T-RECS: Build; Model; System Identification

Kedar More

Abstract

This report documents the modelling and system identification of Transportable Rotorcraft Electronic Control System (T-RECS). It goes through the process step by step from the construction of the system to the initial testing. Some experiments are run to find out the exact dynamical model and convert it into a transfer function. This will be used to study the system and get the best PID control parameters to stabilise it.[2]

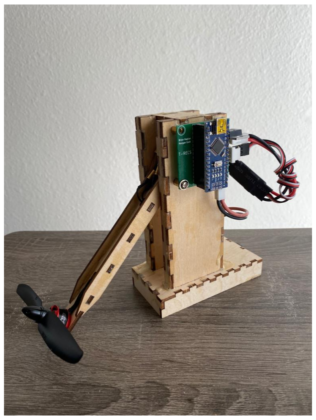


Figure 1: TRECS Assembly

1 Assembly

1. First I created the base for the assembly with 5 part fit into each other by the grooves provided. This will provide some space to clamp and ground the system. A slot for input barrel plug is provided in on of the parts.

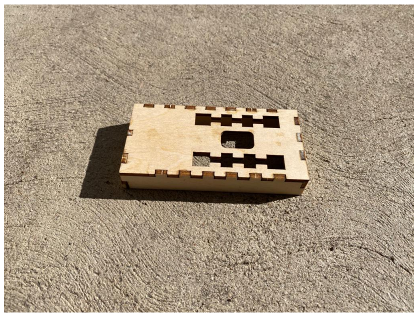


Figure 2: Base of the Assembly

2. After that I assembled 2 pillars with the bearings placed in the larger holes to support the two shafts. The standoffs (both original and custom) were tightened in the holes provided around the larger holes. The pillars were assembled such that I had to place the individual parts inside the slots on the base and then glue them together. The side with bearing in placed inside and the custom hole is placed on the right hand side as viewed from the clamping side.

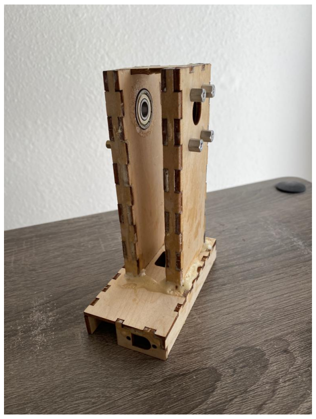


Figure 3: Pillars assembled

3. Next thing to assemble is the arm and attach it to the pillar by the shafts. the top part on the pillars is there just to keep them together and prevent the system from going haywire if the motor starts rotating more than needed. But this was hindering the motion of the arm and an unnecessary friction was introduced. So I opted to give the part a cut from the center.

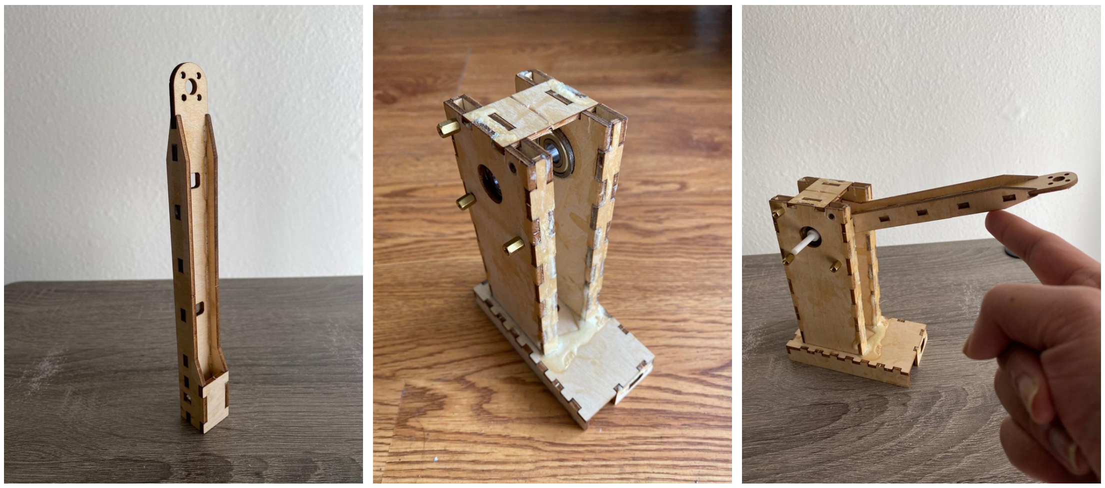


Figure 4: Arm assembled and a cut made on the top part

4. After that make all the electronic connections with the motor, encoder circuit and the Arduino Nano.

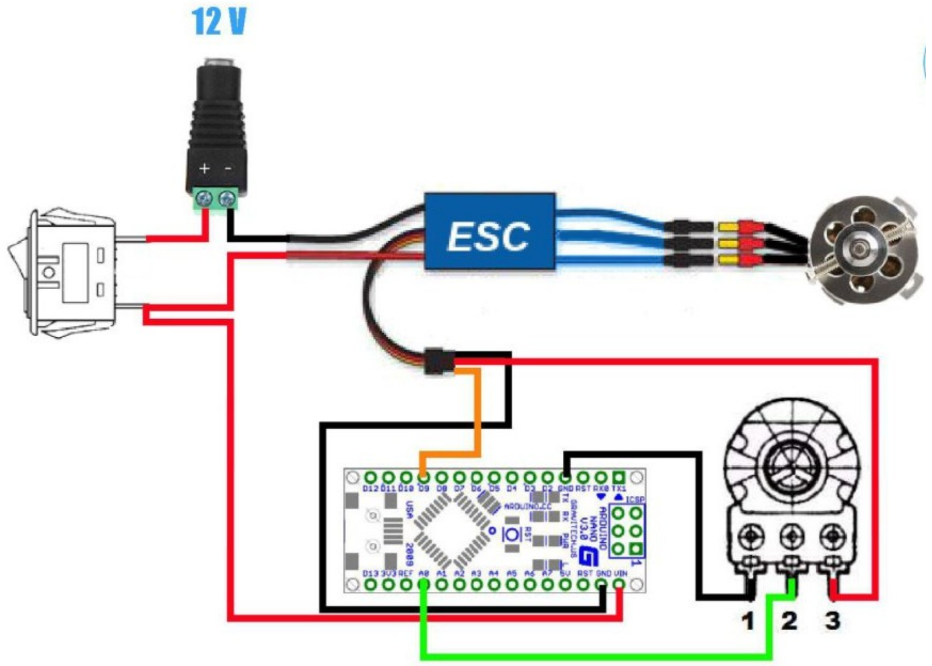


Figure 5: Wiring Schematic[4]

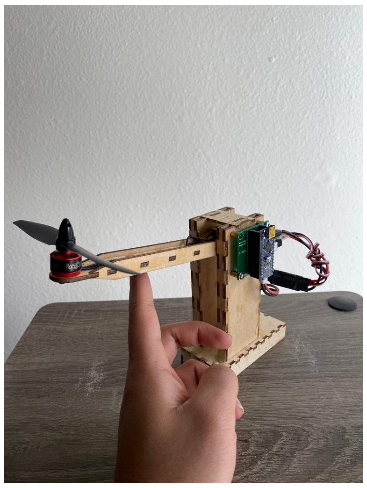


Figure 6: Assembly with wiring and components

2 System Identification

System Modeling

1. Free Body Diagram

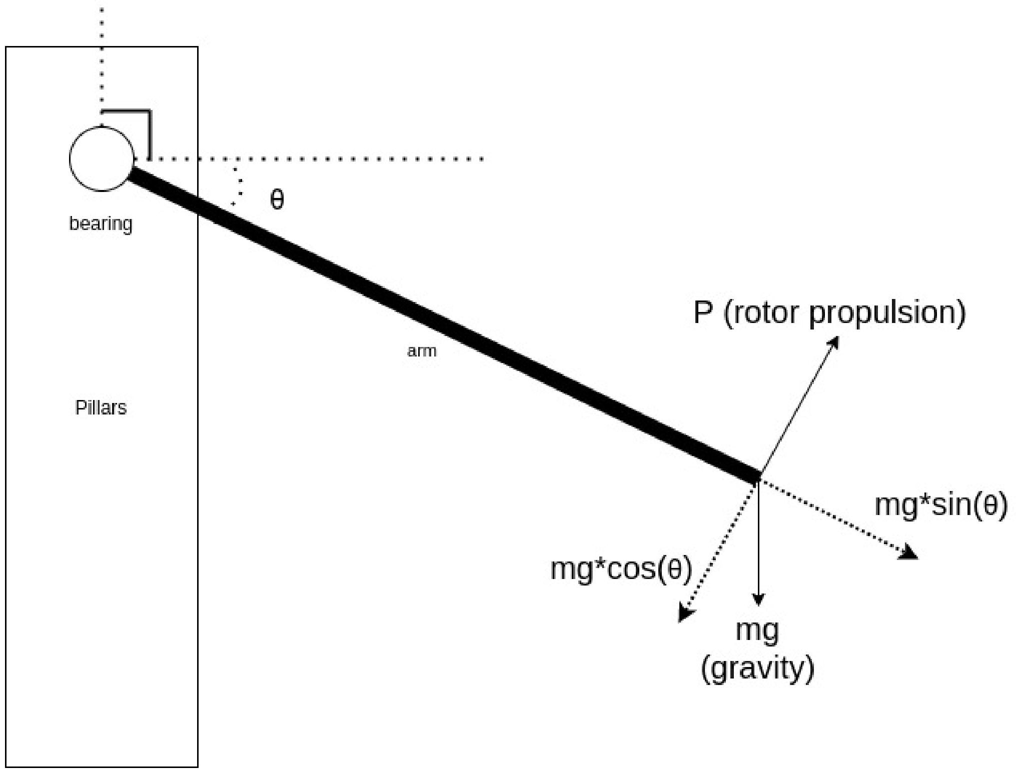


Figure 7: Free Body Diagram of the Arm

2. Calculations

   Here,
   J = Moment of Inertia
   d = Drag
   L = Length of Arm

$$J\ddot{\theta} = PL - dL\dot{\theta} - mgL\cos\theta$$

$$\ddot{\theta} = \frac{L}{J}(P - d\dot{\theta} - mg\cos\theta)$$

   This is the original equation without linearization. The equation should be of the form

$$Output = System * Input$$
$$\theta = G * P$$
$$P = \frac{J}{L}\ddot{\theta} + d\dot{\theta} + mg\cos\theta$$

3 Electronic Components

1. BLDC Motor

   The rotor used is Racerstar Racing Edition BR1306.

Figure 8: BR1306 BLDC Motor[6]

   This motor has the following specifications:

(a) Brand: Racerstar

(b) Item name: BR1306 3100KV Motor

(c) KV: 3100

(d) Stator Diameter: 17.6mm

(e) Stator Length: 12.7mm

(f) Shaft Diameter: M5

(g) Motor Dimensions(Dia.*Len): Phi;17.6times;14.5mm

(h) Weight (g): about 11g

(i) No.of Cells(Lipo): 1-2S

(j) Max Continuous current(A): 6.6A

(k) Max Continuous Power(W): 49W

(l) Internal resistance: Ω

(m) Usage: for 150-200 Glass Multirotor Frame Kit

Physics of Rotor:

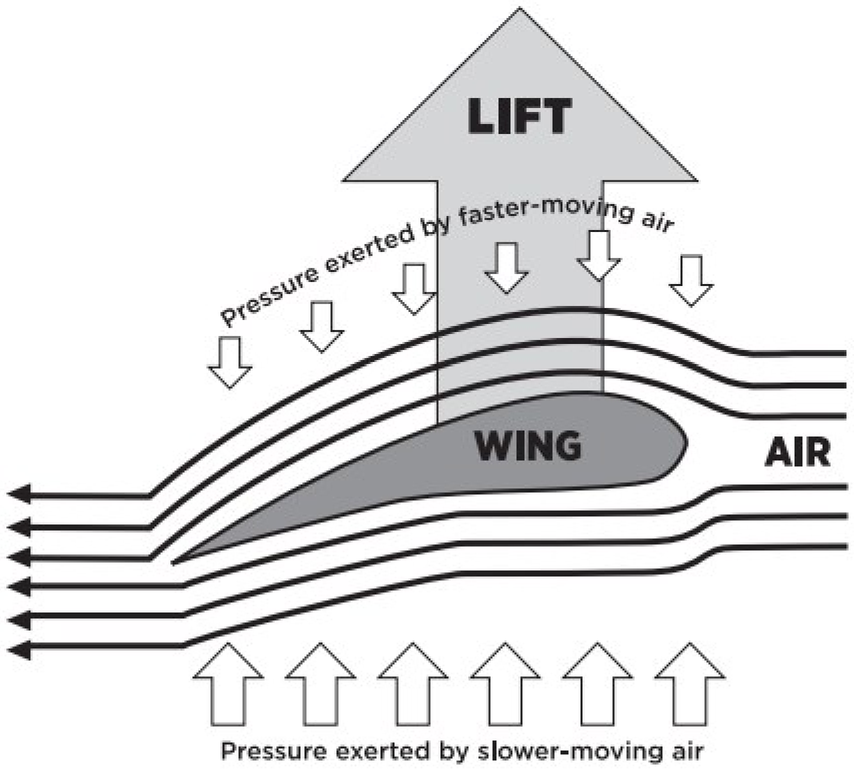

LIFT
Pressure exerted by faster-moving air
WING
AIR
Pressure exerted by slower-moving air

Figure 9: Principle of propeller lift[5]

There is a simple Bernoulli's equation which helps us find the upward force.

$$P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho v_2^2$$

where P is the force, ρ is the density of the medium and v is the velocity of the medium.

The shape of propeller enables the air to flow smoothly from above and restricts the air below it.

Let us consider that 1 is the place above the propeller and 2 is the place below it.

Therefore $v_1 > v_2$ and the density $\rho$of the medium is constant. This in turn states that $P_1 < P_2$. This difference in the force promote the upward propeller motion.

2. ESC

   ARRIS Swift Series BLHeli 20A 2-4S BEC 5V/1A Brushless ESC for RC Multi-rotor

Figure 10: ESC[1]

The ESC has following specifications:

(a) Programe: BLheli, support oneshot 125
(b) Continuous Current: 20A
(c) Burst Current (10S): 30A
(d) BEC: 5V/1A

(e) Lipo Cells: 2-4S
(f) Weight: 10g
(g) Size (excluding plugs): 28 x 15 x 6mm
(h) Typical Applications (for reference): 330-550 Multi-rotor

3. Potentiometer
   Schematic:

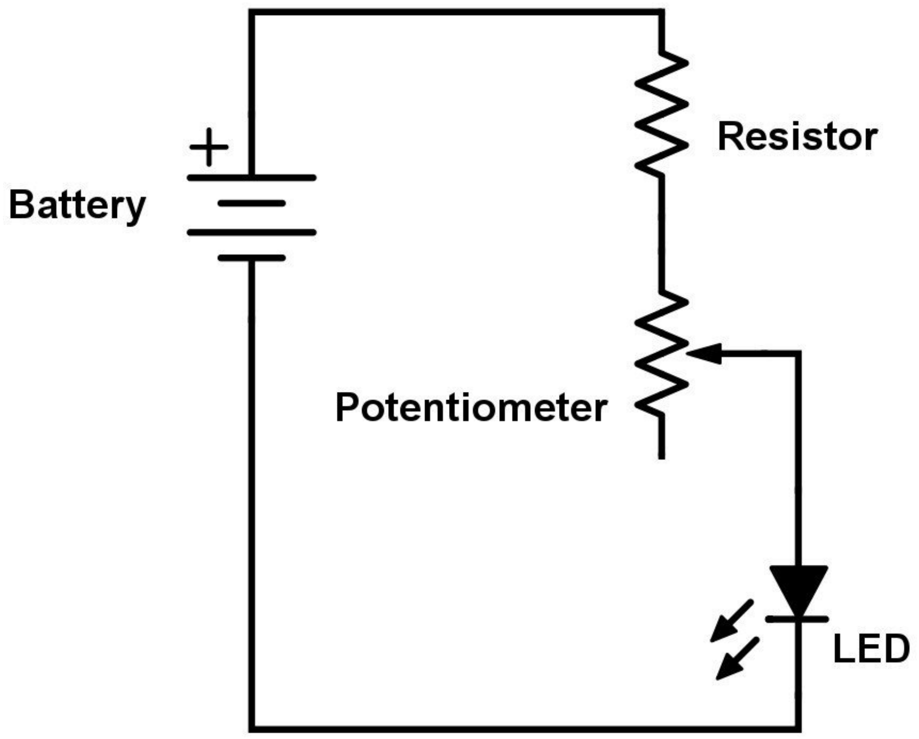


Figure 11: Schematic of Potentiometer[7]

Potentiometer is a three pin component with a Vcc, Ground and the sliding pointer which changes the internal resistance. The output of the pointer pin is the ratio of the the two resistances on the opposite sides of the pointer pin. There is another resistor in series to the potentiometer resistor which is the current limiting resistor. This prevents the current from being too high while it is inpput into a microcontroller pin.

4 Experiments

According to the lumped parameter analysis the following equation is to be solved.

$$P = \frac{J}{L}\ddot{\theta} + d\dot{\theta} + mg\cos\theta$$

$$\theta'' + \frac{d}{mL^2}\theta' + \frac{g}{L}\cos\theta = \frac{PL}{J}$$

$$A = \frac{d}{mL^2}$$

$$B = \frac{g}{L}$$

$$C = \frac{PL}{J}$$

The experiments done are based on the open loop system with the only one sensor used. Potentiometer is used to take angle from the system in degrees. This feedback is used as the output of our equation in degrees with the input being input of the motor.

1. Drop Test

   Let us eliminate of of the parameter that is the propeller force P by making it 0. In the physical model it represents that the motor is turned off. Under the influence of gravity the arm will fall down when we give an initial speed as 0. Here I recorded the angles of the arm with the corresponding time stamp.

   The angles and time were plotted against each other and the curve was fitted as a 3rd degree polynomial. This is the equation for the angle of the arm with the horizontal. This curve was differentiated to get the value for θ'. The same procedure was done with the θ' values to get θ'' values.

   Here we get a system of over defined equations with the coefficients as:

$$\theta'' + A\theta' + B\cos\theta = 0$$

   The method of solving these equations is by using the pseudo-inverse of the input matrix with a zero output matrix. This will fetch us the non trivial solution of the equations of the form:

$$[\theta''\theta'\cos\theta]x = [0]$$

   Calculations:

   The fitted equation of the 3^{rd} degree polynomial fitted curve between cos of the angle and time is

$$\cos\theta = -237.6t^3 + 5.743t^2 + 2.21t + 0.8971$$

and the curve between time and angle is

$$\theta = 487.7t^3 - 1751t^2 - 295t + 26.24$$

with the subsequent derivatives of angle, curve between time and angle' is

$$\theta' = -6.958t^2 - 1.599t + 0.1738$$

curve between time and angle' is

$$\theta'' = -0.02698t + 0.0001268$$

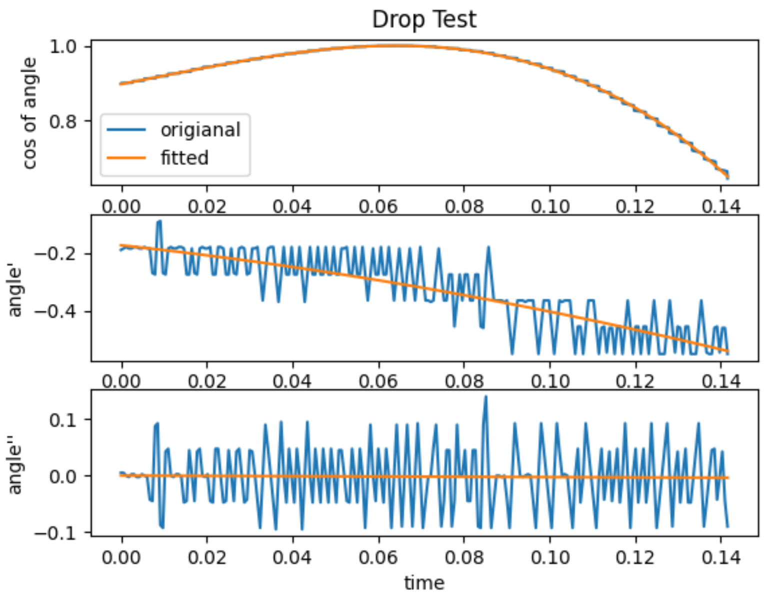
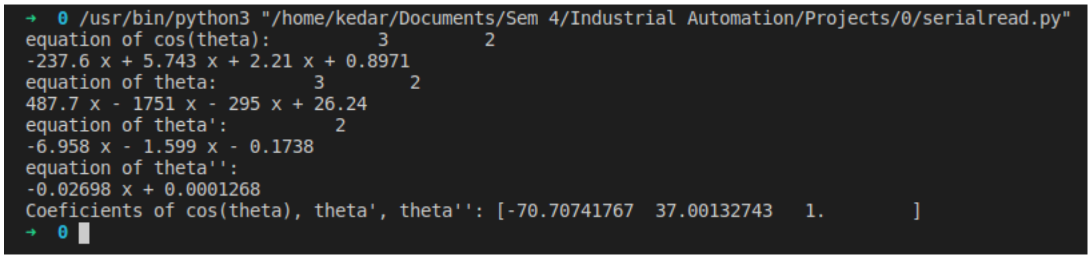


Figure 12: Drop Test



```
→ 0 /usr/bin/python3 "/home/kedar/Documents/Sem 4/Industrial Automation/Projects/0/serialread.py"
equation of cos(theta):          3         2
-237.6 x + 5.743 x + 2.21 x + 0.8971
equation of theta:         3       2
487.7 x - 1751 x - 295 x + 26.24
equation of theta':          2
-6.958 x - 1.599 x - 0.1738
equation of theta'':
-0.02698 x + 0.0001268
Coeficients of cos(theta), theta', theta'': [-70.70741767  37.00132743   1.        ]
→ 0
```

Figure 13: Drop Test Results

From this experiment we get the following lumped parameters:

$$A = 37.001, B = -70.707$$

$$B = \frac{g}{L} = -70.707$$

$$L = 0.1387m = 13.874cm$$

2. Steady State Test

   Now let us consider some input given to the motor. I gave the input to the motor from 70 to 130 and give each input some time to reach the steady state. The sine of angles are plotted against the given input . The line is fitted with 1st degree polynomial to get the slope.[3]

$$mg\cos\theta = P$$
$$mg\cos\theta = Ku^2$$
$$\frac{\cos\theta}{u^2} = \frac{K}{mg}$$

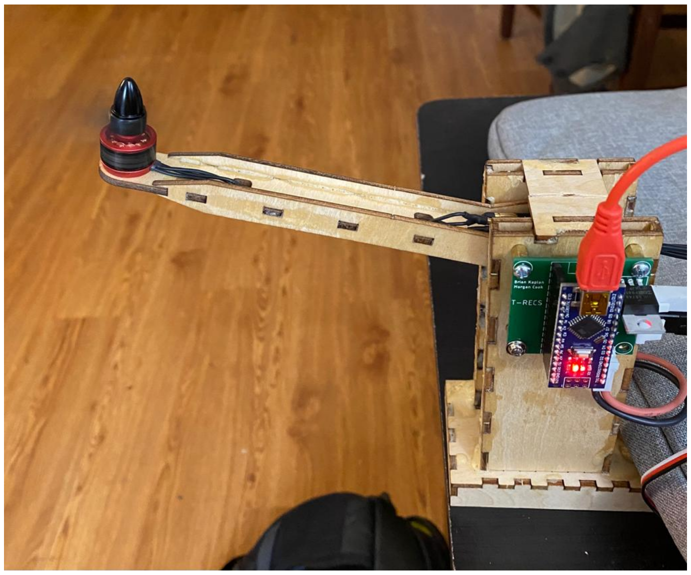


Figure 14: System at a steady state

Calculations:

After the curve was fitted as a line the equation of the line was found out to be

$$\cos\theta = 5.42798454 * 10^{-05} u + 6.33526554 * 10^{-02}$$

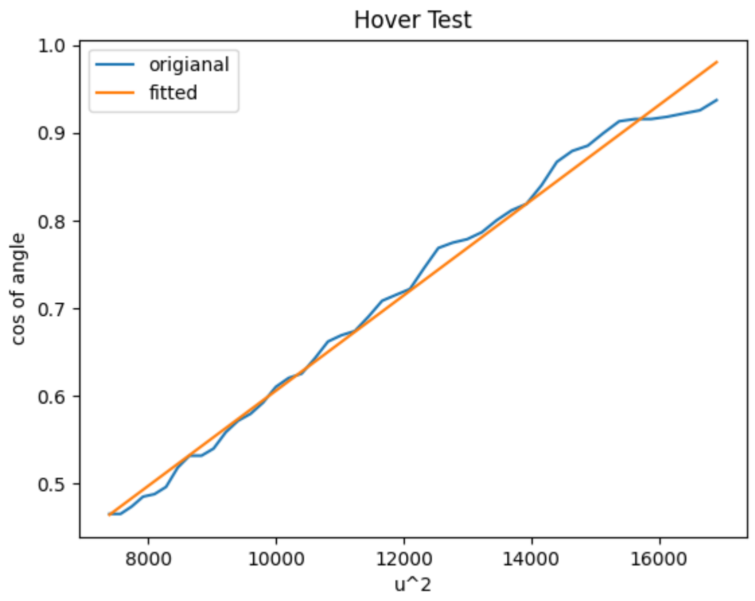


Figure 15: Steady State Test

Here we get the equation:

$$\frac{K}{mg} = 5.42798454 * 10^{-05}$$

The mass of the assembly was calculated to be 15g.

$$K = 7.98 * 10^{-6} N s^2$$

$$C = \frac{PL}{mL^2}$$

$$C = \frac{Ku^2}{mL} = 3.834 * 10^{-3} u^2$$

Final equation for the system comes out to be:

$$\theta'' + A\theta' + B\cos\theta = C$$

$$\boxed{\theta'' + 37.001\theta' - 70.707\cos\theta = 3.834 * 10^{-3} u^2}$$

Transfer Function:
Taking the Laplace transform of $\theta'' + A\theta' + B\cos\theta = C$
Linearizing about $\theta = 0$ we get $\cos\theta = 1$

$$s^2\mathcal{L}(\theta) + As\mathcal{L}(\theta) + B\frac{1}{s}\mathcal{L}(\theta) = 3.834 * 10^{-3}\frac{2}{s^3}\mathcal{L}(u)$$

$$\frac{\mathcal{L}(\theta)}{\mathcal{L}(u)} = \frac{3.834 * 10^{-3}\frac{2}{s^3}}{s^2 + As + B\frac{1}{s}}$$

$$\boxed{\frac{\mathcal{L}(\theta)}{\mathcal{L}(u)} = G = \frac{7.668 * 10^{-3}}{s^5 + 37.001s^4 + 70.707s^2}}$$

5 Problems Faced with Solutions

1. High Friction between the pillars and arm:

   To solve this problem I tried to shift the position of bearings axially. But they were permanently glued together. Hence, I gave a cut to the top part so that there is no force to hold the two pillars together.

2. Hysteresis of the shaft inserted into the encoder:

   While recording the angles there is a lag of $5°$ to $10°$. It was due to the clearance fit between the shaft and encoder. I wrapped the shaft with tape to make it an interference fit.

6 Conclusion

By conducting the above tests, we can experimentally verify the system coefficients. This is the open loop system which will be used with a controller. The equation found is very robust and should be used with a feedback loop. The system parameters are robust due to the imperfect construction of the T-REX.

References

[1] *ARRIS Swift Series 20A 2-4S BEC 5V/1A Brushless ESC for RC Multi-rotor.* `https://www.rc-wing.com/arris-swift-20a-2-3s-blheli-brushless-esc.html`. (Accessed on 02/20/2021).

[2] *Downloads — Tangibles That Teach.* `https://www.tangiblesthatteach.com/downloads`. (Accessed on 02/20/2021).

[3] Eniko T Enikov and Giampiero Campa. “Mechatronic aeropendulum: demonstration of linear and nonlinear feedback control principles with matlab/simulink real-time windows target”. In: *IEEE transactions on education* 55.4 (2012), pp. 538–545.

[4] *F.Lab’s DIYbio Centrifuge by F_Lab_TH - Thingiverse.* `https://www.thingiverse.com/thing:1175393`. (Accessed on 02/20/2021).

[5] Ruqiong Qin and Chunyi Duan. “The principle and applications of Bernoulli equation”. In: *Journal of Physics: Conference Series*. Vol. 916. 1. IOP Publishing. 2017, p. 012038.

[6] *Racerstar Racing Edition 1306 BR1306 4000KV 1-2S Brushless Motor CW/CCW For 150 180 200 Multirotor.* `https://www.racerstar.com/racerstar-racing-edition-1306-br1306-4000kv-1-2s-brushless-motor-cw-or-ccw-for-150-200-rc-drone-fpv-racing-multi-rotor-p-32.html`. (Accessed on 02/20/2021).

[7] *The Potentiometer And Wiring Guide - Build Electronic Circuits.* `https://www.build-electronic-circuits.com/potentiometer/`. (Accessed on 02/20/2021).

Appendices

1. Python: Import data from a Serial Port

```
import serial
import time
import matplotlib.pyplot as plt
import csv

# set up the serial line
ser = serial.Serial('/dev/ttyUSB0', 115200)
start=time.time()

# Read and record the data
data =[]                       # empty list to store the data
t=[]
# while True:
print("START")
for i in range(1000):
    # print("ready")
    b = ser.readline()         # read a byte string
    string_n = b.decode('ISO-8859-1')  # decode byte string into Unicode
    string = string_n.rstrip() # remove \n and \r
    # flt = float(string)        # convert string to float
    print(string)
    data.append(string)           # add to the end of data list
    t.append(time.time()-start)
    # time.sleep(0.2)            # wait (sleep) 0.1 seconds

with open("full.csv","w") as f:
    wr = csv.writer(f,delimiter="\n")
    # wr.writerow(data)
    wr.writerows((data,t))

plt.plot(t,data)

plt.show()

ser.close()
```

2. Python: Process the data

```
import csv
import numpy as np
import matplotlib.pyplot as plt
import math

a=np.empty((0))

with open('out.csv', newline='\n') as csvfile:
     spamreader = csv.reader(csvfile, delimiter=' ', quotechar='|')
     for row in spamreader:
        a=np.append(a,float(''.join(row)))


poly=[3,2,1]

final=np.resize(a,(2,int(len(a)/2)))

t=final[1][0:-5]
t=t-final[1][0:-5][0]
angle=final[0][0:-5]
# t=final[1]
# t=(t-final[1][0])
# angle=final[0]
y=angle
ycos=np.cos(np.radians(angle))



ax=plt.subplot(311)

z = np.polyfit(t, ycos, poly[0])
# print(z)
f = np.poly1d(z)


print("equation of cos(theta): ",f)

x_new = np.linspace(t[0], t[-1], len(angle))
y_new_cos = f(x_new) # fitted


ax.plot(t,ycos)
ax.plot(x_new, y_new_cos)
plt.legend(['origianal', 'fitted'])
plt.xlabel("time")
plt.ylabel("cos of angle")
plt.title("Drop Test")
# plt.show()



z = np.polyfit(t, y, poly[0])
# print(z)
f = np.poly1d(z)


print("equation of theta: ",f)

x_new = np.linspace(t[0], t[-1], len(angle))
y_new = f(x_new) # fitted




ax=plt.subplot(312)

ydash=np.gradient(y)
ax.plot(x_new, ydash)

z = np.polyfit(t, ydash, poly[1])
# print(z)
f = np.poly1d(z)


print("equation of theta': ",f)

x_new = np.linspace(t[0], t[-1], len(angle))
y_new_dash = f(x_new) # fitted

ax.plot(x_new, y_new_dash)

plt.xlabel("time")
plt.ylabel("angle'")
```

```
ax=plt.subplot(313)

yddash=np.gradient(ydash)
ax.plot(x_new, yddash)

z = np.polyfit(t, yddash, poly[2])
# print(z)
f = np.poly1d(z)

print("equation of theta'': ",f)

x_new = np.linspace(t[0], t[-1], len(angle))
y_new_ddash = f(x_new) # fitted

ax.plot(x_new, y_new_ddash)

plt.xlabel("time")
plt.ylabel("angle''")


# y_new, ydash, yddash

theta=np.transpose(np.vstack((y_new_cos,ydash,yddash)))
b=np.zeros(y_new.shape)+0.0001

# x=np.linalg.lstsq(theta,b)
x=np.matmul(np.linalg.pinv(theta),b)
x=x/x[2]
print("Coeficients of cos(theta), theta', theta'': "+str(x))


plt.show()
```