ e di momento $\mu_c = (\mu_x, \mu_y, \mu_z)$:

$$\begin{bmatrix} f_x^s \\ f_y^s \\ f_z^s \\ \mu_x^s \\ \mu_y^s \\ \mu_z^s \end{bmatrix} = \begin{bmatrix} 0 & 0 & c_{13} & 0 & 0 & 0 & c_{17} & 0 \\ c_{21} & 0 & 0 & 0 & c_{25} & 0 & 0 & 0 \\ 0 & c_{32} & 0 & c_{34} & 0 & c_{36} & 0 & c_{38} \\ 0 & 0 & 0 & c_{44} & 0 & 0 & 0 & c_{48} \\ 0 & c_{52} & 0 & 0 & 0 & c_{56} & 0 & 0 \\ c_{61} & 0 & c_{63} & 0 & c_{65} & 0 & c_{67} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{bmatrix}$$

Tuttavia, queste sono le forze f_s^s e i momenti μ_s^s visti dal sensore, mentre per ottenere le medesime grandezze riferite alla pinza, ovvero f_c^c e μ_c^c, dobbiamo impiegare la matrice di trasformazione in evidenza nella seguente relazione:

$$\begin{bmatrix} f_c^c \\ \mu_c^c \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_s^c} & \boldsymbol{O} \\ \boldsymbol{S(r_{cs}^c)R_s^c} & \boldsymbol{R_s^c} \end{bmatrix} \begin{bmatrix} f_s^s \\ \mu_s^s \end{bmatrix}$$

Infine è da notare come è possibile impiegare la matrice di calibrazione vista poc'anzi per discriminare la qualità dei sensori di forza. Infatti, un sensore risulterà essere tanto più preciso quanto più sono presenti zeri o comunque valori molti piccoli nella propria matrice di calibrazione in quanto sarà in grado di disaccoppiare meglio le componenti di forza e momenti.

4.3.2 Sensori di Distanza

I **Sensori di Distanza** *rilavano la presenza di un oggetto nell'area di lavoro ed eventualmente misurano la distanza dal robot lungo una determinata direzione*. Tali sensori sono caratterizzati da un **Range Sensibile**, la quale corrisponde alla distanza entro cui essi riescono a rilevare oggetti. Le due principali categorie di sensori di distanza sono:

- **Sensori di Prossimità**: sono sensori in grado di rilevare la presenza di oggetti nelle vicinanze della parte sensibile del sensore senza la necessità di contatto fisico con essi.
- **Sensori di Portata**: hanno la medesima funzione dei sensori di prossimità ma forniscono una misurata strutturata (più complessa e ricca di informazioni), grazie alla quale è possibile costruire mappe dell'ambiente, sono utili alla navigazione e alla collision detection.

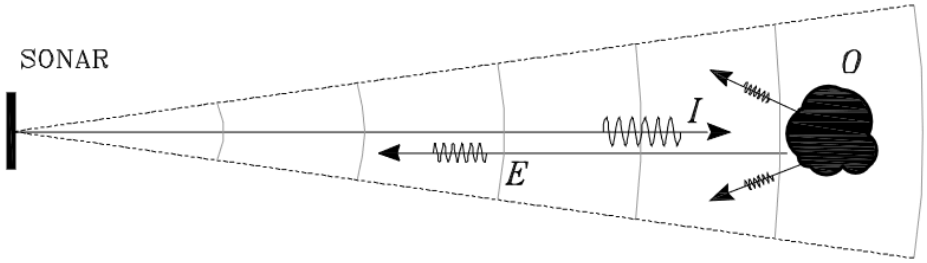
Poiché i sensori di portata vengono maggiormente impiegati nella robotica ci focalizziamo su questa categoria e in particolare su:

- **Sonar**:

E' un sensore che impiega le *onde sonore* per rilevare la presenza/distanza degli oggetti ed eventualmente anche dei dati direzionali qualitativi su di essi. Ciò è possibile in quanto la velocità nel suono in un mezzo come l'aria o l'acqua è nota, per cui medianti semplici formule è possibile calcolare la distanza. Il sonar è costituito da due elementi principali:

- Trasduttore che viene fatto vibrare e trasforma l'energia acustica in energia elettrica e viceversa;
- Circuiteria per l'eccitazione del trasduttore e la rilevazione del segnale riflesso.

Dunque, supponendo di avere una situazione come in figura con un sonar davanti al quale è presente un oggetto, studiamo il suo funzionamento.



Il sonar genera dei suoni che, in ambito robotico, hanno frequenze comprese nell'intervallo $[20,200]\ kHz$, per cui le onde sonore si possono considerare concentrate in un volume conico la cui ampiezza (tipicamente sopra i 15 gradi) dipende da:

- Diametro del sensore;
- Frequenza.

Un certo impulso I andrà ad intercettare l'oggetto O, il quale rifletterà le onde sonore ed un certo eco E andrà ad intercettare il sonar. A questo punto il sonar calcola la distanza dell'oggetto mediante la seguente formula: $\boldsymbol{d_O = c_s t_v / 2}$

Dove:

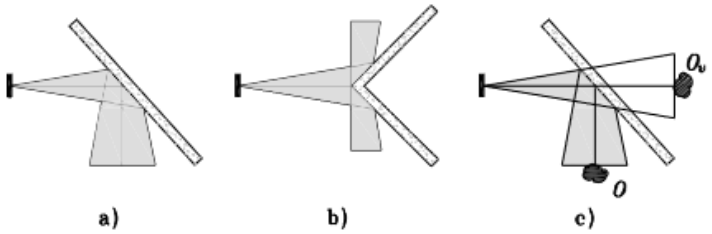
- $c_s \approx 20.05\sqrt{T + 273.16} =$ velocità di propagazione dell'onda sonora nell'aria o nell'acqua ed è funzione della temperatura T;
- $t_v =$ tempo di volo dell onda sonora, ovvero tempo che intercorre tra l'emissione dell'impulso e la ricezione dell'eco.

I principali vantaggi dei sonar che ne hanno portato un ampio utilizzo nella robotica mobile e subaquea sono:

- ✓ Basso costo;
- ✓ Pesi leggero;
- ✓ Basso consumo energetico;
- ✓ Basso sforzo computazionale;
- ✓ Funzionano in scenari con bassa visibilità.

Tuttavia, essi presentano degli svantaggi non trascurabili come:

- o Il tempo di latenza nella commutazione da trasmettitore a ricevitore, il quale dipende dalla durata dell'impulso e dall'inerzia meccanica del trasduttore. Ciò influisce sulla minima distanza che siamo in grado di misurare;
- o Oltre ai limiti sulla distanza minima, abbiamo limiti anche su quella massima. Se aumentiamo la frequenza avremo una minore ampiezza del cono ma un incremento della sua lunghezza e una migliore risoluzione. Risultato di ciò potremo effettuare misurazioni sempre più lontane dal sonar. Il problema però è che frequenze troppo alte possono esasperare fenomeni di assorbimento dipendenti dalla caratteristiche delle superifici che generano eco. Tali fenomeni riducono la potenza del segnale trasmesso con il quadrato della distanza coperta dall'onda sonora. In realtà anche frequenze troppo basse sono un problema perché portano interferenze con il segnale riflesso. A causa dei limiti citati fin'ora spesso si impiegano coppie di sonar dove uno è l'emettitore e l'altro è il ricevitore;
- o I sonar possono commettere errori a causa delle proprietà di alcuni materiali. Per esempio superfici liscie, ovvero le cui irregolarità sono paragonabili alla lunghezza d'onda corrispondente alla frequenza dell'onda sonora, oppure angoli eccessivi (figure a e b) possono produrre echi non rilevabili dal sensore. Oppure, come nel caso della figura c, la presenza di superfici a specchio può causare riflessioni multiple con conseguenti errori di misura o falsi rilevamenti.



Vediamo le due tipologie di sonar maggiormente diffuse, le quali impiegano entrambe le onde sonore ma con principi fisici differenti:

a) **Trasduttori Sonar Piezoelettrici**: sono costituiti da cristalli che si deformano sotto l'azione di un campo elettrico e vibrano se eccitati alla frequenza di risonanza del materiale, il quale genera una onda acustica. L'efficienza della corrispondenza acustica di questi trasduttori con fluidi comprimibili come l'aria è piuttosto bassa. Spesso sul cristallo è montata una tromba conica concava per abbinare acusticamente l'impedenza cristallo-acustica a quella dell'aria. Essendo di tipo risonante, questi trasduttori sono caratterizzati da una larghezza di banda piuttosto ridotta e presentano una notevole inerzia meccanica che limita fortemente la portata minima rilevabile, giustificando così l'utilizzo di due distinti trasduttori come trasmettitore e ricevitore;

b) **Trasduttore Sonar Elettrostatici** (o capacitivi): sono condensatori la cui capacità varia muovendo e/o deformando una delle armature. Quando funzione da emettitore la membrana del trasduttore viene fatta vibrare applicando una sequenza di impulsi elettrici attraverso il condensatore. Le oscillazioni elettriche generano, per effetto del campo

elettrico indotto, una forza meccanica che fa vibrare la piastra mobile. Invece, quando funzione da ricevitore, supponendo di avere una carica costante, la variazione di capacità causata dall'onda sonora produce una variazione proporzionale della tensione ai capi del condensatore. I sonar capacitivi sono caratterizzati da una elevata ampiezza di banda, elevata sensibilità e bassa inerzia. Inoltre, presentano un buon accoppiamento con l'aria. Il problema dei capacitivi è che non possono lavorare a frequenze molto elevate con conseguente peggioramento della risoluzione oltre che di utilizzare una elettronica di controllo complessa.

- **Laser**:

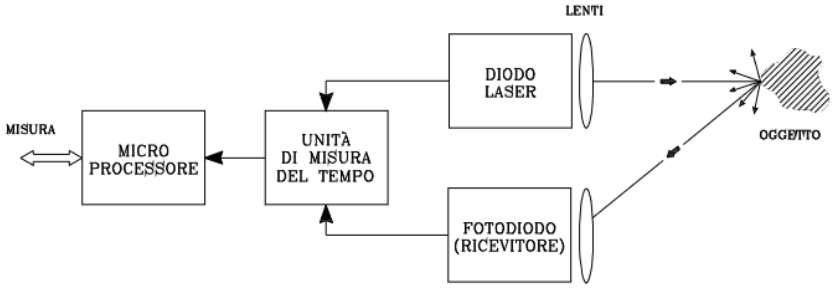
E' un sensore che misura la distanza e la posizione di un oggetto attraverso l'utilizzo della luce. In particolare viene impiegato un raggio laser per diversi motivi:

- ✓ Possono facilmente generare fasci luminosi con sorgenti leggere;
- ✓ I raggi infrarossi possono essere utilizzati in modo discreto;
- ✓ Mettono a fuoco bene per fornire fasci stretti;
- ✓ Le sorgenti a frequenza singola consentono un più facile filtraggio del rifiuto delle frequenze indesiderate e non si disperdono dalla rifrazione tanto quanto le sorgenti a spettro completo.

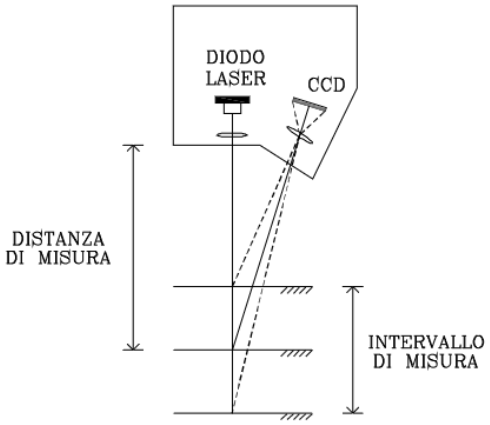
Esistono due tipi di sensori di distanza basati su laser di uso comune:

A. Sensore Laser a Tempo di Volo: misurano la distanza tramite la misura del tempo di volo del laser per andare e tornare. Il tempo di percorrenza moltiplicato per la velocità della luce (opportunamente regolata per la temperatura dell'aria) fornisce la misura della distanza. I sensori laser a tempo di volo possono effettuare sia una misura mediante un singolo raggio, ma è possibile far spaziare quest'ultimo lungo la scena per avere maggiori informazioni su di essa. Il sensore laser a tempo di volo sono caratterizzati da:

- ➢ **Intervallo di Latenza**: ovvero l'intervallo nel passaggio da emettitore a ricevitore come per i sonar in cui il sensore non può effettuare misure. Perciò l'accuratezza è limitata dal minimo intervallo di tempo osservabile. Tale problema può essere risolto mediante l'utilizzo di laser a costo maggiore oppure una coppia di laser dove uno fa solo da emettitore e l'altro da ricevitore;
- ➢ **Intervallo di Ambiguità**: il sensore emette impulsi di luce e calcola la distanza media dell'oggetto dal momento di ritorno degli impulsi. Tipicamente per semplificare l'elettronica di rilevamento di questi sensori, il ricevitore accetta solo segnali che arrivano entro il tempo Δt, ma questa finestra temporale potrebbe anche osservare impulsi precedenti riflessi da superfici più distanti. Da ciò ne deriva che una misura sfalsata è multiplo della distanza $c\Delta t/2$, dove c è la velocità della luce.



B. Sensore Laser a Triangolazione: sono sensori laser basati sul principio fisico della triangolazione geometrica. Il sensor già nativamente non ha problemi di latenza nello switch da emettitore a ricevitore in quanto solo due elementi separati. In particolare, il ricevitore, chiamato CCD, è una matrice di pixel. L'emettitore emette il fascio luminoso, il quale riflette sull'oggetto e la luce riflessa colpisce il CCC. Sulla matrice di pixel si analizzano le aree maggiormente colpite dalla luce. Poiché è nota la posizione relativa tra emettitore e ricevitore, sarà possibile ricavare una misura della distanza dell'oggetto. Uno dei vantaggi di questo tipo di laser è la possibilità di controllare il raggio laser utilizzando filtri sulla lunghezza d'onda, raggi multipli per misure simultanee, osservazioni selettive. Tuttavia, a causa delle riflessioni, questi sensori sono rischiosi per gli occhi e quindi vanno utilizzati con le corrispondenti protezioni.



4.3.3 Sensore di Visione

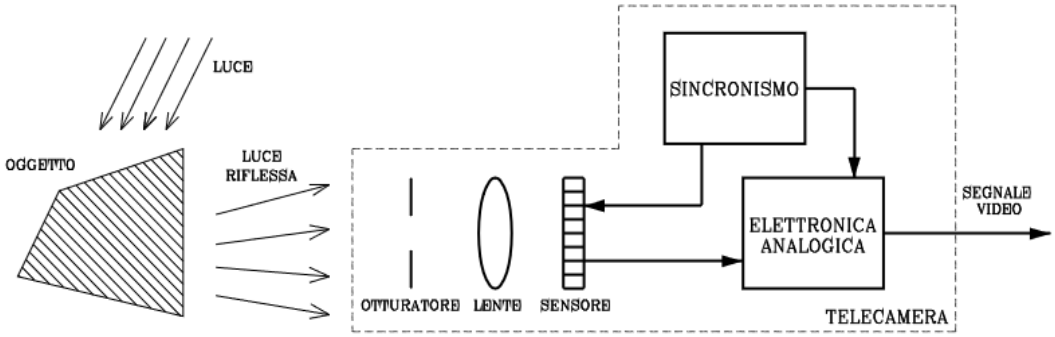
Un **Sensore di Visione** è un sensore costituito da una matrice rettangolare di pixel, ovvero di elementi fotosensibili, che colpiti dalla luce realizzano una conversione del segnale luminoso in un segnale elettrico. Di fatto questo sensore si basa proprio sul principio fisico per cui, quando la luce colpisce l'elemento fotosensibile, quest'ultimo genera elettroni. Il risultato della conversione può essere ottenuto in modo differenti a seconda delle tecnologia impiegata per il sensore di visione, per esempio abbiamo:

A. **CCD (Charge Couled Device)**: accumula carica secondo l'*integrale* dell'illuminazione incidente. I pixel dunque eseguono una misura di volume e ciò comporta che in alcuni casi la quantità di carica del singolo pixel può trabordare quest'ultimo andando ad influenzare quelli limitrofi. Risultato di ciò l'immagine presenterà un effetto di sfocatura.

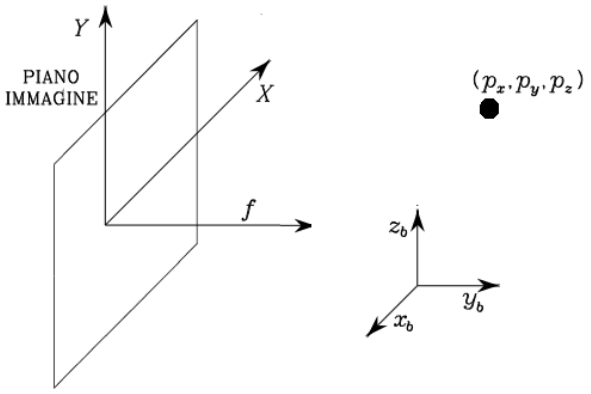
B. **CMOS (Complementary Metal Oxide Semiconductor)**: rilascia carica *proporzionalmente* alla quantità di illuminazione incidente. I pixel dunque eseguono una misura di quantità di fotoni per cui essa non può influenzare i pixel limitrofi come avviene nei CCD.

Studiamo nel dettaglio il sensore di visione nell'ambito del suo utilizzo in una telecamera. Una telecamera è costituta, come si può osservare in figura, da diversi elementi:

- **Otturatore**: è un dispositivo meccanico o elettronico che esegue un campionamento temporale dell'immagine. In altre parole, consente il passaggio della luce solo per un determinato periodo di tempo;
- **Lente**: è un disco si vetro il cui scopo è concentrare la luce;
- **Sensore di Visione**: esegue il campionamento spaziale dell'immagine;
- **Elettronica Analogica**: converte il segnale elettrico in segnale video;
- **Sincronizzatore**: come dice il nome serve a sincronizzare i due precedenti elementi.



A questo punto ci chiediamo come sia possibile rappresentare un generico punto $p = [p_x \ p_y \ p_z]^T$ dello spazio, di cui quindi conosciamo le coordinate rispetto ad una terna base (x_b, y_b, z_b), ad essere mappato sul piano di pixel che chiameremo d'ora in poi **Piano dell'Immagine**. Anzitutto ricordiamo che tra piano dell'immagine e il punto p in questione è presente la lente, il cui scopo è mettere a fuoco la luce riflessa dall'oggetto sul piano stesso.



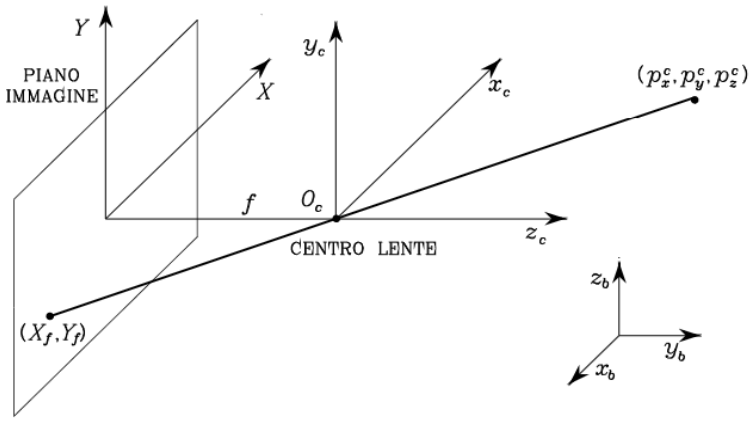
Consideriamo una terna $O_c x_c y_c z_c$ la cui origine coincide con il centro della lente della telecamera. Per conoscere le coordinate del punto p rispetto alla terna della lente dobbiamo applicare una trasformazione omogenea: $\tilde{p}^c = T_b^c \tilde{p}$

Ora che conosciamo le coordinate del punto p nel riferimento della lente e sapendo la lunghezza focale f della lente, ovvero la distanza tra piano immagine e centro lente, possiamo calcolare le coordinate (X_f, Y_f) del suddetto punto sul piano dell'immagine. In particolare, possiamo ricavare (X_f, Y_f) come due possibili trasformazioni:

A questo punto l'obiettivo è ricavare le coordinate X_f e Y_f del punto p^c sul piano immagine (CCD o CMOS). Ci sono due tipologie di trasformazioni che ci consentono di ottenere ciò:

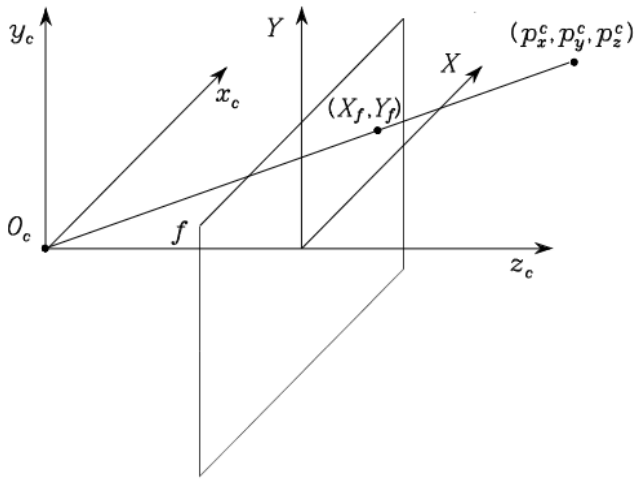
A. Trasformazione Prospettica: il piano dell'immagine si trova prima della terna c della lente alla distanza f. Le coordinate del punto sul piano immagine sono date dalle relazioni di fianco, nelle quali è presente il segno negativo in quanto sul piano dell'immagine l'oggetto viene capovolto.

$$\boldsymbol{X_f = -f\frac{p_x^c}{p_z^c}} \qquad \boldsymbol{Y_f = -f\frac{p_y^c}{p_z^c}}$$



B. Trasformazione Prospettica Frontale: il piano dell'immagine si trova dopo la terna c della lente alla distanza f. In tal caso le coordinate del punto di sul piano immagine sono ottenute mediante le stesse relazioni viste poc'anzi ma prive del segno negativo.

$$\boldsymbol{X = f\frac{p_x^c}{p_z^c}} \qquad \boldsymbol{Y = f\frac{p_y^c}{p_z^c}}$$



Queste trasformazioni sono però ideali in quanto, nella realtà, le lenti presentano delle imperfezioni e distorsioni che vengono corretti in diversi modi.

Altro problema che si pone è che la matrice di pixel esegue un campionamento spaziale dell'immagine con un'unità di campionamento che è il pixel. Allora le coordinate di un punto nel piano dell'immagine devono essere espressi in pixel. Invece, le coordinate (X_f, Y_f) ricavate poc'anzi presentano come unità di misura i metri. Motivo per cui dovremo effettuare una conversione da $m \rightarrow pixel$ mediante le seguenti relazioni:

$$X_I = a_x X_f + X_0 = a_x f \frac{p_x^c}{p_z^c} + X_0$$
$$Y_I = a_y X_f + Y_0 = a_y f \frac{p_y^c}{p_z^c} + Y_0$$

Dove:

- $a_x, a_y =$ sono fattori di scala legati alla lente e sono necessari a causa delle dimensioni finite della matrice di pixel;
- $X_0, Y_0 =$ sono offset che tengono conto della posizione dell'origine del sistema di coordinate dei pixel rispetto all'asse ottico. Tale distanza dipende di fatto dal tipo di tecnologia impiegata se CCD oppure CMOS.

Questa trasformazione non lineare (X_I, Y_I) può essere scritta in forma lineare ricorrendo alla rappresentazione omogenea del punto (x_I, y_I, z_I) tramite le relazioni di fianco in cui λ è un parametro maggiore di zero. Otteniamo così le seguenti uguaglienze:

$$X_I = \frac{x_I}{\lambda} \qquad Y_I = \frac{y_I}{\lambda}$$

$$\begin{bmatrix} x_I \\ y_I \\ \lambda \end{bmatrix} = \lambda \begin{bmatrix} X_I \\ Y_I \\ 1 \end{bmatrix} = \Omega\Pi \begin{bmatrix} p_x^c \\ p_y^c \\ p_z^c \end{bmatrix} = \Omega\Pi T_b^c \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

In pratica, questa relazione ci dice che le coordinate in pixel linearizzate del punto p sul piano dell'immagine (primo membro) possono essere ricavate moltiplicando le coordinate di quest'ultimo per la **Matrice di Calibrazione $\boldsymbol{\Theta}$**, la quale si calcola mediante la seguente relazione:

$$\boldsymbol{\Theta = \Omega\Pi T_b^c} = \begin{bmatrix} \boldsymbol{f_{a_x}} & \mathbf{0} & \boldsymbol{X_0} \\ \mathbf{0} & \boldsymbol{f_{a_y}} & \boldsymbol{Y_0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{f_{a_z}} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix} \boldsymbol{T_b^c}$$

Dove in particolare:

- $\Omega =$ matrice dei parametri intriseci, la quale tiene conto (a_x, a_y, X_0, Y_0, f);
- $\mathrm{T}_b^c =$ matrice dei parametri estrinseci in T_c^b, la quale dipende dalla posizione relativa e l'orientamento della telecamera rispetto alla terna base b. Tramite appositi strumenti è possibile calibrare la matrice dei parametri estrinseci attraverso l'identificazione della posizione e orientamento relativo della terna c della lente e la terna di riferimento b.

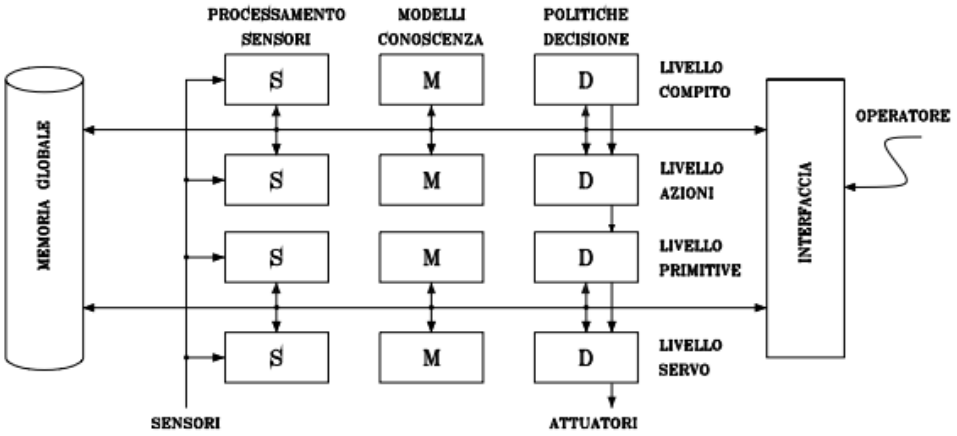
5. Unità di Governo

5.1 Architettura Funzionale

L'**Unita di Governo** di un sistema robotico, al fine di supervisionare le attività di quest'ultimo, dovrebbe essere dotato di una serie di strumenti che conferiscono le seguenti capacità:

- ✓ **Manipolazione**: abilità di muovere oggetti fisici nell'ambiente di lavoro;
- ✓ **Sensorialità**: abilità di ottenere informazioni sullo stato del sistema e dell'ambiente di lavoro;
- ✓ **Intelligenza**: abilità di utilizzare le informazioni per modificare il proprio comportamento;
- ✓ **Processing dei Dati**: abilità di elaborare e rendere disponibili dati relativi all'attività del sistema.

L'implementazione di tali abilità in un sistema robotico può essere ottenuta mediante l'utilizzo di un modello di riferimento costruito come *struttura gerarchica* come mostrato in figura.



Nello schema in figura sono presenti una serie di livelli, ognuno con uno specifico scopo, collegati tra di loro mediante flussi di dati:

- ➢ I livelli superiori sono orientati alla pianificazione logica dell'azione, per cui i flussi di dati scambiati con i livelli inferiori sono indicazioni e comandi.
- ➢ I livelli inferiori orientati all'esecuzione del movimento fisico, per cui i flussi di dati scambiati con i livelli superiori sono misurazioni e risultati di azioni.

Ciascun livello a sua volta sarà costituito da tre moduli:

- **Modulo Sensoriale**: gestisce i dati sensoriali mediante funzioni di acqusizione ed elaborazione dei dati nello spazio e del tempo con il fine ultimo di riconoscere lo stato del sistema e la caratteristiche ambientali. Ciascun modulo sensoriale gestisce dei dati che sono comprensibli dai corrispettivi moduli del medesimo livello di appartenenza;
- **Modulo Modellistica**: dedicato alla conoscenza del mondo di riferimento. Il modulo di modellistica predispone dei modelli del sistema e dell'ambiente le cui informazioni vengono aggiornate dal modulo sensoriale;
- **Modulo Decisionale**: dedicato alla decisione dell'azione e all'attivazione delle funzioni richieste. Inoltre, tali moduli eseguono la suddivisione dei compiti di alto livello in azioni di basso livello sotto due aspetti:
  - Temporalmente: realizzando delle azioni sequenziali;
  - Spazialmente: realizzando azioni concorrenti.

  E' da notare che, l'appartenenza ad un certo livello della gerarchia non è direttamente riconducibile ai moduli sensoriali e di modellistica, poiché compiti dello stesso modulo per

diverse esigenze possono essere realizzati a diversi livelli. Invece, la gestione di certe azioni elementari per la pianificazione ed esecuzione dei compiti è riconducibile solo ad un particolare modulo decisionale di un determinato livello.

L'architettura appena descritta necessita di una interfaccia operatore *ad ogni livello* della gerarchia. In questo modo l'operatore può:

- Essere informato sullo stato del sistema dai moduli sensoriali;
- Mediante idonei strumenti, arricchire i moduli di modellazione;
- Intervenire e supervisione sui moduli decisionali.

Ultimo elemento fondamentale dell'architettura è una memoria globale che contenga tutta la mole di dati gestita da tutti gli elementi citati fin'ora.

Prima di analizzare le funzionalità dei diversi livelli è interessante osservare come più scendiamo nella gerarchia e più i dati risultano essere meno complessi e strutturati. Di contro però, al fine di rispettare i vincoli dinamici del manipolatore, si avranno vincoli temporali più stringenti che possiamo definire realtime. I livelli della gerarchia sono i seguenti:

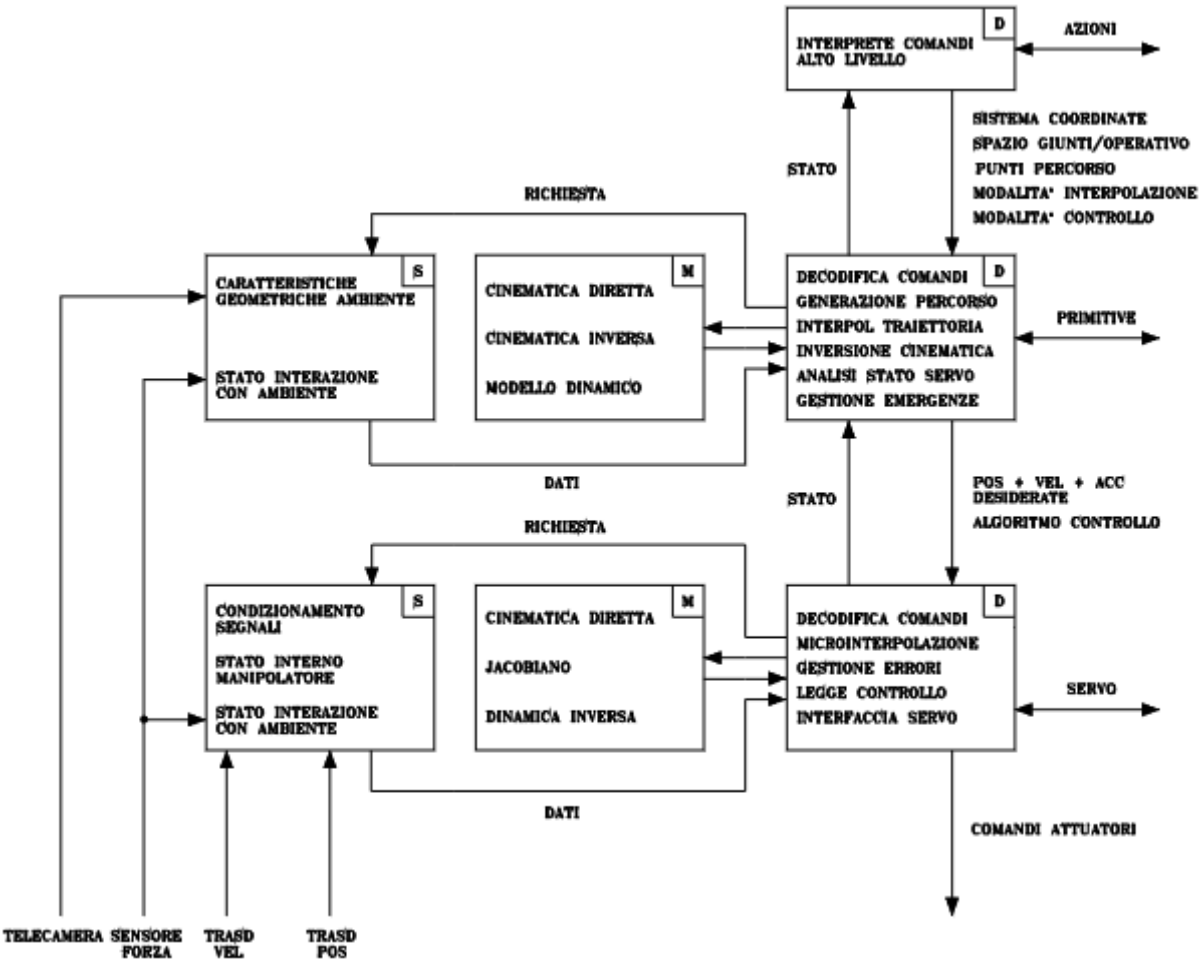
- **Livello Compito**: l'utente specifica il compito che il robot deve svolgere ad un alto livello di astrazione. Dopodichè, sulla base dei modelli forniti dal modulo di modellistica le cui informazioni sono aggiornate dal modulo sensoriale, tale compito viene analizzato e scomposto in una sequenza di azioni elementari coordinate spazialmente e temporalmente.
- **Livello Azione**: ad ogni azione elementare viene associata una sequenza di configurazioni intermedie che definiscono il percorso. Il modulo decisionale sceglie una particolare sequenza da attuare sulla base di un confronto tra quest'ultima e le informazioni provenienti dal modulo di modellistica del medesimo livello. In questo modo viene confermata la fattibilità dell'azione in termini di controllo collisione, movimento in prossimità di configurazioni singolari e di limiti meccanici dei giunti;
- **Livello Primitive**: sulla base della sequenza di configurazioni ricevute dal livello di azione, vengono calcolate le traiettorie di movimento ammissibili e viene decisa, sulla base del modello dinamico e delle informazioni del modulo sensoriale, la strategia di controllo. La traiettoria del movimento viene interpolata in modo da generare i *riferimenti* per il livello inferiore;
- **Livello Servo**: grazie ai riferimenti provienienti dal livello superiori, vengono calcolati i segnali errore, i quali misurano lo scostamento tra i riferimenti e i valori effettivi della grandezze controllate. I segnali errore vengono impiegati dagli algoritmi di controllo implementati a questo livello per generare in uscita i segnali di pilotaggio ai servomotori dei giunti. In particolare, il modulo decisionale elabora la legge di controllo sfruttando:
  - **Modulo di Modellistica**: il quale, grazie al modello dinamico del manipolatore, elabora i termini della legge di controllo in funzione della configurazione corrente;
  - **Modulo Sensoriale**: il quale fornisce misurazioni dei sensori propriocettivi sia per aggiornare i termini della legge di controllo che il modulo di modellistica deve calcolare che per il modulo decisionale circa il calcolo dei segnali errore.

  Inoltre, il modulo decisionale, per sfruttare appieno la caratteristica dinamica degli azionamenti, esegue una microinterpolazione sulla traiettoria di riferimento. Un aspetto fondamentale da tener presente è che a livello servo la frequenza di campionamento deve essere dell'ordine del kHz affinchè il controllo possa reagire velocemente in presenza di interazione con l'ambiente.

5.2 Architettura Funzionale per Robot Industriali

Nei robot industriali, per limiti tecnologici e di costo, non vengono implementati tutti i livelli e tutti i rispettivi moduli con le relative funzionalità prima trattate. Per esempio, i moduli di modellazione e sensori sono sempre presenti al livello più basso, a causa dei requisiti esigenti a livello di servo per robot ad alte prestazioni dinamiche da impiegare anche in applicazioni relativamente semplici. Andiamo ora ad analizzare le differenze dall'architettura precedente:

- **Livello Compito**: è assente in quanto ancora non esistono pacchetti software applicativi efficaci ed affidabili che supportano le complesse funzionalità di questo livello**;**
- **Livello Azione**: è presente *solo il modulo decisionale* che ha il solo scopo di interpretare i comandi ad alto livello dell'operatore. A quest'ultimo è affidata la scomposizione del compito e quindi sono assenti sia il modulo di modellistica che sensoriale. L'eventuale verifica della fattibilità dell'azione viene spostata al livello primitivo dove esiste un modulo di modellazione;
- **Livello Primitive**: tipicamente è presente il modulo di modellazione, mentre quello sensoriale solo per applicazioni di interazione tra robot e ambiente poco strutturato;
- **Livello Servo**: è praticamente invariato rispetto all'architettura viste in precedenza.



5.3 Ambiente di Programmazione

La programmazione di un sistema robotico richiede la definizione di un ambiente di programmazione che, non solo deve presentere tutte le classiche caratteristiche di questi ambienti, ma anche delle ulteriori dovute fondamentalmente all'interazione tra il robot e il mondo esterno. Ricordiamo infatti che il robot esegue i suoi compiti in un ambiente di lavoro reale il quale, seppur modellato in maniera molto accurata, presenterà sempre delle situazioni impreviste. Conseguenza di ciò, un ambiente di programmazione per robot dovrebbe essere dotato delle seguenti caratteristiche:

- Sistema operativo in tempo reale;
- Modellazione dell'ambiente di lavoro;
- Lettura dei dati sensoriali;
- Controllo del movimento;
- Capacità di rilevamento degli errori,
- Ripristino corrette funzioni operative;

I requisiti su un ambiente di programmazione possono essere estratti proprio dal'architettura funzionale gerarchica:

- **Modulo Sensoriale**:
  - **Livello Compito**: strumenti che consentano il riconoscimento degli oggetti presenti nella scena;
  - **Livello Azione**: le caratteristiche geometriche degli oggetti interessati all'azione devono essere estratte da dati sensoriali di alto livello;
  - **Livello Primitive**: i dati sensoriali devono essere espressi nei relativi quadri di riferimento;
  - **Livello Servo**: condizionamento dei dati sensoriali propriocettivi *in tempo reale*.
- **Modulo Modellistica**:
  - **Livello Compito**: la programmazione ambiente dovrebbe assumere le funzioni di un sistema esperto;
  - **Livello Azione**: strumenti che consentano la simulazione del sistema e la modellazione CAD di oggetti elementari;
  - **Livello Primitive**: devono essere effettuate trasformazioni di coordinate;
  - **Livello Servo**: calcolo numerico in linea dei modelli utilizzati dagli algoritmi di controllo deve essere effettuato sulla base di dati sensoriali.
- **Modulo Decisionale**:
  - **Livello Compito**: l'ambiente di programmazione dovrebbe consentire la gestione di processi concorrenti e dovrebbe essere dotato di strumenti per testare, individuare e rimuovere gli errori da un programma (debugger) a un livello altamente interattivo;
  - **Livello Azione**: dovrebbero essere disponibili opzioni di sincronizzazione dei processi per implementare cicli annidati, calcolo parallelo e sistema di interruzione;
  - **Livello Primitive**: deve essere presente il condizionamento logico;
  - **Livello Servo**: capacità di calcolo in linea per generare i segnali di guida per il sistema meccanico.

Analizziamo tre diverse ambienti di programmazione che nel corso del tempo di sono sviluppati, andando di fatto ad inglobare le funzionalità della generazione precedente:

1. **Insegnamento per Esposizione**: come suggerisce il nome, è un approccio di programmazione ai robot caratterizzata da una fase "espositiva" dei movimenti che il manipolatore dovrà riprodurre. Tipicamente è sufficiente un operatore con delle

conoscenze basilari di programmazione che, mediante un apposito dispositivo, controlla manualmente il robot. Il manipolatore in questa fase espositiva salva in memoria i valori dei trasduttori di posizione dei giunti e li utilizzera in seguito come riferimenti per riprodurre fedelmente il movimento impartito. Questo ambiente presenta due svantaggi:

- Si ha la necessità di avere a disposizione fisicamente il robot e quindi di doverlo separare dalla produzione;
- Considerando l'architettura funzionale di una unità di governo, l'operatore è in grado di accedere fino al livello primitive però potrà far eseguire al robot solo delle funzioni molto semplici come come saldatura a punti, verniciatura e pallettizzazione.

Da ciò si evince come l'insegnamento per esposizione può essere facilmente superato da ambienti di programmazione strutturati che consentono di programmare funzioni complesse grazie alla possibilità di:

- Acquisire una postura;
- Calcolare la posa dell'end-effector mediante cinematica diretta;
- Calcolo dei servoriferimenti mediante cinematica inversa;
- Correzione di sequenza di movimento grazie a semplici editor di testo.

2. **Programmazione Orientata ai Robot**: gli ambienti di programmazione orientati ai robot hanno intrapreso due differenti strade:
   a. Sviluppare linguaggi ad hoc per applicazioni robotiche;
   b. Sviluppare librerie di programmi orientati ai robot che supportino linguaggi di programmazione standard.

   E' da specificare che la modalità di programmazione "insegnamento per esposizione" è stata integrata nella programmazione orientata ai robot vista l'elevata accuratezza nell'essere interfacciata con il mondo fisico. Il problema fondamentale della programmazione ai robot è che attualmente esistono solo linguaggi di programmazione proprietari piuttosto che linguaggi basati su standard univocamente accettati dall'intero settore. Mentre le caratteristiche comuni di tali linguaggi sono:

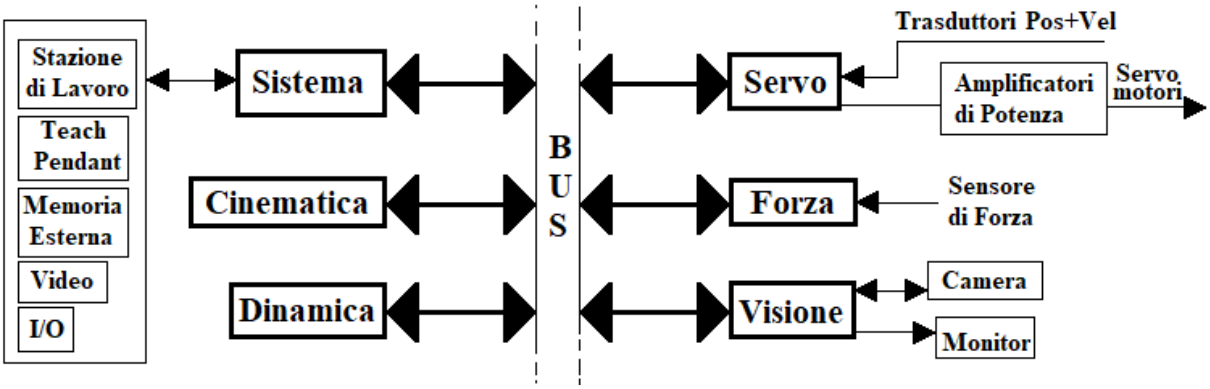
   - La presenza di un editor di testo;
   - Complesse strutture di rappresentazione dei dati;
   - Esecuzione di operazioni di algebra matriciale;
   - Capacità di calcolo parallelo;
   - Funzioni di PLC.

   Riprendendo l'architettura funzionale, l'operatore è in grado di accedere al livello azione. Poiché stiamo parlando di linguaggi strutturati, è necessario che l'operatore presenti delle competenze notevolmente superiori all'ambiente trattato precedentemente. Da ciò però derivano due vantaggi, il primo è che l'operatore non necessità fisicamente del robot in quanto può eseguire modifiche del programma offline e il secondo vantaggio è la possibilità di realizzare compito complessi per il manipolatore (interazione con altre macchine, assemblaggio, ecc.).

3. **Programmazione Orientata agli Oggetti**: è un approccio alla programmazione dei robot che consente all'operatore di accedere al livello compito, quindi quello con il più alto grado di astrazione, dell'architettura funzionale. Tale ambiente consente di specificare attività mediante istruzioni di alto livello che portano all'esecuzione di serie di azioni sugli oggetti presenti nella scena. Tuttavia, i linguaggi di programmazione ai robot di questa categoria sono ancora in fase di sviluppo e vengono applicati principalmente negli scenari di sistemi esperti o intelligenza artificiale.

5.4 Architettura Hardware

Dopo aver scelto di adottare l'architettura gerarchica per definire l'unità di governo del nostro robot industriale, il passo successivo è studiare la sua implementazione hardware, la quale dovrebbe avere come filosofia quella di sfruttare risorse computazionali distribuite ed interconnesse mediante opportuni canali di comunicazione. Circa quest'ultimo aspetto una possibile implementazione hardware dell'unità di governo è mostrata nella seguente figura, nella quale diversi moduli si affacciano al medesimo bus. Tuttavia, ciò non implica che tutti i moduli hanno la medesima frequenza di scambio dati. Infatti, il livello primitive e servo, poiché leggeno dati dai sensori propriocettivi, hanno vincoli di tipo real-time e quindi richiedono frequenza elevate tali da garantire le prestazioni dinamiche per il controllo (per esempio le collisioni). Infatti , la larghezza di banda del bus deve essere sufficientemente ampia da soddisfare i requisiti imposti dai vincoli in tempo reale.



Andiamo ad analizzare i moduli di questa architettura:

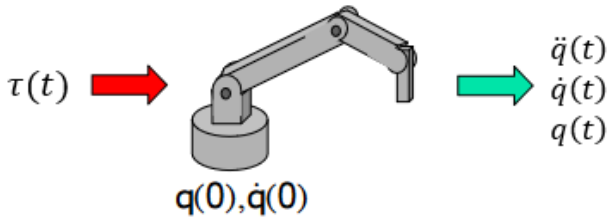
- **Sistema**: presenta una CPU dotati di microprocessore con co-processore matematico, una EPROM per il bootstrap, una RAM locale e condivisa con altre schede, una serie di porte di interfacciamento con il bus e infine registri, contatori e temporizzatori. Le principali funzioni del modulo sistema, oltre ovviamente al bootstrap, riguardano sostanzialmente tutta la parte di interfacciamento con l'utente (mediante le varie periferiche di I/O) e interpretazione di linguaggio di programmazione;
- **Cinematica**: le sue funzioni sono il calcolo delle primitive di movimento, cinematica diretta, inversa, gestione delle ridondanza, calcolo dello Jacobiano e test di fattibilità della traiettoria;
- **Dinamica**: si occupa principalmente del calcolo della dinamica inversa;
- **Servo**: le sue funzioni sono la microinterpolazione dei riferimenti, calcolo dell'algoritmo di controllo, interfacciamento con gli amplificatori di potenza, gestione dei dati dai sensori ed eventuale interruzione del movimento in caso di malfunzionamento;
- **Forza**: le sue funzioni sono il condizionamento dei dati forniti dal sensore di forza e la rappresentazione delle forze in un dato sistema di coordinate.
- **Visione**: si occupa dell'elaborazione dei dati forniti dalla fotocamera ed estrae caratteristiche geometriche dalla scena.

6. Dinamica

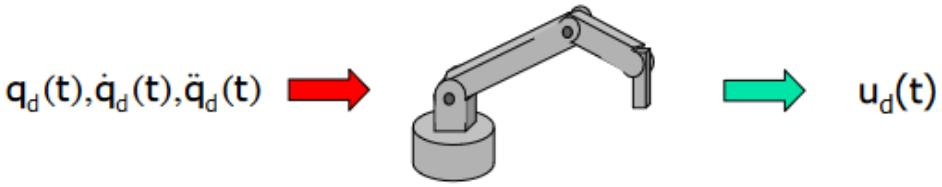
6.1 Caratterizzazione della Dinamica

La **Dinamica** *è lo studio delle equazioni del moto del manipolatore in funzione delle forze e dei momenti agenti su di esso*. Anche la dinamica, come la cinematica e la cinematica differenziale, si articola in due differenti studi:

- **Dinamica Diretta**: *date le forze generalizzate* $\tau(t), h_e(t)$ *e lo stato iniziale del manipolatore* $q(t_0), \dot{q}(t_0)$ *si vuole determinare* $\ddot{q}(t), \dot{q}(t), q(t)$. E' da specificare che in realtà noi calcoliamo algebricamente $\ddot{q}(t)$, mentre dall'integrazione di quest'ultima $\dot{q}(t)$ e $q(t)$ che invece definiscono lo stato del sistema. Supponendo di avere un manipolatore ad n giunti, il numero di operazioni per il calcolo della dinamica diretta è $O(n^2)$.



- **Dinamica Inversa**: *date* $\ddot{q}(t), \dot{q}(t), q(t)$ *e le wranch* $h_e(t)$ *si vuole determinare le* $\tau(t)$. Tipicamente essa viene impiegata per la pianificazione della traiettoria e l'implementazione dell'algoritmo di controllo. Inoltre, il calcolo delle coppie di controllo ci consente anche di stabilire la fattibilità della traiettoria ed eventualmente compensare termini non lineari del modello dinamico del manipolatore. E' da notare che, nella dinamica inversa non abbiamo alcuna condizione iniziale in quanto stiamo assegnando tutta la traiettoria per ogni istante $t > t_0$. Supponendo di avere un manipolatore ad n giunti, il numero di operazioni per il calcolo della dinamica inversa è $O(n)$.



E' da notare come apparentemente il problema di dinamica inversa sembrerebbe un problema di dinamica diretta e viceversa. Tuttavia, dobbiamo pensare che i dati a nostra disposizione, quindi $q_d, \dot{q}_d, \ddot{q}_d$, sono ciò che vogliamo ottenere dal robot e per farlo su quest'ultimo dobbiamo applicare in ingresso le coppie τ che calcoliamo con la dinamica diretta. Discorso al contrario vale invece per la dinamica diretta in cui si scambiano di posto causa ed effetto.

Entrambi i problemi in prima battuta potremmo affrontarli con soluzioni di natura sperimentale. Nella dinamica diretta, per esempio, si potrebbero applicare delle coppie e poi si misurano le variabili di giunti con gli encoder. Tuttavia, nel seguito, utilizzeremo un approccio di natura analitica che fa ampio uso del **Modello Dinamico del Manipolatore nello Spazio dei Giunti**, ovvero delle equazioni che descrivono il moto della struttura robotica. Capiamo perciò che il problema centrale si posta sulla determinazione di tale modello, il quale può essere ottenuto con due differenti formulazioni che analizzeremo nel dettaglio:

A. **Formulazione di Lagrange**;
B. **Formulazione di Newton-Eulero**.

6.2 Formulazione di Lagrange

6.2.1 Definizione

La **Formulazione di Lagrange** consente di ricavare il modello dinamico del manipolatore grazie all'impiego della funzione **Lagrangiana**, la quale è definita come la differenza tra l'energia cinetica $\mathcal{T}$ e l'energia potenziale $\mathcal{U}$: $\boldsymbol{\mathcal{L} = \mathcal{T} - \mathcal{U}}$

Entrambe le energie dovranno essere espresse in funzione delle cosidette **Coordinate Generalizzate** $[\boldsymbol{\lambda_1} \ \dots \ \boldsymbol{\lambda_n}]^{\boldsymbol{T}}$ del sistema che, nel caso di un robot manipolatore, corrispondono proprio alle variabili di giunto: $q = [q_1 \ \dots \ q_n]^T = [\lambda_1 \ \dots \ \lambda_n]^T$

Grazie a questo procedimento otterremo le equazioni del moto del modello dinamico nella seguente forma in cui ξ_i sono le forze non conservative (applicate dall'esterno) applicate sul sistema:

$$\boldsymbol{\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\lambda}_i} - \frac{\partial \mathcal{L}}{\partial \lambda_i} = \xi_i \qquad i = 1, \dots, n}$$

Il **Modello Dinamico** ottenuto dalla formulazione di Lagrange, a seguito della caratterizzazione completa delle precedenti equazioni, è il seguente:

$$\boldsymbol{B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + f_s(\dot{q}) + g(q) = \tau - J^T(q)h_e}$$

Dove:

- $\boldsymbol{B(q)\ddot{q}}$: sono i termini inerziali;
- $\boldsymbol{C(q,\dot{q})\dot{q}}$: sono i termini quadratici nelle velocità e dipendono appunto da $\dot{q}$ ma anche dalla configurazione q del manipolatore. In questi termini sono inclusi i contributi delle forze centrifughe e di Coriolis;
- $\boldsymbol{g(q)}$: sono i termini relativi alla forza di gravità e dipendenti esclusivamente dalla configurazione del manipolatore.

Mentre i seguenti termini sono tutte le **Forze Non Conservative**:

- $\boldsymbol{\tau}$: coppie di attuazione;
- $\boldsymbol{F_v\dot{q}}$: coppie di attrito viscoso;
- $\boldsymbol{f_s(\dot{q})}$: coppie di attrito statico;
- $\boldsymbol{-J^T(q)h_e}$: coppie di bilanciamento di forze di contatto.

Prima di procedere con la dimostrazione di questo modello mediante la formulazione di lagrange, concludiamo questa panoramica definendo alcune caratteristiche operative di questo approccio. La formulazione di Lagrange ha il pregio di essere sistematica e di facile comprensione, inoltre è in grado di fornire delle equazioni del moto in forma analitica/chiusa. Infine, si tratta delle formulazione più adatta in ambito controllistico poiché oltre ad evidenziare i termini inerziali, delle forze centrifughe, ecc. risulta essere efficace se c'è l'esigenza di portare in conto effetti meccanici complessi come le deformazioni eleastiche dei bracci.

Per concludere, se volessimo risolvere la cinema diretta mediante la formulazione di Lagrange, basta che riprendiamo il modello dinamico sopra citato ed eseguire alcuni passaggi:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + g(q) = \tau - J^T(q)h_e$$
$$B(q)\ddot{q} = \tau - J^T(q)h_e - C(q,\dot{q})\dot{q} - F_v\dot{q} - g(q)$$
$$B(q)\ddot{q} = \tau - [J^T(q)h_e + C(q,\dot{q})\dot{q} + F_v\dot{q} + g(q)]$$
$$B(q)\ddot{q} = \tau - \tau'(q,\dot{q})$$
$$\boldsymbol{\ddot{q} = B^{-1}(q)(\tau - \tau')}$$

Discretizziamo il sistema grazie alla conoscenza $\tau_k, q_k, \dot{q}_k$, dai quali ricaviamo $\ddot{q}_k$. Integrando con un certo passo di integrazione Δt, ricavo $\dot{q}_{k+1}$ e q_{k+1}. Il fatto che dobbiamo calcolare l'inversa di una matrice ci fa anche capire perché la dinamica diretta è computazionalmente più onerosa della dinamica inversa.

6.2.2 Dimostrazione del Modello Matematico

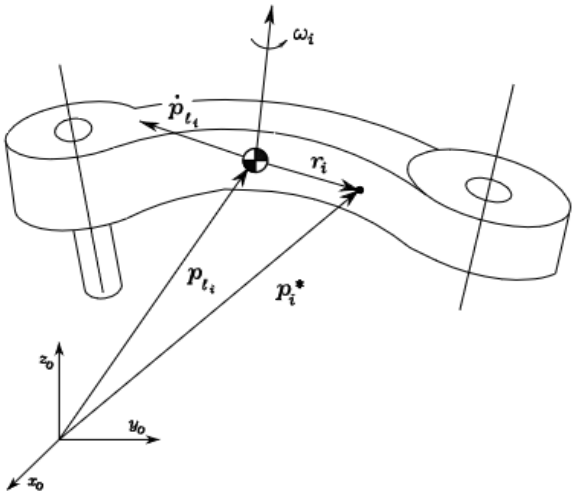
Partiamo anzitutto dalla caratterizzazione della Lagrangiana, la quale è la differenza tra la forza cinematica e quella potenziale: $L(q,\dot{q}) = \mathcal{T}(q,\dot{q}) - \mathcal{U}(q)$

Capiamo perciò che dobbiamo definire le espressione che riguardano appunto le due energie:

- **Energia Cinetica**:

L'energia cinetica totale è data dalla somma dei contributi relativi al moto di ciascun braccio e dei contributi relativi al moto di ciascun giunto:

$$\mathcal{T} = \sum_{i=1}^{n} (\mathcal{T}_{li} + \mathcal{T}_{mi})$$



A questo punto dobbiamo calcolare $\mathcal{T}_{li}$ e $\mathcal{T}_{mi}$ in funzione delle coordinate generalizzate. Partiamo dall'energia cinetica $\mathcal{T}_{li}$ dovuta al braccio i rappresentato in figura. Supponiamo di avere una terna di riferimento 0 e le coordinate del baricentro del braccio i rispetto a quest'ultima terna. Per calcolare $\mathcal{T}_{li}$, facciamo riferimento ad una particella elementare del braccio i di posizione p_i^*, per la quale l'energia cinetica è dato dal seguente integrale di massa sulla massa m_{li} del singolo braccio. La risoluzione di questo tipo di integrale non è particolarmente agevole, per cui possiamo pensare di ipotizzare per semplicità che la distribuzione di massa ρ dei bracci, la quale è data dal rapporto massa e volume, è costante. Per cui possiamo scrivere: $dm = \rho dV$

Passeremo così da un integrale di massa ad uno volumetrico:

$$\mathcal{T}_{li} = \frac{1}{2}\int_{m_{li}} \dot{\rho}_i^{*T}\dot{\rho}_i^* dm = \frac{1}{2}\int_{V_{li}} \dot{\rho}_i^{*T}\dot{\rho}_i^* \rho DV$$

Dobbiamo ora caratterizzare la velocità lineare del generico punto del braccio $\dot{\rho}_i^*$:

$$\dot{\rho}_i^* = \dot{\rho}_{li} + \omega_i \times r_i = \dot{\rho}_{li} + S(\omega_i) r_i$$

Andando a sostituire quest'ultima nella $\mathcal{T}_{li}$ avremo tre termini:

- **Termine Traslazionale**:

$$\frac{1}{2}\int_{V_{li}} \dot{\rho}_{li}^T \dot{\rho}_{li} \rho dV = \frac{1}{2} m_{li} \dot{\rho}_{li}^T \dot{\rho}_{li}$$

- **Termine Mutuo**:

$$2\left(\frac{1}{2}\int_{V_{li}} \dot{\rho}_{li}^T S(\omega_i) r_i \rho dV\right) = 2\left(\frac{1}{2}\dot{\rho}_{li}^T S(\omega_i)\int_{V_{li}} (\dot{\rho}_i^* - p_{li})\rho dV\right) = 0 \quad Dove: \; \underbrace{\rho_{li} = \frac{1}{m_{li}}\int_{V_{li}} \rho_i^* \rho dV}_{\textbf{Baricentro}}$$

- **Termine Rotazionale**:

$$\frac{1}{2}\int_{V_{li}} r_i^T S^T(\omega_i) S(\omega_i) r_i \rho dV \xrightarrow{S(\omega_i) r_i = -S(r_i)\omega_i} = \frac{1}{2}\omega_i^T \left(\int_{V_{li}} S^T(r_i) S(r_i) \rho dV\right)\omega_i = \frac{1}{2}\omega_i^T I_{li} \omega_i$$

Dove I_i è una matrice simmetrica detta **Tensore di Inerzia**, il quale ci viene fornito dal costruttore del braccio.

$$I_{li} = \begin{bmatrix} I_{l_i xx} & -I_{l_i xy} & -I_{l_i xz} \\ * & I_{l_i yy} & -I_{l_i yz} \\ * & * & I_{l_i zz} \end{bmatrix} = \begin{bmatrix} \int \left(r_{iy}^2 + r_{iz}^2\right)\rho dV & -\int r_{ix}^2 r_{iy}^2 \rho dV & -\int r_{ix}^2 r_{iz}^2 \rho dV \\ * & \int (r_{ix}^2 + r_{iz}^2)\rho dV & -\int r_{iy}^2 r_{iz}^2 \rho dV \\ * & * & \int \left(r_{ix}^2 + r_{iy}^2\right)\rho dV \end{bmatrix}$$

In definitiva, avremo la $\mathcal{T}_{li}$ descritta dalla relazione di sinistra. Tuttavia, In questa espressione la massa m_{li} del braccio è costante, mentre il tensore di inerzia è variabile ed inoltre è calcolato rispetto al braccio. Se volessimo rendere I_{li} costante dovremmo considerare I_{li}^i e $\omega_i^0 = R_i^0 \omega_i^i$. In questo modo scarico la dipendenza dalla configurazione del manipolatore sulle metrici di rotazione, rendendo così il tensore di inerzia costante. Allora avremo che la $\mathcal{T}_{li}$ è:

$$\mathcal{T}_{li} = \frac{1}{2} m_{li} \dot{\rho}_{li}^T \dot{\rho}_{li} + \frac{1}{2} \omega_i^T I_{li} \omega_i \quad \rightarrow \quad \mathcal{T}_{li} = \frac{1}{2} m_{li} \dot{\rho}_{li}^T \dot{\rho}_{li} + \frac{1}{2} \omega_i^T R_i I_{li} R_i^T \omega_i$$

A questo punto dobbiamo esprimere le velocità che compaioni in questa formule in funzione delle varibili di giunto. Quindi andiamo a calcolare velocità lineare e angolare del braccio l_i in funzione delle velocità dei giunti precendenti.

Velocità Lineare

$$\dot{p}_{li} = J_{P1}^{(l_i)} \dot{q}_1 + \cdots + J_{Pi}^{(l_i)} \dot{q}_i = J_P^{(l_i)} \dot{q}$$

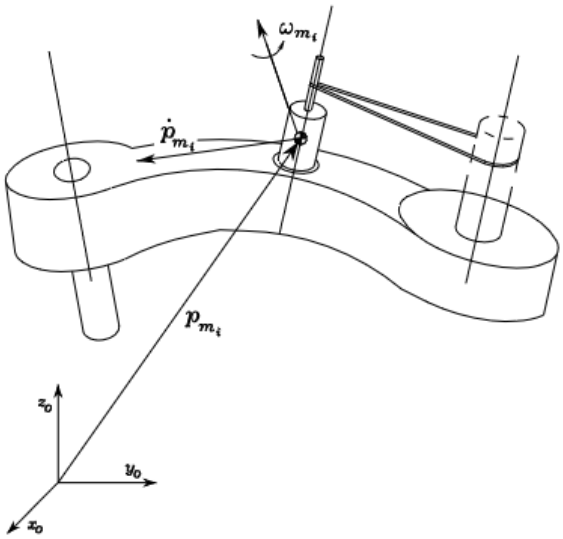
$$J_{Pj}^{(l_i)} = \begin{cases} z_{j-1} & giunto\ rotoidale \\ z_{j-1} \times \left(p_{li} - p_{j-1}\right) & giunto\ prismatico \end{cases}$$

Velocità Angolare

$$\omega_i = J_{O1}^{(l_i)} \dot{q}_1 + \cdots + J_{Oi}^{(l_i)} \dot{q}_i = J_O^{(l_i)} \dot{q}$$

$$J_{Oj}^{(l_i)} = \begin{cases} 0 & giunto\ rotoidale \\ z_{j-1} & giunto\ prismatico \end{cases}$$

Passiamo ora al calcolo dell'energia cinetica $\mathcal{T}_{mi}$ dovuta ai motori. Facciamo l'ipotesi di lavoro che l'asse del motore non è coassiale con l'asse del giunto. In particolare, il motore si trova disposto come in figura al centro del braccio $i-1$ con un asse che ruota a velocità ω_{mi}, mentre a destra troviamo l'asse del giunto i che collaga il braccio $i-1$ e il braccio i. Questa ipotesi di lavoro è anche plausibile perché, nella pratica, tale scelta viene attuata per ridurre i contributi inerziali. Allora la velocità angolare del giunto i sarà dato da due contributi, uno che è proprio il suo moto (il secondo termine) e l'altro (il primo termine) dovuto al moto del giunto $i-1$:



$$\omega_{mi} = \omega_{i-1} + k_{ri} \dot{q}_i z_{mi}$$

In maniera del tutto analoga a quanto fatto precedentemente, abbiamo che $\mathcal{T}_{mi}$:

$$\mathcal{T}_{mi} = \frac{1}{2} m_{mi} \dot{p}_{mi}^T \dot{p}_{mi} + \frac{1}{2}\ \omega_{mi}^T I_{mi} \omega_{mi} = \frac{1}{2} m_{mi} \dot{q}^T J_P^{(mi)T} J_P^{(mi)} \dot{q} + \frac{1}{2}\ \dot{q}^T J_O^{(mi)T} R_{mi} I_{mi}^{mi} R_{mi}^T J_o^{(mi)} \dot{q}$$

A questo punto dobbiamo esprimere le velocità che compaioni in questa formule in funzione delle varibili di giunto. Quindi andiamo a calcolare velocità lineare e angolare del braccio l_i in funzione delle velocità dei giunti precendenti.

Velocità Lineare

$$\dot{p}_{li} = J_{P1}^{(m_i)}\dot{q}_1 + \cdots + J_{Pi}^{(m_i)}\dot{q}_i = J_P^{(m_i)}\dot{q}$$

$$J_{Pj}^{(l_i)} = \begin{cases} z_{j-1} & giunto\ rotoidale \\ z_{j-1} \times (p_{mi} - p_{j-1}) & giunto\ prismatico \end{cases}$$

Velocità Angolare

$$\omega_i = J_{O1}^{(m_i)}\dot{q}_1 + \cdots + J_{Oi}^{(m_i)}\dot{q}_i = J_O^{(m_i)}\dot{q}$$

$$J_{Oj}^{(l_i)} = \begin{cases} J_{Oj}^{(l_i)} & giunto\ rotoidale \\ k_{ri}z_{j-1} & giunto\ prismatico \end{cases}$$

In conclusione, l'**energia Cinetica Complessiva** sarà data dalla seguente espressione:

$$\boldsymbol{\mathcal{T} = \sum_{i=1}^{n}(\mathcal{T}_{li} + \mathcal{T}_{mi}) = \dot{q}^T \frac{1}{2} B(q)\dot{q}}$$

Nella quale definiamo la **Matrice di Inerzia $\boldsymbol{B(q)}$**, la quale avrà la seguente espressione:

$$B(q) = \sum_{i=1}^{n}\left(m_{li}J_P^{(li)T}J_P^{(li)} + J_O^{(li)T}R_i I_{li}^i R_i^T J_O^{(li)} + m_{mi}J_P^{(mi)T}J_P^{(mi)} + J_O^{(mi)T}R_{mi}I_{mi}^{mi}R_{mi}^T J_O^{(mi)}\right)$$

Ed è dotata delle seguenti proprietà:

- Quadrata (quindi anche invertibile);
- Simmetrica;
- Definita positiva;
- Dipendente dalla configurazione (in generale).

- **Energia Potenziale**:

In maniera del tutto analoga per l'energia potenziale avremo che:

$$\mathcal{U} = \sum_{i=1}^{n}(\mathcal{U}_{li} + \mathcal{U}_{mi}) \qquad dove: \qquad \begin{aligned} \mathcal{U}_{li} &= -\int_{V_{li}} g_0^T p_i^* \rho dV = -m_{li}g_0^T p_{li} \\ \mathcal{U}_{mi} &= -m_{mi}g_0^T p_{mi} \end{aligned}$$

Allora l'**energia Potenziale Totale** risulta essere:

$$\boldsymbol{\mathcal{U} = \sum_{i=1}^{n}(\mathcal{U}_{li} + \mathcal{U}_{mi}) = -\sum_{i=1}^{n}\left(m_{li}g_0^T p_{li} + m_{mi}g_0^T p_{mi}\right)}$$

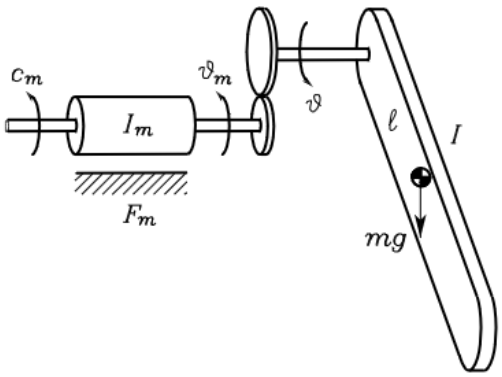
In definitiva, la Lagrangiana avrà la seguente espressione:

$$\boldsymbol{L(q,\dot{q}) = \mathcal{T}(q,\dot{q}) - \mathcal{U}(q) = \dot{q}^T \frac{1}{2} B(q)\dot{q} + \sum_{i=1}^{n}\left(m_{li}g_0^T p_{li}(q) + m_{mi}g_0^T p_{mi}(q)\right)}$$

Esempio: supponiamo di avere un motore e riduttore che azionano un braccio. Indicando con $k_r \gg 1$ il rapporto di trasmissione del riduttore, abbiamo che esso riduce la velocità del braccio $\dot{\theta}$ di un fattore k_r: $\dot{\theta} = \dot{\theta}_m / k_r$

Mentre incrementa la coppia c agente sul carico di un fattore k_r: $c = k_r c_m$

Di questo sistema conosciamo tutti i parametri presenti in figura incluso k_r. Andiamo anzitutto a calcolare l'energia cinetica e l'energia potenziale:



$$\mathcal{T} = \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} I_m \dot{\theta}_m{}^2 = \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} I_m k_r^2 \dot{\theta}^2 \qquad \mathcal{U} = mgl(1 - \cos\theta)$$

Definiamo la funzione lagrangiana:

$$L = \mathcal{T} - \mathcal{U} = \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} I_m k_r^2 \dot{\theta}^2 - mgl(1 - \cos\theta)$$

L'equazione del moto sarà:

$$(I + I_m k_r^2)\ddot{\theta} + mgl \sin\theta = \xi = \tau - F\dot{\theta} - F_m k_r^2 \dot{\theta}$$

Dove:

- $\tau =$ è la coppia vista dal carico (è diversa da c);
- $F\dot{\theta} =$ attrito viscoso lato carico;
- $F_m k_r \dot{\theta} =$ attrito viscoso lato motore.

Ora che abbiamo caratterizzato la funzione Lagrangiana, sostituiamola nell'equazione del moto:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i$$

Andiamo ora a caratterizzare i termini a primo membro di tale equazione:

- **Termine Verde**: la parte di energia potenziale è assente in quanto la sua derivata rispetto a a $\dot{q}$ è nulla (visto che dipende solo da q). Dunque, avremo solo la componente relativa a quella cinetica:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} = \frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{q}_i}\right) = \sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n} \frac{db_{ij}(q)}{dt}\dot{q}_j = \sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{db_{ij}(q)}{q_k}\dot{q}_k\dot{q}_j$$

- **Termine Blu**: in questo caso avremo i contributi di entrambe le energie:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial q_i} = \frac{\partial \mathcal{T}}{\partial q_i} - \frac{\partial \mathcal{U}}{\partial q_i} &= \frac{1}{2}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{db_{jk}(q)}{q_i}\dot{q}_k\dot{q}_j\right] - \left[-\sum_{j=1}^{n}\left(m_{li} g_0^T \frac{\partial p_{lj}}{\partial q_i} + m_{mi} g_0^T \frac{\partial p_{mj}}{\partial q_i}\right)\right] \\
&= \frac{1}{2}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{db_{jk}(q)}{q_i}\dot{q}_k\dot{q}_j\right] - \left[-\sum_{j=1}^{n}\left(m_{li} g_0^T J_{Pi}^{(l_j)}(q) + m_{mi} g_0^T J_{Pi}^{(m_j)}(q)\right)\right] \\
&= \frac{1}{2}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{db_{jk}(q)}{q_i}\dot{q}_k\dot{q}_j\right] - g_i(q)
\end{aligned}$$

Andando a sostituire:

$$\sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{db_{ij}(q)}{q_k}\dot{q}_k\dot{q}_j - \frac{1}{2}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{db_{jk}(q)}{q_i}\dot{q}_k\dot{q}_j\right] + g_i(q) = \xi_i$$

E poi ad imporre:

$$h_{ijk} = \frac{db_{ij}(q)}{q_k} - \frac{1}{2}\frac{db_{jk}(q)}{q_i}$$

Il risultato è il seguente set di equazioni del moto:

$$\boldsymbol{\sum_{j=1}^{n} b_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_k\dot{q}_j + g_i(q) = \xi_i \qquad i = 1, \dots, n}$$

Dove identifichiamo:

- **Termini di Accellerazione**: il coefficiente b_{ii} rappresenta il momento di inerzia visto all'asse del giunto i nella configurazioen corrente del manipolatore quando gli altri giunti sono bloccati. Il coefficiente b_{ij} tiene conto dell'effetot dell'accellerazione del giunto j sul giunto i;
- **Termini Quadratici in Velocità**: il termine $h_{ijj}\dot{q}_j^2$ rappresenta l'effetto centrifugo indotto al giunto i dalla velocità del giunto j. E' da notare che $h_{iii} = 0$ poiché $\partial b_{ii}/\partial q_i = 0$. Fisicamente ciò avviene perché il momento di inerzia del giunto i non può dipendere da q_i, cioè da una posizione angolare di una terna (quella di D-H) che ho scelto arbitrariemente. Il termine $h_{ijk}\dot{q}_j\dot{q}_k$ rappresenta l'effetto di Coriolis indotto al giunto i dalle velocità dei giunti j e k;
- **Termini Dipendenti Solo dalla Configurazione**: il termine g_i rappresenta la coppia generata all'asse del giunto i nella configurazione corrente del manipolatore per effetto della gravità.

In alternativa, utilizzando una forma compatta: $\boldsymbol{B(q)\ddot{q} + n(q,\dot{q}) = \xi}$

Dove $n(q,\dot{q})$ è un termine non lineare che è dato da:

$$n(q,\dot{q}) = \dot{B}(q)\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}(\dot{q}^T B(q)\dot{q})\right)^T + \left(\frac{\partial U(q)}{\partial q}\right)^T$$

Infine, andando ad esplicitare le forze non conservative presenti nel termine ξ_i, riusciamo finalmente a ricavare il modello dinamico nello spazio dei giunti:

$$\boldsymbol{B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + f_s(\dot{q}) + g(q) = \tau - J^T(q)h_e}$$

Dove:

$$\sum_{j=1}^{n} c_{ij}\,\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\,\dot{q}_k\dot{q}_j \qquad \forall\, i = 1, \dots, n$$

6.2.3 Proprietà del Modello Dinamico

Vediamo ora le proprietà notevoli del modello dinamico secondo la formulazione di Lagrange:

➢ **Anti-Simmetria della Matrice $\dot{B} - 2C$**

Tale proprietà afferma che, se fattorizzo $C(q,\dot{q})\dot{q}$ come $\dot{B} - 2C$, allora quest'ultima matrice che chiameremo $N(q,\dot{q})$ è antisimmetrica. In formule dobbiamo dimostrare che:

$$\boldsymbol{N(q,\dot{q}) = \dot{B}(q) - 2C(q,\dot{q}) = -N^T(q,\dot{q})}$$

Per dimostrare questa proprietà andiamo a studiare i seguenti termini:

$$\begin{aligned}\sum_{j=1}^{n} c_{ij}\,\dot{q}_j &= \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\,\dot{q}_k\dot{q}_j \\ &= \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j \\ &= \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial b_{ij}}{\partial q_k}\dot{q}_k\dot{q}_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j\end{aligned}$$

I coefficienti c_{ij} saranno dati dalla seguente sommatoria in cui gli c_{ijk} sono definiti come **Simboli di Christofell del Primo Ordine**, i quali sono dati dalla relazione di fianco.

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)$$

$$\begin{aligned}c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k &= \sum_{k=1}^{n}\left(\frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\right)\dot{q}_k \\ &= \frac{1}{2}\sum_{k=1}^{n}\frac{\partial b_{ij}}{\partial q_k}\dot{q}_k + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k \\ &= \frac{1}{2}\dot{b}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\end{aligned}$$

Prima moltiplichiamo ambo i membri per due e successivamente portiamo allo stesso membro c_{ij} e $\dot{b}_{ij}$, infine chiamiamo n_{ij} quest'ultimo. Osserviamo che, n_{ij} sarà uguale alla sommatoria a terza membro, inoltre se inseriamo il meno nella sommatoria di fatto il penultimo membro è proprio $-n_{ji}$:

$$n_{ij} = \dot{b}_{ij} - 2c_{ij} = -\frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{jk}}{\partial q_i} - \frac{\partial b_{ik}}{\partial q_j}\right)\dot{q}_k = -n_{ji}$$

Per cui, se mettiamo la matrice N in una qualisasi forma quadratica, quest'ultima verrà sempre annullata: $w^T N(q,\dot{q})w = 0 \quad \forall\, w$

La proprietà per cui N è antisimmetrica viene anche definita come **Proprietà Di Passività** e, all'atto pratico, ci dice che la matrice N è in grado di annullare qualsiasi forma quadratica.

Strettamente legata a questa proprietà ne discende un'altra più debole che afferma l'annullamento della forma quadratica per $w = \dot{q}$ per qualunque C:

$$\dot{q}^T N(q, \dot{q})\dot{q} = 0 \qquad \forall\, C$$

Questa proprietà non è banale in quanto, poiché N dipende da q e $\dot{q}$, potrebbe succedere che tale relazione è vera anche per una $N = \dot{B} - 2C$ in cui la C non è definita mediante i simboli di Christofell del primo ordine. A darci la conferma che questa proprietà è verificata ce lo dice il seguente principio. Il **Principio di Conservazione dell'Energia di Hamilton** afferma che che la derivata temporale dell'energia cinetica è data dal lavoro di tutte le forze agenti sul sistema:

$$\boldsymbol{\frac{1}{2}\frac{d}{dt}(\dot{q}^T B(q)\dot{q}) = \dot{q}^T(\tau - F_v\dot{q} - f_s(\dot{q}) - g(q) - J^T(q)h_e)}$$

L'energia cinetica abbiamo visto essere anche:

$$\boldsymbol{\frac{1}{2}\frac{d}{dt}(\dot{q}^T B(q)\dot{q}) = \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{q}^T B(q)\ddot{q}}$$

Ora, dal modello dinamico, possiamo ricavare $B(q)\ddot{q} = (\tau - F_v\dot{q} - f_s(\dot{q}) - g(q) - J^T(q)h_e) - C\dot{q}$:

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}(\dot{q}^T B(q)\dot{q}) &= \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{q}^T B(q)\ddot{q} \\
&= \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{q}^T[(\tau - F_v\dot{q} - f_s(\dot{q}) - g(q) - J^T(q)h_e) - C\dot{q}] \\
&= \dot{q}^T\left(\frac{1}{2}\dot{B}(q) - C\right)\dot{q} + \dot{q}^T(\tau - F_v\dot{q} - f_s(\dot{q}) - g(q) - J^T(q)h_e) \\
&= \frac{1}{2}\dot{q}^T\left(\dot{B}(q) - 2C(q,\dot{q})\right)\dot{q} + \dot{q}^T(\tau - F_v\dot{q} - f_s(\dot{q}) - g(q) - J^T(q)h_e)
\end{aligned}$$

Ma allora, affinchè sia verificata l'uguaglianza tra la relazione in blu e quella in verde è necessario che il termine di fianco si annuli.

$$\frac{1}{2}\dot{q}^T\left(\dot{B}(q) - 2C(q,\dot{q})\right)\dot{q}$$

Esempio:

➢ Linearità dei Parametri Dinamici

Supponendo di trascurare gli attriti (F_v ed f_s) e le forze in punta (h_e), il modello dinamico è il seguente: $B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$

La proprietà di linearità nei parametri dinamici afferma che, seppur il modello dinamico è non lineare, esso può essere espresso in forma lineare rispetto ai parametri dinamici π del modello mediante la **Matrice Regressore** $\boldsymbol{Y}$, la quale dipende da $q, \dot{q}, \ddot{q}$: $\boldsymbol{\tau = \mathcal{Y}(q,\dot{q},\ddot{q})\pi}$

Per dimostrare questa proprietà, dobbiamo anzitutto verificare che la Lagrangiana è lineare nei parametri $\pi_i = \left[m_i \;\; m_i l_{Cix} \;\; m_i l_{Ciy} \; m_i l_{Ciz} \;\; I_{ixx} \;\; I_{ixy} \;\; I_{ixz} \;\; I_{iyz} \; I_{izz} \; I_{mi}\right]^T$ (è un vettore colonna $11x1$), ovvero possiamo scriverla come:

$$\mathcal{L} = \sum_{i=1}^{n} (\beta_{\mathcal{T}_i}^T - \beta_{\mathcal{U}_i}^T)\pi_i$$

Poiché la Lagrangiana è data dalla differenza tra energia cineatica e potenziale, dobbiamo verificare che queste due energie siano lineari rispetto ai parametri dinamici. La dimostrazione di ciò tiene conto di due aspetti:

- Si ragionerà considerando il cosidetto Braccio Aumentato i, il quale combina sia il braccio i −esimo che il rotore $i+1$;
- Si farà ampio uso del seguente teorema che ci conviene enunciare, il quale è da specificare che ragiona in maniera scalare e dunque noi dovremo opportunamento impiegarlo nel caso vettoriale.

Teorema di Steiner: stabilisce che il momento di inerzia di un corpo rigido rispetto ad un'asse è pari alla somma tra il momento di inerzia rispetto all'asse parallelo passante per il baricentro e il prodotto della massa per il quadrato della distanza tra gli assi.

Andiamo ora a dimostrare la linearità dell'energia cinetica e dell'energia potenziale rispetto a π:

- **Energia Cinetica**: consideriamo l'insieme **Braccio Aumentato** costituito dal braccio i e il rotore $i+1$, la cui energia cinetica sarà: $\mathcal{T}_i = \mathcal{T}_{li} + \mathcal{T}_{mi+1}$

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

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

A questo punto possiamo fare riferimento al baricentro p_{Ci} dell'insieme braccio-rotore, allora:

- **Velocità Lineare del Braccio**:

$$\dot{p}_{li} = \dot{p}_{Ci} + \omega_i \times r_{Ci,li} = \dot{p}_{Ci} + \omega_i \times (p_{li} - p_{Ci})$$

- **Velocità Lineare del Motore**:

$$\dot{p}_{mi+1} = \dot{p}_{Ci} + \omega_i \times r_{Ci,mi+1} = \dot{p}_{Ci} + \omega_i \times (p_{mi+1} - p_{Ci})$$

Sostiuiamo nell'energia cinetica del braccio T_{li} le velocità appena ricavate. Applichiamo il teorema di Steiner per definire la matrice:

$$\bar{I}_{li} = I_{li} + m_{li} S^T(r_{Ci,li}) S(r_{Ci,li})$$

Dunque:

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell i} \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \dot{\boldsymbol{p}}_{C_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i) m_{\ell_i} \boldsymbol{r}_{C_i,\ell_i} + \frac{1}{2} m_{\ell_i} \boldsymbol{\omega}_i^T \boldsymbol{S}^T(\boldsymbol{r}_{C_i,\ell_i}) \boldsymbol{S}(\boldsymbol{r}_{C_i,\ell_i}) \boldsymbol{\omega}_i + \frac{1}{2} \boldsymbol{\omega}_i^T \boldsymbol{I}_{\ell_i} \boldsymbol{\omega}_i$$

$$= \frac{1}{2} m_{\ell i} \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \dot{\boldsymbol{p}}_{C_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i) m_{\ell_i} \boldsymbol{r}_{C_i,\ell_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\boldsymbol{I}}_{\ell_i} \boldsymbol{\omega}_i$$

Per quanto riguarda l'energia cinetica del motore facciamo dei ragionamenti analoghi impiegando il teorema di steiner: $\bar{I}_{mi+1} = I_{mi+1} + m_{mi+1} S^T(r_{Ci,mi+1}) S(r_{Ci,mi+1})$

$$\mathcal{T}_{m_{i+1}} = \frac{1}{2} m_{m_{i+1}} \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \dot{\boldsymbol{p}}_{C_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i) m_{m_{i+1}} \boldsymbol{r}_{C_i,m_{i+1}} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\boldsymbol{I}}_{m_{i+1}} \boldsymbol{\omega}_i + k_{r,i+1} \dot{q}_{i+1} \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{\omega}_i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}$$

In definitiva, l'energia cinetica complessiva sarà:

$$\mathcal{T}_i = \frac{1}{2} m_i \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i + k_{r,i+1} \dot{q}_{i+1} \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{\omega}_i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}$$

Nella pratica il rotore ha una distribuzione di massa a simmetria cilindrica intorno al suo asse di rotazione. Se il tensore di inerzia del rotore lo calcolo rispetto alla terna di quest'ultimo, esso avrà la forma di fianco. Grazie a ciò, potrò scrivere il termine $I_{mi+1} z_{mi+1}$ come di seguito. Perciò abbiamo scritto il prodotto tensore di inerzia per il versore dell'asse come momento di inerzia per l'asse in questione:

$$I_{mi}^{mi} = \begin{bmatrix} I_{mixx} & 0 & 0 \\ 0 & I_{mixx} & 0 \\ 0 & 0 & I_{mizz} \end{bmatrix}$$

$$I_{mi+1} z_{mi+1} = R_{mi+1} I_{mi+1}^{mi+1} R_{mi+1}^T z_{mi+1}$$

In definitiva, dall'ultima espressione della $\mathcal{T}_i$, con l'ultimo artificio fatto, abbiamo complessivamente per il singolo braccio 8 parametri:

- 1 parametro $m_i = m_{li} + m_{mi+1}$;
- 6 parametri per il tensore di inerzia complessivo $\bar{I}_i$;
- 1 parametro I_{mi+1}.

A questo punto il problema è che, seppur i parametri m_i ed I_{mi+1}, lo stesso non si può dire per $\bar{I}_i$ in quanto è riferita alla terna base. Perciò dobbiamo fare in modo da riferirlo ad una terna solidale al braccio e per faro riscriviamo $\mathcal{T}_i$ rispetto alla terna del braccio i. Anche in questo caso, riapplichiamo il teorema di steiner: $\hat{I}_i^i = \bar{I}_i^i + m_i S^T(r_i^i, c_i) S(r_i^i, c_i)$

$$\mathcal{T}_i = \frac{1}{2} m_i \dot{\boldsymbol{p}}_{C_i}^{iT} \dot{\boldsymbol{p}}_{C_i}^{i} + \frac{1}{2} \boldsymbol{\omega}_i^{iT} \bar{\boldsymbol{I}}_i^i \boldsymbol{\omega}_i^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}^{iT} \boldsymbol{\omega}_i^i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 I_{m_{i+1}}$$

$$= \frac{1}{2} m_i \dot{\boldsymbol{p}}_i^{iT} \dot{\boldsymbol{p}}_i^i + \dot{\boldsymbol{p}}_i^{iT} \boldsymbol{S}(\boldsymbol{\omega}_i^i) m_i \boldsymbol{r}_{i,C_i}^i + \frac{1}{2} \boldsymbol{\omega}_i^{iT} \hat{\boldsymbol{I}}_i^i \boldsymbol{\omega}_i^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}^{iT} \boldsymbol{\omega}_i^i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 I_{m_{i+1}}$$

In particolare, $m_i r_{i,Ci}^i$ sono gli altri tre parametri del modello e vengono definiti **Momenti Prima d'Inerzia**, ovvero i momenti del baricentro per la massa.

Abbiamo così dimostrato che l'energia cinetica è lineare rispetto a:
- Massa m_i → 1 parametro;
- Momento primo d'inerzia $m_i r_{i,Ci}^i$ → 6 parametri
- Tensore di inerzia $\hat{I}_i^i$ → 3 parametri;
- Momento d'inerzia del motore I_{mi+1} → 1 parametro.

Energia Potenziale: in maniera analoga all'energia cinetica ricaviamo l'energia potenziale. In particolare, in questo caso è più semplice quando la vado a riferire al baricentro complessivo p_{ci}:

$$\mathcal{U}_i = -m_i \boldsymbol{g}_0^{iT} \boldsymbol{p}_{C_i}^i = -\boldsymbol{g}_0^{iT} (m_i \boldsymbol{p}_i^i + m_i \boldsymbol{r}_{i,C_i}^i)$$

Abbiamo così dimostrato che l'energia potenziale è lineare rispetto a:
- Massa m_i → 1 parametro;
- Momento primo d'inerzia $m_i r_{i,Ci}^i$ → 6 parametri

Avendo dimostrato laa linearità di entrambe le energia, sarà lineare anche la lagrangiana rispetto ai parametri $\pi_i = [m_i \;\; m_i l_{Cix} \;\; m_i l_{Ciy} \; m_i l_{Ciz} \;\; \widehat{I_{ixx}} \;\; \widehat{I_{ixy}} \;\; \widehat{I_{ixz}} \;\; \widehat{I_{iyz}} \;\; \widehat{I_{izz}} \;\; I_{mi}]^T$ (è un vettore colonna $11x1$) dove:
- m_i = massa del braccio i;
- $[m_i l_{Cix} \;\; m_i l_{Ciy} \; m_i l_{Ciz}]$ = momento primo di inerzia, ovvero le coordinate del baricentro l_{Ci} del braccio aumentato pesate per la massa m_i;
- $[\widehat{I_{ixx}} \;\; \widehat{I_{ixy}} \;\; \widehat{I_{ixz}} \;\; \widehat{I_{iyz}} \;\; \widehat{I_{izz}}]$ = tensore d'inerzia del braccio, il quale è costante ed è riportato alla terna di D-H;
- I_{mi} = momento di inerzia del motore.

In definitiva, abbiamo dimostrato come la Lagrangiana è lineare nei parametri dinamici grazie ai vettori $\beta_{\mathcal{T}_i}^T$ e $\beta_{\mathcal{U}_i}^T$:

$$\mathcal{L} = \sum_{i=1}^{n} (\beta_{\mathcal{T}_i}^T - \beta_{\mathcal{U}_i}^T)\pi_i$$

E' da specificare che le operazioni di derivazione richieste dalle equazioni di Lagrange di seguito non vanno ad alterare la proprietà di linearità nei parametri dinamici. Questo implica che potremo scrivere le forze generalizzate ξ_i al giunto i-esimo come:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\lambda}_i} - \frac{\partial \mathcal{L}}{\partial \lambda_i} = \xi_i \quad i = 1, \dots, n \quad \rightarrow \quad \xi_i = \sum_{i=1}^{n} y_{ij}^T \pi_j = \sum_{i=1}^{n} \left[\frac{d}{dt}\frac{\partial B_{\mathcal{T}i}}{\partial \dot{q}_i} - \frac{\partial B_{\mathcal{T}j}}{\partial \dot{q}_i} + \frac{\partial B_{\mathcal{U}i}}{\partial q_i}\right]\pi_j$$

Osservando poi che la forza ξ_i al giunto i può dipendere solo dai parametri dei bracci successivi, possiamo costruire una matrice triangolare alta che di fatto è proprio il regresso Y.

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \cdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} y_{11}^T & y_{12}^T & \cdots & y_{1n}^T \\ 0^T & y_{22}^T & \cdots & y_{21}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \cdots & y_{nn}^T \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \cdots \\ \pi_n \end{bmatrix} \quad \rightarrow$$

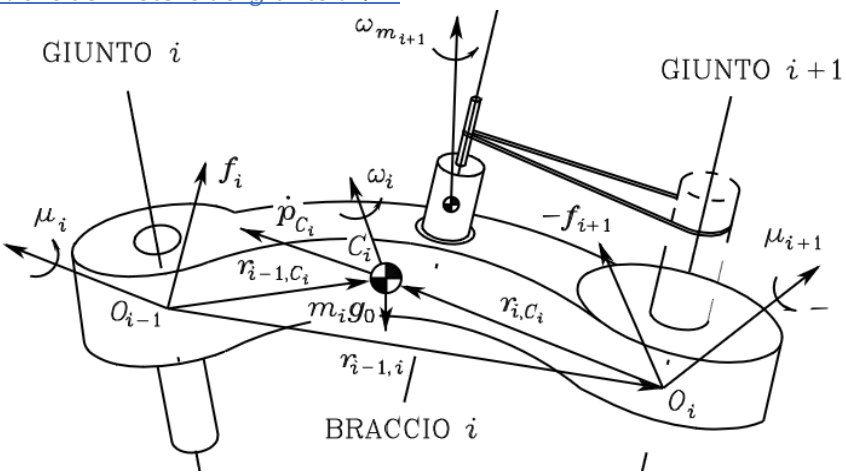
In forma compatta: $\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\pi}$

La formulazione di Lagrange può essere estesa anche al caso di giunti o bracci elastici. In realtà tipicamente si parla di flessibilità dei bracci, i quali possono essere immaginate come delle travi elastiche. Questa precisazione non casuale in quanto se abbiamo una elasticitià dei giunti, allora avremo un modello a parametri concentrati. Per cui l'introduzione dell'elasticità comporterà solo una modifica dell'energia potenziale nella lagrangiana ma le coordinate generalizzate saranno sempre n. Invece, per la flessibilità dei bracci, sarà necessario impiegare un modello a parametri distributi. Perciò, in questo caso, sarà necessario:

- Modificare il termine dell'energia potenziale $\mathcal{U}$;
- Aggiungere dei parametri alle coordinate generalizzate che tengano conto della flessibilità.

6.3 Formulazione di Newton-Eulero

La **Formulazione di Newton-Eulero** consente di ricavare il modello dinamico del manipolatore attraverso lo studio dell'equilibrio di forze e momenti agenti sul generico braccio. Il set di equazioni così ottenute presenteranno una struttura tale che la soluzione può essere calcolata mediante un algoritmo ricorsivo. Lo studio della formulazione di NE verrà effettuta considerando il **Braccio Aumentato *i*** come mostrato nella seguente figura, il quale tiene conto sia del braccio i che del motore del giunto $i+1$.



Facendo riferimento al baricentro C_i del braccio aumentato possiamo caratterizzare i seguenti parametri:

- m_i = massa del braccio aumentato;
- $\bar{I}_i$ = tensore d'inerzia del braccio aumentato;
- I_{mi+1} = momento di inerzia del motore;
- $r_{i-1,ci}$ = distanza dell'origine della terna $i-1$ dal centro di massa C_i;
- $r_{i,ci}$ = distanza dell'origine della terna i dal centro di massa C_i;
- $r_{i-1,i}$ = distanza delle origini delle terne $i-1$ e i.

Per velocità e accellerazioni possiamo considerare:

- $\dot{p}_{ci}$ = velocità lineare del centro di massa:
- $\ddot{p}_{ci}$ = accellerazione lineare del centro di massa;
- $\dot{p}_i$ = velocità lineare dell'origine della terna i;
- $\ddot{p}_i$ = accellerazione lineare dell'origine della terna i;
- ω_i = velocità angolare dell'origine della terna i;
- $\dot{\omega}_i$ = accellerazione angolare dell'origine della terna i;
- $\omega_{mi+1} = \omega_i + k_{r,i+1}\dot{q}_{i+1}$ = velocità angolare del motore;
- $\dot{\omega}_{mi+1}$ = accellerazione angolare del motore;
- g_0 = vettore accellerazione di gravità.

Infine per forze e momenti:

- f_i = forza che il braccio $i-1$ esercita sul braccio i;
- $-f_{i+1}$ = forza che il braccio $i+1$ esercita sul braccio i rispetto all'origine del braccio i (ecco perché è presente il meno);
- μ_i = momento esercitato dal braccio $i-1$ sul braccio i rispetto all'origine della terna $i-1$;
- $-\mu_{i+1}$ = momento esercitato dal braccio $i+1$ sul braccio i rispetto all'origine della terna i;

Come anticipato a inizio paragrafo la formulazione Newton-Eulero descrive il moto dei link in termini di equilibrio di forze e momenti agenti su di esso. A tal proposito si considerano le seguenti equazioni:

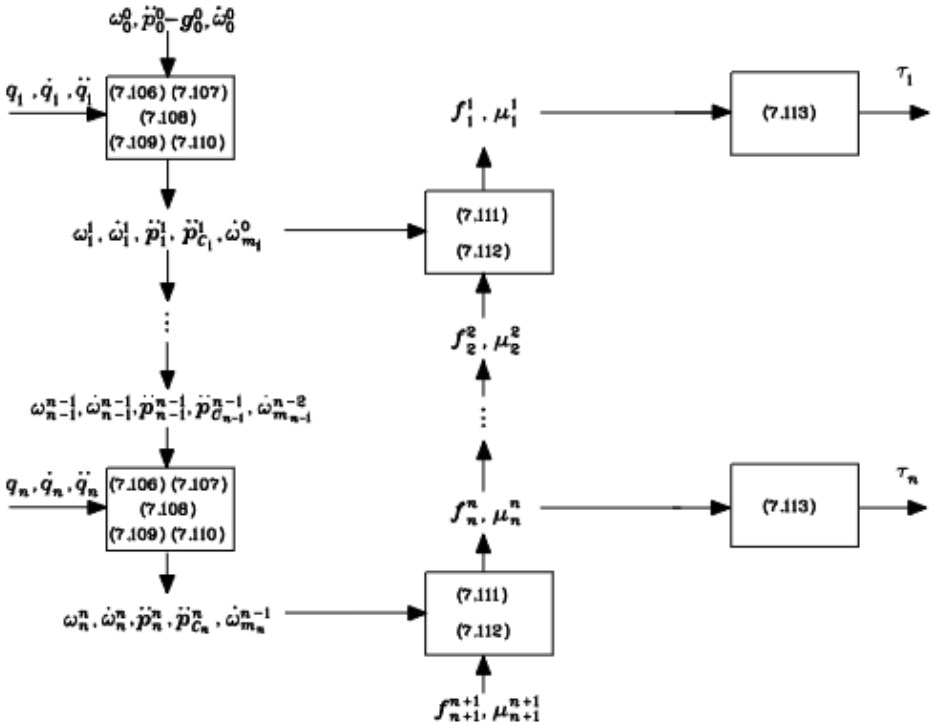
- **Equazione di Newton – Moto Traslazione del Baricentro**:

$$\boldsymbol{f_i - f_{i+1} + m_i g_0 = m_i \ddot{p}_{Ci}}$$

- **Equazione di Eulero – Moto Rotazionale del Baricentro**: le coppie e il termine di trasporto che il braccio i subisce dal precedente braccio e applica al successivo (con segno invertito) sono pari alla derivata del momento della quantità di moto, la quale fa riferimento al braccio aumentato:

$$\boldsymbol{\mu_i + f_i \times r_{i-1,Ci} - \mu_{i+1} - f_{i+1} \times r_{i,Ci} = \frac{d}{dt}\left(\bar{I}_i \omega_i + k_{r,i+1}\dot{q}_{i+1} I_{m_{i+1}} z_{m_{i+1}}\right)}$$

Le equazioni del moto di Newton-Eulero, a causa dell'accoppiamento del movimento del singolo giunto a quello degli altri, non sono in forma chiusa. Tuttavia, questo problema può essere aggirato costruendo un *algoritmo ricorsivo* schematizzato nella seguente figura:



Esso sarà costituito da due ricorsioni in cui tutti i vettori ci conviene riferirli alla terna corrente sul braccio i:

1. **Ricorsione in Avanti di Velocità e Accelerazioni**: partendo dalle condizioni iniziali $\omega_0, \ddot{p}_0 - g_0, \dot{\omega}_0$ e sapendo che per ogni braccio i abbiamo una dipendenza da $q_i \dot{q}_i \ddot{q}_{ii}$ del relativo giunto, vado a calcolare $\omega_i, \dot{\omega}_i, \dot{\omega}_{mi}, \ddot{p}_i, \ddot{p}_{Ci}$. Esegue questi calcoli fino ad arrivare al giunto n. E' da notare che una delle condizioni iniziali è $\ddot{p}_0 - g_0$, perciò potremmo chiederci perché non abbiamo messo $\ddot{p}_0$. Poiché nell'equazione di Newton metteremo un $-g_0$, facendo questa modifica andiamo a tenere conto del termine di gravità braccio per braccio.

2. **Ricorsione all'Indietro di Forze e Momenti**: a partire dall'ultimo giunto, se su di esso agiscono delle forze f_{n+1}^{n+1} e coppie μ_{n+1}^{n+1} (nel caso non ci fossero non si considerano), a ritroso mediante le formule di Newton ed Eulero calcoliamo le f_i^i e le μ_i^i. Da queste ultime poi calcoliamo le coppie τ_i.

Quanto detto circa la formulazione di Newton-Eulero si evince come essa si presti bene per la risoluzione della dinamica diretta. Infatti, immaginando di impiegare lo schema appena visto come un sistema che, dati in ingresso $q, \dot{q}, \ddot{q}$, in uscita calcolo τ. Ponendo $\ddot{q} = 0$ avrò che la coppia τ calcolata precedenemente e pari proprio a τ'. A questo punto dobbiamo calcolare la matrice $B(q)$ e per farlo possiamo possiamo ragionare in maniera analoga, dove però questa volta poniamo in ingresso $g_0 = 0, \dot{q} = 0, \dot{q}_i = 1, \dot{q}_j = 0$ per $i \neq j$. Ripetiamo questo procedimento per $i = 1, \dots, n$ in modo da ricostruire la matrice $B(q)$. Successivamente calcoliamo l'inversa in modo da poter sfruttare la relazione: $\boldsymbol{\ddot{q} = B^{-1}(q)(\tau - \tau')}$

Inoltre, tale formulazione si presta adeguatamente anche per quanto riguarda la dinamica inversa, tuttavia la formulazione di Lagrange è in grado di fornire prestazioni simili mediante una implementazione ricorsiva però con uno sforzo di riformulazione non trascurabile.

6.4 Identificazione dei Parametri Dinamici

La **Identificazione dei Parametri Dinamici** *è una procedura il cui scopo è ricavare appunto i parametri dinamici del modello dinamico in modo da poterli impiegare per la simulazione e il controllo del manipolatore.* Notoriamente, la questione della definizione dei parametri dinamici come masse, momenti primi di inerzia e tensori d'inerzia è sempre più complessa rispetto all'identificazione dei parametri delle convenzione di D-H. Anzitutto, i parametri dinamici non sono direttamente ricavabili dai datasheet del robot in quanto, vista l'esigenza dell'utenza di programmare il robot per un certo task, sono necessari solo i parametri della convenzione di D-H. Infatti, quest'ultimi possono essere direttamente indicati o comunque facilmente ricavabili dai dati di targa. Invece, il costruttore del manipolatore è interessato ai parametri dinamici perché il modello dinamico viene impiegato, come detto precedentemente, per la simulazione e per il controlo con il fine ultimo di progettare l'unità di governo. Allora, se non siamo in grado di ottenere dai datasheet i parametri dinamici, l'identificazione dei parametri dinamici può essere eseguite con le seguenti metodologie:

A. Metodo 1: si tratta di un approccio euristico in cui l'idea è smontare il manipolatore ed effettuare una serie di misurazioni direttamente sui suoi componenti. Tale tecnica come si può immaginare non è di facile attuazione;

B. Metodo 2: se siamo il costruttore del manipolatore potremmo impiegare i modelli CAD oppure se siamo l'utente possiamo richiedere tali modelli al costruttore. Grazie ai file CAD possiamo calcolare i valori dei parametri dinamici dei diversi componenti (collegamenti, attuatori e trasmissioni) in base alla loro geometria e tipo di materiale. Anche questo metodo presenta però un difetto che riguarda l'imprecisione delle stime dei parametri dinamici. Tale imprecisione è causata dai seguenti fattori:
- Semplificazioni nella modellazione geometrica;
- Effetti dinamici complessi come l'attrito non modellati.

C. Metodo 3: per capire l'idea di questo metodo ricordiamo che nella calibrazione cinematica abbiamo eseguito una linearizzazione nell'intorno dei valori nominali dei parametri e alla fine abbiamo effettuato una serie di raffinazioni di questi ultimi fino a quando non raggiungevamo una condizione in cui non era più possibile effettuare delle migliorie. Nel caso dinamico, seppur il modello è più articolato e ci sono fenomeni più complessi, esso gode della *proprietà di linearità* rispetto ai parametri dinamici: $\tau = Y(q, \dot{q}, \ddot{q})\pi$
Ciò si traduce praticamente in una estrema semplicità nell'identificazione dei parametri dinamici, per i quali infatti basterà una sola iterazione della procedura che vedremo di seguito.

Avendo compreso che il metodo 3 è quello corretto focalizziamoci su di esso. Essendo basato sulla proprietà di linearità rispetto ai parametri dinamici, partiamo proprio dalla suddetta relazione nella quale il nostro obiettivo centrale sarà dunque il vettore π: $\tau = Y(q, \dot{q}, \ddot{q})\pi$
L'aspetto interessante da sottolineare è che, seppur i valori dei parametri dinamici contenuti in π non sono esattamenti quelli reali, tali valori saranno tali che, per quel modello dinamico e insieme alle $q, \dot{q}, \ddot{q}, \tau$ vere, *minimizzano lo scarto* tra valori ricavati e reali.

Ora, il passo seguente per ricavare i parametri dinamici è calcolare le seguenti grandezze:
- q e $\dot{q}$ mediante misure dirette con encoder e tachimetri;
- $\ddot{q}$ mediante misure indirette con filtri ricostruttori;
- τ mediante misure dirette con sensori di coppia al giunto oppure misure indirette con misure di forza al polso o corrente nel caso di attuatori elettrici.

In particolare, dovremo eseguire una serie di misurazioni in determinati istanti di tempo $t_1, \dots, t_N$ lungo una data traiettoria, la quale deve essere scelta opportunamente secondo questi criteri:
- La traiettoria è preferibile che sia polinomiale cubica o al più quintica in quanto, all'aumentare del grado del poliniomio, aumenta il suo carattere oscillatorio. Ciò si traduce nell'insorgenza di effetti dinamici non modellati come l'elasticità articolare;
- La traiettoria deve essere sufficientemente ricca in modo da consentire una accurata individuazione dei parametri. Ciò si traduce nella pratica nell'effettuare un alto numero di misurazioni. Tipicamente si sceglie un numero N di misurazioni tale che $Nn \gg p$, in modo da evitare mal condizionamenti della matrice Y;
- La traeittoria non deve impiegare un profilo di velocità trapezoidale in quanto, nel tratto di crociera, l'accellerazione è nulla e come vedremo nella seguente relazione si ha che la coppia non dipenderà dai termini inerziali. A causa di ciò, in questa fase, non saremo in grado di ricavare i parametri dinamici.

Scelta la traiettoria secondo questi criteri, otterremo il seguente sistema di equazioni in cui, grazie alle misurazioni fatte, siamo passati da una matrice $Y \in R^{n\times p}$ bassa ad una matrice $\bar{Y} \in R^{(N\cdot n)\times p}$ alta. A noi interessano i parametri dinamici del vettore π e poiché abbiamo più equazioni che incognite, dovremo adottare una *soluzione ai mimini quadrati*. Tale soluzione è caratterizzata dalla **Matrice Pseudo-Inversa Sinistra** $(\bar{Y}^T\bar{Y})^{-1}\bar{Y}^T$:

$$\bar{\tau} = \begin{bmatrix} \tau(t_1) \\ \vdots \\ \tau(t_N) \end{bmatrix} = \begin{bmatrix} Y(t_1) \\ \vdots \\ Y(t_N) \end{bmatrix} \pi = \bar{Y}\pi \qquad \rightarrow \qquad \boldsymbol{\pi = (\bar{Y}^T\bar{Y})^{-1}\bar{Y}^T\bar{\tau}}$$

Perciò il problema dell'identificazione dei parametri dinamici si riconduce al calcolo di una pseudo-inversa sinistra. In realtà, possiamo pensare di scegliere a questo punto tra due strade:

a. **Metodo 1**: sapendo che la matrice Y ha una struttura triangolare alta destra (come mostrato di fianco), possiamo pensare di fare in modo che $\bar{Y}$ presenti la medesima struttura. In questo modo, partendo dall'ultima riga (quindi braccio), ricaviamo agevolmente π_n in quanto l'ultima riga della matrice $\bar{Y}$ avrà un solo elemento diverso da zero (un braccio non può dipendere da alcun parametro relativo ai bracci a monte ma solo a valle). Poi ripeto il procedimento con la riga di sopra dove questa volta non avremo $N-1$ valori nulla ma bensì $N-2$, però avremo anche che π_n è noto. Continuando a ritroso calcoleremo tutti i π_i. Questo approccio però presenta l'inconveniente di accumulare eventuali errori dovuti al mal condizionamento delle matrici coinvolte passo dopo passo;

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \cdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} y_{11}^T & y_{12}^T & \cdots & y_{1n}^T \\ 0^T & y_{22}^T & \cdots & y_{21}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \cdots & y_{nn}^T \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \cdots \\ \pi_n \end{bmatrix}$$

b. **Metodo 2**: banalmente calcoliamo per intero tutta la pseudo-inversa sinistra, il che si tratta di una procedura abbastanza onerosa a livello computazionale ma che ha il vantaggio di distribuire in maniera uniforme gli errori su tutte le componenti contrariamente all'approccio precedente.

A questo punto però bisogna fare alcuni ragionamenti che riguardano il rango della matrice $\bar{Y}$, la quale essendo alta presenterà un rango colonna:

a) **Se $\boldsymbol{rank(\bar{Y}) = p}$**: allora la matrice $\bar{Y}$ ha rango pieno e di conseguenza anche la matrice $\bar{Y}^T\bar{Y}$, che è quadrata, avrà rango pieno $rank(\bar{Y}^T\bar{Y}) = p$. Quest'ultima condizione ci garantisce l'invertibilità di $\bar{Y}^T\bar{Y}$ cosìcchè possiamo adottare la soluzione della pseudo-inversa sinistra citata in precedenza. Inoltre, in questa casistica, possiamo essere certi che *tutti* i parametri dinamici danno un contributo al modello dinamico;

b) **Se $\boldsymbol{rank(\bar{Y}) < p}$**: allora la matrice $\bar{Y}$ non ha rango pieno e di conseguenza la matrice $\bar{Y}^T\bar{Y}$ non è invertibile, per cui non possiamo impiegare la soluzione della pseudo-inversa sinistra. Il fatto che $\bar{Y}$ non sia di rango pieno è dovuta a due criticità:
   - **Parametri Non Identificabili**: si trattano di parametri che, a causa della disposizione dei link e dei giunti, non danno alcun contributo all'equazioni del moto per qualsiasi traiettoria assegnata al manipolatore. Una diretta conseguenza è che le colonne della matrice Y corrispondenti a tali parametri sono nulle e quindi devono essere rimosse dalla matrice stessa;
   - **Combinazione Lineare dei Parametri**: talvolta può capitare che uno o più parametri dinamici siano combinazione lineari di altri parametri. In tal caso è necessario, per ogni combinazione lineare, rimuovere tante colonne della matrice Y quanti sono i parametri della combinazione lineare meno uno.

   Allora per risolvere questi problemi possiamo pensare pensare di utilizzare una matrice ai minimi quadrati smortaza $Y^* = \bar{Y}^T\bar{Y} + k^2 I$ portando così ad un piccolo scostamento causato dal non perfetto svolgimento del problema (infatti $YY^* \cong I$).

Per concludere, citiamo un aspetto che non è stato in alcun modo considerato, ovvero la presenza di un carico utile sull'end-effector. In realtà, questo aspetto non è particolarmente complesso da integrare nel discorso appena fatto in quanto il carico può essere integrato come modificata strutturale dell'ultimo collegamento. Così facendo, si può procedere all'identificazione dei parametri dinamici del collegamento modificato. A tal fine, se è disponibile un sensore di forza al polso del manipolatore, è possibile caratterizzare direttamente i parametri dinamici del carico utile partendo dalle misurazioni del sensore di forza.

6.5 Scalatura Dinamica della Traiettoria

La **Scalatura Dinamica della Traiettoria** *è una tecnica che ridimensiona quest'ultima qualora essa violi i vincoli dinamici senza la necessità il ricalcolo della dinamica inversa.* Ricordiamo che nella pratica gli attuatori che impieghiamo presentano dei vincoli ben specifici, in particolare riguardo le massime coppie esprimibili. Talvolta può accadere che la traiettoria pianificata richieda delle prestazioni dinamiche troppo gravose. Allora, in prima battuta, si potrebbe procedere impiegando un tempo maggiore per percorrerla, il che però richiederebbe una ripianificazione della traiettoria. Proprio per evitare di fare ciò si impiega la scalatura dinamica, la quale in pratica adatta il movimento lungo il percorso specificato con una tempistica adeguata.

Consideriamo il modello dinamico del manipolatore in cui supponiamo per semplicità che $F_v = 0$, $F_s = 0$ e $h_e = 0$. Ricordiamo che il termine $C(q, \dot{q})$ tiene conto delle forze centrifughe e di Coriolis, le quali hanno una dipendenza quadratica dalle velocità dei giunti $\dot{q}$. Vedremo essere conveniente eseguire la sua fattorizzazione impiegando una notazione simbolica $[\dot{q}\dot{q}]$, la quale è un vettore colonna di $(n(n+1)/2)$ elementi: $[\dot{q}\dot{q}] = [\dot{q}_1^2 \quad \dot{q}_1\dot{q}_2 \quad \dots \quad \dot{q}_{n-1}\dot{q}_n]^T$
Ricordiamo poi di esplicitare la dipendeza dal tempo t dei diversi termini che compaiono nell'espressione. Otteniamo così i seguenti passaggi:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + F_s sgn(\dot{q}) + g(q) = \tau - J^T(q)h_e$$
$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + +g(q) = \tau$$
$$B(q)\ddot{q} + \Gamma(q)[\dot{q}\dot{q}] + g(q) = \tau$$
$$B\big(q(t)\big)\ddot{q}(t) + \Gamma\big(q(t)\big)[\dot{q}(t)\dot{q}(t)] + g\big(q(t)\big) = \tau(t)$$

Consideriamo ora una nuova funzione del tempo scalare $r(t)$ con il vincolo che $r(0) = 0$ e $r(t_f) = \bar{t}_f$ dove se l'obiettivo è rallentare $\bar{t}_f > t_f$. In tal modo posso riscrivere la $q(t)$ nel seguente modo: $q(t) = \bar{q}(r(t))$
Differenziando rispetto al tempo due volte:

$$\dot{q} = \dot{r}\bar{q}'(r)$$
$$\ddot{q} = \dot{r}^2\bar{q}''(r) + \ddot{r}\bar{q}'(r)$$

Sostituendole entrambe nella τ:

$$\dot{r}^2\big(B\big(\bar{q}(r)\big)\bar{q}''(r) + \Gamma\big(\bar{q}(r)\big)[\bar{q}'(r)\bar{q}'(r)]\big) + \ddot{r}B\big(\bar{q}(r)\big)\bar{q}'(r) + g\big(\bar{q}(r)\big) = \tau$$
$$\dot{r}^2\big(\bar{\tau}_s(t)\big) + \ddot{r}B\big(\bar{q}(r)\big)\bar{q}'(r) + g\big(\bar{q}(r)\big) = \tau$$
$$\big[\dot{r}^2\big(\bar{\tau}_s(t)\big) + \ddot{r}B\big(\bar{q}(r)\big)\bar{q}'(r)\big] + g\big(\bar{q}(r)\big) = \tau$$
$$\tau_s(t) + g\big(\bar{q}(r)\big) = \tau$$

Indichiamo con $\tau_s(t)$ i contributi di coppia dipendenti da velocità e accellerazioni richieste dal manipolatore quando è soggetto a moti aventi il medesimo percorso ma leggi di temporizzazioni differenti ottenute mediante scalatura temporale con $q(t) = \bar{q}\big(r(t)\big)$. Focalizziamoci su di essa in quanto le coppie gravitazionali sono solo funzione della posizione dei giunti e non sono in alcun modo inflienzate dalla scalatura temporale. La coppia $\tau_s(t)$ sarà pari a: $\tau_s(t) = \dot{r}^2\bar{\tau}_s(t) + \boldsymbol{\ddot{r}B\big(\bar{q}(r)\big)\bar{q}'(r)}$

Il problema di tale espressione è la presenza del termine in grassetto. Per eliminarlo, la scelta più semplice è prendere la funzione di scala $r(t)$ come funzione lineare: $r(t) = Ct$
Dove:

A. **Se $C > 1$ stiamo decellerando la traiettoria;**
B. **Se $C < 1$ stiamo accellerando la traiettoria.**

Grazie a questo artificio, arriviamo alla **Legge di Scalatura Dinamica**:

$$\boldsymbol{\tau_s(t) = C^2 \bar{\tau}_s(Ct)}$$

Osserviamo infine che, il problema citato inizialmente potrebbe riguardare diversi giunti. Allora si può pensare di utilizzare un algoritmo ricorsivo per il calcolo della dinamica inversa in cui si verifica se le coppie ai giunti superano i limiti consentiti durante l'esecuzione della traiettoria. Identificati i giunti in cui avviene la violazione si prende solo il giunto i che ha la violazione maggiore e mediante la formula di fianco si calcola il fattore di scalatura C. Così facendo stiamo andando a considerare il caso peggiore in modo da garantire il rispetto dei vincoli per tutti i giunti. Dovrebbe essere evidenziato, tuttavia, che con questo tipo di ridimensionamento lineare l'intera traiettoria può essere penalizzata, anche quando un limite di coppia su un singolo giunto viene superato solo per un breve intervallo di tempo.

$$\frac{|\tau_s|}{|\bar{\tau}_i + g(q_i)|} = C^2$$

6.6 Modello Dinamico nello Spazio Operativo

Il **Modello Dinamico nello Spazio Operativo** è una modalità alternativa per descrivere le equazioni del moto del manipolatore, questa volta nello spazio operativo anziché in quello dei giunti, in funzione delle forze e dei momenti agenti su di esso. Dunque questo modello dinamico vuole descrivere le relazioni tra le forze generalizzate e il numero minimo di variabili che esprimono la posa dell'end-effector nello spazio operativo.

$$\boldsymbol{B_A(x_e)\ddot{x}_e + C_A(x_e,\dot{x}_e)\dot{x}_e + g_A(x_e) = \gamma_A - h_A}$$

Il pedice A vuole evidenziare che si trattano di termini differenti da quelli visti per il modello dinamico nello spazio dei giunti. Tipicamente questo modello viene impiegato nell'ambito interazione robot-ambiente. Come il modello dinamico nello spazio dei giunti, anche quello nello spazio operativo deve trattare con dovuta accortezza ai casi di:

- **Ridondanza**: se $r < n$, in ogni caso il sistema si muove con n gradi di libertà. Allora utilizzando la formulazione di Lagrange, quindi un approccio energetico, a causa della ridondanza non abbiamo una descrizione completa della dinamica del manipolatore ridondante. Infatti, in questo caso, è ragionevole aspettarsi il verificarsi di moti interni della struttura causati da quelle forze generalizzate ai giunti che non influenzano il movimento dell'effettore finale.
- **Singolarità**: a prescindere dalla ridondanza, quindi se lo Jacobiano sia quadrato oppure rettangolare, se per esempio il manipolatore si trova in singolarità di spalla tale matrice perde di rango. In questo caso ruotando intorno ad un certo asse non cambia la posizione dell'end-effector, ma ci sarà comunque la presenza di un moto interno e quindi ritorniamo al discorso citato per la ridondanza.

Allora potremmo chiederci per quale motivo adottare il modello dinamico nello spazio operativo viste queste problematiche. Essenzialmente l'idea sarebbe gestire ridondanza e singolarità direttamente nel loop di controllo, anziché nello stadio a monte di questo ultimo di inversione cinematica come avviene nel modello dinamico nello spazio dei giunti.

Vediamo come pervenire al modello dinamico nello spazio operativo. Prima di tutto riprendiamo il modello dinamico nello spazio dei giunti in cui, per semplicità, trascuriamo gli attriti:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J^T(q)h_e$$

Ora possiamo pensare che, invece di scegliere come coordinate generalizzate q e $\dot{q}$, consideriamo x_e e $\dot{x}_e$. Allora se vogliamo scrivere una equazione in termini di x_e e le sue derivate, dobbiamo capire come queste sono legate a q. Queste equazioni le possiamo ricavare partendo dall'equazione cinematica diretta e derivando due volte quest'ultima. Arrivati a questo punto, il passo fondamentale è unire la cinematica del secondo ordine $\ddot{x}_e = J_A(q)\ddot{q} + \dot{J}_A(q,\dot{q})\dot{q}$ e il modello dinamico nello spazio dei giunti. Per fare ciò abbiamo due strade:

$$x_e = k(q)$$
$$\dot{x}_e = J_A(q)\dot{q}$$
$$\ddot{x}_e = J_A(q)\ddot{q} + \dot{J}_A(q,\dot{q})\dot{q}$$

A. **Metodo 1**: calcoliamo $\ddot{q}$ dal modello dinamico nello spazio dei giunti e sostituisco in $\ddot{x}_e$;
B. **Metodo 2**: calcoliamo $\ddot{q}$ dalla cinematica differenziale del secondo ordine e sostituiamo nel modello dinamico nello spazio dei giunti.

Prima di procedere è doveroso osservare che nelle equazioni di $\dot{x}_e$ e $\ddot{x}_e$ non a caso è presente lo Jacobiano analitico, visto che ci troviamo nello spazio operativo. Se volessimo avere una trattazione più precisa dovremmo fare riferimento allo Jacobiano geometrico, tuttavia questo a spese di dover considere un orientamento espresso in modo differente dagli angoli di Eulero. Quest'ultimo aspetto di fatto porta ad una maggiore complessità della trattazione analitica.

Ora visto che la matrice $B > 0$ essa sarà certamente sempre invertibile, per cui conviene seguire il metodo 1: $\ddot{q} = -B^{-1}C\dot{q} - B^{-1}g + B^{-1}J^T(\gamma_e - h_e)$
Ricordando che $J^T\gamma_e = \tau$ dalla statica e $\tau = J_A^T\gamma_A$ (analogamente per $J^T h_e$), sostituiamo ora $\ddot{q}$ in $\ddot{x}_e$:

$$\begin{aligned}\ddot{x}_e &= J_A\ddot{q} + \dot{J}_A\dot{q}\\ &= J_A[-B^{-1}C\dot{q} - B^{-1}g + B^{-1}J^T(\gamma_e - h_e)] + \dot{J}_A\dot{q}\\ &= -J_AB^{-1}C\dot{q} - J_AB^{-1}g + J_AB^{-1}J_A^T(\gamma_e - h_e) + \dot{J}_A\dot{q}\\ &= -J_AB^{-1}C\dot{q} - J_AB^{-1}g + J_AB^{-1}J_A^T(\gamma_A - h_A) + \dot{J}_A\dot{q}\end{aligned}$$

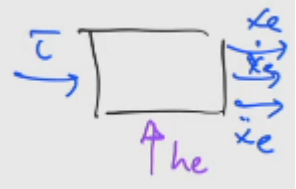
Arrivati a questo punto dobbiamo cercare di isolare il termine $(\gamma_A - h_A)$, quindi portiamo tutti gli altri termini a primo membro e poi dividiamo entrambi per $J_A(q)B^{-1}(q)J_A^T(q)$, ovvero moltiplichiamo per l'inversa di questa matrice. Imponendo $B_A(x_e) = (J_A(q)B^{-1}(q)J_A^T)^{-1}$ consideriamo due casi:

a. **Manipolatore Non Ridondante $\boldsymbol{n = r}$**: tutte le matrici che compogono B_A sono quadrate e quindi: $B_A = (J_AB^{-1}J_A^T)^{-1} = J_A^{-T}BJ_A^{-1}$
   Dopodichè, definendo le seguenti relazioni riusciamo ad ottenere il modello dinamico descritto inizialmente:
   - $C_A\dot{x}_e = J_A^{-T}C\dot{q} - B_A\dot{J}_A\dot{q}$
   - $g_A = J_A^{-T}g$
b. **Manipolatore Ridondante $\boldsymbol{n > r}$**: $B_A = (J_AB^{-1}J_A^T)^{-1}$
   Dopodichè, definendo le seguenti relazioni riusciamo ad ottenere il modello dinamico descritto inizialmente:
   - $C_A\dot{x}_e = B_A(x_e)J_A(q)B^{-1}(q)C(q,\dot{q}) - B_A(x_e)\dot{J}_A(q,\dot{q})$
   - $g_A = B_A(q)J_A(q)B^{-1}(q)g(q)$

Una piccola osservazione da fare è che in realtà, con la formulazione Lagrangiana, non riusciamo ad isolare la matrice C_A in quanto tutto il secondo membro, formalmente, è $C_A\dot{x}_e$. Risultato di ciò C_A è un $C_A(x_e, \dot{x}_e)$ come scritto nel modello dinamico.

Concludiamo perciò che, sia nel caso di un manipolatore ridondante che non ridondante riusciamo a pervenire ad una formulazione del modello dinamico nello spazio operativo analoga a quanto enunciato inizialmente: $\boldsymbol{B_A(x_e)\ddot{x}_e + C_A(x_e,\dot{x}_e)\dot{x}_e + g_A(x_e) = \gamma_A - h_A}$
L'unica differenza è che il suddetto modello, nel caso della ridondanza, non riesce a descrivere completamente la dinamica del sistema in quanto ragiona sullo spazio operativo che impiega meno variabili. Tuttavia, un aspetto fondamentale che accumuna entrambe le casistiche è la loro validità in *assenza di singolarità cinematiche*. In entrambi i casi, infatti, lo Jacobiano analitico perde di rango e la matrice $J_A(q)B^{-1}(q)J_A^T$ non risulta essere più invertibile. Conseguenza di ciò non possiamo affermare che B_A è una matrice di inerzia in quanto è solo semi-definitiva positiva, ovvero $B_A \geq 0$. Fisicamente vuol dire che ci saranno delle velocità $\dot{x}_e$ che non generano energia cinetica.

Andiamo ora a risolvere il problema della dinamica diretta e della dinamica inversa con il modello dinamico nello spazio operativo.



- **Dinamica Diretta**: note τ e h_e vogliamo ricavare $x_e, \dot{x}_e, \ddot{x}_e$. Per farlo sviluppiamo una soluzione in due stadi:
  1. Dinamica diretta nello spazio dei giunti.
  2. Cinematica diretta.

- **Dinamica Inversa**: note $h_e, x_e, \dot{x}_e, \ddot{x}_e$ vogliamo ricavere τ. Per farlo possiamo scegliere due differenti strade ciascuna delle quali con due stadi:
  a) Risolviamo la ridondanza a livello cinematico:
     1. Cinematica differenziale inversa da cui calcolo $\dot{q}$
     2. Dinamica inversa nello spazio dei giunti.
  b) Risolviamo la ridondanza a livello dinamico:
     1. Modello dinamico nello spazio operativo (ricavo γ_A);
     2. Coppie con gestione dei moti interni.

Soffermiamoci sulla strada (b) della ridondanza a livello dinamico. In realtà, noi non possiamo scrivere semplicemente $\tau = J_A^T \gamma_A$ in quanto, con la ridondanza, abbiamo dei gradi di libertà in più. Partiamo dal modello dinamico nello spazio operativo:

$$\begin{aligned}
\gamma_A - h_A &= B_A(x_e)\ddot{x}_e + C_A(x_e, \dot{x}_e)\dot{x}_e + g_A(x_e) \\
&= B_A(x_e)\ddot{x}_e + \left[-B_A\dot{J}_A\dot{q} + B_A J_A B^{-1} C\dot{q}\right] + g_A(x_e) \\
&= B_A\left(\ddot{x}_e - \dot{J}_A\dot{q}\right) + B_A J_A B^{-1} C\dot{q} + B_A J_A B^{-1} g \\
&= B_A J_A \ddot{q} + B_A J_A B^{-1} C\dot{q} + B_A J_A B^{-1} g
\end{aligned}$$

Definiamo la **Pseudo-Inversa Destra Pesata Secondo $\boldsymbol{B^{-1}}$ $\boldsymbol{\bar{J}_A^T}$**: $\boldsymbol{\bar{J}_A^T = B_A(q) J_A(q) B^{-1}(q)}$
Ottenendo così:

$$\gamma_A - h_A = \bar{J}_A^T(B\ddot{q} + C\dot{q} + g) = \bar{J}_A^T(\tau - J_A^T h_A)$$

Da cui possiamo ricavare la coppia τ che risolve la ridondanza a livello dinamico, in cui τ_0 sono sono coppie arbitrarie che non provocano forze in punta (in quanto $\bar{J}_A^T(q)$ moltiplicato per la parentesi tonda da come risultato zero):

$$\boldsymbol{\tau = J_A^T(q)\gamma_A + \left(I - J_A^T(q)\bar{J}_A^T(q)\right)\tau_0}$$

6.7 Ellisoide di Manipolabilità Dinamica

L' **Ellisoide di Manipolabilità Dinamica** *rappresenta l'attitudine del manipolatore a generare dei moti in accelerazione all'organo terminale in una data configurazione nelle diverse direzioni*. Come vedremo tale ellissoide può essere impiegato come strumento di analisi delle prestazioni dinamiche del manipolatore.

Affrontiamo lo studio dell'ellisoide di manipolabilità dinamica supponendo anzitutto che lo spazio delle coppie di giunto sia una sfera a norma costante unitaria, perciò può essere descritto mediante l'equazione: $\tau^T\tau = 1$

L'obiettivo è descrivere le accelerazioni spaziali operative che possono essere generate dalle coppie τ. Per affrontare questo studio consideriamo le seguenti ipotesi:

- ✓ Studiamo il manipolatore in una specifica configurazione, perciò $q = costante$ e da ciò possiamo affermare che la velocità $\dot{q} = 0$;
- ✓ Il manipolatore non è contatto con l'ambiente, perciò $h_e = 0$.

Il modello dinamico presenterà dunque a primo membro sono i termini inerziali e quelli di accellerazione di gravità:

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

Dalla cinematica differenziale del secondo ordine (a sinistra) possiamo ricavare l'accellerazione dei giunti: $\dot{v}_e = J\ddot{q} + \dot{J}\dot{q} = J\ddot{q} + \dot{J}\cdot 0 = J\ddot{q} \quad \rightarrow \quad \ddot{q} = J^+(q)\dot{v}_e$

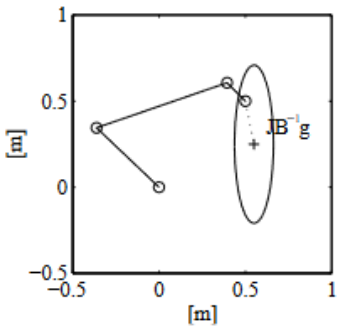
Sostituiamo $\ddot{q}$ nel modello dinamico: $B(q)J^+(q)\dot{v}_e + g(q) = \tau$

Andiamo a studiare nel dettaglio il termine a primo membro:

$$\begin{aligned} B(q)J^+(q)\dot{v}_e + g(q) &= B(J^+\dot{v}_e + B^{-1}g) \\ &= B(J^+\dot{v}_e + B^{-1}g + J^+JB^{-1}g - J^+JB^{-1}g) \\ &= B(J^+\dot{v}_e + J^+JB^{-1}g + B^{-1}g - J^+JB^{-1}g) \\ &= B(J^+\dot{v}_e + J^+JB^{-1}g + (I - J^+J)B^{-1}g) \\ &= BJ^+(\dot{v}_e + JB^{-1}g) \end{aligned}$$

E' da osservare che il termine $(I - J^+J)B^{-1}g$ è stato trascurato perché $B^{-1}g$ sono le accellerazioni che competono alle coppie di gravità. Queste accellerazioni sono proiettate nel nullo dello Jacobiano e dunque non danno alcun contributo all'accelerazione all'organo terminale. A questo punto avendo espresso $\tau = BJ^+(\dot{v}_e + JB^{-1}g)$, sostituendo tale relazione in $\tau^T\tau = 1$ ricaviamo l'equazione che definisce l'ellissoide di manipolabilità dinamica:

$$\boldsymbol{(\dot{v}_e + JB^{-1}g)^T J^{+T} B^T B J^+ (\dot{v}_e + JB^{-1}g) = 1}$$

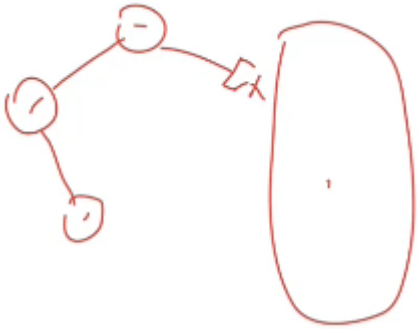


Il nucleo dell'ellissoide è la matrice $J^{+T}B^TBJ^+$, la quale dipende dalle caratteristiche geoemtriche e inerziali del manipolatore, inoltre determine il volume e gli assi principali dell'ellissoide.

La presenza nell'equazione dell'ellissoide del vettore $JB^{-1}g$ produce una traslazione costante del centro dell'ellissoide (per

ogni configurazione di manipolatore) rispetto all'origine del sistema di riferimento come mostrato in figura. In effetti questo ha anche un significato fisico in quanto è più facile accellerare verso il basso anziché verso l'alto andando contro la forza di gravità.

Un caso molto particolare è quello mostrato in figura nel quale, se l'ellissoide di manipolabilità dinamica non intercetta l'organo terminale allora vuol dire che quest'ultimo non è in grado di realizzare un'accellerazione (peso eccessivo). Dunque l'ellissoide può essere utilizzato come strumento di verificare di fattibilità delle accellerazioni a parità di coppie spese.



7. Controllo del Moto

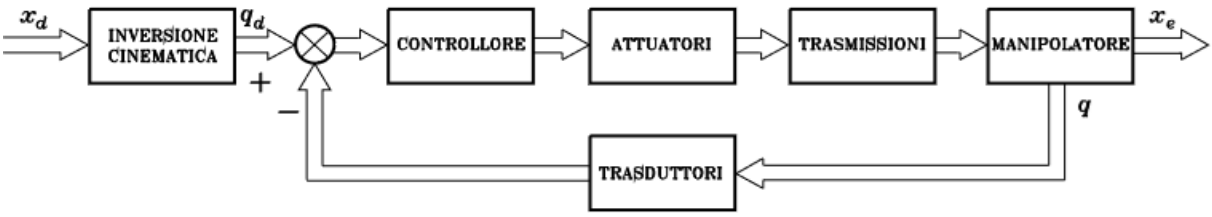
7.1 Problema del Controllo del Moto

Il **Controllo del Moto** consiste nel ricavare le forze generalizzate (forze e momenti) affinchè il manipolatore svolga la traiettoria pianificata. Indipendentemente dal tipo di manipolatore utilizzato, è importante notare che mentre il compito (task) che il robot dovrà eseguire è normalmente formulato nello spazio operativo, in termini di moto dell'organo terminale, le azioni di controllo sono invece applicate nello spazio dei giunti. E' quindi naturale considerare un controllo del moto nello spazio dei giunti e un controllo del moto nello spazio operativo. Gli schemi che implementano questi due controlli, seppur con le loro peculiarità, presentano degli aspetti come:

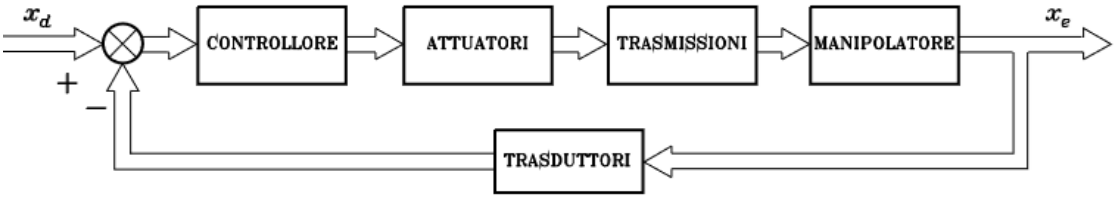
- Ingresso la traiettoria pianificata x_d e uscita la posa dell'end-effector x_e;
- Sfruttano la retroazione vista la sua robustezza a possibili incertezze del modello e reiezione degli effetti di disturbo (sulla linea diretta);
- Presentano dei trasduttori sulla linea di retroazione.

Identifichiamo ora gli schemi a blocci dei due tipi di controllo del moto con le relative particolarità:

A. Controllo nello Spazio dei Giunti: tale controllo si articola in due sotto-problemi. Il primo è anzitutto risolvere la cinematica inversa in modo da ricavare, a partire dai riferimenti x_d nello spazio operativo, dei riferimenti q_d nello spazio dei giunti. Allora lo scopo di questo controllo è ricavare le coppie τ tale che $q = q_d$. Questa soluzione, comunemente adottata nei manipolatori industriali, presenta tuttavia lo svantaggio di controllare le variabili nello spazio operativo (ovvero x_e) in anello-aperto. Come conseguenza qualsiasi incertezza nella struttura meccanica (tolleranza di costruzione, mancanza di calibrazione, gioco degli ingranaggi, elasticità) e qualsiasi imprecisione nella conoscenza della posa dell'organo terminale rispetto all'oggetto manipolato possono produrre errori di posizionamento.



B. Controllo nello Spazio Operativo: tale controllo ha il vantaggio concettuale di agire direttamente nelle variabili nello spazio operativo, al contempo però è richiesta una misura delle variabili nello spazio operativo, generalmente ottenuta in maniera indiretta mediante il calcolo della cinematica diretta a partire dalla misura delle variabili di giunto. Inoltre, la cinetica inversa è inserita all'interno del loop di controllo. Il principale svataggio è che il controllo dello spazio operativo richiede una complessità algoritmica notevolmente maggiore rispetto al controllo nello spazio dei giunti. Infine, questo controllo tipicamente viene impiegato per la specifica di un certo task e quando abbiamo l'interazione del robot con l'ambiente.



7.2 Controllo del Moto nello Spazio dei Giunti

Anzitutto riprendiamo il modello dinamico ottenuto nel capitolo precedente con la formulazione Lagrangiana. Per semplicità, supponiamo di trascurare il termine relativo all'attrito statico e mantenere solo quello di attrito viscoso.

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

Il problema del controllo è quello di determinare le coppie ai giunti τ in modo che la traiettoria reale sia uguale a quella desiderata: $q(t) = q_d(t)$

Tale problema può però presentare due sfacciattature:

- **Tracking**: il problema di controllo si esplica come inseguimento di un ingresso di riferimento;
- **Regolazione**: il problema di controllo si esplica come mantenenimento dell'uscita al valore desiderato $q_d = cost$.

In entrambe la tipologia di problemi, anche se in particolar modo per il tracking, sarà necessario tollerare un minimo errore tra q e q_d. Anzi sarà da scegliere se privilegiare un errore di posizione (vogliamo che sia piccolo) a scapito di un errore di velocità maggiore o viceversa. Ricordiamo che il manipolatore è un sistema meccanico, per cui certamente sarà dotato di trasmissioni, le cui relazioni sono note dallo studio degli attuatori. In effetti, se $k_r\dot{\theta} = \dot{\theta}_m$ e ciò deve valere per tutti i giunti possiamo passare ad una forma matriciale in modo da considerare tutte le variabili di giunto $K_r\dot{q} = \dot{q}_m$. Questa relazione però se ci pensiamo, come deve valere per le velocità certamente deve valere anche per le posizioni: $q_m = K_r q$

Riprendendo la relazione $K_r\dot{q} = \dot{q}_m$, sfruttiamo la dualità cineto-statica per ricavare invece la seguente relazione: $\tau = K_r^T\tau_m$

Poiché certamente K_r è diagonale e quadrata sarà certamente invertivile per cui possiamo scrivere: $\tau_m = K_r^{-1}\tau$

Inoltre, il sistema sarà dotato di n azionamenti che possono essere descritti in forma compatta dalla seguenti equazioni dove la prima legala seconda è l'equilibrio elettrico e la terza lega la parte meccanica e quella elettrica:

- **Prima Equazione**: lega la parte meccanica a quella elettrica in quanto leghiamo la coppia alla corrente del circuito di armatura: $K_r^{-1}\tau = K_t i_a$
- **Seconda Equazione**: è l'equilibrio elettrico del circuito di armatura: $v_a = R_a i_a + K_v\dot{q}$
- **Terza Equazione**: la tensione sul circuito di armatura deve essere l'uscita dell'azione di controllo, per cui sarà necessario un amplificatore con guadagno G_v. In realtà, ci dovrebbe essere G_v diviso s moltiplicato per la costante di tempo, però poiché quest'ultima ormai è dell'ordine del kHz può essere trascurata. In pratica stiamo dicendo che il legame tra v_a e v_c è istantaneo. $v_a = G_v v_c$

Dove:

- K_t = matrice diagonale delle costanti di coppia;
- i_a = vettore delle correnti di armatura degli n motori;
- v_a = vettore delle tensioni di armatura;
- R_a = matrice diagonale delle resistenze di armatarua;
- K_v = matrice diagonale delle costanti di tensione degli n motori;
- G_v = matrice diagonale dei guadagni degli n amplificatori;
- v_c = vettore delle tensioni di comando degli n servomotori.

L'azionamento può essere controllato mediante due differenti tecniche:

A. Manipolatore Controllato in Velocità: noi abbiamo indicato con τ le coppie ai giunti, però dobbiamo pensare che la presenza di trasmissioni meccanica tra queste e le coppie lato motore τ_m c'è un matrice K_r. A noi però per l'azione di controllo interessa il lato motore, anziché il lato giunti che è di fatto il lato carico. A sua volta l'azione di controllo è rappresentata dalla v_c che, mediante l'amplificatore, diventa la tensione di armatura v_a che alimenta il motore.
Indicando con u la coppia di controllo che devo sintizzare, essa sarà:

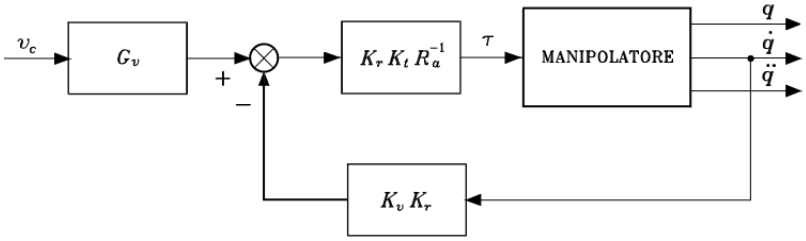
$$u = K_r K_t R_a^{-1} G_v v_c$$

Grazie ad essa possiamo riscrivere il modello dinamico in cui il termine legato alla velocità $\dot{q}$ sarà: $F = F_v + K_r K_t R_a^{-1} K_v K_r$
Dunque il modello dinamico sarà: $B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) = u$
Allora a causa della presenza della maglia del circuito di armatura, il legame tra tensione di controllo v_c e coppia ai giunti τ. Da questo legame possiamo ricavare proprio τ:

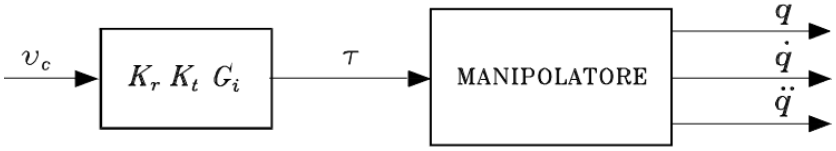
$$u = K_r K_t R_a^{-1} G_v v_c = \tau + K_r K_t R_a^{-1} K_v \dot{q}_m = \tau + K_r K_t R_a^{-1} K_v K_r \dot{q}$$
$$\downarrow$$
$$\tau = K_r K_t R_a^{-1} (G_v v_c - K_v K_r \dot{q})$$

Ora noi sappiamo che K_r ha elementi $\gg 1$ e R_a ha valori piccoli per cui R_a^{-1} ha valori molto grandi. Un buon sistema di controllo è tale se τ ha valori limitati e da ciò ne deriva che la parentesi tonda deve assumere valori limitati. In definitiva, il segnale di controllo v_c, a meno di tutte le matrici in parentesi, la velocità $\dot{q}$.

Se ci mettiamo in un ottica di controllo decentralizzato, ovvero l'azione di controllo su ciascun giunto è indipendente dagli altri giunti, si ha che: $G_v v_c \approx K_v K_r \dot{q}$
Quanto detto può essere implementato mediante il seguente schema che ragruppa sia il manipolatore che gli attuatori a monte.



B. Manipolatore Controllato in Coppia: se il mio obiettivo è una riduzione della sensibilità alle variazioni parametriche posso pensare di eseguire una retroazione della corrente di armatura: $i_a = G_i v_c$
Da cui otteniamo il legame diretto tra τ e u: $\tau = u = K_r K_t G_i v_c$



7.3 Controllo Decentralizzato

7.3.1 Modello Decentralizzato

Il **Controllo Decentralizzato nello Spazio dei Giunti** *consiste nel controllare la struttura robotica vedendola come un sistema indipendente di* n *azionamenti di giunto ad un ingresso e una uscita e in cui gli effetti di accoppiamento sono trattati come ingressi di disturbo*.

Prima di affrontare le tecniche con cui si realizza il controllo decentralizzato, c'è da dire che esso si basa su una costituzione del sistema manipolatore che a sua volta sia decentralizzato. Inoltre, altro aspetto da tenere presente è che il controllo non viene effettuato lato carico/giunti ma bensì lato motore. Allora il nostro scopo ultimo, date queste premesse, sarà ricavare il sistema manipolatore decentralizzato. Anzitutto, a partire dal modello dinamico del manipolatore: $B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + g(q) = \tau$

Dobbiamo ricavare il modello dinamico lato motore e quindi dobbiamo introdurre le seguenti relazioni:

- $\tau_m = K_r^{-1}\tau$
- $\ddot{q}_m = K_r\ddot{q}$ (stessa relazione vale per $\dot{q}_m$ e q_m)

Otteniamo così il seguente modello dinamico lato motore:

$$B\ddot{q} + C\dot{q} + F_v\dot{q} + g = \tau$$
$$BK_r^{-1}\ddot{q}_m + CK_r^{-1}\dot{q}_m + F_vK_r^{-1}\dot{q}_m + g = K_r\tau_m$$
$$K_r^{-1}BK_r^{-1}\ddot{q}_m + K_r^{-1}CK_r^{-1}\dot{q}_m + K_r^{-1}F_vK_r^{-1}\dot{q}_m + K_r^{-1}g = \tau_m$$

A questo punto osserviamo che la matrice di inerzia può essere scomposta nella somma di un termine diagonale costante e di un termine residuo che può essere trascurato in presenza di *rapporti di riduzione elevati*: $B(q) = \bar{B} + \Delta B(q)$

Grazie a questo artificio possiamo al **Modello Decentralizzato** del manipolatore lato motore come:

$$\boldsymbol{K_r^{-1}\bar{B}K_r^{-1}\ddot{q}_m + F_m\dot{q}_m + d = \tau_m}$$

Dove:

- $F_m = K_r^{-1}F_vK_r^{-1}$ = attrito viscoso;
- $d = K_r^{-1}\Delta B(q)K_r^{-1}\ddot{q}_m + K_r^{-1}C(q,\dot{q})K_r^{-1}\dot{q}_m + K_r^{-1}g(q)$ = disturbo.

Prima di definire lo schema a blocchi di tale modello è necessario analizzare se effettivamente d è una quantità limitata che possiamo definire come disturbo. Infatti, se così non fosse non potremmo chiudere un anello di controllo su quello che vedremo essere la parte lineare del sistema. Osservando le matrici a blocchi che compongono d:

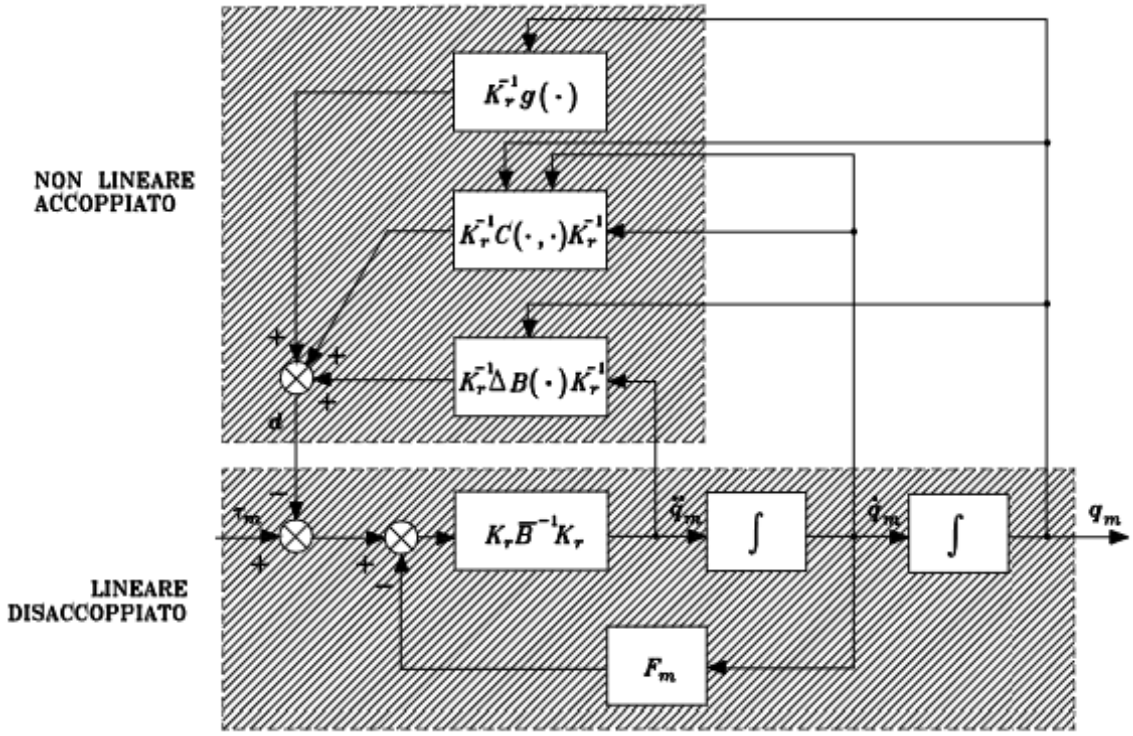
- Una variazione dell'inerzia ai giunti è scalata (divisa) di una quantità K_r^2;
- Una variazione quadratica di velocità, quindi parliamo dei termini di Corilis e forza centrifuga, è scalata di una quantità K_r^2;
- Il termine di gravità è scalato di K_r.

Perciò, seppur la presenza del riduttore porta a svantaggi come attriti, gioghi, ecc., esso contribuisce, grazie all'elevato rapporto di trasmissione, di garantire che il disturbo d sia limitato. In effetti tutto quello che abbiamo detto fino ad ora si poggia sulla condizione di avere elevati rapporti di trasmissione perché altrimenti sarebbe poco giusticato integrare in un disturbo la variabilità dell'inerzia del carico e gli effetti di accoppiamento tra i giunti.

In conclusione, possiamo rappresentare lo schema a blocchi del modello decentralizzato composto da due sotto-sistemi:

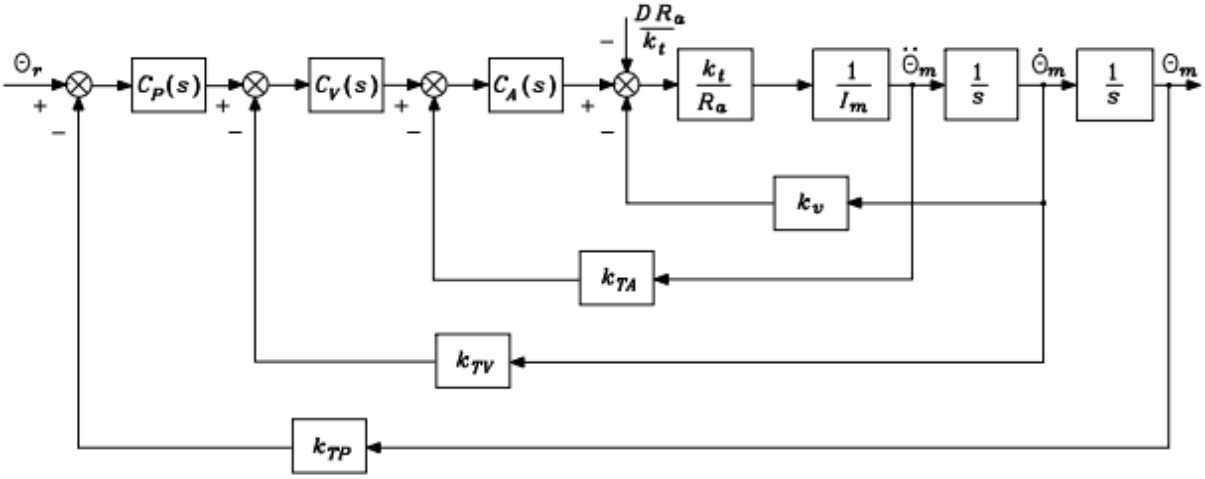
- **Sistema Lineare Disaccoppiato**: l'ingresso è τ_m e l'uscita è q_m;
- **Sistema Non Lineare Accoppiato**: che ha come ingressi $\ddot{q}_m, \dot{q}_m, q_m$ e uscita il disturbo d appena definito che agisce sul sistema lineare disaccoppiato.

Così facendo, la dinamica non-lineare e accoppiata del robot può essere approssimata con quella di un sistema lineare completamente disaccoppiato soggetto a un disturbo.



7.3.2 Controllo Indipendente ai Giunti

Il **Controllo Indipendente ai Giunti** *consiste nel controllare ciascuna variabile di giunto mediante uno o più anelli di retroazione che riportano all'ingresso posizione, velocità e accellerazione.* Lo schema a blocchi generale che implementa il controllo indipendente ai giunti è il seguente in cui, partendo da quello più esterno, identifichiamo l'anello di retroazione della posizione Θ_m, della velocità $\dot{\Theta}_m$ e accelerazione $\ddot{\Theta}_m$.



A seguito andremo a studiare questo schema generale considerando la presenza sono dell'anello di posizione e poi aggiungeremo via via gli altri. Prima di fare ciò dobbiamo però chiederci che tipo di controllore dobbiamo adottare per rigettare il disturbo. Il **Controllore PI** (**Proporzionale-Integrale**) risulta essere il più adatto per ridurre gli effetti di d su Θ:

$$\boldsymbol{C(s) = k_P + \frac{k_I}{s} = K_c \frac{1 + sT_c}{s}}$$

Esso infatti garantisce::

- ✓ Elevato valore del guadagno degli amplificatori a monte del punto di applicazione del disturbo;
- ✓ Annullare a regime (per θ costante) l'effetto della componente gravitazionale sull'uscita.

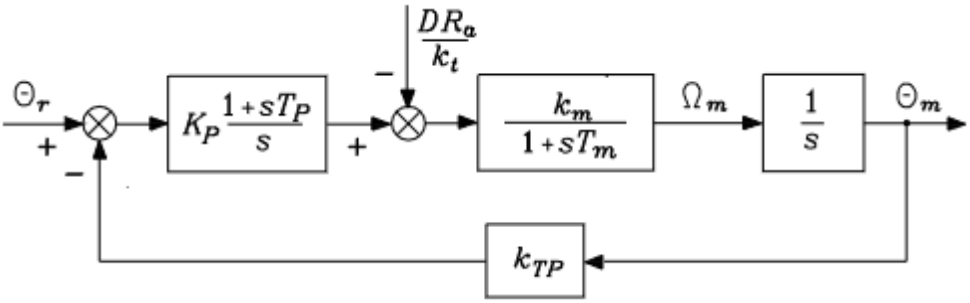
E' da notare che la sola azione integrale non basta perché, è vero che risulta essere centrale per annullare la componente gravitazionale a regime, però su un sistema del secondo ordine introduce un ritardo di fase che ne pregiudica i margini di stabilità del sistema. Una azione puramente integrale, quindi l'introduzione di un polo nell'origine, è destabilizzante per cui si mitica ciò inserendo un guadagno integrale k_I molto piccolo. A sua volta un k_I molto piccolo riduce la pulsazione critica del sistema e dunque la banda passante. Di conseguenza abbiamo un tempo maggiore per andare a regime e una minore prontezza ai disturbi. Per risolvere questi problemi è necessario inserire anche una azione proporzionale.

Ultimo aspetto da osservare circa gli schemi che studieremo di seguito è che il controllore PI deve essere sempre il controllore relativo all'*anello più interno*. Infatti, non seguendo questo criterio, il controllore PI agirà su un disturbo diverso da quello per cui è stato progettato in quanto non agisce su d ma bensì su quest'ultimo già circolato nell'anello di retroazione più interno.

Analizziamo ad analizzare i diversi schemi derivati da quello generale:

- **Retroazione di Posizione**: essa si realizza mediante il seguente schema. In questo caso abbiamo:

$$C_P(s) = K_P \frac{1 + sT_P}{s} \qquad C_V(s) = 1 \qquad C_A(s) = 1 \qquad k_{TV} = k_{TA} = 0$$



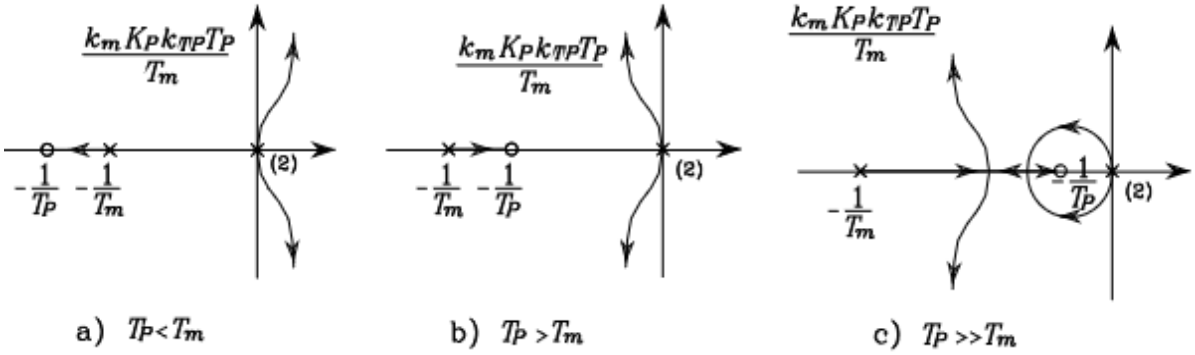
In pratica stiamo impiegando lo schema visto nel capitolo sugli attuatori per il controllo di posizione. Abbiamo infatti un controllore PI e i due blocchi di seguito sono la funzione di trasferimento tra l'ingresso di controllo e la posizione dell'attuatore. La funzione di trasferimento in questione è la seguente (ricordiamo che stiamo usando il generatore controllato di velocità):

$$M(s) = \frac{k_m}{s(1 + sT_m)} \qquad dove: \quad k_m = \frac{1}{k_v} \qquad T_m = \frac{R_a I_m}{k_v k_t}$$

Ora, se io devo generare un controllo in tensione v_c vuol dire che il disturbo in coppia d devo renderlo equivalente ad un disturbo in tensione. Il disturbo d è una coppia, per cui diviso per k_t diventa una corrente, se moltiplichiamo per la R_a del circuito di armatura ottengo una tensione. La f.d.t del ramo di azione diretta e quella del ramo in retroazione sono rispettivamente:

$$C(s)G(s) = \left[K_P \frac{1 + sT_P}{s}\right]\left[\frac{k_m}{s(1 + sT_m)}\frac{1}{s}\right] = \frac{k_m K_P (1 + sT_P)}{s^2(1 + sT_m)} \qquad H(s) = k_{TP}$$

Andiamo ad eseguire una analisi con il luogo delle radici in cui la costante K_P può essere visto come la **Costante di Trasferimento del Luogo delle Radici** che ci consente di tarare quest'ultimo. L'utilizzo del luogo presuppone una certa conoscenza del motore o in altre parole dei parametri k_m e T_m. Nella funzione di trasferimento CG abbiamo tre poli fissi e lo zero $-1/T_P$ la cui posizione dipende da T_P. Studiamo perciò diversi casi in relazione a T_m. Il caso $T_P = T_m$ si scarta apriori perché comporta una cancellazione polo/zero che porta il l.d.r. ai limiti della regione di stabilità. Anche il caso $T_P < T_m$ è da scartare perché è una situazione instabile vista la presenza dei rami nel semi-piano destro. Da ciò capiamo che *lo zero deve essere sempre alla destra del polo* come nei casi (b) e (c) in cui il baricentro del luogo si trova nel semi-piano sinistro.



a) $T_P < T_m$ b) $T_P > T_m$ c) $T_P >> T_m$

Come si può osservare il caso (c) è ancora più conveniente perché avvicinando lo zero all'asse reale il baricentro si sposta sempre più a sinistra. Il risultato di ciò è che, a parità di pulsazione naturale, il sistema è più veloce perché riduciamo la costante di tempo ed inoltre abbiamo anche una riduzione della sovraelongazione.

Tipicamente per l'analisi delle prestazioni del sistema si studia la f.d.t. a ciclo chiuso $\Theta_m(s)/\Theta_r(s)$, la quale può essere nella forma indicata da $W(s)$ in cui si evidenziano la frequenza naturale ω_n e lo smorzamento ξ di una coppia di poli complessi coniugati $-\xi\omega_n \pm j\sqrt{1-\xi^2\omega_n}$ e un polo reale in $-1/\tau$:

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{1/k_{TP}}{1+\dfrac{s^2(1+sT_M)}{k_m K_P k_{TP}(1+sT_P)}} \quad \rightarrow \quad W(s) = \frac{\dfrac{1}{k_{TP}}(1+sT_P)}{\left(1+\dfrac{2\xi}{\omega_n}s+\dfrac{1}{\omega_n^2}s^2\right)(1+s\tau)}$$

Tuttavia, nell'ottico di un controllo indipendenti ai giunti, sarà necessario studiare anche la f.d.t. tra disturbo e uscita:

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\dfrac{sR_a}{k_t K_P k_{TP}(1+sT_P)}}{1+\dfrac{s^2(1+sT_m)}{k_m K_P k_{TP}(1+sT_P)}}$$

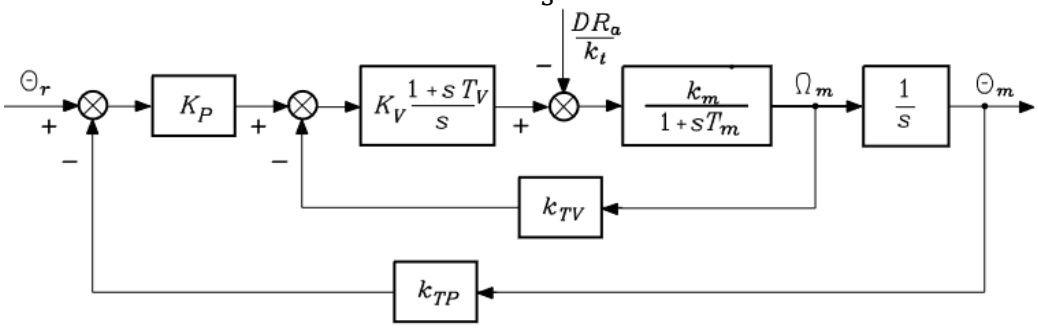
Analizzando queste due funzioni di trasferimento possiamo definire due importanti grandezze utili per valutare le prestazioni del controllore:

- **Fattore di Reiezione al Disturbo X_R**: è il fattore moltiplicativo dello zero presente nel regolatore PI, il cui scopo è annullare gli effetti della gravità sulla posizione angolare quando $\Theta_m = cost$ ed è definito come: $X_R = K_P k T_P$

  Tale fattore è caratterizzato da K_P, per cui più aumentiamo quest'ultimo e maggiore sarà X_R con conseguente migliore reiezione ai disturbo. Tuttavia, non è consigliabile aumentare troppo K_P in quanto, per piccoli rapporti di smorzamento, si avrebbero delle oscillazioni inaccettabili sull'uscita.
- **Tempo di Recupero del Uscita T_R**: come dice il nome è il tempo necessario al sistema di controllo per recuperare l'uscita affetta dal disturbo sulla posizione angolare ed è definita come: $T_R = \max\{T_P, 1/\xi\omega_n\}$

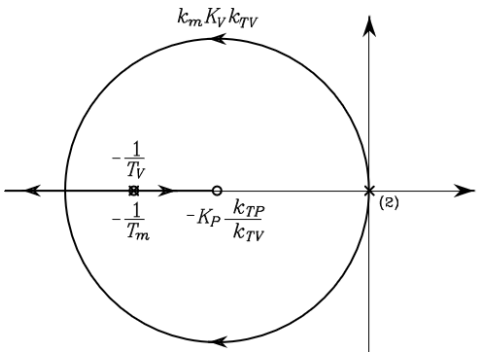
- **Retroazione di Posizione e Velocità**: essa si realizza mediante il seguente schema. In questo caso abbiamo:

$$C_P(s) = K_P \qquad C_V(s) = K_V \frac{1 + sT_V}{s} \qquad C_A(s) = 1 \qquad k_{TA} = 0$$



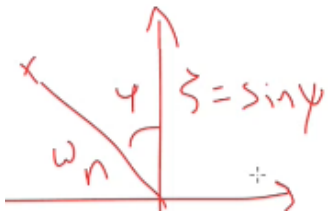
In questo caso abbiamo il problema che non possiamo applicare il luogo della radici in quanto sono presenti due anelli di retroazione. Allora per poterlo impiegare dobbiamo spostare i rami verticali dopo K_P e prima dell'integratore rispettivamente a monte del primo e a valle di quest'ultimo. Dopodichè dobbiamo fare il parallelo delle due azioni in retroazione. In tal modo, possiamo definire le f.d.t. del ramo di azione diretta e del ramo di retroazione:

$$C(s) = \frac{k_m K_P K_V (1 + sT_v)}{s^2(1 + sT_m)} \qquad H(s) = k_{TP}\left(1 + s\frac{k_{TV}}{K_P k_{TP}}\right)$$



Contrariamente al caso precedente, non dobbiamo solo progettare K_p e T_v ma anche K_v, per cui abbiamo guadagnato un grado di libertà. Osservando il luogo delle radici abbiamo un polo meccanico $-1/T_m$ e il doppio polo nell'origine (uno è di struttura e l'altro e del controllore). Ci conviene cancellare il polo meccanico con lo zero $-1/T_V$ ponendo $T_m = T_v$. Dopodichè avremo un zero aggiuntivo $-K_P k_{TP}/k_{TV}$ che risulta variabile perché varia K_P, infatti riducendo quest'ultimo lo zero si sposta sempre più a destra. Questo inciderà ovviamente sul luogo delle radici poiché se tale zero si sposta verso destra la circonferenza si riduce e viceversa se lo zero va verso sinistra. Per cui dobbiamo scegliere su quale curva ci vogliamo trovare, dopodichè progetto K_V per determinare la posizione dei poli complessi coniugati in retroazione. Per capire ciò andiamo a calcolare la fdt ingresso-uscisa:

$$W(s) = \frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\frac{1}{kT_P}}{1 + \frac{k_{TV}}{K_P k_{TP}} s + \frac{s^2}{k_m K_P k_{TP} k_V}} = \frac{\frac{1}{kT_P}}{1 + \frac{2\xi}{\omega_n} s + \frac{s^2}{\omega_n^2}}$$



Allora se noi assegniamo da progetto ξ e ω_n, grazie alle seguenti formule possiamo calcolare K_V e K_P stabilendo dove dovranno trovarsi i poli complessi coniugati del sistema ad anello chiuso.

In definitiva, più che adottare il luogo della radici abbiamo effettuato una vera e propria sintesi diretta del controllore scegliendo:

$$\boldsymbol{K_V = \frac{2\xi\omega_n}{k_m k_{TV}}} \qquad \boldsymbol{K_P = \frac{\omega_n^2}{k_m k_{TP} K_V}}$$

Analizzando la f.d.t. tra disturbo e uscita:

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\dfrac{sR_a}{k_t K_P k_{TP} K_V (1+sT_P)}}{1+\dfrac{k_{TV}}{K_P k_{TP}}s+\dfrac{s^2}{k_m K_P k_{TP} k_V}}$$

Possiamo nuovamente definire:

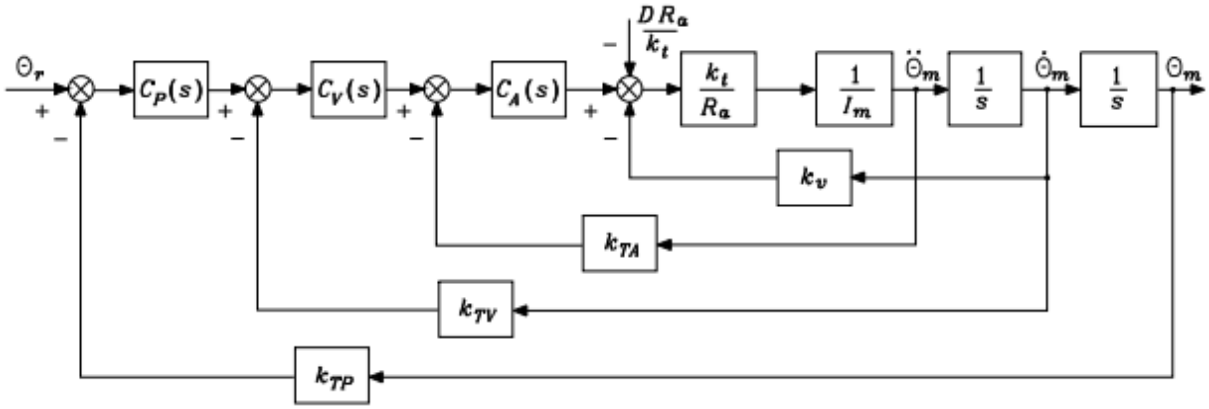
- **Fattore di Reiezione al Disturbo X_R**: $X_R = K_P k_{TP} K_V$
- **Tempo di Recupero del Uscita T_R**: $T_R = \max\{T_m, 1/\xi\omega_n\}$

Rispetto al caso precedente abbiamo sempre la presenza di uno zero nell'origine però ora il disturbo non è ridotto del fattore $K_P k_{TP}$ ma bensì di $K_p k_{TP} K_V$. I poli complessi coniugati hanno una costante di tempo di $1/\xi\omega_n$, mentre un altro polo ha una costante di tempo $T_m = T_V$. Nell'approccio precedente ci siamo trovati nella condizione svantaggiosa per cui avevamo scelto $T_P > T_m$, mentre stavolta il T_R non dipende da T_P. In definitiva, la retroazione appena studiata è da preferire per tre motivi:

- ✓ Ci consente di effettuare una sintesi diretta;
- ✓ Riduce il disturbo in transitorio;
- ✓ Riduce il tempo di recupero T_R.

- **Retroazione di Posizione, Velocità e Accellerazione**: : essa si realizza mediante il seguente schema. In questo caso abbiamo:

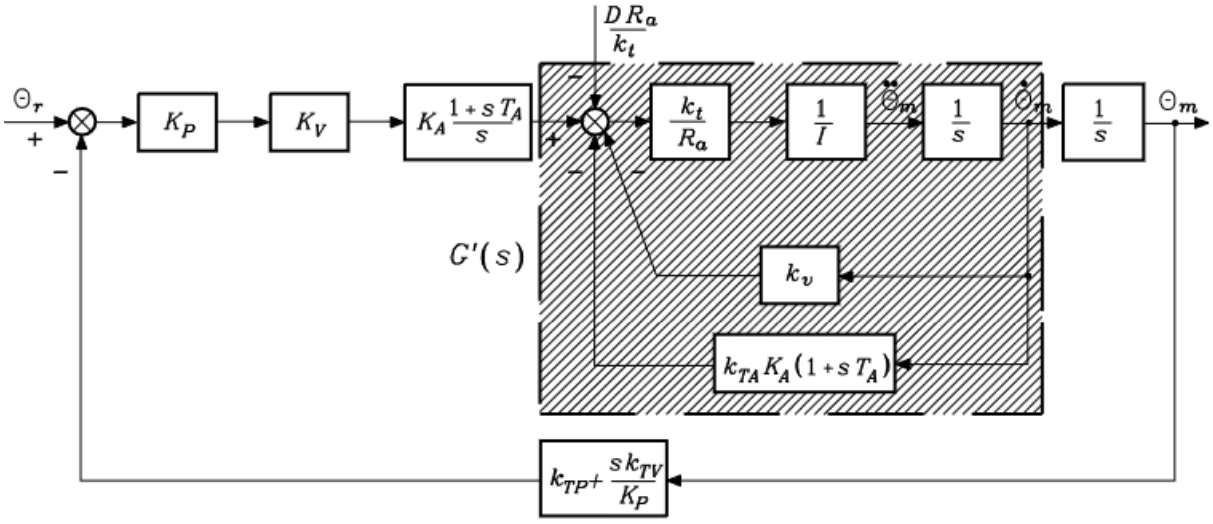
$$C_P(s) = K_P \qquad C_V(s) = K_V \qquad C_A(s) = K_A\frac{1+sT_A}{s}$$



Eseguendo degli spostamenti analoghi a quelli visti per il caso precedente, possiamo definire una funzione di anello $G'(s)$:

$$G'(s) = \frac{k_m}{(1+k_m K_A k_{TA})\left(1+\dfrac{sT_m\left(1+k_m K_A k_{TA}\dfrac{T_A}{T_m}\right)}{(1+k_m K_A k_{TA})}\right)}$$

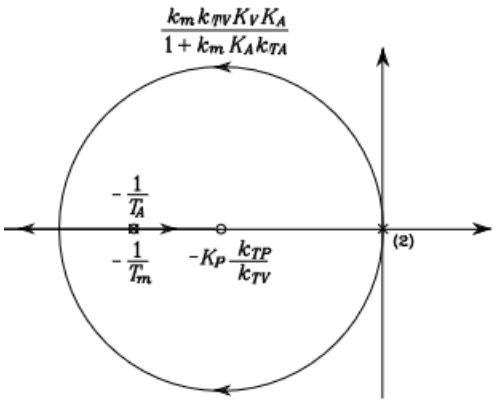
E ottenere così il seguente schema:



Le f.d.t. del ramo di azione diretta e del ramo di retroazione sono:

$$C(s)G(s) = \frac{K_P K_V K_A (1+sT_A)}{s^2} G'(s) \qquad H(s) = k_{TP}\left(1 + \frac{k_{TV}}{K_P k_{TP}} s\right)$$

Ponendo $T_m = T_A$ otteniamo un luogo delle radici analogo al caso precedente, infatti abbiamo la cancellazione polo $-1/T_m$ -zero $-1/T_A$ e lo zero variabile. Questa volta però abbiamo un grado di libertà in più dovuto a K_A. Anche in questo scenario possiamo soddisfare dei requisiti circa ξ e ω_n .



$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{k_{TV}}{K_P k_{TP}} s + \frac{(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}}$$

In realtà, grazie al gradi di libertà aggiuntivo possiamo imporre un vincolo di progetto riguardo X_R.

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\frac{sR_a}{k_t K_P k_{TP} K_V K_A (1+sT_A)}}{1 + \frac{k_{TV}}{K_P k_{TP}} s + \frac{(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A} s^2}$$

Possiamo nuovamente definire:

- **Fattore di Reiezione al Disturbo X_R**: $X_R = K_P k_{TP} K_V K_A$
- **Tempo di Recupero del Uscita T_R**: $T_R = \max\{T_A, 1/\xi\omega_n\}$

La seconda qunatità deve essere positiva altrimenti non avremmo più una retroazione negativa.

$$\frac{2K_P k_{TP}}{k_{TV}} = \frac{\omega_n}{\zeta}$$

$$k_m K_A k_{TA} = \frac{k_m X_R}{\omega_n^2} - 1$$

$$K_P k_{TP} K_V K_A = X_R$$

In realtà, un'altra possibile soluzione è scegliere $T_A < T_m$ per cui valga la seguente condizione:

$$k_m K_A k_{TA} \gg 1 \quad \rightarrow \quad k_m K_A k_{TA} T_A \gg T_m$$

In questo modo nella $G'(s)$ gli uno nella parentesi tonda sono trascurabili e comunque possiamo eseguire la cancellazione polo-zero vista in precedenza. Otteniamo così un T_A minore della precedente soluzione $T_A = T_m$.

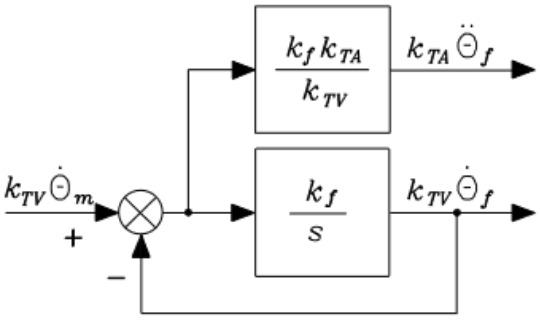
A questo punto si pone il problema di come retroazionare l'accellerazione e per farlo sarà necessario un **Filtro Ricostruttore**. Supponendo di avere sia un trasduttore di posizione ma soprattuto un trasduttore di velocità, grazie a quest'ultimo siamo in grado di calcolare $k_{TV}\dot{\Theta}_m$. Grazie a questa informazione possiamo impiegare lo schema di fianco per realizzare il nostro filtro ricostruttore. L'ingresso e l'uscita del filtro risultano essere esattamente la stessa velocità:

$$k_{TV}\dot{\Theta}_m = k_{TV}\dot{\Theta}_f$$

Tuttavia, a valle del nodo sommatore abbiamo l'accellerazione:

$$k_{TV}\dot{\Theta}_f \frac{s}{k_f} \qquad dove: \quad \dot{\Theta}_f s = \ddot{\Theta}_f$$

Dando tale accellerazione in ingresso al blocco superiore otteniamo la velocità filtrata che ci interessa: $k_{TA}\ddot{\Theta}_f$



Tutto il discorso tuttavia, funziona solo se k_f è elevato anche se questo non è un problema perché un sistema meccanico come un manipolatore sono dell'ordine delle decine di Hert e quindi questo filtro certamente non inficia in alcun modo la risposta del sistema.

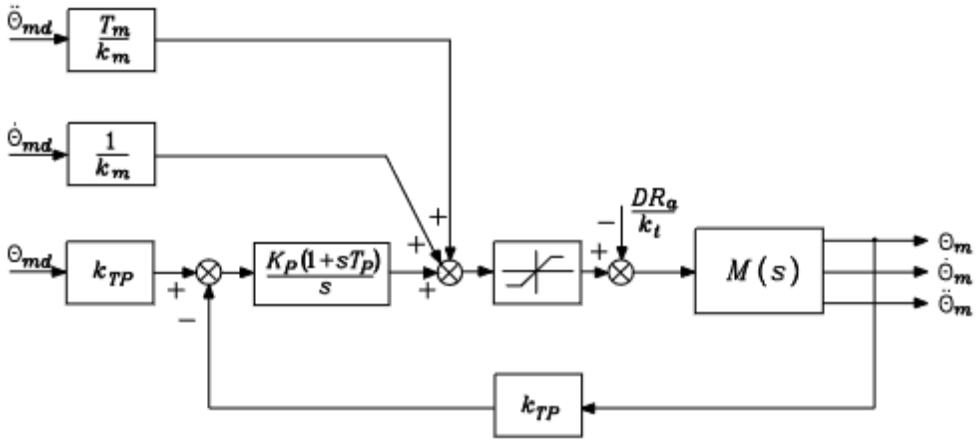
7.3.3 Compensazione in Avanti Decentralizzata

La **Compensazione in Avanti Decentralizzata** *è una tecnica che consiste nell'impiego di una azione di feedforward per inseguire una traiettoria specificata da un andamento desiderato della variabile motore in termini di posizione,velocità e accellerazione*. Tale tecnica viene impiegata quando abbiamo elevati valori di velocità e accellerazione che inevitabilmente portano ad un degrado dell'inseguimento del riferimento. Questa tecnica di compensazione presenta le seguenti tre varianti ciascuna delle quali relative al particolare schema adottato nel controllo indipendenti ai giunti:

- **Retroazione di Posizione**: la funzione di trasferimento ingresso/uscita a ciclo chiuso può essere modificata come segue:

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\dfrac{1}{k_{TP}}}{1 + \dfrac{s^2(1+sT_m)}{k_m K_P k_{TP}(1+sT_P)}} \quad \rightarrow \quad \Theta'_r(s) = \left(k_{TP} + \frac{s^2(1+sT_m)}{k_m K_P(1+sT_P)}\right)\Theta_{md}(s)$$

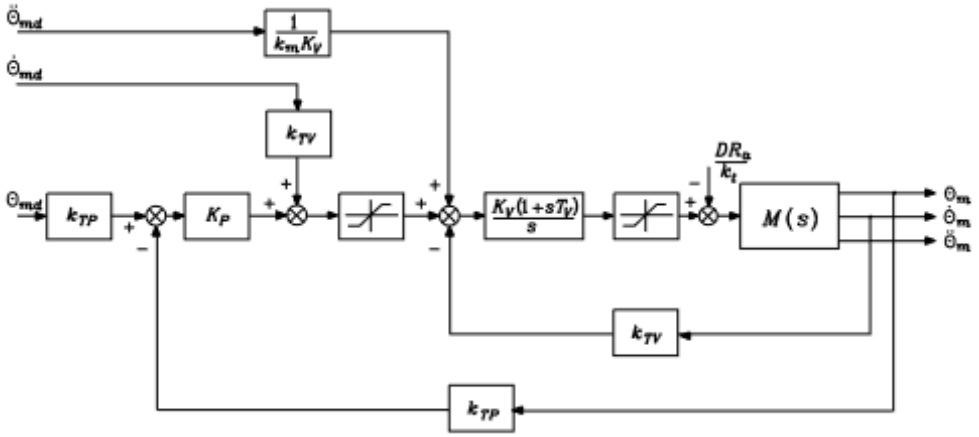
Lo schema che implementa questa soluzione è il seguente:



- **Retroazione di Posizione e Velocità**: la funzione di trasferimento ingresso/uscita a ciclo chiuso può essere modificata come segue:

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\dfrac{1}{k_{TP}}}{1 + \dfrac{sk_{TV}}{K_P k_{TP}} + \dfrac{s^2}{k_m K_P k_{TP} K_V}} \quad \rightarrow \quad \Theta'_r(s) = \left(k_{TP} + \frac{sk_{TV}}{K_P} + \frac{s^2}{k_m K_P K_V}\right)\Theta_{md}(s)$$

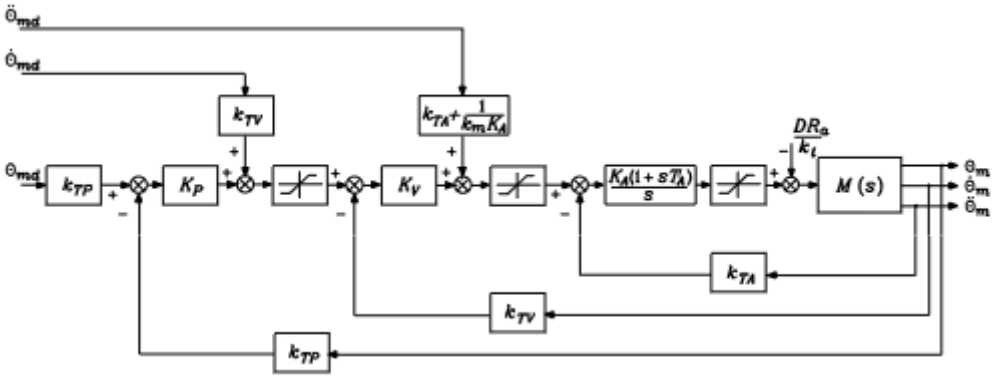
Lo schema che implementa questa soluzione è il seguente:



- **Retroazione di Posizione, Velocità e Accellerazione**: la funzione di trasferimento ingresso/uscita a ciclo chiuso può essere modificata come segue:

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2(1 + k_m K_A k_{TV})}{k_m K_P k_{TP} K_V K_A}} \rightarrow \Theta'_r(s) = \left(k_{TP} + \frac{sk_{TV}}{K_P} + \frac{s^2(1 + k_m K_A k_{TV})}{k_m K_P K_V K_A}\right)\Theta_{md}(s)$$

Lo schema che implementa questa soluzione è il seguente:



Queste tre soluzioni presentano degli aspetti in comune:

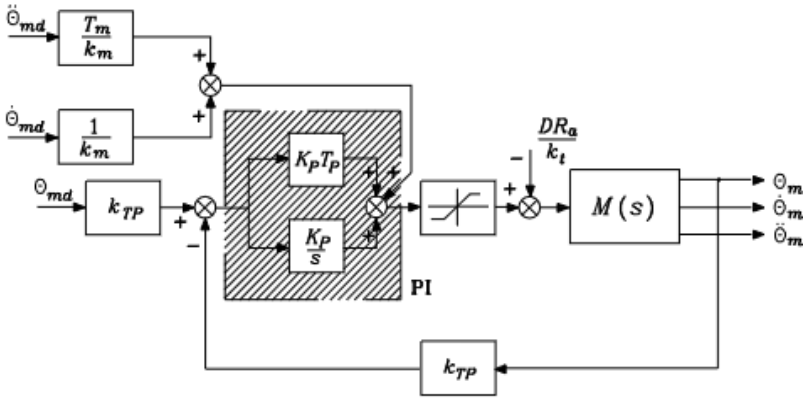
- **Prima Analogia**: le $\Theta'_r(s)$ sono in pratica $\Theta_{md}(s)$ moltiplicata per la funzione di trasferimento inversa tra $\Theta_{md}(s)/\Theta_r(s)$. Per esempio, considerando la retroazione di posizione e velocità in cui $\Theta_m(s) = \Theta_{md}(s)$:

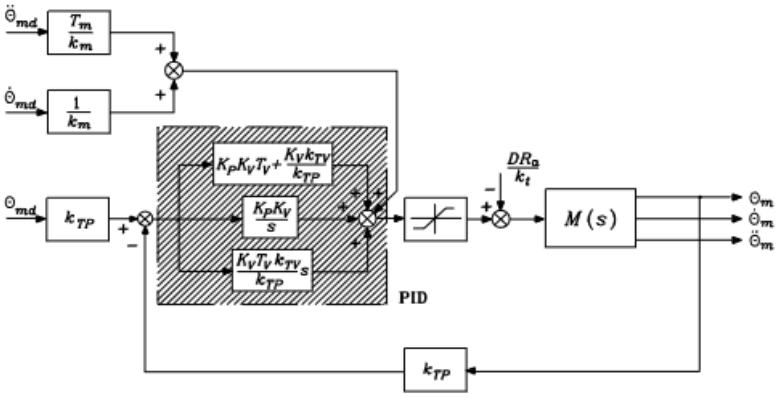
$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2}{k_m K_P k_{TP} K_V}} = W_d(s) \quad \rightarrow \quad \Theta'_r(s) = \frac{1}{W_d(s)}\Theta_{md}(s)$$

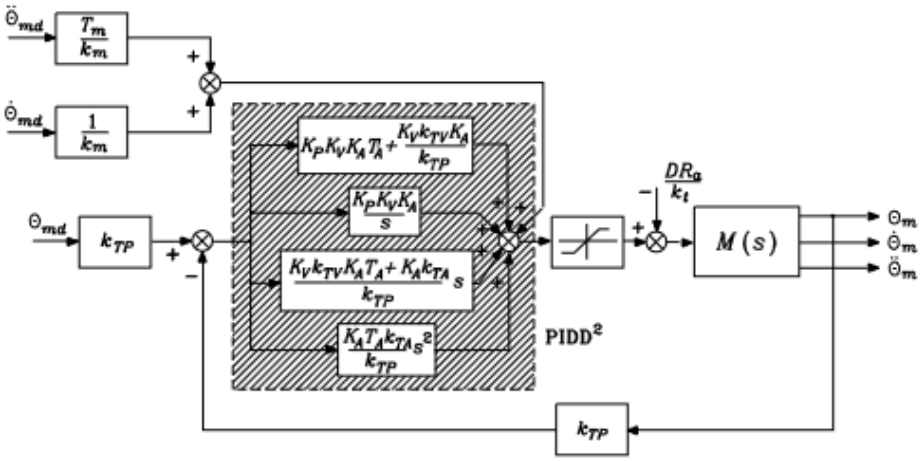
  Osservando $W_d(s)$ abbiamo un eccesso poli/zeri pari a due (cioè 2 poli e nessun zero). Questo vuol dire che $1/W_d(s)$ avrà un eccesso zeri/poli pari a due e nella realtà non è possibile realizzare un sistema che ha più zeri che poli. Tuttavia, non dobbiamo dimenticare che noi abbiamo a disposizione il set completo di posizione, velocità e accellerazioni desiderate per cui all'atto pratica non stiamo dando $\Theta_{md}(s)$ in ingresso ad un blocco non realizzabile che ci restituisce $\Theta'_r(s)$;
- **Seconda Analogia**: sulla scia della prima analogia capiamo che tutte e tre le soluzioni richiedono una inversione cinematica completa, ovvero è necessario impiegare un algoritmo CLIK del secondo ordine proprio per avere posizione, velocità e accellerazione;
- **Terza Analogia**: in tutti e tre gli scenari è richiesta una certa conoscenza del sistema meccanico ma che si riduce via via che aumentiamo il numero di anelli di retroazione. Nel primo caso dobbiamo conoscere T_m e k_m, nel secondo caso solo k_m e nell'ultimo sempre k_m ma con un peso minore;
- **Quarta Analogia**: tutti gli schemi adottano dei blocchi saturazione, i quali non sono da intendersi come non linearità ma bensì come limitatori dei segnali di controllo (in pratica delle protezioni). Nel primo schema abbiamo un limitatore di tensione, nel secondo si aggiunge anche un limitatore di velocità (seconda saturazione partendo da destra) e nel terzo schema si aggiunge un limititatore di accellerazione (terza saturazione partendo da destra).

Ultimo aspetto da osservare riguarda la possibile adozioni dei seguenti schemi equivalenti. In realtà, essi sono degli schemi equivalenti dal punto di vista matematico ma non lo sono dal punto di vista pratico per due motivi:

- Richiedono delle conoscenze differenti, tipicamente maggiori, circa il sistema. Per esempio il secondo prima richiedeva solo la conoscenza di k_m mentre in quello equivalente è richiesto in aggiunto T_m;
- E' preferibile avere gli anelli di retroazione fisici in quanto consentono l'adozione di blocchi di saturazione separati per tensione di controllo, velocità e accellerazione. Invece i seguenti schemi equivalenti adottano una sola saturazione. Risultato di ciò, l'adozione di un unico blocco regolatore non mi consente di diagnosticare malfunzionamenti in tempi rapidi in quando sono rilevabili solo dall'aumento eccessivo del segnale di controllo. Invece, l'adozione di saturazioni separate mi consente di capire che qualcosa non funziona per esempio già osservando dei sovragiri del motore (saturazione sull'accellerazione o velocità).





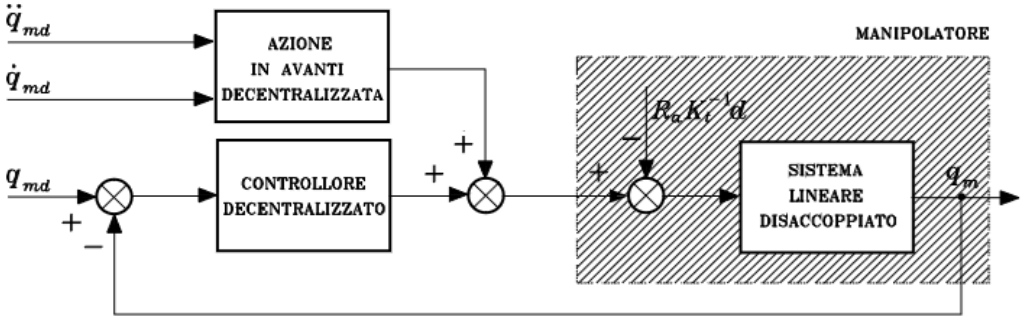


7.3.4 Compensazione in Avanti a Coppia Precalcolata

In precedenza abbiamo eseguito i seguenti studi:

1. Modellato il manipolatore come un sistema lineare disaccoppiato soggetto ad un disturbo d;
2. Realizzato un controllo decentralizzato che si occupare di fare in modo che la singolare variabile di giunto Θ_m segua il riferimento Θ_{md}. Complessivamente dovremo ragionare sui vettori q_m e q_{md};
3. Abbiamo migliorato l'inseguimento della traiettoria nel caso in cui si abbiano elevati valori di velocitò e accelerazione.

A questo punto però, sempre a fronte di velocità e accelerazione levati, rimane comunque il problema del disturbo d che rende complessivamente l'intero controllo piuttosto oneroso.



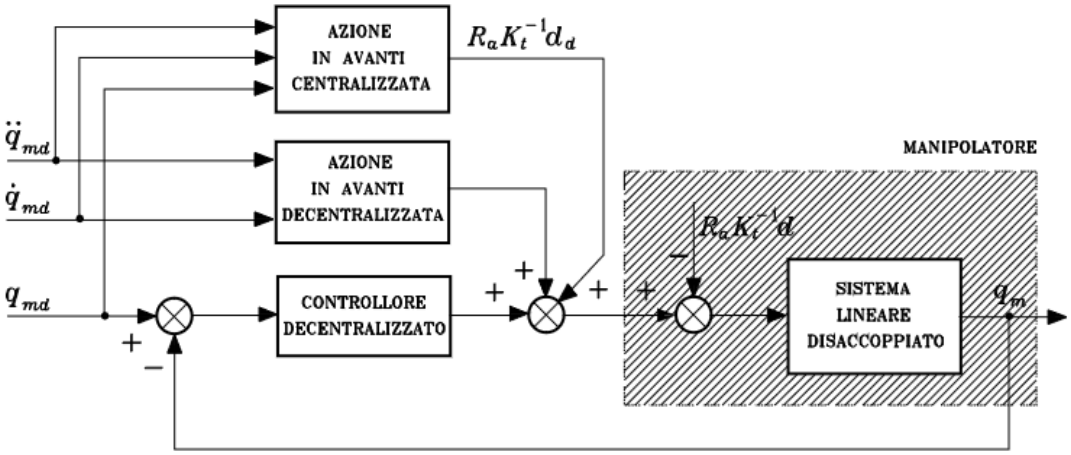
La **Compensazione in Avanti a Coppia Precalcolata** realizza un'azione di feedforward di tipo centralizzata che introduce nel sistema un segnale **Disturbo Desiderato d_d** il cui scopo è *compensare* il disturbo d. Dalla conoscenza di:

- Modello dinamico del manipolatore;
- Traiettoria desiderata in termini di posizione q_{md}, velocità $\dot{q}_{md}$ e accelerazione $\ddot{q}_{md}$.

E' possibile calcolare il disturbo desiderato d_d mediante la seguente relazione:

$$\boldsymbol{d_d = K_r^{-1}\Delta B(q_d)K_r^{-1}\ddot{q}_{md} + K_r^{-1}C(q_d,\dot{q}_d)K_r^{-1}\dot{q}_{md} + K_r^{-1}g(q_d)}$$

E' stato messo in evidenza il termine compensare in quanto non si va a cancellare totalmente il disturbo d in quanto il suo andamento non è noto. Tuttavia, l'azione di compensazione fa si che avremo in disturbo residuo $\tilde{d} = d_d - d$ che ovviamente sarà inferiore a d, rendendo così più leggero il controllo. Ciò dal punto di vista pratica si traduce nell'impiego di guadagni nel controllo più bassi e quindi minor rischio di saturazione che comporterebbe aumento dell'errore e degrado delle prestazioni.



7.4 Controllo Centralizzato

7.4.1 Definizione di Controllo Centralizzato

Il **Controllo Centralizzato** *consiste nel controllare la struttura robotica vedendola come un sistema multivariabile non lineare a n ingressi, ovvero le coppie ai giunti τ, e ad n uscite, ovvero le posizioni dei giunti q, le quali interasciscono mediante appunto relazioni non lineari* (come ci suggerisce il modello dinamico). L'approccio del controllo centralizzato, come vedremo di seguito, si basa ampiamente su una conoscenza parziale oppure completa del modello dinamico del manipolatore. La motivazione di ciò risiede nello sviluppo di algoritmi di controllo che siano in grado di compensare i termini non lineari del modello e conferendo così elevate prestazioni del manipolatore. Il controllo centralizzato per sua natura porta a risultati che non sarebbero ottenibili neanche con il miglior algoritmo decentralizzato, in quanto considerare gli effetti di interazione e accoppiamento tra i giunti come un disturbo in un sistema con alte velocità o magari azionamenti diretti (contatto diretto motore/giunto per cui $K_r = I$) porta ad elevati errori di tracking.

Data questa panoramica circa il controllo centralizzato, ricordiamo che il problema del controllo del moto nello spazio dei giunti può avere due sfaccettature:

A. Problema di Tracking: il problema di controllo si esplica come inseguimento di un ingresso di riferimento: $q(t) = q_d(t)$

B. Problema di Regolazione: il problema di controllo si esplica come mantenenimento dell'uscita al valore desiderato: $q(t) = q_d = cost$

Allora quello che faremo in seguito è, dato il modello dinamico del nostro manipolatore:

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

Definire i diversi algoritmi di controllo centralizzato il cui scopo sarà ricavare le coppie (ai giunti) di controllo u (sarebbe τ) tali da risolvere il problema di tracking oppure di regolazioni. Gli algoritmi di controllo centralizzato che analizzeremo sono:

A. Controllo a Dinamica Inversa;
B. Controllo PD con Compensazione di Gravità;
C. Controllo Robusto;
D. Controllo Adattativo.

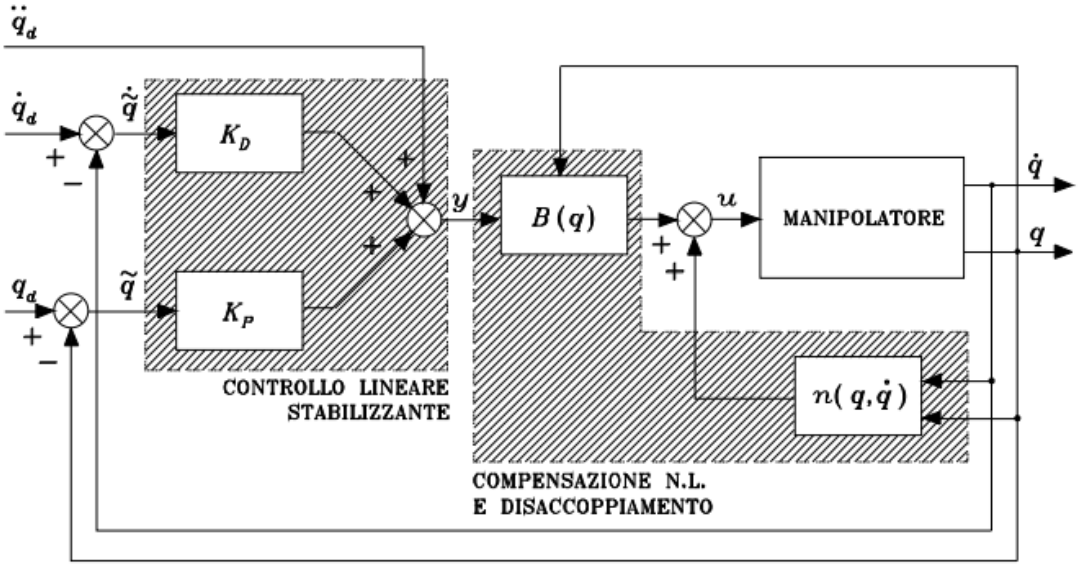
7.4.2 Controllo a Dinamica Inversa

Il **Controllo a Dinamica Inversa** si pone l'obiettivo di ricavare una legge di controllo u, funzione dello stato del sistema, tale da realizzare una relazione ingresso/uscita di tipo lineare con il fine di risolvere il *problema di tracking*. La legge di controllo che realizza il controllo a dinamica inversa è la seguente:

$$\boldsymbol{u = B(q)y + n(q,\dot{q}) = B(q)y + [C(q,\dot{q})\dot{q} + F\dot{q} + g(q)]}$$

In cui: $\boldsymbol{y = \ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q}}$ *dove*: $\boldsymbol{errore\ di\ posizione = \tilde{q} = q_d - q}$

Lo schema a blocchi che implementa il controllo a dinamica inversa è il seguente:



Anzitutto riprendiamo il modello dinamico del nostro manipolatore:

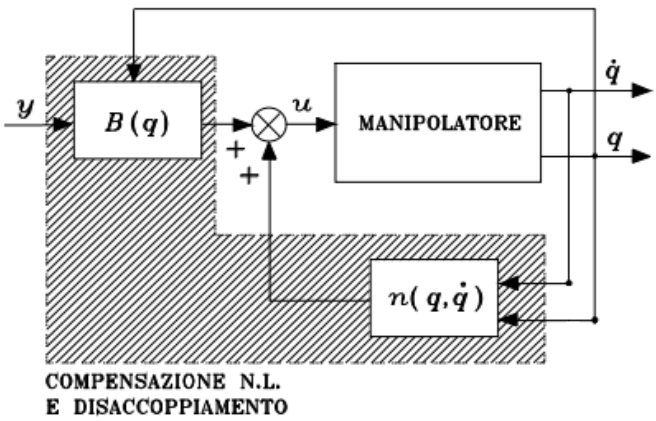
$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) = B(q)\ddot{q} + n(q,\dot{q}) = u$$

Il controllo a dinamica inversa, come accennato dalla definizione, vuole realizzare una retroazione di stato non lineare che effettui una *linearizzazione esatta* della dinamica del sistema. Ricordiamo che nello schema CLIK del primo ordine andavamo a compensare il termine non lineare $J_A(q)$, quindi realizzavamo un feedback linearization, scegliendo come soluzione alla cinematica inversa: $\dot{q} = J_A^+(q)(\dot{x}_d + Ke)$

In tal modo, il sistema errore $\dot{e} + Ke = 0$ era globalmente asintoticamente stabile. Allo stesso modo, nel CLIK del secondo ordine andavamo a compensare il termine non lineare $\dot{J}_A\dot{q}$ scegliendo come soluzione: $\ddot{q} = J_A^+(q)\left(\ddot{x}_d + K_D\dot{e} + K_P e - \dot{J}_A\dot{q}\right)$

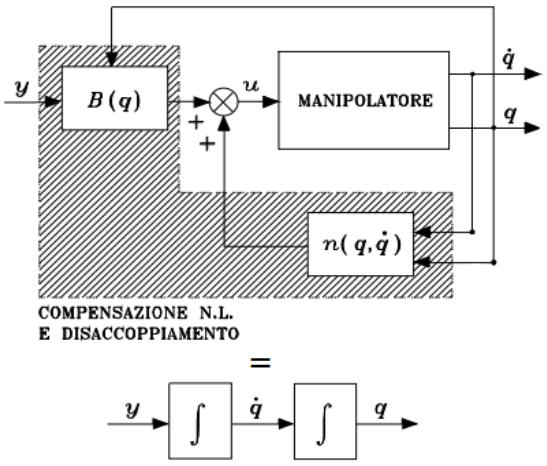
In tal modo il sistema errore $\ddot{e} = \ddot{x}_d + K_D\dot{e} + K_P e = 0$ era globalmente asintoticamente stabile. Quanto detto ci fa intuire che la strada da seguire per realizzare la feeback linearization per il controllo a dinamica inversa è sfruttare la matrice a fattore del termine non lineare. E' da osservare che nel caso del modello dinamico la matrice in questione è proprio la matrice di inerzia B, la quale grazie alle sue proprietà di essere *quadrata e definita positiva*, dunque sempre invertibile. Tutto ciò ci assicura la possibilità di trovare un regolatore che effettui la feeback linearization. Contrariamente invece allo Jacobiano che può essere rettangolare (a causa della ridondanza) e non di rango pieno (a causa delle singolarità).

Poiché il nostro scopo è trovare una u tale da disaccoppiare e linearizzare il sistema non è necessaria alcuna inversione della matrice d'inerzia B, ma bensì possiamo scegliere la legge di controllo come: $\boldsymbol{u = B(q)y + n(q,\dot{q})}$
Realizzando così la compensazione non lineare e disaccoppiamente indicata dello schema di fianco. Il controllo a dinamica inversa come si può osservare dalla definizione della u deve il suo nome al fatto che la legge di controllo non lineare si basa sul calcolo della dinamica inversa del manipolatore.



Il vettore y rappresenta un nuovo vettore di input di cui dovremo determinare l'espressione. Tuttavia, sostituendo l'espressione della u nel modello dinamico:

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



Tale uguaglianza può essere verificata in due modi:

a) $B(q)\ddot{q} = B(q)y \;\rightarrow\; B(q)(\ddot{q} - y) = 0$
Ora $B(q)(\ddot{q} - y) = 0 \;\Leftrightarrow\; (\ddot{q} - y) \in \mathcal{N}(q)$
Tuttavia, essendo la matrice d'inerzia quadrata e definita positiva, allora $\mathcal{N}(B) = \emptyset$.
Dunque l'uguaglianza $B(q)\ddot{q} = B(q)y$ è verificata solo se :
$$\ddot{q} = y$$

b) $B(q)\ddot{q} = B(q)y \quad\rightarrow\quad \ddot{q} = B^{-1}By = Iy = y \quad\rightarrow\quad \ddot{q} = y$

In pratica stiamo dicendo che ciascuna componente y_i del vettore y influenza la corrispettiva variabile di giunto q_i di q con una relazione di doppio integratore e il tutto indipendentemente dal movimento degli altri giunti. A questo punto, possiamo scegliere y come azione PD con aggiunta di un feedforward di velocità e accellerazione:

$$\begin{aligned}\boldsymbol{y} &= -K_D\dot{q} - K_P q + r \\ &= -K_D\dot{q} - K_P q + [\ddot{q}_d + K_D\dot{q}_d + K_P q_d] \\ &= \boldsymbol{\ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q}}\end{aligned} \qquad dove: \begin{aligned}\boldsymbol{\tilde{q} &= q_d - q} \\ \boldsymbol{\dot{\tilde{q}} &= \dot{q}_d - \dot{q}}\end{aligned}$$

Grazie a questa scelta della y arriviamo alla condizione per cui il sistema errore, qualora presenti $K_D > 0$ e $K_P > 0$, risulta essere globalmente asintoticamente stabile in quanto fare tendere l'errore di posizione $\tilde{q} = q_d - q \rightarrow 0$: $\quad \ddot{\tilde{q}} + K_D\dot{\tilde{q}} + K_P\tilde{q} = 0$
Scegliendo K_P e K_D come: $\quad K_P = diag\{\omega_{n1}^2, \dots, \omega_{nn}^2\} \qquad K_D = diag\{2\xi_1\omega_{n1}, \dots, 2\xi_n\omega_{nn}\}$
E poiché abbiamo compensato i termini non lineari ottenendo un sistema lineare disaccoppiato, possiamo imporre la dinamica desiderata mediante la tecnica di assegnamento degli autovalori scegliendo opportunamento i ω_{ni} e ξ_i.

In alcuni casi potrebbe capitare che in realtà noi abbiamo solo delle stime (indicato con $\hat{}$) di C, F e g:

$$B(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} + \hat{F}\dot{q} + \hat{g}(q) = u$$

Ora questo scenario è problematico anzitutto qualora C e F non siano compensati, ma qualora lo fossero rimane comunque il problema circa il termine di gravità. A causa di ciò a regime, seppur $\dot{q}$ e $\ddot{q}$ tendono a zero, a causa del fatto che $g(q) \neq \hat{g}(q)$ si ha che l'errore $\hat{q}$ non tende a zero, portando alla presenza di un errore a regime non nullo. Per risolvere questo problema dobbiamo passare da un controllo PD con feedforward ad un controllo PID con feedforward scegliendo y come:

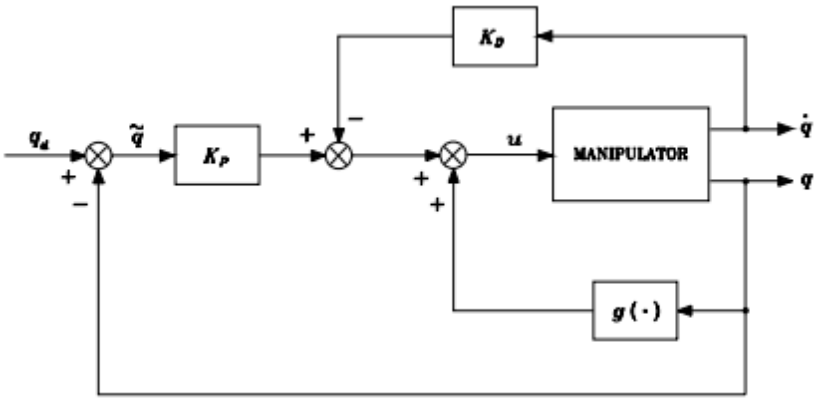
$$\boldsymbol{y = \ddot{q}_d + K_P\tilde{q} + K_I \int \tilde{q} d\tau + K_D\dot{\tilde{q}}}$$

Questo produrrà un aumento dell'ordine del sistema di controllo da ordine 2 a 3, per cui bisognerà scegliere opportunamento K_P, K_I, K_D. Infatti, bisogna osservare che per un sistema del terzo ordine avere queste tre matrici definite positiva è solo una *condizione necessaria ma non sufficiente* per l'asintotica stabilità.

7.4.3 Controllo PD+

Il **Controllo PD con Compensazione di Gravità** (o **PD+**) è la soluzione di controllo centralizzato che risolve il problema di regolazione scegliendo la coppia u necessaria a far convergere q a $q_d = cost$ per $t \to \infty$ (in pratica un tempo sufficientemente lungo):

$$\boldsymbol{u = K_P\tilde{q} - K_D\dot{q} + g(q)}$$



Possiamo progettare una legge di controllo lineare a meno della compensazione di gravità che è in grado di controllare un sistema non lineare e accoppiato non compensando i termini non lineari in quanto nel modello dinamico ci sono delle proprietà notevoli (precisamente quella di passività). Trovandoci nel campo non lineare dobbiamo ricorrere a Lyapunov, infatti il nostro scopo sarà trovare una funzione di Lyapunov funzione di $\tilde{q}$ per cui valgono queste condizioni:

- ✓ $V(\tilde{q}) > 0 \quad e \quad V(0) = 0$
- ✓ $\dot{V} < 0$

Se tali condizioni sono verificate la V è tale che $\tilde{q} \to 0$. E' bene ricordare che le condizioni del teorema di Lyapunov sono condizioni sufficienti ma non necessarie, per cui se troviamo una V che non rispetta la seconda condizione non vuol dire necessariamente che il sistema è instabile ma che forse non abbiamo trovato la funzione giusta. La V chiaramente verrà scelta come funzione quadratica tuttavia abbiamo due possibili strade:

A. Scelta Basata sulla Matematica: indicando lo stato del sistema meccanico come $\xi = [\tilde{q} \quad \dot{\tilde{q}}]^T$, scegliamo la funzione di Lyapunov come:

$$V(\tilde{q}) = \frac{1}{2}\xi^T Q \xi$$

A questo punto dobbiamo verificare che la derivata $\dot{V} < 0$, ciò comporterà con l'operazione di derivazione una presenza di $\dot{\xi}, \ddot{\tilde{q}}$ e $\dddot{\tilde{q}}$. In particolare, da quest'ultimo deriveremo la coppia di controllo u e quindi il problema si sposterà sul progettare quest'ultima tale che $\dot{V} < 0$. Proprio quest'ultimo problema non è di semplice risoluzione.

B. Scelta Basata sulla Fisica: sulla scia del precedente approccio scegliamo la matrice Q come matrice diagonale a blocchi, per cui la funzione V sarà data da un termini che hanno un preciso significato fisico:

$$V(\tilde{q}) = \frac{1}{2}\xi^T Q \xi = \frac{1}{2}[\tilde{q} \quad \dot{\tilde{q}}]^T \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \frac{1}{2}\tilde{q}^T Q_1 \tilde{q} + \frac{1}{2}\dot{\tilde{q}}^T Q_2 \dot{\tilde{q}}$$

Dove:

- **Primo Termine**: è il termine quadratico nell'errore di posizione $\tilde{q}$ e rappresenta una energia potenziale. In pratica essa una molla di rigidezza Q_1 al sistema che ha uno scostamento di $\tilde{q}$. Di fatto un azione di controllo del tipo $K_p e$ la posso interpretare fisicamente una azione elastico.
- **Secondo Termine**: è il termine quadratico nell'errore di velocità $\dot{\tilde{q}}$ e rappresenta una energia cinetica. Questo comperterà che Q_2 dimensionalmente sarà una matrice di inerzia costante. E' bene osservare che Q_2 è una matrice arbitraria che sceglie il progettista e non è direttamente uguale alla matrice di inerzia B del modello matematico, ma bensì siamo noi che scegliamo di porre questa uguaglianza.

Seguiamo dunque questa strada basata sulla fisica in cui $Q_1 = K_P$ e $Q_2 = B(q)$. Inoltre, ricordando che questo è un problema di regolazione per cui $q_d = cost$:

$$\tilde{q} = q_d - q = cost - q$$
$$\downarrow$$
$$\dot{\tilde{q}} = \dot{q}_d - \dot{q} = 0 - \dot{q}$$

Da cui otteniamo una funzione V che presentarà due termini quadratici entrambi definiti positivi e che si annullano solo per $\tilde{q} = 0$:

$$V = \frac{1}{2}\tilde{q}^T Q_1 \tilde{q} + \frac{1}{2}\dot{\tilde{q}}^T Q_2 \dot{\tilde{q}} = \frac{1}{2}\tilde{q}^T K_P \tilde{q} + \frac{1}{2}\dot{q}^T B(q)\dot{q}$$

Calcolando la derivata di V ed eseguendo alcuni passaggi algebrici vedremo la comparsa del termine $\dot{B}(q) - 2C(q,\dot{q})$ studiato nelle proprietà del modello dinamico. In particolare, indipendentemente da come fattorizzo la C, il termine in verde per il principio di conservazione dell'energia è sempre nullo:

$$\begin{aligned}
\dot{V} &= -\dot{q}^T K_P \tilde{q} + \dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} \\
&= -\dot{q}^T K_P \tilde{q} + \dot{q}^T [u - C\dot{q} - F\dot{q} - g] + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} \\
&= -\dot{q}^T K_P \tilde{q} + \dot{q}^T u - \dot{q}^T C\dot{q} - \dot{q}^T g + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} \\
&= \frac{1}{2}\dot{q}^T \left(\dot{B}(q) - 2C(q,\dot{q})\right)\dot{q} - \dot{q}^T F\dot{q} + \dot{q}^T(u - g(q) - K_P\tilde{q}) \\
&= -\dot{q}^T F\dot{q} + \dot{q}^T(u - g(q) - K_P\tilde{q})
\end{aligned}$$

A questo punto dobbiamo scegliere l'azione di controllo u tale da rendere negativa la $\dot{V}$. Osservando l'espressione finale di quest'ultima il primo termine non da alcun problema perché è gia negativo e fisicamente è l'attrito dovuto alla presenza di un riduttore meccanico il chè è una azione stabilizzante. Il secondo termine è problematico in quando è positivo per cui dobbiamo compensarlo scegliendo: $u = g(q) + K_P\tilde{q}$

In realtà, ci potrebbero essere dei casi in cui l'azione stabilizzante del primo termine non è sufficiente, per cui all'azione di controllo deve presentare anche un termine che rafforza questo termine utile: $\boldsymbol{u = g(q) + K_P\tilde{q} - K_D\dot{q}}$

Abbiamo così ottenuto la legge di controllo definita inizialmente e andandola a sostituire nalla $\dot{V}$ verifichiamo che essa sia definita negativa: $\dot{V} = -\dot{q}^T(F + K_D)\dot{q}$

Quando $\dot{q} = 0$ si che $\dot{V} = 0$, però l'aspetto su cui dobbiamo fare attenzione riguarda cosa accade a $\tilde{q}$ perché solo quando quest'ultimo va a zero possiamo dire di avere realizzato

correttamente il sistema di controllo. Ciò non è secondario perché attualmente non sappiamo se il sistema si è fermato a $q = q_d$ oppure ad un'altra q. Per affrontare questo problema partiamo dal modello del sistema controllato con il PD+:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) = g(q) + K_P\tilde{q} - K_D\dot{q}$$

Studiando il sistema da fermo si ha che $\dot{q} = \ddot{q} = 0$ e quindi, se ho anche compensato correttamente il termine di gravità ho:

$$\cancel{B(q)\ddot{q}} + \cancel{C(q,\dot{q})\dot{q}} + \cancel{F\dot{q}} + \cancel{g(q)} = \cancel{g(q)} + K_P\tilde{q} - \cancel{K_D\dot{q}} \quad \rightarrow \quad K_P\tilde{q} = 0 \quad \rightarrow \quad \tilde{q} = 0$$

Risultato di ciò: $\tilde{q} = q_d - q = 0 \quad \rightarrow \quad q = q_d = cost$

Allora possiamo affermare che con il controllo PD+ abbiamo ricavato un sistema globalmente asintoticamente stabile.

Ultimo aspetto da valutare riguarda la possibilità di non compensare la gravità puntualmente ma di inserire nella legge di controllo u solo il valore di gravità finale alla q_d:

$$u = g(q_d) + K_P\tilde{q} - K_D\dot{q}$$

Il problema che ci poniamo è se effettivamente il manipolatore a destinazione ha un termine di gravità $g(q_d)$. Affinchè ciò avvenga bisogna avere un K_P sufficientemente elevato e in particolare quest'ultimo dovrà essere scelto come matrice diagonale sulla quale sono presenti dei pesi tali da rispettare questa relazione: $K_P > \partial g/\partial q$

Dove $\partial g/\partial q$ ci da una stima del valore massimo di g e quindi di fatto stiamo studiando un worst case. Infine un'ultima scelta sarebbe quella di impiegare un azione integrale:

$$u = K_P\tilde{q} - K_D\dot{q} + K_I\int \tilde{q}d\tau$$

7.4.4 Controllo Robusto

Il **Controllo Robusto** *è un algoritmo di controllo centralizzato che si pone l'obiettivo di ricavare una legge di controllo del moto per il manipolatore negli scenari in cui si ha una incertezza sul modello dinamico e/o sono state adottate delle semplificazioni* (trascurare parte della matrice di inerzia, i termini di Coriolis, di forza centrifuga, ecc.). La legge di controllo u definita nel controllo robusto è la seguente ed è identica a quella vista per il controllo a dinamica inversa ma, invece di avere B e n, saranno presenti le rispettive stime $\hat{B}$ e $\hat{n}$:

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

In cui:

$$\boldsymbol{errore\ di\ posizione = \tilde{q} = q_d - q}$$

$$\boldsymbol{errore\ di\ velocit\grave{a} = \dot{\tilde{q}} = \dot{q}_d - \dot{q}}$$

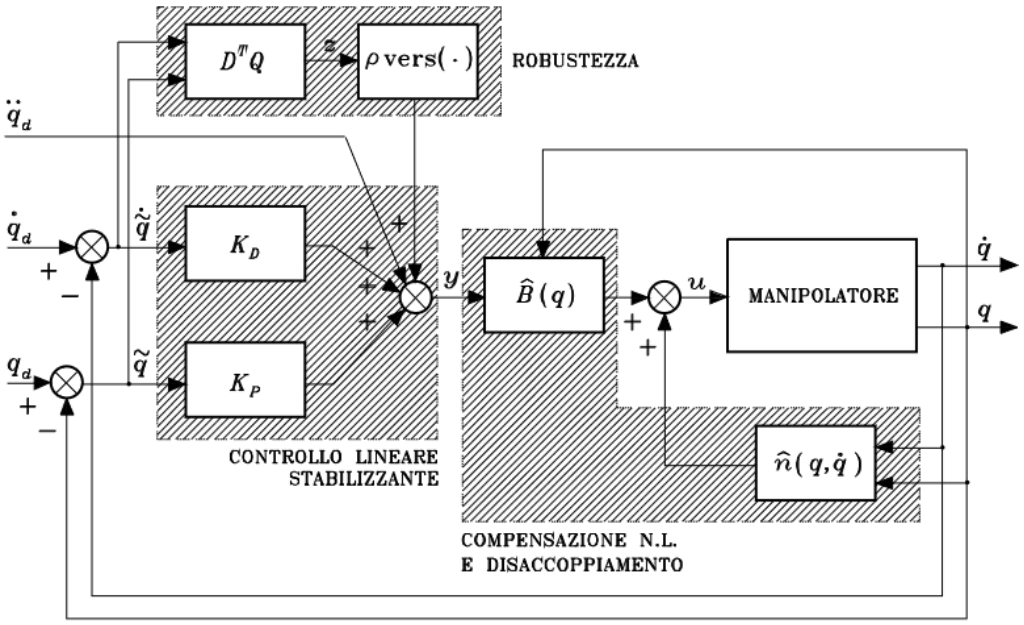
$$\boldsymbol{y = \ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q} + w} \qquad dove: \qquad \boldsymbol{z = D^T Q\xi = D^T Q\begin{bmatrix}\tilde{q}\\ \dot{\tilde{q}}\end{bmatrix}}$$

$$\boldsymbol{w = \begin{cases} \rho\dfrac{z}{\|z\|} & per\ \|z\| \geq \epsilon \\ \rho\dfrac{z}{\epsilon} & per\ \|z\| \geq \epsilon \end{cases}}$$

La legge di controllo che abbiamo definito avrà perciò tre fondamentali contributi:

- Termine che assicura la compensazione degli effetti non lineari e disaccoppiamento tra i giunti;
- Termine che introduce un'azione lineare in avanti che stabilizza il sistema dinamico di errore;
- Termine che rappresenta il contributo in robustezza che contrasta l'indeterminazione.

Lo schema a blocchi che realizza il controllo robusto è il seguente, il quale corrisponde esattamente a quello del controllo con dinamica inversa ma invece di avere B e n avremo le loro versione stimate e dunque sarà necessario un blocco aggiuntivo superiore che tiene conto della compensazione.



Cominciamo lo studio del controllo robusto andando a sostituire la sua legge di controllo nel modello dinamico reale: $B(q)\ddot{q} + n(q,\dot{q}) = u = \hat{B}(q)y + \hat{n}(q,\dot{q})$

Sfruttando il fatto che la matrice B è sempre invertivile ricaviamo la $\ddot{q}$. Eseguendo alcuni passaggi matematici in pratica otterremo in parte il risultato $\ddot{q} = y$ però in aggiunta avremo un termine di disturbo η:

$$\begin{aligned}\ddot{q} &= B^{-1}\hat{B}y + B^{-1}(\hat{n} - n)\\ &= B^{-1}\hat{B}y + B^{-1}(\hat{n} - n) + y - y\\ &= y + \left(B^{-1}\hat{B} - I\right)y + B^{-1}(\hat{n} - n)\\ &= y + \left(B^{-1}\hat{B} - I\right)y + B^{-1}\hat{\eta}\\ &= y - \eta\end{aligned}$$

A questo punto potremmo pensare di scegliere la y come l'abbiamo definita nel controllo a dinamica inversa vista : $y = \ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q}$

Tuttavia, contrariemente a quanto avviene nel controllo a dinamica inversa, l'errore non risulterà essere nullo ma bensì pari ad una quantità η: $\ddot{\tilde{q}} + K_D\dot{\tilde{q}} + K_P\tilde{q} = \eta$

Inoltre, anche scegliendo K_D e K_P definite positive non c'è alcuna garanzia che ciò rende il sistema asintoticamente stabile in quanto il sistema errore ha un *ingresso forzato*. Capiamo dunque che dobbiamo modificare la y inserendo una azione di controllo w che contrasti il disturbo η: $y = \ddot{q}_d + K_D\dot{\tilde{q}} + K_P\tilde{q} + w$

Allora per completare lo studio del controllo robusto ci serve definire il termine w. Per fare ciò Andiamo a studiare nel dettaglio il sistema errore, il quale presenterà come stato il vettore $\xi = [\tilde{q} \quad \dot{\tilde{q}}]^T$ costituito dall'errore di posizione e di velocità. Le equazioni di tale sistema sono indicate nel sistema di sinistra, mentre nel sistema di destra abbiamo l'equazione dell'errore nello spazio di stato:

$$\begin{cases}\dot{\tilde{q}} = \dot{\tilde{q}}\\ \ddot{\tilde{q}} = -K_P\tilde{q} - K_D\dot{\tilde{q}} + \eta - w\end{cases} \qquad \begin{aligned}\dot{\xi} &= \begin{bmatrix}0 & I\\ -K_P & -K_D\end{bmatrix}\xi + \begin{bmatrix}0\\ I\end{bmatrix}(\eta - w)\\ &= \tilde{H}\xi - D(\eta - w)\end{aligned}$$

Dove:

- $\tilde{H}\xi =$ è la parte lineare del sistema errore che è asintoticamente stabile e dove la matrice $\tilde{H}$ è definita dalla seguente espressione: $\tilde{H} = H - DK = \begin{bmatrix}0 & I\\ 0 & 0\end{bmatrix} - \begin{bmatrix}0\\ I\end{bmatrix}[K_P \quad K_D]$
- $D(\eta - w) =$ è la parte non lineare del sistema errore ed infatti la parte su cui dovremo agire in modo che il sistema complessivo sia globalmente asintoticamente stabile.

Adottiamo il metodo di Lyapunov scegliendo una funzione V dello stato ξ del sistema errore.

- ✓ **Condizione 1**: scegliamo la V come forma quadratica in modo che sia definita positiva e si annulli solo $\xi = 0$: $V(\xi) = \xi^T Q\xi > 0 \quad \forall\xi \neq 0 \qquad V(0) = 0 \qquad dove\colon\ Q > 0$
- ✓ **Condizione 2**: calcoliamo la derivata $\dot{V}$:

$$\begin{aligned}\dot{V} &= \dot{\xi}Q\xi + \xi^T Q\dot{\xi}\\ &= \xi^T\left(\tilde{H}^T Q + Q\tilde{H}\right)\xi + 2\xi^T QD(\eta - w)\\ &= -\xi^T P\xi + 2\xi^T QD(\eta - w)\\ &= -\xi^T P\xi + 2z^T(\eta - w)\\ &= -\xi^T P\xi + 2z^T\eta - 2z^T w\end{aligned}$$

↲ ***Equaz di Lyapunov***: $\tilde{H}^T Q + Q\tilde{H} = -P \quad \forall P > 0$

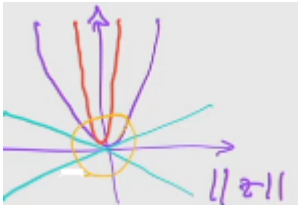
E' da notare che il termine $z^T = \xi^T QD$ andando ad analizzare i termini di cui è composto risulterà essere un vettore riga (perciò si metto trasposto):

$$z^T = [1 \times n] = [1 \times 2n][2n \times 2n][2n \times n]$$

Osservando l'espressione finale di $\dot{V}$ ci accorgiamo i termini negativi aiutano la stabilità contrariamente al termine positivo in cui compare il disturbo η. Allora poiché l'incognita è w, possiamo sceglierla in modo tale contrastare il termine di segno positivo. A tal proposito abbiamo diverse scelte:

A. $\boldsymbol{w = \rho z \ con \ \rho > \|\eta\|}$: quindi scegliamo w come il prodotto di un guadagno $\rho > \|\eta\|$ e $z = (z^T)^T = (\xi^T QD)^T = D^T Q\xi$. In tal modo otteniamo: $\dot{V} = -\xi^T P\xi + 2z^T\eta - 2\rho z^T z$
Valutiamo questa scelta andando a graficare l'andamento in funzione della $\|z\|$ del termine di segno negativo $2\rho z^T z$, il quale è quadratico rispetto a z, e del termine $2z^T\eta$, il quale è lineare rispetto a z di segno incerto:

a) Nel caso il termine $2z^T\eta$ è la retta che va verso il basso non ci sono problemi perché il termine quadratico prevale sempre su di esso. Assicurandoci così che $\dot{V}$ è sempre negativo;
b) Nel caso il termine $2z^T\eta$ è la retta che va verso l'alto, nell'intorno dell'origine, indipendentemente da quanto sia elevato il termine quadratico, avremo sempre un tratto lineare superiore ad esso. Risultato di ciò avremo dei valori di z per cui $\dot{V}$ non è negativa.

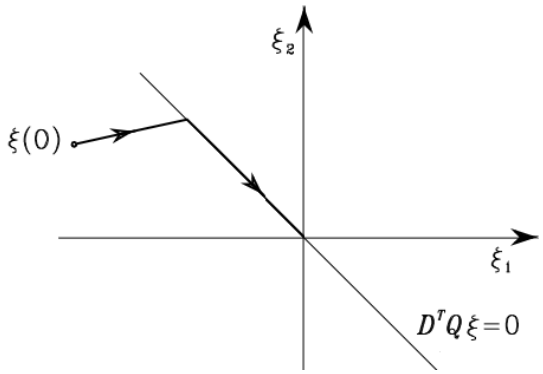


B. $\boldsymbol{w = \rho z/\|z\|}$: con questa modifica rispetto al caso precedente, scegliendo $\rho > \|\eta\|$, saremo sicuri che il termine di segno negativo sia sempre maggiore rispetto al termine di segno incerto. In definitiva, otterremo che la derivata di V risulterà essere definita negativa garantendo così la globale asintotica stabilità del sistema:

$$\dot{V} = -\xi^T P\xi + 2z^T\left(\eta - \rho\frac{z}{\|z\|}\right) < 0 \quad \forall\xi \neq 0$$

Allora dal punto di vista pratico sapendo che η di fatto misura la discrepanza nella compensazione dei termini del modello matematico, dovremo aumentare il valore di ρ fino a quando non siamo sicuri di stabilizzare il sistema. Questa scelta però presenta una diversa problematica a cui bisogna prestare attenzione. Per comprendere il problema in questione, andiamo a considerare il diagramma in figura dove sull'asse delle ascisse troviamo la norma dell'errore di posizione ξ_1 e sulle ordinate la norma dell'errore di velocità ξ_2. Definiamo inoltre il **Sottospazio di Scivolamento** $\boldsymbol{D^T Q\xi = 0}$, ovvero quella retta passante per l'origine tra 2° e 4° sulla quale devono evolversi le traiettorie del sistema errore. Allora supponiamo che all'istante iniziale l'errore $\xi(0) \neq 0$, grazie al controllo che abbiamo adottato la sua traiettoria è quello mostrato in figura in cui abbiamo:

1. Un primo tratto per cui l'errore finisce sul sottospazio di scivolamento;
2. A questo punto la z può assumere valori positivi o negativi, per cui il termine w per come è stato definito effettua delle compensazioni tali da far rimanere l'errore lungo il sottospazio di scivolamento fino a che esso non raggiunga l'origine dello spazio di stato.



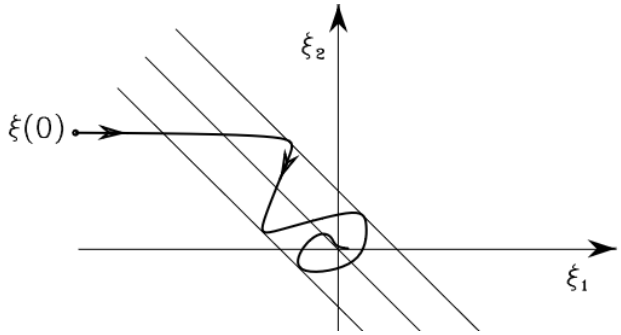
Questo scenario presenta il cosidetto **Chattering**, ovvero quel fenomeno di oscillazioni ad alta frequenza del segnale di controllo dovute proprio alle continue compensazioni di w.

C. w scelta a seconda di $\|z\|$: per mantenere il meglio della precedente scelta e in più eliminare il chattering dobbiamo *saturare l'azione di controllo* scegliendo w secondo il seguente criterio:

a. Se la norma di z è sopra una certa soglia ho uno legge di controllo a versore;
b. Se la norma di z è al di sotto di una certa soglia ho una legge di controllo lineare.

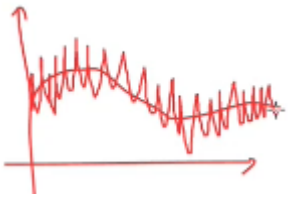
Più piccola è la soglia ε e più saranno alte le componenti di alta frequenza nel segnale di controllo.

$$\boldsymbol{w} = \begin{cases} \boldsymbol{\rho \frac{z}{\|z\|}} & \boldsymbol{per\ \|z\| \geq \epsilon} \\ \boldsymbol{\rho \frac{z}{\epsilon}} & \boldsymbol{per\ \|z\| \geq \epsilon} \end{cases}$$



Scegliendo la w in questo modo avremo che le traiettorie del sistema evolveranno in una regione a tubo attorno al sottospazio di scivolamento chiamato **Boundary Layer**. Tuttavia, questa scelta *non garantisce* la convergenza dell'errore a zero, ma assicura solo errori a norma limitata.

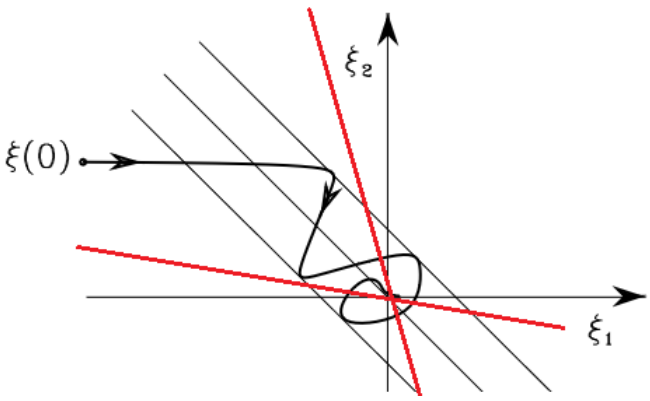
Questa azione di controllo porterà ad una oscillazione della coppia analoga a come mostrata in figura. E' da notare che il valor medio del segnale oscillatorio ricalca l'andamento della coppia desiderata. Nella pratica questa situazione non è in alcun modo problematica perché se il controllo va campionato con frequenze dell'ordine del kHz, questa frequenza viene tagliata dalla banda del sistema meccanico che al più potrà essere di qualche decina di Hz.



Ricordiamo che z è data dalla seguente relazione in cui ξ contiene sia l'errore di posizione che di velocità. Ciascuno di questi termini è moltiplicato per il rispettivo peso. Sapendo che dovremo tollerare un z che non è proprio zero ma bensì ε, ci chiediamo come conviene scegliere questi pesi (lo possiamo fare perché l'equazione di Lyapunov vale sia se scelgo Q e ricavo P o viceversa):

$$z = D^T Q \xi = [\cdot]\tilde{q} + [\cdot]\dot{\tilde{q}}$$

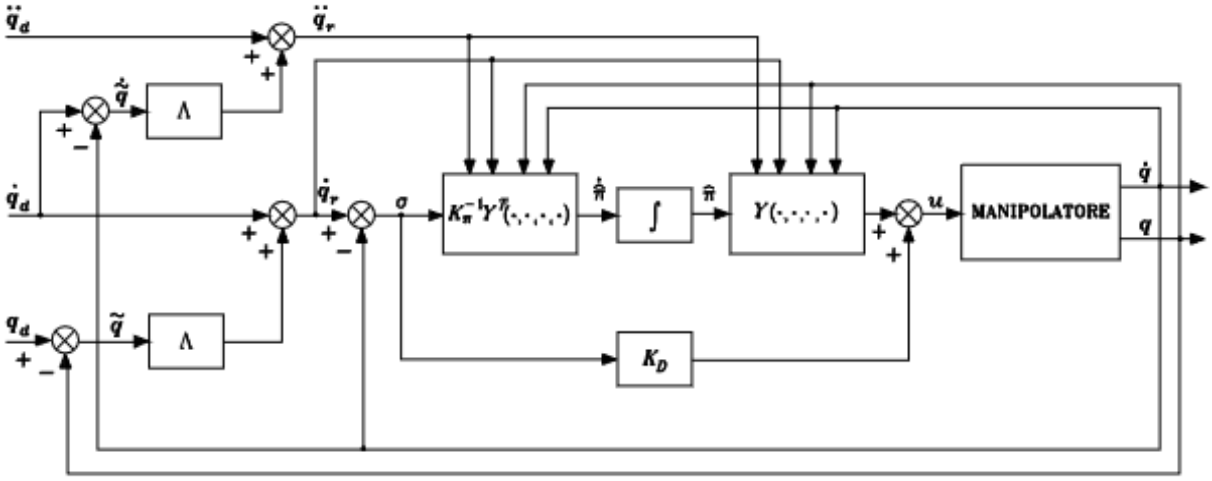
A. **Equità tra gli errori**: scelgo i pesi uguali ottenendo il sottospazio di scivolamento della precedente figura come bisettrice di secondo e quarto quadrante;
B. **Priorità all'errore di velocità**: scelgo un peso maggiore per $\dot{\tilde{q}}$ ottenendo così la retta a pendenza minore.
C. **Priorità all'errore di posizione**: scelgo un peso maggiore per ottenendo così la retta a pendenza maggiore. Nelle applicazioni di robotica in cui vogliamo garantire il corretto inseguito della traiettoria è chiaramente la scelta da adottare.



7.4.5 Controllo Adattativo

Il **Controllo Adattativo** *è un algoritmo di controllo centralizza che si pone l'obiettivo di trovare una legge di controllo del moto del manipolatore nel caso in cui esista una incertezza circa i parametri dinamici del modello (mal forniti dal produttore o presenza di un carico che li ha modificati). In particolare, tale legge di controllo andrà compensare tutti i parametri dinamici e aggiungerà una azione di adattamento online per contrastarne la variabilità*. La legge di controllo che implementa il controllo adattativo è: $\boldsymbol{u = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{\pi} + K_D\sigma}$

Lo schema a blocchi che implementa il controllo adattativo è il seguente:



Il controllo adattativo fonda le proprie basi sul proprietà di linearità $u = Y(q, \dot{q}, \ddot{q})\pi$ del modello dinamico rispetto ai parametri dinamici $\pi \in R^{p\times 1}$ per mezzo della matrice regressore $Y \in R^{n\times p}$: $\quad B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F\dot{q} + g(q) = u = Y(q, \dot{q}, \ddot{q})\pi$

Come descritto nella definizione di controllo adattativo la legge di controllo deve dapprima compensare tutti i parametri dinamici e ciò sarà la prima cosa di cui ci occuperemo, dopo fatto ciò ci occuperemo di aggiungere l'azione adattativa per contrastarne la variabilità. In primo luogo, adottiamo una legge di controllo mista tra coppia precalcolata e dinamica inversa:

$$\boldsymbol{u = B(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + F\dot{q}_r + g(q) + K_D\sigma}$$

Dove:

- $\dot{q}_r = \dot{q}_d + \Lambda\tilde{q}$ = contributo di velocità desiderata pesata, mediante una matrice $\Lambda > 0$, per l'errore di posizione $\tilde{q}$;
- $\ddot{q}_r = \ddot{q}_d + \Lambda\dot{\tilde{q}}$ = contributo di accelerazione desiderata pesata, mediante una matrice $\Lambda > 0$, per l'errore di velocità $\dot{\tilde{q}}$;
- $\sigma = \dot{q}_r - \dot{q} = \dot{\tilde{q}} + \Lambda\tilde{q}$ = questo termine moltiplicato per la matrice $K_D > 0$ definisce un azione PD sull'errore. La derivata di σ è: $\quad \dot{\sigma} = \ddot{q}_r - \ddot{q} = \ddot{\tilde{q}} + \Lambda\dot{\tilde{q}}$

Come si è potuto capire la matrice Λ, tipicamente scelta diagonale, permette di esprimere i termini di compensazione e disaccoppiamento non lineare in funzione della velocità e dell'accelerazione desiderate corrette però dallo stato corrente $(q, \dot{q})$ del manipolatore. Andiamo ora a sostituire la u del controllo adattativo nel modello matematico:

$$B\ddot{q} + C\dot{q} + F\dot{q} + \cancel{g} = B\ddot{q}_r + C\dot{q}_r + F\dot{q}_r + \cancel{g} + K_D\sigma$$
$$B(\ddot{q}_r - \ddot{q}) + C(\dot{q}_r - \dot{q}) + F(\dot{q}_r - \dot{q}) + K_D\sigma = 0$$
$$B\dot{\sigma} + C\sigma + F\sigma + K_D\sigma = 0$$

Poiché ci troviamo in ambito non lineare dobbiamo adottare la teoria di Lyapunov per dimostrare che la legge di controllo scelta rende il sistema globalmente asintoticamente stabile. Scegliamo la seguente funzione V come forma quadratica in modo che soddisfi già la prima condizione del teorema di Lypaunov. E' da osservare che, essa dovrà presentare anche il termine in $\tilde{q}$ in quanto per $\sigma = 0$ non siamo certi che l'errore di posizione $\tilde{q}$ vada a zero:

$$V(\sigma, \tilde{q}) = \frac{1}{2}\sigma^T B \sigma + \frac{1}{2}\tilde{q}^T M \tilde{q} \qquad con\ M > 0$$

Dobbiamo ora verificare la seconda condizione calcolando la $\dot{V}$. Osserviamo che ad un certo punto comparirà il termine $(\dot{B} - 2C)$, il quale sarà nullo se C è stata fattorizzata con i simboli Christofell. E' da notare che, vista la presenza di σ anziché di $\dot{q}$, non possiamo ricorrere al principio di Hamilton, tuttavia comunque abbiamo la cancellazione descritta.

$$\begin{aligned}
\dot{V} &= \sigma^T B \dot{\sigma} + \frac{1}{2}\sigma^T \dot{B} \sigma + \tilde{q}^T M \dot{\tilde{q}} \\
&= \sigma^T [-C\sigma - F\sigma - K_D \sigma] + \frac{1}{2}\sigma^T \dot{B} \sigma + \tilde{q}^T M \dot{\tilde{q}} \\
&= -\sigma^T C \sigma - \sigma^T F \sigma - \sigma^T K_D \sigma + \frac{1}{2}\sigma^T \dot{B} \sigma + \tilde{q}^T M \dot{\tilde{q}} \\
&= -\sigma^T 2(\dot{B} - 2C)\sigma - \sigma^T F \sigma - \sigma^T K_D \sigma + \tilde{q}^T M \dot{\tilde{q}} \\
&= 0 - \sigma^T F \sigma - \sigma^T K_D \sigma + \tilde{q}^T M \dot{\tilde{q}}
\end{aligned}$$

A questo punto non rimane che eliminare l'unico termine non negativo $\tilde{q}^T M \dot{\tilde{q}}$. Sostituiamo al termine $\sigma^T K_D \sigma$ l'espressione di $\sigma = \dot{\tilde{q}} + \Lambda \tilde{q}$. Poiché la matrice M è arbitraria, visto che non è presente nel controllo, la impongo in modo tale da cancellare il termine positivo: $M = \Lambda^T K_D$

$$\begin{aligned}
\dot{V} &= -\sigma^T F \sigma - \sigma^T K_D \sigma + \tilde{q}^T M \dot{\tilde{q}} \\
&= -\sigma^T F \sigma - (\dot{\tilde{q}} + \Lambda \tilde{q})^T K_D (\dot{\tilde{q}} + \Lambda \tilde{q}) + \tilde{q}^T M \dot{\tilde{q}} \\
&= -\sigma^T F \sigma - \dot{\tilde{q}}^T K_D \dot{\tilde{q}} - \tilde{q}^T \Lambda K_D \Lambda \tilde{q} - \tilde{q}^T \Lambda^T \dot{\tilde{q}} - \tilde{q}^T \Lambda^T \Lambda \tilde{q} + \tilde{q}^T M \dot{\tilde{q}} \\
&= -\sigma^T F \sigma - \dot{\tilde{q}}^T K_D \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_D \Lambda \tilde{q} \cancel{- \tilde{q}^T \Lambda^T K_D \dot{\tilde{q}}} - \tilde{q}^T \Lambda^T K_D \Lambda \tilde{q} + \cancel{\tilde{q}^T [\Lambda^T K_D] \dot{\tilde{q}}} \\
&< 0
\end{aligned}$$

In defintiva, lo stato del sistema $[\tilde{q} \quad \sigma] \to 0$ e quindi il sistema è globalmente asintoticamente stabile. Possiamo dunque affermare di aver trovato una legge di controllo che è in grado di compensare tutti i parametri dinamici

Sulla base di questo risultato, la legge di controllo può essere resa adattativa rispetto al vettore dei parametri π. Supponiamo ora di avere una incertezza sui parametri dinamici considerando non il vettore π ma una sua stima $\hat{\pi}$, allora il modello dinamico sarà il seguente:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F\dot{q} + g(q) = u = Y(q, \dot{q}, \ddot{q})\hat{\pi}$$

La legge di controllo adottata precedentemente dovra essere modificata come segue, dove saranno presenti questa volte le stime di tutte le matrici costituenti il modello dinamico. Sfruttando poi la proprietà di linearità rispetto ai parametri dinamici otteniamo l'ultimo membro:

$$\boldsymbol{u} = \hat{B}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{F}\dot{q}_r + \hat{g}(q) + K_D\sigma = \boldsymbol{Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{\pi} + K_D\sigma}$$

Sostituendo tale legge di controllo nel modello dinamico otterremo la seguente uguaglianza in cui $\tilde{\pi} = \hat{\pi} - \pi$ è il vettore errore dei parametri dinamici mentre le matrici $\tilde{B}, \tilde{C}, \tilde{F}, \tilde{g}$ caratterizzano l'errore di modellazioene:

$$\begin{aligned} B(q)\dot{\sigma} + C(q,\dot{q})\sigma + F\sigma + K_D\sigma &= -\tilde{B}(q)\ddot{q}_r - \tilde{C}(q,\dot{q})\dot{q}_r - \tilde{F}\dot{q}_r - \tilde{g}(q) \\ &= -Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)(\hat{\pi} - \pi) \\ &= -Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\tilde{\pi} \end{aligned}$$

In maniera analoga a quanto fatto precedentemente dobbiamo adottare Lyapunov per verificare che la legge di controllo scelta rende il sistema globalmente asintoticamente stabile. In questo caso però nella funzione di Lyapunov, sempre scelta quadratica, dovremo aggiungere un terzo termine che si riferirà all'asintotica stabilità rispetto all'incertezza dei parametri dinamici:

$$V(\sigma,\tilde{q},\tilde{\pi}) = \frac{1}{2}\sigma^T B\sigma + \frac{1}{2}\tilde{q}^T \Lambda K_D \tilde{q} + \frac{1}{2}\tilde{\pi}^T K_\pi \tilde{\pi} > 0 \quad \forall \sigma,\tilde{q},\tilde{\pi} \neq 0$$

Calcolando la derivata avremo una prima parte uguale a quanto ricavato in precedenza e ovviamente un ulteriore termine dovuto a quello che abbiamo aggiunto:

$$\dot{V} = -\sigma^T F\sigma - \dot{\tilde{q}}^T K_D \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_D \Lambda \tilde{q} + \tilde{\pi}^T\left(K_\pi \dot{\tilde{\pi}} - Y^T(q,\dot{q},\dot{q}_r,\ddot{q}_r)\sigma\right)$$

Osserviamo però che $\dot{\tilde{\pi}} = \dot{\hat{\pi}} - \dot{\pi} = \dot{\hat{\pi}}$ in quanto i parametri dinamici reali π che non conosciamo hanno valore costante per cui la derivata $\dot{\pi} = 0$. Allora posso scegliere arbitrariamente $\dot{\hat{\pi}}$ e lo faccio in modo da annullare il termine tra parentesi in modo da ritrovarmi nuovamente $\dot{V} < 0$: $\dot{\hat{\pi}} = K_\pi^{-1} Y^T(q,\dot{q},\dot{q}_r,\ddot{q}_r)\sigma$

In definitiva, abbiamo aggiunto alla nostra legge di controllo anche una azione adattativa che fa fronte alle variazione dei parametri dinamici.

Facciamo alcune osservazione circa questo tipo di controllo. Ricordiamo che il focus di questa legge è portare a zero l'errore di inseguito, quindi $\tilde{q} \to 0$. Invece, tale legge di controllo non garantisce che i parametri dinamici con incertezza convergano a quelli reali, ovvero $\hat{\pi} \to \pi$. Infatti, ricordiamo che la matrice Y è una matrice rettangolare bassa $2n \times \rho$ mentre il vettore dei parametri dinamici è una vettore colonna $\rho \times 1$. La Y per le sue caratteristiche possiederà sempre un nullo diverso dall'insieme vuoto e il vettore $\hat{\pi} - \pi$ potrebbe entrare in esso senza che tale differenza sia nulla. Questo dal punto di vista pratico vuol dire che $\hat{\pi} \neq \pi$, quindi i parametri con incertezza non coincidano con quelli reali. In realtà, il sistema può inseguire la traiettoria con errore nullo nonostante tale condizione e questo perché i parametri dinamici sono in numero maggiore dei gradi di libertà del robot.

Infine, facciamo un esempio pratico in cui abbiamo un manipolatore che deve trasportare un carico sconosciuto che sbilancerà uno o più parametri dinamici (creando l'incertezza). In particolare, supponiamo che $p-1$ parametri li conosciamo mentre l'ultimo è incerto. Allora vogliamo realizzare un controllo adattativo: $u = Y\hat{\pi} + K_D\sigma$

Per risolvere il problema possiamo adottare una di queste soluzioni:

a. **Scelta dei Pesi di $\boldsymbol{K_\pi}$**: tipicamente la matrice K_π si sceglie come matrice diagonale e isotropa (tutti guadagni uguali) in modo che l'azione di adattamente sia uguale per tutti i parametri. Nel nostro caso, sapendo che l'ultimo parametro dinamico è incerto, riduciamo il guadagno del relativo elemento in K_π. Così facendo, quando calcoleremo K_π^{-1}, il relativo termine avrà un valore elevato concentrando di fatto l'azione di adattamento sul parametro problematico;
b. **Partizionamento della $\boldsymbol{Y}$**: partizioniamo la Y in una matrice Y'_{P-1} e un vettore colonna Y_L. Riscriviamo il controllo: $u = Y'\pi' + Y_L\hat{\pi}_L + K_D\sigma$
   Così facendo abbiamo una compensizione perfetta dei parametri π', mentre l'adattamento verrà effettuato solo sul parametro incerto ponendo $\dot{\hat{\pi}}_L = K_\pi^{-1}Y_L^T K_D\sigma$. In questo caso, visto che il vettore colonna Y_L è di rango pieno pari ad 1, per cui con questo controllo avremo che $Y_L(\hat{\pi}_L - \pi_L) \to 0$ e quindi possiamo identificare puntualmente il parametro incerto.

Facendo un confronto l'approccio (b) è certamente più prestante di (a), tuttavia il primo è preferibile perché non possiamo avere certezza assoluto di conoscere i parametri π' del secondo approccio.

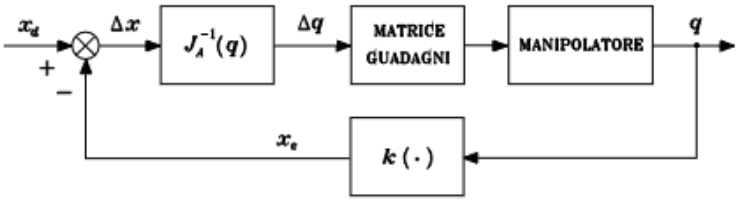
7.6 Controllo del Moto nello Spazio Operativo

7.6.1 Linee Guida per il Controllo nello Spazio Operativo

In precedenza abbiamo messo in evidenza l'immediatezza di pianificare traiettorie nello spazio operativo visto che il task eseguito dal manipolatore è definito in questo spazio. Tuttavia, sappiamo anche che l'azione del controllore, la quale sarà un set di forze generalizzate, lavora nello spazio dei giunti poiché lo stato del manipolatore è definito in quest'ultimo. Da ciò denotiamo la necessità di effettuare una conversione dello specifiche/riferimenti definite nello spazio operativo e nello spazio dei giunti mediante un algoritmi di inversione cinematica. Ora nel controllo nello spazio dei giunti l'inversione cinematica viene eseguita da un blocco a se stante fuori dall'anello di controllo, invece nello spazio operativo avvengono delle trasformazioni di coordinate che fanno le veci dell'inversione cinematica. Quest'ultima soluzione, contrariamente a quanto si può immaginare, è molto più complesso del blocco di inversione a se stante. Lo studio del controllo del moto perciò è motivato dall'esistenza di applicazioni in cui il manipolatore deve interagire con l'ambiente circostante. Esistono due linee guida principali per controllare un manipolatore nello spazio operativo:

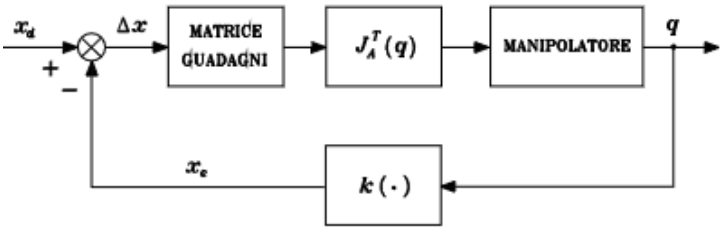
A. Controllo con Inversa dello Jacobiano

Supponiamo che il riferimento x_d è generato da un algoritmo di pianificazione delle traiettorie che opera nello spazio operativo. Questo riferimento viene confrontato con la posa effettiva del manipolatore, la quale è ottenuta mediante la cinematica diretta che sfrutta la misura delle variabili di giunto q letta dagli encoder del robot stesso. Dal confronto tra x_d e x_e viene generato un segnale errore Δx definito nello spazio operativo. Moltiplicando quest'ultimo per J_A^{-1} deriviamo un errore Δq nello spazio dei giunti. A sua volta dal prodotto di Δq per una matrice dei guadagni otteniamo una forza generalizzata che agisce sul manipolatore e che di fatto rappresenta l'azione di controllo nello spazio dei giunti. La matrice dei guadagni può essere vista come una molla generalizzata che produce una forza nello spazio dei giunti tale l'errore $\Delta q \to 0$. Se la matrice dei guadagni è diagonale i cui elementi sono costanti, allora possiamo immaginare la presenza di N molle nello spazio dei giunti.



B. Controllo con Trasposta dello Jacobiano

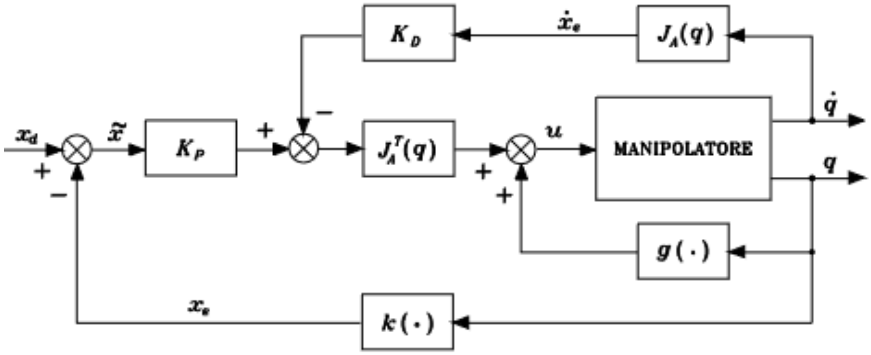
Esattamente come nello schema precendete calcoliamo l'errore Δx da moltiplicare questa volta con la matrice dei guadagni, la quale mi da come output una forza generalizzata nello spazio operativo. Anche in questo caso possiamo immaginare la matrice dei guadagni come una molla e in particolare essa sarà applicata all'end-effector in modo tale da annullare l'errore. Convertiamo, mediante J_A^T, la forza generalizzata nello spazio operativo in una forza nello spazio dei giunti, che andrà poi applicata al manipolatore.



7.6.2 Controllo PD con Compensazione di Gravità

Il **Controllo PD con Compensazione di Gravità** viene realizzato impiegando la seguente legge di controllo, la quale è costituita da una azione di compensazione non lineare della gravità e una azione PD lineare esercitata nello spazio operativo: $\boldsymbol{u = g(q) + J_A^T K_p \tilde{x} - J_A^T K_D J_A \dot{q}}$

Lo schema implementativo di questo controllo è il seguente:



Il controllo PD+ nello spazio operativo ha come principale obiettivo quello di stabilizzare una postura e annullare l'errore di posa nello spazio operativo: $\tilde{x} = x_d - x_e \to 0$

Inoltre, come si può osservare dalla struttura dello schema a blocchi, esso segue la seconda linea guida citata in precedenza che adotta $J_A^T(q)$.

Andiamo a studiare questo controllo in maniera analoga a quanto fatto nello spazio dei giunti, perciò facciamo ricorso a Lyapunov definendo la seguente funzione:

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

Calcoliamo la deriata:

$$\begin{aligned}
\dot{V} &= \dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{\tilde{x}}^T K_P \tilde{x} \\
&= \dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{q}^T J_A^T(q) K_P \tilde{x} \\
&= \dot{q}^T[-C(q,\dot{q})\dot{q} - F\dot{q} - g(q) + u] + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{q}^T J_A^T(q) K_P \tilde{x} \\
&= -\dot{q}^T C(q,\dot{q})\dot{q} - \dot{q}^T F\dot{q} - \dot{q}^T g(q) + \dot{q}^T u + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{q}^T J_A^T(q) K_P \tilde{x} \\
&= -\dot{q}^T \frac{1}{2}\left(\dot{B}(q) - 2C(q,\dot{q})\right)\dot{q} - \dot{q}^T F\dot{q} + \dot{q}^T(u - g(q) - J_A^T K_P \tilde{x}) \\
&= -\dot{q}^T F\dot{q} + \dot{q}^T(u - g(q) - J_A^T K_P \tilde{x})
\end{aligned}$$

Progettiamo il controllore: $\boldsymbol{u = g(q) + J_A^T K_p \tilde{x} - J_A^T K_D J_A \dot{q}}$

Sostituendo nella derivata di V otteniamo due termini che sono entrambi negativi e quindi di fatto l'azione di controllo rende il sistema globalmente asintoticamente stabile:

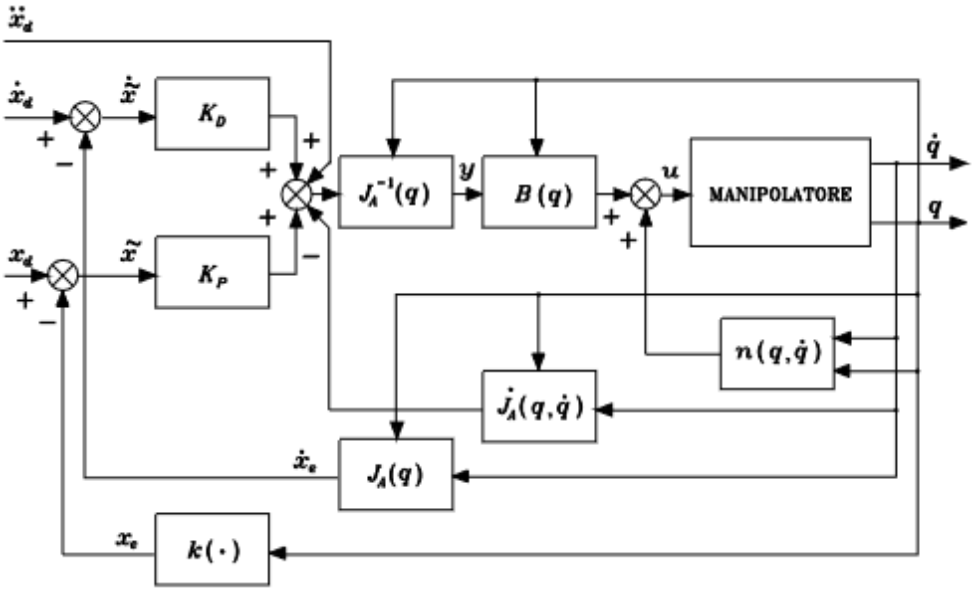
$$\begin{aligned}
\dot{V} &= -\dot{q}^T F\dot{q} + \dot{q}^T\left(g(q) + J_A^T K_p \tilde{x} - J_A^T K_D J_A \dot{q} - g(q) - J_A^T K_P \tilde{x}\right) \\
&= -\dot{q}^T F\dot{q} - \dot{q}^T J_A^T K_D J_A \dot{q} < 0
\end{aligned}$$

Se lo Jacobiano analitico è di rango pieno e la matrice $K_P > 0$ ed è simmetrica siamo in grado di raggiungere la cosidetta postura di equilibrio $J_A^T K_P \tilde{x} = 0$, la quale implicherà che $\tilde{x} = x_d - x_e = 0$ e quindi $x_d = x_e$.

7.6.3 Controllo a Dinamica Inversa

Il **Controllo a Dinamica Inversa** viene realizzato impiegando la seguente legge di controllo, la quale è costituita da: $\boldsymbol{u = B(q)y + n(q,\dot{q})}$

Lo schema implementativo di questo controllo è il seguente:



Questo tipo di controllo, vista la tempo-varianza dei riferimenti, risolve problemi di tracking contrariamente al controllo PD+ che è orientato alla stabilizzazione di una postura. Inoltre, come si può osservare dalla struttura dello schema a blocchi, esso segue la prima linea guida citata in precedenza che adotta $J_A^{-1}(q)$.

Iniziamo lo studio di questo controllo partendo dal modello dinamico del manipolatore nello spazio dei giunti in forma compatta raccogliendo i termini di Corilis, forza centrifuga e gravitazionali nel termine $n(q,\dot{q}) = C(q,\dot{q})\dot{q} + F\dot{q} + g(q)$: $B(q)\ddot{q} + n(q,\dot{q}) = u$

Applichiamo un azione di controllo linearizzante a dinamica inversa: $\boldsymbol{u = B(q)y + n(q,\dot{q})}$

Essa mi portarà ad un sistema doppio integratore: $\ddot{q} = y$

In pratica questa relazione lega l'accellerazione nello spazio dei giunti a questo segnale di controllo ausilirio che dovremo definire e progettare in maniera tale che i riferimenti x_d di spazio operativi vengano inseguiti. Per fare ciò, sfruttiamo l'equazione della cinematica differenziale del secondo ordine: $\dot{x}_e = J_A\ddot{q} + \dot{J}_A + \dot{q}$

Allora il mio controllo ausiliario può essere visto come:

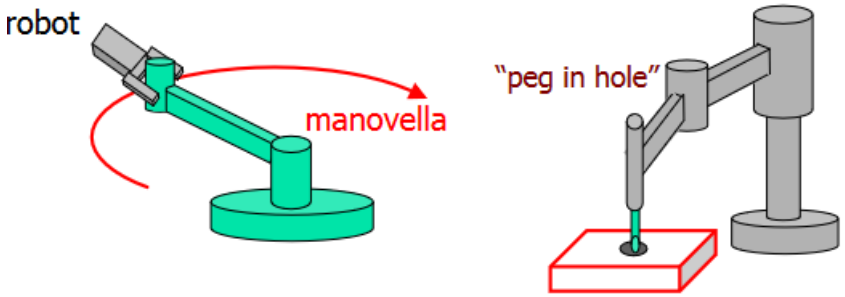
$$y = J_A^{-1}(q)\left(\ddot{x}_d + K_D\dot{\tilde{x}} + K_P\tilde{x} - \dot{J}_A(q,\dot{q})\dot{q}\right)$$

Grazie a questo controllo il sistema errore $\ddot{\tilde{x}} + K_D\dot{\tilde{x}} + K_P\tilde{x} = 0$ risulterà essere globalmente asintoticamente stabile poiché $\tilde{x} \to 0$.

8. Force Control

8.1 Characterization of Interaction Control

The interaction control between manipulator and enviroment is a fundamental aspect in all tasks where these two elements are in contact, for example mechanical processing or assembly tasks. In these situations, the environment impose strong *motion constraints* about geometrical path for end-effector or applied force on a particular objects and so on.



Now, we can think to use a motion control or positional control strategy, however these approachs require:

- A perfect knowledge of kinematic and dynamic model of manipulator;
- A perfect knowledge of environment's geometry;
- A perfect trajectory execution.

But probably in practice at least one of these elements is partially know or completely unknow and for these reasons born different problems:

- Big position error of end-effector.
- Increasing of contact forces.
- Saturation of actuators.
- Breaking of contact elements.

These considerations give us two important results:

- ✓ We should assicurate a compliant behavior (a sort of elasticity) for the end-effector of manipulator.
- ✓ The contact force is the quantity that better describes the state of the interaction. So if we have a measurement of contact force probably we can improve the performance.

The interaction control strategies can be differentiate in two macro-categories:

A. **Strategy for Indirect Force Control**: the force control is implemented by a motion control without an explicit control loop for the force. Examples of this approach are:
   A. **Compliance Control**.
   B. **Impendence Control**.

B. **Strategy for Direct Force Control**: the force control is implemented thanks a control loop for the force and, thanks to this, we can impose a desired contact force. Example of this approach are:
   A. **Force Control**;
   B. **Force/Motion Ibrid Control**.

For conclude it's necessary specify that all these control strategy will study are thinking in operational space because contact force is naturally defined in this space.

8.2 Compliance Control

8.2.1 Compliance Behaviour

First of talk about compliance control it's necessary explain the concept of compliance. The **Compliance** *is the ability to give up applied force and is definite as the inverse of rigidity, so the tendency of a body to oppose the deformation due to the effect of a force*. Thanks this particular behaviour we can facilitate the interaction and the execution of specific task.

For understand this concept from analytical point of view, we consider two elastically coupled rigid bodies R and S with relative frames as in figure. We suppose that, in absence of interaction forces and moments, the two frames coincide. The elastic force h_s between R and S is given by product between **Stiffness Matrix** $\boldsymbol{K} \in \boldsymbol{R}^{6\times 6}$, which is symmetric and positive semi-definite matrix, and spring's deformation dx_{rs}:

$$h_s = \begin{bmatrix} f_s \\ \mu_s \end{bmatrix} = \begin{bmatrix} K_f & K_c \\ K_c^T & K_\mu \end{bmatrix} \begin{bmatrix} do_{rs} \\ \omega_{rs} dt \end{bmatrix} = \begin{bmatrix} K_f & K_c \\ K_c^T & K_\mu \end{bmatrix} \begin{bmatrix} o_s - o_r \\ (\omega_s - \omega_r) dt \end{bmatrix} = K dx_{rs}$$

Notice that the stiffness matrix K can be see a blocks matrix where these blocks are 3×3 matrix:

- K_f = matrix of traslational stiffness.
- K_μ = matrix of rotational stiffness;
- K_c = matrix of coupling stiffness.

If we consider the inverse relation, we can said that deformation dx_{rs} is give by product between **Compliance Matrix** $\boldsymbol{C}$, which is symmetric and positive semi-definite, and elastic forces h_s:

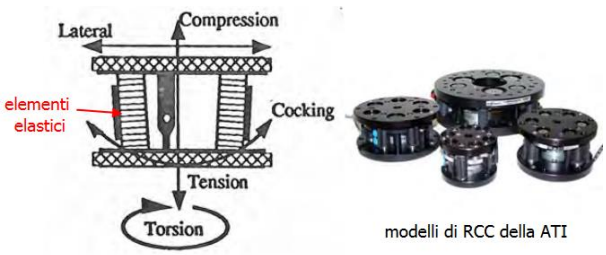
$$\boldsymbol{dx_{rs} = Ch_s}$$

So we can conclude that <u>the compliance matrix is the inverse of stiffness matrix</u>.

At this point, we can distinguish two different types of compliance:

A. Passive Compliance

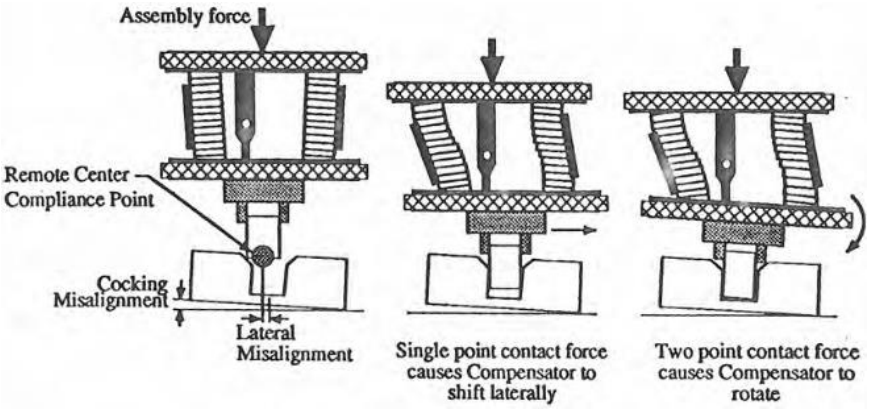
Is the compliance obtained thanks a RCC (Remote Center of Compliance), a particular mechanical device collocated *between wrist and end-effector* characterized by the presence of different elastical elements.



For understand how to work this device, we can think a task where the manipulator must insert a pin into a hole. The RCC assicurate:

- High vertical rigidity.
- High horizontal compliance.

Thanks to this, also in presence of error position, the contact force help mechanical coupling between two elements.



Now thanks the RCC we able to impose a matrix of coupling stiffness K_c symmetric or null and obtain a decoupling between translation and rotational part. The fundamental problem of this device is low versatility to different operating conditions and generic interaction tasks. For overcome the problem of passive compliance, it is necessary to switch to active compliance.

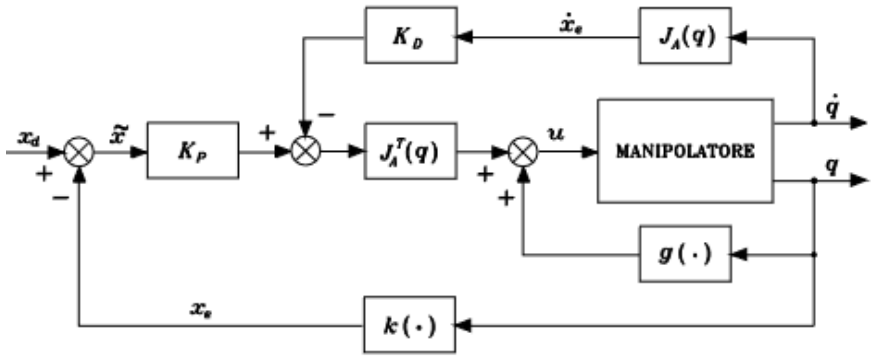
B. Active Compliance

the robot is equipped with a control that reacts appropriately to the forces applied (and measured by the sensor) to the end-effector.

8.2.2 Definition of Compliance Control

The **Compliance Control** *consist to realize an active compliance where we control the contact force through positioning of end-effector, which it has a behaviour similary to generalized spring*. The follow block scheme and PD+ control law implement the compliance control: $\boldsymbol{u = J_{Ad}^T(q,\tilde{x})\left(K_p\tilde{x} - K_D J_{Ad}(q,\tilde{x})\dot{q}\right) + g(q)}$

Thanks this definition, we understand the compliance control is indirect force control because we don't acting directly on force but on position.



First of all, we want understand how it was modify classic PD+ control law $u = g(q) + J_A^T K_p \tilde{x} - J_A^T K_D J_A \dot{q}$ for obtain the previously control law. Consider dynamic model of manipulator: $B(q)\ddot{q} + C(q,\dot{q})\dot{q} + Fq + g(q) = u - J^T(q)h_e$

We want use a PD+ control for tend to zero the pose error: $\tilde{x} = x_d - x_e \to 0$

The standard PD+ control law has the fundamental problem that it's not able reach this scope when external forces $h_e \neq 0$. In fact, at equilibrium, the position error $\tilde{x}$ is dependent by compliance properties of manipulator. We can see this consideration from analytical point of view substituting the control law in dynamic model, in this way we obtain that pose error depend by compliance matrix K_P^{-1}:

$$J_A^T(q)K_p\tilde{x} = J^T(q)h_e \;\rightarrow\; \boldsymbol{\tilde{x} = K_P^{-1}T_A^T(x_e)h_e = K_P^{-1}h_A}$$

Only case such that where error $\tilde{x} = 0$ in presence of external forces $h_e \neq 0$ is when $h_e \in \mathcal{N}(J^T)$ because contact forces are completely balanced by mechanical structure of manipulator (similary when it is in shoulder singularity). But green relation give us a important result, the **<u>manipulator has a behaviour similary to generalized spring in the operatinal space with compliance K_P^{-1} with respect to the equivalent forces h_a</u>** (o stiffness K_P if we consider inverse relation $h_A = K_p\tilde{x}$). So, through software, we can modify in very easy way the compliance's properties of system.

At this point for modify standard PD+ contro law it's necessary change error's definition because it is dependent from end-effector orientation respect to external force h_e also in case of elementary displacements. This condition makes the choice of compliace properties difficult respect to the task. Considering:

- Tool frame $O_e - x_e y_e z_e$ with homogenous transformation matrix T_e.
- Desired frame $O_d - x_d y_d z_d$ with homogenous transformation matrix T_d.

$$T_e = \begin{bmatrix} R_e & o_e \\ 0^T & 1 \end{bmatrix} \qquad T_d = \begin{bmatrix} R_d & o_d \\ 0^T & 1 \end{bmatrix}$$

Now, the variations of position and orientation of first frame respect the second frame can be express similary as homogenous transformation matrix T_e^d:

$$T_e^d = (T_d)^{-1}T_e = \begin{bmatrix} R_e^d & o_{d,e}^d \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} R_d^T R_e & R_d^T(o_e - o_d) \\ 0^T & 1 \end{bmatrix}$$

At this point we can define a new pose error where angular part is the vector ϕ_{de} of Eulero angle extract by rotation matrix R_e^d between two frames. Notice that, the sign minus is write only for difference between quantities desired and effective. However, in PD+ control law we have also a derivative term in $\dot{\tilde{x}}$ which, rembering differential cinematic $v_e = J(q)\dot{q}$:

$$\tilde{x} = -\begin{bmatrix} o_{de}^d \\ \phi_{de} \end{bmatrix}$$
$$\downarrow$$
$$\dot{\tilde{x}} = -T_A^{-1}(\phi_{de})\begin{bmatrix} R_d^T & O \\ O & R_d^T \end{bmatrix} v_e = -T_A^{-1}(\phi_{de})\begin{bmatrix} R_d^T & O \\ O & R_d^T \end{bmatrix} J(q)\dot{q} = -J_{Ad}(q,\tilde{x})\dot{q}$$

Thanks to definition of $\tilde{x}$ and $\dot{\tilde{x}}$, we can modify the PD+ control law. In practice, if the operational space variables are only position variables, the PD+ control law see in motion control has the error pose defined respect base frame of robot, the compliance control law is referred to desired frame (in fact there is pedix "d"):

$$u = g(q) + J_A^T(q)K_p\tilde{x} - J_A^T(q)K_D J_A(q)\dot{q}$$
$$\downarrow$$
$$\boldsymbol{u = g(q) + J_{Ad}^T(q,\tilde{x})K_p\tilde{x} - J_{Ad}^T(q,\tilde{x})K_D J_{Ad}(q,\tilde{x})\dot{q}}$$

At this point we can realize a stability analysis considering two cases:

A. Contactless: asymptotical stability, so $\tilde{x} = 0$, can be proved with Lyapunov function on the right in the hypothesis that K_P and K_D are symmetric and definite positive.

$$V = \frac{1}{2}\dot{q}^T B(q)\dot{q} + \frac{1}{2}\tilde{x}^T K_P \tilde{x}$$

B. With Contact: to equilibrium it's verify this equality: $J_{Ad}^T(q)K_P\tilde{x} = J^T(q)h_e$

It's possible demonstrate these relation about the forces h_e that manipulator apply on environment and about expression of pose error dx_{ed} of end-effector to equilibrium:

$$\boldsymbol{h_e = \left(I_6 + KK_P^{-1}\right)Kdx_{rd}}$$
$$\boldsymbol{dx_{ed} = K_P^{-1}h_e = K_P^{-1}\left(I_6 + KK_P^{-1}\right)Kdx_{rd}}$$

Where:

- K = is stiffness matrix.
- K_P = is a matrix where its element should choose based on enviroment's properties with the manipulator must interact.
- dx_{rd} = is the elementary displacement between frame $O_r - x_r y_r z_r$ relative to environment and desired frame.

These red relations say that the pose error of end-effector depends from:

- ❖ Rest position of the environment.
- ❖ Desired pose impose to control system of manipulator. In pratice, acting on K_P^{-1} we can module the weight such that to make compliant to the manipulator or the enviroment.

In example of pin in the hole, we choose K_p^{-1} such that along pin insertion direction the enviroment is compliant and manipulator rigid, in opposite in lateral direction we have the opposite configuration. For conclude:

- ✓ Along directions in which the environment is compliant, pose error dx_{ed} tends to zero. In particular, the elastic force of interaction depends on the stiffness of the environment and on the deviation dx_{re} between the equilibrium pose and the rest position of the environment: $dx_{ed} \to 0$
- ✓ Along the directions in which the manipulator is compliant, the pose of the end-effector tends to coincide with the rest pose of the environment: $dx_{ed} \cong dx_{rd}$
  In particular, the elastic force of interaction depends on the stiffness of the manipulator and on the deviation dx_ed between the desired pose and the equilibrium pose of the end effector.

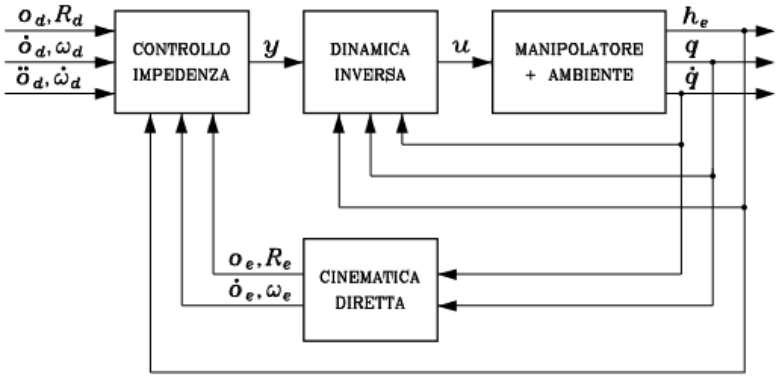
8.3 Impedence Control

8.3.1 Definition of Impedence Control

The **Impedence Control** *imposes that the manipulator in contact with the environment shows a behavior similar to a mechanical impedance, assimilable to a mass-spring-damper system.*
The following control law and scheme implement impedence control:

$$\boldsymbol{u = B(q)y + n(q,\dot{q}) + J^T(q)h_e}$$

With: $$\boldsymbol{y = J_A^{-1}(q)M_d^{-1}\left(M_d\ddot{x}_d + K_D\dot{\tilde{x}} + K_P\tilde{x} - M_d\dot{J}_A(q,\dot{q})\dot{q} - h_A\right)}$$



The basic idea of this strategy is that power flow between two physical systems is always definite as product between generalized flow and generalized effort. Now:

a) In electrical systems, the flow is current and the effort is tension.
b) In mechanical systems, the flow is (linear or angular) velocity and the effort is force or moment.

The mechanical impedence define a dynamic relation between velocity (or displacement) and force in a mechanical system. So, thanks to impendence control, we can impose a desired dynamics behaviour for our manipulator when it is in contact with the environment.

At this point we can study formally this control strategy and for this scope considering this dynamic model: $$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + Fq + g(q) = u - J^T(q)h_e$$
But, especially, two differente cases:

A. Absence of Contact Force Measurement:

Our robot isn't equipped with a force sensor for obtain a force measurement. In this case we choose a standard inverse dynamic control law u: $$u = B(q)y + n(q,\dot{q})$$
And control y: $$y = J_A^{-1}(q)M_d^{-1}\left(M_d\ddot{x}_d + K_D\dot{\tilde{x}} + K_P\tilde{x} - M_d\dot{J}_A(q,\dot{q})\dot{q}\right)$$
We obtain this error's dynamic $\tilde{x}$ of second order: $M_d\ddot{\tilde{x}} + K_D\dot{\tilde{x}} + K_P\tilde{x} = M_dB_A^{-1}(q)h_A$
Where:

- M_d = is a diagonal positive definite matrix.
- $B_A(q) = J_A^{-T}(q)B(q)J_A^{-1}(q)$ = is the inertial matrix of manipulator in operational space, which depends by configuration q and it's positive definite if J_A is full rank.

The previously relation define **<u>a relation of Generalized Mechanical Impedence between resultant forces vector $M_dB_A^{-1}(q)h_A$ and displacement vector $\tilde{x}$ in operational space</u>**.
In particular, this impedence is assimilable to the system characterized by:

- Mass matrix M_d.
- Dumper matrix K_D.
- Rigidity matrix K_P.

We understand that we can characterize completely the dynamic of system through specifying the matries M_d, K_D, K_P. Now, the fundamental problem of this scenario is that the inverse of inertial matrix B_A^{-1} makes the *system coupled*.

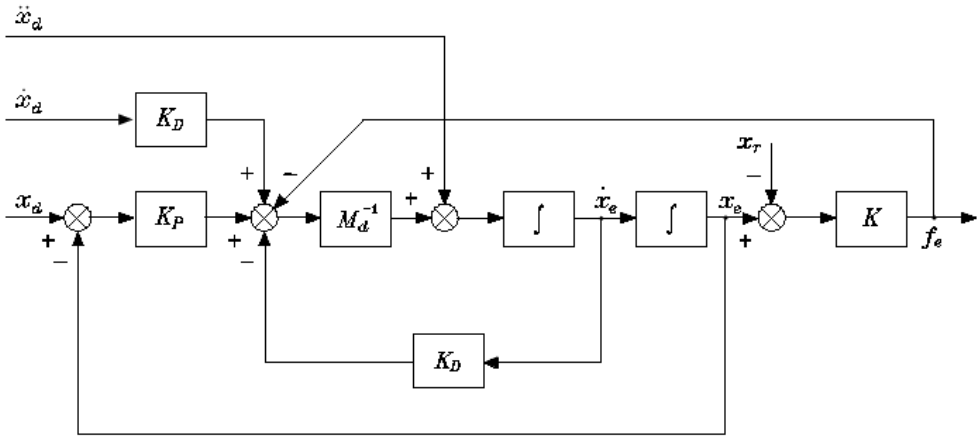
B. Presence of Force Contact Measurement:

This measurement is provided by a force sensor mounted on manipulator's wrist. Thanks this measurement we can keep linearity and decoupling properties during the interation with environment. In this case we choose the control law: $\boldsymbol{u = B(q)y + n(q,\dot{q}) + J^T(q)h_e}$

And control input: $\boldsymbol{y = J_A^{-1}(q)M_d^{-1}\left(M_d\ddot{x}_d + K_D\dot{\tilde{x}} + K_P\tilde{x} - M_d\dot{J}_A(q,\dot{q})\dot{q} - h_A\right)}$

We obtain this error's dynamic $\tilde{x}$ of second order: $\boldsymbol{M_d\ddot{\tilde{x}} + K_D\dot{\tilde{x}} + K_P\tilde{x} = h_A}$

As in previously case, we have obtained a similary relation but in this case it's a *linear mechanical impedence* between equivalent forces h_A and error $\tilde{x}$. Also the previously control scheme of impedence control become:



Notice that, in absense of force sensor, we can think to measure the current of actuators, so the joint's torque and, thanks the Jacobian of manipulator, the force to end-effector. The only problem of this approach is that we lost some precision for two reasons:

- We can have low precision on mechianical reduction coefficient between actuators and joints.
- We can have an uncertainty regarding the parameters of the mathematical model of the actuator.

But, the most important problem about impedence control is the choose of parameter M_d, K_D and K_P. In fact we can have two tendences opposite:

a. If we want a more compliance for impose to limit interaction forces, then we choose a little values for the elements of K_P.
b. If we want more rejection to disturbance caused by model uncertains and approximations introduced with calculations of inverse dynamic, then we choose a big values for the elements of K_P. For example this condition happen when we havent' interaction along free motion direction where the impedence control is equivalent to simple inverse dynamics control.

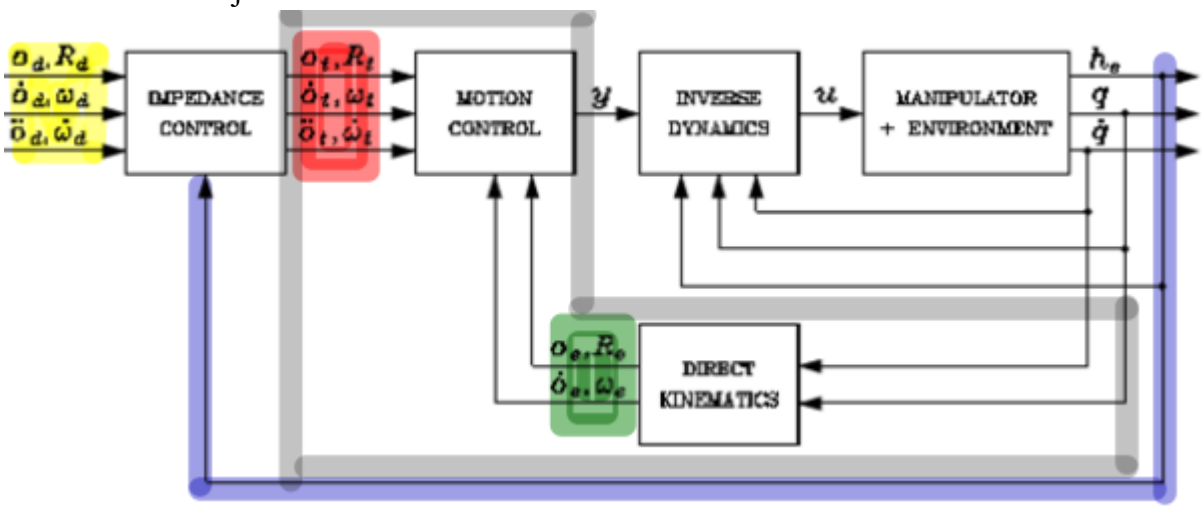
This critical situations is more difficult in case of articulated task where we have for example different interaction scenarios that require a changing of parameters. For resolve all these problems we can decide to use the admittance control.

8.3.2 Admittance Control

The **Admittance Control** *consist to decouple the motion control problem and force control problem for choose correctly gains of two controller for obtain respectively a high value of rejection disturbance factor and a desired behaviour during the interaction with environment*. How suggests the definition of admittance control, the block scheme is characterized by two control loop:

- **Outer Loop**: this loop realize the impedence control through force/torque measurements h_e (in blue) and introduction of a particular frame. The **Compliant Frame** is a reference cartesian frame that describe the ideal behaviour of end-effector through the impedence control. This frame is specify in terms of:
  - Posizion o_t of origin.
  - Rotation matrix R_t.
  - Linear and angular velocities and accellerations calculate for integration of following impedence equation starting from force measurement h_e where the matries M_t, K_{Dt}, K_{Pt} are compliance frame's parameters and $\tilde{z}$ represents the operational space error between the desired frame and the compliant frame: $\boldsymbol{M_t\ddot{\tilde{z}} + K_{Dt}\dot{\tilde{z}} + K_{Pt}\tilde{z} = h_e^d}$

  The impedence control produce in output the kinematic variables of the compliant frame.
- **Inner Position Control Loop**: this loop realize the motion control using kinematic variables of the compliant frame as reference. Also the inner loop (in grey) have as scope disturbance rejection.



In definitely, we choose the compliant frame such that to guarantee:

✓ A choice of impedence control gains for obtain a good behaviour during the interaction with environment.

✓ A choice of motion control gains for obtain a high values of rejection disturbances coefficient.

Notice that the name "admittance control" derive from equations $M_t\ddot{\tilde{z}} + K_{Dt}\dot{\tilde{z}} + K_{Pt}\tilde{z} = h_e^d$, which represent a mechanical admittance. In particular:

The admittance characterize the mapping: $forces \rightarrow kine.var$

The impendence characterize the mapping: $kine.var \rightarrow forces$

For conclude, about the stability of system it's necessary that *pass band of motion control should be bigger than pass band of impendence control*.

8.4 Force Control

The **Force Control** *consist to control directly this quantities and allows us to specify a desired interaction force.* Initially we can think to realize a control scheme similary to other control strategy, so with:

- PD stabilizable action that act on force error.
- Non linear compensation.

The problem is that force measurement are corrupted by noise and for this *reason a derivative action is not avaible*. This consideration is the reason why the following schemes are based on force, position and velocity measurements. Now, for study these control strategies indipendently by enviroment's geometry we impose these hypothesis:

- ✓ Suppose that operational space is characterized only position variables: $x_e = o_e$
- ✓ Thanks the previously hypothesis we obtain that geometrical jacobian and analitical jacobian are equal: $J_A(q) = J(q)$
- ✓ The interaction force with the enviroment is characterized by a elastic model, so stiffness matrix for displacement: $f_e = K(x_e - x_r)$

The different version of direct control force that we show are:

A. Force Control with Inner Position Loop:

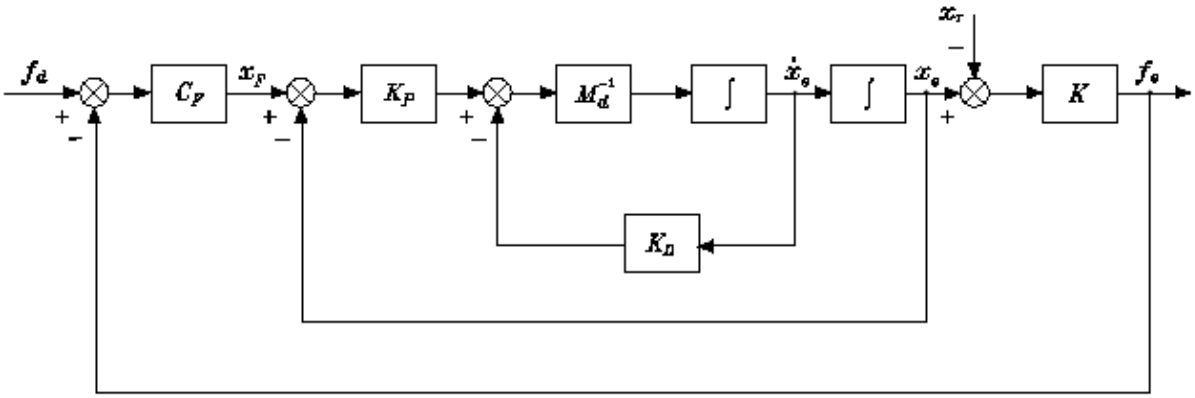
The control law and block scheme are following: $u = B(q)y + n(q,\dot{q}) + J^T(q)h_e$

$$\boldsymbol{y = J^{-1}(q)M_d^{-1}\left(-K_D\dot{x}_e + K_P(x_F - x_e) - M_d\dot{J}(q,\dot{q})\dot{q}\right)}$$

Notice that x_F is a reference linked to error force thanks this relation where compare the compliance matrix C_F:

$$x_F = C_F(f_d - f_e) \qquad where: \qquad C_F = K_F + K_I\int_0^t (\cdot)d\xi$$

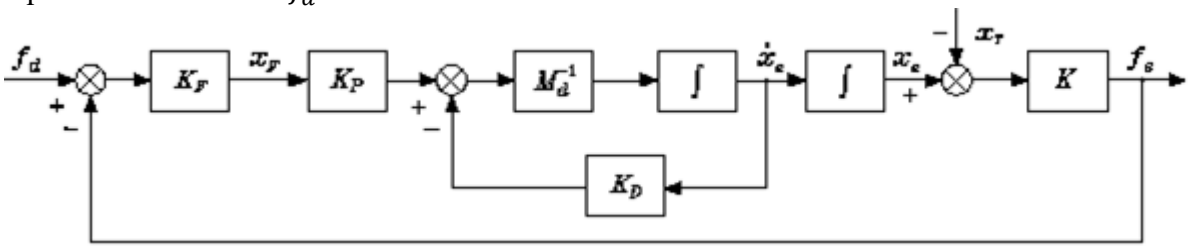
The compliance matrix is a diagonal matrix where its element characterize the control action. For obtain to steady state the condition $f_e = f_d$ it's necessary a PI action. In definite, we obtain to steady state that interacting force reach a desired value, however the final result is a closed loop system of third order. For this reason it's necessary a accurate choice of K_P, K_D, K_F, K_I and this problem is not trivial. In fact K_P and K_I influence the compliance behaviour and, to same time,K_F and K_I influence bandwidth and margins of stability.



B. Force Control with Inner Velocity Loop:
The control law and block scheme are following:

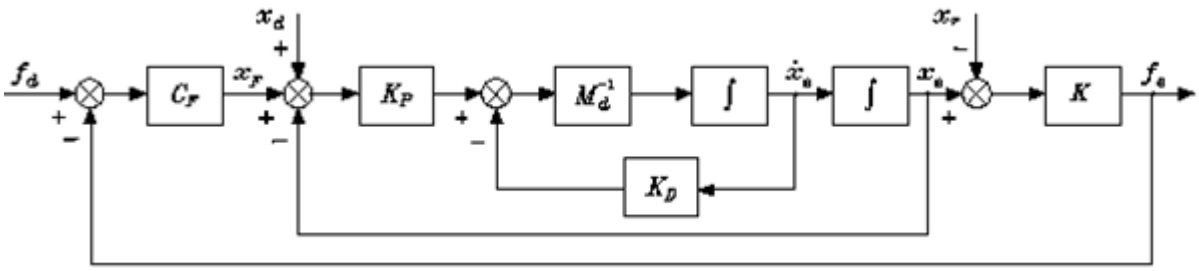
$$y = J^{-1}(q)M_d^{-1}\left(-K_D\dot{x}_e + K_P x_F - M_d\dot{J}(q,\dot{q})\dot{q}\right)$$

Also we choose $C_F = K_F$, so a pure proportional action on force error, in this way we obtain this relation: $x_F = K_F(f_d - f_e)$
In practice, respect to previously control, we open position loop and in this way between x_F and x_e born a integration relationship because x_F become a velocity reference. The final result is a closed loop system easer respect to previously case because is the second order, however without an integral action in presence of unmodelled dynamic effect interaction force f_e is not equal to desired force f_d.



C. Parallel Force/Position Control:
The control law and block scheme are following:

$$y = J^{-1}(q)M_d^{-1}\left(-K_D\dot{x}_e + K_P(\tilde{x} + x_F) - M_d\dot{J}(q,\dot{q})\dot{q}\right)$$



The previously control scheme don't allow to impose, together a desired force f_d, also a desired position x_d. In fact, if we don't plan correctly f_d this quantity is seen a velocity reference and this produce a derive phenomena of end-effector's position. For resolve this problem we can adopt the parallel force/position control charcaterized by presence of parallel action of position control $K_P\tilde{x} = K_P(x_d - x_e)$ and force control $K_P C_F(f_d - f_e)$. In practice, this control technique follow the same idea of impedence control, so a inner control loop with large proportional gain and outer loop with small proportional gain. In this way, steady state, equilibrium position satisfy this equation: $x_e = x_d + C_F(K(x_r - x_e) + f_d)$
And in particular:

a. In directions that not belongs to $\mathcal{R}(K)$ where the free motion is ammissible, we obtain that x_e reach the reference x_d: $x_e = x_d$
b. In directions that belong to $\mathcal{R}(K)$ where we have constraint motion, x_d is seen as an additional disturbance. However, if we use also an integral action, a steady state, are sure that $f_e = f_d$ but position error assume a value dependent by compliance properties of environment.

8.5 Motion Constraint

A real manipulation task is complex distributed phenomena because link local properties of contact surfaces and global properties between manipulator and enviroment. In these scenarios some directions we are subject to constraint about end-effector and in other directions constraints about interaction forces. Also in control point of view, where we choose pose and force reference, we should consider some problems:

- The pose and force reference must be compatible with constraints imposed by environment.
- It isn't possible to simultaneously impose a desired velocity and force on the environment along the same direction.

This scenario become more complex in case all these constraints aren't static.

Now for realize our interaction task we can think to define an analytical description of interaction forces, however this approach is very expensive for modelling. Another approach is assicurate that our control strategy realize a **Motion Contraint**, a motion that respect two class of constraints:

- Kinematics constraints, which regards to motion,
- Dynamics constraints, which regard to force and moments.

At this point, it's important to understand under what assumptions about the environment the control algorithm is based:

✓ **Rigid Enviroment**:

Manipulator and enviroment are perfectively rigid and this last impose *only kinematic constraints*, which can be represent by a set of algebric equation that should satisfy by end-effector. The **Holonomic Constraints** are kinematic constraints expressed in following form, where φ is a $m \times 1$ vector with m less than number n of manipulator's joints: $\boldsymbol{\varphi(q) = 0}$

Computing the time derivative of holonomic constraints, we obtain velocity constraint defined by **Constraint Jacobian** $\boldsymbol{J_\varphi(q)} \in R^{m\times n}$: $J_\varphi(q)\dot{q} = 0$

Thanks the constraint jacobian and kineto-static duality we can demonstrate follow relations. The **Selection Matries** $\boldsymbol{S_f, S_f^+, S_v, S_v^+}$ allow to describe completely the interaction task assigning desired motion for end-effector and contact forces/moments such that satisfy the constraints:

$$\boldsymbol{h_e = S_f(q)\lambda \in \mathcal{R}(S_f)} \quad \Leftrightarrow \quad \boldsymbol{\lambda = S_f^+(q)h_e}$$
$$\boldsymbol{v_e = S_v(q)v \in \mathcal{R}(S_v)} \quad \Leftrightarrow \quad \boldsymbol{v = S_v^+(q)v_e}$$

Where:

- The range $\mathcal{R}(S_f)$ of S_f correspond to subspace of force control.
- The range $\mathcal{R}(S_v)$ of S_v correspond to subspace of velocity control.

✓ **Compliance Enviroment**:

The manipulator is perfectively rigid and all compliance is localized in the enviroment. The contact force between manipulator and enviroment is approximates with an elastic linear model in form: $h_e = Kdx_{re}$

Notice that:

a. If the stiffness matrix K is positive definite, then we have a fully constrained motion in which the deformation of the environment coincides with the displacement of the end-effector.
b. If the stiffness matrix K is positive semi-definite, then we have a partially constrained motion.

In particular, in case of partially constraint motion, the K matrix can be calculated imposing that compliance enviroment are modelled as two rigid bodies S and R, with relative frames, link by a ideal six-dimensions spring. Supposing that the manipulator is link with S and assuming that tool frame coincide with frames solidal with S and R, the geometry of contact between S and end-effector allow to evaluate the selection matries S_f and S_v as relative force and velocity control subspace. In fact, supposing frictionless contact, the interaction force applied by the end-effector on body S: $h_e = S_f \lambda$

For conclude, it should be noted that the control should be able to manage acceptably even situations in which we do not have such hypotheses and which therefore represent non-ideal situations.

8.6 Natural Constraints and Artificial Constraints

An interaction task can be planning in terms of force h_d and velocity v_d for end-effector, but these constraints are valid only if:

✓ The vector h_d belongs to the force control subspace.
✓ The vector v_d belongs to the velocity control subspace.

These condition ca be guarantee if we assigne h_d and v_d through this relations using the selection matries and where we choose λ_d and ν_d: $h_d = S_f \lambda_d$ $v_d = S_v \nu_d$

Also, for many robotics tasks, we can define a **Constraint Frame $O_c - x_c y_c z_c$,** which is the cartesian frame eventually time-variant with we able to identify in easy way constraints imposed by the enviroment and assign reference force and velocity. Each axis of constraint frame correspond two degree of freedom:

- A grade associated to force and traslational velocity along axe's direction.
- A grade associated to torque and angular velocity aroud axe.

At this point, we can define two different type of constraint:

- **Natural Constraints**: are constraints imposed by the environment for each degree of freedom and depend to task's geometry. For example, natural velocity constraints allow the motion along or aroud an axe, and a force constraints avoid to apply a force o moment long or around an axe.
- **Artificial Constraints**: are constraints on value of variables not subject to natural constraits and, in fact, are only variables that the manipulator can control. These constraint are imposed by control strategy for execute task.

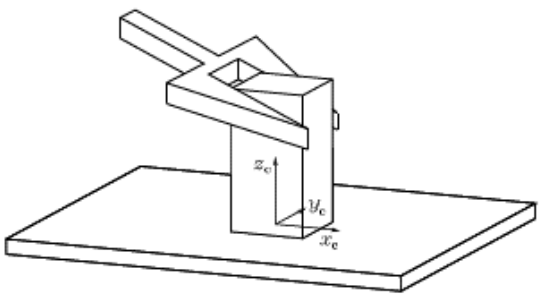
The two set of constraint are complementary because interess different variables for each degreed of freedom, in fact their union completely specify the task. We observe that, in case of compliance enviroment, for each degree of freedom where we have interaction we can choose if we want impose a force o velocity constraint. In any case, it's recommended follow this guideline:

A. High Stiffness – Rigid Environment: the velocity is the natural constraint and force is articial constraint.

B. Low Stiffness – Compliance Enviroment: the force is the natural constraint and velocity is articial constraint.

Notice that, in presence of friction, we have forces and moments different to zero also on degree of freedom associated a force natural constraint.

Esempio 1 – Sliding on Plane: the task consist to slide a prismatic object on the plane, which we suppose rigid and without friction. The constraint frame should be solidal to contact plane and orthogonal to this, for these reasons we choose it as in figure. We analize the natural and artificial constraints:



- **Natural Constraints**:
- $\dot{o}_z^c$: velocity constraint to prevent along linear velocity z_c;
- ω_x^c: velocity constraint to prevent long linear velocity along x_c;
- ω_y^c: velocity constraint to prevent long linear velocity around y_c;
- f_x^c: force constraint for prevent force along x_c;

- f_y^c: force constraint for prevent force along y_c;
- μ_z^c: force constraint for prevent moment around z_c.

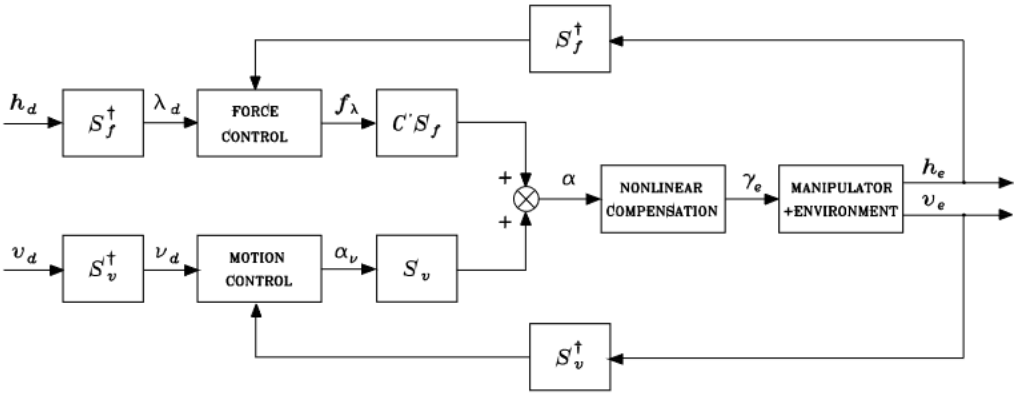
- **Artificial Constraints**:

- f_z^c: force constraint along z_c;
- μ_x^c: force constraint for prevent moment around x_c;
- μ_y^c: force constraint for prevent moment around y_c;
- $\dot{o}_x^c$: linear velocity constraint along x_c:
- $\dot{o}_y^c$: linear velocity constraint along y_c:
- ω_z^c: angular velocity constraint around z_c.

The dimension of the force control subspace is $m = 3$, so the velocity one has dimension $n - m = 6 - 3 = 3$.

8.7 Hybrid Force/Motion Control

The **Hybrid Force/Motion Control** *consists to implement a control that acts only on the variables associated with artificial constraints in terms of force, position and velocity, but has no effect on the variables associated with natural constraints to avoid undesired behaviour caused by the conflict between control and interaction with the environment.*



For the synthesis of hybrid control, we employ follow dynamic model writing respect acceleration $\dot{v}_e$ of end-effector:

$$B_e(q)\dot{v}_e + n_e(q,\dot{q}) = \gamma_e - h_e \qquad where: \begin{array}{l} \dot{v}_e = J(q)\ddot{q} + \dot{J}(q)\dot{q} \\ B_e = J^{-T}BJ^{-1} \\ n_e = J^{-T}(C\dot{q} + g) - B_e\dot{J}\dot{q} \end{array}$$

We will study hybrid control in two different cases. Each case has its own control law, however both are inverse dynamic control laws for obtain linearization and decoupling. Also these control laws are characterized by presence of auxiliary controls to impose a desired dynamic:

A. Compliance Environment: in this case we adopt follow inverse dynamics control law and blocks scheme:

$$\boldsymbol{\gamma_e = B_e(q)\alpha + n_e(q,\dot{q}) + h_e} \quad with: \ \boldsymbol{a = \dot{v}_e = S_v\dot{\mathrm{v}} + C'S_f\ddot{\lambda} = S_v a_v + C'S_f f_\lambda}$$

Where α is a new control input, in particular an acceleration, with we able to decoupling force control by velocity control. In fact:

a. If we want a desired velocity v_d, then we impose:

$$a_v = \dot{v}_d + K_{Pv}(v_d - v) + K_{Iv}\int_0^t \big(v_d(\tau) - v(\tau)\big)d\tau$$

b. If we want a desired force λ_d, then we impose: $f_\lambda = \ddot{\lambda}_d + K_{D\lambda}\big(\dot{\lambda}_d - \lambda\big) + K_{P\lambda}(\lambda_d - \lambda)$
Where $K_{Pv}, K_{Iv}, K_{D\lambda}, K_{P\lambda}$ are positive definite matrices.

B. Rigid Environment: In the case of rigid environment, the interaction force and moment can be written in the form: $h_e = S_f\lambda$

In this case we adopt follow inverse dynamics control law:

$$\boldsymbol{\gamma_e = B_e(q)(S_v a_v + \dot{S}v) + n_e(q,\dot{q}) + S_f f_\lambda}$$

Where α_v and f_λ are new control inputs with we able to decoupling force control by velocity control. In fact:

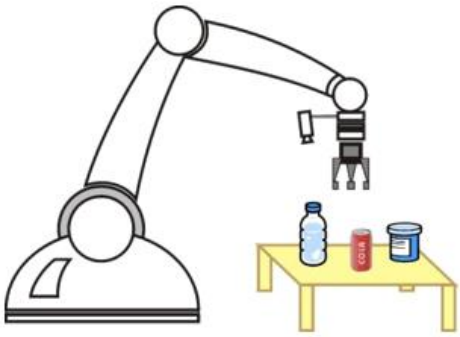
a. If we want a desired velocity v_d, then we impose:

$$a_v = \dot{v}_d + K_{Pv}(v_d - v) + K_{Iv}\int_0^t \big(v_d(\tau) - v(\tau)\big)d\tau$$

b. If we want a desired force λ_d, then we impose: $f_\lambda = \lambda_d + K_{P\lambda}(\lambda_d - \lambda)$
Where $K_{Pv}, K_{Iv}, K_{P\lambda}$ are positive definite matrices.

9. Visual Servoing

9.1 Characterization of Visual Servoing



The vision, together interaction control, is another foundamental element for any robotic system because allow to obtain geometrical and qualitative information about environment without a physical interaction. These information can be use for:

- Low level control, for example in control algorithm.
- High level control, for example in task planning.

Considering, for example, a robotic system constitueted by:

- A manipulator that have as end-effector a pliers.
- A camera that look the workspace.

The manipulator should grasp an object with a specific grasping configuration. This scenario can be resolved with two different approach:

A. Look-And-Move Approach: trajectory is planned using a motion control algorithm that employ a visual measurements to *open loop*. This approach is very easy to implement but it isn't robust to low position precision.

B. Visual Servoing Approach: visual measurements are used in a control loop algorithm for evaluate a current pose error between end-effector and object. This approach is more complex to realize but resolve the problem of robustness.

As we can understand, we are interested to second approach and for this reason we define formally this control problem.

The **Visual Control Problem** *consists to guarantee that end effector of manipulator, through a real-time visual measurements, reach a desired pose (constant or time variant) with respect to the observed object*.

Now, we observe that:

- The visual measurements obtained by vision system provides feature parameters of image.
- The task is defined in operational space in terms of end-effector's pose respect to observed object.

This observation identifies two different control approach that we will study:

A. Visual Servoing in The Operational Space: we use a visual measurements for evaluate pose of object respect to end-effector or viceversa.

B. Visual Servoing in The Image Space: it base on comparision of feature parameters of image, currente pose of end-effector and desired pose.

In really exists a third approach called **Hybrid Visual Control**, which want combine the advantages of previously approaches. In the sequel, we will study different approach of visual servoing in the operational space and image space. In particular, we suppose that:

- ✓ We use one camera.
- ✓ The object is fix respect to camera.
- ✓ The camera have a perfect calibration.
- ✓ The camera is eye-in-hand.
- ✓ The camera frame coincides with end-effector frame.

A foundamental aspect that characterize visual servoing, respect to motion and force control, is that the controlled quantities are not those directly obtained by the sensor, but are obtained after a sequence elaborations of images and computer vision. For this reason, we are interessed by following topics:

- **Camera Type**: fixed or variable resolution cameraca, CCD and CMOS technology. In following, we suppose that cameras are *black and white with fixed focal lenght.*
- **Number of Camera**: this aspect have a contribution on performance of visual control.
- **Image Processing Algorithm**: these algorithms want obtain numerical information from image called **Characteristic Parameters of Image**. These parameters can be used for estimate pose of object in the enviroment respect to camera or viceversa. About this, we can have:
  a. **Analitical Elaboration Algorithm**: based on measurement of a specific number of points.
  b. **Numerical Elaboration Algorithm**: based on the integration of the differential link between the speed of the camera in the operating space and the speed of the characteristic parameters in the image plane.

  It is possible to obtain more complete information thanks to stereo vision techniques when several images of the same scene from different angles are available.
- **Camera Calibration**: is a procedure for evaluate intrinsec parameters, that link quantitis measure in image plane and quantities referred to camera frame, and existrenc parameters that link quantities referred to camera frame and quantities solid to manipulator's frame.

9.2 Visual System

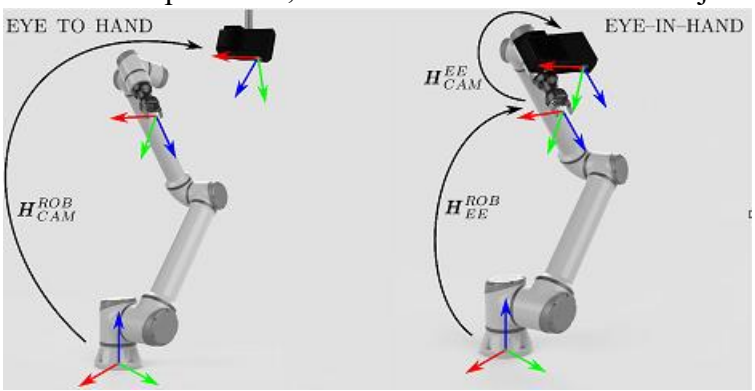
A **Visual System** is charcterized by two foundamental aspects::

A. Number of Cameras: a visual system can have one o more cameras for extract information from scene. The **3D Vision** o **Stereo Vision** is the capability of define the depth of scene using more images of same scene from different angles (similary to human's eyes). The stereo vision can be obtained with more cameras or, in opposite, with a single camera through a prior knowledge of geometrical characteristic of scene for estimate the depth. Typically, the system with single camera are preferred because a simpler and cheap but have different disvantages as low accuracy.

B. Position of Camera: for mono o multiple camera system we can adopt two different solutions:

A. Eye-To-Hand: the camera is installed on a fixed base. So the visualized object are referred to base frame of manipulator.
Advantage: filed of view of camera don't change during task, in this way we can assume measurement accuracy is constant.
Disvantage: in some applications as assembly, the manipulator can be a obstacle between camera and grasping object.

B. Eye-In-Hand: the camera is installed on the robot in particular on the wirst.
Advantage: this configuration resolve the disvantage of eye-to-hand configuration.
Disadvantage: we have a high variable accuracy because the visual of camera change with motion of robot. In particular, when the camera is near the object



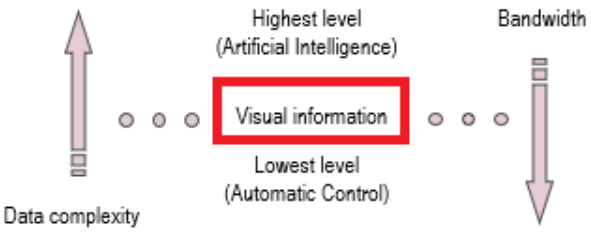
The choice of configuration respect to each other depends only by type of robot and type of task. Considering, for example, the case of a drone equipped with to a manipulator.

a) In configuration eye-in-hand the advantage is that object never covered by the camera. However, we have a variable accuracy cause to manipulator's motion
b) In configuration eye-to-hand the advantage is that a constant accuracy. However, the object may covered by the camera.

In this scenario, probably it's better the configuration eye-in-hand because in any case the accuracy of all system is variable to cause of drone's motion. In pratice, we can adopt a hybrid configuration in case of multi-camera system for obtain the better of two solutions. The robotic heads is a separate category because are characterized by stereo vision obtained from a couple of cameras mounted onf motorized mechanism which allow yaw, or pan, and pitch, or tilt, hence the name of **pan-tilt cameras**.

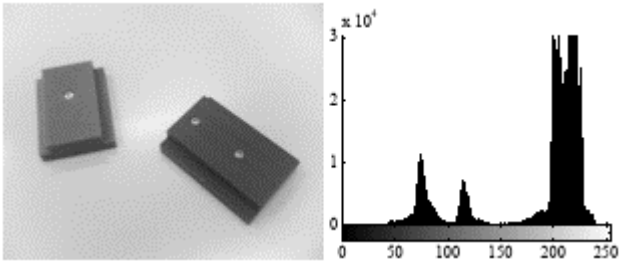
For conclude, in follow we consider only visual system *mono-camera* and in configuration *Eye-In-Hand*. The extension of algorithm in case of camera eye–to–hand and multicamera require some changes.

9.3 Image Processing



The visual information is the richest but at the same time most complex type of data to process because require a sequence of complex trasformation for used these informations in a control loop. These transformations have as fundamental scope to obtain **Feature Parameters** of image, in practice are numerical information that descrive in a complete and robust way the objects contained in image. However, if we want study these trasformations, they require the definition **Image Function**, a function defined on set of pixel and its components represent the values of one o more physical quantities associated to pixel. Let see some examples:

- The image function $I(X_I, Y_I)$ of black/white image is scalar and is associated to grey intesity. The minimum value is black and maximum value image is white. In this case is very useful the grey histogram where:
  - **Abscissa Axis**: we have the gray levels p $p \in [0,255]$.
  - **Ordinate Axis**: we have the frequency of occurrence of each level of gray p.



- The function $I(X_I, Y_I)$ of color image have three components $I_r(X_I, Y_I), I_g(X_I, Y_I), I_b(X_I, Y_I)$ associated to each primary color reg, green, blue.

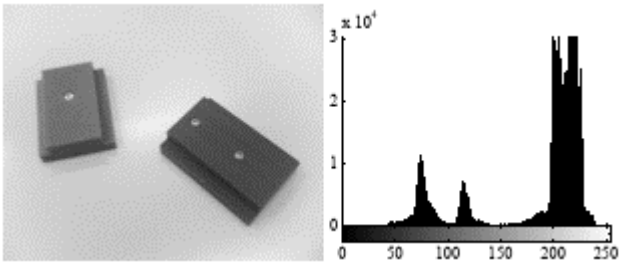
At this point, we can study these elaboration for obtain the feature parameters:

1. Image Segmentation:

Is the partition of image in a specific number of **Segments**, so parts that result to be homogenous respect to a one o more properties. Most typically segmentation approaches are:

A. Region-Based Segmentation: consist to identify the connected regions of image, in practice a set of pixels, that have a common characteristic. The binary segmentation is a particular type of segmentation based on regions where we define a threshold for distinguish the object respect to background. For example:

- All pixels with grey intensity bigger than threshold form the S_o object.
- All pixels with grey intensity smaller than threshold form the set S_b that represents background.



For this reason, is very important the choose of threshold for obtain the better results.

B. **Boundary-Based Segmentation**: consist to identify the boundaries of objects contained in image. If we consider a black/white image, we can see the boundaries of object a grey levels discontinuity. This type of segmentation is articulated in two steps:
   1. **Filtering Process**: this filtering is implemented via hardware for identify to local boundary of object, it is associated to different grey levels. Better we realize this step and more easy the second step.
   2. **Boundaries Rilevation**: using mathematical tool as gradient of image function or Laplacian of image function, we connect the local boundaries both them for obtain complete boundary of a particular object.

Notice that these segmentation apprach are complementary because an object in a image can be see or though its boundary or set of pixel inside close edge. For conclude, edge-based segmentation is less computationally expensive respect regions-base segmentation. However, region-based segmentation is preferred as it requires fewer memory accesses.

2. Image Interpretation:

Consist to evaluate the feature parameters of image starting from segments (regions or bords in base to segmentation type). These parameters can be used in visual servoing applications but, sometimes, require to calculate particular quantities.

- **Moments $\boldsymbol{m_{ij}}$**: are parameters defined on region $\mathcal{R}$ that characterize position, orientation and form of bidimensional object in this region. Fomally are defined as on the right:

$$\boldsymbol{m_{ij}} = \sum_{X_I, Y_I} \boldsymbol{I(X_I, Y_I) X_I^i Y_I^i}$$

- **Centroid** $(\bar{x}, \bar{y})$: are the coordinate that univocavely identify position of region in image. They are defined as respectively division between moment m_{10} and moment m_{00} and division between moment m_{01} and moment m_{00}:

$$\bar{x} = \frac{\boldsymbol{m_{10}}}{\boldsymbol{m_{00}}} \qquad \bar{y} = \frac{\boldsymbol{m_{01}}}{\boldsymbol{m_{00}}}$$

- **Central Moments $\boldsymbol{\mu_{ij}}$**: are the moments m_{ij} centered in the centroid of regione $\mathcal{R}$

$$\boldsymbol{\mu_{ij}} = \sum_{X_I, Y_I} \left(X_I^i - \bar{x}\right)^i \left(Y_I^i - \bar{y}\right)^i$$

For understand what mean these quantities we can do a mechanical comparison imaging that region R is a bidimensional rigid body where its density corresponds to lumen intensity:

- m_{00} corresponds to mass of R.
- The centroid $(\bar{x}, \bar{y})$ corresponds to center of mass.
- The matrix correspond to inertia tensor where on principale diagonal we find inertia moments:

$$I = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix}$$

- The eigenvalues of I are principal moments of inertia and eigenvectors to the principal axes of inertia.

Now moments and their parameters can be evulated starting from edge of generic object, but usually an observed object is built by a human as set of geometrical primitive elements (point, segments and so on). For example we can evaluated feature parameters thanks object's corner that we can see as feature points. This consideration can be extend to other geometrical primitive.

9.4 Pose Estimation

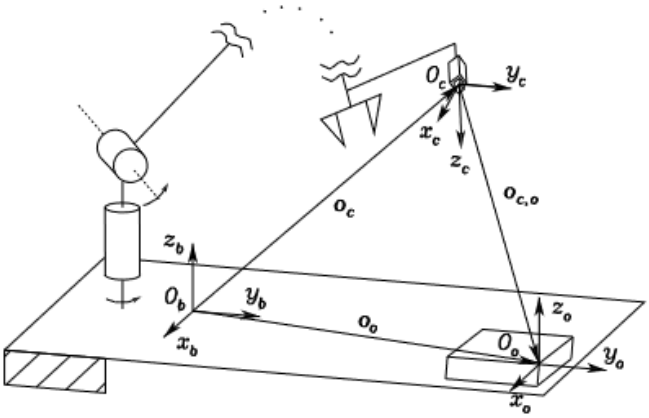
The visual servoing requires a mapping operation between characteristic quantities measured in image plane and quantities in operational space that define pose of object respect to camera or viceversa. Typically, it employs a differential mapping in terms of velocity for two reasons:

- ✓ Similary to invers kinematics of manipulator, we have a linear relation.
- ✓ It able to calculate the pose through a procedure of numerical integration.

For start, we define:

- $s = [X \quad Y]^T$: is column vector $k \times 1$ where its elements are characteristic parameters.
- $\bar{s} = [X \quad Y \quad 1]^T$: is homogeonous represenatation of s.

Notice that, we don't use coordinate pixel (X_I, Y_I) but normalized coordinates (X, Y). Considering the scenario in figure, where we have a manipulator with a camera in configuration eye-in-hand that see a particular object on the plane, our scope is *estimate pose (position and orientation) of object in image*, which is given from matrix T_o^c:



$$T_o^c = \begin{bmatrix} R_o^c & o_{co}^c \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} R_o^c & o_o^c - o_c^c \\ 0^T & 1 \end{bmatrix}$$

For resolve this problem we can adopt two different approachs:

A. Analytical Solution

Consist to define the **PnP Problem** (**Prespective-*n*-Point**), which is articuled in these steps:

1. We consider n points of object that we know to prior the position vector r_{oi}^o respect to object frame (for example thanks a CAD model): $r_{oi}^o = p_i^o - o_o^o$
2. Now, the projection of i-th point on image plane have coordinates $s_i = [X_i \quad Y_i]^T$. In this way, we can define the characteristic vector $s = [s_1 \quad s_2 \quad \dots \quad s_n]^T$.
3. At this point, if homogenous coordinates of each i-th point is: $\tilde{r}_{oi}^c = T_o^c \tilde{r}_{oi}^o$
   Then, the homogenous coordinates of projection of i-th point on image plane can be obtained though follow **Prospective Trasformation**: $\lambda_i \tilde{s}_i = \Pi T_o^c \tilde{r}_{oi}^o$
4. The PnP problem is a system of n equations of type see in step 3 and the unknown elements are components of matrix T_o^c.

Computing the solution of this problem is a difficult task because we can have multiple solutions. However, it can be simplify if the points belong to same plane. For example, we can suppose that the point belong to plane $z_0 = 0$ (last coordinate is zero). In this case, the prospective transformation becomes a **Planar Homography** of type:

$$S(\tilde{s}_i)[r_1 \quad r_2 \quad o_{co}^c]\begin{bmatrix} r_{xi} & r_{yi} & 1 \end{bmatrix}^T = S(\tilde{s}_i)H\begin{bmatrix} r_{xi} & r_{yi} & 1 \end{bmatrix}^T = S(\tilde{s}_i)[R_o^c \quad o_{co}^c]\tilde{r}_{oi}^o = 0$$

Where:

- r_1, r_2: are first and second column of rotation matrix R_o^c.
- r_{xi}, r_{yi}: are first components note qual to zero of vector.

At this point, we calculate the inverse of previously relation for evaluate the elements of matrix T_o^c. The problems of this method is that we need to know:

- The geometry of the object or to be more precise the position of the points with respect to the object frame.
- The intrinsic parameters of the camera to calculate the normalized coordinates s_i starting from those in pixels.
- The extrinsic parameters if we want to calculate the pose of the object with respect to the base frame, in the case of an eye-to-hand camera, or with respect to end-effector frame, in the case of an eye-in-hand camera.

B. Algorithmic Solution

For understand this strategy suppose to have a relative motion between object and camera, in this way the vector s of feature parameters is time variant: $s = s(t)$

Thanks to this, we can evaluate velocity vector of feature parameters $\dot{s} \in R^{k\times 1}$, which is defined as product between **Image Jacobian** $\boldsymbol{J_s} \in \boldsymbol{R^{k\times m}}$ and relative velocity v_{co}^c camera/object. Now we can write relative velocity in form contained in square brackets, in this way we can identify the **Interaction Matrix** $\boldsymbol{L_s} \in \boldsymbol{R^{k\times m}}$, which define the linear transformation between absolute velocity v_c^c of camera and velocity $\dot{s}$ on image plane in *hypothesis of static object* ($v_o^c = 0$):

$$\begin{aligned}\dot{s} &= J_s(s, T_o^c)v_{co}^c \\ &= J_s(s, T_o^c)[v_o^c + \Gamma(o_{co}^c)v_c^c] \\ &= J_s(s, T_o^c)\Gamma(o_{co}^c)v_c^c \\ &= L_s v_c^c\end{aligned} \qquad where: \quad \Gamma(\cdot) = \begin{bmatrix} -I & S(\cdot) \\ O & -I \end{bmatrix}$$

Typically, we prefer to use interaction matrix because it has an analythical expression easier respect to image Jacobian. How we said previously, image jacobian and interaction matrix are both $k \times m$ matries where:

- $k =$ number of feature parameters.
- $m =$ dimension of velocity vector v_c^c. Usually $m = 6$ if the object has free motion respect to camera but in opposite case $m < 6$.

Now, for estimation pose probably in relation $\dot{s} = J_s(s, T_o^c)v_{co}^c$ we are interested to v_{co}^c. Suppose a interaction matrix L_s is full rank, so also J_s is full rank, we can evaluate v_{co}^c respect to follow cases:

A. $\boldsymbol{k = m}$: we adopt basically the inverse relation: $v_{co}^c = J_s^{-1}\dot{s} = \Gamma L_s^{-1}\dot{s}$

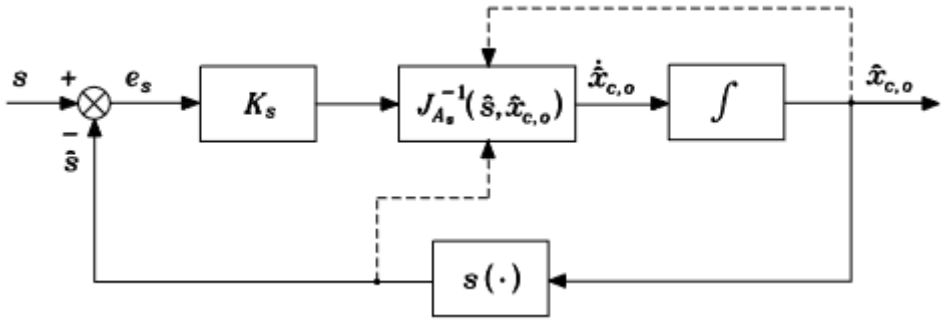
B. $\boldsymbol{k > m}$: the interaction matrix has more rows than columns and we can adopt a least-squares technique: $v_{co}^c = \Gamma(L_s^T L_s)^{-1} L_s^T \dot{s}$

C. $\boldsymbol{k < m}$: the interaction matrix has more column than row and this mean there are infinite movements of the object with respect to the camera which are *not associated* with a variation of the parameters. This implies that we have chosen a low number of parameters. This case for pose estimation is not of interest.

At this point we have all tools for estimate univocaly the pose of object respect to camera though an approach similary to resolution of inverse kinenamtics of manipulator. For this scope, we define two quantities:

- $x_{co} = [o_{co}^{c} \quad \phi_{co}]^T \in R^{m\times 1}$: is pose of object respect to camera with a minum number m of coordinates.
- $e_s = s - \hat{s}$: is the error between real and estimated vector of parameters. Notice that, we suppose $s = const$ and $\hat{s}$ function of current time of integration.

At this point, we implement this block scheme.



The evolution of error is the follow, where J_{As} is meaning similary to analythical jacobian of manipulator, so square and no singular: $\dot{e} = -\dot{\hat{s}} = -J_{As}\dot{\hat{x}}_{co} = -L_s \Gamma T_A \dot{\hat{x}}_{co}$

If we choose: $\dot{\hat{x}}_{co} = J_s^{-1} K_s e_s$

We arrive to linear error system: $\dot{e}_s + K_s e_s = 0$

Where is matrix K_s is positive definite, the error system converge asymptothically to zero with velocity depending on eighvalues of K_s.

However, the foundamental problem of this approach is given by relation between jacobian J_{As} and interaction matrix L_s. In fact, jacobian matrix has two singularities type:

- Singularity for representation of orientation.
- Singularity of interaction matrix.

This last singularity are more critical because depend on choice of feature parameters s. For obtain simultaiously a separation of singularities'effect and resolution of interaction matrix's singularities we can adopt follow steps:

1. We compute estimated velocity of object respect to camera:
2. After, thanks left pseudo-inverse of L_s, we compute estimated velocity of object respect to camera.
3. Integration for obtain estimated pose $\hat{x}_{co}$

$$\hat{v}_{co}^{c} = \Gamma (L_s^T L_s)^{-1} L_s^T K_s e_e$$
$$\downarrow$$
$$\dot{\hat{x}}_{co} = T_A^{-1} \hat{v}_{co}^{c}$$
$$\downarrow$$
$$\hat{x}_{co}$$

For conclude, the convergence can be proved with this Lyapunov function: $V = e_s^T K_s e_s / 2$

9.5 Camera Calibration

The **Camera Calibration** is procedure which we estimate follow parameters:

- **Intrinsic Parameters**: are the characteristic parameters of camera system, for example focal lenght, coordinates of optical centre, pixel skew and so on. These parameters characterize the matrix of intrinsic Ω.
- **Extrinsic Paramters**: these parameters characterize pose of camera respect to fixed base, in case of camera eye-to-hand, or respect end-effector's frame, in case of camera eye-in-hand.

There are different calibration techniques and we are studying is based on a procedure similar to the one used to estimate the pose of an object with respect to the camera. In particular, we analize two different cases:

A. Unknown Intrinsic Parameters: is the case more tipically in pratice and it's articulated in this steps:

1. We calculate the planar homography: $H' = \Omega H$
2. We calculate the intrinsic matrix Ω.
3. We calculate the extrinsic parameters thought method of resolution for PnP problem with n points belongs to same plane that we see in estimation pose. Resolving this problem we obtain the transformation matrix T_o^c and after, in base to configuration camera, it's possible evaluate the exitrinsic parameters thanks this relations:
   a. **Camera Eye-To-Hand**: $T_c^b = (T_o^c)^{-1} T_0^b$
   b. **Camera Eye-In-Hand**: $T_c^e = T_o^e (T_o^c)^{-1}$

   Where:
   - T_o^c: is soluzion matrix of PnP problem;
   - T_o^b: represent position and orientation of object frame respect to base frame.
   - T_o^e: represent position and orientation of object frame respect to end-effector frame.

B. Know Intrisic Parameters: is simpler case because we can start directly from third step of previously procedure.

However, this methods is only teorical because:

- Don't provides good solutions in presence of measurement noise.
- Don't consider lens distorsion.
- The accuracy can be improved employing more complex non linear model that considered distorsion.

From sperimetal point of view, calibration procedure showed require to use calibration plane where we can identify very easy a specific number of reference points with high accuracy. Tipically these calibration planes are checkerboard plane is used where the points of interest are the edges of the squares.

9.6 Stereo Vision

The bidimensional image doesn't provide any explicit information about depth or the distance observed object/camera. This information can be recovered with two way:

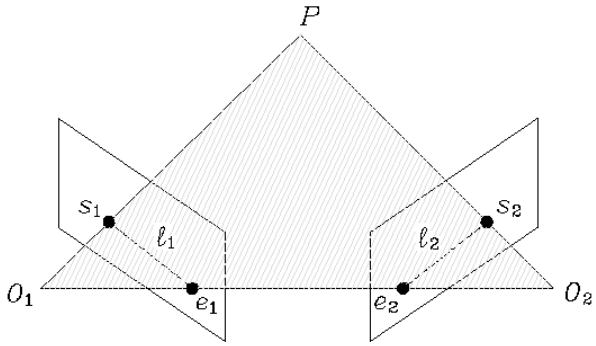
A. Indirect Way: we can use the geometrical model of object but this method assumes that the model is know;

B. Direct Way: we can use the **Stereo Vision**, then use two camera that capture two images of the same scene from two different points of view for computed information about depth. Alternatively we can use a single camera in movement for capture sequenatially the images of scene.

In the stereo vision we are two fundamental problem:

- **Correspondence Problem**: identification of the points of the two images that are projections of the same point of the scene. This problem hasn't a easy solution because it is based on:
  - Existence of details of the scene which appear to be similar in the different images.
  - Existence of geometric constraints between two images of the same point.

  The **Epipolar Constraints** are analytical form the geometric constraint existing between these points. Looking at the image on the side, the epipolar constraints impose that segment O_1P, O_2P and O_1O_2 must lie on the same plane.



- **3D Recostrution**: computation of the relative pose of the cameras (calibrated and not) and thus, starting from this pose, the position in the 3D space of the points of the observed object.

A second of the known information, in the context of 3D reconstruction we have several scenarios:

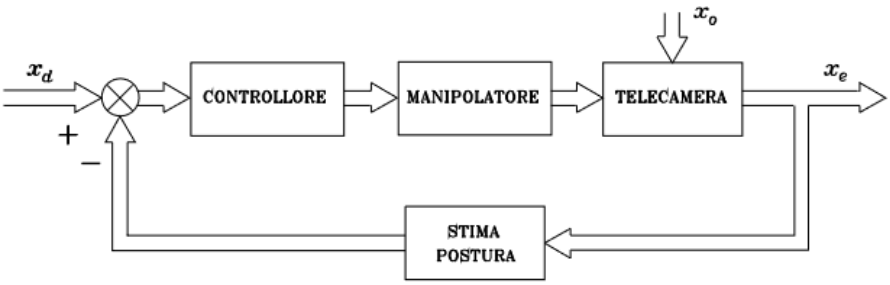
a. Triangulation: we know intrisec and existrisec parameters of two cameras. The 3D reconstruction problem is reduced to the use of triangulation to calculate the positions in the scene of the points projected on the two image planes.

b. Absolute Orientation: we don't know shape of rigid observed object. The calculation of the absolute orientation consists in the use of triangulation to calculate the variation in pose between the object and the system of cameras caused by the relative motion of the first with respect to the second.

c. Planar Homography: we have that the characteristic points of the observed object belong to the same plane. This observation characterized a constraint called planar homography. Then we have like constraints planar homography and epipolar constraint.

9.7 Visual Servoing in the Operational Space

The **Visual Control In The Operational Space** *resolve the visual control problem implementing a control action based on a error defined as difference between desired pose and real-time estimation pose calculated through visual measurements.*



The basic idea of visual control in operational space is:

1. **Definition of Desired Pose**: we impose the homogenous transformation matrix T_o^d that describe desired pose of object frame respect to camera frame. Thanks T_o^d, we can define the operational space vector $x_{do} \in R^{m\times 1}$: $\boldsymbol{T_o^d \rightarrow x_{do} \in R^{m\times 1}}$:
2. **Definition of Current Pose**: in practice we realize a pose estimation using the visual measurements and we can choose to adopt an analitycal solution or algorithmic solution. The result is the homogeneous transformation matrix T_o^c that describe current pose of object frame respect to camera frame. Thanks T_o^c, we can define the operational space vector $x_{co} \in R^{m\times 1}$: $\boldsymbol{T_o^c \rightarrow x_{co} \in R^{m\times 1}}$
3. **Definition of Relative Pose**: we evaluate the matrix T_c^d that describe the desired value to the relative pose of the object frame with respect to the camera:
$$T_c^d = T_o^d (T_o^c)^{-1} = \begin{bmatrix} R_c^d & o_{dc}^c \\ 0^T & 1 \end{bmatrix}$$
4. **Definition of Error**: we define the error vector in operational space: $\tilde{x} = -\begin{bmatrix} o_{dc}^d & \phi_{dc} \end{bmatrix}^T$
5. **Control Law**: at this point we can choose a particular control law for obtain that the space error $\tilde{x}$ tends to zero asymptotically.

Before studying the different control law choices, we want to look at the main advantages and disadvantages of visual servoing in operational space. In particular, the advantage is:

- ✓ The possibility of acting directly on operational space variables. This simplifies the choice of controller parameters and the resulting trajectories for the end effector are easily predictable.

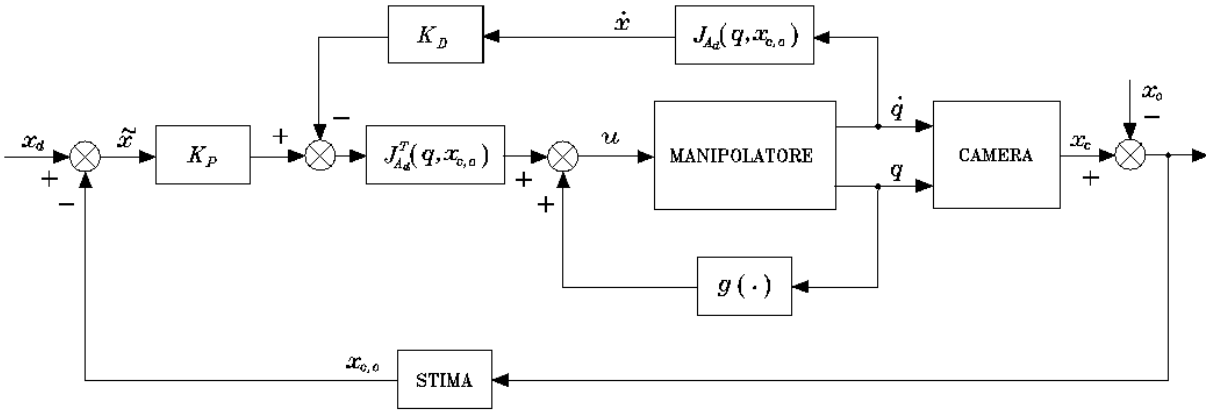
The disadvantages are:

- The object can fall out of the camera's field of view during the transient or due to a planning error. This produces an opening of the feedback loop and thus a loss of stability.
- This approach isn't robust to calibration errors. In fact, uncertain on camera's parameters influence error on pose estimation and the closed-loop control isn't able to reject these errors.
- If we use one camera is necessary to need know a geometric model of object for pose estimation.

The visual servoing in operationl space techiniques that we study are:

A. PD control with Gravity Compensation in Operational Space: is implement by follow block scheme and control law:

$$\boldsymbol{u = J_{A_d}^T(q,\tilde{x})K_p\tilde{x} - J_{A_d}^T(q,\tilde{x})K_D J_{A_d}(q,\tilde{x})\dot{q} + g(q)}$$



As we can observe is very similar to PD+ that we have study in motion control. In fact the block scheme have same structure, like the control law, and the stability demonstration can be realized with following Lyapunov function V. The only difference are:

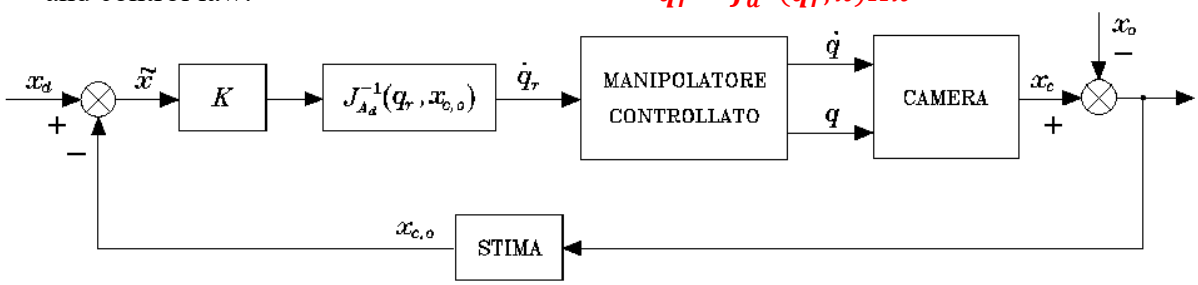
$$V = \frac{1}{2}\dot{q}^T B(q)\dot{q} + \frac{1}{2}\tilde{x}^T K_P \tilde{x}$$

- There is a camera that, thanks q and $\dot{q}$ like input, calculate the end-effector pose x_e.
- We have a different definition of error $\tilde{x}$.
- On feedback we have a block that compute estimation of relative position x_{co} camera/object. In PD+ study in motion control we had the direct kinematics $k(\cdot)$.

Notice that the sum blocks computing error $\tilde{x}$ and that computing the output of the controlled system have a purely conceptual meaning and *do not correspond to algebraic sums*.

B. Resolve-Velocity Control in Operational Space: is implement by follow block scheme and control law:

$$\boldsymbol{\dot{q}_r = J_a^{-1}(q_r,\tilde{x})K\tilde{x}}$$



How we can see, the control's name derives from control law based on computation of velocity $\dot{q}_r$ in operational space. Notice that the sum blocks computing error $\tilde{x}$ and that computing the output of the controlled system have a purely conceptual meaning and *do not correspond to algebraic sums*.

Most characteristic aspect about this control is that information deriving from visual measurements are computed to a frequency $f \leq f_r$, where f_r is frame rate of camera. Also the frame rate f_r, above all camera CCD, is as least of one order of magnitude lower than the typical frequencies f_c used for motion control of robot manipulators: $f_r \ll f_c$

Now, if we want realize a PD+ control law as in previously approach, for preserve the stability of the closed-loop system, we should set a small value for controller gains. However this condition mean impose a limit for convergence speed of error and disturbance rejection capability. For this reason, typically in visual control we assume that the motion control, for example in joint space, is ideal and so joint variables are about equal to reference of joint variables: $q(t) \approx q_r(t)$

We know that: $\dot{\tilde{x}} = -J_a(q, \tilde{x})\dot{q}$

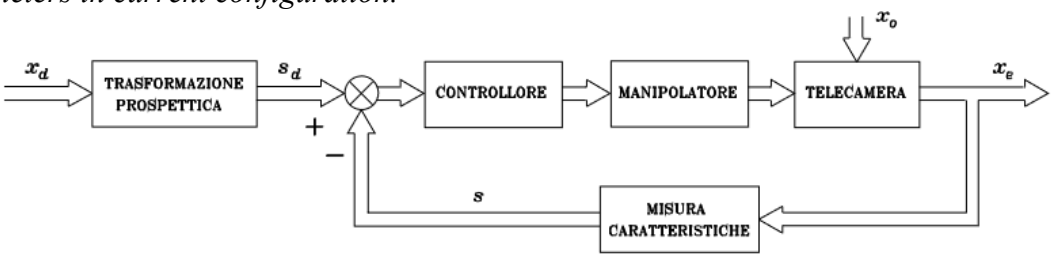
If we choose: $\dot{q} = \dot{q}_r = J_a^{-1}(q_r, \tilde{x})K\tilde{x}$

We obtain the following linear equation: $\dot{\tilde{x}} + K\tilde{x} = 0$

If matrix K is definite positive, then error tends asymptotically to zero with a convergence of exponential type and speed depending on the eigenvalues of matrix K (more big eigenvalues and more fast).

9.8 Visual Servoing in the Image Space

The **Visual Servoing in The Image Space** *resolve the visual control problem implementing a control action based on a error defined as difference between feature parameters in desired configuration* (calculated for example using a prospectic transformation) *and feature parameters in current configuration.*



The basic idea of visual control in image space is:

1. **Definition of Desired Feature Parameters**: using in input a desired configuration x_d and a prospective transformation we compute feature parameters s_d in x_d: $s_d = s(x_d)$
2. **Definition of Current Feature Parameters** We calculate, through feature measurements, the feature parameters s in current configuration: $s = s(x_e)$
3. **Definition of Error**: we calculate the error in image space e_s: $e_s = s_d - s$
4. **Control Law**: at this point we can choose a particular control law for obtain that the error in image space e_s tends to zero asymptotically. $e_s = s_d - s \to 0$

Before studying the different control law choices, we want to look at the main advantages and disadvantages of visual servoing in image space. The advantages are:

- ✓ We don't need pose estimation of observed object.
- ✓ It's possible to keep the object in the camera field of view during the motion.
- ✓ It's robust respect to calibration error of camera. In particular, this errors act only in the direct chain of control structure and uncertain on camera's parameters are balanced by control. The visual servoing in image space we don't realize a pose estimation because camera's parameters can be obtained directly by the camera without prospective transformation from desired pose x_d.
- ✓ If we use one o two camera we don't need to know a geometric model of observed object.

The disadavantages are:

- Due to the nonlinearity of the mapping between the image feature parameters and the operational space variables, singular configurations may occur, which cause instability or saturation of the control action.
- The end-effector trajectories cannot be easily predicted in advance and may produce collisions with obstacles or joint limits violation.

For conclude, the control approachs in image space that we will study are charcaterized by image Jacobian and interaction matrix dependent to z_c. This vector $z_c = [z_{c1}\ \ z_{c2}\ \ z_{cn}]^T$ contain the third coordinate z_{ci} of generic characteristic point respect to camera frame. We choose coordinate of these n points of object like feature parameters and in particular:

a) $n \geq 4$ if coplanar points.
b) $n \geq 6$ if non coplanar points.

The control techinique in image space that we study are:

A. PD control with Gravity Compation in Image Space:

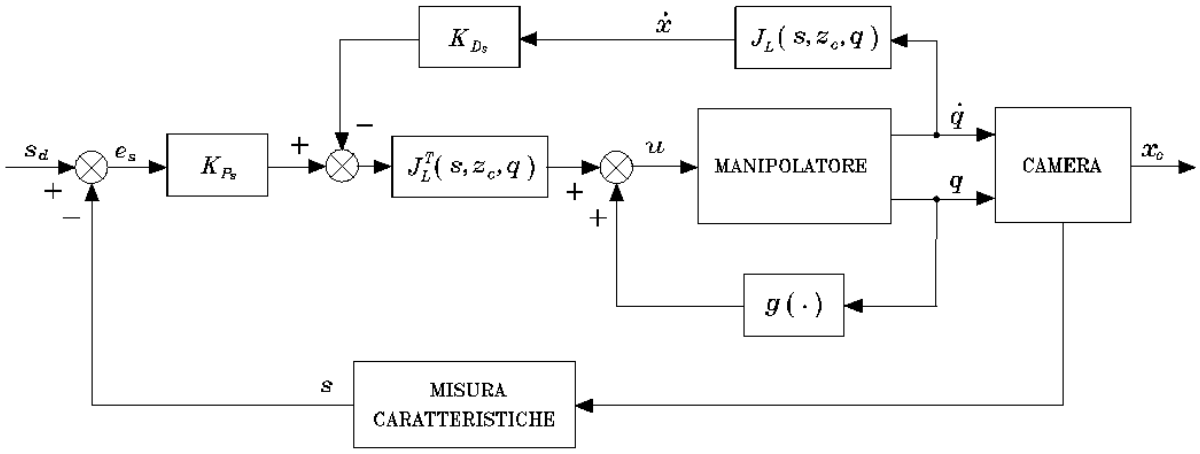
is implement by following block scheme and it has this control law:

$$\boldsymbol{u = J_L^T(s, z_c, q)K_p e_s - J_L^T(s, z_c, q)K_D J_L(s, z_c, q)\dot{q} + g(q)}$$

Where:

- First and second terms are PD action in image space;
- Third term is a nonlinear compensation action of gravitational forces in the joint space.

We can observe that this control law is equal to control law of PD+ in operational space, but it based on new errore in image space e_s.



Similary for dimostration of stability we use the function where $K_p \in R^{k\times k}$ is a matrix symmetric and definitive positive:

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

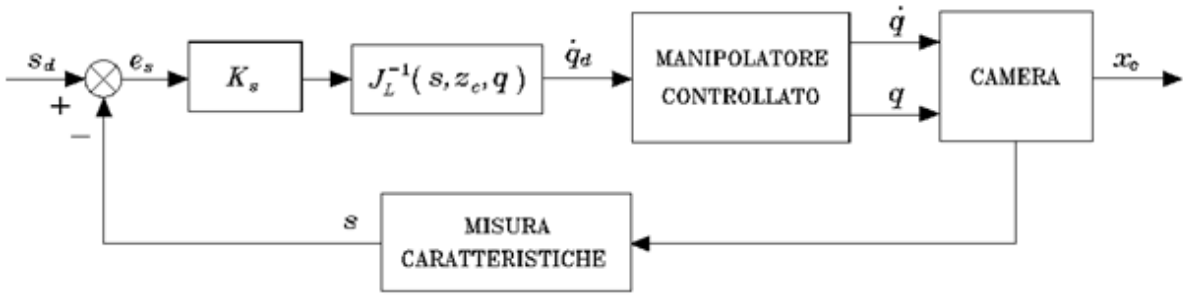
If we calculate $\dot{V}$ find that system arrive to equilibrium state if: $J_L^T(s, z_c, q)K_p e_s = 0$

The problem of all consideration to do is the calculate of z_c. In practice z_c is know with good approximation, but in alternative we can use a estimation of z_c. This choice have the problem that a estimation of interaction matrix and this is a complication for stability analysis.

B. Resolve-Velocity Control in Image Space.

is implement by following block scheme and it has following control law:

$$\boldsymbol{\dot{q}_r = J_L^{-1}(s, z_c, q_r)Ke_s}$$



This approach is similary to resolved-velocity control in operational space, in fact we assume that motion control, for example in joint space, is ideal and so joint variables are about equal to reference of joint variables : $q(t) \approx q_r(t)$

We know that: $\dot{e}_s = -\dot{s} = -J_L(s, z_c, q)\dot{q}$
If we choose: $\dot{q} = \dot{q}_r = J_L^{-1}(s, z_c, q_r)Ke_s$
We obtain the following linear equation: $\dot{e}_s + Ke_s = 0$
If matrix K is definite positive, then error tends asymptotically to zero with a convergence of exponential type and speed depending on the eigenvalues of matrix K (more big eigenvalues and more fast). Also the choice of K has effect on characteristic parameters trajectory, in fact if we choose K diagonal and with equal gains then projection point on image plane follow straight lines. The corresponding camera motion, however, cannot be easily predicted, because of the nonlinearity of the mapping between image plane variables and operational space variables.Notice that the sum blocks computing error e_s and that computing the output of the controlled system have a purely conceptual meaning and *do not correspond to algebraic sums*. Also this approach is based on calculate of J_L^{-1} and so it is sensible to the singularity problem of inverse Jacobian and interaction matrix. For this problem we can resolve it thank a calculation in two steps:

1. We calculate vector: $v_r^c = L_s^{-1}(s, z_c)K_s e_s$
2. We calculate the reference velocity in joint space:

$$\dot{q}_r = J^{-1}(q)\begin{bmatrix} R_c & 0 \\ 0 & R_c \end{bmatrix} v_r^c$$

Contririwise, in hypothesis of work aways singulary we can consider a number of parameters $k > m$, where m is the minimum number of characteristic parameters needed. In this case, the algoritm is modified using the pseudo-inverse: $v_r^c = (L_s^T L_s)^{-1} L_s^T K_s e_s$

In both cases, we need to know z_c that we can compute thanks to estimation of inverse or pseudo-invers of interaction matrix L_s. This consideration shows how control approaches in image space do not require knowledge of object geometry.

9.9 Hybrid Visual Control

The **Hybrid Visual Control** is based on definition of error where:

- Some components are defined in operational space. Thanks to these components we can assign a desired motion and have more control on trajectory during visual servoing.
- Some components are defined in image space. Thanks to these components we can avoid loss of visibility of characteristic points.

So this approach want take the advantages of visual control in operational space and visual control in image space. The hybrid visual control require a estimation of some variables in operational space. We can study this aspect by reasoning, for example, assuming that we have a object with a planar surface and on this surface select at least four feature points three to three not aligned. We can compute a planar omography H using the coordinates of these points in the camera image plane, both in the current and in the desired pose of the camera frame. Notiche that we can do this calculation without know currente and desired pose but only charactericit parameters vectors s and s_d. In particular condition, the expression of planar omography is:

$$H = R_d^c + \frac{1}{d_d} o_{cd}^c n^{dT}$$

Where:

- R_d^c: is rotation matrix between desidered orientation and current orientation of camera frame.
- o_{cd}^c: is position vector of the origin of the camera frame in the desired pose with respect to the current pose.
- d_d: is the distance between this plane and the origin of the camera frame in the desired pose.
- n_d: is the unit vector normal to the plane containing the feature points.

The quantities R_d^c, n^d, $(1/d_d)$, o_{cd}^c in the current camera pose, can be computed at each sampling time from matrix H.

9.10 Visual and Force Control

New interaction control approaches consist to combinate force control and visual servoing. Suppose to want control interaction of a manipulator with a rigid object of:

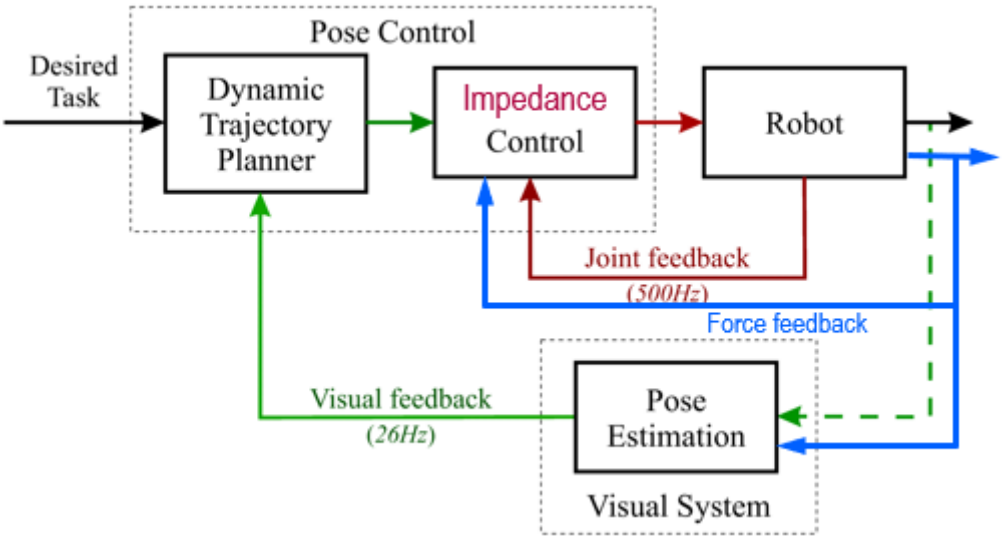
- ✓ Known geometry: $\varphi({}^{o}p) = 0$
- ✓ Unknow position and orientation: $\{o_o, R_o\} \rightarrow x_0$

The solution of this control problem is:

1. When robot is *far from object*, we adopt a position-based visual servoing where the relative pose of the robot with respect to the object is estimated recursively using only vision.
2. When robot is *in contact with object*, we adopt any kind of interaction control strategy that we have study (impedance control, parallel force/position control). In this case, the relative pose of the robot with respect to the object is estimated recursively using vision, force and joint position measurements.

A possible implementation of this solution is realized through the follow control strategy. The **Position-based Visual Impedence** is a control technique characterized by two different type of control loops:

- Visual feedback for gross motion that provides global information about environment, motion planning, obstacle avoidance.
- Force feedback for fine motion and to ensure that local constraints imposed by the environment are not violated.



10. Robots with Flexible Elements

10.1 Characterization

Typically we study kinematics, dynamics and control design of our robots with assumption that it is composed *by only of rigid bodies*. However, this hypothesis is only ideal and valid when there are *slow motion and small interacting forces*. In practice, mechanical flexibility in robot manipulators is present for two main reasons:

- Presence of transmission elements.
- Use of lightweight material and slender design.

These two factor of flexibility introduce new aspects which, if not considered, lead to a significant degradation of the manipulator's performance.

First of all, it's necessary understand how introduce flexibility in robot modelling and this can be realized in two different ways:

A. **Flexibility Concentrated in the Robot's joints**.
B. **Flexibility Distributed along the Robot's links**.

In particular, we are interested to first type of flexibility. The **Robots with Flexible Joints** are robots of which we suppose that all aspects of flexibility are concentrated in robot's joints. This modellation have different advantages:

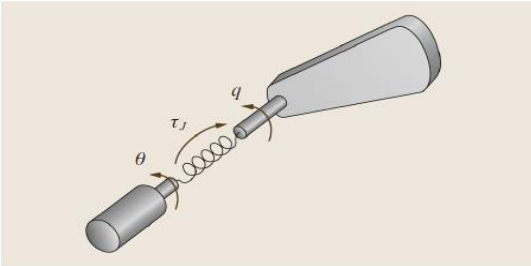
- ✓ It has a sense because in industrial practice we employ trasmissions, long shaft, harmonic drive for improve the dynamic efficiency and performances. But all this elements, when subjected to force/torque, are intrinsically flexible and introduce:
  - An oscillatory behavior typically with small amplitude and high frequency.
  - Different instabiliy form that involve contact not desired with the environment.
- ✓ We limit the complexity of motion equation.
- ✓ Motor torques and disturbance torques act work on the same joint axes, in this way though the control inputs that acting before source of flexibility we can ensure that output variables defined beyond the flexibility have a desired behaviour.

However, respect to rigid body case, there are two complicated factor for the control:

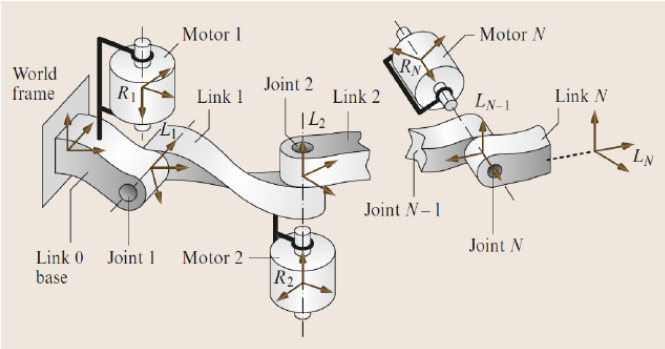
- Are requires twice the number of generalized coordinates to completely characterize the configuration of all rigid bodies;
- Are requires twice the number of sensors.

10.2 Dynamic Modelling

Now, we want study the dynamic model of robot with flexible joint for obtain a quantitively evaluation of vibratory effect that sum on rigid motion and understand if our control strategy, built in case of only rigid motion, continue to work in case of flexibility joints. We realize this study imposing this set of hypothesis:



- ✓ **Assumption 1**: joint deflections are small. This easy assumption mean that flexibility effects are limited in linear domain. In this way the flebility can be modelled as spring with stiffness $K_i > 0$.
- ✓ **Assumption 2**: the rotor of actuartors are modeled as uniform bodies and so their center of mass is collocated on the rotation axe. This condition imples that the inertia matrix and the gravity term in the robot dynamic model will be independent of the angular position of the motors. Also this assumption is very reasonable because is a property that we want for our electrical drive for improve its life.
- ✓ **Assumption 3**: each motor is located on the robot's arm in a position preceding the driven link (like in following image).



At this point suppose to have a manipulator with:

- $N + 1$ rigid bodies, so N links and fixed base.
- N flexbile joints that interconnect the links and actuating by N electrical drive.

In this situation we need to set $2N$ frames, for example through Denavit-Harteberng convections, because we have N links and N joints. For this reason, it's necessary a $2N$ generalized coordinates:

$$\Theta = \begin{pmatrix} q \\ \theta \end{pmatrix} \qquad where: \quad \begin{matrix} q \in R^{N\times 1} = position\ of\ links \\ \theta \in R^{N\times 1} = position\ of\ joints \end{matrix}$$

This choice of variables is quite convenient because:

- ✓ The model will be formally independent of the reduction ratios.
- ✓ These position variables will have a similar dynamic range.
- ✓ Robot kinematics only a function of link variables q.

At this point we follow the Lagrangian approach defining the Lagrangian function like difference between kinectic energy $\mathcal{T}$ and potential energy $\mathcal{U}$: $\quad \mathcal{L} = \mathcal{T}(\Theta, \dot{\Theta}) - \mathcal{U}(\Theta)$

Now we analize both energy:

- **Kinectic Energy**: this energy have a component relative to links and a component relative to rotors: $\mathcal{T}(\Theta, \dot{\Theta}) = \mathcal{T}_{link} + T_{rotor}$

  The links are rigid body and so we can use a quadratic form where there is the link inertia matrix $M_L(q)$ positive-definite symmetric. For rotor we can demonstrate that the kinectic energy is: $\mathcal{T}_{link} = \frac{1}{2}\dot{q}^T M_L(q)\dot{q}$

  $$\mathcal{T}_{rotor} = \frac{1}{2}\dot{q}^T[M_R(q) + S(q)B^{-1}S^T(q)]\dot{q} + \dot{q}^T S(q)\dot{\theta} + \frac{1}{2}\dot{\theta}^T B\dot{\theta}$$

  Where:
  - $M_R(q)$: rotor masses.
  - $S(q)$: inertial couplings between rotors and previous links.
  - B: constant diagonal inertia matrix.

  Summing up we calculate the total kinectic energy where we define **Total Inertia Matrix** $\boldsymbol{\mathcal{M}(q)}$:

  $$\mathcal{T}(\Theta, \dot{\Theta}) = \frac{1}{2}\dot{\Theta}^T \mathcal{M}(\Theta)\dot{\Theta} = \frac{1}{2}\begin{pmatrix}\dot{q}^T & \dot{\theta}^T\end{pmatrix}\begin{pmatrix}M(q) & S(q) \\ S^T(q) & B\end{pmatrix}\begin{pmatrix}\dot{q} \\ \dot{\theta}\end{pmatrix}$$

  With $M(q) = M_L(q) + M_R(q) + S(q)B^{-1}S^T(q)$.

- **Potential Energy**: this energy have a gravity component, for links e motors, and a elastic component for joints thanks assumption 1:

  $$\begin{aligned}\mathcal{U}(\Theta) &= \mathcal{U}_{grav}(q) + \mathcal{U}_{elast}(q - \theta) \\ &= [\mathcal{U}_{grav,link}(q) + \mathcal{U}_{grav,motor}(q)] + \left[\frac{1}{2}(q-\theta)^T K(q-\theta)\right]\end{aligned}$$

  Where $K = diag(K_1, \dots, K_N)$. Notice that for assumption 2, the term $\mathcal{U}_{grav}$ is independent of θ.

Finally we can describe the complete dynamic model with respectively a equation for link motion and a equation for joint motion. In particular:

- **Blu Terms**: is the inertial terms relative to matrix $\mathcal{M}(q)$.
- **Red Terms**: is the Coriolis and centrifugal terms.
- **Green Terms**: is the potential terms where $\tau_j = K(\theta - q)$

$$\begin{pmatrix}\boldsymbol{M(q)} & \boldsymbol{S(q)} \\ \boldsymbol{S^T(q)} & \boldsymbol{B}\end{pmatrix}\begin{pmatrix}\ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{\theta}}\end{pmatrix} + \begin{pmatrix}\boldsymbol{c(q,\dot{q}) + c_1(q,\dot{q},\dot{\theta})} \\ \boldsymbol{c_2(q,\dot{q})}\end{pmatrix} + \begin{pmatrix}\boldsymbol{g(q) + K(q-\theta)} \\ \boldsymbol{K(\theta - q)}\end{pmatrix} = \begin{pmatrix}\boldsymbol{0} \\ \boldsymbol{\tau}\end{pmatrix}$$

On the right-hand side of model there are all non-conservative generalized forces. For example, if we want consider viscous friction of transmission or spring damping, on the right-hand side of model appear the terms:

$$\begin{pmatrix}-F_q\dot{q} - D(\dot{q} - \dot{\theta}) \\ -F_\theta\dot{\theta} - D(\dot{\theta} - \dot{q})\end{pmatrix}$$

In case of contact with environment we have a product between traspost Jacobian and generalized forces acting from the environment on the robot: $\tau_{ext} = J^T(q)F$

This model have three fundamental properties:

- **Property 1**: all elements in the velocity-dependent terms are independent of motors position θ, to be computed via Christoffel symbols.
- **Property 2**: the dynamic model shares some properties of the rigid case, such as:
  - The model equations admit a linear parametrization in terms of a suitable set of dynamic coefficients, including joint stiffnesses and motor inertias, which is useful for model identification and adaptive control.
  - The Coriolis and centrifugal terms can always be factorized as following if the matrix $\dot{\mathcal{M}} - 2\mathcal{C}$ is skew-symmetric: $c_{tot}(\Theta, \dot{\Theta}) = \mathcal{C}(\Theta, \dot{\Theta})\dot{\Theta}$
  - For robots having only rotational joints, the gradient of the gravity vector g is globally bounded in norm by a constant.
- **Property 3**: when the joint stiffness is extremely large, $K \to \infty$, then $\theta \to q$ while $\tau_J \to \tau$. It is easy to check that the dynamic model collapsing into standard model of fully rigid robots, including links and motors.

In general, the link and motor equations in dynamic model are dynamically coupled through the elastic torque τ_j at the joints and inertial component of matrix $S(q)$. However, in presence of large reduction ratios, like $n_i \approx 100 - 150$, we can simply the dynamic model neglecting energy contributions due to the inertial couplings between the motors and the links. This is equivalent to considering the following simplifying assumption:

✓ **Assumption 4**: angular velocity of rotors due only to their own spinning.

Thanks this assumption in total kinectic energy we have that the matrix $S(q) = 0$ and dynamic model reduces to:

$$\begin{cases} \boldsymbol{M(q)\ddot{q} + c(q,\dot{q}) + g(q) + K(q-\theta) = 0} \\ \boldsymbol{B\ddot{\theta} + K(\theta - q) = \tau} \end{cases}$$

The main feature of this model is that:

- **Property 1**: The link and motor equations are dynamically coupled through only elastic torque τ_J.
- **Property 2**: the motor equation (the second) is fully linear.

For conclude, it is interesting to show that, when the joint stiffness K is relatively large but still finite, we can define a particular dynamics model characterized by a two-time-scale dynamic behavior. Notice that, for simplicity, we consider the reduced model. For show this characteristic we write K through a scale factor ε: $K = \hat{K}/\varepsilon^2 = diag(K_1, \dots, K_N)/\varepsilon^2$

The **Singular Perturbation Model** is a dynamics model for robot with flexible joint characterized by a slow subsystem (first equation) and a fast subsystem (second equation):

$$\begin{cases} \boldsymbol{M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau_J} \\ \boldsymbol{\varepsilon^2\ddot{\tau}_J = \hat{K}\{B^{-1}\tau - [M^{-1}(q) + B^{-1}]\tau_J + M^{-1}(q)[c(q,\dot{q}) + g(q)]\}} \end{cases}$$

The fast subsystem is obtained by deriving the slow subsystem twice. This model is a basis for **Composite Control Schemes**, where the control torque has the general form costitueted by two different terms: $\tau = \tau_s(q, \dot{q}, t) + \varepsilon\tau_f(q, \dot{q}, \tau_J, \dot{\tau}_J)$

Where:

- τ_s: is a slow action designed when neglecting joint elasticity.
- τ_f: is a fast action for locally stabilizing the fast flexible dynamics around a suitable manifold in the state space.

If we set $\varepsilon = 0$ then we obtain rigid robot model: $[M(q) + B]\ddot{q} + c(q, \dot{q}) + g(q) = \tau_s$

10.3 Inverse Dynamics

The inverse dynamics consist to use desired motion of robot for calculate the torque τ_d necessary to reproduce it. After we can use this torque as feedforward terms in a control for of trajectory tracking. Remember that, the inverse dynamics in case of rigid robots is a simple algebraic computation obtained by replacing the desired motion of the generalized coordinates in the dynamic model. However, in case of flexible robots there are addintion completixy and for this reason we articuled the study of inverse dynamics in two steps:

1. **Reduced Model**:

$$\begin{cases} M(q_d)\ddot{q}_d + n(q_d, \dot{q}_d) + K(q_d - \theta_d) = 0 \\ B\ddot{\theta}_d + K(\theta_d - q_d) = \tau_d \end{cases}$$

Now for compute τ_d we need θ_d and $\ddot{\theta}_d$. From link motion equation we can obtain θ_d, after it's necessary a twice differentiation respect to time of link motion equation for evaluate $\ddot{\theta}_d$:

$$M(q_d)\ddot{q}_d + n(q_d, \dot{q}_d) + Kq_d = K\theta_d$$
$$\downarrow Diff.\ respect\ to\ time \downarrow$$
$$M(q_d)q_d^{(3)} + \dot{M}(q_d)\ddot{q}_d + \dot{n}(q_d, \dot{q}_d) + K\dot{q}_d = K\dot{\theta}_d$$
$$\downarrow Diff.\ respect\ to\ time \downarrow$$
$$M(q_d)q_d^{(4)} + 2\dot{M}(q_d)q_d^{(3)} + \ddot{n}(q_d, \dot{q}_d) + \left[\ddot{M}(q_d) + K\right]\ddot{q}_d = K\ddot{\theta}_d$$

At this point substituting θ_d and $\ddot{\theta}_d$ in joint motion equation $B\ddot{\theta}_d + K(\theta_d - q_d) = \tau_d$, we obtain desired torque:

$$\boldsymbol{\tau_d = [M(q_d) + B]\ddot{q}_d + n(q_d, \dot{q}_d) + BK^{-1}\left[M(q_d)q_d^{(4)} + 2\dot{M}(q_d)q_d^{(3)} + \ddot{M}(q_d)\ddot{q}_d + \ddot{n}(q_d, \dot{q}_d)\right]}$$

Notice that, similarly to the rigid case, we can defined a recursive numerical Newton–Euler algorithm, which computes efficiently the inverse dynamics torque. But, in case of robots with elastic joints, the algorithm involves:

1) Forward recursion of motion variables up to the fourth differential order.
2) Backward recursion of second time derivatives of forces and torques.

2. **Complete Model**:

$$\begin{pmatrix} M(q) & S \\ S^T & B \end{pmatrix}\begin{pmatrix} \ddot{q} \\ \ddot{\theta} \end{pmatrix} + \begin{pmatrix} c(q, \dot{q}) + c_1(q, \dot{q}, \dot{\theta}) \\ c_2(q, \dot{q}) \end{pmatrix} + \begin{pmatrix} g(q) + K(q - \theta) \\ K(\theta - q) \end{pmatrix} = \begin{pmatrix} 0 \\ \tau \end{pmatrix}$$

For simplify and for not lost generality we suppose that matrix S is constant and for this reason both c_1 and c_2 vanish: $\quad S(q) = S \quad \rightarrow \quad c_1 = c_2 = 0$

$$\begin{cases} M(q_d)\ddot{q}_d + S\ddot{\theta}_d + n(q_d, \dot{q}_d) + K(q_d - \theta_d) = 0 \\ S^T\ddot{q}_d + B(q_d)\ddot{\theta}_d + K(\theta_d - q_d) = \tau_d \end{cases}$$

Now, contrariwise the reduced model, in this case we can't compute θ_d as function of $(q_d, \dot{q}_d, \ddot{q}_d)$ to due of desired motor acceleration $\ddot{\theta}_d$ on the left-hand side of link equation.

For resolve this problem we can use *upper triangular structure* of matrix S for build a recursive approach.

$$S(q) = \begin{pmatrix} 0 & S_{12} & \dots & S_{1N} \\ 0 & 0 & S_{23} & \vdots \\ \vdots & \ddots & 0 & S_{N-1,N} \\ 0 & \dots & 0 & 0 \end{pmatrix}$$

We can observe that the $N-th$ equation in previously form is independent of $\ddot{\theta}_d$ (because $S_N = 0$ and $0^T\ddot{\theta}_d = 0$). So we can define the N-th component position, which after double differentiation we can calculate $\ddot{\theta}_{dN}$. Now, the $(N-1)-th$ depend to $\ddot{\theta}_{dN}$ that we have calculate previously. In this way we can repeat the previously calculate. We can execute the same procedure backwards until we find the first component of θ_d and $\ddot{\theta}_d$:

$$\begin{cases} N-th\ equ. \ \rightarrow \ \theta_{dN} = f_N(q_d, \dot{q}_d, \ddot{q}_d) \ \rightarrow \ \ddot{\theta}_{dN} = f_N''\left(q_d, \dot{q}_d, \dots, q_d^{(4)}\right) \\ \qquad \circlearrowleft \boldsymbol{Substitute\ \ddot{\theta}_{dN}\ in\ (N-1)-th\ equ} \\ (N-1)-th\ equ \ \rightarrow \theta_{dN-1} = f_{N-1}\left(q_d, \dot{q}_d \dots, q_d^{(4)}\right) \ \rightarrow \ \ddot{\theta}_{dN-1} = f_{N-1}''\left(q_d, \dot{q}_d, \dots, q_d^{(6)}\right) \\ \qquad \circlearrowleft \boldsymbol{Substitute\ \ddot{\theta}_{dN-1}\ in\ (N-2)-th\ equ} \\ \qquad\qquad \dots and\ so\ on \dots \\ First\ equ. \ \rightarrow \ \theta_{d1} = f_1\left(q_d, \dot{q}_d \dots q_d^{(2N)}\right) \ \rightarrow \ \ddot{\theta}_{dN} = f_1''\left(q_d, \dot{q}_d, \dots, q_d^{(2(N+1))}\right) \end{cases}$$

At this point we can substitute θ_d and $\ddot{\theta}_d$ in this equation: $B\ddot{\theta}_d + K(\theta_d - q_d) = \tau_d$
For obtain nominal torque:

$$\boldsymbol{\tau_d = [M(q_d) + S^T]\ddot{q}_d + n(q_d, \dot{q}_d) + (B+S)\ddot{\theta}_d\left(q_d, \dot{q}_d, \dots, q_d^{(2(N+1))}\right)}$$

Per passare dal modello completo a quello ridotto si trascura l'effetto giroscopico (domanda fatta all'esame)

The introductions of **Dissipative Terms** in the inverse dynamics deserves some additional comments. The reduced model with dissipative terms is the following:

$$M(q_d)\ddot{q}_d + n(q_d, \dot{q}_d) + K(q_d - \theta_d) = 0$$
$$\downarrow \boldsymbol{Dissipative\ Terms} \downarrow$$
$$M(q_d)\ddot{q}_d + n(q_d, \dot{q}_d) + \left[\boldsymbol{(D + F_q)\dot{q}_d + D\dot{\theta}_d}\right] + K(q_d - \theta_d) = 0$$

Differenting these equations, we obtain a first-order linear asymptotically stable dynamical system with state $\dot{\theta}_d$ and forcing signal $w_d(t)$ that we will solve for a given initial condition $\dot{\theta}_d(0)$:

$$D\ddot{\theta}_d + K\dot{\theta}_d = M(q_d)q_d^{(3)} + \left[\dot{M}(q_d) + D + F_q\right]\ddot{q}_d + \dot{n}(q_d, \dot{q}_d)K\dot{q}_d$$
$$D\ddot{\theta}_d + K\dot{\theta}_d = w_d$$

Thanks for solution $\dot{\theta}_d(t)$ e the associated solution derivative $\ddot{\theta}_d$, we can evaluate the nominal torque: $\boldsymbol{\tau_d = M(q_d)\ddot{q}_d + B\ddot{\theta}_d + n(q_d, \dot{q}_d) + F_q\dot{q}_d + F_\theta\dot{\theta}_d}$

Where $\ddot{q}_d$ is the desired link trajectory should have a continuously differentiable. A similar procedure can also be applied to the complete model in the presence of spring damping. This discourse is considerably completed in the case of robots with flexible connections. These robots can have unstable internal dynamics and this is a problem for finding a feasible solution.

10.4 Regulation Control

The **Regulation Problem** *consist to conduce our robot with flexible joints in a constant configuration* q_d *(*$\dot{q}_d(t) = 0$*).* Now if $q_d = cost$, we obtain:

- A reference for position motor variables : $\theta_d = q_d + K^{-1}g(q_d)$.
- The static torque applied in steady state: $\tau_d = g(q_d)$.

First that we start the study of this control problem it's necessary observe a particular aspect. In case of robot with flexible joints, the feedback part of control law can depends of four variables for any joints:

- Position motor;
- Position link;
- Velocity of motor;
- Velocity of link.

For this reason, it's necessary position and velocity measurements. Notice that, typically the robot are projected not considering the flexible joints and so they can have only position sensor. However, this condition is not a big problem because we can calculate velocity through a numerical differentiation thanks high definition of position measurements. At this point, we can see different control approaches:

A. PD Control: we employ a PD control law, in fact there are K_P and K_D positive-definite gain matrices, without any gravity compensation: $\boldsymbol{\tau = K_P(\theta_d - \theta) - K_D\dot{\theta}}$

From relation $\theta_d = q_d + K^{-1}g(q_d)$ we obtain that the reference value for the motor position is: $\theta_d = q_d$

The control law τ is globally asymptotically stabilizes the desired equilibrium state $(q = \theta = q_d, \dot{q} = \dot{\theta} = 0)$. This can be show through a Lyapunov's theorem choosing the following candidate Lyapunov function. It is sum of total energy of the system, kinetic plus elastical potential, and energy of control energy, a virtual elastical potential:

$$\boldsymbol{V = \frac{1}{2}\dot{\Theta}^T\mathcal{M}(\Theta)\Theta + \frac{1}{2}(q-\theta)^T K(q-\theta) + \frac{1}{2}(\theta_d - \theta)^T K_P(\theta_d - \theta) \geq 0}$$

This function have a $\dot{V} = -\dot{\theta}^T K_D \dot{\theta} \leq 0$ and if there are dissipative terms is only negative semi-definite. For solve this complication and demonstrate the global stability it's necessary completed Lyapunov with La Salle's principle.

B. PD with Constant Gravity Compensation: if we consider the presence of gravity it's necessary introduce an additional compensation terms: $\boldsymbol{\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + g(q_d)}$

The control law τ is globally asymptotically stabilizes the desired equilibrium state $(q = \theta = q_d, \dot{q} = \dot{\theta} = 0)$. This can be show through a Lyapunov's theorem choosing the following candidate Lyapunov function given by Lyapunov function V see in PD control plus other terms:

$$\boldsymbol{V_{g1} = V + \mathcal{U}_{grav}(q) - \mathcal{U}_{grav}(q_d) - (q - q_d)^T g(q_d) - \frac{1}{2}g(q_d)^T K^{-1} g(q_d) \geq 0}$$

Another method for evaluate the stability is adopt another assumption:

- **Assumption 5**: the lowest joint stiffness is larger than the upper bound on the gradient of gravity forces. Formally: $\min_{i=1,\dots,N} K_i > a$

Where the positive constant a can be obtained by assumption 2 impose initially. The assumption 2 says that the actuators' rotors are modeled as uniform bodies having their center of mass on the rotation axis. This involve that for a robots with rotational joints, elastic or not, the norm of gradient of gravity forces is limited by a positive constant a. A sufficient condition guaranteeing that $q = q_d, \theta = \theta_d = q_d + K^{-1}g(q_d), \dot{q} = \dot{\theta} = 0$ will be the unique globally asymptotically stable equilibrium for the system under the control law τ (in red) is that the lowest eightvalues of matrix in square brackets is larger than the upper bound on the gradient of gravity forces:

$$\left\| \frac{\partial g(q)}{\partial q} \right\| \leq a \quad \forall q \in R^N$$

$$\lambda_{Min} \begin{bmatrix} K & -K \\ -K & K + K_P \end{bmatrix} > a$$

Taking into account the diagonal structure of K and KP, and thanks to assumption 5, it is always possible to fulfill this condition by increasing the smallest proportional gain in the controller.

C. PD with Online Gravity Compensation: we employ a PD control law with a gravity compasation for all configuration during motion based on quantity **Gravity-Biased** $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{K}^{-1}\boldsymbol{g}(\boldsymbol{q_d})$ which is the modification of the measured motor position θ caused by vector gravity.

$$\boldsymbol{\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + g(\tilde{\theta})}$$

Notice that, the term $g(\tilde{\theta})$ provide an almost perfect gravity compensation during the motion and a perfect gravity compensation at steady state. Eventually we can adopt a easier control law characterized by a quasi-static estimation $\bar{q}(\theta)$ of q. So we can realize this control law:

$$\boldsymbol{\tau = K_p(\theta_d - \theta) - K_D\dot{\theta} + g(\bar{q}(\theta))}$$

The control law τ is globally asymptotically stabilizes the desired equilibrium state $(q = \theta = q_d, \dot{q} = \dot{\theta} = 0)$. This can be show through a Lyapunov's theorem choosing the following candidate Lyapunov function given by Lyapunov function V see in PD control plus other terms:

$$\boldsymbol{V_{g2} = V + \mathcal{U}_{grav}(q) - \mathcal{U}_{grav}(\tilde{\theta}) - \frac{1}{2}g(q_d)^T K^{-1} g(q_d) \geq 0}$$

Or, in alternately, thought method equal to PD with constant gravity compensation.
For conclude, these control laws realize a compliance control in the joint space with only motor measurements. Can be extended to operational space control via Jacobian transpose.

D. Full State Feeback: we employ a control law based on full set of measurements of the robot state. For this scope, we exploit the presence of joint torque sensor for implement a control law with spring damping as a dissipation effect. Under this consideration we propose this control law: $\boldsymbol{\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + g(q_d) + K_T[g(q_d) - \tau_J] - K_s\dot{\tau}_J}$

Where:

- $\boldsymbol{K_P = BB_\theta^{-1}K_{P\theta}}$.
- $\boldsymbol{K_D = BB_\theta^{-1}K_{D\theta}}$.
- $\boldsymbol{K_T = BB_\theta^{-1} - I}$.
- $\boldsymbol{K_S = (BB_\theta^{-1} - I)DK^{-1}}$.

Notice that the control law can also be rewritten in terms of $(\theta, q, \dot{\theta}, \dot{q})$ if the torque sensor is not avaible.

10.5 Trajectory Tracking

The **Trajectory Tracking Problem** *consist to realize a controller for able the robot to track a particular reference*. For resolve this control problem we can use two different approach:

A. Feeback Linearization:

The feedback linearization control is non linear control techinque where we synthetize a control law for obtain a full linear system and, in this case, a decoupled behaviour for all N joints. The control law is: $\boldsymbol{\tau = BK^{-1}\left[M(q)v + a\left(q,\dot{q},\ddot{q},q^{(3)}\right)\right] + [M(q) + B]\ddot{q} + n(q,\dot{q})}$

Where:

- $\boldsymbol{a\left(q,\dot{q},\ddot{q},q^{(3)}\right) = \dot{M}(q)\ddot{q} + 2\dot{M}(q)q^{(3)} + \dot{n}(q,\dot{q})}$.
- $\boldsymbol{v = q^{(4)}}$: chains of 4 input–output integrators from each new input v_i to each link position output q_i.

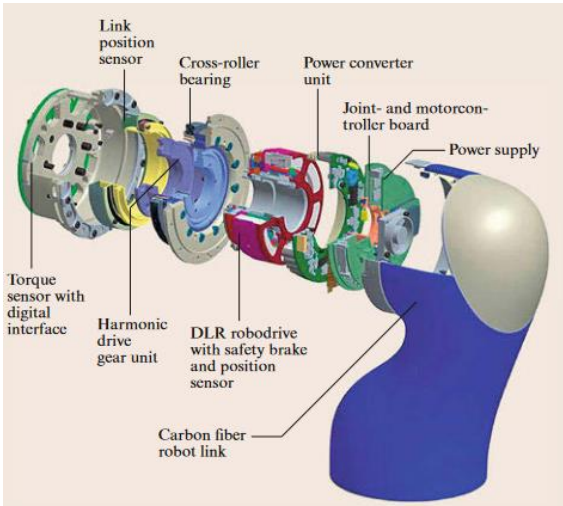
For understand this control law, it's important observe that the motion of a link attached to an elastic joint, due to the inertial couplings present between motors and links, is already affected to the second differential order by torques that have origin in other elastic joints. Now, the control law in practice stiffens the robot's dynamics and the addition of integrators delays the influence of inertial couplings to the fourth differential order. In conclusion, we obtain two results:

✓ An exact linearization of the extended system (robot+controller) which is decoupled.

✓ The tracking error is forced to be globally exponentially stable with convergence rate specified through the choice of scalar feedback gains in the controller.

Notice that the control law is expressed respect linearizing coordinates $\left(q,\dot{q},\ddot{q},q^{(3)}\right)$. Therefore, accelleration $\ddot{q}$ e jerk $q^{(3)}$ have two problem:

- Their direct measure are impossible to obtain with currently avaible sensors;
- If we want use multiple numerical differentiation of position in real-time there is many noise.



This complication can be resolve observing that following set of coordinate are equivalent:

- $\left(q,\dot{q},\ddot{q},q^{(3)}\right)$.
- $\left(q,\tau,\dot{q},\dot{\tau}_J\right)$.

Thanks the lastest technology with joint torque sensors, we can evaluate very easly:

❖ Rotor position θ, from which we obtain q and by differentiating we obtain $\dot{q}$.

❖ Torque τ, from which thanks relation $\tau = \theta - K^{-1}\tau_j$ we obtain τ_j. After with diffrentiating we calculate $\dot{\tau}_J$.

The main advantages is impose a globally enforcing a linear and decoupled behavior to the trajectory error dynamics. The principle disadvantages is gives rise to a rather complex nonlinear control law.

B. Linear Control Design:

We can adopt this approach in presence of a sufficiently smooth reference link trajectory $q_d(t)$ where we can realize follow control law costitued by two control actions:

- PD action, standard linear feedback, that locally stabilizes the system around reference state trajectory.
- The feedforward torque τ_d what matain the robot along the desired motion when error is vanished.

$$\boldsymbol{\tau = K_P(\theta_d - \theta) + K_D(\dot{\theta}_d - \dot{\theta}) + \tau_p}$$

The advantages of this approach are:

✓ The implementation is more easy.

✓ The computational cost in real-time is very low than approach with feedback linearization.

The main disadvantage is that the linear control design guaranteed only locally convergence.

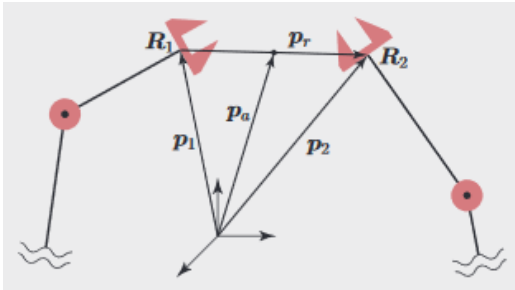
11. Cooperative Manipulation

11.1 Dual Manipulator System

Different applications require the adoption of multiple robots and in this cases are necessary particular properties as:

- A correct task space formulation;
- A coordination about relative motion between robot manipulators and mobile robots;
- Avoiding undesired collisions.

Suppose to have the dual manipulator system (as in figure) with we want to realize our task. A first approach consist to control *each robot individually*, in this way the variables of cooperative task will be final pose of end-effector of each manipulator. However, this approach has two problems:



- The final user should manage manually the cooperation between two manipulators.
- We do not efficiently employ the presence of two manipulators.

In this context, it's better specify control two robot as a single system and for this it's necessary a different formulation of task. For this reason, we define two different frames:

- **Absolute Frame**: this frame describes the task as a composition of the poses of manipulator's end-effectors. This frame is characterize by a position vector $p_a = (p_1 + p_2)\,/2$ e rotational matrix R_a.
- **Relative Frame**: this frame completes the descrition of task adding pose of a robot respect to each other. Also this is important because exist infinite end-effector configurations giving the same absolute pose. This frame is charcterized by a position vector $p_r = p_2 - p_1$ e rotational matrix ${}^1R_r = {}^1R_2$.

Definitely the cooperative task space is define by variables $p_a, R_a, p_r,\ {}^1R_r$. Two examples are the following:

- **First Example**: move tightly grasped rigid object. In this case we should assign a trajectory to p_a and R_a, while keeping p_r and R_r^1 constant.
- **Second Example**: stretch, bend or shear object. In this case we should assign a trajectory in terms of p_a, R_a, p_r, R_r^1.

The fundamental advantage of this formulation is not require any special assumption on the nature of the manipulated object and/or the grasp (For examples rigid o flexible object). Also this topic can be deal in terms of twist, in fact we can define:

- **Absolute Velocities**:

$$v_a = \frac{1}{2}(\dot{p}_1 + \dot{p}_2) \qquad \omega_a = \frac{1}{2}(\omega_1 + \omega_2)$$

- **Relative Velocities**:

$$v_r = \dot{p}_2 - \dot{p}_1 \qquad \omega_r = \omega_2 - \omega_1$$

This cooperative task space formulation can be adopted to solve the inverse kinematics problem for a dual manipulator system. Consider a system of two cooperative 6-DoF manipulators, the **Jacobian Inverse CLIK Algoritm** is given by following equation where indicate with t_d the desired twist, which is composed by absolute velocity t_{ad} and relative velocity t_{rd}:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})(\boldsymbol{t_d} + \boldsymbol{Ke})$$

Where:

$$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \qquad J = \begin{pmatrix} J_a \\ - \\ J_r \end{pmatrix} = \begin{pmatrix} J_1/2 & | & J_2/2 \\ - & + & - \\ -J_1 & | & J_2 \end{pmatrix}$$

$$t_d = \begin{pmatrix} t_{ad} \\ t_{rd} \end{pmatrix} = \begin{pmatrix} v_{ad} \\ \omega_{ad} \\ \hline R_a^a v_{rd} + S(\omega_a) R_a^a p_{rd} \\ {}^1\omega_{rd} \end{pmatrix} \qquad e = \begin{pmatrix} e_a \\ e_r \end{pmatrix}$$

For verify red relation we start from differential kinematics equation evaluated for each robot of dual robotic system: $t_i = J_i(q_i)\dot{q}_i \qquad with:\ i = 1,2$

Abolute twist t_a is given by an abosulte linear velocity v_a and a absolute angular velocity ω_a that we know their expression. These twists are associate a joint velocity $(\dot{q}_1 \ \ \dot{q}_2)^T$ thanks the absolute Jacobian $J_a \in R^{6\times 12}$:

$$t_a = \begin{pmatrix} v_a \\ \omega_a \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\dot{p}_1 + \dot{p}_2) \\ \frac{1}{2}(\omega_1 + \omega_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}J_1 & \frac{1}{2}J_2 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = J_a(q_1, q_2) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

Similarly we can build relative twist vector with expression of v_r and ω_r. The relative jacobian $J_r \in R^{6\times 12}$:

$$t_r = \begin{pmatrix} v_r \\ \omega_r \end{pmatrix} = J_r(q_1, q_2) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

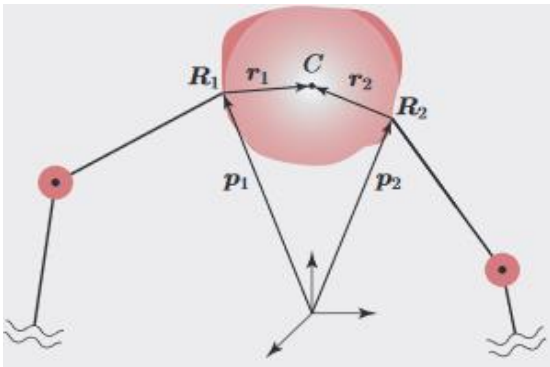
At this point, remember the standard inverse jacobian CLIK algorithm, renaming $\dot{x}_d$ as $t_d = (t_a \quad t_r)^T$, we obtain the red relation:

$$\dot{q} = J^{-1}(q)(\dot{x}_d + Ke) \quad \rightarrow \quad \dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})(\boldsymbol{t_d} + \boldsymbol{Ke})$$

11.2 Object Manipulation

Consider now a system of two cooperative robots manipulating a common object. Let C be a fixed point of the object (for example the center of mass of the object) and p_c its vector position of base frame. Assuming that the object is rigid and tightly grasped by each manipulator, p_c can be expressed as: $p_1 + r_1 = p_2 + r_2 = p_C$

Where r_i with $i = 1,2$ denotes the distances of C from each point of contact. The our goal is identify wrench and twist of manipulated object.



- **External Wrench $\boldsymbol{w_E}$**: represents the wrench that cause the motion of object. In formula, this quantity is given by sum of wrenchs $w_{i,obj}$ provide by end-effectors of two manipulators or, in a compact form, as product between **Grasp Matrix $\boldsymbol{G} = (\boldsymbol{G_1} \quad \boldsymbol{G_2}) \in \boldsymbol{R^{6\times 12}}$** that describe the geometry of the grasp and vector $w = (w_1^T \quad w_2^T)^T$:

$$\boldsymbol{w_E} = \boldsymbol{w_{1,obj}} + \boldsymbol{w_{2,obj}} = \boldsymbol{Gw} \qquad with:\ w_{i,obj} = G_i(r_i)w_i = \begin{pmatrix} I & O \\ -S(r_i) & I \end{pmatrix} w_i \qquad i = 1,2$$

- **Internal Wrench $\boldsymbol{w_I}$**: represent mechanical stress and doesn't contribute to the object motion. For compute this wrench it's necessary invert the relation of external expression w_E: $w = G^+ w_E + V w_I$

  In this expression we can see the pseudoinverse G^+ of G and the full-column rank matrix $V \in R^{12\times 6}$ that spanning the null space of G. In fact Vw_I is the wrench lying in the null space of G and so w_I is the internal wrench.
- **Absolute Twist $\boldsymbol{t_e}$**: identify this twist we start from the kineto-statics duality where the twist t_i at each end-effector is:

$$t_i = G_i^T(r_i)t_{i,obj} = \begin{pmatrix} I & S(r_i) \\ O & I \end{pmatrix} t_{i,obj} \qquad i = 1,2$$

  If we invert this equation obtain: $t_{i,obj} = G_i^{+^T}(r_i)t_i$

  The resulting absolute twist on the object is the sum of each absolute twist t_1 and t_2 associated to each manipulator:

$$t_e = G^{+^T} t = G^{+^T} \begin{pmatrix} t_1^T \\ t_2^T \end{pmatrix}$$

- **Relative Twist $\boldsymbol{t}$**: the relative twist is the component of null space of G: $V^T t = 0$

The **Equations of Motion of Dual Manipulator System is**:

$$\boldsymbol{M_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i)\dot{q} + g_i(q_i) = \tau_i - J_i^T w_i} \qquad \boldsymbol{with:\ \ i = 1,2}$$

$$\updownarrow \boldsymbol{or} \updownarrow$$

$$\boldsymbol{M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J_e^T w}$$

The **Equation of Motion of Rigid Object** is described by the classical Newton–Euler equations of rigid body:

$$\boldsymbol{M_E\dot{t}_E + C_E(t_E)t_E + g_E = w_E = Gw} \qquad \boldsymbol{where:} \quad \boldsymbol{M_o = \begin{pmatrix} m_o I & O \\ O & I_o \end{pmatrix}}, \quad \boldsymbol{C_o = \begin{pmatrix} m_o S(\omega_o) & O \\ O & S(\omega_o) I_o \end{pmatrix}}, \quad \boldsymbol{g_o = \begin{pmatrix} -m_o g_o \\ O \end{pmatrix}}$$

These equations must be completed by imposing the closed-chain constraints that born from the kinematic coupling between the two manipulators through the commonly manipulated rigid object:

$$V_1^T t_1 - V_2^T t_2 = G_1^{-T} t_1 - G_2^{-T} t_2 = 0 \ \rightarrow \ V_1^T J_1 \dot{q}_1 - V_2^T J_2 \dot{q}_2 = 0$$

In definitely, if we have $N_1 + N_2$ generalized coordiantes related by constraint, so the total number of degrees of freedom is $N_1 + N_2 - 6$ and model has form of a set of differential-algebraic equations.

11.3 Control

The object manipulation, the most typical control problem for for a multi-arm system, can be divided into two different control problems:

✓ Absolute motion control.
✓ Internal stress control.

In fact many control techinique for multi-arm systems are approach force/motion costituted by a control loop for motion and a control loop for force.

A first idea about control of multi-arm systems is **Leader/Follower Control**, which define:

- A master arm that control abolute motion of object, so this arm is drive by a position control for good performance, robust tracking and rejection respect to external disturbances, ecc.
- A slave arm that is controlled in force that must follow master arm.

This approach have different problem:

- The compliance of follower needs to be very large, so as to follow motion imposed by leader smoothly.
- In some cases it's necessary a dynamic role switching between master and slave.

For resolve these problems we should see the system *as a whole*.

In particular, we present the following approach where torques inputs τ to each arm in dynamic model $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J^T\omega$ is computed as the combination of two contributions:

$$\boldsymbol{\tau = \tau_p + \tau_w}$$

Where:

- τ_p: is action for control the object motion.
- τ_w: is action for control the internal forces.

For calculate τ_p e τ_w we have two ways:

A. PD Force/Motion Control: we use a PD control action τ_p with gravity compensation for regulate the motion of the object to a desidered pose. The internal force control term τ_w can be designed as follows:

$$\boldsymbol{\tau_p = K_P e_q - K_D \dot{q} + g(q) + J^T(q) G^+ g_e}$$

$$\boldsymbol{\tau_w = J^T(q) V \left(w_{Id} + K_h \int^t e_\omega d\xi \right)}$$

Where:

- $e_q = q_d - q$: is the error between vector of desired joint variables q_d and q;
- K_P and K_D: are positive definite matrix gains;
- g: is vector of gravitational torques action at the manipulators joint;
- g_e: is vector of gravity forces at the manipulated object;
- $e_w = w_{Id} - w_I$: is error between vector of constant desidered internal wrench ω_{Id} and vector of internal wrench ω_I.
- The integral action make the internal wrench error converges to zero at steady state.

This law of τ_p acts at the joint level, so we need a desired pose for each end-effector and thanks the CLIK algorithm we can calculate the desired joint trajectories q_d.

In alternatively we can use a law acts at the end-effector level:

$$\boldsymbol{\tau_p = J^T(q)(K_P e - K_V t) - K_D \dot{q} + g(q) + J^T(q) G^+ g_e}$$

Where:

- e: is error is in terms of end-effector pose variables;
- t: is the vector collecting the end-effector twists;
- K_P, K_V, K_D: are positive definite matrix gains.

Another ways is a control law direct in object space where errore e is in terms of object absolute pose variables and t_E is the object twist:

$$\boldsymbol{\tau_p = J^T(q) G^+ (K_P e - K_V t_E) - K_D \dot{q} + g(q) + J^T(q) G^+ g_e}$$

B. Inverse Dynamics: the idea is enhance force/motion control in terms of object motion tracking with an inverse dynamics control action. For this reason we choose the following law for τ_p, while τ_w is the same as the previous approach:

$$\boldsymbol{\tau_p = J^T(q) G^+ (M_E(\ddot{y}_{Ed} + K_V \dot{e}_E + K_P e_E) + C_E(t_E) t_E + g_E)}$$

$$\boldsymbol{\tau_w = J^T(q) V \left(\omega_{Id} + K_h \int^t e_\omega d\xi \right)}$$

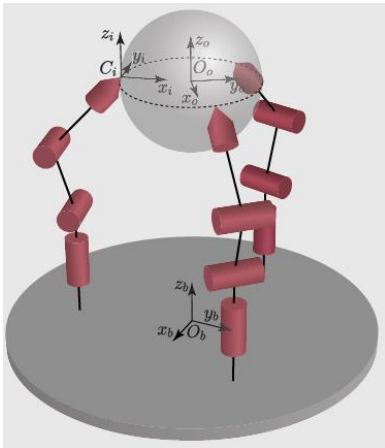
This control law yields a linear and decoupled closed-loop dynamics and this guarantees asymptotically vanishing motion and force errors.

$$\ddot{e}_E + K_V \dot{e}_E + K_P e_E = 0$$

$$e_q + K_h \int^t e_\omega d\xi = 0$$

11.4 Grasping

11.4.1 Multifinger Hand



The **Multifinger Hand** is the generalization of dual manipulator system where each finger corresponds to a manipulator. This particular system should have the following properties:

- ✓ Assicurate the grasping in different payload conditions, basically object with different size, geometry and mass.
- ✓ Prevent contact loss or undesired contact sliding.
- ✓ Assicurate the grasping in presence of unknow disturbing wrenches applied to the object.

Suppose to have a hand with N fingers where each finger is a sequence of links. Assume that links of each finger and the manipulated object are rigid bodies. Define the follow quantities:

- $M = \sum_i^N n_i$: is sum of degrees of freedom n_i of each finger ("i" is index of a specific finger).
- $q = (q_1 \dots \ q_M)^T$: is vector of joint variables.
- $\tau = (\tau_1 \dots \tau_M)^T$: is vector of all finger' torque.

For a formal study of grasping system, we need to remember a very intuitive and easy but important concept about the contact: **each contact should be considered as two coincident points, one on the object and one on the hand** (for this reason we define same quantities side hand and side object). Our final scope is characterize the dynamic model of our grasping system in contact with to object and derivate it we need to characterize two important element:

- **Grasp Matrix and Hand Jacobian Matrix**

In general these matries characterize relative velocity and transmitted forces to the contact. In particular:

- **Grasp Matrix $\bar{G}^T$**: is matrix that define the relation between aboluste twist t_e and twist $t_{c,obj}$ of all contact points *on the object*:

$$\boldsymbol{t_{c,obj} = \bar{G}^T t_E = \begin{pmatrix} \bar{G}_1 \\ \dots \\ \bar{G}_N^T \end{pmatrix} t_E \qquad where: \ {}^i t_{i,obj} = \bar{G}_i^T t_E \qquad i = 1, \dots, N}$$

- **Hand Jacobian $\bar{J}$**: is matrix that define the relation between joint velocities $\dot{q}$ and twist $t_{c,hnd}$ of all contact points *on the hand*:

$$\boldsymbol{t_{c,hnd} = \bar{J}\dot{q} = \begin{pmatrix} \bar{J}_1 \\ \dots \\ \bar{J}_N \end{pmatrix} \dot{q} \qquad where: \ {}^i t_{i,hnd} = \bar{J}_i \dot{q} \qquad i = 1, \dots, N}$$

The twists on all contacts with object is a vector where the components are twists on each contact. These components can be computed from the absolute object twist t_E and matrix G_i^T. Notiche that all quantities are refer to frame i attached to each contact, in fact in all relations we have: $\bar{G}_i = \bar{R}_i^T G_i \qquad \bar{J}_i = \bar{R}_i^T J_i$

- **Contact Model**

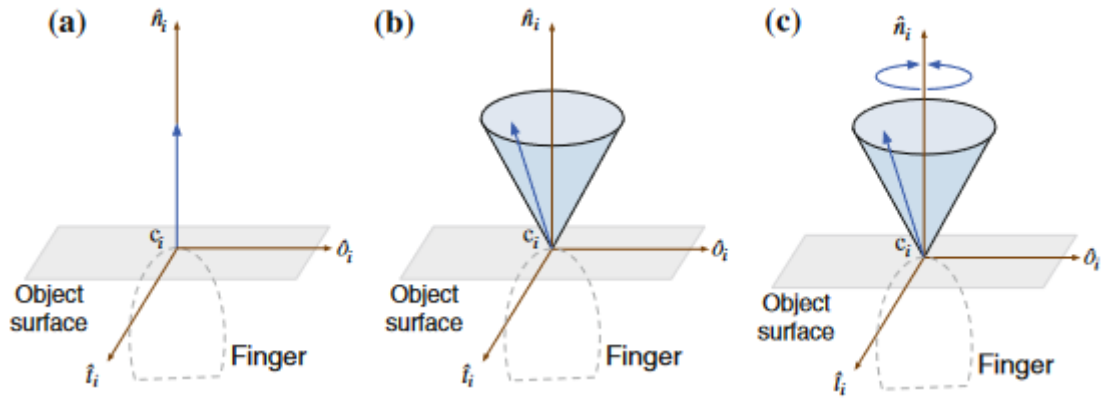
This model characterises which velocities and forces are transmitted to the contact. It's important aspect because through it we can impose a particular motion or apply a specific force to the object. The three contact model that we study are:

A. Point-Contact-Without-Friction (PwoF): is used when the contact zone is very small and we haven't friction. In this model:
- **Transmitted**: normal component of the translational velocity and normal component of contact force.
- **Not Transmitted**: two components of tangential velocity, the three components of angular velocity, two components of contact force and all moments.

B. Hard-Finger (HF): is used when there is significant contact friction but contact zones are small. In this model:
- **Transmitted**: all three translational velocity components of contact point on hand transmitted to object and all three components of contact force transmitted.
- **Not Transmitted**: none of angular velocity components transmitted and none of moment components.

C. Soft-Finger (SF): is used when surface friction and contact zones are large enough to generate significant friction forces and friction moment:
- **Transmitted**: three translational velocity components of contact point on the hand and three components of contact force and normal component of contact moment transmitted through contact.

Substantially equivalent to admitting that finger links and object are not rigid, yet applicable when amount of deformation is small vs large contact patch.



The definition of contact model allow to specialised the grasp matrix and hand jacobian matrix respect a particular contact model. Each contact model is associates a particular diagonal block matrix that characterize it to each i contact: $H = blockdiag(H_1, \dots, H_n)$

At this point we can re-write $\bar{G}$ and $\bar{J}$ specialized for a particular contact model:

- **Grasp Matrix** $\boldsymbol{G_H^T = H\bar{G}^T \in R^{6\times K}}$
- **Hand Jacobian Matrix** $\boldsymbol{J_H = H\bar{J} \in R^{K\times M}}$

In particular:

- J_H and G_H^T transmit only selected components of contact twists.
- J_H^T and G_H transmit only selected components of contact wrenches.

The **Dynamic Model of Multifinger Hand** is the following and it is subject to constraint on the right:

$$\begin{pmatrix} J_H^T \\ -G_H \end{pmatrix} \lambda = \begin{pmatrix} \tau_{hand} \\ w_{obj} \end{pmatrix} \qquad s.a. \quad \begin{pmatrix} J_H & -G_H^T \end{pmatrix} \begin{pmatrix} \dot{q} \\ t_E \end{pmatrix} = 0$$

$$\updownarrow or \updownarrow$$

$$\begin{matrix} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J_H^T \lambda \\ M_E \dot{v}_E + C_E(t_E) t_E + g_E = G_H \lambda \end{matrix} \qquad s.a. \quad J_H \dot{q} = t_{c,hnd} = t_c = t_{c,obj} = G_H^T t_E$$

Where:

- J_H: is the hand jacobian matrix.
- G_H: is the grasp matrix.
- $t_c =$ is twist trasmitted through the contact.
- λ: is the vector of contact wrench components transmitted through the contacts and expressed in the contact frames.

$$\lambda = (\lambda_1^T \ \dots \ \lambda_N^T)^T \qquad where: \ \lambda_i = H_i(f_{ix} \ \ f_{iy} \ \ f_{iz} \ \ \mu_{ix} \ \ \mu_{iy} \ \ \mu_{iz})$$

- $\tau_{hnd} = \tau - M(q)\ddot{q} - C(q,\dot{q})\dot{q} - g(q)$
- $w_{obj} = -M_E \dot{v}_E - C_E(t_E) t_E - g_E$

Notice that, the physical meaning of constraint said that *in contact phase* twist side hand and twist side object are equal.

The **Quasi-Static Model** is a version of dynamic model valid when the inertia terms are negligible and it doesn't depend on joint and object velocities. In this case we obtain a force/velocity decoupled solution that is not accettable when dynamic effects are appreciable, because first equation depends on the second through equation about w_{obj}.

$$\begin{matrix} \tau_{hnd} = \tau - g(q) \\ w_{obj} = -g_E \end{matrix}$$

11.4.2 Grasp Classification

In hand design and manipulation planning, it is important to know two different sets:

- ❖ The set of twists that the hand can apply to the object.
- ❖ The set of conditions under which the hand can prevent all possible motions of the object.

Or, for kineto-static duality:

- ❖ The set of wrenches that hand can apply to object
- ❖ The set of conditions under which any wrench can be applied through contacts

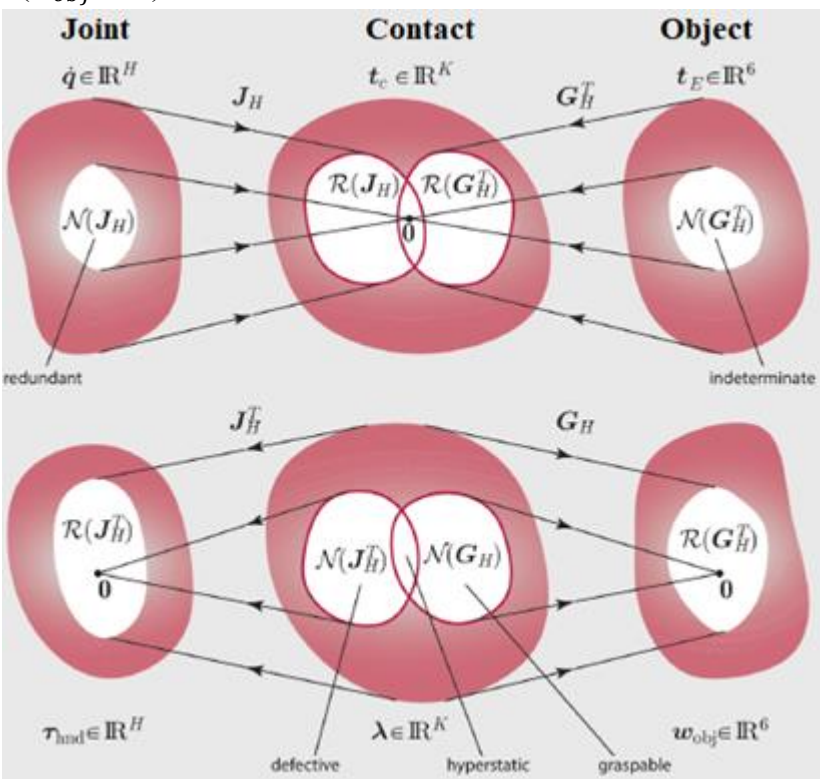
This knowledge will be obtained by studying the different subspaces associated with grasp matrix G_H and hand jacobian J_H. At this point, it's important remember the foundamental relations on the right where there are two particularity:

$$\dot{\boldsymbol{q}} = \boldsymbol{J_H^+ t_c} + \boldsymbol{N(J_H)\gamma}$$
$$\boldsymbol{t_E} = \left(\boldsymbol{G_H^T}\right)^+ \boldsymbol{t_c} + \boldsymbol{N\left(G_H^T\right)\gamma}$$
$$\boldsymbol{\lambda} = \left(\boldsymbol{J_H^T}\right)^+ \boldsymbol{\tau_{hnd}} + \boldsymbol{N\left(J_H^T\right)\gamma}$$
$$\boldsymbol{\lambda} = -\boldsymbol{G_H^+ w_{obj}} + \boldsymbol{N(G_H)\gamma}$$

- The matrix A^+ (can be J_H o G_H) denotes the generalized inverse (pseudo-inverse) and $N(A)$ is a matrix where columns form a basis for $\mathcal{N}(A)$.
- The quantity γ is an arbitrary vector of appropriate dimension that parameterizes the solution sets.

Through the previously relation, in a grasping system:

- The hand jacobian matrix J_H associates joint's velocities to contact twists that belong to $\mathcal{R}(J_H)$. However, all $\dot{q} \in \mathcal{N}(J_H)$ are mapped on null twist contact ($t_c = 0$). In duality view, J_H^T associates wrench to contact λ with torque τ_{hand} trasmitted by hand. However, all $\lambda \in \mathcal{N}(J_H^T)$ are mapped on null joint velocity ($\dot{q} = 0$).
- The grasp matrix G_H^T associates abolute twists t_e to contact twist t_c that belong to $\mathcal{R}(G_H^T)$. However, all $t_e \in \mathcal{N}(G_H^T)$ are mapped on null twist contact ($t_c = 0$). In duality view, G_H associates wrench to contact λ with wrench w_{obj}. However, all $\lambda \in \mathcal{N}(G_H)$ are mapped on null wrench ($w_{obj} = 0$).



Observing the previously scheme, the different nullspace of J_H, G_H and relative transpose provide a classification of grasping system:

Condition	Class	Many-to-one
$\mathcal{N}(\boldsymbol{J}_H) \neq \emptyset$	Redundant	$\dot{\boldsymbol{q}} \rightarrow \boldsymbol{t}_c$ $\boldsymbol{\tau}_{\text{hnd}} \rightarrow \boldsymbol{\lambda}$
$\mathcal{N}(\boldsymbol{G}_H^T) \neq \emptyset$	Indeterminate	$\boldsymbol{t}_E \rightarrow \boldsymbol{t}_c$ $\boldsymbol{w}_{\text{obj}} \rightarrow \boldsymbol{\lambda}$
$\mathcal{N}(\boldsymbol{G}_H) \neq \emptyset$	Graspable	$\boldsymbol{\lambda} \rightarrow \boldsymbol{w}_{\text{obj}}$ $\boldsymbol{t}_c \rightarrow \boldsymbol{t}_E$
$\mathcal{N}(\boldsymbol{J}_H^T) \neq \emptyset$	Defective	$\boldsymbol{\lambda} \rightarrow \boldsymbol{\tau}_{\text{hnd}}$ $\boldsymbol{t}_c \rightarrow \dot{\boldsymbol{q}}$

A. **Redundant**: grasping system is said to be redundant if $\mathcal{N}(J_H)$ is non-trivial. Joint velocities $\dot{q} \in \mathcal{N}(J_H)$ are called **Internal Hand Velocities** because they correspond to finger motions, but they don't generate motion of the hand in the constrained directions at the contact points.
B. **Indeterminate**: a grasping system is said to be indeterminate if $\mathcal{N}(G_H^T)$ is non-trivial. Object twists $t_E \in \mathcal{N}(G_H^T)$ are called **Internal Object Twists** because they correspond to motions of the object, but they don't cause motion of the object in the constrained directions at the contacts.
C. **Graspable**: a grasping system is said to be graspable if $\mathcal{N}(G_H)$ is non-trivial. Wrench intensities $\lambda \in \mathcal{N}(G_H)$ are called **Internal Object Forces**. These wrench play a fundamental role in maintaining grasps in presence of friction, but they don't contribute to the acceleration of the object because $G_H\lambda = 0$.
D. **Defective**: a grasping system is said to be defective if $\mathcal{N}(J_H^T)$ is non-trivial. Wrench intensities $\lambda \in \mathcal{N}(J_H^T)$ are called **Internal Hand Forces**. These forces don't influence the hand joint dynamics and they are absorbe by the structure of the hand.

In really exist another class called **Hyperstatic** where a grasping system is said to be hyperstatic if $\mathcal{N}(J_H^T) \cap \mathcal{N}(G_H)$ is non-trivial. In such systems there are internal forces, like glasplable system, belonging $\mathcal{N}(J_H^T)$ that *are not controllable*.

11.4.3 Control Properties

For a general purpose grasping system, there are three main desirable control properties:

- Control of the object twist t_E.
- Control of object wrench w_{obj}.
- Control of the internal forces.

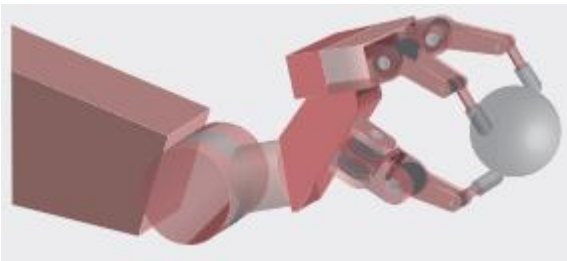
Control of these quantities implies that, appropriately choosing of joint velocities and actions, robot's hand deliver the desired t_E and w_{obj} with specified grip pressure. These control properties can be traduce in conditions on J_H and G_H, which are summerized in following table:

Task requirement	Required condition
All wrenches possible $\boldsymbol{w}_{\text{obj}}$ All twists possible $\boldsymbol{t}_E$	$\text{rank}(\boldsymbol{G}_H) = 6$
Control all wrenches $\boldsymbol{w}_{\text{obj}}$ Control all twists $\boldsymbol{t}_E$	$\text{rank}(\boldsymbol{G}_H \boldsymbol{J}_H) = \text{rank}(\boldsymbol{G}_H) = 6$
Control all internal forces	$\mathcal{N}(\boldsymbol{J}_H^T) \cap \mathcal{N}(\boldsymbol{G}_H) = \emptyset$

We analyze one by one:

- **All Object Twists Possible t_E**: if we want achieve any object twist then $\mathcal{N}(G_H) = \emptyset$ and these implies that $rank(G_H) = 6$. Any grasp with three non-collinear hard contacts or two distinct soft contacts satisfies this condition. In particular, given a set of contact locations and types, by solving the twist constraint equation for t_E or observing the map G_H. In fact we have that:
  - The achievable object twists are those in $\mathcal{R}(G_H)$.
  - The unfeasible object twists are those in $\mathcal{N}(G_H)$.
- **All Object Wrench Possibile w_{obj}**: this is the dual case of the previous case but relative to wrench. In fact, from the wrench constraint equation we obtain $\mathcal{N}(G_H^T) = \emptyset$ and again $rank(G_H) = 6$.
- **Control All Object Twists t_E**: if we resolve the twist constraint equation for t_E we can see that for cause any object twist t_E, by choice of joint velocities $\dot{q}$, it must be $\mathcal{R}(G_H J_H) = R^6$ and $\mathcal{N}(G_H^T) = \emptyset$. These conditions are equivalent to says that $rank\big(\mathcal{R}(G_H J_H)\big) = rank(G_H) = 6$.
- **Control All Object Wrench w_{obj}**: this is the dual case of the previous case but relative to wrench. Analysis of the wrench constraint equation yields the same condition $rank\big(\mathcal{R}(G_H J_H)\big) = rank(G_H) = 6$.
- **Controll All Internal Forces**: in general, not all the internal forces may be actively controlled by joint actions. It can be shown that all internal forces in $\mathcal{N}(G_H)$ are controllable if and only if $\mathcal{N}(J_H^T) \cap \mathcal{N}(G_H) = \emptyset$.

Example: the Salinsbury Hand is designed to have the smallest number of joints that would meet all the task requirements in the table above. Assuming HF contacts, three non-collinear contacts is the minimum number such that $rank = 6$. So we have:



- G_H has siz row and nine columns.
- $\mathcal{N}(G_H)$ has dimension three.

- The ability to control all internal forces and apply an arbitrary wrench to the object requires that $\mathcal{N}(J_H^T) \cap \mathcal{N}(G_H) = \emptyset$. For obtain this, the minimum dimension of the range space of J_H should be nine. In practice, the hand must have at least nine joints implemented as three fingers, each with three revolute joints.

In definitely, the Salinsbury Hand execute the manipulation forming a grasp triangle with three non-collinear points. In this way the manipulation can be thought like a control of the vertices of the grasp triangle. Therefore, we must do attention to control internal forces for maintained the contact points to avoid sliding.