Università Di Bologna

Fisica dei sistemi complessi

Relazione Progetto

Prototipo di catapulta automatizzata con Arduino

Author:	*Student Number:*
Calvigioni Giacomo	0000784488
Contigiani Riccardo	0000784451
Pagliari Alessio	0000784408
Traini Stefano	0000778487

Luglio 2016

Indice

1 Introduzione

La catapulta automatizzata è un dispositivo a trazione gestito da un micro-controllore Arduino in grado di colpire un bersaglio posto frontalmente ad una distanza arbitraria. Il progetto prevede il connubio tra le discipline della fisica e dell'informatica al fine di estrapolare un algoritmo che risolva la classe dei problemi del calcolo della trazione data in input la gittata. Il sistema prevede che il lancio del proiettile venga effettuato per mezzo di un braccio propellente messo in movimento dalla elongazione di una molla e bloccato da un'asta che definisce quindi l'angolo di lancio. I componenti descritti sono messi in movimento dal micro-controllore tramite un set di tre servo motori, il primo effettua la trazione della molla, il secondo inclina l'asta che blocca il braccio mentre il terzo si occupa del blocco/sgancio di quest'ultimo; la rilevazione del bersaglio viene effettuata tramite un sensore ad ultrasuoni anch'esso collegato al micro-controllore.

2 Fase 1 - Costruzione della catapulta

Il sistema è stato assemblato quasi interamente utilizzando legno di abete fatta eccezione per le componenti elettroniche e l'asta di arresto del braccio propellente in metallo. Lo scheletro della catapulta escludendo le parti elettriche ed i relativi supporti è formato da una base, 4 parallelepipedi componenti la struttura che inclina e sorregge l'asta di arresto, l'asta stessa, il braccio propellente, una cerniera metallica ed il cesto per l'alloggiamento del proiettile.
Le parti elencate sono state assemblate utilizzando viti autofilettanti secondo lo schema in figura:

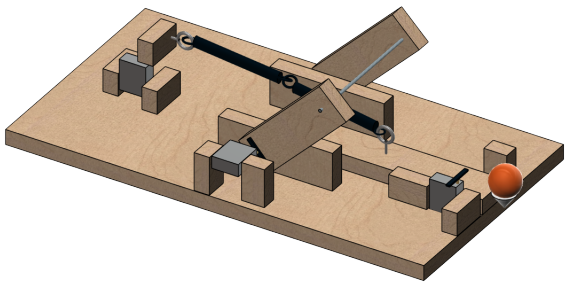


Figure 1: Modello 3D del prototipo

2.1 Componenti Meccaniche

Di seguito saranno descritti in dettaglio e mostrati con una figura i singoli pezzi elencati in precedenza:

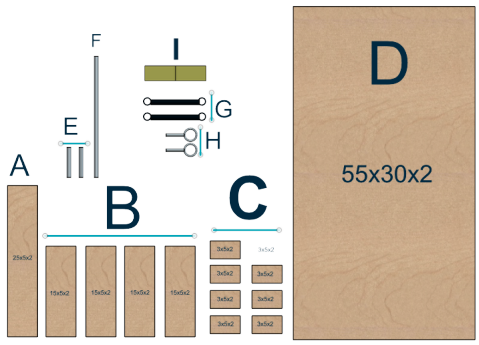


Figure 2: Modello dei componenti meccanici del prototipo

Componente **A** Braccio propellente attaccato alla base tramite la cerniera (componente I)

Componente **B** componenti della struttura di arresto del braccio; connessi a due a due tramite aste metalliche (componente E), determinano l'angolazione del braccio di arresto (componente G)

Componente **C** supporti dei servo motori e giunzione tra il servo motore e il perno metallico (componente H)

Componente **D** base della catapulta

Componente **E** aste metalliche di giunzione degli elementi della struttura di arresto (componente B), misurano 5 cm

Componente **F** asta metallica di arresto del braccio (20 cm)

Componente **G** molle a trazione collegate tra loro e alle estremità ai perni (componente H); la parte elastica di ciascuna molla misura 8 cm

Componente **H** perni di giunzione tra il braccio propellente e la molla e tra il servo motore traente e la molla

Componente **I** cerniera che collega la base della catapulta al bracci propellente

2.2 Componenti Elettroniche

Le componenti elettroniche invece sono composte da:

- Arduino UNO R3

Figure 3: Microcontrollore Arduino REV 3

- 3x servo motori [TowerPro, MG995]

Figure 4: Servo motore Tower Pro MG-995

- Sensore di distanza [Vorcool, HC-SR04]

Figure 5: Sensore di distanza Vorcool HC-SR04

3 Fase 2 - Test Preliminari

I primi test sono stati effettuati per verificare il corretto funzionamento delle componenti elettroniche.

3.1 Test dei servomotori

I servo motori sono stati montati e testati singolarmente per verificarne le funzionalità e il range d'azione.
Nell'ordine è stato montato dapprima il servo che regola l'angolo di lancio, poi quello che regola lo sblocco del braccio ed infine quello che gestisce l'allungamento della molla. Terminati i test singoli è stato scritto il seguente sketch al fine di testare i 3 componenti all'unisono.

```
1  #include <Servo.h>
2
3  Servo gancio;
4  Servo inclinazione;
5  Servo molla;
6
7  void setup() {
8    Serial.begin(9600);
9    gancio.attach(2);
10   inclinazione.attach(4);
11   molla.attach(6);
12   delay(1000);
13 }
14
15 void loop() {
16 gancio.write(100);
17 molla.write(100);
18 inclinazione.write(100);
19 delay(2000);
20 gancio.write(0);
21 molla.write(0);
22 inclinazione.write(0);
23 }
```

Alimentazione dei servo motori

Da questo secondo test è stato riscontrato un problema di alimentazione, la tensione in uscita dell'Arduino (5 Volt) non è sufficiente a far operare tutti i servo contemporaneamente. Tramite il data sheet dei componenti utilizzati è stato verificato che ogni motore richiede un voltaggio tra 4,5 e 6 Volt per un totale tra 13,5 e 18 Volt utilizzando 3 componenti.

Soluzione: E' stata aggiunta una batteria da 9 Volt con lo scopo di alimentare 2 dei 3 servo mentre il terzo resta alimentato dall'Arduino.

3.2 Primi test di lancio

Successivamente sono stati effettuati i primi lanci fissando l'inclinazione dell'asta a 45° e non collegando il sensore di distanza per verificare le potenzialità della catapulta. Da questi test sono sorte varie problematiche:

1. La cerniera che regola il movimento del braccio di lancio ha del gioco, compie pertanto movimenti laterali che determinano una direzione errata del proiettile.

   Soluzione: E' stato inserito un ulteriore elemento di legno fissato alla base all'altezza del servo che regola lo sgancio del braccio al fine di ridurre l'oscillazione di quest'ultimo.

2. 2. L'urto tra il braccio di lancio e l'asta in ferro che ne arresta la corsa provoca un'usura del legno che lo compone; a lungo andare potrebbe provocare una variazione della gittata.

   Soluzione: è stata aggiunta un'imbottitura in gomma piuma che circonda la barra metallica in modo tale da attutire la violenza dell'urto.

3. Il contraccolpo prodotto dal lancio provoca una traslazione in avanti dell'intera catapulta di alcuni millimetri ad ogni lancio;

   Soluzione: sono stati applicati dei gommini sotto la base per ridurre il contraccolpo prodotto dal lancio;

4. La forza dell'urto tra il braccio e l'asta metallica produce una variazione della posizione di quest'ultima causando un'inclinazione del tiro maggiore rispetto a quella prestabilita.

Soluzione: è stata applicata della colla a caldo sui punti di giuntura tra l'asta metallica ed i componenti in legno al fine di ridurre il movimento; successivamente sono stati effettuati nuovi lanci ed è stata calcolata in modo empirico la variazione dell'angolo di lancio, circa 5°; in conseguenza di ciò è stato applicato un offset all'algoritmo di lancio al fine tener in considerazione nel calcolo la variazione di inclinazione dovuta all'urto.

5. Il proiettile ed il relativo recipiente non sono adeguati per i test.

   Soluzione: Il proiettile utilizzato inizialmente, un piccolo cubo in legno, è stato sostituito con una pallina da ping-pong e come recipiente si è utilizzato un elemento che si adatta alla forma sferica della palla.

6. Durante il lancio, nella fase di allungamento della molla, il punto di giuntura tra il servo motore tirante e la molla ha uno scivolamento dovuto alla forma circolare di entrambi gli elementi; questo causa un errore nel calcolo dell'allungamento.

   Soluzione: Il perno è stato posizionato verticalmente e fissato con della colla a caldo per annullare lo scivolamento prodotto dall'allungamento della molla.

3.3 Accuratezza del sensore di distanza

In seguito sono stati effettuati dei test per analizzare l'accuratezza e il range del sensore di distanza tramite un apposito sketch Arduino.
Test del sensore di distanza:

```
1  #define echoPin 12 // Echo Pin
2  #define trigPin 10 // Trigger Pin
3
4  void setup() {
5   Serial.begin (9600);
6   pinMode(trigPin, OUTPUT);
7   pinMode(echoPin, INPUT);
8  }
9
10 void loop() {
11  digitalWrite(trigPin, LOW);
12  delayMicroseconds(2);
13  digitalWrite(trigPin, HIGH);
14  delayMicroseconds(10);
15  digitalWrite(trigPin, LOW);
```

```
16   duration = pulseIn(echoPin, HIGH);
17   distance = duration/58.2;
18   Serial.print(distance);
19   println
20   delay(50);
21 }
```

Il test sul sensore di distanza ha portato alla luce delle imprecisioni sulla distanza percepita dall'hardware del sensore.

Soluzione: La precisione è stata migliorata effettuando più misurazioni e calcolando la media delle stesse. Inoltre è stato necessario aggiungere un offset alla distanza rilevata dal sensore essendo i valori in difetto di circa 10cm.

4 Fase 3 - Test Empirici

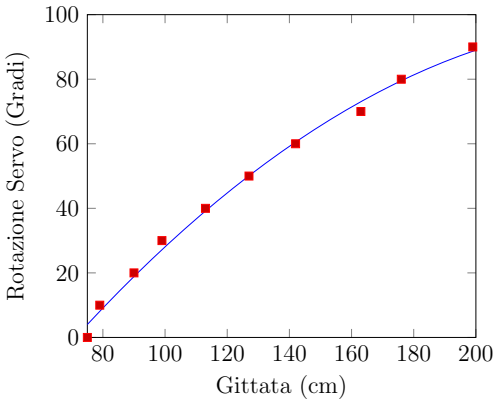
4.1 Test per distanza con angolazione a 45°

Dopo aver applicato queste modifiche al sistema sono stati effettuati ulteriori test impostando l'inclinazione del braccio di lancio a 45° e variando l'allungamento della molla di 10 step del servo motore ad ogni serie di prove per verificare la gittata del lancio. Tramite una fettuccia posta a terra sono state effettuate 10 misurazioni per ciascun allungamento riprendendo la prova con una videocamera a 60 fotogrammi al secondo e riproducendo il video al rallentatore.
I dati della gittata [**Tabella 1**] sono stati inseriti su un foglio di calcolo ed è stata tracciata la retta interpolante per trovare la relazione che lega l'allungamento della molla tirante e la gittata del lancio.

Table 1: Tabella valori risultati da test con inclinazione a 45°

Allungamento Molla (cm)	Rotazione Motore (Gradi)	Gittata (cm)
5.9	0	75.4
6.1	10	78.7
6.6	20	90.2
6.9	30	99.4
7.7	40	113.2
8.7	50	127.3
9.3	60	141.5
10.3	70	163
11.2	80	176.4
11.9	90	199.4

Il grafico della funzione interpolante e dei punti che la hanno determinata è il seguente:



Essendo descrivibile con buona approssimazione attraverso l'equazione di una parabola è possibile descrivere direttamente la relazione che lega la rotazione del motore alla

gittata.
E' stata quindi ricavata la funzione di interpolazione che data in input la gittata determina di quanto è necessario ruotare il servo motore; l'equazione della parabola è :

$$y = -0.002822x^2 + 1.456x - 89.314 \quad (1)$$

4.2 Test per distanza con angolazione a 30°

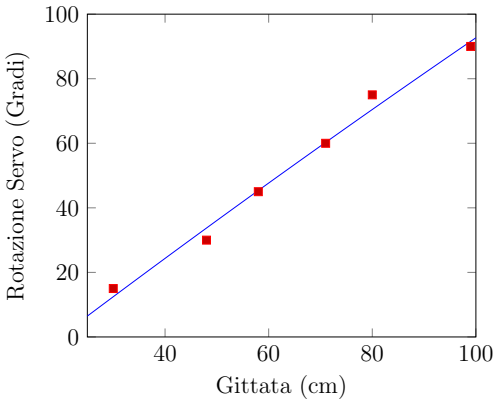
Successivamente sono state realizzati gli stessi test impostando il braccio di arresto a un'angolazione di poco superiore ai 30 gradi al fine di ampliare il range della catapulta e far si che possa colpire bersagli posti ad una distanza più ravvicinata; infatti con un angolo di 45° la gittata è massima ed è pertanto impossibile centrare obiettivi inferiori agli 80 cm. Di seguito la tabella [**Tabella 2**] e la parabola interpolante che fanno riferimento ai lanci effettuati col braccio a 30°, l'equazione della parabola ricavata è:

$$y = -0,0007001 * x^2 + 1,237 * x - 24,02 \quad (2)$$

Table 2: Tabella valori risultati da test con inclinazione a 30°

Allungamento Molla (cm)	Rotazione Motore (Gradi)	Gittata (cm)
TODO	15	30.4
6.9	30	48.3
TODO	45	58.2
9.3	60	70.7
TODO	75	80.1
11.9	90	99.4

Il grafico della funzione interpolante e dei punti che la hanno determinata è il seguente:



5 Fase 4 - Test con sensore di distanza

Nella fase successiva sono state inserite nello sketch arduino le leggi ricavate ed è stato testato il sistema nella sua interezza.
Da questo test sono emerse nuove problematiche, il sensore richiede ulteriori 5 Volt per essere alimentato che vanno a sommarsi a quelli relativi ai servo motori.
Soluzione: E' stato aggiunto un alimentatore regolabile che fornisce fino a 12 Volt di tensione; la configurazione finale prevede che il sensore di distanza sia alimentato direttamente dal micro-controllore mentre i 3 servo motori dall'alimentatore regolabile.

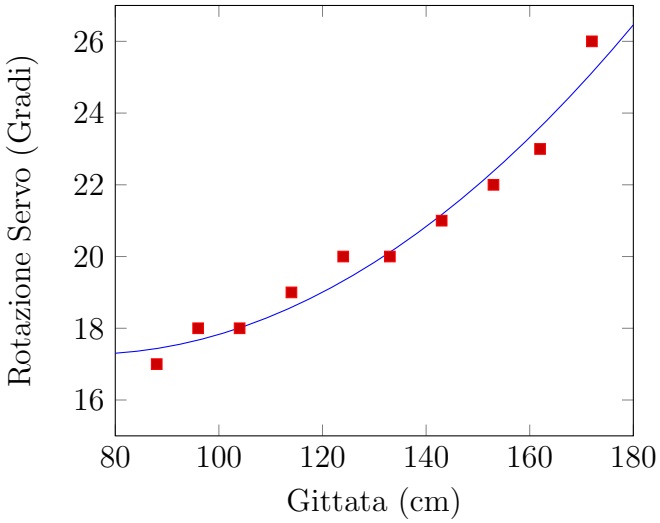
Inoltre è stato riscontrato un errore crescente al crescere della distanza del bersaglio sia della valutazione da parte del sensore che del sistema di lancio stesso.
Gli errori commessi sono stati analizzati separatamente per i lanci a 30° e a 45° al fine di trovare una funzione che descriva l'offset da utilizzare.
Per i lanci a 30° è stato sufficiente inserire 3 condizioni all'interno dei controlli dell'algoritmo che fanno variare l'offset di lancio. Per i lanci a 45° che coprono una range più ampio è stata approssimata ed inserita una funzione di calcolo dell'offset tramite un'equazione di secondo grado, utilizzando i test descritti in precedenza.

$$y = 0,0008175x^2 - 0,121x + 21,752 \tag{3}$$

Di seguito la tabella ed il grafico della parabola di interpolazione per i lanci a 45°.

Table 3: Tabella risultati ed offset da test con inclinazione a 45°

Distanza Obiettivo(cm)	**Offset (Gradi)**	**Distanza Rilevata(cm)**
90	17	88
100	18	96
110	18	104
120	19	114
130	20	124
140	20	133
150	21	143
160	22	153
170	23	162
180	26	172



La seguente è la tabella contenente i dati inseriti per l'aggiustamento del tiro dei i lanci a 30°.

Table 4: Tabella valori risultati ed offset da test con inclinazione a 30°

Distanza Obiettivo(cm)	**Offset (Gradi)**	**Distanza Rilevata(cm)**
30	5	76
40	3	67
50	3	57
60	3	47
70	3	38
80	1	28

6 Codice Sketch Arduino

```
#include <Servo.h>
#include <math.h>

/*
   DEFINIZIONE VARIABILI DISTANCE SENSOR HC-SR04
*/

#define echoPin 12 // Echo Pin
#define trigPin 10 // Trigger Pin

/*
  DEFINIZIONE PIN COLLEGAMENTI CON ARDUINO
*/
#define gancioPin 2
#define incliPin  4
#define mollaPin  6
#define butPin    8

/*
   DEFINIZIONE VARIABILI ED OFFSET OPERATIVI
*/
#define AGGANCIA    85 //Servo agganciato
#define SGANCIA     0 //Servo sganciato
#define OFFSETMOLLA 30 //Per tirare al minimo la molla
int OFFSETDIST; //Per colpiere in pieno l'obiettivo

/*
   DEFINIZIONE ANGOLI DI LANCIO E DISTANZE
*/
#define MINGIT          30
#define MAXGIT          180
#define CAMBIOGIT       85
#define TRENTA          150
#define QUARANTACINQUE  135

/*
   VARIABILI DI STATO BOTTONE
*/

int initialSet = 1;//Stato iniziale bottone
int stat = 0;//Stato del bottone
```

```
42 int val = 0;//Lettura stato del bottone
43
44 /*
45    DEFINIZIONE DEI SERVO
46 */
47 Servo gancio;//dichiarazione servo per gancio/sgancio
48 Servo inclinazione;//dichiarazione servo inclinazione asta
49 Servo molla;//dichiarazione servo che tira la molla
50
51 /*
52    SETUP DEL SISTEMA
53 */
54 void setup() {
55   gancio.attach(gancioPin);  // servo
56   delay(500);
57   inclinazione.attach(incliPin);
58   delay(500);
59   molla.attach(mollaPin);
60
61   pinMode(butPin, INPUT);
62   pinMode(trigPin, OUTPUT);
63   pinMode(echoPin, INPUT);
64
65   Serial.begin(9600);
66   delay(500);
67 }
68
69 /*
70    DEFINIZIONE DEL LOOP DI ARDUINO
71 */
72 void loop() {
73   long dist, all;
74   //Stato iniziale del sistema
75   if (initialSet == 1) {
76     relax();
77     initialSet = 0;
78   }
79
80   //leggo valore bottone
81   val = digitalRead(butPin);
82
83   //controllo bottone
84   if (val == LOW) {         // check if the input is HIGH (button released)
```

```
85
86      if (stat == 1) {
87        relax();
88        stat = 0;
89        delay(1000);
90      }
91      else {
92        //Serial.println("AGGANCIA");
93        gancio.write(AGGANCIA);
94
95        /* ----------CALCOLA COSE------------ */
96
97        dist = distanza();
98        Serial.print("Media Dist: ");
99        Serial.println(dist);
100
101       if (dist >= MINGIT  dist <= MAXGIT) {
102         if (dist <= CAMBIOGIT) {
103           inclinazione.write(TRENTA); //TODO inclinazione 30 gradi
104           OFFSETDIST = offset(dist, TRENTA);
105           all = allungamento(dist + OFFSETDIST, TRENTA);
106           Serial.println("LANCIO A 30 GRADI ");
107         } else {
108           inclinazione.write(QUARANTACINQUE); //TODO inclinazione 45 gradi
109           OFFSETDIST = offset(dist, QUARANTACINQUE);
110           all = allungamento(dist + OFFSET, QUARANTACINQUE);
111           Serial.print("OFFSET: ");
112           Serial.println(OFFSETDIST);
113           Serial.println("LANCIO A 45 GRADI");
114         };
115
116         Serial.print("CARICA MOLLA A ");
117         Serial.println(all);
118         molla.write(OFFSETMOLLA + all);
119         stat = 1;
120       }
121       else Serial.println("ERRORE: rilevazione errori fuori dal range");
122       delay(1000);
123     }
124   }
125 }
126
127 /*
```

```
128     DEFINIZIONE FUNZIONE PER RILASSARE IL SISTEMA
129 */
130 void relax() {
131   //inclinazione.write(135);//inclinazione a 45 gradi
132 
133   Serial.println("SGANCIA");
134   gancio.write(SGANCIA);
135   delay(1500);
136 
137   Serial.println("RIALASSA MOLLA");
138   molla.write(OFFSETMOLLA);
139   delay(1000);
140 }
141 
142 /*
143     FUNZIONE PER CALCOLARE OFFSET DI LANCIO
144 */
145 int offset(long d, int ang){
146   int os; //offset
147   double a, b, c, err;
148 
149   switch(ang){
150     case QUARANTACINQUE:
151       /*calcolo a 45 gradi*/
152       a = 0.0008175;
153       b = -0,121;
154       c = 21,752;
155       err = a * d * d + b * d + c;
156       os = round(err);
157       break;
158     case TRENTA:
159       /*calcolo a 30 gradi*/
160       if(d>75) os = 5;
161       else if(d<=75  d>=40) os = 3;
162       else os = 1;
163     break;
164   }
165 
166   return os;
167 }
168 
169 /*
170     CALCOLO ALLUNGAMENTO DELLA MOLLA
```

```
171 */
172 long allungamento(long x, int ang) {
173   long mot;
174   double a, b, c, g;
175 
176   Serial.print("Fin Dist: ");
177   Serial.println(x);
178 
179   switch(ang){
180     case QUARANTACINQUE:
181       /*calcolo a 45 gradi*/
182       a = -0.002822;
183       b = 1.456;
184       c = -89.314;
185       break;
186     case TRENTA:
187       /*calcolo a 30 gradi*/
188       a = -0.0007001;
189       b = 1.237;
190       c = -24.02;
191     break;
192   }
193 
194   g = a * x * x + b * x + c;
195 
196   mot = round(g);
197 
198   return mot;
199 }
200 
201 /*
202    CALCOLO VALORI SENSORE DISTANZA
203 */
204 long distanza() {
205 
206   long duration, distance;
207   long somma, media;
208   long nt = 5;
209   long nd = 0;
210   somma = 0;
211 
212   //rilvemento con sensore e media
213   for (int i = 0; i < nt; i++) {
```

```
214       digitalWrite(trigPin, LOW);
215       delayMicroseconds(2);
216       digitalWrite(trigPin, HIGH);
217       delayMicroseconds(10);
218       digitalWrite(trigPin, LOW);
219       duration = pulseIn(echoPin, HIGH);
220       distance = duration / 58.2;
221
222       somma += distance;
223       Serial.print("Dist  ");
224       Serial.print(": ");
225       Serial.println(distance);
226     }
227
228     media = somma / nt;
229     return media;
230   }
```

7 Problematiche e guasti

Durante l'assemblaggio del sistema e lo svolgimento dei test si sono verificato il guasto di un servo motore nella seguente modalità:
Lo svolgimento di test consecutivi ha causato una distorsione del perno interno al meccanismo del servo motore che gestisce l'inclinazione della barra matellica di arresto del braccio propellente. Effettuando lanci con una torsione del servo motore superiore ai 90 step l'urto prodotto distorce il perno. La distorsione rende il componente inutilizzabile.

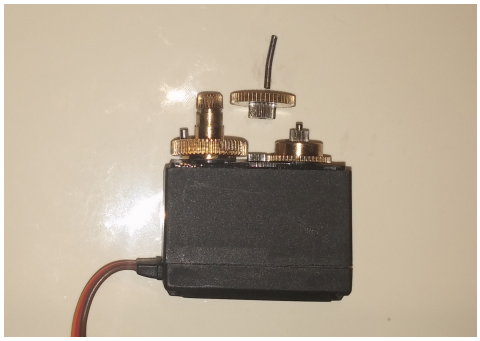


Figure 6: Servo motore con il perno piegato

Soluzione: Inizialmente si è tentato di raddrizzare e sostituire il perno con un pezzo metallico compatibile; entrambe le soluzioni sono risultate inefficaci in quanto il perno raddrizzato tende ad essere più facilmente soggetto a distorsione essendo diventato più malleabile dopo una prima deformazione mentre la sostituzione con un perno nuovo non è stata possibile in quanto il componente ha una tolleranza di dimensioni molto bassa e non sono presenti specifiche abbastanza dettagliate a riprodurlo in maniera idonea. E' stato per tanto necessario sostituire l'intero servo motore con uno nuovo limitare la rotazione del motore di trazione abbassando di fatto il range del sistema a 180 cm.

L'esecuzione ripetuta dei test ha inoltre causato una variazione sulla gittata non imputabile ad una causa specifica. Sono state fatte diverse ipotesi sulle cause della variazione: il mutamento potrebbe essere dovuto al gioco della cerniera che si è accentuato nel corso delle prove, all'usura delle molle o all'usura dei servo motori.

8 Conclusioni

In conclusione l'obiettivo posto dal progetto è stato raggiunto, è stata realizzata una catapulta a trazione che colpisce un bersaglio posto ad una distanza arbitraria.
Lo sviluppo del progetto ha portato il gruppo ad affrontare problematiche nel campo della programmazione con le quali non ci si era confrontati in precedenza, ovvero realizzare un software partendo da una situazione reale che tenesse conto delle imprecisioni hardware; inesattezze dovute alle componenti del sistema, sia meccaniche che elettroniche, al montaggio di queste ultime, alle tecniche di rilevazione dei test empirici e al degrado delle parti coinvolte, sia per quanto riguarda le capacità effettive che per la precisione.

8.1 Test di precisione

8.2 Sviluppi Futuri

Il sistema ha ampi margini di miglioramento dal punto di vista dell'hardware utilizzato. La struttura può essere realizzata in lega metallica garantendo una maggiore uniformità e delle proprietà meccaniche migliori, la molla traente può essere sostituita con un pezzo singolo avente delle specifiche dichiarate per una definizione migliore del range d'azione e delle forze applicabili, il cesto di lancio e il proiettile possono essere cambiati con due componenti aventi forma complementare al fine di annullare totalmente la traslazione verso l'alto in fase di lancio, inoltre il proiettile può essere realizzato con un materiale più aerodinamico, la cerniera che regola il movimento del braccio propellente può essere sostituita con una che non permetta traslazioni laterali.
Per quanto riguarda le componenti elettroniche il sensore ad ultrasuoni può essere sostituito con uno al laser che al prezzo di un costo maggiore fornisce una precisione molto superiore, i servo motori possono essere sostituiti con una versione avente il meccanismo interno in titanio presente in commercio, meno soggetto quindi alle sollecitazioni dovute agli urti in fase di lancio, inoltre la barra di arresto può essere inclinata da una coppia di servo motori anziché da un singolo per distribuire meglio gli urti ed evitare lo sbilanciamento da un lato, l'alimentatore può essere sostituito con uno avente voltaggio maggiore in grado di far operare al massimo potenziale tutti i servo motori del circuito.

References

[1] Arduino official website.

[2] Servo database - rc servo specs and reviews -.