

POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea in
Ingegneria Aeronautica



PROGETTAZIONE DI COMANDI PER UN ROBOT DI TIPO SCARA

Relatore: Prof. Marco MORANDINI

Tesi di Laurea di:

Antonio VENTILII Matr. 783079

Anno Accademico 2012 - 2013

Indice Generale

1	Obiettivo del robot	5
1.1	Descrizione del sistema	5
1.2	Applicazioni.....	5
1.3	Approccio all'ottimizzazione.....	6
1.4	Programmi utilizzati	6
2	Modello del sistema	8
2.1	Descrizione fisica del modello.....	8
2.2	Descrizione cinematica e geometrica del modello.....	8
2.3	Approssimazioni e sistema-esempio.....	9
3	Metodi scartati	12
3.1	Metodo della distanza minima.....	12
3.2	Metodo della doppia cinematica	12
3.3	Sistema-esempio con il metodo della doppia cinematica	13
3.4	Inefficacia del metodo della doppia cinematica.....	15
4	Algoritmo point-by-point proposto	18
4.1	Considerazioni iniziali	18
4.2	Premessa e definizioni	18
4.3	Spiegazione dell'algoritmo	18
5	Studio di soluzione ottima	22
5.1	Valutazione dell'ottimo	22
5.2	Razionale	22
6	Applicazione al sistema-esempio	24
7	Esecuzione dell'algoritmo e applicazioni tipo	32
7.1	Estremi del raggio di azione	32
7.2	Applicazione tipo: disposizione e prelievo di elementi.....	33
7.3	Applicazione tipo: deposito e azione di elementi	33
7.4	Disegno e tracciamento.....	34
8	Sviluppi e risultati	38
8.1	Risultati elaborati	38
8.2	Generalizzazione.....	47

Elenco delle Figure

1	Schema di implementazione.....	7
2	Angolo relativo dell'asta k con l'asta k-1 e punti di estremità.....	9
3	Sistema-esempio in 3 dimensioni.....	10
4	Sistema-esempio in 2 dimensioni.....	11
5	Metodo della distanza minima.....	12
6	Primo tipo di cinematica nel sistema esempio	14
7	Secondo tipo di cinematica nel sistema esempio	15
8	Formazione iniziale per esempio 1.....	16
9	Esempio di soluzione del metodo della doppia cinematica.....	17
10	Esempio di soluzione ottima	17
11	Passo 0 nel sistema-esempio	24
12	Passo 1 nel sistema-esempio	25
13	Passo 2-a nel sistema-esempio	26
14	Passo 2-b nel sistema-esempio	27
15	Passo 1bis-a nel sistema-esempio.....	28
16	Passo 1bis-b nel sistema-esempio	29
17	Passo 0bis nel sistema-esempio.....	30
18	Soluzione ottima nel sistema-esempio	31
19	Raggio di azione massimo.....	32
20	Disposizione di elementi	33
21	Deposito di elementi.....	34
22	Disegno di quadrato: fase 1	35
23	Disegno di quadrato: fase 2	36
24	Disegno di quadrato: fase 3	36
25	Disegno di quadrato: fase 4	37
26	Simulazione 1: tragitto completo.....	38
27	Simulazione 1: asta 1.....	39
28	Simulazione 1: asta 2.....	40
29	Simulazione 1: asta 3.....	41
30	Simulazione 1: asta 4.....	42
31	Simulazione 1: asta 5.....	43
32	Simulazione 1: asta 1 - confronto.....	43
33	Simulazione 1: asta 2 - confronto.....	43
34	Simulazione 1: asta 3 - confronto.....	43
35	Simulazione 1: asta 4 - confronto.....	43
36	Simulazione 1: asta 5 - confronto.....	43

Al Prof. Morandini, per essere riuscito a seguirmi mentre ero dall'altra parte del mondo.

Ai miei genitori, i migliori professori che un figlio possa desiderare.

Ai miei fratelli, senza i quali non sarei spinto a dare il buon esempio.

A quelli del S. Leonardo, che hanno trasformato in casa un posto inimmaginabile.

A quelli dell'università, che hanno reso le lezioni degne di essere vissute.

E soprattutto a Isa, la metà che stavo aspettando da una vita.

GRAZIE!

Per avermi fatto arrivare qui e avermi fatto diventare quello che volevo.

1 OBIETTIVO DEL ROBOT

1.1 Descrizione del sistema

La seguente tesi tratta dell'ottimizzazione dei comandi da fornire ad un robot SCARA (*Selective Compliance Arm for Robotic Assembly*) che deve agire in determinati punti di lavoro forniti dall'utente. Questo tipo di robot è costituito da una serie di braccia meccaniche, ognuna delle quali ha un solo grado di libertà: la rotazione libera attorno ad un asse perpendicolare al piano di azione. Il robot ha perciò la possibilità di agire in qualsiasi punto di un piano x-y ruotando le proprie braccia rispetto all'asse z perpendicolare al piano.

Fu creato nel 1981 come sviluppo dei robot industriali, per fornire maggiore libertà di azione e maggiore versatilità nei processi di assemblaggio perché ha contemporaneamente la capacità di definire il proprio raggio di azione e di restringerlo entro una determinata area del raggio di azione massimo.

Ogni braccio è azionato da un motore inserito nelle cerniere che li legano; ciò vuol dire che se non viene fornita energia ai motori, le cerniere bloccano qualsiasi rotazione libera o spontanea. Ad esempio se viene fornita energia solo al primo motore (quello che lega il primo braccio al terreno) il robot intero ruota rigidamente attorno all'asse di rotazione del primo braccio, mantenendo fisse le posizioni dei rimanenti. Si considerano simili i motori di azionamento del sistema, al fine di poter studiare esclusivamente l'ottimizzazione del sistema dal punto di vista fisico.

1.2 Applicazioni

Principalmente viene usato nei processi industriali al fine di spostare determinati elementi in posizioni stabilite oppure unirli per il montaggio di un elemento più grande o complesso. In questo caso prima di iniziare l'esecuzione al robot viene fornita una serie ordinata di punti sui quali deve agire.

Tuttavia si è sviluppato un altro tipo di applicazione con l'avvento dell'automazione complessa: questo tipo di robot viene integrato più volte all'interno di robot più complessi come parte con obiettivo di movimentazione e interazione con oggetti esterni. In questo modo il robot affronta una serie di obiettivi mutabili durante il lasso di tempo di esecuzione, conformemente alle alterazioni o implementazioni negli obiettivi generali del robot principale; quest'ultima possibilità è evidentemente quella

che lascia più libertà all'utente in quanto, mentre il robot lavora, egli può decidere di alterare i punti di esecuzione, eliminandoli, modificandoli o aggiugnendone di nuovo.

1.3 Approccio all'ottimizzazione

Considerando questa la condizione al contorno del problema, si deve pensare che l'algoritmo di esecuzione non può prevedere a priori il numero di punti per cui sarà necessario ottimizzare i comandi, ma con certezza sa quale è il punto obiettivo successivo in qualsiasi istante di tempo di esecuzione. Quindi non può decidere quale sia la soluzione ottima globale sapendo che i dati forniti di input potrebbero variare una volta avviato il robot. Da ciò si deduce che la soluzione ottima in un qualsiasi istante di tempo durante l'azionamento fa parte del dominio di soluzioni in cui i punti obiettivo non cambiano, ovvero coincide con la soluzione ottima per il raggiungimento del prossimo punto previsto in quello stesso istante.

In conclusione i metodi che verranno presentati si basano sullo studio di ottimizzazione *point-by-point* cioè cercando la soluzione ottima per far muovere il robot da un punto di esecuzione al suo successivo. Questa verrà studiata tramite l'algoritmo al fine di scoprire quale dovrebbe essere la rotazione di ogni braccio; si lascia quindi l'esecuzione ad un programma di elaborazione a cui viene dato come input il modello e la soluzione ottenuta.

1.4 Programmi utilizzati

Viene sviluppata l'implementazione informatica tramite due diversi programmi in intercomunicazione tra essi: Matlab e MBDyn.

Lo studio e l'applicazione dell'algoritmo al sistema verranno in primo luogo implementati nel programma Matlab: agli script verranno forniti in ingresso le proprietà fisiche del sistema e i punti di lavoro del robot, al fine di restituire i vincoli imposti dall'algoritmo di ottimizzazione, cioè gli angoli di rotazione di ogni braccio, e i file di costruzione del modello per la simulazione.

Il programma utilizzato per l'elaborazione dei dati e la simulazione è MBDyn, il software di analisi di Sistemi Dinamici *Multibody* sviluppato dal Dipartimento di Ingegneria Aerospaziale del Politecnico di Milano: questi realizza la simulazione di un sistema lineare e non lineare di corpi rigidi soggetti a vincoli cinematici. Il sistema inoltre può essere integrato con dinamiche differenti come reti elettriche, azionamenti

idraulici e sistemi aeronautici già modellati. Come già descritto, viene modellato il sistema tramite i file forniti da Matlab includendo i vincoli di ottimizzazione e vengono restituiti i dati di simulazione del sistema.

Questi infine verranno posti sotto forma di grafico o animazione tramite ulteriori script di Matlab. In Figura 1 viene mostrato lo schema di implementazione completo con i rapporti di input/output.

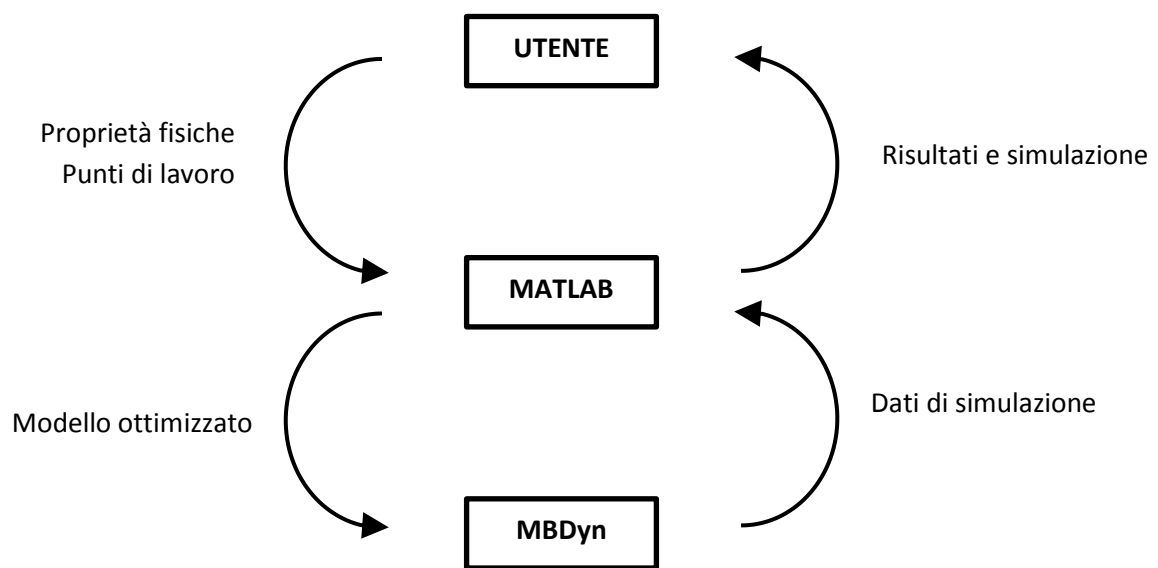


Fig.1 Schema di implementazione

2 MODELLO DEL SISTEMA

2.1 Descrizione fisica del modello

Essenzialmente per descrivere il sistema viene utilizzato un modello bidimensionale sul piano x-y: si tratta di un insieme di n aste cilindriche che rappresentano le braccia del robot, ognuna con definite proprietà fisiche:

- lunghezza l_k
- raggio r_k
- massa m_k
- matrice di inerzia J_k

Si considereranno le aste con rigidità elevata sull'asse z rispetto alle altre due direzioni cartesiane x e y.

Le aste sono legate le une alle altre tramite cerniere piane: partendo dall'origine l'asta 1 ha un'estremità legata a terra e l'altra legata all'asta 2. In generale l'asta k ha un'estremità legata all'asta k-1 e un'altra all'asta k+1. L'unica eccezione è l'asta n che possiede un'estremità libera, quella che deve coincidere con i punti obiettivo.

Ciò porta il sistema ad avere un numero di gradi di libertà di base pari a:

$$n_{gdl} = 3n - 2n = n$$

2.2 Descrizione cinematica e geometrica del modello

Per definire le proprietà cinematiche del sistema è necessario stabilire come definire la posizione del sistema. Si è scelto di identificare il sistema con un vettore di angoli α_k con $k = 1:n$ che rappresentano la posizione angolare di ogni asta rispetto alla precedente:

$$A = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_{n-1} \quad \alpha_n]$$

Come mostrato in Figura 2 si definiscono inoltre i punti estremità $P_k = (x_k \quad y_k)$ con $k = 1:n$ di ogni asta come il punto dell'asta k che la lega all'asta k+1, per cui:

$$x_k = x_{k-1} + l_k \cos \alpha_k$$

$$y_k = y_{k-1} + l_k \sin \alpha_k$$

dove si considera $P_0 = (x_0 \ y_0) = (0 \ 0)$

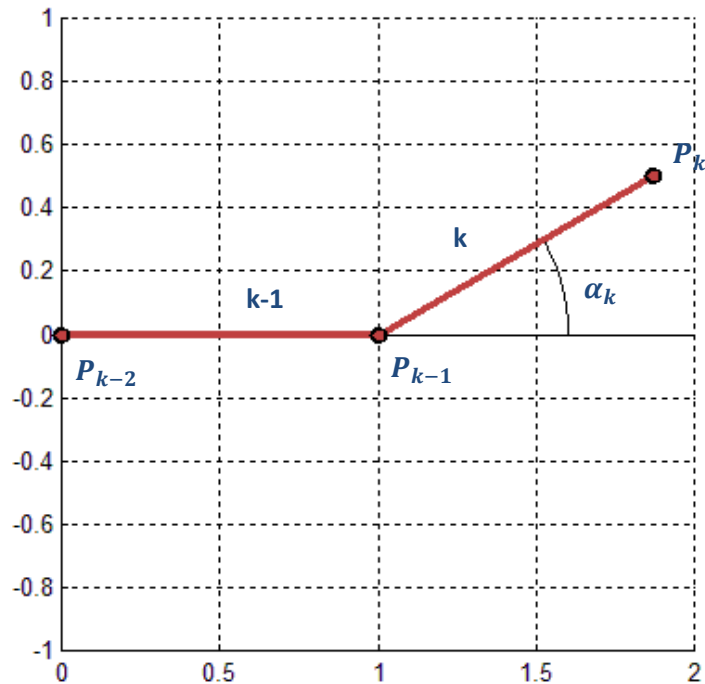


Fig.2 Angolo relativo dell'asta k con l'asta k-1 e punti di estremità

2.3 Approssimazioni e sistema-esempio

Il modello scelto risulta come buona approssimazione del sistema reale ai fini dello studio di ottimizzazione; difatti se si volesse realizzare un modello più complesso, ciò che varierebbe realmente sarebbe la conformazione fisica e le proprietà delle aste. Tuttavia questa variazione influenzerebbe maggiormente la simulazione fornita dal programma, piuttosto che l'algoritmo di ottimizzazione, poiché qualsiasi tipo di braccio fosse scelto potrebbe essere modellato come un'asta rigida, a causa dello scopo per cui verrà utilizzato.

Nelle Figure 3 e 4 è raffigurato un sistema con $n=5$ e con aste con stesse lunghezze l e proprietà fisiche. Questo sistema verrà considerato come sistema-esempio per i prossimi ragionamenti e schemi logici; si aggiunge che inizialmente il sistema si trova in formazione con tutte le aste giacenti sull'asse x .

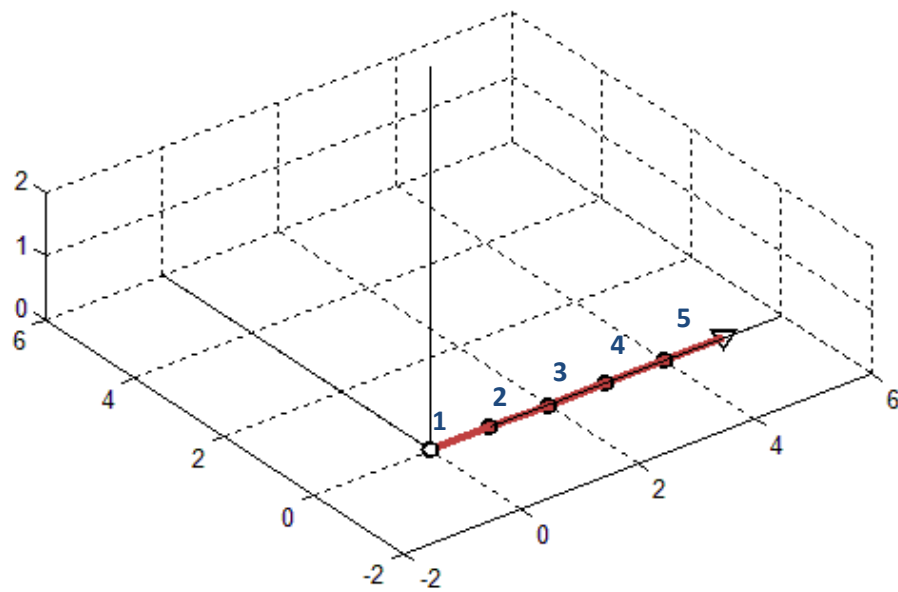


Fig.3 Sistema-esempio in 3 dimensioni

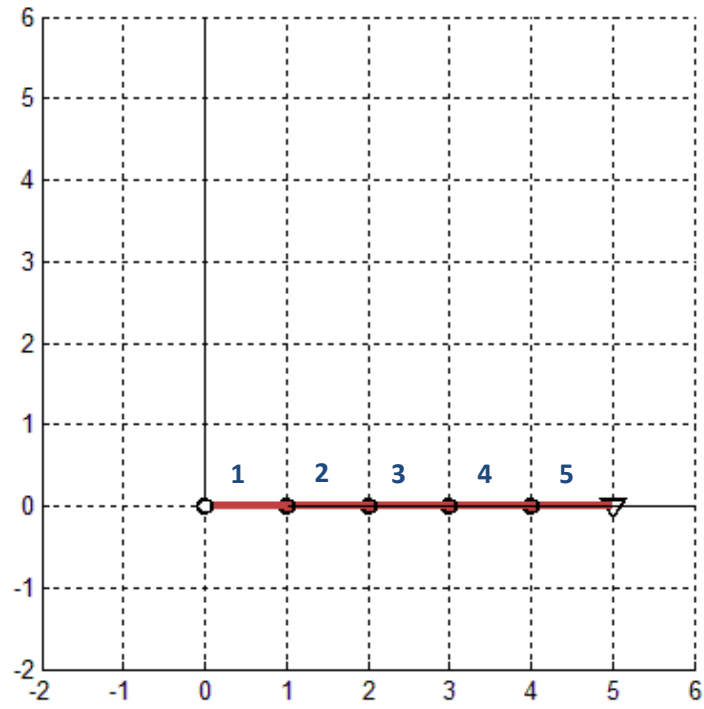


Fig.4 Sistema-esempio in 2 dimensioni

3 METODI SCARTATI

3.1 Metodo della distanza minima

Viene affrontato lo studio di ottimizzazione valutando inizialmente il comportamento del sistema a seguito di situazioni operative semplici. Ad esempio il primo metodo considerato è il più semplice oltre che il meno efficace: si tratta del tragitto diretto con distanza minima dell'estremità libera da un punto obiettivo all'altro.

L'unico vincolo imposto al sistema è la traiettoria dell'estremità libera, che è anche l'unico vincolo imposto al programma di simulazione. Il sistema viene perciò solamente vincolato presso l'ultimo braccio e si lasciano i motori in regime libero, cioè essi non oppongono resistenza alle rotazioni libere delle aste. In queste condizioni le cerniere vedono applicate delle rotazioni non controllate alle aste come risultato del moto conseguente al movimento dell'estremità libera, senza vincoli di alcun genere.

Il tragitto complessivo è formato semplicemente dall'insieme dei segmenti che hanno come estremità i punti obiettivo. Questo rende la soluzione poco efficace considerando che le aste risultano essere in moto libero, senza movimenti imposti di alcun tipo, soggette a forze e moti oscillatori perlopiù non voluti. Un possibile miglioramento si può effettuare cercando di ridurre le rotazioni relative libere, ovvero aggiungendo fattori di attrito viscoso e di smorzamento nei motori che limitino il movimento di ogni asta. Uno studio di questo genere non verrà affrontato perché servirebbe solo a migliorare una soluzione già di per sé non efficiente e non sotto diretto controllo dell'utente.

3.2 Metodo della doppia cinematica

Un ulteriore metodo che è stato soggetto di studio è costituito dalla combinazione di due tipi di cinematiche differenti. Il punto obiettivo viene identificato in coordinate polari (ρ_f θ_f) rispetto all'origine e all'asse x e vengono calcolati due diversi tipi di rotazione per ogni asta. Le due tipologie differiscono per lo scopo:

- per il primo si vuole raggiungere la coordinata angolare del punto nel piano; vengono considerate le aste vincolate le une alle altre, come se fossero un unico corpo rigido (le cerniere diventano vincoli di incastro), con l'unica possibilità di ruotare attorno all'origine (1 gdl), e viene raggiunta la coordinata θ_f del

- punto obiettivo; si tratta perciò della rotazione semplice dell'asta 1, a cui seguono le altre per moto rigido di trascinamento;
- per il secondo si raggiunge la coordinata ρ_f del punto obiettivo lungo la retta che congiunge l'estremità libera con l'origine, cioè si diminuisce la distanza tra il punto P_n (l'estremità non vincolata dell'asta n) e l'origine P_0 , con una rotazione equamente divisa tra tutte le aste.

3.3 Sistema-esempio con il metodo della doppia cinematica

Si consideri il sistema-esempio e un punto obiettivo di coordinate cartesiane $(x_f \ y_f) = (2 \ 2)$, ovvero di coordinate polari $(\rho_f \ \theta_f) = (2\sqrt{2} \ 45^\circ)$. Il metodo considerato propone la seguente risoluzione:

- il primo tipo di cinematica del sistema ha come obiettivo il raggiungimento di una rotazione totale del sistema di 45° , cioè si considera il sistema come un'asta rigida unica di lunghezza $5l$ che ruota di 45° rispetto all'asse x ;

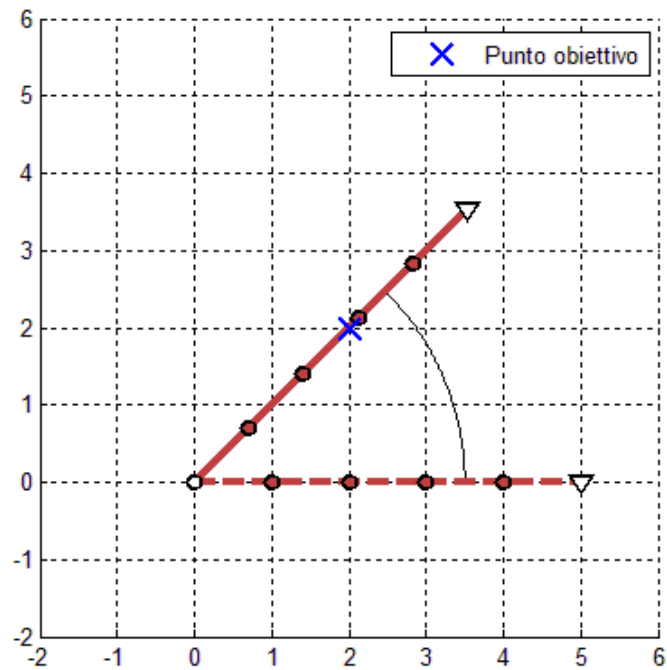


Fig.6 Primo tipo di cinematica nel sistema esempio

- il secondo tipo di cinematica fornisce una rotazione equi divisa ad ogni asta, alternatamente in senso orario e in senso antiorario; per capire meglio come è calcolato l'angolo di rotazione si tenga in considerazione che in questo caso la distanza tra l'estremità libera e l'origine deve essere pari a $2\sqrt{2}$ quando nella posizione iniziale è pari a 5 (per semplicità si sta considerando $l = 1$).

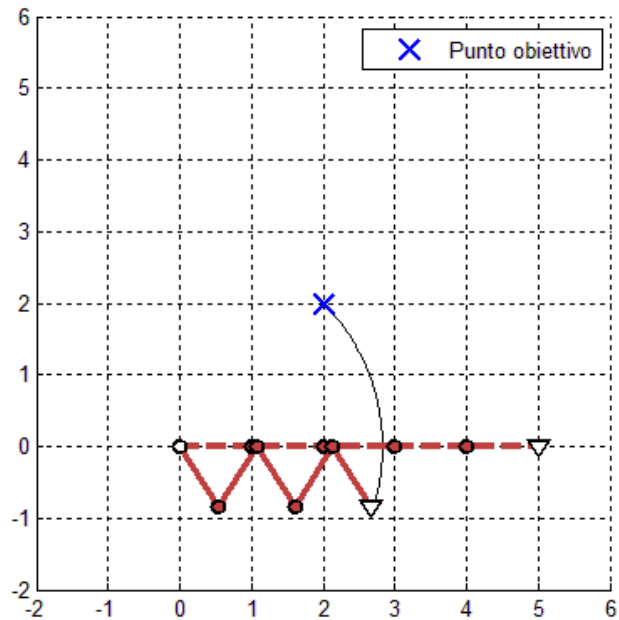


Fig.7 Secondo tipo di cinematica nel sistema esempio

La combinazione lineare delle due cinematiche costituisce una soluzione del problema.

3.4 Inefficacia del metodo della doppia cinematica

È facilmente dimostrabile la poca efficienza di questo metodo considerando la posizione iniziale delle aste come tutte giacenti sull'asse x (come in Figura 8) e definendo il punto obiettivo coincidente con la cerniera che lega la penultima asta con la terzultima. Per questo esempio si considerano 5 aste di pari lunghezza l .

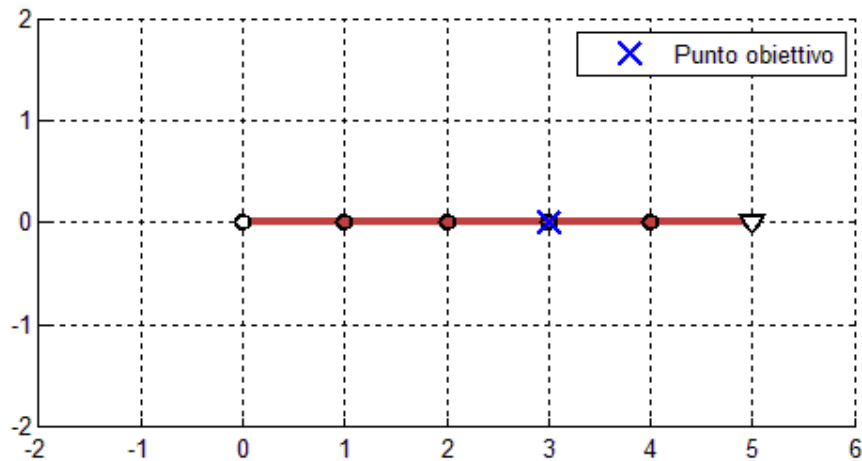


Fig.8 Formazione iniziale per esempio 1

Si analizza il sistema con il metodo proposto: il punto obiettivo ha coordinate polari $(3 \mid 0^\circ)$.

- Il sistema non ha bisogno di ruotare rigidamente come se fosse un unico corpo rigido, in quanto la coordinata angolare del punto coincide con quella del sistema complessivo. In questo caso il primo tipo di cinematica è nullo.
- È necessario diminuire la distanza tra l'estremità libera e l'origine, ruotando ogni asta di un angolo relativo equamente diviso.

Unendo i due schemi si ottiene la soluzione completa fornita da questo metodo (Figura 9)

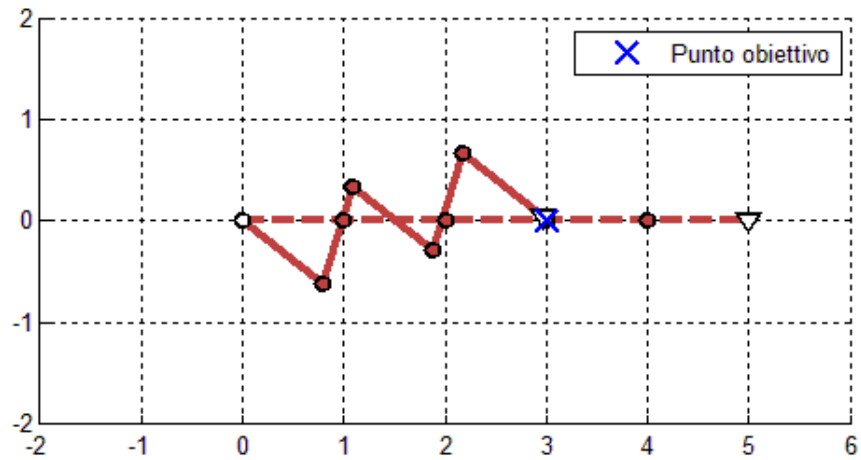


Fig.9 Esempio di soluzione del metodo della doppia cinematica

La soluzione trovata è evidentemente la più “dispendiosa”, dovuto all’elevato numero di rotazioni necessarie al raggiungimento del punto obiettivo. Come è facile intuire la soluzione più semplice (e allo stesso tempo la soluzione ottima) è costituita dalla rotazione di 180° della singola asta finale (Figura 10).

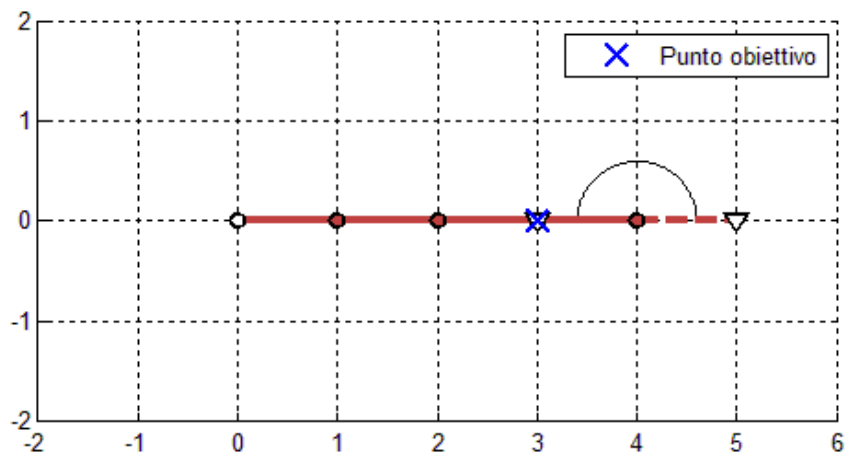


Fig.10 Esempio di soluzione ottima

4 ALGORITMO POINT-BY-POINT PROPOSTO

4.1 Considerazioni iniziali

Dalle considerazioni precedenti si può fare una prima elucubrazione per lo studio di un nuovo metodo: si è dedotto come lo spostamento di una singola asta possa fornire una soluzione migliore dello studio sul sistema totale di aste. Ciò vuol dire che a fronte di un punto obiettivo è meglio considerare un sottoinsieme del gruppo di aste e verificare se esiste una soluzione agendo solamente su di esso.

Inoltre è meglio cominciare lo studio dal sottosistema di ordine minore, per poi proseguire lo studio fino a quello di ordine maggiore. In questo modo se, per esempio, il punto obiettivo è lo stesso dell'esempio del paragrafo precedente, allora lo studio del sottosistema costituito dalla sola asta finale è sufficiente a trovare una soluzione ottima.

4.2 Premessa e definizioni

Per spiegare l'algoritmo utilizzato sono necessarie alcune premesse:

- viene definita *formazione* la posizione ad un dato istante dell'insieme di aste nel loro complesso, cosicché, per esempio, la formazione iniziale del sistema consiste nella posizione delle insieme di aste, considerandone gli angoli e le posizioni rispetto al sistema di riferimento;
- vengono numerate le aste in ordine crescente da 1 a n da quella con una cerniera legata al sistema di riferimento fisso fino all'ultima legata ad una sola;
- viene definita *asta precedente* di un'asta i , l'asta $i-1$;
- vengono definite *aste carico* di un'asta i , le aste da $i+1$ a n ;
- ogni volta che un'asta viene ruotata, si considera l'insieme costituito dall'asta i e dalle sue aste carico che ruota rigidamente rispetto all'asta precedente alla i .

4.3 Spiegazione dell'algoritmo

L'idea alla base di questo metodo di ottimizzazione è “quanto più l'asta è vicina all'origine, meno deve ruotare”. Essenzialmente si basa sul concetto di “stressare” il meno possibile i motori che movimentano il sistema, in quanto ogni asta ruotando deve “sostenere” il peso e l'inerzia delle aste rimanenti unicamente legate ad essa. Per fare un semplice esempio, si considera l'asta più lontana dalla cerniera: farla ruotare

rigidamente di 1° richiede energia necessaria solo per un'asta. Se invece si prende la penultima asta, per farla ruotare rigidamente di 1° , si ha bisogno dell'energia necessaria per ruotare rigidamente due aste (ultima e penultima). Con la terzultima asta energia per tre e così in avanti.

Si sottolinea di nuovo che l'algoritmo si basa sull'idea di spostamento *point-by-point*, ovvero a partire da una formazione iniziale si cerca di raggiungere una formazione finale in cui l'ultima asta ha l'estremità non vincolata coincidente con il punto obiettivo. Per realizzare questo obiettivo ho pensato al seguente metodo.

Passo 0

- 1) Viene data la matrice della formazione iniziale.
- 2) Si parte dall'ultima asta (asta n), vincolando le rimanenti aste.
- 3) Si costruisce una circonferenza di raggio pari alla lunghezza dell'asta n , con centro nell'estremità vincolata di questa.
- 4) Si costruisce una circonferenza di raggio nullo, con centro nel punto obiettivo.
- 5) Si cercano le intersezioni tra le due circonferenze.

Si possono verificare due eventi in questo caso: nel caso esistano intersezioni, vuol dire che è sufficiente ruotare l'asta n di un determinato numero di gradi per poter raggiungere il punto, lasciando le rimanenti nelle posizioni originali, nel caso invece in cui non esistano intersezioni, vuol dire che è necessario ruotare almeno un'altra asta affinché si possa raggiungere il punto obiettivo. In quest'ultimo caso si passa al passo successivo.

Passo 1

- 1) Si considera l'insieme delle aste da 1 a $n-1$, ovvero scartando l'ultima asta.
- 2) Si costruisce una circonferenza di raggio pari alla lunghezza dell'asta $n-1$, con centro nell'estremità vincolata di questa.

- 3) Si costruisce una circonferenza di raggio pari alla somma delle lunghezze delle aste scartate (in questo caso solo l'asta n), con centro nel punto obiettivo.

Di nuovo si possono verificare due casi (esistenza e non esistenza di intersezioni). Per semplicità di spiegazione supponiamo che non esistano intersezioni; ciò vuol dire che qualsiasi combinazione di rotazioni delle due aste n e $n-1$ non è sufficiente per il raggiungimento del punto obiettivo senza muovere perlomeno un'altra asta. Si passa perciò al passo successivo.

Supponiamo che non esistano intersezioni fino al passo k , ovvero fino a considerare solo le aste da 1 a $n-k$. Se esistono delle intersezioni fra la circonferenza costruita sulle aste e quella costruita sul punto obiettivo, significa che questi è raggiungibile, perlomeno con una formazione. Inoltre il punto obiettivo è raggiungibile se l'asta $n-k$ avesse la sua estremità libera in uno qualsiasi dei punti dell'arco evidenziato dalle intersezioni. Chiaramente per essere ottimizzato deve essere il punto più vicino all'attuale posizione dell'estremità, perciò l'asta $n-k$ ruota fino a raggiungere il suddetto punto.

Passo k

- 1) Si considera l'insieme delle aste da 1 a $n-k$.
- 2) Si costruisce una circonferenza di raggio pari alla lunghezza dell'asta $n-k$, con centro nell'estremità vincolata di questa.
- 3) Si costruisce una circonferenza di raggio pari alla somma delle lunghezze delle aste scartate, con centro nel punto obiettivo.
- 4) Si considera l'intersezione più vicina, ovvero quella che richiede un angolo di rotazione minore per l'asta $n-k$, oppure l'unica intersezione nel caso in cui esista un solo punto in comune tra le due circonferenze.
- 5) Si fissa la formazione delle $n-k$ aste, considerate quindi per i passi successivi come fisse.

La formazione attuale risulta essere quindi la formazione finale delle aste comprese tra 1 e $n-k$. Si aggiunge di nuovo l'asta $n-k+1$; ovvero si ripete il passo $k-1$, raggiungendo l'ultima asta eliminata (asta $n-k+1$) nella stessa posizione relativa all'asta $n-k$.

Ripetendo il passo precedente, questa volta si ha la certezza che esisteranno delle intersezioni, poiché il punto obiettivo rientra nell'area raggiungibile con la nuova formazione. Procedendo così a ritroso fino al passo 0 si ottiene la formazione finale per il punto obiettivo. La formazione alla fine del passo 0 risulta infine essere la formazione finale del sistema.

5 STUDIO DI SOLUZIONE OTTIMA

5.1 Valutazione dell'ottimo

La linea logica da seguire per considerare l'algoritmo come ottimo è un flusso ad esclusione: il punto obiettivo è raggiungibile muovendo una sola asta? Ovvero, è nella circonferenza di azione dell'ultima asta? Se sì, è sufficiente ruotare quell'asta. Se no, è nel fascio di circonferenze di azione delle ultime due aste? Se sì, l'asta più vicina all'origine viene spostata il minimo possibile al fine di far rientrare il punto nella circonferenza di azione dell'asta rimanente. Se no, è nel fascio di tre aste?... E così si prosegue fino all'asta legata all'origine.

Chiaramente se a priori il punto obiettivo non rientra nella raggio di azione massimo, ovvero lo spazio interno alla circonferenza di raggio pari alla somma delle lunghezze delle aste, allora la soluzione non esiste.

5.2 Razionale

Si considera il punto obiettivo di coordinate $P_f = (x_f \ y_f)$ e la sua distanza dall'origine $d_f = \sqrt{x_f^2 + y_f^2}$. Se si definisce $R_{max} = \sum_{i=1}^n l_i$ il raggio della circonferenza che delimita lo spazio di azione, si possono ottenere i seguenti casi:

- 1) $d_f > R_{max} \rightarrow \nexists$ soluzioni al problema
- 2) $d_f \leq R_{max} \rightarrow \exists$ almeno una soluzione al problema

Si considera il secondo caso. Ricordando la definizione di P_i di α_i e definendo l'operatore $dist(P_k \ P_h)$ come la distanza tra due punti, si procede al passo 0.

Passo 0

- 1) Si calcola $d_n = dist(P_{n-1} \ P_f)$.
- 2) $d_n = l_n \rightarrow$ si ruota l'asta n di un angolo pari a $\beta_n = \min(\widehat{P_n P_{n-1} P_f})$
 \rightarrow algoritmo terminato.
- 3) $d_n \neq l_n \rightarrow$ passo 1.

Il passo generico $k > 0$ è differente dal passo 0 ma segue l'algoritmo generalizzato, per cui si darà di seguito illustrazione solamente del generico passo k .

Passo k

- 1) Si definisce la circonferenza base dell'asta $n-k$ come la circonferenza di raggio pari a l_{n-k} e centro in P_{n-k-1} .

$$C_{n-k} = \{P = (x \ y) : (x - x_{n-k-1})^2 + (y - y_{n-k-1})^2 = (l_{n-k})^2\}$$

- 2) Si definisce il cerchio residuo dell'asta $n-k$ come il luogo dei punti interni alla circonferenza di raggio pari a $\sum_{i=n-k+1}^n l_i$ e centro in P_f .

$$F_{n-k} = \left\{ P = (x \ y) : (x - x_f)^2 + (y - y_f)^2 \leq \left(\sum_{i=n-k+1}^n l_i \right)^2 \right\}$$

- 3) Si studia l'insieme intersezione $I_{n-k} = C_{n-k} \cap F_{n-k}$:

- $I_{n-k} = \emptyset \rightarrow$ passo $k+1$
- $I_{n-k} \neq \emptyset \rightarrow$ si procede come segue

- a. si costruisce l'insieme degli angoli formati dal punto P_{n-k} , dal punto P_{n-k-1} e dai punti appartenenti all'intersezione (ovvero all'arco della circonferenza C_{n-k} interno al cerchio F_{n-k})

$$B_{n-k} = \{\beta \in R : \beta = P_{n-k} \widehat{P_{n-k-1} P} , \quad \forall P \in I_{n-k}\}$$

- b. come precedentemente spiegato, si sceglie il minore degli angoli evidenziati

$$\beta_{n-k} = \min B_{n-k}$$

- c. si fa ruotare l'asta $n-k$ del suddetto angolo e si fissa così la formazione finale delle prime $n-k$ aste

$$[\alpha_1 \ \alpha_2 \ \dots \ \alpha_{n-k-1} \ \alpha_{n-k} + \beta_{n-k}]$$

- d. si aggiunge l'asta $n-k+1$ (mantenendo l'angolo α_{n-k+1}) e si ripete il passo $k-1$.

L'algoritmo si ferma esclusivamente alla voce 2 del passo 0, cioè quando la formazione finale delle prime $n-1$ aste permette al punto obiettivo di appartenere alla circonferenza base C_n .

6 APPLICAZIONE AL SISTEMA-ESEMPIO

Viene fornita di seguito una spiegazione del metodo applicata al sistema-esempio.

Passo 0: viene costruita la circonferenza base dell'asta 5 e il cerchio residuo di raggio nullo.

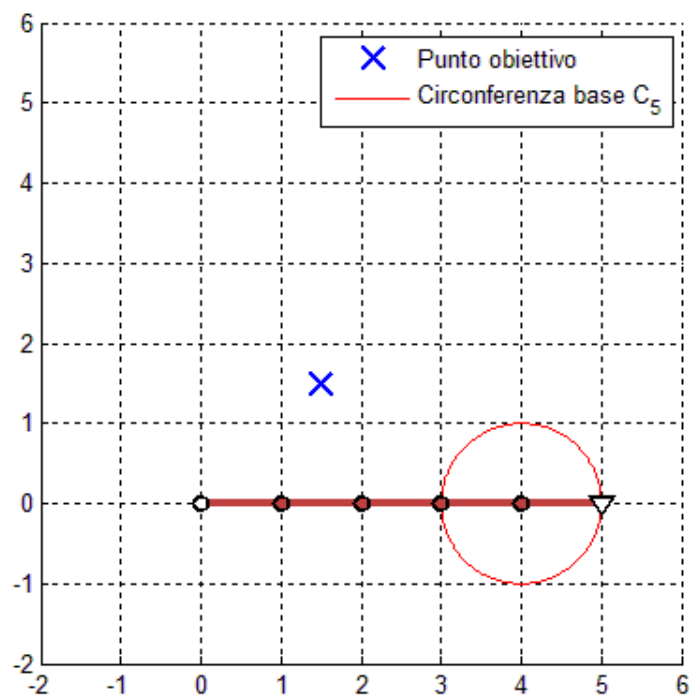


Fig.11 Passo 0 nel sistema-esempio

Risulta che non esistono intersezioni fra i due. Quindi si prosegue con il prossimo passo.

Passo 1: è considerato il sottosistema delle prime 4 aste e vengono costruite la circonferenza base e il cerchio residuo dell'asta 4.

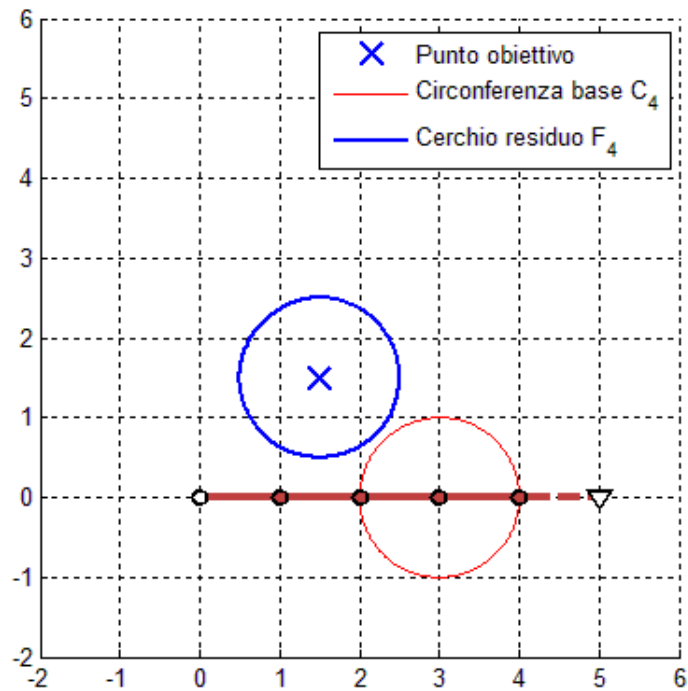


Fig.12 Passo 1 nel sistema-esempio

Anche in questo caso non esistono intersezioni, per cui si prosegue con l'algoritmo.

Passo 2: costruiti la circonferenza base e il cerchio residuo si evidenziano immediatamente le intersezioni che delimitano l'arco di circonferenza per cui esistono delle soluzioni.

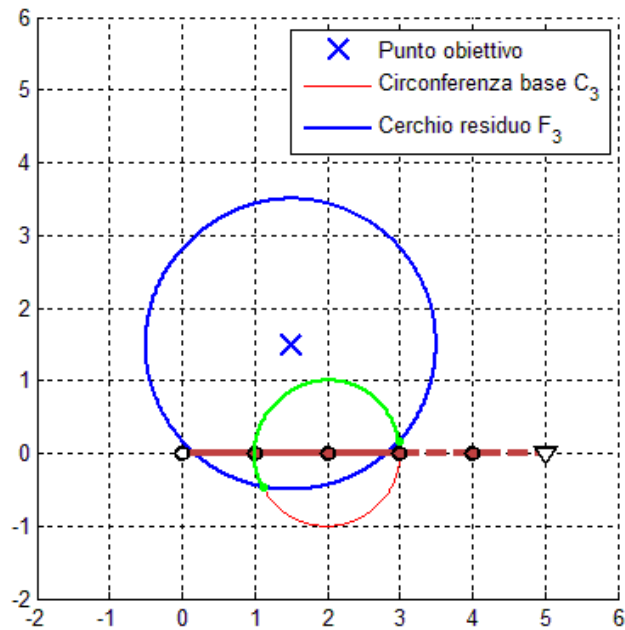


Fig.13 Passo 2-a nel sistema-esempio

Coerentemente ai ragionamenti precedenti viene scelto il punto di intersezione che causa la minor rotazione possibile all'asta. Una volta effettuata la rotazione si ripete il passo 1 considerando l'attuale formazione delle prime 3 aste.

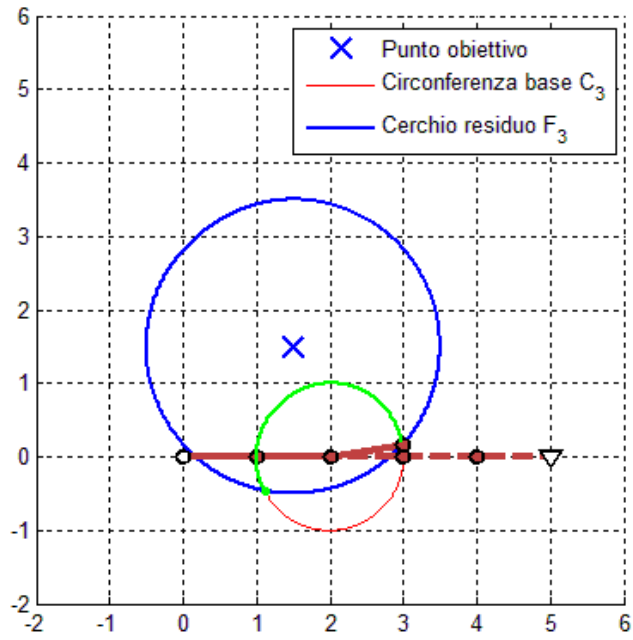


Fig.14 Passo 2-b nel sistema-esempio

Passo 1bis: costruiti di nuovo la circonferenza base e il cerchio residuo, si ottiene un'unica intersezione.

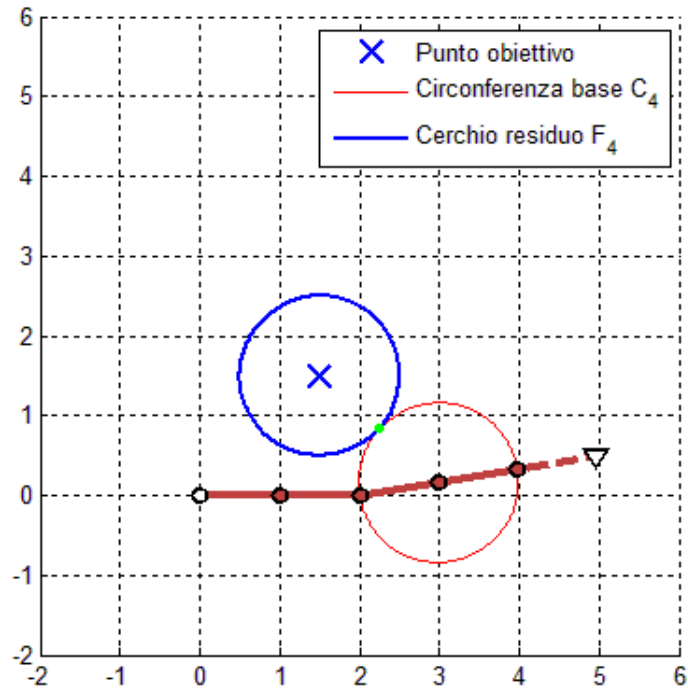


Fig.15 Passo 1bis-a nel sistema-esempio

Si ruota l'asta 4 fino a far coincidere l'intersezione con l'estremità di quest'ultima.

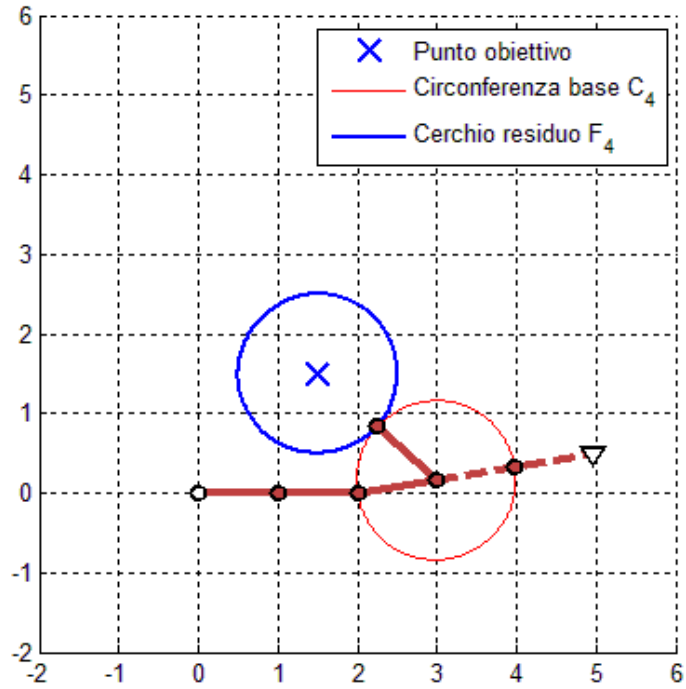


Fig.16 Passo 1bis-b nel sistema-esempio

Passo 0bis: si conclude l'algoritmo constatando che il punto obiettivo giace sulla circonferenza base dell'asta 5, come doveva verificarsi.

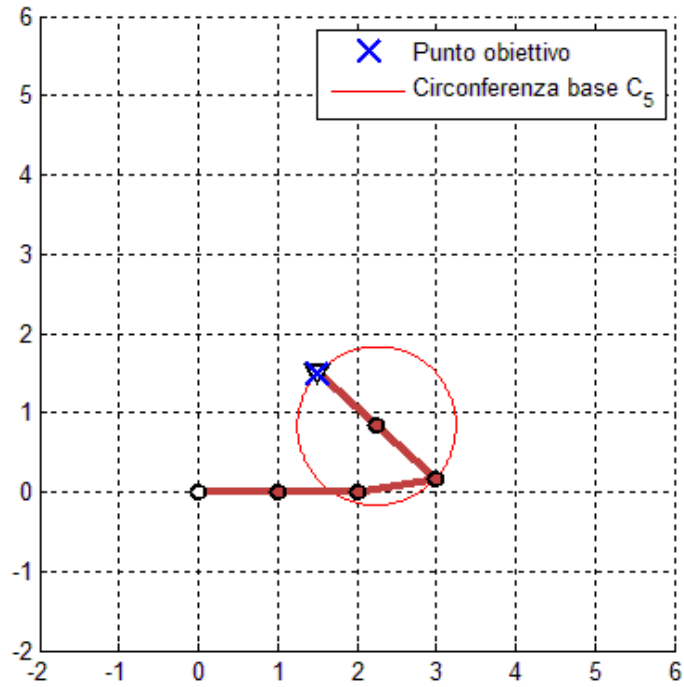


Fig.17 Passo 0bis nel sistema-esempio

In conclusione si ottiene la formazione finale per il punto obiettivo considerato.

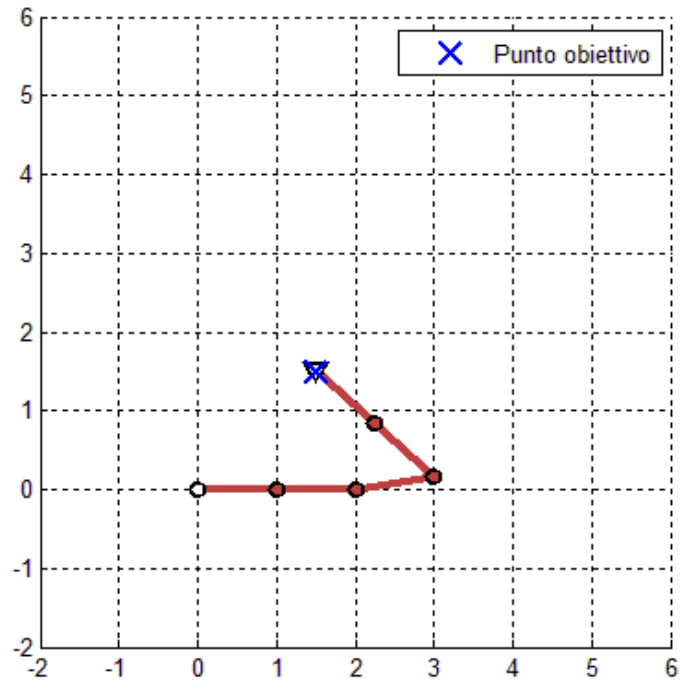


Fig.18 Soluzione ottima nel sistema-esempio

7 ESECUZIONE DELL'ALGORITMO E APPLICAZIONI TIPO

7.1 Estremi del raggio di azione

Si vuole far agire il robot in alcuni degli estremi del raggio di azione massimo, ad esempio

$$(5l \ 0) \quad (0 \ 5l) \quad (-5l \ 0) \quad (0 \ -5l)$$

e si studia l'effettivo tragitto dell'estremità libera. Con il metodo della distanza minima si sarebbe descritto un rombo avente come vertici i punti obiettivo, ma ciò nel complesso avrebbe significato un numero di rotazioni nelle cerniere superfluo rispetto al dovuto.

Come si può intuire la soluzione ottima in questo caso risiede nella sola rotazione della prima asta che a sua volta trascina con se le successive. Difatti l'algoritmo proposto fornisce come soluzione la circonferenza passante per i punti di estremità, ovvero la circonferenza che delimita il raggio di azione massimo (Figura 19).

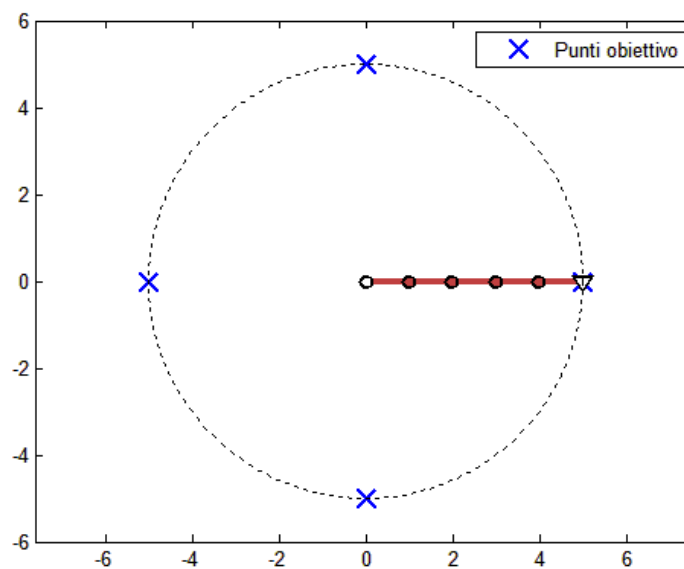


Fig.19 Raggio di azione massimo

7.2 Applicazione tipo: disposizione e prelievo di elementi

Una delle applicazioni tipiche per questo tipo di robot è il deposito o il prelievo di elementi o oggetti in punti caratteristici del processo in cui sono coinvolti. Si pensi, ad esempio, ad un robot SCARA che deve disporre una serie di viti su un elemento maggiore; oppure deve collocare in sequenza diversi pezzi di assemblaggio.

La Figura 20 mostra un esempio di deposito di elementi per un *pattern* semplice.

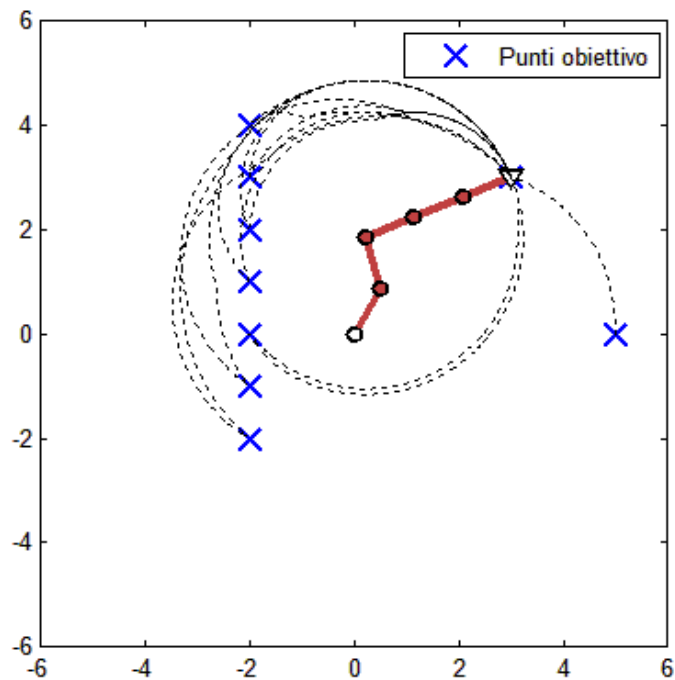


Fig.20 Disposizione di elementi

7.3 Applicazione tipo: deposito e azione di elementi

Una ulteriore applicazione tipica è nel caso in cui nell'estremità libera sia presente uno strumento di azione propria, ovvero nel caso in cui il robot, una volta lasciata la posizione iniziale, non dipenda da un punto fisso per l'esecuzione della sua missione,

ma necessiti solo dei punti obiettivo. Ad esempio si consideri un robot che deve avvitare dei bulloni tra due lamiere in sequenza oppure debba saldare dei punti ben specifici di un elemento più grande; sono innumerevoli e forse più comuni casi di questo genere, e possono avere applicazioni in svariati campi, dal montaggio industriale alla medicina chirurgica. È necessario ricordare che presentano nell'estremità libera degli elementi atti all'esecuzione del compito nei punti prefissati.

Nella Figura 21 si mostra ad esempio un robot che deve avvitare una serie di bulloni disposti in doppie file.

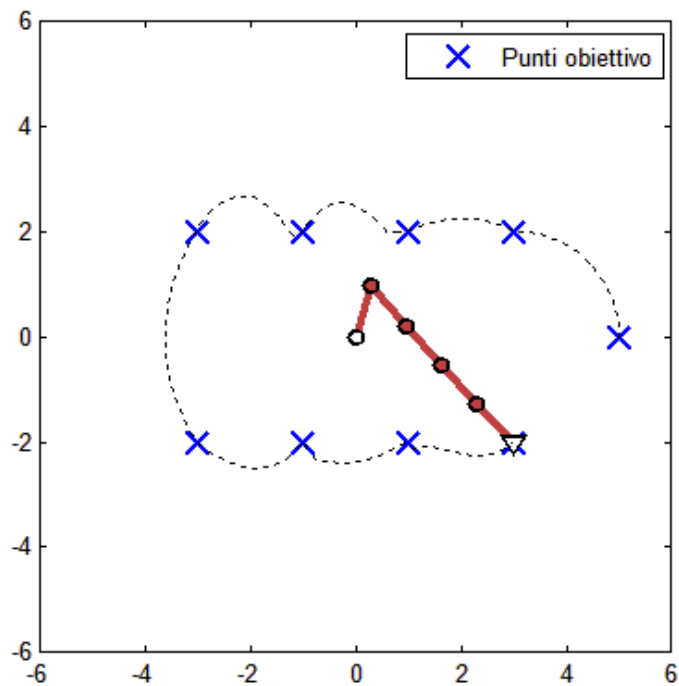


Fig.21 Deposito di elementi

7.4 Disegno e tracciamento

Si può constatare dai precedenti esempi come il tragitto dell'estremità libera non sia regolare e non presenti un aspetto pratico in caso di tragitto breve tra i punti; quindi nel caso si volesse far eseguire una traiettoria continua assegnata all'estremità libera (come

ad esempio nel caso di processi come un disegno al laser o un'incisione) si potrebbe supporre che l'algoritmo sia incoerente con questo caso.

Questa supposizione risulta errata perché l'algoritmo ha insito il principio di ottimizzazione, per cui se un punto obiettivo è molto prossimo al seguente il tragitto fornito dal metodo tra i due risulta con buona approssimazione un segmento.

Supponiamo ad esempio che si voglia disegnare una forma quadrata; si forniscono in primo tempo solo i vertici come punti obiettivo. Il risultato, mostrato in Figura 22, è evidentemente lontano dalla soluzione desiderata.

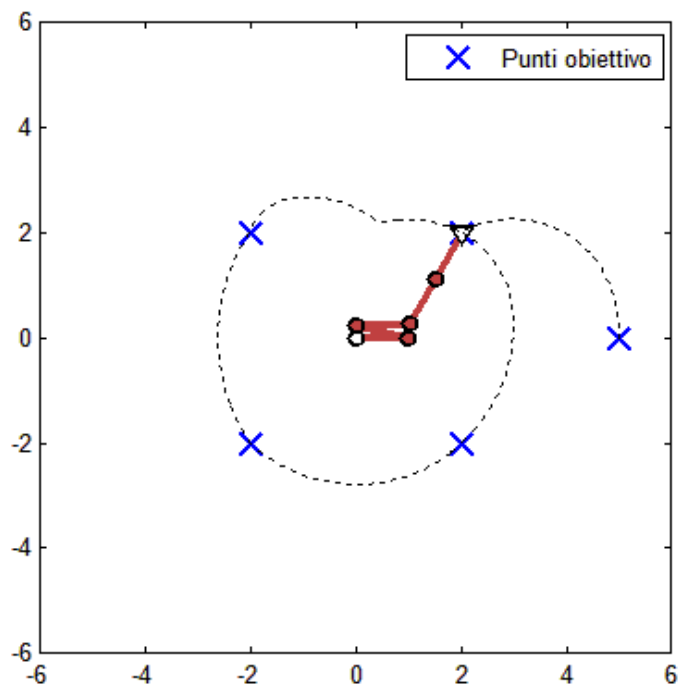


Fig.22 Disegno di quadrato: fase 1

Tuttavia se si prova ad infoltire il numero di punti obiettivo, tutti comunque appartenenti alla forma desiderata, si scopre che il tragitto effettivo piano piano risulta combaciante e approssimato con il quadrato finale (Figure 23 e 24).

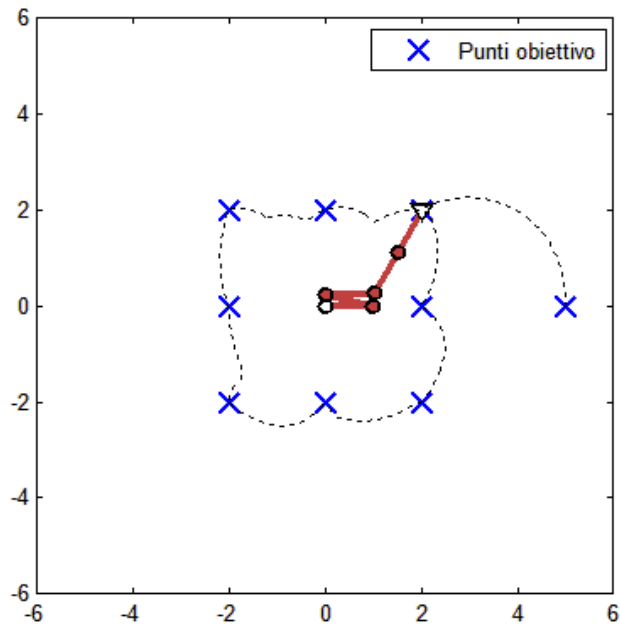


Fig.23 Disegno di quadrato: fase 2

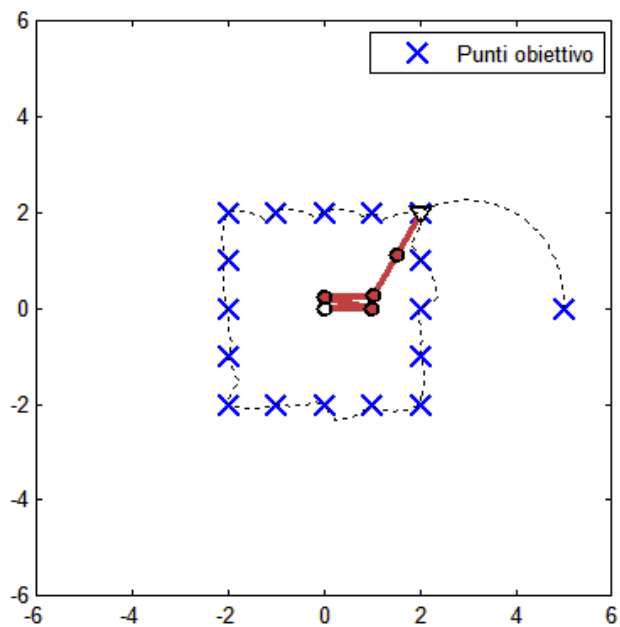


Fig.24 Disegno di quadrato: fase 3

In conclusione se si fornisce un numero pressoché elevato di punti obiettivo il tragitto finale sarà esattamente la forma considerata. Nella Figura 25 viene mostrato il risultato ottenuto fornendo 36 punti del quadrato (per semplicità non verranno riportati i punti forniti).

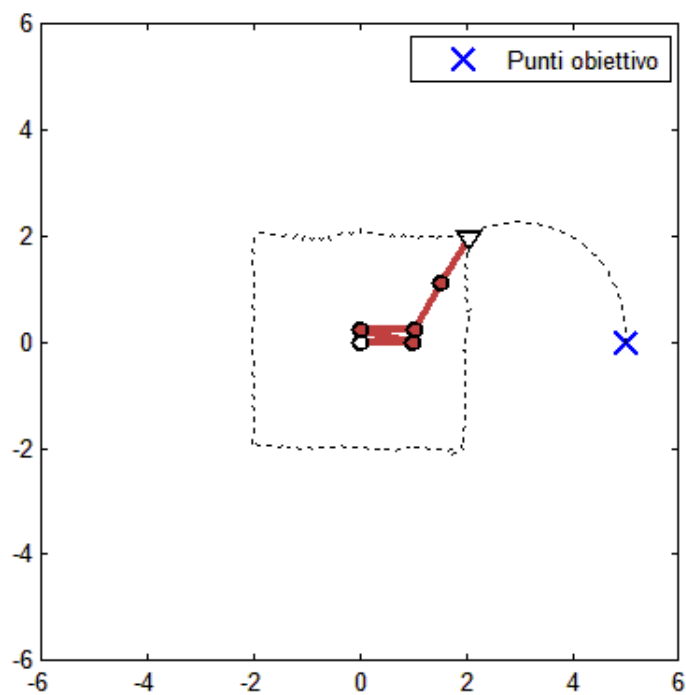


Fig.25 Disegno di quadrato: fase 4

8 SVILUPPI E RISULTATI

8.1 Risultati elaborati

Di seguito verranno rappresentati gli andamenti temporali di alcune variabili del sistema-esempio a cui vengono forniti 5 punti random e l'obbligo di tornare al punto di partenza iniziale. La Figura 26 mostra il tragitto completo del sistema.

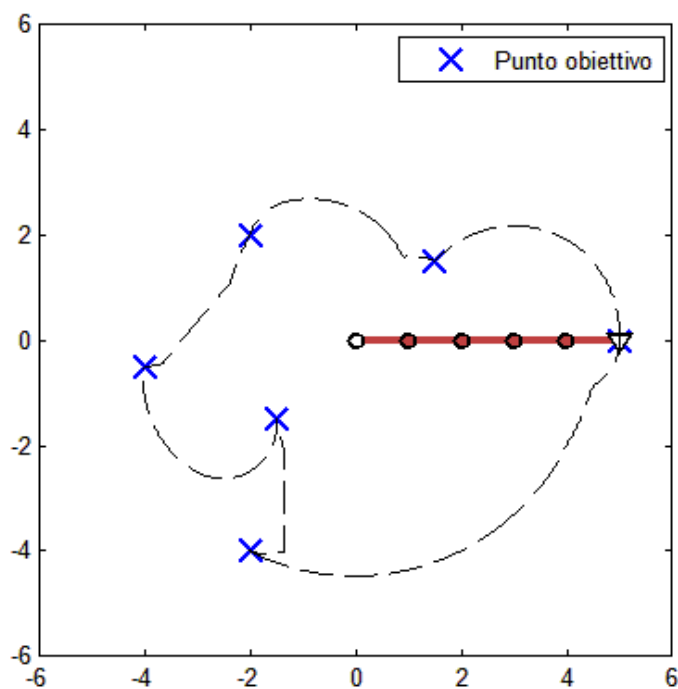


Fig.26 Simulazione 1: tragitto completo

Le Figure da 27 a 31 mostrano invece l'andamento temporale delle rotazioni assolute delle aste (grafico in alto) insieme agli impulsi di coppie agenti in ogni cerniera durante l'esecuzione (grafico in basso). La presenza degli impulsi é dovuta al fatto che le cerniere ne necessitano per iniziare la rotazione ma soprattutto per fermarla una volta raggiunto il punto obiettivo; in ciò si includono le rotazioni di trascinarsi delle aste che ognuna ha a carico. Difatti si può notare come per le aste più vicine all'origine gli impulsi richiesti risultino maggiori e come vengono limitate le rotazioni delle stesse, in

linea con l'obiettivo dell'algoritmo; allo stesso modo si nota che ad impulsi minori delle aste più lontane dall'origine corrispondono rotazioni maggiori.

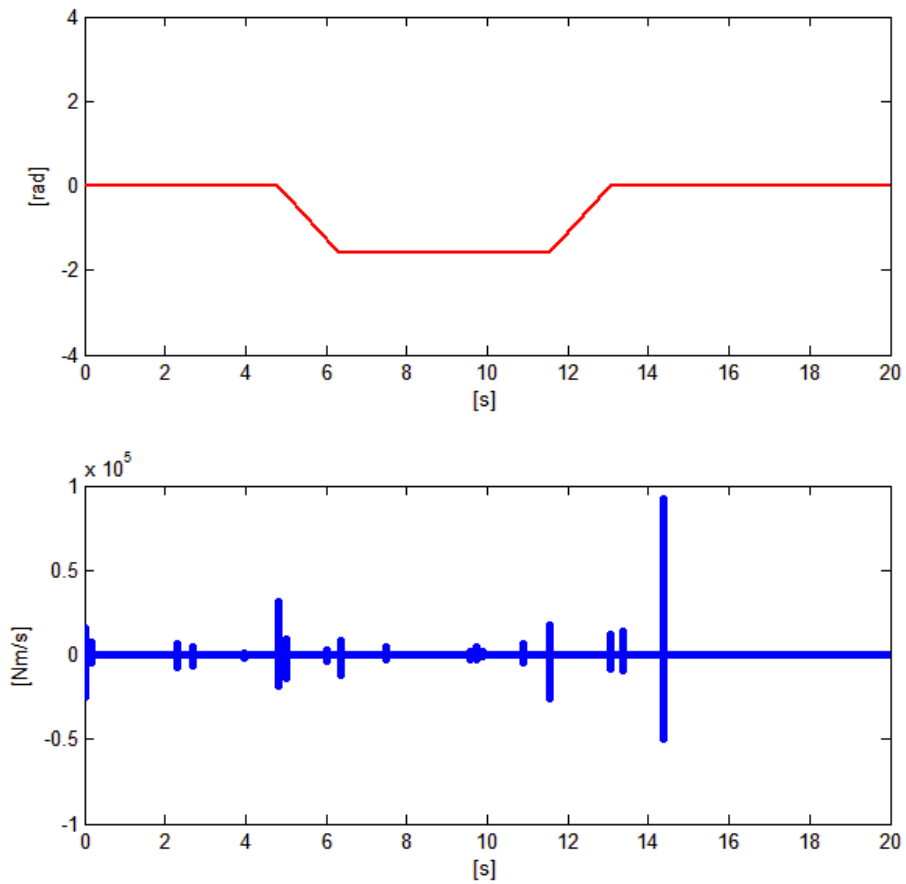


Fig.27 Simulazione 1: asta 1

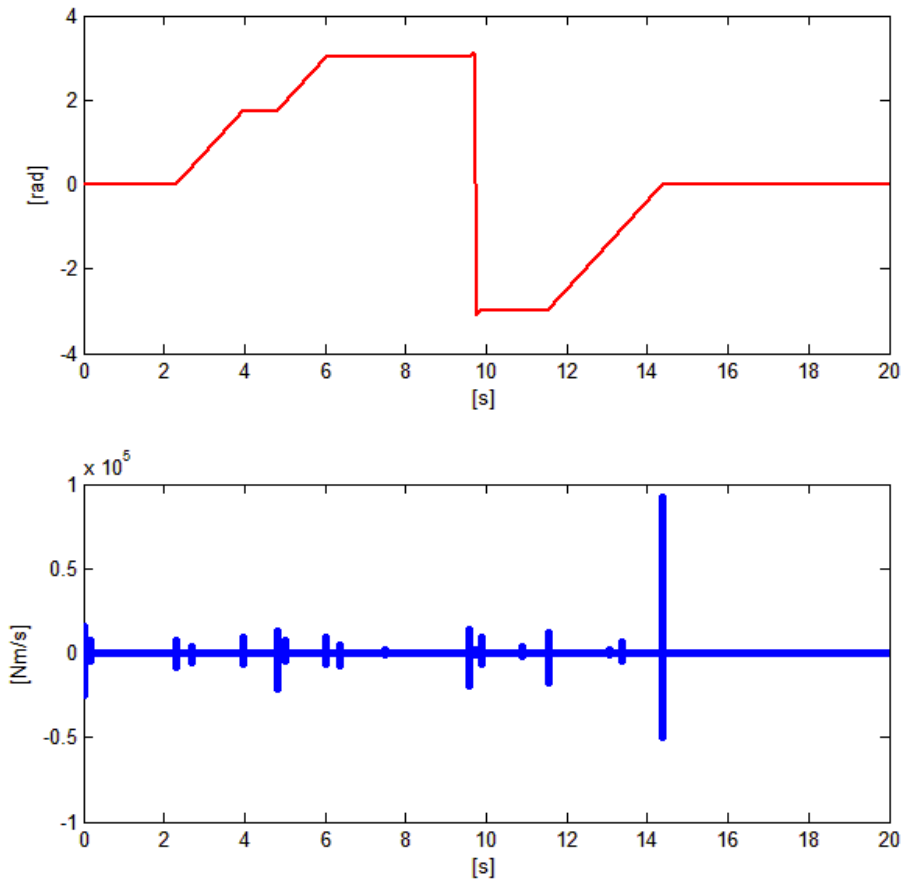


Fig.28 Simulazione 1: asta 2

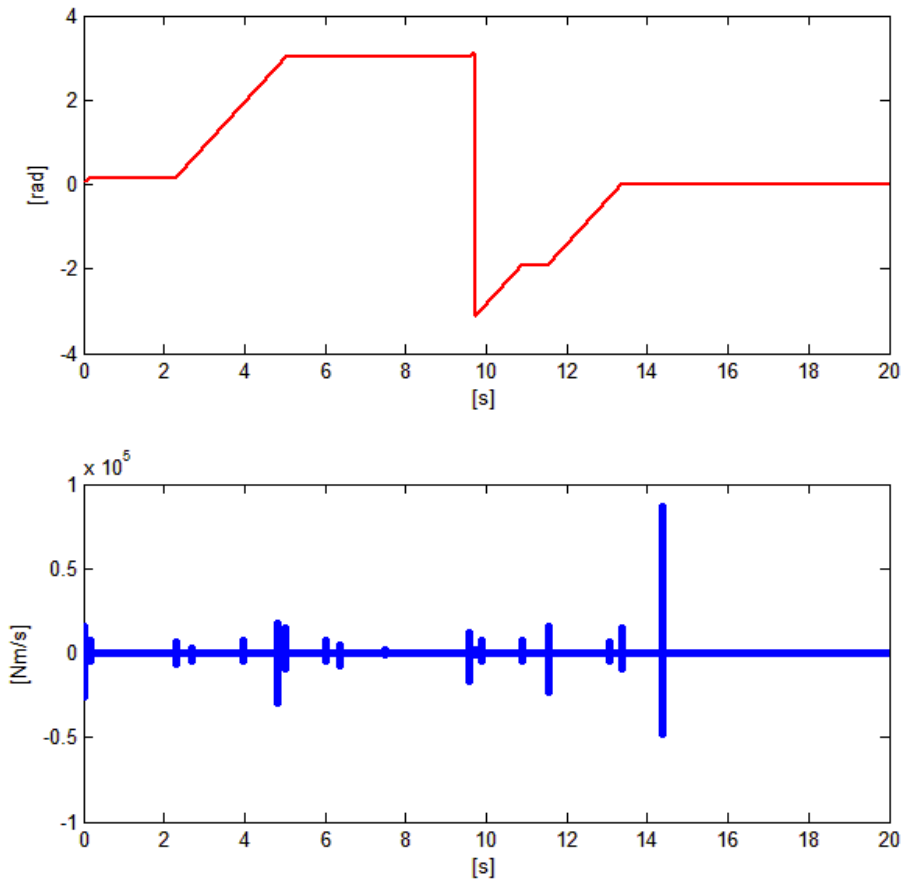


Fig.29 Simulazione 1: asta 3

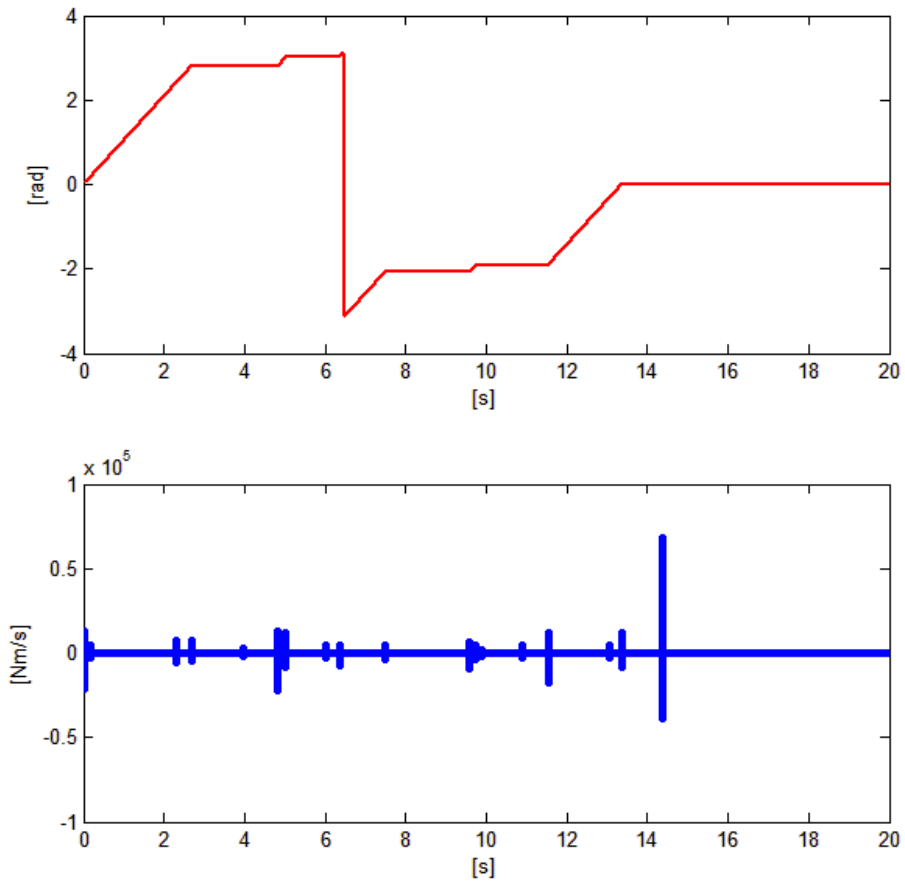


Fig.30 Simulazione 1: asta 4

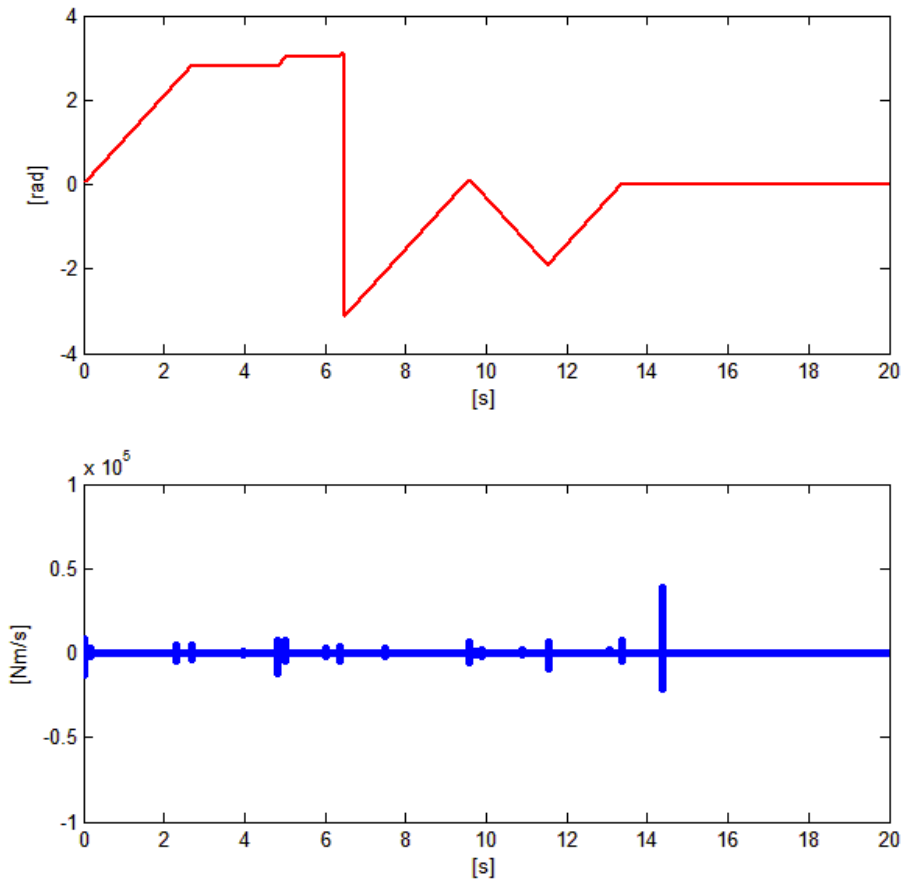


Fig.31 Simulazione 1: asta 5

Per fornire una rappresentazione migliore dell'andamento temporale delle coppie viene fornito un grafico in scala ridotta nelle Figure da 32 a 36, escludendo i valori di impulso; i grafici risultano quindi decisamente irregolari, ma vengono confrontati con i valori di coppia della soluzione che si ottiene per gli stessi punti obiettivo attraverso il metodo studiato e pubblicato dai Prof. i Morandini, Msarati, Bargigli e Vaccani nell'articolo *"Feedforward Control Design From General-Purpose Multibody Analysis for an Original Parallel Robot Concept"*.

La prima macro-differenza appare nell'esecuzione della missione: viene infatti effettuato un tragitto differente tra le due soluzioni, risaltando la maggior continuità di quella proposta nell'articolo. Può essere considerato un punto a sfavore dell'algoritmo proposto, in quanto minore è l'oscillazione dell'input fornito ai motori, minore risulta lo stress subito dagli stessi. Per quanto riguarda i valori di coppia in modulo, invece, appaiono delle congruenze nelle applicazioni delle prime aste, quelle più vicine all'origine, mentre sono più conservativi i valori forniti dall'algoritmo proposto alle ultime due cerniere. Si può dedurre quindi che, a parità di missione, la soluzione trovata in questa tesi rientra tra le soluzioni ammissibili ed è ben paragonabile con altre soluzioni di ottimo.

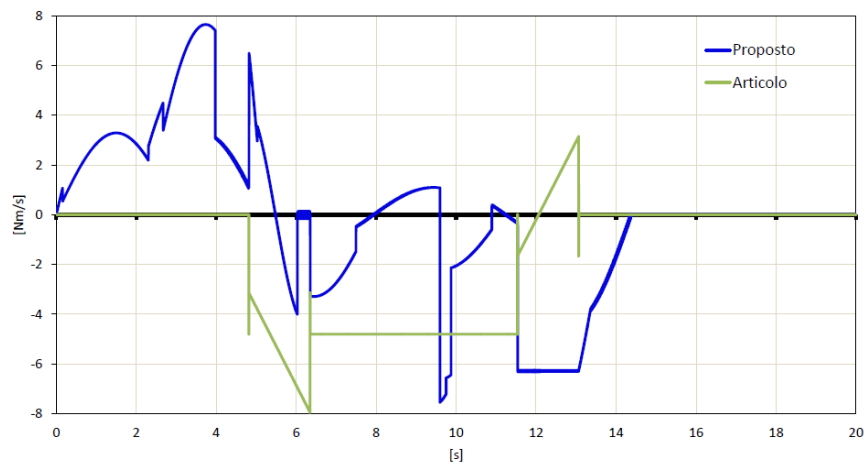


Fig.32 Simulazione 1: asta 1 - confronto

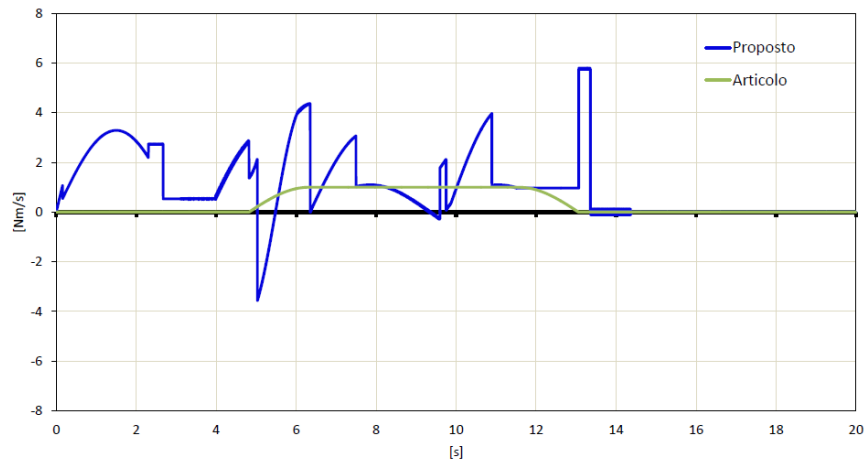


Fig.33 Simulazione 1: asta 2 - confronto

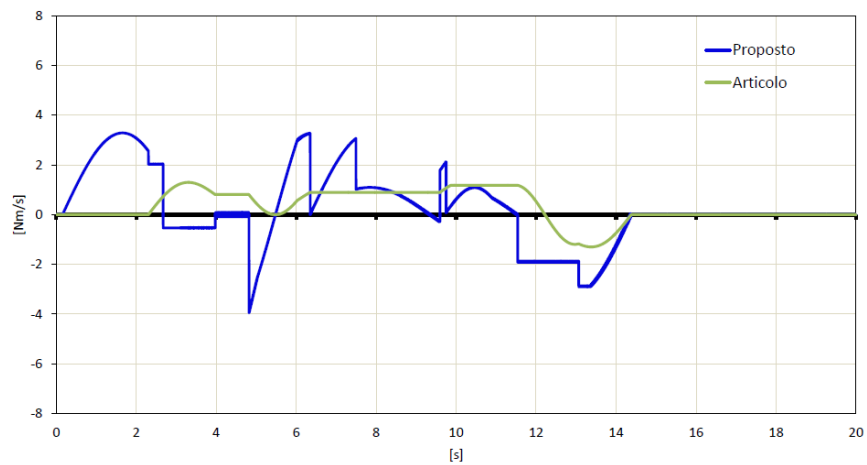


Fig.34 Simulazione 1: asta 3 - confronto

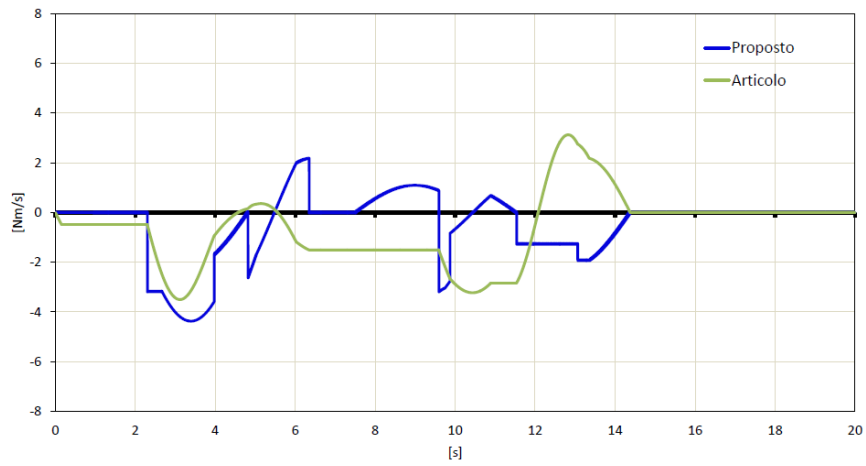


Fig.35 Simulazione 1: asta 4 - confronto

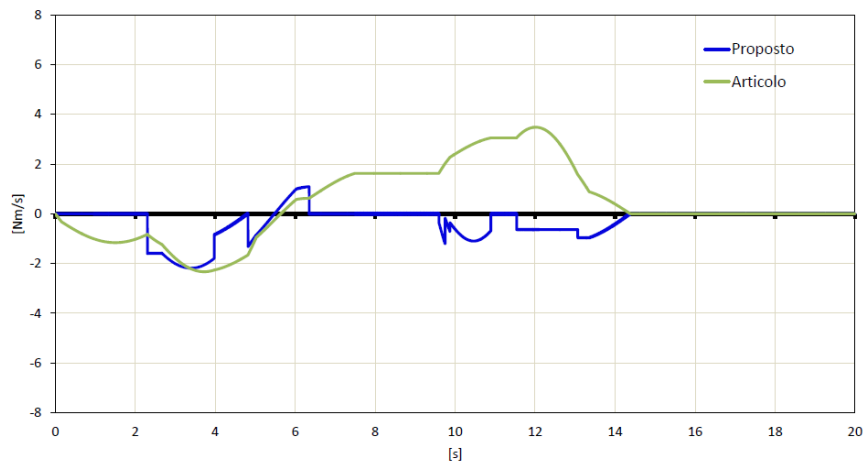


Fig.36 Simulazione 1: asta 5 - confronto

8.2 Generalizzazione

Il metodo presentato può essere generalizzato al fine di proporre uno schema logico di ottimizzazione per un sistema per cui si chiede un'ottimizzazione di tipo *the simpler, the better* su insiemi di elementi con stesso obiettivo e stessi gradi di libertà.

Ciò che si vuole evidenziare è la possibilità di realizzare algoritmi per sistemi e problemi che non necessitano solo di una soluzione ottima ma hanno bisogno della più semplice fra le soluzioni ottime.

Si prenda ad esempio la situazione seguente: esiste un gruppo di n persone che devono trasportare un numero indefinito di elementi da un punto A ad un punto B; le persone sono già in formazione a catena, equamente disposti tra i due punti. Si cerca di ottimizzare il trasporto totale degli elementi sotto ogni aspetto (minor tempo impiegato e minor fatica effettuata), ma si tiene presente che alterare la posizione iniziale di due persone non altererebbe il sistema poiché le proprietà di ognuno sono considerate uguali. Si comprende che passando gli oggetti da una persona all'altra, come una catena di montaggio, si ottiene una soluzione ottima. Si supponga quindi che invece di arrivare alla persona più vicina al punto B, gli elementi si fermino alla penultima persona e questa li porti personalmente al punto finale. Si supponga anche che la differenza di tempo nei due tipi di soluzione sia trascurabile.

La convenienza nell'usare il secondo tipo sta nel non usufruire di una persona della catena, cioè di usare $n-1$ strumenti del sistema. Si tratta quindi di usare un numero minore di persone, trascurandone una non necessaria e ottenendo una soluzione comunque ottima.

La differenza trascurabile nel tempo di esecuzione tuttavia può considerarsi tale fino ad un numero $n-k$ di persone utilizzate. In conclusione si ottiene una soluzione ottima ma sfruttando le risorse del sistema il minimo possibile, non in termini di energia ma di mezzi di esecuzione.

Generalizzando si arriva a definire un algoritmo *the simpler, the better* attraverso una idea basilare: partendo dall'obiettivo di missione del sistema, può il primo strumento fornire una soluzione ottima da solo? Se no, possono i primi due? Infine qual è il numero minimo necessario di strumenti affinché si ottenga una soluzione e per di più ottima?

L'algoritmo proposto suggerisce di fermarsi alla prima domanda e in caso di risposta negativa:

- 1) considerare il sottosistema costituito dall'esclusione del primo elemento;

- 2) ridefinire come obiettivo della missione la sfera delle soluzioni in cui sarebbe necessario utilizzare solo il primo elemento;
- 3) verificare che si riesca a raggiungere l'obiettivo.

In caso di ulteriori risposte negative, si procede con i successivi sottosistemi fino a trovare la soluzione ottima.