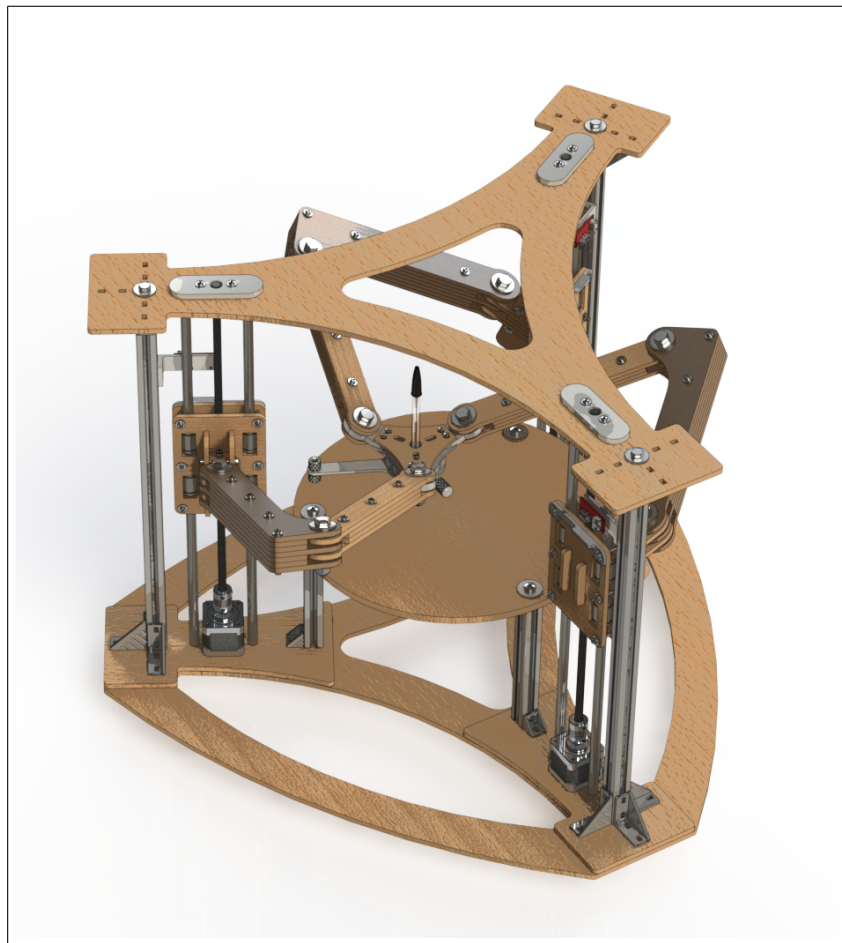


UNIVERSITÀ DEGLI STUDI DI PISA

SCUOLA DI INGEGNERIA

Laurea Magistrale in Ingegneria Robotica e dell'Automazione
Meccanica dei Robot - Prof. M. Gabiccini



Delteron G8.0

Roberto Mauceri
rob_mau@hotmail.it

Anno Accademico 2018/2019

Indice

1	Introduzione	2
2	Geometria	2
3	Hardware	7
	3.1 Materiali e Tecniche di Produzione	7
	3.2 Soluzioni Costruttive	8
	3.3 Componentistica Elettronica e Azionamenti	10
4	Software	12
5	Suggerimenti Pratici	14

1 Introduzione

Il presente elaborato raccoglie le procedure seguite e i risultati ottenuti durante gli stadi di progettazione e costruzione di un prototipo di robot Delteron in grado di disegnare. Il robot riceve in ingresso una sequenza di istruzioni in G-code che codificano l'immagine da tracciare (prodotta in precedenza da un software di grafica vettoriale), le quali vengono opportunamente elaborate e trasformate in comandi da inviare agli attuatori. Ad una preliminare fase di studio cinematico della struttura seguono le fasi di modellazione CAD e realizzazione dei componenti, sfruttando i materiali e le tecnologie di produzione disponibili in laboratorio. Si passa dunque alla scelta della componentistica elettronica e degli azionamenti e, infine, alla stesura del codice che gestisce la macchina.

2 Geometria

Il Delteron (talvolta chiamato Tripteron collineare) è un manipolatore a cinematica parallela: è costituito cioè da più catene cinematiche seriali poste "in parallelo", che collegano il telaio all'organo mobile terminale (end-effector). In una simile struttura le relazioni che esprimono i vincoli di chiusura delle catene generano delle dipendenze tra alcune delle variabili di giunto e quindi, tipicamente, non tutti i giunti sono attuati. Uno dei vantaggi è quello di poter posizionare gli attuatori sul basamento, rendendo la parte mobile del robot più leggera e dotata di una più spiccata destrezza nei movimenti.

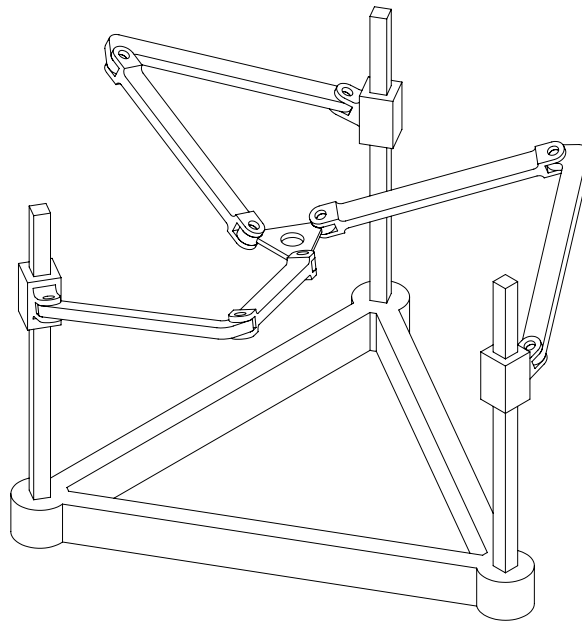


Figura 1: geometria del robot.

Come il nome stesso suggerisce, il Delteron è una sorta di fusione tra la geometria di un robot Delta e quella di un Tripteron. Ciascuna delle tre identiche catene cinematiche che concorrono alla movimentazione dell'end-effector è costituita da tre elementi, vincolati

tra loro mediante coppie rotoidali. Alle estremità della catena, da un lato, un giunto prismatico (attuato) collega questa al telaio, mentre, dall'altro, un ulteriore giunto rotoidale la collega all'end-effector. Gli assi delle cerniere di una stessa catena sono tra loro paralleli, ma inclinati rispetto ad una retta verticale di un angolo δ (Figura 2).

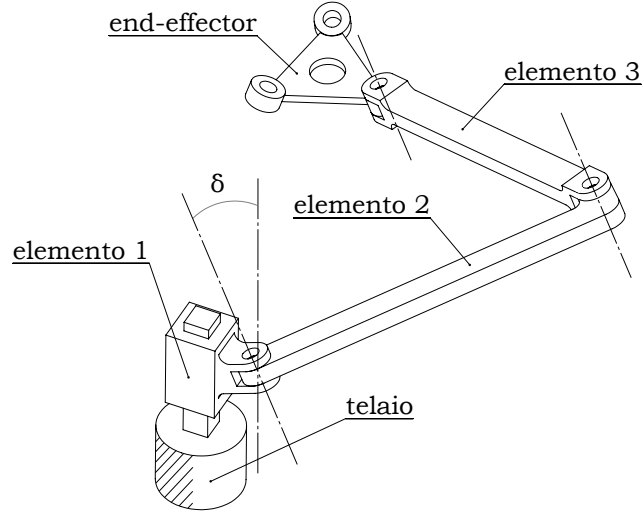


Figura 2: geometria della catena cinematica PRRR.

Così vincolato l'end-effector possiede solamente i tre gradi di libertà traslazionali e il suo orientamento nello spazio rimane immutato al variare della configurazione assunta dal robot. Poiché nella configurazione coi giunti prismatici alla stessa altezza, per simmetria della struttura, l'end-effector si trova posizionato con un orientamento orizzontale, per quanto detto sopra, sarà orientato orizzontalmente in ogni altra possibile configurazione.

Per affrontare lo studio della cinematica inversa è conveniente, in questo caso, adottare un approccio geometrico in quanto, seppur non rigoroso, snellisce la trattazione e fornisce una più immediata comprensione del sistema. Facendo riferimento alla Figura 3, il problema della cinematica inversa si traduce nel calcolo della coordinata verticale ζ del punto H , variabile associata al giunto prismatico, avendo noto il vettore posizione del punto E , centro dell'end-effector. La soluzione può essere ottenuta facilmente intersecando il piano perpendicolare all'asse delle cerniere e passante per il vertice V (la cui posizione relativa ad E è perfettamente nota considerando l'invarianza dell'orientamento dell'end-effector) con l'asse di scorrimento della coppia prismatica. In formule:

$$E = [x_e, y_e, z_e]$$

la posizione del vertice V è data da:

$$V = E + (V - E)$$

$$[x_v, y_v, z_v] = [x_e, y_e, z_e] + [0, d, 0] = [x_e, y_e + d, z_e]$$

il versore direttore dagli assi delle cerniere è:

$$\vec{n} = [0, -\sin \delta, \cos \delta]$$

mettendo insieme le equazioni appena ricavate, cioè intersecando i luoghi geometrici sopra definiti, ottengo infine la soluzione cercata, ovvero le coordinate del punto H e, in particolare, la sua quota ζ :

$$\begin{cases} z_h = z_e + \tan \delta (y_h - y_e - d) \\ x_h = l \\ y_h = 0 \end{cases}$$

\Downarrow

$$\zeta \stackrel{\text{def}}{=} z_h = z_e - \tan \delta (y_e + d)$$

Analogamente è possibile ottenere la coordinata verticale dei punti che individuano la posizione della coppia prismatica sugli altri due assi del robot (basta considerare la precedente geometria ruotata di $2/3\pi$ rad e $4/3\pi$ rad).

Adottando per tali coordinate i simboli ζ_1 , ζ_2 , ζ_3 (procedendo in senso antiorario), le equazioni che risolvono la cinematica inversa del robot sono:

$$\begin{aligned} \zeta_1 &= z_e - \tan \delta (d + y_e) \\ \zeta_2 &= z_e - \tan \delta (d - \sqrt{3}/2 x_e - 1/2 y_e) \\ \zeta_3 &= z_e - \tan \delta (d + \sqrt{3}/2 x_e - 1/2 y_e) \end{aligned}$$

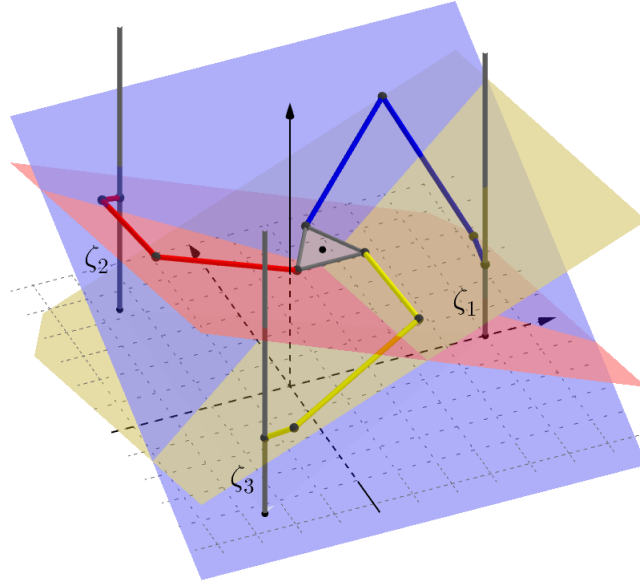


Figura 4: interpretazione geometrica delle equazioni della cinematica inversa.

È interessante notare come la soluzione sia del tutto indipendente da gran parte dei parametri geometrici che caratterizzano la struttura. Essa infatti necessita della conoscenza dei soli parametri d (distanza tra V e E) e δ (angolo di inclinazione dell'asse dei giunti rotoidali rispetto alla verticale). Il valore di l (distanza dell'asse del giunto prismatico dall'origine O), per quanto a prima vista possa sembrare necessario, è in

realità irrilevante. Infatti, sempre in riferimento alla Figura 3, il versore normale al piano inclinato ha componente nulla lungo x e quindi il piano intercetterà l'asse della guida sempre alla stessa quota se quest'ultimo viene fatto traslare in direzione parallela ad x stesso (variazione del valore di l).

Le relazioni ottenute, inoltre, godono della proprietà di linearità e possono allora essere riscritte nella più compatta forma matriciale:

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix} = \begin{bmatrix} 0 & -\tan \delta & 1 \\ \sqrt{3}/2 \tan \delta & 1/2 \tan \delta & 1 \\ -\sqrt{3}/2 \tan \delta & 1/2 \tan \delta & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} - d \tan \delta \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

dalla quale sono facilmente ottenibili le relazioni inverse (cinematica diretta):

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \frac{1}{3 \tan \delta} \begin{bmatrix} 0 & \sqrt{3} & -\sqrt{3} \\ -2 & 1 & 1 \\ \tan \delta & \tan \delta & \tan \delta \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix} + d \tan \delta \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Queste ultime sono utili per visualizzare lo spazio di lavoro del manipolatore e, di conseguenza, per operare una scelta sul valore da assegnare al parametro δ .

Come visibile in Figura 5, aumentando il valore dell'angolo di inclinazione dei bracci, a parità di ogni altro parametro progettuale, lo spazio di lavoro tende a ridursi. Per questo manipolatore si è scelto di porre $\delta = \pi/8$. Il valore del parametro d invece è stato valutato in modo tale da poter garantire sull'end-effector spazio sufficiente ad accogliere i cuscinetti e il sistema di serraggio della penna. Le lunghezze dei link che costituiscono i bracci, pur non influenzando direttamente le equazioni della cinematica, devono essere scelte in modo da permettere al robot di raggiungere tutte le posizioni previste evitando configurazioni singolari o strutturalmente impossibili.

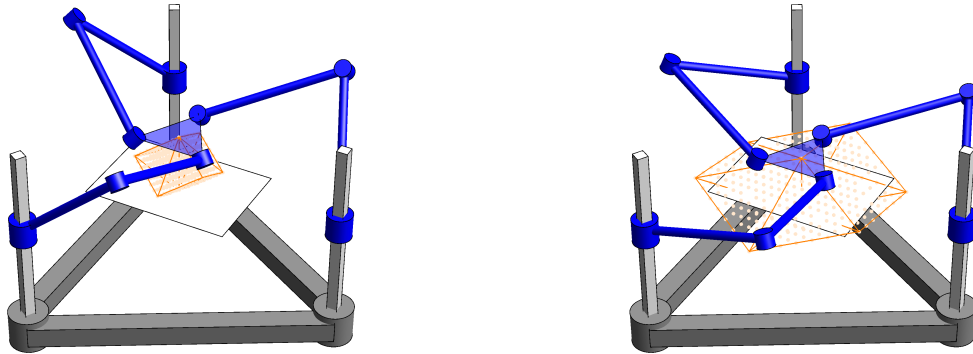


Figura 5: spazio di lavoro: a sinistra $\delta = \pi/4$, a destra $\delta = \pi/8$.

3 Hardware

3.1 Materiali e Tecniche di Produzione

La struttura meccanica è stata realizzata principalmente in legno multistrato, potendo far affidamento in laboratorio su una macchina per il taglio laser a controllo numerico, mentre componenti piccole o di geometria elaborata sono state realizzate in acido polilattico (PLA) con una stampante 3D, usando la tecnica di modellazione a deposizione fusa (FDM): un filamento plastico fuso è depositato da un ugello estrusore su strati successivi, che, disponendosi l'uno sopra l'altro, creano il volume del pezzo.

Per migliorare la robustezza della struttura, sono stati impiegati inoltre profilati e componenti modulari in alluminio Bosch Rexroth.

Tutte le parti del robot sono state disegnate con l'ausilio di Solidworks. I file .sldprt (solidworks part) generati dal software CAD, per poter essere letti ed elaborati, vanno convertiti in .dxf (drawing exchange format) per la macchina a taglio laser e .stl (standard triangulation language) per la stampante 3D.

Per gestire la stampante 3D si è usato il software Repetier-Host, il quale legge il file .stl, che è sostanzialmente una rappresentazione discretizzata mediante triangoli della superficie del modello, e, attraverso un programma di slicing (Slic3r o CuraEngine), suddivide l'oggetto in strati (layer). Per ogni strato genera poi le traiettorie che l'estrusore dovrà seguire (Figura 6). In questa fase è possibile settare diversi parametri, come lo spessore dei layer o la densità di riempimento del pezzo. In output il programma produce un file in G-code che invia direttamente alla macchina. Si veda il video nella cartella "Media" per un esempio di stampa.

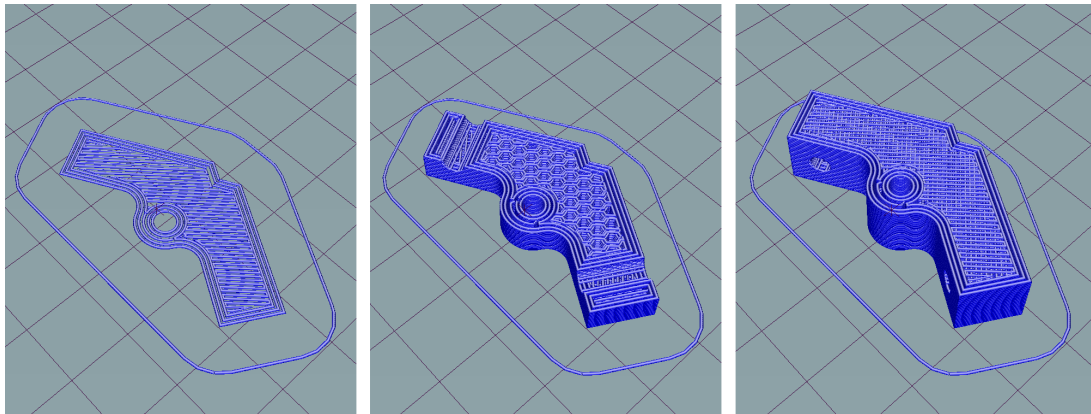


Figura 6: differenti layer del pezzo "griffa".

In Figura 7 sono riportati due esempi di pezzi realizzati con le tecniche appena esposte. A sinistra sono raffigurati i modelli disegnati in Solidworks, a destra ciò che è stato prodotto con le lavorazioni.

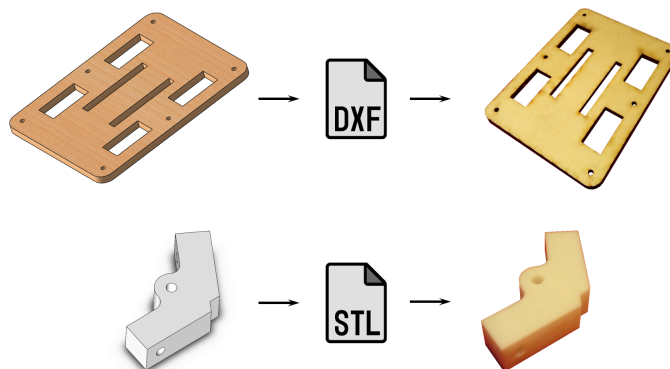


Figura 7: schema di realizzazione dei pezzi.

3.2 Soluzioni Costruttive

Il robot è ottenuto assemblando tra loro pezzi che, a causa del particolare processo di realizzazione (taglio laser), possono definirsi a ragion veduta "bidimensionali". Questi, impilandosi e incastrandosi tra loro come in un puzzle tridimensionale, generano le forme "solide". Per favorire il disassemblaggio o la sostituzione dei pezzi danneggiati, tutti gli elementi sono tenuti insieme attraverso collegamenti smontabili.

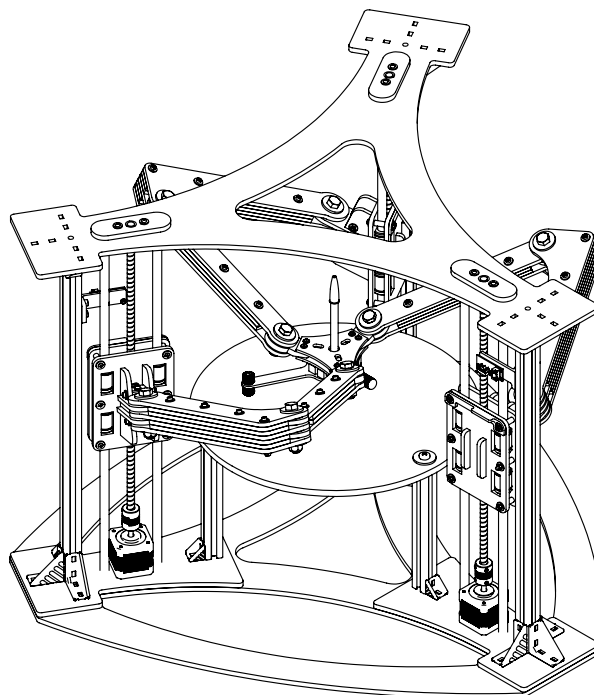


Figura 8: modello Solidworks del robot.

Nelle sottosezioni a seguire vengono mostrate le soluzioni costruttive adottate per la realizzazione di alcuni degli elementi più delicati della struttura: i giunti e il sistema di serraggio della penna.

Giunto Rotoidale

Il giunto rotoidale è stato realizzato mediante l'utilizzo di cuscinetti radiali a sfere $8 \times 22 \times 7$ mm il cui anello esterno è stato inserito con interferenza nell'apposito alloggiamento ricavato su uno dei due link da connettere. L'anello interno invece entra in contatto col secondo link attraverso la frapposizione di rosette. Il tutto è opportunamente serrato con una vite M8. Per rendere lo snodo più rigido si sono utilizzati due cuscinetti per asse, realizzando una configurazione nella quale fogli di legno appartenenti ad entrambi i link si alternano lungo lo spessore del giunto (Figura 9).

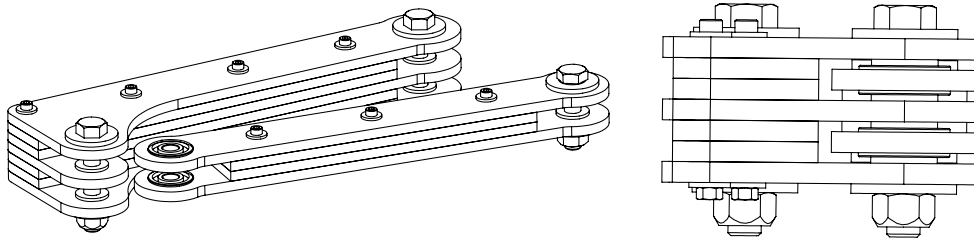


Figura 9: dettaglio del giunto rotoidale.

Giunto Prismatico

Il giunto prismatico è stato costruito impiegando cuscinetti a sfere lineari $10 \times 19 \times 29$ mm che scorrono lungo barre in acciaio M10. I cuscinetti sono stati alloggiati in fori rettangolari ritagliati su due piastrine, le quali impacchettano questi tenendoli fermi in posizione. Poiché la trasmissione del moto è realizzata attraverso un accoppiamento vite-madrevite, il blocco accoglie al suo interno un dado che andrà ad avvitarsi su una barra filettata M8, il cui moto è reso solidale all'albero del motore per mezzo di un coupler flessibile. A completamento del pezzo, tramite un sistema di incastri, è stato aggiunto il supporto per i cuscinetti radiali (che costituiscono parte del primo snodo rotoidale), il quale impone l'inclinazione dell'angolo δ al piano su cui idealmente giacerà l'intero braccio. Alcuni dettagli sono visibili in Figura 10.

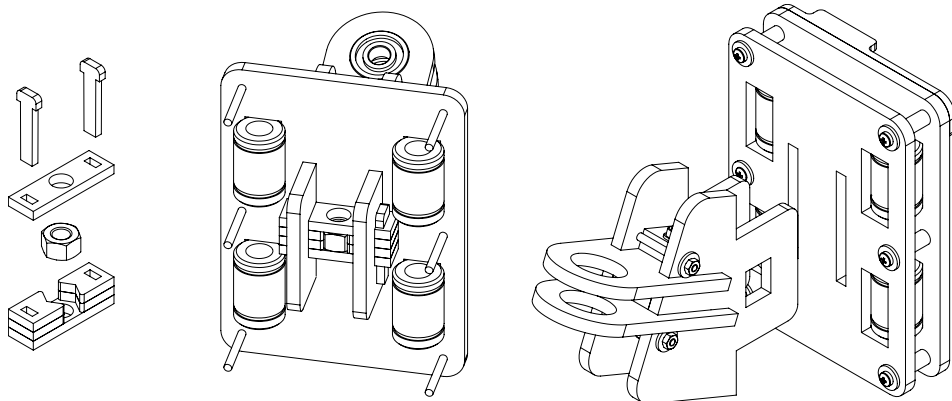


Figura 10: dettaglio del giunto prismatico.

Gripper

Il gripper è l'elemento preposto al posizionamento della penna ed è l'unico interamente realizzato in PLA per consentire alle parti in moto relativo un più fluido scorrimento. Esso svolge sia la funzione di centraggio che quella di serraggio.

La prima è adempiuta da un sistema di griffe scorrevoli movimentate da una ruota su cui son state ricavate delle asole curve. Ruotando in senso orario il braccio della ruota (se osservato dall'alto), le asole allontanano i perni su cui sono vincolate le griffe e il meccanismo si apre. Un anello elastico (in questo caso una sezione di camera d'aria di bicicletta), montato su delle pulegge avvitare sui perni, provvede invece alla chiusura automatica del dispositivo. Da notare che la sagoma dell'asola, in corrispondenza del punto di massima apertura, consente il movimento dei perni solo in direzione tangenziale alla ruota e, poiché le forze elastiche spingono invece in direzione radiale, il meccanismo rimane bloccato nella configurazione aperta, permettendo un comodo posizionamento della penna. Dopodiché basta ruotare leggermente il braccio per far scattare il movimento di chiusura (si veda il video nella cartella "Media").

La seconda funzione è invece svolta da una vite che attraversa trasversalmente il volume di una delle griffe ed entra in contatto con la superficie laterale della penna. La vite, stringendo leggermente, impedisce che la penna possa accidentalmente sfilarsi a causa delle forze di contatto col piano del foglio che tenderebbero a sollevarla. Una rappresentazione del dispositivo smontato è riportata in Figura 11.

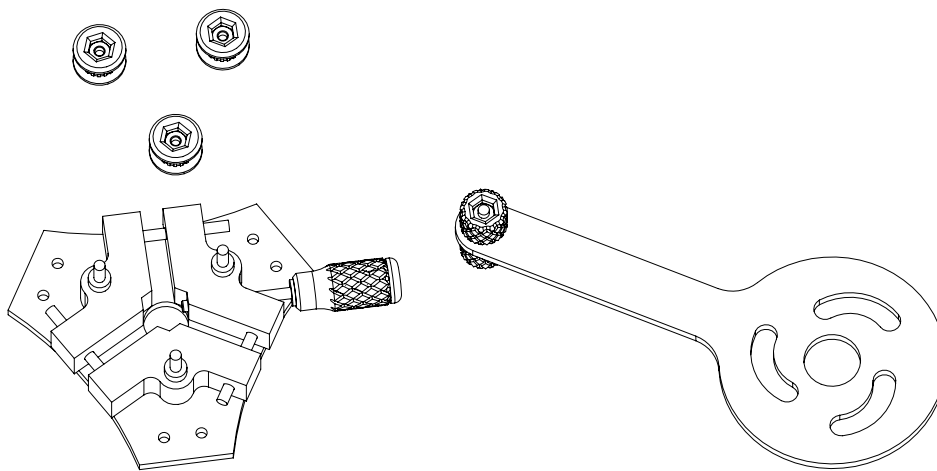


Figura 11: dettaglio del gripper.

3.3 Componentistica Elettronica e Azionamenti

Per quanto riguarda l'elettronica si è scelto di utilizzare una scheda Arduino Uno, affiancata da una CNC shield volta ad accogliere tre driver Pololu A4988 per motori stepper. A completare la dotazione ci sono tre motori stepper Nema 17 modello 17HS4401, tre endstop, i rispettivi cavi di collegamento e un alimentatore da 12 V in grado di erogare corrente sufficiente ad alimentare i tre motori. In Figura 12 sono riportati i vari elementi collegati tra loro e i pin della scheda Arduino coinvolti nello scambio di segnali.

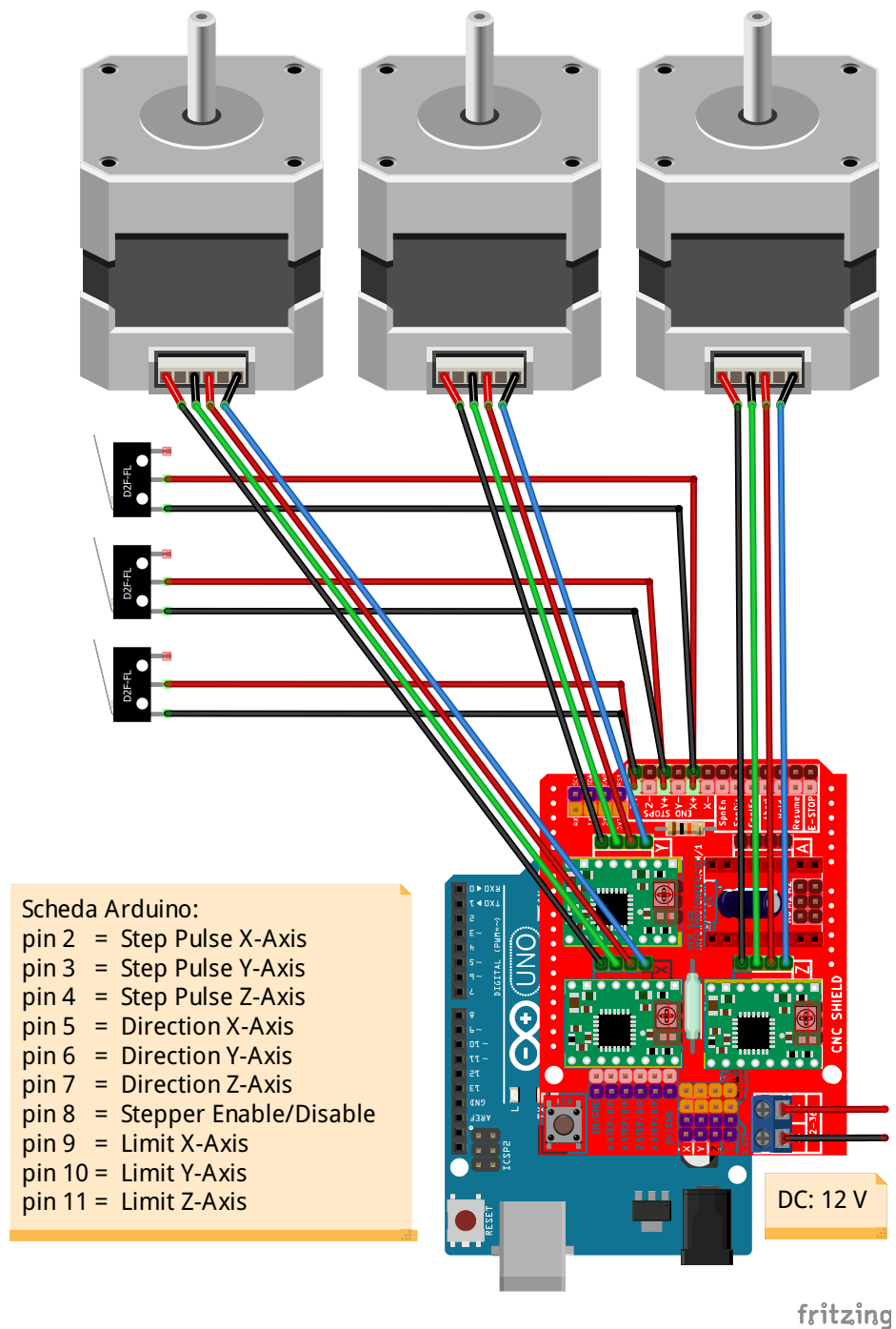


Figura 12: schema circuitale.

4 Software

Il software consiste essenzialmente in due script cooperanti (uno in esecuzione sul PC, l'altro sulla scheda Arduino), che comunicano tra loro per mezzo di messaggi inviati attraverso la porta seriale.

L'obiettivo del programma è quello di leggere e interpretare, riga dopo riga, le istruzioni in G-code memorizzate in un file di testo e di far assumere ai motori le posizioni richieste, coordinandone il movimento. Così facendo l'end-effector si sposterà da un punto dello spazio al successivo tracciando un'approssimazione della traiettoria voluta. Più traiettorie daranno luogo a un disegno.

Per far ciò il primo passo consiste nel trasformare un disegno in una sequenza di istruzioni che codificano l'insieme di punti su cui dovrà posizionarsi l'end-effector. Ciò può essere ottenuto facilmente con un software di grafica vettoriale, come ad esempio Inkscape. L'output del programma sarà un file di testo con righe del seguente tipo:

```
...
G01 X20.813110 Y-17.187250 Z-0.500000
G01 X20.293060 Y-19.078350 Z-0.500000
...
```

Python

Ottenuto il file bisogna adesso scrivere un programma per il PC che legga il testo un'istruzione per volta e comunichi la stringa di caratteri ad Arduino tramite porta seriale. In questa fase viene a crearsi un problema di sincronizzazione: il PC infatti, una volta comunicato il comando, deve attendere che Arduino esegua le elaborazioni sui dati ricevuti e posizioni i motori dove richiesto prima di poter inviare il comando successivo. Per tener conto di ciò il programma, dopo aver inviato l'istruzione, si mette in ascolto sulla porta seriale e attende finché non riceve il messaggio "Positioned" da parte della scheda. Dopodiché il ciclo si ripete uguale a se stesso fino a completa lettura del file. Questo script è stato scritto in Python ed è accessibile nella cartella "Python" col nome "serial_com". Può essere lanciato tramite l'interfaccia grafica IDLE.

Arduino - C

Ora che Arduino può ricevere l'istruzione in G-code, ciò che resta da fare è scrivere un programma in C da caricare sulla scheda che permetta di estrarre dalla stringa di caratteri le informazioni numeriche relative alla posizione dell'end-effector, le usi per calcolarsi gli spostamenti angolari dei motori attraverso le equazioni della cinematica inversa e invii ai driver segnali contenenti informazioni su direzione del moto e numero di passi da far effettuare ai motori stepper.

Di seguito viene riportato lo script in pseudo-linguaggio in modo da chiarire le operazioni fondamentali che il microcontrollore esegue.

All'avvio viene lanciata la funzione homing. Questa è una procedura che porta il robot in una configurazione nota (home), usata come riferimento per ogni posizionamento

futuro. Sostanzialmente i motori iniziano a ruotare facendo traslare le guide prismatiche verso l'alto finché queste non incontrano gli endstop, i quali innescano l'arresto di ogni movimento. Ovviamente gli endstop vanno fissati sulla struttura con una buona precisione in modo che risultino tutti alla stessa altezza.

Dopodiché si entra in un loop in cui Arduino si pone in ascolto sulla seriale e non appena arriva un nuovo comando esegue una serie di operazioni ben definite. Per prima cosa avvia la procedura di parsing. Questa consiste nel frammentare la stringa ricevuta in sottoparti. Il carattere che indica il punto in cui effettuare la separazione è lo spazio. Data la struttura tipica di una riga di G-code ci si trova quindi con un frammento costituito da una lettera seguita da un numero, e.g. "Y-19.078350". Interpretata la lettera, questa viene eliminata e ciò che resta (caratteri che rappresentano numeri) viene convertito in una variabile numerica vera e propria. Così per ogni frammento generato. Ottenute le variabili numeriche il codice prosegue col calcolo della cinematica inversa e col posizionamento dei motori. Quest'ultima operazione viene effettuata usando delle comode funzioni definite nelle librerie Accelstepper.h e Multistepper.h di Arduino che semplicemente generano dei segnali HIGH/LOW da inviare sui pin STEP e DIR di ciascun driver. Ad ogni transizione basso-alto sul canale STEP corrisponde l'avanzamento di un passo del motore, mentre il livello del segnale sul canale DIR definisce il senso orario o antiorario della rotazione. Infine la scheda comunica attraverso la seriale il termine delle proprie operazioni scrivendo la stringa di caratteri "Positioned".

```
void setup() { // esegue in avvio
  inizializza pin;
  inizializza motori;
  inizializza seriale;

  stampa su seriale: "Homing . . .";
  avvia procedura di homing;
  stampa su seriale: "Homing Completed";
  stampa su seriale: "Start!";
}

void loop() { // esegue ciclicamente
  leggi da seriale;
  if (è presente un nuovo comando) {
    estrapola info numeriche dal comando; (parsing)
    stampa su seriale le info numeriche estrapolate;

    calcola cinematica inversa;
    posiziona i motori;
    stampa su seriale: "Positioned";
  }
}
```

Nella cartella "Arduino" è disponibile lo sketch completo chiamato "delteron_code" pronto per essere caricato sulla scheda.

In figura 13 è riportato uno schema che illustra visivamente quanto finora esposto.

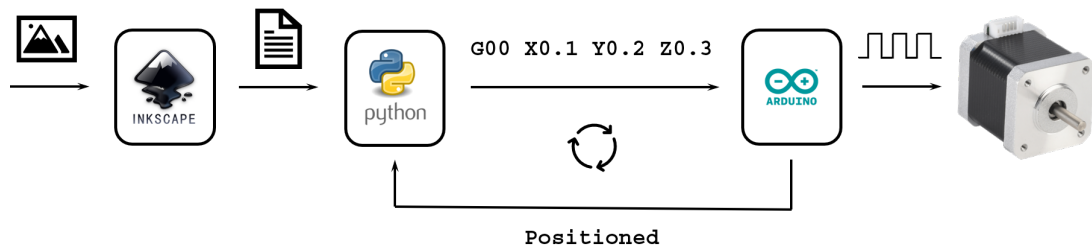


Figura 13

Nella cartella "Media" è possibile trovare il video del robot in funzione mentre è intento a tracciare i lineamenti di un volto.

5 Suggerimenti Pratici

Prima di avviare lo script in Python "serial_com" si consiglia di controllare il nome della porta seriale alla quale è connesso il dispositivo ed eventualmente, se diverso da quello riportato nel codice, modificarlo. Stessa cosa per quanto riguarda il nome del file G-code che rappresenta il disegno da tracciare.

```
port = 'COM6'
file_name = 'ritratto.ngc'
```

Prima di posizionare la penna sull'end-effector si suggerisce di mandare in esecuzione tramite lo script "serial_com" il file di testo "origin.ngc" contenuto nella cartella "Python". Questo file riporta un'unica istruzione in G-code: G00 X0.0 Y0.0 Z0.0. Quando eseguito, il comando fa posizionare l'end-effector in prossimità del piano di disegno. In questa posizione si può fissare la penna sul gripper facendo in modo che la punta sfiori la superficie del foglio. Controllare inoltre che il ripiano del foglio sia ben livellato.

Si consiglia poi, in fase di generazione del file G-code del disegno, di settare un valore di Z negativo ma piccolo in modulo (e.g. -0.5 mm) in modo che la punta della penna penetri nel foglio esercitando una leggera pressione.

Si suggerisce infine di usare un software per la visualizzazione dei listati di G-code (e.g. CAMotics) in modo da verificare in anticipo che tutte le istruzioni siano corrette e che il disegno sia ben posizionato e delle giuste dimensioni.