



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Meccanica

**Progettazione di un sistema mecatronico per
la gestione del movimento delle palpebre
di un robot umanoide**

**Design of a mechatronic system for the motion handling
of humanoid robot eyelids**

Relatore:

Prof. Matteo Claudio Palpacelli

Tesi di Laurea di:

Andrea Tisba

Anno Accademico 2020 / 2021

Abstract

L'obiettivo di questo lavoro di tesi è quello di analizzare il modo in cui è possibile dare espressività al volto di un robot umanoide proponendo il progetto funzionale di un sistema meccatronico in grado di gestire in maniera controllata il movimento delle relative palpebre, e gettando le basi per il successivo traguardo del movimento oculare. Considerando che i robot stanno acquistando sempre più spazio nella società è necessario migliorare la loro interazione con l'uomo aggiungendo delle funzionalità per renderli in maggior misura simili agli esseri umani come le emozioni che di solito vengono manifestate tramite le espressioni facciali. In questo aspetto il sistema oculare ha un ruolo importante dato che implementa le emozioni e permette una forma di comunicazione non verbale basata su un semplice sguardo oppure su un battito di ciglia. Quindi gli occhi di un robot umanoide devono essere in grado di produrre determinati movimenti che si possono realizzare attraverso diversi mezzi tecnologici. Nel presente lavoro si è utilizzata la scheda Arduino per progettare un piccolo banco meccatronico finalizzato alla movimentazione controllata della palpebra di un occhio robotico; la scheda alimenta un servomotore che deve muovere l'intero sistema e siccome il tutto deve posizionarsi all'interno di una testa di un bambolotto o di un androide si verifica nei risultati la configurazione del dispositivo adatta a produrre il movimento richiesto. Inoltre non sono state presenti grandi difficoltà anche perché il meccanismo progettato è molto semplice e può essere sviluppato rendendo più complessi e realistici i componenti del sistema, ad esempio gli occhi, o aggiungendo servomotori per un maggior controllo e per aumentare le funzionalità.

Indice

1 Introduzione	1
1.1 Scopo della tesi	1
1.2 Discussione generale	2
1.2.1 Alcuni cenni storici	2
1.3 Sistema oculare	3
1.3.1 Sguardo e battito di ciglia	5
1.3.2 Alcuni esempi	6
1.4 Organizzazione del lavoro	8
2 Tecnologie	9
2.1 Sistema di movimentazione palpebre	9
2.2 Descrizione Arduino	10
2.2.1 Hardware e alimentazione	11
2.3 Componenti hardware del progetto	12
2.3.1 Scheda Arduino Uno	12
2.3.2 Servomotore	13
2.3.3 Collegamento servomotore	14
2.4 Componenti software del sistema	15
2.4.1 IDE di Arduino	15
2.4.2 Solid Edge	17
2.4.3 Foglio di lavoro Excel	19

3 Realizzazione	20
3.1 Programma per muovere il servomotore	20
3.1.1 Aggiunta del potenziometro	22
3.2 Progettazione del banco mecatronico	24
3.2.1 Telaio struttura	25
3.2.2 Sostegno di supporto occhio	27
3.2.3 Occhi	27
3.2.4 Palpebre	28
3.3 Assemblaggio componenti	29
3.4 Analisi di posizione	31
4 Risultati e conclusioni	33
4.1 Configurazione dispositivo	33
4.2 Conclusioni	35
Bibliografia	37

Capitolo 1

Introduzione

I robot stanno assumendo un ruolo sempre più importante nella società odierna tanto che la loro incursione nella nostra vita quotidiana sta diventando inevitabile e nella maggior parte dei casi sempre più indispensabile: ad esempio, i robot di servizio sono utilizzati per soddisfare una vasta gamma di requisiti e nella maggior parte delle applicazioni dovrebbero interagire con l'essere umano. Per facilitare ciò si sviluppano sempre più robot umanoidi, definiti così poiché presentano delle sembianze umane. Infatti i robot, per migliorare l'interazione con l'uomo, richiedono un aspetto umano e delle funzionalità aggiuntive come emozioni, visione e interazione vocale insieme a movimenti degli occhi e della bocca. Inoltre essere in grado di muovere i bulbi oculari, le palpebre, le sopracciglia e la bocca permette di definire un gran numero di emozioni che sono generate come risultato della loro combinazione.

1.1 Scopo della tesi

Lo scopo principale di questa tesi è quello di studiare l'espressività nel volto di un robot umanoide e in particolare di analizzare la tematica legata alla movimentazione delle orbite e delle palpebre. In questo caso si parte con un discorso generale dell'argomento e successivamente si entra più nello specifico, progettando un piccolo banco mecatronico finalizzato alla movimentazione controllata della palpebra di un occhio

robotico tramite Arduino, in modo da fornire delle indicazioni pratiche che possono essere riprese e sviluppate.

1.2 Discussione generale

L'interazione naturale uomo-robot sta diventando sempre più importante quando robot e persone iniziano a coesistere e condividere in modo cooperativo una varietà di compiti.

Un robot per essere emotivamente intelligente dovrebbe essere capace di comprendere le emozioni umane e mostrare le proprie, di solito, usando espressioni facciali.

Quest'ultime sono il modo più espressivo con cui gli esseri umani presentano le emozioni dato che possono fornire spunti sensibili e significativi sulla risposta emotiva e svolgono un ruolo importante nella comunicazione non verbale. Tuttavia le manifestazioni delle facce robotiche generalmente non sono realistiche a causa di limiti della progettazione e del controllo mecatronico: la transizione da un'espressione all'altra tende a essere brusca e avviene in modo rapido e improvviso. I principali elementi del viso utilizzati per comunicare emozioni sono la bocca grazie anche alle labbra, le guance, gli occhi, le sopracciglia, il collo e la fronte.

Per realizzare espressioni facciali su un robot la ricerca si è focalizzata principalmente su algoritmi e programmi computazionali come la sintesi gestuale da un linguaggio dei segni per umanoidi animati, sintesi di immagini da audio e video per la realtà virtuale e l'interazione tra essere umano e robot tenendo conto della reazione di un gruppo di soggetti. I diversi sforzi fatti non contribuiscono alla progettazione dell'espressione facciale artificiale sugli umanoidi. Infatti la mappatura delle emozioni facciali da un essere umano a un robot si è rivelata efficace per la generazione di caratteristiche nel volto di sistemi robotici autonomi.

1.2.1 Alcuni Cenni storici

L'analisi dell'espressione facciale ha origine da Darwin nel 19° secolo quando propose il concetto universale di espressione facciale in Uomo e Animali. Per ragioni sia teoriche che pratiche, alcuni ricercatori definiscono le emozioni secondo una o più dimensioni. Una versione popolare di questo è il modello dimensionale di Russell, 1979, in cui si utilizzano i modelli dell'eccitazione e della valenza. Nel 1980 Plutchik

offre un modello tridimensionale in cui dispone le emozioni in cerchi concentrici e i cerchi interni sono più basilari, mentre i cerchi esterni più complessi; in particolare quest'ultimi sono formati anche mescolando le emozioni dei cerchi interni.

Nel 1982 Ekman ha sviluppato il sistema di codifica dell'azione facciale (FACS) per sistemare in modo discreto ogni espressione facciale umana. In più Ekman ha ideato un elenco base di emozioni contenente rabbia, disgusto, paura, gioia, tristezza e sorpresa (figura 1.1).

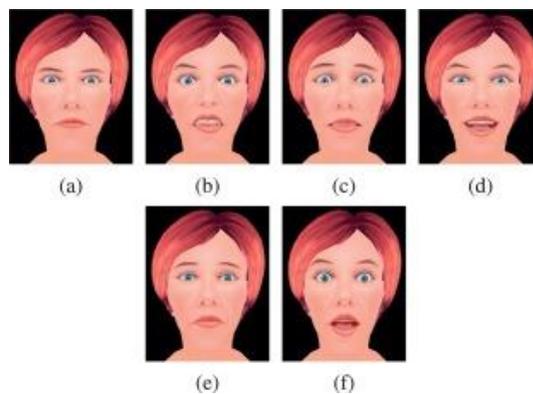


Figura 1.1. Le sei espressioni facciali di base alla loro massima intensità: a) rabbia, b) disgusto, c) paura, d) gioia, e) tristezza, f) sorpresa.

Invece per quanto riguarda i robot umanoidi, dal 1992 alcuni scienziati hanno iniziato la ricerca su robot antropomorfi nel progetto umanoide al centro di ricerca avanzata per la scienza e l'ingegneria presso la *Waseda University*, in Giappone. Nel 1996-1997 hanno sviluppato Hadaly-2 che è capace di eseguire una varietà di azioni mentre interagisce con gli umani; inoltre questo robot umanoide è in grado di dialogare con i visitatori utilizzando un sistema di elaborazione vocale e di visione.

Negli ultimi anni grazie ai progressi della tecnologia informatica e dell'intelligenza artificiale si stanno incrementando i robot che riescono a comunicare e interagire con gli esseri umani in modo emotivo e sociale: nel 2008 sono stati realizzati 8,6 milioni di agenti robotici, compresi i robot sociali e umanoidi.

1.3 Sistema oculare

Gli occhi hanno un ruolo importante nel fornire un aspetto simile a quello umano alla testa del robot, però sono oggetti molto complessi e intricati. Un automa per assomigliare a un essere umano deve possedere gli occhi e questi dovrebbero includere una certa complessità nei dettagli della superficie, nella forma dell'occhio, nel bulbo oculare, nell'iride e nella pupilla. Il sistema oculare è fondamentale nell'interazione uomo-robot; innanzitutto si utilizzano gli occhi per mostrare le espressioni facciali e dunque si possono implementare le emozioni umane. Nella figura 1.2 sono mostrate alcune emozioni usando gli occhi.

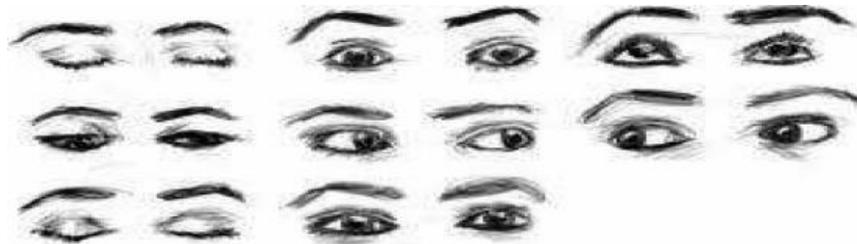


Figura 1.2. Emozioni umane usando gli occhi

Inoltre gli occhi possono essere impiegati per consentire l'interazione basata sullo sguardo che è altrettanto importante come forma di divulgazione non verbale. In particolare quando due persone incrociano lo sguardo hanno un contatto visivo che permette la comunicazione come il cercare informazioni, ad esempio per capire come qualcosa che diciamo viene ricevuto oppure per regolare l'interazione con gli altri, segnalando quando è il turno di qualcun altro di parlare. Pertanto è necessario che i robot umanoidi abbiano comportamenti dello sguardo simili a quelli umani così da poter emulare la comunicazione umana.

Un'altra ragione per dotare il robot di occhi è che questi possono essere utilizzati come un sistema di visione attivo. Anche le palpebre danno un buon contributo nell'esprimere emozioni dal momento che consentono agli occhi di aprirsi e chiudersi a vari gradi.

I bulbi oculari sono progettati per migliorare l'interazione con le persone sulla base di dati antropomorfici umani.

1.3.1 Sguardo e battito di ciglia

Lo sguardo degli occhi è una delle caratteristiche più basilari e importanti del volto umano; infatti le persone incorporano lo sguardo sia consciamente che inconsciamente in vari schemi di interazione uomo-uomo. Lo sguardo svolge diverse funzioni come catturare l'attenzione, mantenere il coinvolgimento, trasmettere informazioni su stato emotivo e mentale e aumento verbale di comunicazione.

Considerando l'importanza dello sguardo nell'interazione sociale, non è sorprendente che si sia studiato il suo comportamento in molte piattaforme robotiche che controllano lo sguardo adoperando attuatori nei bulbi oculari. Tuttavia, questi attuatori potrebbero non essere abbastanza veloci o precisi da replicare il movimento degli occhi umani. D'altra parte, animazioni in computer grafica hanno la capacità di produrre uno sguardo dall'aspetto più naturale, anche se la mancanza di incarnazione e presenza fisica può limitare la percezione dello sguardo degli agenti virtuali.

Quando si analizzano i comportamenti dello sguardo per i robot umanoidi ci sono due problemi da considerare: il primo è la pianificazione dei movimenti; il secondo è quello di controllare lo sguardo dell'androide per seguire la traiettoria pianificata. Per risolvere ciò è necessario compiere una calibrazione dello sguardo (figura 1.3).

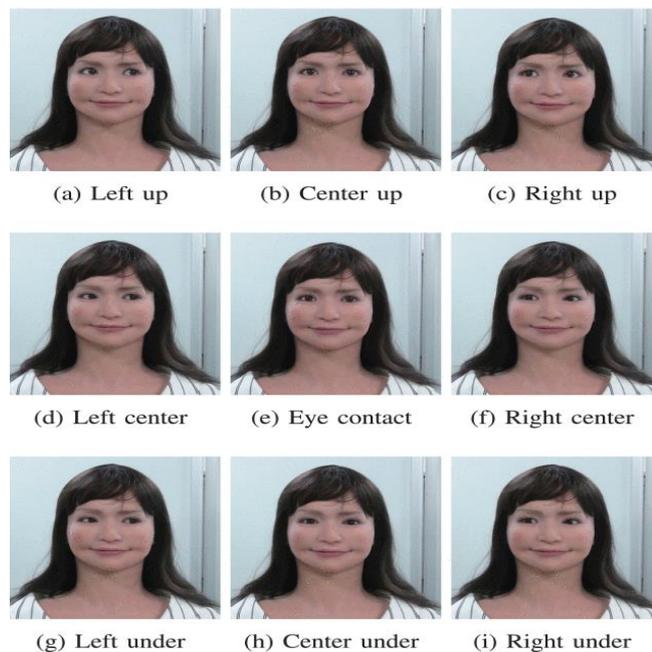


Figura 1.3. Nove direzioni dello sguardo in un robot umanoide

Il battito di ciglia ha una valenza altrettanto importante nella comunicazione così le palpebre sono state implementate in diversi robot. Tuttavia sono state condotte pochissime indagini su come modellare il battito di ciglia umano per robot con occhi fisici e dunque il movimento delle palpebre è stato aggiunto in modo casuale all'interazione sociale. Questo è dovuto in gran parte alle restrizioni tecniche imposte dagli occhi fisici robotici e alla complessità dei fattori che influenzano il battito delle ciglia negli esseri umani.

Ciononostante, grazie alla ricerca fisiologica sulle varie dipendenze del comportamento delle palpebre dovuto a diversi fattori fisiologici e psicologici, si sono raggiunti vari risultati che possono essere impiegati per modellare il battito di ciglia nei robot.

In sintesi il movimento delle palpebre viene descritto come: una funzione di variabili fisiologiche, ad esempio la velocità media di un singolo battito di ciglia, la frequenza media di ammiccamento e la durata media degli intervalli di ammiccamento tra gli occhi; una funzione di variabili di stato del sistema, esempio cambiamenti nell'espressione facciale e nella comunicazione verbale; una funzione delle informazioni del contesto sociale, per esempio durante la lettura, conversando oppure guardando qualcosa di specifico; una funzione dello stato psicologico della persona che manifesta il comportamento; una funzione del comportamento dell'interlocutore sociale.

1.3.2 Alcuni esempi

Hadaly-2 è uno dei primi robot antropomorfi capace di comunicare con gli umani ed è dotato di un sistema testa-occhio chiamato WE-3R che permette agli occhi e alle palpebre di muoversi in maniera coordinata e regola gli angoli di apertura di quest'ultime secondo la luminosità di un oggetto. Inoltre le ciglia svolgono un ruolo importante nell'adattare la luminosità di un oggetto sulla retina e nel chiudere improvvisamente gli occhi quando qualcuno tocca le palpebre le quali sono costituite da una cornice in duralluminio e da una copertura in gomma simile alla pelle. Un tendine costituito da filo e puleggia guida il meccanismo utilizzando un motore a corrente continua e una molla è impiegata per il bulbo oculare; invece le ciglia sono realizzate con fili metallici collegati a un modulo di rilevamento tattile.

Per il robot iCub si è sviluppato un modulo per sbattere le palpebre simile a quello umano. Il robot ha occhi molto pronunciati che ricordano le sembianze umane come la pupilla nera, la sclera bianca e palpebre superiori e inferiori mobili (figura 1.4).



Figura 1.4. a) iCub occhi; b) iCub meccanismo delle palpebre

Il meccanismo delle palpebre è comandato da un servomotore e costruito in modo tale che entrambi gli occhi si chiudano contemporaneamente e le ciglia si incontrino al centro dei bulbi oculari. Le palpebre possono essere controllate di continuo da una posizione aperta a una chiusa e si possono impartire dei comandi per aprirle o chiuderle completamente. Il battito di ciglia viene diviso in tre fasi: la prima quando l'occhio si chiude; la seconda corrispondente all'occhio chiuso; la terza nel momento in cui l'occhio si apre. A differenza dell'occhio umano, durante l'ammicciamento, le palpebre superiori e inferiori del robot si muovono allo stesso modo.

Il robot umanoide Actroid-SIT prodotto dalla Kokoro Company Ltd, invece, ha un aspetto molto simile a quello degli esseri umani e per imitarne i comportamenti sono stati utilizzati vari approcci (figura 1.5). Uno di questi, modella i movimenti dello sguardo utilizzando l'osservazione dei soggetti e prevede un moto della vista dagli stimoli visivi; inoltre adatta la relazione tra la frequenza dei movimenti oculari e quello dei battiti di ciglia. In questo androide la direzione degli occhi e i battiti di ciglia sono controllati indipendentemente. Per controllare lo sguardo si definiscono nove direzioni, quelle della figura 1.3, scegliendone una; passando dalla direzione del contatto visivo alle altre otto si imita il comportamento degli occhi. Per controllare le palpebre si seguono i tempi dei battiti di ciglia.



Figura 1.5. Robot umanoide Actroid-SIT

1.4 Organizzazione del lavoro

Nei capitoli successivi sono illustrate le tecnologie utilizzate per progettare il banco meccatronico di prova e viene descritta la sua realizzazione, indicando anche le difficoltà incontrate. Infine sono raccolti e presentati i risultati ottenuti che hanno come scopo quello di modificare, se necessario, la struttura per rispondere al fine richiesto.

Capitolo 2

Tecnologie

In questo capitolo sono illustrati i principali strumenti utilizzati per realizzare un piccolo banco mecatronico finalizzato alla movimentazione controllata della palpebra di un occhio robotico tramite Arduino.

2.1 Sistema di movimentazione palpebre

Il banco mecatronico ha lo scopo di analizzare la tematica legata alla movimentazione delle palpebre per iniziare a dare espressività al volto di un robot umanoide già discussa nel capitolo 1.3. Il progetto cerca di creare e condividere una serie di istruzioni per realizzare un'applicazione con Arduino in grado di controllare il movimento di apertura e di chiusura di una palpebra, utilizzando solo strumenti e materiali facilmente reperibili e usando una stampante 3D. Il telaio su cui deve essere appoggiato il motorino, che ha il compito di mobilitare il sistema, viene progettato su un software CAD (progettazione assistita dall'elaboratore) e può essere stampato in metallo o in un altro materiale: non è importante perché le dimensioni sono piccole dovute al fatto che bisogna inserirlo nella testa di un bambolotto o di un robot umanoide; di conseguenza anche il motorino deve essere di piccola taglia e quindi le velocità non sono molto elevate. Nel caso in cui non

si dovesse avere una stampante 3D, la struttura di sostegno si può realizzare con i comuni strumenti contenuti nella cassetta degli attrezzi; in questo modo si perde di precisione a causa delle lavorazioni e il materiale utilizzato può essere il legno poiché ha una lavorabilità migliore.

È importante aggiungere che il sistema CAD utilizzato per la creazione del banco meccatronico e il software di Arduino si possono scaricare nel computer in maniera completamente gratuita e ciò rende il progetto oltre che interessante per via della tematica trattata anche accessibile a quasi la totalità delle persone: può offrire una possibile base di partenza alla comunità di utenti di Arduino che possono sviluppare e condividere soluzioni per migliorare l'elaborato; inoltre può coinvolgere ed essere d'aiuto a persone che hanno meno dimestichezza e vogliono cimentarsi nell'argomento.

2.2 Descrizione Arduino

Arduino è una famiglia di schede elettroniche open source, cioè con licenza libera, a basso costo dotate di un microcontrollore con cui è possibile creare circuiti di vario tipo per molte applicazioni, la maggior parte nel campo della robotica e dell'automazione. Alla sua base si ha l'idea di rendere la creazione di progetti elettronici più veloce e agevole, ma soprattutto facile e alla portata di tutti: si possono realizzare in maniera relativamente rapida e semplice piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, automatismi per il controllo della temperatura e dell'umidità e molti altri meccanismi che utilizzano sensori, attuatori e comunicazione con altri apparecchi. Le schede di Arduino sono reperibili in commercio in forma preassemblata o come kit fai da te oppure si possono acquistare singoli componenti.

La scheda è associata a una parte software chiamata ambiente di sviluppo integrato (*Integrated Development Environment – IDE*) che deve essere installata sul computer e permette di programmare il microcontrollore, ossia scrivere il programma in codice informatico che verrà successivamente caricato nella scheda stessa. L'IDE aiuta lo sviluppatore segnalando errori di sintassi del codice in fase di scrittura e permettendo la loro correzione, oltre a una serie di strumenti e funzionalità di supporto tra le quali delle librerie generali per pilotare diversi tipi di sensori e attuatori. Inoltre utilizza un linguaggio di programmazione semplice, facile da imparare e intuitivo chiamato *Wiring*,

derivato dal C¹ e dal C++², il quale permette a neofiti del settore e anche a persone meno esperte di iniziare a programmare.

2.2.1 Hardware e alimentazione

Arduino è costituito da un microcontrollore, prodotto dalla Atmel, installato su una scheda con pin connessi alle porte input/output (I/O), un regolatore di tensione a 5 volt, un'interfaccia USB che permette di comunicare con il computer e molte volte viene incluso un oscillatore a cristallo a 16 MHz.

Il microcontrollore ha il compito di gestire tutta la scheda e in alcuni casi viene pre-programmato grazie ad un *bootloader*, un programma che nella fase di avvio carica il nucleo del sistema operativo dalla memoria secondaria alla memoria primaria (solitamente la RAM), il quale facilita il caricamento dei programmi sulla memoria flash incorporata nella scheda.

Quasi tutti i pin di I/O sono disposti sulla parte superiore della scheda e sono identificati mediante dei connettori femmina; le funzionalità di input/output consentono di effettuare uno scambio di dati o segnali grazie a queste interfacce le quali vengono rese disponibili da un sistema operativo per assicurare la comunicazione tra due apparati altrimenti incompatibili.

Tramite le funzionalità di input la scheda riceve impulsi raccolti da sensori esterni, mentre l'interazione con l'esterno avviene attraverso attuatori, come ad esempio i motori elettrici, pilotati dal programma.

L'alimentazione del circuito può avvenire con cavo USB che permette di collegare la porta presente sulla scheda di Arduino al computer oppure mediante un alimentatore in corrente continua di solito a 9 volt, costituito spesso da un contenitore di batterie, il quale si connette con un filo in un'altra porta di diametro minore rispetto alla

¹Linguaggio di programmazione imperativo di natura procedurale

²Linguaggio di programmazione *general purpose*, letteralmente 'scopo generale' traducibile come 'per uso generale'

precedente. In quest'ultimo caso la scheda commuta automaticamente sull'alimentazione esterna quando l'attacco dell'alimentatore è inserito, invece converte autonomamente sull'alimentazione USB nel caso in cui il connettore è disconnesso oppure assente.

2.3 Componenti hardware del progetto

Si elencano alcune delle parti utilizzate per realizzare il banco mecatronico che verranno approfondite successivamente: la scheda Arduino Uno e il servomotore. Inoltre è necessario un computer per il controllo della scheda e potrebbe essere utile disporre di una fonte di alimentazione esterna nel caso si usasse un motorino più potente.

2.3.1 Scheda Arduino Uno

Alla base del progetto è stata utilizzata la scheda Arduino Uno dato che è programmabile ed economica, quindi elevata disponibilità nel mercato (figura 2.1). Essa è dotata di 14 pin di I/O digitali numerati da 0 a 13 e situati nella parte superiore, 6 dei quali possono essere usati come segnali modulazione di larghezza d'impulso (*Pulse-width modulation – PWM*) con i quali è possibile, ad esempio, regolare la velocità di rotazione di un motorino elettrico. I pin digitali lavorano solamente con due valori, 0 e 1; i pin PWM si identificano sulla scheda con un numero preceduto da un cancelletto (esempio: ~9) e differiscono rispetto agli altri poiché, oltre a lavorare con i valori 0 e 1, si possono impiegare per controllare segnali analogici.

Inoltre è bene ricordarsi di stabilire attraverso apposite istruzioni da inserire nello *sketch* programmato sull'IDE la direzione di funzionamento delle porte utilizzate, input o output.

Sempre sulla Uno sono presenti altri 6 connettori dedicati a ingressi di segnali analogici che ricevono valori di tensione dati da sensori esterni i quali sono convertiti in 1024 livelli, da 0 a 1023. Essi sono posizionati nella parte bassa della scheda e accanto sono collocati i pin che riguardano l'alimentazione: 5V (5Volt) e GND sono i più utilizzati.

Molto importante è il pulsante RESET, situato di fianco ai pin digitali e sopra la porta che permette di collegare la scheda al computer, dato che serve per riavviare Arduino qualora si fosse bloccato o si fosse aggiornato il programma caricato.

Nel banco mecatronico la scheda Arduino Uno si occupa di ricevere i comandi forniti da un computer esterno su cui è installato l'IDE e di convertirli in segnali elettrici per il motorino. Essa ha il compito di controllare i movimenti del servomotore e può gestire eventuali input o output aggiuntivi, analogici o digitali. Ad esempio, al sistema si può aggiungere un potenziometro in modo da decidere manualmente dove posizionare il braccio del servomotore.

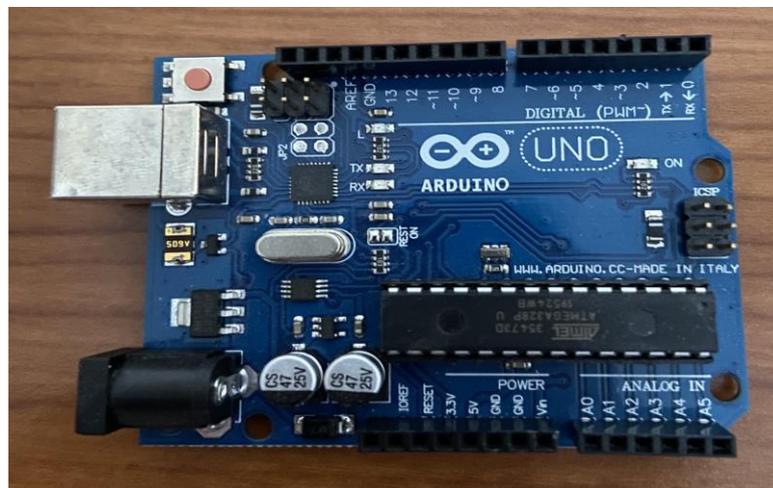


Figura 2.1. Scheda Arduino Uno utilizzata per controllare i movimenti del servomotore

2.3.2 Servomotore

Per muovere l'asta collegata alle due palpebre è stato utilizzato un servomotore SG90 peso 9 grammi che ha elevata coppia in uscita ed è ideale per principianti, ma anche per progetti di media complessità (figura 2.2). Infatti esistono molti modelli con differenti dimensioni, potenze e tipi di ingranaggi, tuttavia la maggior parte di loro ha un funzionamento molto simile indipendentemente dal modello scelto. Tipicamente il servo SG90 può effettuare una rotazione di 180° ed è dotato di diversi bracci intercambiabili in nylon: monoleva, bileva e croce; per il progetto si può utilizzare il braccio monoleva che viene fissato sulla testa del servomotore grazie a una vite.

È importante aggiungere che si utilizza solo un servo poiché è semplice da pilotare ed è sufficiente alimentarlo direttamente da Arduino; nel caso in cui il circuito sia composto da due o più servomotori si ha la necessità di adoperare un'alimentazione esterna e si deve ricordare di mettere in comune il polo negativo di quest'ultima e il pin GND della scheda Uno. Al posto del servomotore è consentito impiegare un motorino elettrico che, tuttavia, rende più complesso il circuito poiché si deve aggiungere: un amplificatore di corrente o di potenza elettrica chiamato mosfet dato che la potenza richiesta dal motore è maggiore rispetto a quella che può offrire Arduino; un dispositivo per regolare la velocità di un motore in corrente continua; un sistema a ponte H che permette di invertire il senso di marcia del motore.

Il servo SG90 è costituito da un unico cavo di lunghezza circa 25 cm composto da tre connettori di colori rosso, marrone, arancione e dotati di pin terminale femmina.



Figura 2.2. Servomotore utilizzato per muovere il sistema oculare

2.3.3 Collegamento servomotore

Il servo è connesso alla scheda Uno tramite una *breadboard*, basetta formata da fori. Essa è costituita da due righe etichettate dal + per indicare il polo positivo dell'alimentazione e dal – per denotare quello negativo, situate in alto e in basso; invece

le file centrali sono contrassegnate da lettere e sono utilizzate per realizzare il circuito. Dopo aver collegato i pin 5V e GND della scheda rispettivamente con la riga positiva e quella negativa della basetta, si possono congiungere i tre connettori del servomotore: il marrone con un foro della linea negativa; il rosso con uno di quella positiva, mentre l'arancione con uno dei pin PWM che in base alla modulazione della larghezza dell'impulso impostano l'angolo di uscita sul servomotore (figura 2.3).

Nell'applicazione viene preso il pin numero 9 e il braccio monoleva deve variare di un angolo compreso tra 0° e 90° .

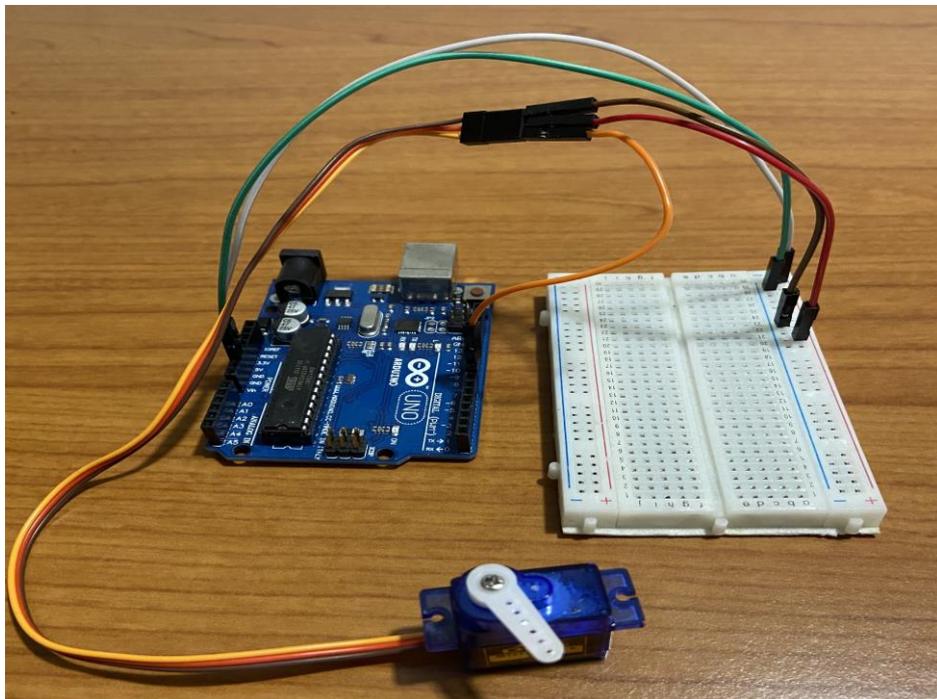


Figura 2.3. Circuito realizzato collegando il servomotore con la scheda Arduino Uno

2.4 Componenti software del sistema

Di seguito vengono elencate le componenti software utilizzate per controllare la scheda e per realizzare il banco mecatronico.

2.4.1 IDE di Arduino

L'ambiente di sviluppo integrato viene scaricato e installato direttamente sul computer in maniera gratuita come è stato detto in precedenza.

Terminato il download è necessario procedere con l'installazione guida per poter usufruire al meglio del software.

Nell'interfaccia principale sono presenti vari pulsanti che hanno il compito di assistere e aiutare lo sviluppatore a scrivere il programma che poi verrà caricato sulla scheda. I più rilevanti sono: verifica che serve a valutare la correttezza del codice digitato; carica che permette di caricare lo sketch creato nella scheda; nuovo che serve per aprire un documento di lavoro pulito e inutilizzato; apri che permette di aprire un programma realizzato; salva utile a salvare volta per volta un programma in corso di scrittura; monitor seriale il quale permette di visualizzare dei valori ottenuti, ad esempio, da un sensore collegato all'hardware di Arduino (figura 2.4).



Figura 2.4. Pulsanti nell'interfaccia principale del software di Arduino

L'IDE richiede all'utente di definire soltanto le due funzioni SETUP e LOOP: la prima viene chiamata una sola volta all'inizio di un programma e può essere utilizzata per i settaggi iniziali; la seconda invece viene chiamata ripetutamente e la sua esecuzione si interrompe solamente con lo spegnimento della scheda (figura 2.5).

Al di sopra della funzione SETUP si definiscono le variabili utilizzate nel programma e si possono includere delle librerie per semplificare la comunicazione quando è presente un sensore o un dispositivo esterno come, per esempio uno schermo LCD o un servomotore. Inoltre nell'IDE si possono esprimere dei commenti che rappresentano una parte di testo non eseguita dal processore e vengono utilizzati per descrivere il codice o per aiutare a comprendere parti di programma: una linea di commento viene indicata con il simbolo //.

La programmazione viene assistita dall'*utility* dell'IDE: indentazione automatica che consente di inserire un rientro di una linea di testo; controllo parentesi ovvero ad ogni parentesi aperta deve corrisponderne una chiusa, uno sbilanciamento di parentesi impedisce la compilazione e quindi l'esecuzione del programma; *syntax highlighting* che evidenzia automaticamente le parole riconosciute dal compilatore, dunque è importante anche l'utilizzo della lettera maiuscola o minuscola.

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

Figura 2.5. Funzioni SETUP e LOOP con alcuni esempi di linee di commento

Nel progetto l'IDE è stato utilizzato per redigere e caricare il codice sulla scheda Arduino Uno. A differenza delle schede di Arduino che sono numerose e vengono impiegate per diversi progetti, la scelta viene eseguita in base all'utilizzo e alle esigenze, il software è il medesimo per tutti i tipi di hardware. Infatti la scheda utilizzata deve essere selezionata nell'interfaccia principale sulla voce strumenti, altrimenti non è possibile compilare e caricare il codice all'interno della scheda stessa. In aggiunta, sempre nella sezione strumenti, si deve indicare la porta seriale di ingresso quando si collega Arduino al computer.

2.4.2 Solid Edge

Software di progettazione 3D sviluppato dall'azienda Siemens che può essere scaricato gratuitamente da determinate persone quali studenti, insegnanti, ma anche Startup o nella versione di prova, altrimenti è necessario possedere una licenza per poterlo utilizzare. Le eccezionali funzionalità di modellazione e di gestione dei flussi di lavoro

e l'ambiente di gestione della progettazione rendono Solid Edge la migliore piattaforma per l'esecuzione di progetti accurati, per questo motivo è stato utilizzato per costruire un prototipo di sistema oculare robotico.

Infatti la progettazione con Solid Edge è più semplice e rapida grazie agli strumenti di modellazione e gestione degli assiemi che consentono di sviluppare con facilità una vasta gamma di prodotti, da singoli componenti ad assiemi contenenti una grande quantità di parti; inoltre nel sito dove viene scaricato si possono trovare numerose esercitazioni guidate per apprendere in modo ancora più veloce il suo utilizzo.

Con Solid Edge i prodotti vengono realizzati nel migliore dei modi e nel minor tempo, anche attraverso comandi personalizzati e flussi di lavoro strutturati che accelerano lo sviluppo di feature comuni impiegate in specifici settori e assicurano il corretto funzionamento e posizionamento dei componenti; questi vengono progettati, analizzati e modificati nell'ambito del modello di assieme (figura 2.6).



Figura 2.6. Interfaccia di Solid Edge in cui è possibile scegliere l'ambiente migliore per realizzare un progetto

2.4.3 Foglio di lavoro Excel

Viene utilizzato Excel per determinare una configurazione accettabile e realizzabile del dispositivo. Sul foglio vengono riportati i dati relativi alle misure del sistema progettato su Solid Edge che vengono in seguito analizzati tramite delle formule matematiche e se necessario modificati, ciò prevede un aggiornamento del progetto.

Ulteriori dettagli sono forniti nel capitolo successivo.

Capitolo 3

Realizzazione

In questo capitolo vengono presentate con maggiore dettaglio le tecnologie introdotte nel capitolo precedente, in particolare si descrive il programma in grado di muovere il servomotore, la realizzazione del banco mecatronico e l'analisi condotta con le relative formule utilizzate per ricavare una configurazione praticabile del sistema.

3.1 Programma per muovere il servomotore

In figura 3.1 si può vedere il programma creato con l'IDE di Arduino che viene in seguito caricato sulla scheda Uno per mettere in movimento il servomotore e quindi l'intero sistema.

Si può notare che il codice digitato non è molto lungo e neanche complesso, tuttavia utilizza un linguaggio e delle funzioni ben precise. In primo luogo è necessario includere la libreria Servo per semplificare la comunicazione con il servomotore: in questo modo non si deve codificare la frequenza del segnale, ma è sufficiente specificare l'angolo per ruotare il motorino. Solitamente la libreria in questione viene già scaricata insieme all'IDE e ad altre librerie di maggior uso ed è possibile selezionarla nella sezione sketch sulla voce `#include libreria`.

```
servomotore
1 #include <Servo.h>
2 #define PIN_SERVO 9
3
4 Servo servo;
5
6 void setup() {
7   servo.attach(PIN_SERVO);
8 }
9
10 void loop() {
11   for(int i = 0; i < 91;i++){
12     servo.write(i);
13     delay(20);
14   }
15   for(int i = 90; i >= 0;i--){
16     servo.write(i);
17     delay(50);
18   }
19 }
```

Figura 3.1. Sketch per muovere il servomotore

Nella seconda riga del programma si definisce il pin per controllare il servomotore che, come detto nel paragrafo 2.3.3, è il pin PWM numero 9.

Successivamente si crea un oggetto della classe servo e in particolare il servo con la lettera iniziale maiuscola specifica la classe dell'argomento, mentre il secondo scritto completamente in minuscolo indica il nome che può essere scelto a piacere: si può chiamare servo1 se vengono utilizzati più servomotori.

Nella funzione SETUP l'oggetto appena creato viene collegato con il pin 9 e così viene utilizzato per comunicare con il servomotore; in seguito si può uscire dalla funzione senza aggiungere nient'altro dal momento che non sono presenti altre variabili da descrivere. Invece nella funzione LOOP si scrive il codice per ruotare il braccio monoleva, posizionato sopra il servomotore, in maniera continua e in grado di spazzare un angolo da 0° a 90°.

Quindi si introducono due cicli *for*: il primo ha il compito di far avanzare il braccio dall'angolo nullo fino a 90°, mentre il secondo ha un comportamento inverso, cioè deve

riportare il sistema indietro fino ad arrivare al punto di partenza che corrisponde a 0°. Il metodo *write* della classe servo imposta la posizione del servomotore ricevendo come unico parametro l'angolo, espresso in gradi, in cui si vuole mettere il braccio del motorino; quest'ultimo ad ogni ciclo nel primo for viene disposto in una diversa collocazione e incrementato fino a 90°. Raggiunta questa quota si passa al secondo for dove a ogni interazione del periodo l'angolo diminuisce sino a 0°.

La funzione *delay* sospende l'esecuzione del programma per il tempo in millisecondi passato tra le parentesi, quindi ha lo scopo di rallentare leggermente il movimento del servomotore. Naturalmente i delay nei due cicli hanno un valore soggettivo e si possono modificare a seconda che si voglia una rotazione più veloce o più lenta.

Siccome la funzione LOOP viene eseguita di continuo, terminato il secondo ciclo for il servomotore ripartirà ruotando nuovamente da 0° a 90° e proseguirà a girare avanti e indietro fino a quando non verrà tolta l'alimentazione alla scheda.

L'applicazione appena creata permette di aprire e chiudere gli occhi in modo ripetuto, tuttavia il braccio del servomotore si muove da 0° fino a 90° e viceversa, quindi non può assumere una determinata posizione iniziale o finale.

3.1.1 Aggiunta del potenziometro

Per poter controllare manualmente il motorino e dunque decidere in quale punto posizionare il braccio o quanto deve ruotare si può utilizzare un potenziometro. Esso viene, dapprima, inserito sulla basetta nelle righe centrali: il piedino a sinistra si collega con un foro della linea negativa dell'alimentazione; quello a destra con un foro della linea positiva e infine il piedino centrale viene associato a uno dei pin analogici della scheda Uno, il pin A0 (figura 3.2); dopodiché si modifica il programma nell'IDE di Arduino.

In figura 3.3 si ha lo sketch alterato dovuto alla presenza del potenziometro di cui si specifica il pin connesso alla scheda nella parte iniziale, insieme al pin del servomotore. Si nota che il contenuto della funzione SETUP può rimanere invariato, mentre la funzione LOOP viene modificata completamente. All'interno di quest'ultima si legge il valore del potenziometro tramite la prima istruzione; in seguito si esegue una conversione utilizzando la funzione *map* dal momento che i valori in ingresso analogici variano da 0 a 1023 e invece devono mutare da 0 a 90: il primo termine dentro le

parentesi rappresenta il valore da convertire, i due successivi indicano gli estremi dell'intervallo di partenza, mentre gli ultimi due denotano gli estremi di destinazione. La trasformazione avviene in modo proporzionale e il valore ritornato viene usato nell'istruzione successiva, funzione write della classe servo, che permette di spostare il braccio monoleva nella posizione indicata dal potenziometro. Infine si aggiunge, anche in questo caso, un leggero ritardo.

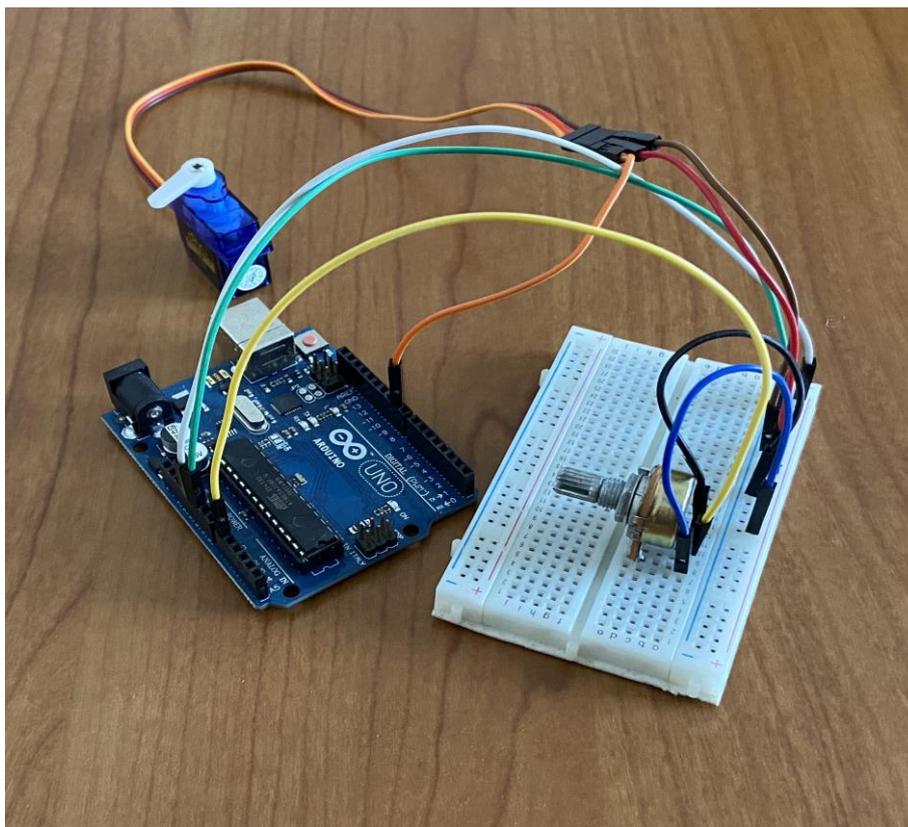


Figura 3.2. Circuito modificato con l'introduzione di un potenziometro

Il sistema con il potenziometro ha il vantaggio di poter impostare manualmente la posizione del braccio e decidere di quanto farlo ruotare, ciò significa che le palpebre possono essere individuate inizialmente anche in una situazione intermedia e si possono socchiudere o aprire in modo parziale; tuttavia il circuito va collocato nella testa di un bambolotto e quindi risulta difficile andarlo a manipolare. Una soluzione potrebbe essere quella di sistemare solamente il servomotore sulla struttura da inserire nella testa e invece disporre la scheda Arduino Uno con la basetta all'esterno; ciononostante si

tratta di un metodo molto complesso dato che necessita di un gran numero di connettori per realizzare i collegamenti e di conseguenza si rischia di intasare il sistema.

```
servo_potenziometro
1 #include <Servo.h>
2 #define PIN_SERVO 9
3 #define PIN_POTENZIOMETRO A0
4
5 Servo servo;
6
7 void setup() {
8   servo.attach(PIN_SERVO);
9 }
10
11 void loop() {
12   int input = analogRead(PIN_POTENZIOMETRO);
13   int angolo = map(input, 0, 1023, 0, 90);
14   servo.write(angolo);
15   delay(20);
16 }
```

Figura 3.3. Sketch per muovere il servomotore con potenziometro

3.2 Progettazione del banco mecatronico

Dopo aver realizzato il programma per controllare e spostare il servomotore si crea una struttura in cui installare il motorino e finalizzata alla movimentazione regolata della palpebra. Un buon consiglio è di eseguire uno schizzo su un foglio bianco cartaceo prima di avviare il lavoro sul software CAD, così da avere già una buona base da utilizzare come linea guida e quindi proseguire in maniera più spedita con meno intoppi. Dopodiché si passa alla modellazione 3D su Solid Edge, progettando i vari componenti singolarmente che in seguito verranno uniti nel modello assieme; in particolare si realizza: un telaio su cui si appoggia il servomotore; due palpebre che sono collegate tra loro da un perno e sono connesse al braccio del motorino tramite un'asta; due occhi, per realizzarne uno è sufficiente una sfera e si può provare a creare l'iride; due sostegni di supporto necessari a congiungere gli occhi con il telaio.

Si può notare come vengono progettati solamente i componenti necessari a produrre un prototipo di prova che, dunque, risulta semplice e poco articolato, ma deve essere in grado di rispondere al fine richiesto.

Di seguito vengono descritti i vari pezzi e la loro realizzazione.

3.2.1 Telaio struttura

Il telaio è una parte molto importante dato che deve sorreggere l'intera struttura e su di esso vanno posizionati il servomotore e la scheda Arduino Uno; pertanto viene creato tramite diverse operazioni (figura 3.4).

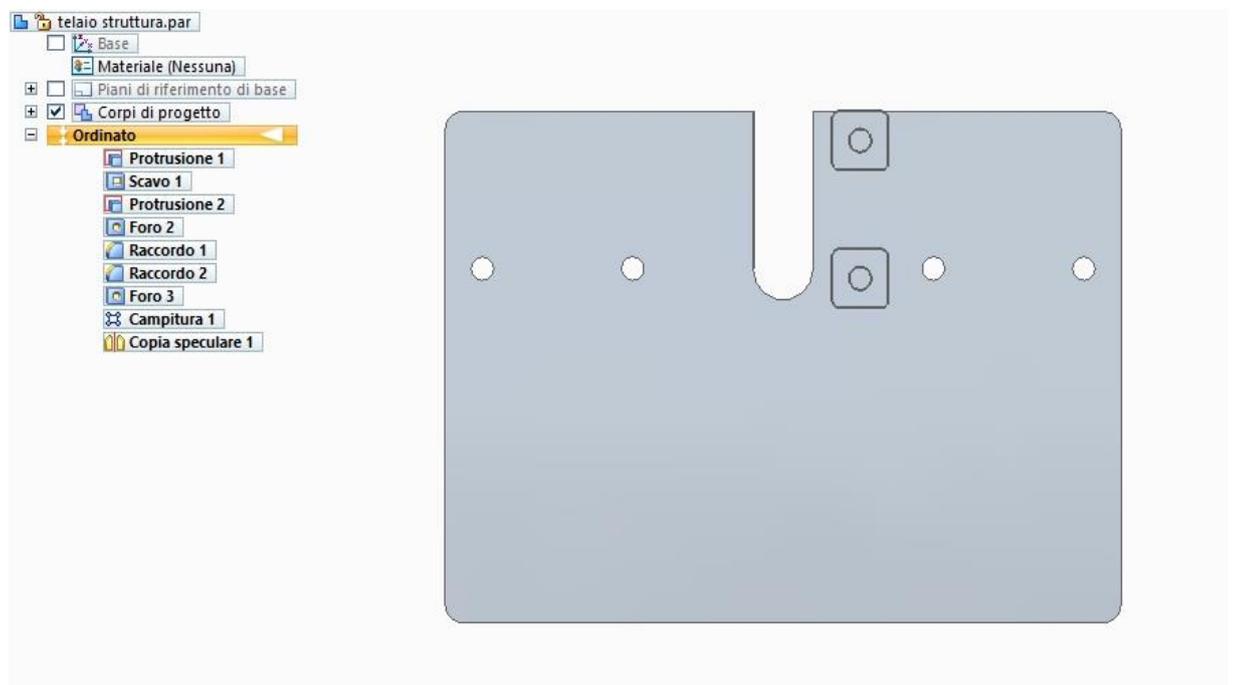


Figura 3.4. Telaio struttura

Le sue dimensioni sono piccole e per evitare problemi di natura logistica si sistema il motorino nella parte sottostante, di conseguenza si realizza anche un supporto con la funzione di immobilizzare il motorino stesso; il sostegno si posiziona dopo aver inserito il servomotore e viene fissato al telaio attraverso due viti che vengono infilate nei medesimi fori (figura 3.5).

La scheda, invece, si può collocare sopra nella parte non occupata dagli altri componenti della struttura.

Su Solid Edge la base viene creata mediante il comando estrudi che realizza una protrusione dopo aver scelto forma e dimensioni: il profilo rettangolare è risultata la migliore scelta progettuale. Successivamente si compie uno scavo al centro, in uno dei due lati più lunghi, per non intralciare la rotazione del braccio del motorino; quest'ultimo si posiziona senza nessun gioco tra due elementi, realizzati grazie ad un'estrusione, di forma quadrata disposti parallelamente vicino allo scavo e nel cui centro viene praticato un foro per le viti introdotte alla fine dopo l'inserimento del supporto di bloccaggio.

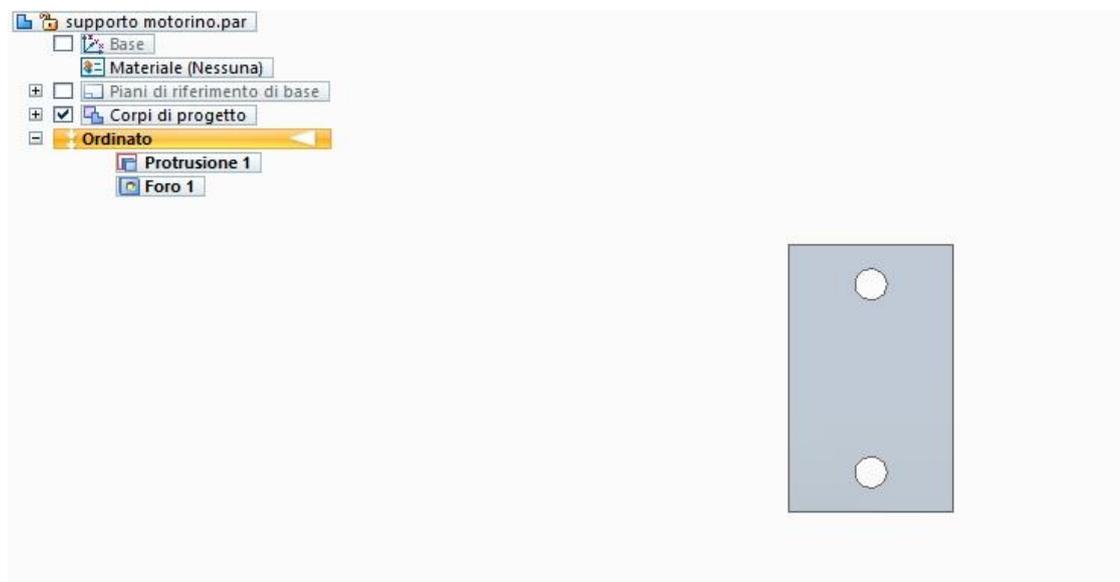


Figura 3.5. Supporto per immobilizzare il servomotore sul telaio

Inoltre nella porzione sinistra del telaio si effettua un foro che viene riprodotto a una certa distanza con il comando campitura; in seguito entrambi i fori realizzati vengono copiati specularmente nella parte destra e servono a posizionare i due appoggi che devono sostenere gli occhi. Infine per una questione per lo più di sicurezza si eseguono dei raccordi negli spigoli principali utilizzando il comando arrotonda: gli arrotondamenti sono necessari per evitare di ferirsi nel maneggiare il telaio, qualora questo fosse prodotto in metallo o altri materiali che possono tagliare.

3.2.2 Sostegno di supporto occhio

Ha la funzione di sostenere e connettere il sistema oculare al telaio e non sono necessarie complesse lavorazioni per progettarlo (figura 3.6).

Innanzitutto si crea un piedistallo, di spessore piccolo e preferibilmente di forma rettangolare tramite il comando estrudi, in cui successivamente si realizzano due fori che si sovrappongono perfettamente ai medesimi presenti nella base e servono a posizionare il supporto, bloccandolo grazie all'immissione di due bulloni.

Dal piedistallo si esegue una seconda estrusione, di forma rettangolare, verso l'alto che rappresenta il sostegno vero e proprio e in successione se ne effettua una terza circolare verso l'esterno che si connette all'occhio tramite un perno; quest'ultimo viene inserito in un foro praticato al centro dell'ultima protrusione. Infine si raccordano gli spigoli vivi più influenti con il comando arrotonda. L'altro sostegno di supporto viene realizzato allo stesso modo.



Figura 3.6. Sostegno di supporto

3.2.3 Occhi

I due occhi vengono realizzati per posizionare nel migliore dei modi le palpebre e anche per avere un sistema più completo e veritiero possibile.

La progettazione di un occhio è veramente complicata, tuttavia esso non ha una vera e propria funzionalità nel prototipo di prova e dunque si riduce tutto alla creazione di una sfera: utilizzando il comando rivoluzione si esegue una rotazione di 360° dopo aver scelto l'asse intorno cui lo schizzo deve girare.

Per esprimere maggiore somiglianza tra la sfera e l'occhio si può tentare di replicare l'iride e la pupilla. Nel nostro progetto i due vengono realizzati in maniera semplificata attraverso due protrusioni effettuate a una specifica distanza nella parte centrale della sfera; in questo modo l'iride è fisso e non varia la posizione.

Invece nella porzione opposta si esegue uno scavo circolare, fino al centro della sfera, in cui si introduce un perno per fissare l'occhio al sostegno di supporto (figura 3.7).



Figura 3.7. Occhio

3.2.4 Palpebre

Sono posizionate sopra gli occhi e devono potersi muovere in modo controllato, dunque sono connesse tra di loro da un perno di collegamento che si congiunge a sua volta con il motorino tramite un'asta. Una palpebra si realizza utilizzando ancora il comando rivoluzione: non si effettua una rotazione completa, bensì una rivoluzione di 120° dato che non si deve intersecare con il perno impiegato per sostenere l'occhio e soprattutto deve consentire il movimento, di chiusura o di apertura, di un angolo pari a 90°.

La palpebra deve possedere un diametro maggiore rispetto all'occhio; inoltre nelle parti laterali si crea una protrusione forata e in una delle due, dipende quale lato si prende in considerazione, si realizza un accoppiamento senza gioco con un perno che si deve connettere all'altra palpebra con il medesimo collegamento (figura 3.8).

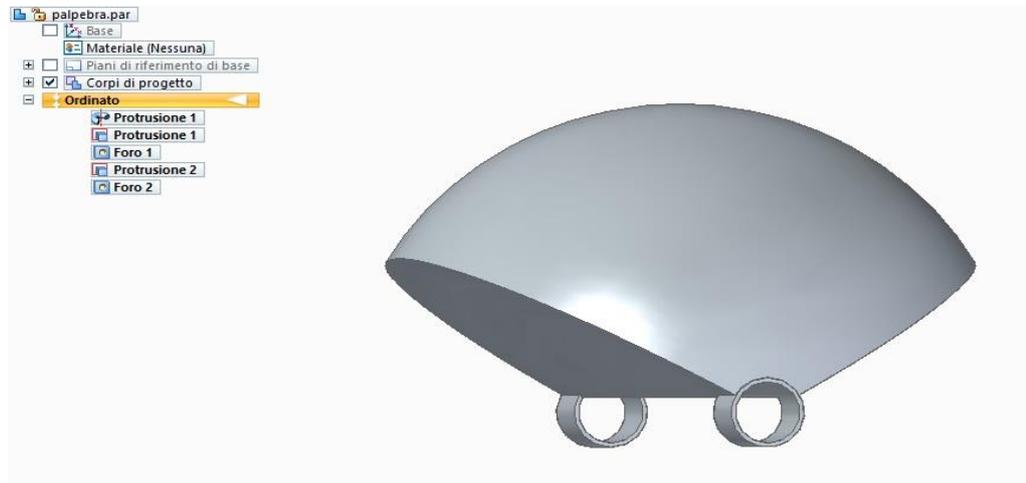


Figura 3.8. Palpebra

3.3 Assemblaggio componenti

Dopo aver progettato tutti gli elementi si passa, sempre su Solid Edge, alla modalità assieme in cui si compone la struttura. Questa parte risulta particolarmente impegnativa poiché per assemblare i vari componenti in modo corretto occorre vincolarli tra di loro seguendo delle relazioni ben precise; pertanto si possono utilizzare le esercitazioni guidate per aiutarsi.

Il primo pezzo disposto è il telaio che viene vincolato a terra. Successivamente si sistemano, uno alla volta, i due supporti per il sistema oculare: il sostegno viene accoppiato con il telaio selezionando la faccia inferiore del piedistallo e la faccia superiore del telaio stesso; poi si accoppiano assialmente i due fori con quelli della base, bloccando la rotazione, così da avere il posizionamento desiderato.

Per semplificare l'assemblaggio dei componenti, il sistema oculare viene montato separatamente in un altro documento assieme e in seguito aggiunto nel complesso principale. Nel nuovo foglio di lavoro si vincola a terra l'occhio; si inserisce e connette il collegamento tra il supporto e l'occhio sia in modo assiale sia in modo fisico:

quest'ultimo avviene unendo una delle estremità del perno con la faccia dello scavo praticato nella sfera; infine si posiziona la palpebra sopra l'occhio accoppiando il centro dello scavo con il centro di una delle due protrusioni forate e indicando la distanza (figura 3.9).



Figura 3.9. Sistema oculare assemblato

Una volta montato il sistema oculare è semplice connetterlo al supporto: si prende l'altra estremità libera del perno di collegamento e si effettua un accoppiamento, anche assiale, con il foro realizzato nel sostegno. Si ripete lo stesso procedimento anche con l'altro sistema oculare; quindi si congiungono le due palpebre tramite un altro perno. Quest'ultimo viene accoppiato assialmente alla protrusione forata conforme delle due palpebre con cui realizza un accoppiamento senza gioco e si posiziona in maniera tale da non toccare l'occhio, inoltre la sua rotazione rimane sbloccata poiché deve trasmettere il movimento del motorino.

L'ultimo componente necessario a completare il prototipo è l'asta di collegamento tra il servomotore e il perno che connette le palpebre; per facilitare il suo posizionamento si progetta un motorino semplificato con identiche dimensioni a quello reale e si colloca nella zona del telaio stabilita. In seguito si dispone l'asta che può essere saldata al braccio del servomotore nella parte inferiore, mentre nella porzione superiore si accoppia l'asse del foro creato nell'asta stessa con l'asse del foro praticato in una piccola diramazione a 90° realizzata, in prossimità del centro, nel perno.

In figura 3.10 si può osservare la struttura finale dopo che i componenti sono stati assemblati, è stato inserito anche il supporto per bloccare il motorino.

Naturalmente per permettere al banco mecatronico di funzionare si sostituisce il motorino utilizzato per posizionare l’asta con il servomotore.

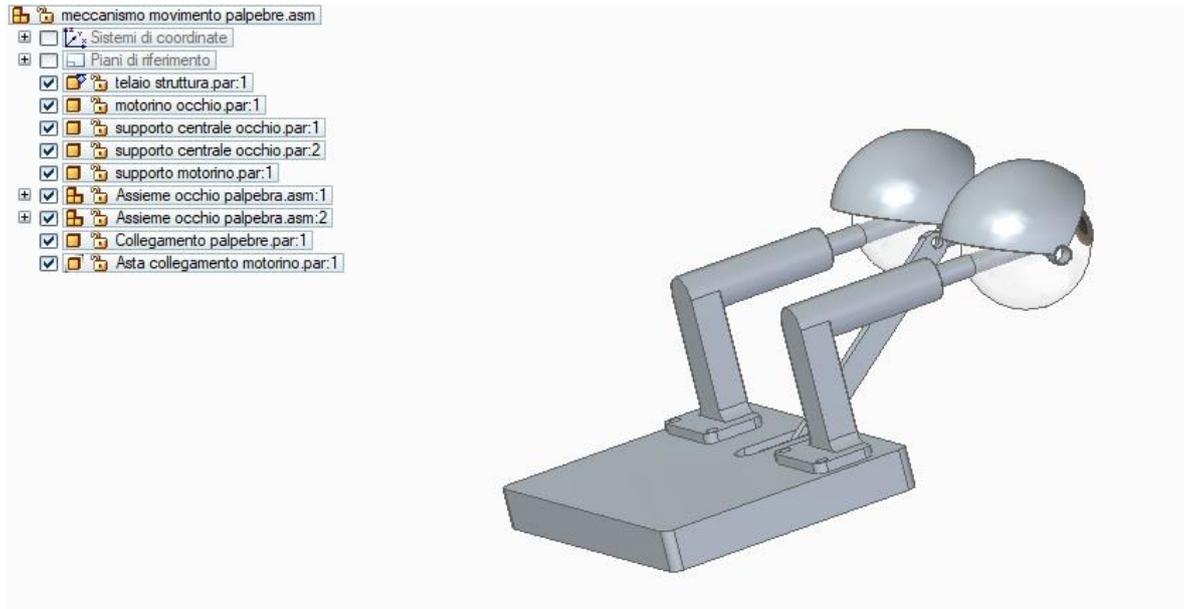


Figura 3.10. Struttura completa finalizzata alla movimentazione controllata delle palpebre

3.4 Analisi di posizione

Il sistema: braccio posizionato all’altezza del telaio - asta di collegamento – perno di connessione delle palpebre – sostegno occhio è possibile associarlo a un quadrilatero articolato, dunque si effettua un’analisi di posizione per ricavare una soluzione realizzabile che risponda alle specifiche richieste.

In un foglio Excel si riportano i valori dei lati costituenti il quadrilatero articolato e indicando con O il punto di partenza situato nel telaio si scrive l’equazione di chiusura (3.1).

$$z_2 + z_3 + z_4 - z_1 = 0 \quad (3.1)$$

L’Eq. (3.1) si proietta sugli assi del riferimento assoluto così da poter visualizzare anche gli angoli e si ottengono le equazioni (3.2)

$$\left\{ \begin{array}{l} z_2 \cos (\theta_2) + z_3 \cos (\theta_3) + z_4 \cos (\theta_4) - z_1 \cos (\theta_1) = 0 \\ z_2 \sin (\theta_2) + z_3 \sin (\theta_3) + z_4 \sin (\theta_4) - z_1 \sin (\theta_1) = 0 \end{array} \right. \quad (3.2)$$

in cui l'angolo θ_1 è nullo poiché z_1 coincide con l'asse x del riferimento assoluto e i moduli dei lati sono costanti. Quindi si assume nota la coordinata libera θ_2 che rappresenta l'angolo del movente e in una colonna si indicano tutti i valori che può assumere, ossia un angolo giro: da 0° a 360° ; questi valori vengono espressi anche in radianti per comodità.

Per una rappresentazione completa del meccanismo si mostrano in altre colonne le parti note c_x e c_y calcolate grazie alle equazioni (3.3) e (3.4) e i coefficienti A, B, C i quali dovrebbero indicare le posizioni dei vertici del quadrilatero e sono ricavati dalle equazioni (3.5)

$$c_x = z_2 \cos (\theta_2) - z_1 \quad (3.3)$$

$$c_y = z_2 \sin (\theta_2) \quad (3.4)$$

$$A = 2c_x z_3 \quad B = 2c_y z_3 \quad C = (c_x^2 + c_y^2 + z_3^2 - z_4^2) \quad (3.5)$$

Dopodiché si calcolano i valori di θ_3 che corrispondono alla configurazione in cui il punto B rimane sopra alla congiungente AC e permettono di ricavare θ_4 tramite l'equazione (3.6), costituita dalla funzione arcotangente nella sua versione con due argomenti.

$$\theta_4 = \arctan 2 (- (c_y + z_3 \sin (\theta_3)), - (c_x + z_3 \cos (\theta_3))) \quad (3.6)$$

L'angolo del cedente, rappresentato da θ_4 , deve poter effettuare una rotazione almeno di 90° e deve rilevare sempre una misura ogni volta che il corpo movente varia. Se viene meno quest'ultima constatazione, quindi risulta impossibile trovare uno o più valori di θ_4 oppure si segnalano uno o più errori nel suo calcolo, significa che il dispositivo non può assumere determinate posizioni e dunque non può essere montato in tutti gli angoli spazzati dal movente. Di conseguenza si modificano le grandezze di posizione, ovvero i lati del quadrilatero, per arrivare a una soluzione realizzabile in grado di assecondare i dati richiesti.

Capitolo 4

Risultati e Conclusioni

4.1 Configurazione dispositivo

Applicando un approccio alla Monte Carlo per la modifica delle grandezze di posizione del quadrilatero articolato sono state individuate le misure ottimali, visibili nella tabella 4.1, per la realizzazione di un dispositivo in grado di produrre il movimento richiesto.

z_2	OA	18	Mm
z_3	AB	138	Mm
z_4	BC	29	mm
z_1	OC	129	mm

Tabella 4.1. Lunghezza finale dei lati del quadrilatero

Nel grafico 4.1 è possibile notare la configurazione con cui la struttura si dispone dopo che sono state assegnate tali misure.

Si nota che il braccio posizionato sul motorino è molto corto rispetto all'asta di collegamento e rappresenta il movente del dispositivo, dunque ha il compito di azionare l'intero sistema. L'asta riceve l'impulso e deve trasmettere il movimento al perno che,

quindi, permette la rotazione della palpebra, chiudendola o aprendola. Inoltre si può osservare che il perno non è perfettamente in asse con l'asta e ha una lunghezza modesta. In aggiunta nel grafico non viene raffigurato il segmento z_1 , ovvero il sostegno di supporto collegato al telaio, che, però, è ben visibile poiché posizionato sull'asse orizzontale: in questo modo si ha la chiusura del quadrilatero.

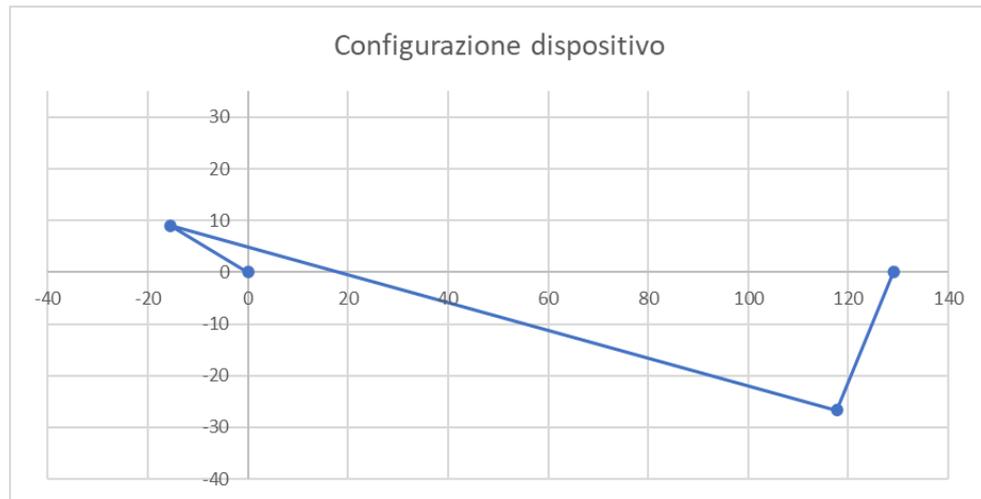


Grafico 4.1. Configurazione finale della struttura

La struttura realizzata con le suddette misure risponde perfettamente alle esigenze di progetto come si può rilevare nel grafico 4.2. Infatti il cedente descrive un angolo maggiore di 90° , per la precisione 90.33° : da 156.01° a 65.67° ; avendo un movimento più rapido quando l'angolo si aggira intorno a 156.36° dopo il quale il meccanismo torna alla configurazione di partenza.

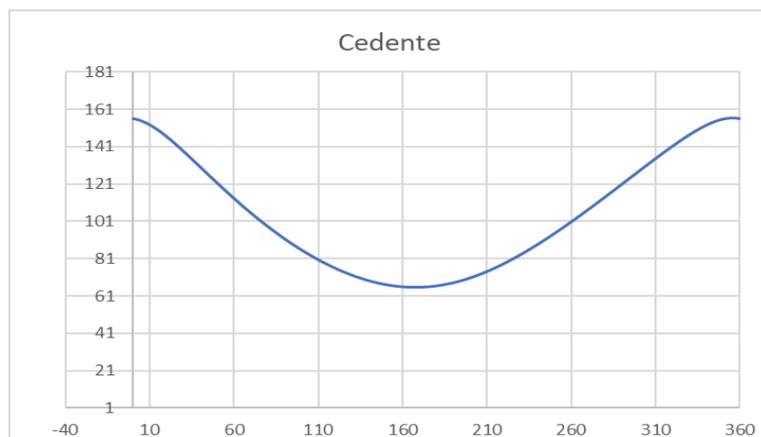


Grafico 4.2. Angolo di oscillazione del cedente

4.2 Conclusioni

L'obiettivo della tesi è stato quello di analizzare l'espressività nel volto di un robot umanoide, proponendo un piccolo banco meccatronico di prova finalizzato a muovere in maniera controllata le palpebre di un occhio robotico tramite Arduino. Tuttavia a causa di alcune problematiche dovute a fattori esterni il prototipo è stato ideato, ma non realizzato; di conseguenza non è stato possibile validare il suo funzionamento sperimentalmente. Comunque sia il dispositivo con la configurazione ricavata dai risultati è stato esaminato sul software Solid Edge inserendo il motorino semplificato di uguali dimensioni al posto di quello reale e si è potuto constatare che esegue il movimento richiesto, quindi l'obiettivo è stato raggiunto.

La struttura, essendo un banco di prova, è molto semplice e può essere utilizzata, ad esempio, nella rappresentazione teatrale dopo che viene inserita nella testa di un bambolotto oppure di un robot umanoide i quali saranno in grado di aprire o chiudere gli occhi. Inoltre è bene ricordarsi che il controllo del movimento delle palpebre avviene tramite la scheda Arduino, alimentata da un computer o da un'alimentazione esterna costituita da batterie che dunque devono essere posti nelle vicinanze. Le difficoltà sono state veramente minime e la maggior parte dovute alla mancanza di esperienza nell'utilizzo della strumentazione. Il progetto può essere sviluppato rendendo i componenti del sistema più complessi e realistici, ad esempio intervenendo sugli occhi che, tuttavia, sono estremamente complicati da realizzare in modo simile a quelli degli esseri umani anche utilizzando software molto avanzati. Altro metodo di sviluppo è

quello di aggiungere un maggior numero di servomotori per un miglior controllo e per aumentare le funzionalità: per esempio impiegando due motorini, uno per ogni occhio, è possibile separare il meccanismo delle palpebre e di conseguenza i due occhi possono muoversi in maniera indipendente. Infine questa tesi si può riprendere per progettare altri componenti del corpo umano che, però, hanno un funzionamento e una progettazione diversi.

Bibliografia

- [CFP17] M. Callegari; P. Fanghella; F. Pellicano. 2017. *Meccanica applicate alle macchine*, Seconda edizione. CittàStudi, Novara.
- [ELLSJ] E. A. N. S. Edirisinghe; R. A. P. Lakmal; T. G. G. Lakshitha; S. W. D. T. S. Sirimanna; A. G. Buddhika; P. Jayasekara. 2016. *Design of a Human-like Robot Head with Emotions*. Robotics and Control Laboratory Department of Electrical Engineering University of Moratuwa. Moratuwa, Sri Lanka
- [PrT12] S. Priya; Y. Tadesse. 2012. *Graphical Facial Expression Analysis and Design Method: An Approach to Determine Humanoid Skin Deformation*. Department of Mechanical Engineering, Virginia Tech, Blacksburg.
- [Add12] F. Adăscăliței; I. Doroftei. 2012. *Expressing Emotions in Social Robotics - A Schematic Overview Concerning the Mechatronics Aspects and Design Concepts*. Technical University of Iasi, Bucarest
- [HST98] S. Hirano; K. Sato; A. Takanishi. 1998. *Development of an anthropomorphic Head-Eye System for a Humanoid Robot: Realization of Human-like Head-Eye Motion Using Eyelids Adjusting to Brightness*. Waseda University, Giappone.
- [ACMMS] H. Abdollahi; R. Cole; M. H. Mahoor; A. Mollahosseini; T. D. Sweeny. 2018. *Role of embodiment and presence in human perception of robots' facial cues*. International Journal of Human-Computer Studies, Denver.
- [LMPr6] H. Lehmann; A. Roncone; U. Pattacini; G. Metta. 2016. *Physiologically Inspired Blinking Behavior for a Humanoid Robot*. Istituto italiano di Tecnologia, iCub Facility, Genova, Italia. Social Robotics Lab, New Haven, CT, USA

- [NORST] A. Nakazawa; T. Ogasawara; G. A. G. Ricardez; T. Sano; J. Takamatsu; A. Yuguchi. 2019. *Evaluating Imitation of Human Eye Contact and Blinking Behavior Using an Android for Human-like Communication*. Conference location New Delhi, India.
- [DORTY] M. Ding; T. Ogasawara; G. A. G. Ricardez; J. Takamatsu; A. Yuguchi. 2017. *Gaze Calibration for Human-Android Eye Contact Using a Single Camera*. Graduate School of Information Science, Nara Institute of Science and Technology, Giappone.