ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Dipartimento dell'Energia e dell'Informazione

Automation Engineering

Relazione Finale attività di Tirocinio

svolta presso

Alma Mater Studiorum, Università di Bologna

Dipartimento di Ingegneria Industriale (DIN)

Analisi cinematica della mano per lo sviluppo di un esoscheletro riabilitativo

Presentata da

Marco Barry

Tutor Accademico

Prof. Nicola Sancisi

Referente della Struttura ospitante

Prof. Nicola Sancisi

Anno Accademico 2023/2024

Indice

Introduzione

Questo elaborato è il prodotto di un tirocinio curriculare da 3-CFU, svolto sotto la supervisione di Prof. Nicola Sancisi e tutela di Dott. Cosimo Fonte. Il lavoro svolto segue la tesi per laurea triennale in Ingegneria dell'Automazione di Sara Querzé, titolata *"Definizione del modello cinematico di un esoscheletro per la riabilitazione della mano basato su misure sperimentali di movimento",* da cui si è partito ricalcolando i risultati trovati, per poi svolgere ulteriori manipolazioni e raggruppamenti di dati.

I dati sperimentali, misurati durante lo svolgimento della tesi sopra citata, assieme al modello cinematico della mano ideato da Dott. Fonte, permettono di risolvere il problema della cinematica inversa. La soluzione della cinematica inversa sono i dati iniziali per le elaborazioni e prove statistiche fatte durante questo tirocinio. Di fatto, si tratta di coordinate angolari delle coppie cinematiche della mano, cioè, gli angoli delle articolazioni tra metacarpo, falange, falangetta e falangina.

Lo scopo del tirocinio era di continuare a svolgere e ampliare il risultato già trovato, avendo campo libero a pensare in quale direzione sviluppare l'argomento. Dopo la riproduzione dei risultati della tesi, la quale termina con una analisi delle componenti principali, si è deciso di validare l'approccio, calcolando l'errore effettivo tra gli angoli ed i marker sperimentali e gli analoghi del modello ridotto. Questo ha comportato di ricostruire le posizioni dei marker a partire dal modello, per confrontarli con gli analoghi sperimentali, a seguito di una minimizzazione che determina la configurazione del modello ridotto che meglio approssima la misura sperimentale.

Misurazione, modello cinematico & dataset

Per una spiegazione più dettagliata della teoria anatomica di fondo, della procedura di misurazione in laboratorio e delle trasformazioni e semplificazioni fatte per creare il modello, è bene riferirsi al testo della tesi su cui è basato questo elaborato. I prossimi sottocapitoli inquadreranno in breve questi aspetti.

Attività in laboratorio

Riprese e trasformazione in coordinate spaziali (*Nexus*)

In letteratura ci sono diverse maniere per raccogliere dati sulla cinematica, alcune delle quali con utilizzo di video camere esterne (Fischer, Jermann, Renate, Reissner, & Calcagni, 2020) e altre senza (ACM UbiComp/ISWC 2023, 2023).

Nel caso trattato si utilizzò si utilizzò un sistema stereofotogrammetrico basato su 8 telecamere che acquisiscono in contemporanea la posizione di 29 marker creati ad hoc e posizionati sulle falangi e sul dorso della mano. Questi oggetti di forma sferica furono posizionati e fissati sulla mano (con diverse modalità in base alla posizione) grazie alla forte proprietà riflettenti degli stessi, essi sono considerabili punti luce dagli applicativi di motion-capture. Le posizioni dei marker vengono misurate dai fotogrammi dei video. Dunque, il primissimo formato dato che viene trattato è la posizione del marker nell'immagine 2D, che però non viene mai trattato da noi perché il software di motion-capture *Nexus* (di *Vicon*) tratta i dati presi e li trasforma in coordinate spaziali X-Y-Z.

Da questo è facile capire la mole di dati con cui si tratta, dato che ogni marker ha tre coordinate, una per ogni coordinata spaziale, si ottiene che si hanno addirittura .87 parametri con cui lavorare.

$$m = 3 \cdot 29 = 87 \ \ d.o.f.$$

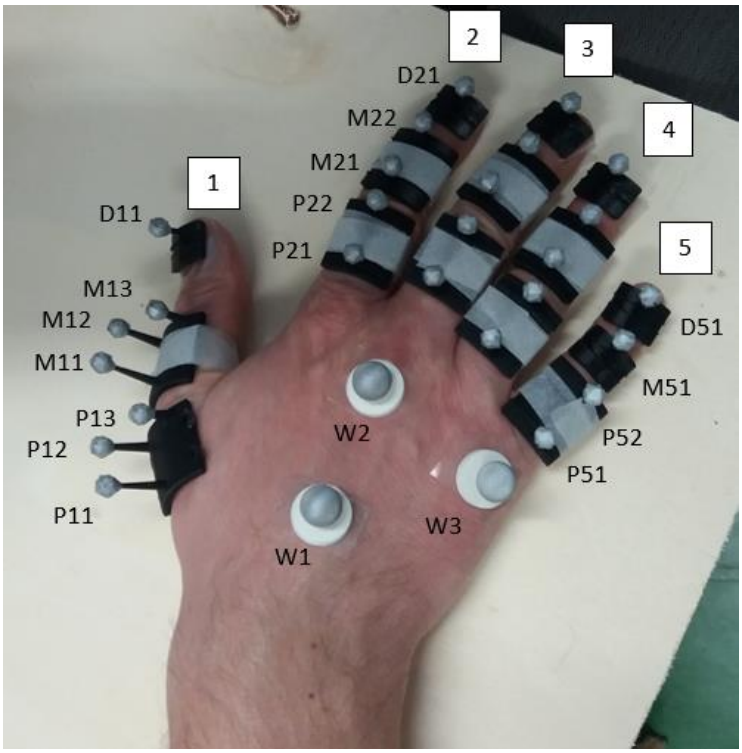


Figura 1 Immagine della tesi. Raffigurata mano con marker, Nomenclatura dei marker definita su Nexus.

Calibrazione (*ImageJ*)

Inoltre, è importante ricordare che al fine di creare un modello cinematico della mano era necessario ottenere in maniera precisa, le grandezze coinvolte. Dunque, al fine di ottenere le dimensioni delle falangi e la distanza tra i giunti, oltre che la posizione dei marker rispetto ad esse, è stato utilizzato il software ImageJ, che come definito sul sito "*is a public domain Java image processing program […] It can calculate area and pixel value statistics of user-defined selections. It can measure distances and angles […] It supports standard image processing functions such as contrast manipulation, sharpening, smoothing, edge detection and median filtering.*" Questo ha permesso la calibrazione e ricavare grandezze da immagini fotografiche di calibrazione.

Modello Cinematico della Mano

In breve, il modello cinematico della mano è frutto dell'unione tra modello geometrico ideale, che tratta la mano un sistema di corpi rigidi e coppie idealizzate (rotoidali e sferiche) e i dati sperimentali ricavati in laboratorio. Come scritto nella tesi di Querzé:

I passaggi da seguire per risolvere questo problema di cinematica inversa ottimizzato sono:

1. Costruire un modello cinematico rigido della mano e dei marker della mano
2. Definire la geometria del modello utilizzando i dati ottenuti dalla calibrazione fotografica
3. Risolvere il problema di cinematica diretta riferito al modello cinematico definito
4. Determinazione degli angoli attraverso un metodo di ottimizzazione numerica
Per approfondire l'argomento si consiglia di vedere operato di Querzé.

Dataset

Questo è il primo capitolo in cui si può dire che è di diretto interesse al tirocinio. Una volta risolto il problema di cinematica inversa e dopo avere ottimizzato, il risultato sono gli angoli dei giunti del sistema falangio-carpale.

Questi a noi sono di interesse perché facilitano la comprensione di dati, innanzitutto, poiché sono dipendenti dalla tipologia di vincolo che esiste tra i corpi. Cioè, in base alla coppia idealizzata che è presente tra due corpi, certi gradi di libertà sono nulli o trascurabili. Un esempio, il giunto tra falange prossimale e mediale dell'indice è pressoché una coppia rotoidale con un solo grado di libertà, che significa che occorre un solo angolo per descriverlo. Questo riduce il numero di coordinate necessarie per descrivere il sistema, rispetto ai dati ricavati dal motion-capturing, oltre a essere forse più intuitivo. In altre parole, il modello cinematico consente già una prima preselezione dei gradi di libertà importanti nel sistema, eliminando quelli secondari che riducono l'interpretabilità dei risultati.

Perciò, considerando ogni falange:

- Pollice, composto da due coppie sferiche e una rotoidale
- Altre falangi, una coppia sferica e due rotoidali

Dunque, il numero di gradi di libertà del sistema è (circa) pari a

$$m = (2 \cdot 3 + 1 \cdot 1) + 4 \cdot (1 \cdot 3 + 2 \cdot 1) = 7 + 20 = 27\ d.o.f$$

Chiaramente il sistema reale è modellabile in diversi modi e quindi il numero di gdl da considerare dipende dal modello cinematico scelto. È bene osservare che sono state considerate anche alcune alternative e quella finale attualmente in fase di analisi non è quella considerata in questo tirocinio, poiché nel frattempo si è cercato di migliorare ulteriormente la preparazione dei dati cinematici. In questo tirocinio si considererà quindi il caso con 27 gdl complessivi.

Inoltre, dato che vennero trattate 15 prese diverse si considerano 15 video-campionamenti, dunque 15 file. Pertanto, il risultato appare sempre come 15 serie di dati. Nel nostro caso abbiamo 15 file o oggetti, con circa 27 variabili indipendenti (righe) e N colonne (dove N sono il numero di punto dato/fotogrammi).[1]Ogni oggetto può essere considerato come distribuzione statistica, oppure si può prendere un sotto-insieme di oggetti da considerare come distribuzione statistica.

I valori degli angoli sono in radianti. È importante da tenere in conto che possibilmente dovuto a errori di misura (per esempio, quando il marker è nascosto nel fotogramma) possono capitare valori spuri o mancanti. Perciò, nelle primissime iterazioni venne applicato un filtraggio dei dati per dare continuità ad essi.

Svolgimento

In questo capitolo verrà spiegato lo svolgimento delle attività, prima genericamente le fasi di attività eseguiti in ordine cronologico, per poi soffermarsi sugli argomenti o problematiche affrontate durante il tirocinio.

[1] N spesso si aggira attorno ai 1600 circa.

Cronologia delle attività

1. *Indicazione date da Prof. Sancisi e breve revisione teorica su PCA e tesi di Sara Querzé:*

In questa iniziale fase fu di grande importanza leggere la tesi, capire le indicazioni date da Prof. Sancisi su cosa fare e dove portare avanti il progetto. Oltre alla lettura dell'elaborato di Sara, è stato importante approfondire gli aspetti legati alla PCA e alla sua applicazione.

2. *Lettura e verifica codice di Sara*:

Una volta consolidata una comprensione sufficiente per affrontare il problema, si è passati alla lettura della analisi compiuta nella tesi. Questo non solo include linee codice, ma anche organizzazione directory, file e dati.

3. *Riproduzione risultati di Sara, con qualche combinazione di presa in più (permutazioni dita):*

Eseguire lo script *MatLab* nelle directory ricevute con modifiche minime per aggiungere qualche combinazione di dati in più, per cui si intende combinazioni di prese e dita prese in considerazione. (Risultati salvati nel file Excel "VarianceCoverageRaggruppamenti.xlsx"). Si aveva provato a ridurre impatto di eventuali outliers, filtrando o togliendo il filtro con pochi risultati. Inoltre, si aveva eseguito una centratura dei dati, che viene già fatto durante la PCA, con ovviamente gli stessi risultati

4. *Modifiche al codice per il calcolo di PCA conseguente a ricezione di nuovo modello di Dott. Fonte:*

Prima comunicazione con il Dott. Fonte, che consegna un modello più recente e quindi dati diversi, da cui si produce nuovamente una analisi PCA. Data però la struttura diversa dei dati (raccolti in strutture) è necessario modificare il codice. Aggiuntive modifiche al codice per renderlo più atomico (scritture file funzione invece di avere un file singolo contente tutto il programma).

5. *Riproduzione risultati Sara ma con dati di modello dinematico aggiornato, alcune combinazioni di prese tralasciate per questioni di tempo, concentrazione su tutte le prese tutte e 5 le dita ("ALL5"):*

Riproduzione dei risultati di Sara coi nuovi dati utilizzando scirpt che raccoglie e genera combinazioni presa e dita in automatico, raccolti in folder "Covariance Coverage" in file *'.mat'*, contenenti tabelle Matlab. Si ha tralasciato situazioni particolari analizzati da Sara di combinazioni particolari di prese (per esempio, "presa 12 e 14, tutte le dita").

6. *Lettura e comprensione "HandModel3" e "Solver" di Dott. Fonte:*

Lettura e comprensione del modello cinematico contenuto in "Solver.m", in quanto per verificare la correttezza del modello a dimensioni ridotte generato da PCA, è necessario calcolare la posizione nello spazio del modello cinematico della mano e confrontare con i dati sperimentali o del modello completo. In questo elaborato, per immediatezza del confronto, si ha utilizzato la posizione dei marker come riferimento.

Di conseguenza, si ha utilizzato la funzione di cinematica diretta presente nel solver e confrontato, scambiando gli angoli già presenti nel modello con quelli approssimati della PCA.

7. *Scrittura codice di minimizzazione in funzione dei coefficienti Z*

Visto il modello complesso, la relazione tra coordinate angolari e posizione spaziale delle falangi non è semplice né lineare. Non necessariamente il modello ottimale in coordinate angolari (risultante da PCA) è ottimale per coordinate spaziali.

Dunque, in questa parte dell'esperienza, si ha scritto codice per minimizzare l'errore quadratico medio della distanza tra marker sperimentali e marker proiettati utilizzando i coefficienti ***Z*** e componenti Principali ***V*** risultanti dalla PCA. In maniera di ottimizzare con il sistema di riferimento ridotto, le pose della mano in tutti gli istanti tempo a presa.

		n° PC										
Raggruppamenti fatti da Querzè	finge	0	1	2	3	4	5	6	7	8	9	10
Ogni singola presa tutte le dita:												
presa1	5	0.000	72.275	91.810	94.725	96.807	97.833	98.533	99.091	99.421	99.586	99.720
presa2	5	0.000	61.715	89.315	93.011	96.201	97.731	98.498	99.013	99.360	99.595	99.704
presa3	5	0.000	57.075	86.809	91.959	95.387	96.633	97.717	98.644	99.227	99.473	99.633
presa4	5	0.000	73.612	91.494	94.282	96.494	97.783	98.460	98.984	99.342	99.525	99.675
presa5	5	0.000	51.308	89.251	93.646	95.569	96.971	97.962	98.753	99.217	99.500	99.691
presa6	5	0.000	71.460	89.146	93.461	96.299	97.968	98.803	99.296	99.491	99.661	99.786
presa7	5	0.000	65.431	83.525	89.310	92.862	95.382	97.098	97.835	98.396	98.874	99.242
presa8	5	0.000	63.235	76.044	87.954	93.145	96.749	98.490	99.169	99.522	99.705	99.794
presa9	5	0.000	58.872	82.637	88.138	92.139	94.475	96.483	97.551	98.286	98.847	99.235
presa10	5	0.000	72.527	92.830	95.662	97.135	98.156	98.831	99.229	99.616	99.735	99.815
presa11	5	0.000	62.192	88.244	93.228	95.509	97.396	98.248	98.919	99.313	99.551	99.722
presa12	5	0.000	78.272	93.553	96.365	97.946	98.860	99.352	99.646	99.756	99.836	99.876
presa13	5	0.000	45.714	83.391	90.884	95.858	97.284	98.163	98.814	99.161	99.471	99.729
presa14	5	0.000	50.257	86.379	92.702	95.038	96.915	98.459	99.051	99.449	99.597	99.710
presa15	5	0.000	72.865	87.314	92.607	95.334	97.047	98.178	98.845	99.201	99.400	99.586
Tutte le prese sommate, tutte le dita	5	0.000	49.173	70.094	82.736	87.288	91.241	93.001	94.569	95.605	96.432	97.187
Tutte le prese sommate, solo pollice	1	0.000	90.926	95.773	97.886	99.219	100.0					
Tutte le prese sommate, solo indice	1	0.000	45.539	86.195	93.980	98.680	100.0					
Tutte le prese sommate, solo medio	1	0.000	67.674	88.004	95.726	98.749	100.0					
Tutte le prese sommate, solo anulare	1	0.000	66.411	87.420	95.221	98.319	100.0					
Tutte le prese sommate, solo migonolo	1	0.000	58.438	91.616	96.629	99.564	100.0					
Tutte le prese sommate, solo pollice, indice, medio	3	0.000	70.833	81.972	89.377	92.470	94.319	95.856	97.132	98.128	98.799	99.431
Tutte le prese sommate, solo indice, medio, anulare, mignolo	4	0.000	45.125	69.815	78.931	86.471	89.691	92.668	94.657	95.909	96.853	97.646
Prese da 1 fino 4, tutte le dita	5	0.000	56.765	77.415	88.123	91.422	93.633	95.313	96.433	97.275	97.996	98.412
Prese da 6 fino 10, tutte le dita	5	0.000	43.453	74.108	83.555	88.388	91.891	93.791	95.146	96.212	97.066	97.667
Presa 5 e da 11 fino 14, tutte le dita	5	0.000	74.013	84.728	90.755	94.980	96.262	97.225	98.032	98.579	98.927	99.169
Presa 5 e 13 (potenza, poll. Addotto), tutte le dita	5	0.000	32.751	61.650	86.935	91.588	95.243	96.637	97.460	98.048	98.594	98.943
Presa 12 e 14, tuttel le dita	5	0.000	64.008	82.751	90.482	93.865	96.062	97.279	98.025	98.711	99.183	99.426

Legend	Red if x>95
	Yellow if 90<x<95

Figura 2 Foglio Excel "Variance Coverage Raggruppamenti", foglio che contiene le riproduzioni dei risultati presentati nella tesi di Sara, questi raggruppamenti (quindi gradi di libertà presi in considerazione) sono quelli presentati da Sara nella tesi. I valori presenti sono la percentuale di varianza dello spazio di 27 g.d.l. riproducibile il numero di componenti principali indicato. Indica quanta varianza è coperta dallo spazio matematico a dimensioni ridotte. Dati utilizzati non aggiornati, di modello cinematico precedente.

	n° PC										
Possibili Raggrappamenti da fare	finge	0	1	2	3	4	5	6	7	8	9	10
Tutte le prese; pollice, indice, medio, anulare	4	0.000	61.481	78.157	85.054	89.219	92.984	94.662	95.895	96.779	97.606	98.307
Tutte le prese; pollice, medio, anulare, mignolo	4	0.000	52.903	74.557	86.895	90.384	93.447	95.284	96.280	97.137	97.712	98.262
Tutte le prese; pollice, indice, anulare, mignolo	4	0.000	52.088	71.741	83.996	88.383	92.133	94.024	95.375	96.366	97.162	97.837
Tutte le prese; pollice, indice, medio, mignolo	4	0.000	40.029	66.243	78.623	87.747	91.425	94.269	95.853	97.089	98.198	98.934
Tutte le prese; pollice, indice, anulare	3	0.000	66.755	79.056	86.027	90.446	94.087	95.801	97.062	97.942	98.619	99.090
Tutte le prese; pollice, indice, mignolo	3	0.000	58.585	75.990	85.881	90.590	94.192	95.595	96.732	97.623	98.356	98.938
Tutte le prese; pollice, medio, anulare	3	0.000	68.150	84.791	89.028	93.172	95.167	96.730	97.732	98.506	98.993	99.297
Tutte le prese; pollice, medio, mignolo	3	0.000	59.612	77.160	88.590	91.812	94.776	96.097	97.196	98.017	98.634	99.119
Tutte le prese; pollice, anulare, mignolo	3	0.000	56.607	76.818	89.057	92.564	94.683	96.384	97.333	98.133	98.736	99.181
Tutte le prese; indice, medio, mignolo	3	0.000	42.787	69.442	80.420	88.015	91.397	94.280	95.840	97.181	98.210	98.962
Tutte le prese; indice, anulare, mignolo	3	0.000	45.232	71.600	80.343	87.805	91.058	93.956	95.840	97.095	98.082	98.729
Tutte le prese; indice, medio, anulare	3	0.000	49.764	67.309	78.313	88.013	91.368	94.027	95.975	97.123	98.222	98.843
Tutte le prese; medio, anulare, mignolo	3	0.000	51.199	78.455	85.655	91.467	94.272	95.747	96.909	97.806	98.602	99.142
Tutte le prese; pollice, indice	2	0.000	78.403	86.130	92.986	95.142	96.952	98.245	99.142	99.672	99.865	100.000
Tutte le prese; pollice, medio	2	0.000	80.385	89.058	93.638	95.537	97.207	98.370	99.194	99.688	99.894	100.000
Tutte le prese; pollice, anulare	2	0.000	75.169	86.921	91.221	95.345	97.107	98.357	99.094	99.529	99.811	100.000
Tutte le prese; pollice, mignolo	2	0.000	64.755	82.011	92.001	95.171	96.847	97.971	98.864	99.513	99.889	100.000
Tutte le prese; indice, medio	2	0.000	47.084	74.326	83.325	91.079	95.291	97.832	98.682	99.213	99.675	100.000
Tutte le prese; indice, anulare	2	0.000	45.430	65.537	77.914	89.918	94.027	96.473	97.966	98.955	99.651	100.000
Tutte le prese; indice, mignolo	2	0.000	46.652	72.571	84.130	91.648	95.090	96.988	98.415	99.377	99.733	100.000
Tutte le prese; medio, anulare	2	0.000	63.839	79.134	87.987	92.970	96.048	97.751	98.780	99.537	99.834	100.000
Tutte le prese; medio, mignolo	2	0.000	49.605	80.320	88.674	93.018	96.236	98.085	98.813	99.431	99.810	100.000
Tutte le prese; anulare, mignolo	2	0.000	53.040	83.848	90.207	94.445	96.744	98.137	98.882	99.399	99.834	100.000

Figura 3 Foglio Excel "Variance Coverage Raggruppamenti", raggruppamenti (quindi gradi di libertà presi in considerazione) nuovi considerati da Marco. Dati utilizzati non aggiornati, ma di modello cinematico precedente.

PCA

L'Analisi delle Componenti Principali (PCA) è una potente tecnica ampiamente utilizzata nell'analisi dei dati e nella riduzione della dimensionalità. È particolarmente utile per ridurre la complessità dei dati ad alta dimensionalità mantenendo la maggior parte delle informazioni essenziali. L'idea fondamentale alla base della PCA è quella di trasformare i dati originali in un nuovo sistema di coordinate, in cui gli assi (componenti principali) sono ortogonali tra loro e allineati alle direzioni di massima varianza nei dati.

Sinteticamente, la PCA cerca di individuare le direzioni lungo le quali i dati variano maggiormente. Queste direzioni, note come componenti principali, sono ordinate in modo che la prima componente principale catturi la varianza più significativa nei dati, seguita dalla seconda, terza e così via. Mantenendo solo le prime componenti principali, che spiegano la maggior parte della varianza, la PCA riduce efficacemente la dimensionalità dei dati minimizzando la perdita di informazioni.

L'algoritmo PCA funziona prima calcolando la matrice di covarianza dei dati. Successivamente, calcola gli autovettori e gli autovalori di questa matrice di covarianza. Gli autovettori rappresentano le direzioni di massima varianza (le componenti principali), mentre gli autovalori indicano la quantità di varianza spiegata da ciascuna componente principale. Questi autovettori e autovalori consentono alla PCA di determinare gli assi più informativi su cui proiettare i dati.

Una volta identificate le componenti principali, la PCA proietta i dati originali su queste componenti per ottenere una rappresentazione a dimensioni ridotte. Questa rappresentazione trasformata può essere utilizzata per vari scopi, come la visualizzazione, la compressione dei dati, la riduzione del rumore e l'estrazione delle caratteristiche. Inoltre, la PCA viene spesso utilizzata come passaggio di preelaborazione prima di applicare altri algoritmi di apprendimento automatico, in quanto può migliorarne le prestazioni riducendo la dimensionalità e concentrandosi sulle caratteristiche più rilevanti.

In generale, la PCA è una tecnica versatile e ampiamente utilizzata nell'analisi dei dati e nell'apprendimento automatico, offrendo preziose informazioni sui dataset ad alta dimensionalità e facilitando il calcolo e l'interpretazione efficienti. La sua capacità di ridurre la dimensionalità preservando informazioni essenziali la rende uno strumento fondamentale nel toolkit di scienziati dei dati e ricercatori in vari settori.

Formulazione per la varianza massima

Esistono due formulazioni principali della PCA: formulazione di massimizzazione della varianza; e formulazione di minimizzazione dell'errore. È stata utilizzata la prima, i cui principali passi sono i seguenti:

1. **Centrare la distribuzione**

Il primissimo passo è centrare la distribuzione. Significa calcolare la media e sottrarla a ogni punto dato della distribuzione P.

P è la matrice di dati, in cui i vettori riga contengono i valori presi nell'istante ***t*** dei gradi di libertà del modello cinematico. Nel nostro caso i gdl sono valori degli angoli dei giunti e sono disposti su 27 colonne della matrice P.

Operazione di centratura:

$$[\mathrm{P_{MOD}} = \mathrm{P} - \mathrm{mean(P)}]$$

2. **Calcolare la Matrice di Covarianza**

Calcolare la matrice di covarianza basata sui dati standardizzati.

Se il dataset originale è indicato come *P*, la matrice di covarianza è data da:

$$\left[\mathrm{C} = \frac{1}{(\mathrm{n}-1)}\mathrm{P_{MOD}^{T}}\mathrm{P_{MOD}}\right]$$

3. **Decomposizione degli Autovalori**

Calcolare gli autovalori (*λ*) e gli autovettori (*v*) della matrice di covarianza.

$$C\lambda = \lambda V$$

Matlab risolve l'equazione sopra con la funzione *eig()*.

4. **Selezionare le Componenti Principali**

Scegliere i primi *k* autovettori corrispondenti agli autovalori più grandi. La matrice *V* costituisce una base dello spazio di *P* dove ogni componente della base è ordinato in base alla dimensione della varianza nella direzione descritta da questo vettore.

5. Proiettare i Dati sulle Componenti Principali

Ora, una volta che è stato creato una base ***V*** e ridotta a k autovettori, è possibile proiettare i dati sulle componenti principali. Questo permette di avere i dati in formato ridotto nel sistema di coordinate calcolato con PCA. La proiezione nel modello sono chiamati scores o coefficienti, poiché è possibile passare da una rappresentazione all'altra, utilizzandoli come coefficienti.

Infatti, per trovare ***Z,*** basta moltiplicare la matrice di dati centrati $\boldsymbol{P_{mod}}$ per gli autovettori selezionati per ottenere la matrice di dati trasformata *Z*:

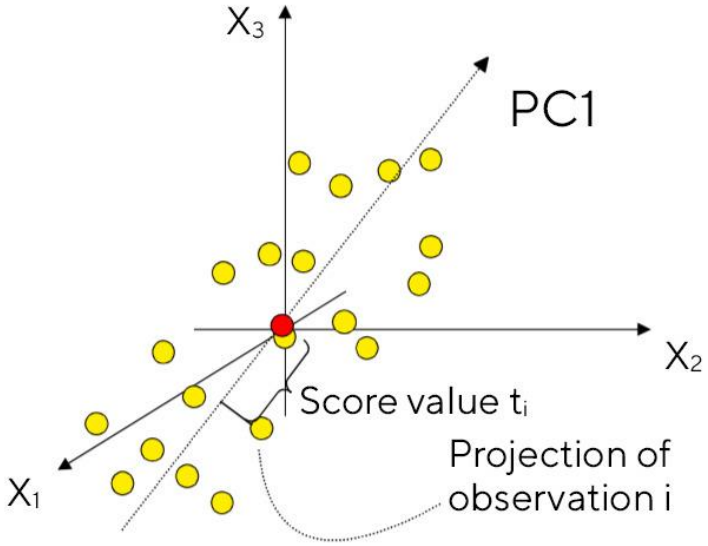
$$[\mathbf{Z} = \mathbf{P_{mod}}\mathbf{V}]$$

Qui, *V* contiene gli autovettori come colonne.

Dato poi che ***V*** è una base, quindi ortogonale, allora $\boldsymbol{V^T = V^{-1}}$:

$$[\mathbf{Z}\mathbf{V}^{\mathbf{T}} \cong \mathbf{P_{mod}}]$$

Figura 4 Illustrazione PCA con 1 componente principale in spazio a 3 dimensioni. È raffigurato anche lo score sulla PCA1 di un punto. Punto rosso indica media della distribuzione.



(Sartorius, 2020)

Rapporto di Varianza Spiegata

La proporzione di varianza totale spiegata da ciascuna componente principale è data da:

$$\left[\text{Rapporto di Varianza Spiegata} = \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}\right]$$

Quindi sul totale intero che rappresenta il totale della varianza (spiegata), noi possiamo calcolare il totale di varianza spiegata (o coperta) dei primi *k* autovettori, sommando i primi k autovalori e dividere per la somma di tutti gli autovalori. Questo è uno dei due indici utilizzato nella tesi di Sara. Vedere Figura 2 e Figura 3, per tabelle di risultati riprodotti della tesi.

L'altro valore utilizzato da Sara è il valore per ogni grado di libertà nello spazio angolare dell'autovettore più influente di ***V,*** quindi una indicazione quanto le dimensioni nello spazio influiscano la componente principale (quella che spiega la varianza di più)***.***

1 ALL5	0	60.2340	81.8641	91.1168	96.2323	97.2439	97.8596	98.4367	98.8232	99.1067	99.35
2 THUMB4NO_LITTLE	0	62.2366	84.5250	94.0931	97.5384	98.4036	98.9192	99.2988	99.4924	99.6572	99.735
3 THUMB4NO_INDEX_angles	0	61.0177	82.9160	92.3041	96.9050	97.8636	98.4450	98.8882	99.2438	99.5239	99.671
4 THUMB4NO_MIDDLE_angles	0	61.3270	83.3604	92.7778	96.9102	97.7692	98.3491	98.7998	99.1057	99.3531	99.536
5 THUMB4NO_RING_angles	0	61.3695	83.3907	92.8126	96.6862	97.6568	98.2403	98.7966	99.1080	99.3514	99.553
6 LAST4FING_angles	0	62.0790	72.9411	80.6232	86.2950	89.5633	92.4405	94.6195	96.4675	97.7118	98.352
7 FIRST3FING_angles	0	63.4599	86.1778	95.9341	98.1600	98.9489	99.3014	99.5926	99.7610	99.8473	99.895
8 THUMB3INDEX_RING_angles	0	63.4154	86.1411	95.8923	98.1599	98.8432	99.2666	99.5041	99.6786	99.7959	99.865
9 THUMB3INDEX_LITTLE_angles	0	62.5102	84.9535	94.5523	97.4366	98.2815	98.8663	99.3072	99.5637	99.7445	99.835
10 THUMB3MIDDLE_RING_angles	0	63.0843	85.6540	95.3695	98.1626	99.0231	99.4763	99.6467	99.7417	99.8211	99.865
11 THUMB3MIDDLE_LITTLE_angles	0	62.1887	84.4862	94.0509	97.3532	98.2527	98.8398	99.2106	99.5081	99.7361	99.834
12 THUMB3RING_LITTLE_angles	0	62.1459	84.4569	94.0170	97.6669	98.4253	99.0080	99.3145	99.5692	99.7374	99.851
13 INDEX3MIDDLE_RING_angles	0	65.2188	77.4213	87.3608	91.0484	94.0470	96.0223	97.0888	98.0954	98.9109	99.506
14 INDEX3MIDDLE_LITTLE_angles	0	59.3656	72.6403	81.2502	87.0202	90.7838	93.9174	96.2507	97.8863	99.0091	99.546
15 INDEX3RING_LITTLE_angles	0	62.6398	75.2124	82.5722	87.1054	90.8364	93.6789	95.9612	97.6993	98.9058	99.524
16 MIDDLE3RING_LITTLE_angles	0	66.3174	77.3208	84.6089	89.9797	93.8038	96.1700	97.4385	98.4072	99.1627	99.612
17 THUMB2INDEX_angles	0	64.6915	87.8667	97.8166	98.9999	99.5929	99.8106	99.8928	99.9371	99.9722	100.000
18 THUMB2MIDDLE_angles	0	64.3471	87.3562	97.2668	98.6815	99.4648	99.7907	99.8740	99.9224	99.9667	10
19 THUMB2RING_angles	0	64.3022	87.3202	97.2257	98.8307	99.5119	99.7537	99.8615	99.9210	99.9682	100.000
20 THUMB2LITTLE_angles	0	63.3668	86.0967	95.8448	98.2308	98.9695	99.5579	99.8248	99.9159	99.9661	100.000
21 INDEX2MIDDLE_angles	0	65.2609	79.8811	89.5794	94.6665	97.2227	98.9013	99.5993	100	100	10
22 INDEX2RING_angles	0	65.0380	79.4203	86.5741	92.1780	95.5177	97.9635	99.2091	100	100	10
23 INDEX2LITTLE_angles	0	60.0579	76.8659	86.8645	92.1301	95.7846	97.9296	99.3479	100	100	10
24 MIDDLE2RING_angles	0	71.4110	87.7148	92.0034	94.7528	97.0437	98.3518	99.5084	100	100	10
25 MIDDLE2LITTLE_angles	0	63.9202	77.8053	85.2248	91.5736	96.2206	98.3064	99.2037	100.0000	100.0000	100.00
26 RING2LITTLE_angles	0	69.7379	82.3121	88.0307	92.7750	96.3588	98.6296	99.7074	100	100	10
27 THUMB_angles	0	65.6202	89.0986	99.2097	99.9116	99.9646	100.0000	100.0000	100	100	10
28 INDEX_angles	0	76.9930	91.6393	97.0286	100	100	100	100	100	100	10
29 MIDDLE_angles	0	69.9289	93.4806	97.5674	100	100	100	100	100	100	10
30 RING_angles	0	78.3967	90.7026	96.1415	100	100	100	100	100	100	10

Figura 5 Variance Coverage permutazioni di dita, tabella Matlab dati aggiornati al modello più recente utilizzato in questo elaborato

1 GRIP1	0	89.1004	93.9767	95.9122	96.9714	97.8765	98.5357	98.9788	99.3244	99.4965	99.6224	99.7265
2 GRIP2	0	80.9469	90.9186	93.4845	95.3917	96.7522	97.7141	98.3814	98.8076	99.0820	99.3380	99.5063
3 GRIP3	0	76.3212	84.5383	90.1531	94.3344	96.7443	97.7654	98.5114	99.0415	99.3446	99.5059	99.6298
4 GRIP4	0	86.9432	92.8478	96.3111	97.7789	98.5058	98.9564	99.2931	99.4865	99.6284	99.7253	99.7881
5 GRIP5	0	83.9317	91.0816	94.5383	96.4080	97.4147	98.3088	98.7582	99.0026	99.2377	99.4156	99.5552
6 GRIP6	0	62.3978	82.5535	88.6498	93.6780	95.8086	97.3841	98.2276	98.7402	99.1096	99.4129	99.6342
7 GRIP7	0	54.5299	81.1628	88.3029	91.8468	94.2176	96.0783	97.1943	97.9537	98.5435	98.9774	99.3412
8 GRIP8	0	58.4225	80.1708	85.6949	90.2789	94.1200	96.1724	97.7377	98.4765	99.0097	99.3322	99.5854
9 GRIP9	0	61.7760	74.3447	84.6894	90.4058	93.4469	95.1189	96.6443	97.5049	98.1770	98.7002	99.1142
10 GRIP10	0	90.9305	94.7269	96.4752	97.7133	98.4736	98.9955	99.2418	99.4079	99.5633	99.6996	99.7893
11 GRIP11	0	59.1336	79.9986	90.7525	94.6850	96.5987	97.6113	98.4378	98.9411	99.2875	99.5184	99.6603
12 GRIP12	0	83.0103	94.9334	97.3508	98.4358	98.9913	99.3787	99.5658	99.6799	99.7607	99.8270	99.8719
13 GRIP13	0	78.0332	90.6600	93.9787	95.8074	97.0059	97.8434	98.5692	98.9108	99.2417	99.4612	99.6164
14 GRIP14	0	81.2951	89.7374	92.7715	95.5324	97.0641	98.2252	98.7202	99.0779	99.3669	99.5282	99.6725
15 GRIP15	0	61.8944	80.7977	86.2890	89.7921	92.4531	94.8135	96.1388	96.9879	97.5948	98.1597	98.5918
16												
17												
18												

Figura 6 Variance Coverage prese singole, tutte le dita, , tabella Matlab dati aggiornati al modello più recente utilizzato in questo elaborato

Implementazione MatLab

“PCA.m” è il file di codice Matlab che esegue la PCA. Inizialmente chiede all’utente di scegliere i file su cui svolgere le operazioni. Prenderà i dati ed eseguirà la PCA per tutte le prese (letto come file) in un ciclo for singolarmente. Dopodiché, esegue funzioni di raggruppamenti e raggruppa in struct i dati, che contengono valori e quali dita sono coinvolte. Eseguirà la PCA anche su questi raggruppamenti. I file che contengono i passaggi matematici chiave sono “func_PCA.m” e “calc_scores.m”.

Quindi, partendo dai componenti principali e dai dati centrati (valore – media) è possibile calcolare i scores ***Z*** che riporta nello sistema iniziale di coordinate

a valori approssimativi di quelli iniziali. Notare bene, che maggiore è il numero di componenti utilizzate, migliore è l'approssimazione.

Nella implementazione utilizzata nella tesi su cui si basa questo testo, gli autovettori più a destra nella matrice degli autovettori in Matlab, matrice calcolata con la funzione *eig(C)* dove *C* è la matrice di covarianza.

```
u = (mean((P)'))'; %matrice della media dei vettori colonna p
h = ones(n,1); %matrice di uni usata per la prossima operazione
P_mod = (P)-u*h'; %centro la distribuzione
C = (P_mod*P_mod')./(n-1); %matrice covarianza
[V,D] = eig(C); %eig calcola autovettori ed autovalori di C
[Z, V_red, results_mod] = calc_scores(V, P_mod, 27);
results = results_mod + u*h';
```

Figura 7 Implementazione PCA in func_PCA

```
%reduce the eigenmatrix of the most varying components (the last columns)
V_red = V(:, m-k+1:m);

%Calculate scores Z
Z = V_red.' * P_mod;%è la matrice dei coefficienti della combinazione
%lineare ottenuta con la PCA
%V contiene le componenti principali, Z i loro coefficienti

%Opposite operation to calculate results i.e. angle
results = V_red * Z;
```

Figura 8 Calcolo degli "scores", cioè i coefficienti, in calc_scores

Minimizzazione

Dopo il calcolo delle PC e dei coefficienti di esse, è necessario avere degli indici più concreti della approssimazione post PCA.

Errore angolare proiezione PCA con dati input

Una prima misura della performance della approssimazione è il confronto degli angoli riproiettati con gli angoli dati in input alla PCA (cioè, i risultati calcolati a seguito del problema della cinematica inversa dai dati presi, che rappresentano i dati inziali della PCA). Il risultato dipende dal numero di componenti principali utilizzate, dato che prendendo un numero minore di componenti *k* si ha *(m – k)* gradi di libertà in meno per descrivere un dato descritto da *m* variabili indipendenti; dunque, minore è il numero di componenti

utilizzate più grossolana sarà la trasformazione nel sistema di riferimento originale a *m* gradi di libertà (coordinate angolari). Questo confronto viene eseguito in un file main addizionale chiamato: “anglesALL5_squaremeandifference.m” nella cartella contenenti la versione finale Marco.

In questo caso utilizzando tutte le componenti, si torna ovviamente ad avere differenza nulla tra angoli di input e quelli di output (lo spazio è completamente descrivibile dalla PCA, sarà ruotato il sistema di riferimento).

Con tutte le PC incluse l’errore è praticamente nullo, come aspettato.

Errore cinematica diretta

In maniera del tutto simile, si può risolvere/calcolare la cinematica diretta e fare il confronto delle approssimazioni delle posizioni nello spazio di certi punti appartenenti alle falangi e al carpo. In questo lavoro, dato che i dati sperimentali provengono da marker rilevati tramite video, in posizioni fisse (o perlomeno con spostamenti minimi dovuti alla deformabilità dei corpi coinvolti) un ottimo indice se la approssimazione è vantaggiosa o efficace l’errore quadratico medio tra le posizioni sperimentali dei marker nello spazio cartesiano e le posizioni approssimate delle proiezioni nel sistema di riferimento di Componenti Principali.

Ora, descrivendo in maniera generale, è necessario minimizzare l’errore che si ha tra la proiezione e i dati sperimentali. Definiamo quindi la funzione di costo, funzione da minimizzare in problema di ottimizzazione, si avrà un problema strutturato nella seguente maniera:

$$D = \left|\left|\theta - \hat{\theta}\right|\right|$$

$$Costo = \sum D$$

dove θ indica la posizione obbiettivo/effettiva e $\widehat{\Theta}$ la posizione predetta/proiettata. D è la distanza o errore. Nel caso trattato, il valore di

distanza da minimizzare è la distanza quadratica media di tutti i marker in un frame (posa). Quindi, prendendo come distanza *D*, dove $\hat{\Theta}$ e Θ sono vettori, la distanza quadratica (non calcoliamo la distanza perché evitiamo la radice per velocità computazionale, senza modificare il risultato):

$$D = \left(\Theta - \hat{\Theta}\right)^2 = \left(\begin{bmatrix}\Theta_1\\ \Theta_2\\ \Theta_3\end{bmatrix} - \begin{bmatrix}\hat{\Theta}_1\\ \hat{\Theta}_2\\ \hat{\Theta}_3\end{bmatrix}\right)^2 = \sum_i^3 \left(\Theta_i - \hat{\Theta}_i\right)^2$$

Per avere la distanza quadratica media (mean squared) di tutti i punti considerati:

$$D_{ms} = \frac{\sum_i^m D_i}{m}$$

Nel programma è poi impropriamente nominata la variabile *"avg_squared_dist",* quando in realtà si tratta non della distanza quadratica media (root mean squared error, RMSE), ma della radice della distanza quadratica media di cui il risultato offre una idea migliore della distanza medio tra punto proiettato e punto dato (pur non essendo effettivamente la distanza.

$$Costo = D_{rms} = \sqrt{D_{ms}} = \sqrt{\frac{\sum_i^m D_i}{m}}$$

Dove m sono punti (che possono essere posizione marker, posizione dito o altro. La funzione di costo utilizzata è quindi la somma delle distanze euclidee di una posa (un time-frame). Il nostro obbiettivo è minimizzare questa funzione di costo, che equivale a minimizzare la distanza complessiva tra posizioni.

Implementazione: minimizzazione metodo *FKupdate*

L'idea di base è di utilizzare i metodi già esistenti dell'oggetto Solver, quindi di scrivere solo la funzione di costo e non le funzioni della cinematica diretta. Dunque, utilizzeremo nella minimizzazione il metodo "*FKupdate"* della classe Solver.

Quindi come scritto prima la nostra funzione di costo è la radice dell'errore quadratico medio (RMSE) delle distanze. I punti considerati, per cui calcoliamo la posizione predetta, saranno tutti i marker; per ogni posa viene calcolato RMSE tra markers effettivi e markers predetti. Noi minimizziamo il valore RMSE per ogni istante di tempo.

Per eseguire la differenza e minimizzare per ogni istante di tempo, ci sono due oggetti Solver nel codice: uno contenente una sola posa utilizzato per calcolare la posizione predetta per quella posa/istante di tempo; un altro Solver "reale" contenente tutte le posizioni marker "reali" per una presa. In questa maniera per ogni instante '*t'* è possibile minimizzare l'errore medio dei marker.

Ora, è necessario capire però in funzione di quale variabile/i si sta minimizzando. Come già spiegato in precedenza, la PCA genera un sistema di riferimento semplificato (a meno gradi di libertà) e ruotato in base alla disposizione/varianza dei dati iniziali. Di conseguenza, abbiamo una base nuova da non modificare con vettori ortogonali fra di loro pari al numero di componenti principali. Dato il sistema complesso con cui si sta lavorando non necessariamente, le proiezioni della PCA meglio approssimano le posizioni e le pose nello spazio tre-dimensionale. Quindi, si prova a minimizzare in funzione dei coefficienti '*Z'* della PCA, mantenendo invariata la base 'V' trovata.

I valori posizione marker da entrambi i solver poi vengono utilizzati nella funzione di costo, che calcola la RMSE, per l'istante di tempo *t*. La funzione di costo scritta in file "*MINIMIZATION_func.m"*, viene utilizzata come funzione simbolica all'interno di "*fminunc()"*.

Risultati

Di seguito sono presentati alcuni risultati rappresentativi. Per questioni di tempo e costo computazionale non si poteva eseguire la minimizzazione di

tutti i risultati. I risultati trovati sono salvati come workspace nella cartella di minimizzazione.

Tutti i risultati presentati sono di prese singole, minimizzate con il sistema di riferimento trovato a seguito di PCA che considera tutte le prese e tutte le dita (*"ALL5"*), prendendo 3 componenti principali. Prendendone 5 l'errore si riduce anche molto, ma il tempo necessario per computare una presa ammontava a varie decine di ore.

Figura 9 Grip 1, 3 compoenti principali, RMSE tra marker sperimentali e marker predetti PCA

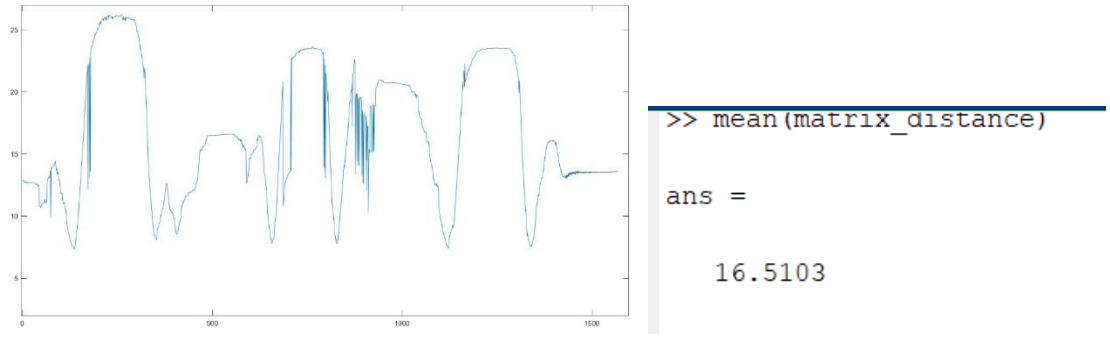


Figura 10 Grip 2, 3 compoenti principali, RMSE tra marker sperimentali e marker predetti PCA

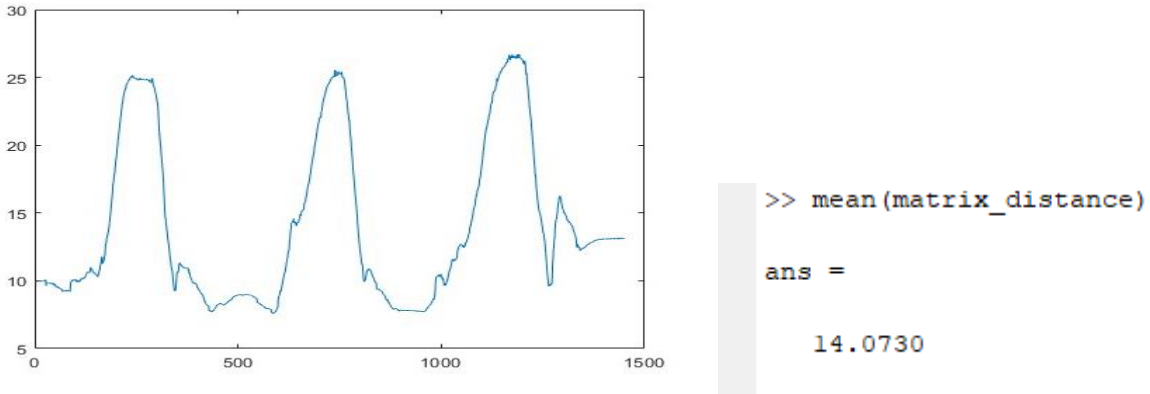


Figura 11 Grip 3, 3 compoenti principali, RMSE tra marker sperimentali e marker predetti PCA

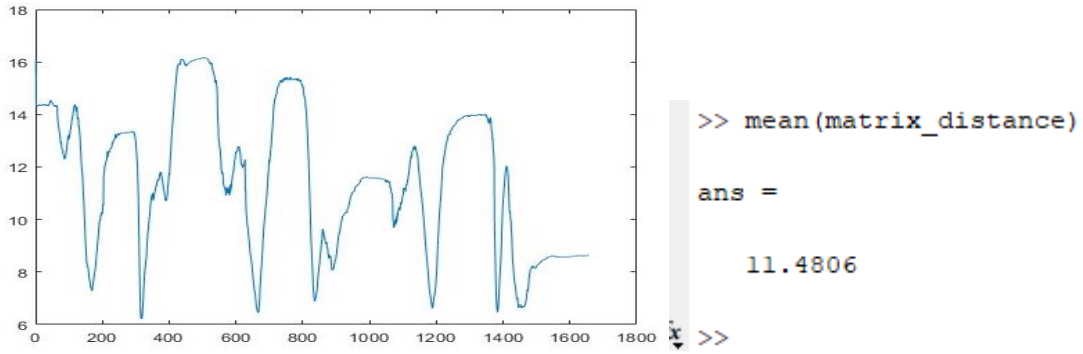
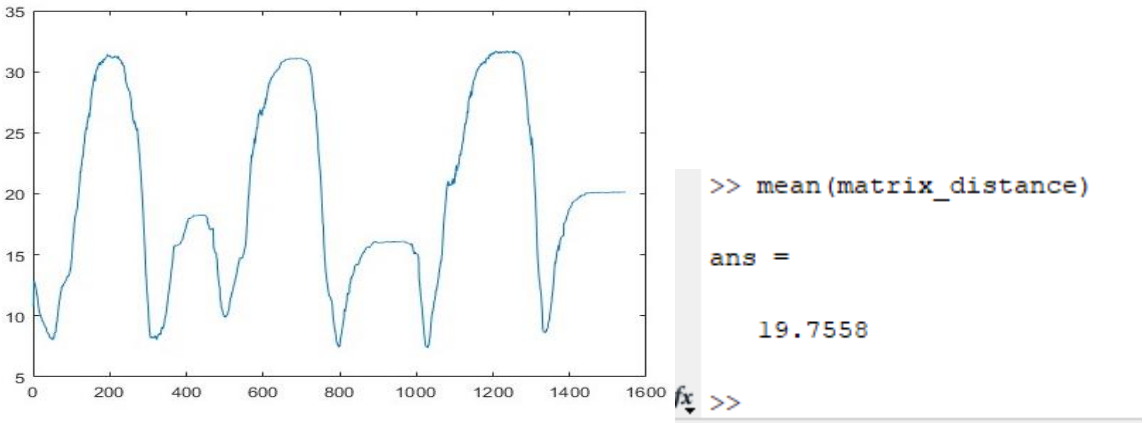


Figura 12 Grip 4, 3 compoenti principali, RMSE tra marker sperimentali e marker predetti PCA



Dall'errore medio raffigurato si può verificare (unità di misura in millimetri) che l'radice dell'errore quadratico medio per le varie prese varia tra 10mm e 20mm circa, con dei picchi massimi fino a 32mm circa.

Ottimizzazione

Di certo è possibile eseguire delle migliorie sull'applicazione. Infatti, un primo tentativo di implementazione non andato a buon fine, era migliore in termini di velocità di calcolo, però i risultati erano completamente errati.

In questa prima implementazione, prendendo spunto e copiando dal codice del metodo *FKUpdate* si ha tentato di comporre la funzione della cinematica diretta, prima di eseguire la minimizzazione, tramite composizione di stringhe. Invece, Nella implementazione funzionante la cinematica diretta viene

eseguita da *FKUpdate* in cui la funzione della cinematica diretta viene in parte composta ed inizializzata, e dato che, minimizzare non è altro che un ciclo con tante piccole variazioni, in cui ogni volta che viene chiamato *FKUpdate* modificando **Z**, è necessario ricomporre la formula della cinematica diretta ogni qualvolta si riparte nel ciclo. Poi, non solo esiste il ciclo di minimizzazione (implicito della funzione di minimizzazione → *fminunc),* si deve considerare che si minimizza per ogni ***t*** in un file di presa, quindi un ciclo for per ogni **t**.

Dunque, si ha un ciclo di minimizzazione all'interno di un ciclo di iterazione, in cui non solo si calcola l'errore per minimizzare, ma si compone la funzione a ogni iterazione.

Infatti, l'implementazione errata, cercava di comporre la funzione simbolica antecedentemente alla minimizzazione calcolando poi direttamente dalla funzione simbolica i dati, ma evidentemente la composizione e/o l'estrazione dei dati era completamente errata.

La seconda implementazione, quella utilizzata e spiegata nelle precedenti pagine, è più facile da un punto di vista di implementazione, ma molto più costosa in termini di risorse poiché rigenera in ogni ciclo di minimizzazione la funzione simbolica per la cinematica.

Conclusioni

Purtroppo, con tre componenti principali, l'errore è ancora incisivo e il sistema risulta non soddisfacente. Infatti, si vorrebbe avere circa questo numero di componenti principali per avere un sistema meno complesso e meno costoso, facile da controllare con errori che si aggirano al di sotto dei 5mm e non i 10-20mm che si hanno in questi risultati.

Inoltre, con la implementazione utilizzata il calcolo risulta poco tempestivo e sicuramente l'esperienza detta che è possibile avere delle migliorie anche in questo ambito.

Di base, entrare nel codice e nell'argomento forse è stato costoso in termini di ore e la difficoltà finale della minimizzazione era percettibile, ma l'operato durante il tirocinio ha fruttato risultati.

Per il futuro, si potrebbe provare a standardizzare i dati non solo centrarli prima della PCA. Questo significa ridurli alla stessa varianza, dividendo il dato per la deviazione standard di quel grado di libertà.

Un'altra cosa a cui si è pensato, ma non si è riuscito a implementare in questo progetto è visualizzazione dell'effetto della variazione di una sola componente principale sul modello cinematico nello spazio (vedere la mano muoversi modificando una componente principale). Questo sarebbe implementabile scegliendo la componente principale *k* e durante il calcolo degli scores moltiplicare la colonna *k* di *V* con la *k*-esima riga di *Z,* avendo valore discreti linearmente discreti in Z.

$$angles = V \cdot Z$$

$$angles_k = V(:,k) \cdot Z(k,:)$$

Sicuramente un altro argomento da ampliare è l'ottimizzazione della minimizzazione.

Infine, c'è da analizzare tutto il resto del grande quantitativo di dati e trovare una possibile combinazione di prese e/o dita che permette una approssimazione con PCA adatta. Inoltre, da valutare quali dati da analizzare dato che sono state analizzate solo le theta (valori angolari presi dalle coppie) e non i valori di riferimento utilizzato nel modello, come alfa, beta e gamma.

Complessivamente, l'esperienza è stata un successo visto che accademicamente il sottoscritto ha appreso molto ed esplorato nuove tematiche, approfondendo argomenti già conosciuti. Poi, per mancanza di tempo, sicuramente non è completa l'esperienza, ma rimane una esperienza vantaggiosa.

Appendice: Diagrammi

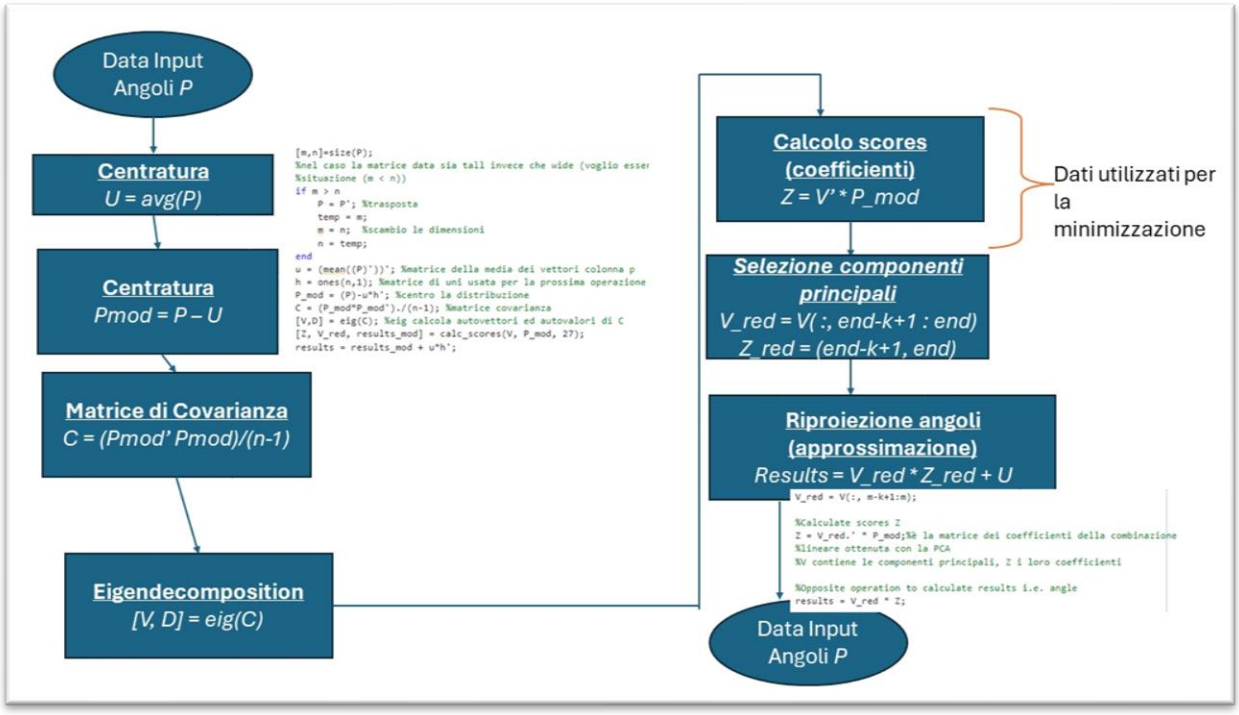


Figura 13 Diagramma PCA

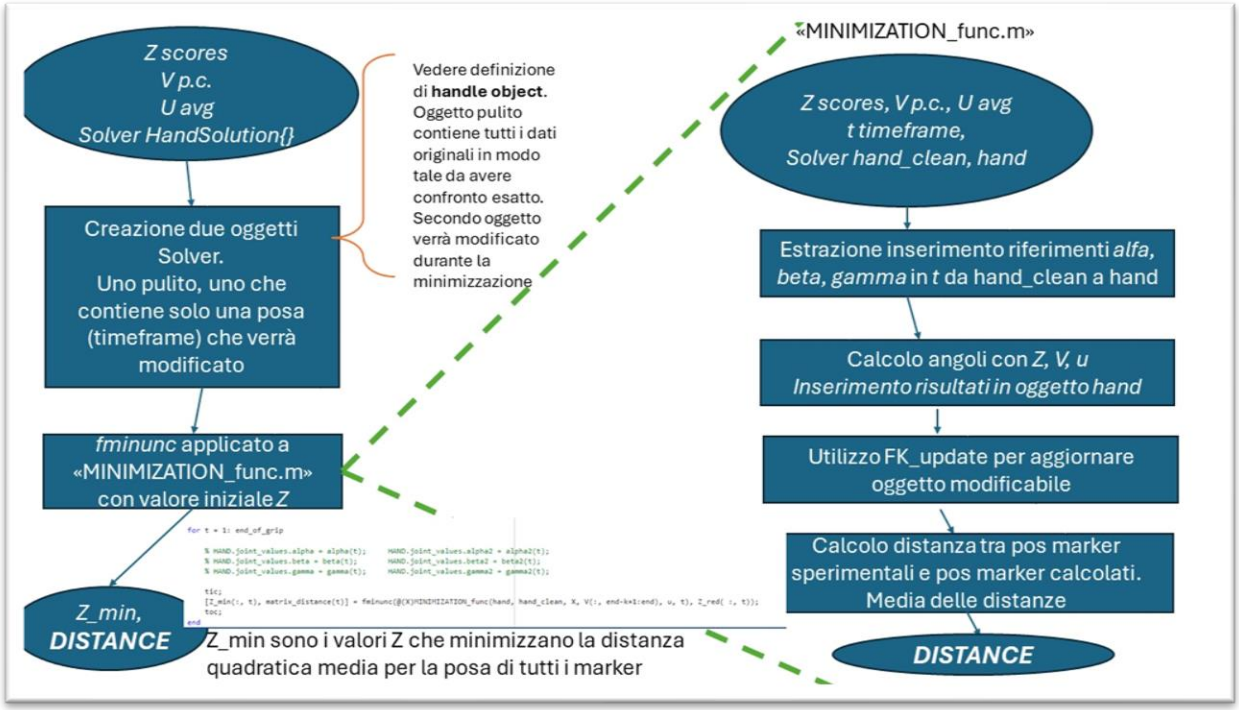


Figura 14 Diagramma Minimizzazione

Bibliography

ACM UbiComp/ISWC 2023. (2023, September 27). *medium.com*. Tratto da medium.com: https://medium.com/ubicomp-iswc-2023/accurate-wearable-3d-pose-tracking-without-external-camera-fa05348551a

Fischer, G., Jermann, D., Renate, L., Reissner, L., & Calcagni, M. (2020). Development and Application of a Motion Analysis. *Applied Sciences*.

Sartorius. (2020, Agosto 18). *www.sartorius.com/*. Tratto da https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186?srsltid=AfmBOooPiRbmVGMgMTmcUUnejMOGCoCsaNoK7O9lNBdLjtWI_DfKsa7-