EE160 Final Project: Kite Controller

Hongxiang Gao

***Abstract*—It is spring now, which is a good time to fly kites. However, it is quite difficult to keep the kite in a steady state with a wind force. This article will give the explicit equations to model the system and apply the concept of PID Control to get the solution. Detailed mechanical analysis equations and simulation codes will be given. In addition, this paper will analyze how the value of K will affect the variance of the result.**

***Index Terms*—PID Control, Wind force, Kite.**

I. Introduction

This article gives the general solution of keep a kite in sky under wind force. This report is divided into five parts.

Part I gives the content and introduction of the whole article.

In part II, it will give a detail procedure of problem formulating, which contains three sections: General Introduction, Constraints and Assumptions, Mechanical Equations, Linearized Model and Some Details About RK4.

Part III gives the simulation code.

Part IV will give the simulation result, which include two parts. And it will also give some interpretations of the plots.

Part V is the conclusion, it will summarize the highlights of this article.

II. Problem Formulation

Before we go through the details of the model, some intuitions and constraints will be declared in advance.

A. General Introduction

The input of the system is the force imposed by hand in both tangential and normal directions. The output is the length of the rope and the angle between the rope and the horizontal level. The aim of the project is to apply the concept of PID to control the input signals so that the final steady state can be adjusted to the presupposed value. It will also give the simulation result of the code and analyze it to give a optimal control parameters.

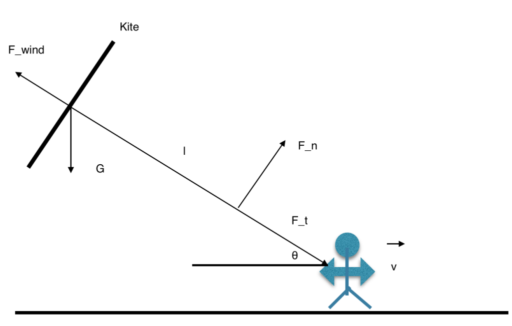

Fig. 1. General Picture of a Kite System

Figure 1 shows the general scene of a kite system. In the picture, the controller is trying to control the kite in the sky, however, it is hard for him to adjust the kite at a steady state as the parameters he wants. There are two input signals, they are tangential force and normal force. The outputs contains the length of the rope and the angle between rope and the horizontal level. There are four force imposed on the kite, they are gravity, tangential force on the rope, the normal force on the rope and the forced imposed by wind. The controller is moving at a uniform velocity of v. Detailed parameters and some mechanical constraint will be discussed in part two: Model Construction.

B. Constraints and Assumptions

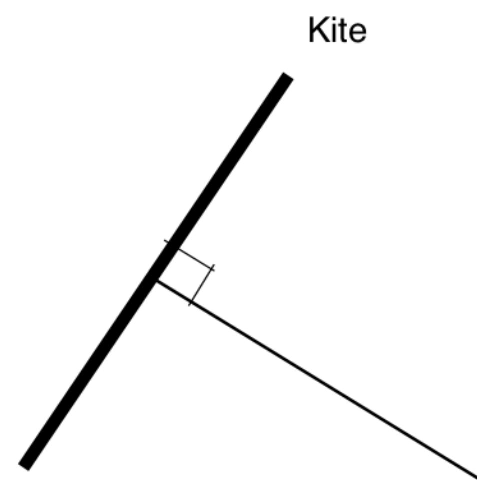

Fig. 2. Detail of The Kite

First assumption is the angle of the kite. As an intuition, why a kite is flying in the sky, the plane of it should be perpendicular to the rope like the picture showed in Figure 2. Under this assumption, the angle of the rope will also has an effect on the angle of the kite, further more, it will affect the wind force imposed on the kite.

Second one is an assumption on the controller, in this scene, the controller is moving at a uniform velocity as 1m/s.

Third assumption is about the natural environment. The wind force is proportional to the absolute velocity of the kite, but there is an extra force F_0, which follows a normal distribution.

C. Mechanical Equations

The key part of the system is to analyze the force imposed on the kite and combine it with the assumptions given above.

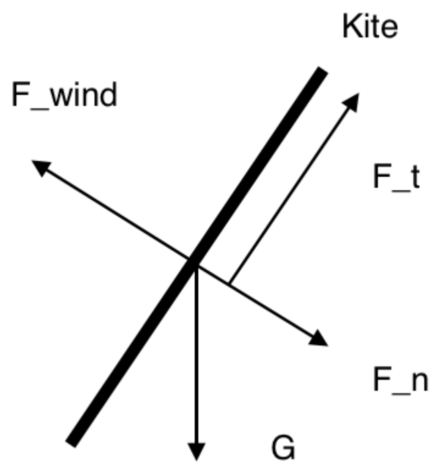

Fig. 3. Mechanical Analysis of The Kite

Figure 3 shows the mechanical analysis of the kite. Different from the traditional method of mechanical analysis at x-y plane, in this model, I did it in tangential and normal directions. The equations are given:

$$F_n = F_{wind} - mg\sin(\theta) - F_{ni} \tag{1}$$

$$F_t = -mg\cos(\theta) + F_{ti} \tag{2}$$

For simplicity, the gravity constant is given as $g = 10m/s^2$ and the mass of the kite is $m = 1kg$.

F_{ni} and F_{ti} in the two equations are the inputs. In Equation (1), it involves a unknown term F_{wind}, this term is related to the absolute value of the kite, the equation that describe it is:

$$F_{wind} = K_{wind} * \sqrt{{v_x}^2 + {v_y}^2} + F_0 \tag{3}$$

There gives Four more unknown terms, K_{wind}, v_x, V_y and F_0. K_wind is the coefficient to convert the velocity to the force. It is proportional to the area of the kite, in this system, it is taken as:

$$K_{wind} = 10(N * s^2/m^2) \tag{4}$$

v_x and v_y are to describe the absolute value of the kite, they are given as:

$$v_x = v_{controller} - \dot{l}\cos(\theta) + l\dot{\theta}\sin(\theta) \tag{5}$$

$$v_y = \dot{l}\sin(\theta) + l\dot{\theta}\cos(\theta) \tag{6}$$

The velocity of the controller is given as $v_{controller} = 1m/s$, which is a constant.

F_0 is the natural force, and since it is unpredictable, it is given as the normal distribution. The mean value is 10N, and the variance is 1N^2. It also can be seen as a disturbance to the system.

At last, the acceleration of the kite in tangential and normal directions can be described as:

$$\ddot{l} = \frac{F_n}{m} \tag{7}$$

$$\ddot{\theta} = \frac{F_t}{ml} \tag{8}$$

The state of the system is z=$[l; \theta; \dot{l}; \dot{\theta}]^T$. And with the help of Equation $(1) - (8)$, the model is completed.

D. Linearized Model

The model given above is quite difficult to calculate, that is say, it will be quite slow for a computer to calculate the numerical solution. To make a better performance in calculation speed, a linearized model is needed. Nevertheless, it will, inevitably, dwindle the accuracy of the result. In linearized model, some simplification is applied to the Equation $(1)-(8)$. And to follow the form of a traditional ODE function given as:

$$\dot{z} = Az + Bu \tag{9}$$

The simplification includes: $\sin(\delta\theta) = \delta\theta$ and $\cos(\delta\theta) = 1$ for $\delta\theta$ is a small value. The two coefficient matrices are given as

$$A = \begin{pmatrix} 0 & 0 & 0 & \frac{g\cos(\theta_s)}{l^2} \\ 0 & 0 & -g\cos(\theta_s) & \frac{g\sin(\theta_s)}{l} \\ 1 & 0 & -\frac{K_{wind}\cos(\theta_s)}{m} & 0 \\ 0 & 1 & \frac{K_{wind}l\sin(\theta_s)}{m} & 0 \end{pmatrix} \tag{10}$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{pmatrix} \tag{11}$$

The state z and the input signal u is given as:

$$z = \begin{pmatrix} l \\ \theta \\ \dot{l} \\ \dot{\theta} \end{pmatrix} \tag{12}$$

$$u = \begin{pmatrix} F_{ni} \\ F_{pi} \end{pmatrix} \tag{13}$$

Both of the two models can be programmed in MATLAB and calculated numerically with the concept of RK4.

E. Some Details About RK4

$$
\begin{aligned}
k_1 &= f(y_n) \\
k_2 &= f\left(y_n + \frac{h}{2}k_1\right) \\
k_3 &= f\left(y_n + \frac{h}{2}k_2\right) \\
k_4 &= f(y_n + hk_3) \\
y_{n+1} &= y_n + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right);
\end{aligned}
$$

Fig. 4. RK4 Algorithm

RK4 algorithm can be used to calculate the numerical solution of a differential equation. And it follows the calculation tactic shown in Figure 4. However, the detailed program will depend on the case of the differential equation.

III. Solution Method

The model is applied in MATLAB and it contains three parts. In this section, I used the non-linear model to get a better accuracy, but it may cause a larger processing time consumption.

A. ODE Function

```
function [ dz ] = ode( t,z,v,Ft,Fn )
    %Parameter Setting
    m=1;
    g=10;
    l=z(1);
    theta=z(2);
    dl=z(3);
    dtheta=z(4);

    %The value of natural wind is a normal
     distribution around 10
    %The variance of F0 is adjustable
    F0=random('norm',10,1,1,1);

    %Set the value of dz
    dz(1,1)=dl;
    dz(2,1)=dtheta;

    %The velocity->force coefficient
    K_wind=10;

    %Velocity of the kite
    vx=v-dl*cos(theta)+l*dtheta*sin(theta);
    vy=dl*sin(theta)+l*dtheta*cos(theta);

    %F_wind
    F_wind=K_wind*sqrt(vx^2+vy^2)+F0;

    %Centripetal force and tangential force on the
     kite
    F_n=F_wind-m*g*sin(theta)-Fn;
    F_t=-m*g*cos(theta)+Ft;

    %Centripetal and tangential acceleration
    dz(3,1)=F_n/m;
    dz(4,1)=F_t/m/l;
end
```

./code/ode.m

This program describe the whole system, which converts the mathematical equations to the code. The equations are non-linear, but the simulation result will have higher accuracy.

B. RK4

```
function [ x,y ] = RK4( ufunc,y0,h,a,b,Ft,Fn,v )
%Set Step
n=floor((b-a)/h);
%Initial Time
x(1)=a;
%Initialization of y
y(:,1)=y0;
for i=1:n
    %Time stepping
    x(i+1)=x(i)+h;
    %Apply RK4 Algorithm Approximation
    k1=ufunc(x(i),y(:,i),v,Ft,Fn);
    k2=ufunc(x(i)+h/2,y(:,i)+h*k1/2,v,Ft,Fn);
    k3=ufunc(x(i)+h/2,y(:,i)+h*k2/2,v,Ft,Fn);
    k4=ufunc(x(i)+h/2,y(:,i)+h*k3,v,Ft,Fn);
    %Add up all the components
    y(:,i+1)=y(:,i)+h/6*(k1+2*k2+2*k3+k4);
end
```

./code/RK4.m

This program is for find the numerical solution of a differential equation. The input of this function is relate to the model which has been presented in ode.m. The numshoot is given as 5, when it is given a larger value, the accuracy of the numerical solution of the ODE function will be improved, but the time consumption will also be increased dramatically.

C. Kite

```
clear all
%%Parameter Statement
m=0.5;
g=10;

%%Initialization
l=7;
theta=pi/3;
dl=0;
dtheta=0;
state0=[l;theta;dl;dtheta];

%%Simulation
%Timing Initialization
N=10000;
T=20;
del_t=T/N;
numshoot=5;

%State/Control Initialization
state=zeros(4,N+1);
state(:,1)=state0;
x_m=zeros(1,N+1);
x_m(1)=0;
Ft=zeros(1,N);
Fn=zeros(1,N);

%The velocity of controller is 1m/s
v=1;

%%Set up PID control
%Set Kt
%Kpt=-10-----20
Kpt=-15;
Kdt=-8;
Kit=-0.0001;
```

```
%Set Kt
%Kpn=5---15
Kpn=10;
Kdn=2;
Kin=0.00001;

%Get error
ie=[0;0];
%The iterative loop number
K=1;
%variance_l=zeros(1,K);
%variance_theta=zeros(1,K);
%for j=1:K
    for i=1:N
        %Calculate Fp
        e(1,1)=state(1,i)-5;
        e(2,1)=state(2,i)-pi/6;
        Fpt=Kpt*e(2,1);
        Fpn=Kpn*e(1,1);

        %Calculate Fd
        de(1,1)=state(3,i);
        de(2,1)=state(4,i);
        Fdt=Kdt*de(2,1);
        Fdn=Kdn*de(1,1);

        %Calculate Fi
        ie(1,1)=ie(1,1)+e(1,1);
        ie(2,1)=ie(2,1)+e(2,1);
        Fit=Kit*ie(2,1);
        Fin=Kin*ie(1,1);

        %%Calculate F_total
        Ft(i)=Fpt+Fdt+Fit+8.66;
        Fn(i)=Fpn+Fdn+Fin+15;
        x_m(i+1)=x_m(i)+v*del_t;

        %Calculate Variance
        %variance_l(j)=variance_l(j)+e(1,1)^2/100;
        %variance_theta(j)=variance_theta(j)+e(2,1)
    ^2/100;

        %Use ode45 to solve ode
        [~,sta]=RK4(@ode,state(:,i),del_t/numshoot,(
    i-1)*del_t,i*del_t,Ft(i),Fn(i),v);
        ste=sta';
        state(:,i+1) = ste(end,:);
    end
    %end

%%Plot
%Plot States
figure;
i=0:T/N:T;
subplot(221)
plot(i,state(1,:));grid;
title('l(m)');
subplot(222)
plot(i,state(2,:));grid;
title('theta(rad)');
subplot(223)
plot(i,state(3,:));grid;
title('dl(m/s)');
subplot(224)
plot(i,state(4,:));grid;
title('dtheta(rad/s)');
%Plot Control Signal
figure;
subplot(211)
plot(i(2:N+1),Ft);grid;
title('Ft(N)');
subplot(212)
plot(i(2:N+1),Fn);grid;
title('Fn(N)');

figure;
i=10:T/N:T;
subplot(121)
plot(i,state(1,5001:end)-5);grid;
title('Error of l(m)');
subplot(122)
plot(i,state(2,5001:end)-(pi/6));grid;
title('Error of theta(rad)');
% Kpn=-19.9:0.1:-10;
% figure;
% subplot(211)
% plot(Kpn,variance_l);grid;
% title('Variance of l(m^2) with Kpt')
% subplot(212)
% plot(Kpn,variance_theta);grid;
% title('Variance of theta(rad^2) with Kpt')
```

./code/kite.m

This part will give general simulation of the system. The program has two functions, the first one is simulate the states changing tendency as the input control signals are given to the system, and the second one is to calculate the variance under different conditions. Some parameters in this program can be adjusted, such as K_{pt}, K_{pn} and initial states.

IV. Simulation Result

This section includes two parts: the state simulation and variance simulation.

A. State Simulation

In this section, we take the value of each control parameter as:

$$K_{pt} = -15; K_{dt} = -8; K_{it} = -0.0001 \tag{14}$$

$$K_{pn} = 10; K_{dt} = 2; K_{it} = 0.00001 \tag{15}$$

The steady state set is $\theta = \frac{\pi}{6}$ and the length of the rope is $l = 5m$. The initial state is $\theta_0 = \frac{\pi}{3}$ and $l_0 = 7m$.

Under these conditions, the simulation result is:

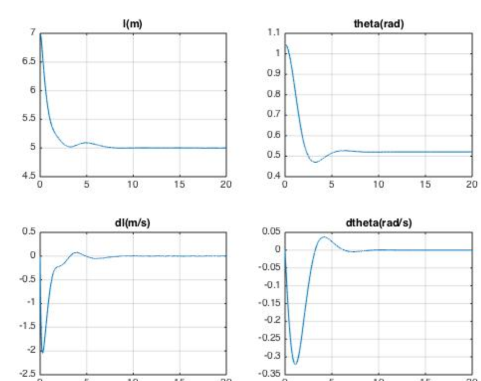

Fig. 5. States Plot

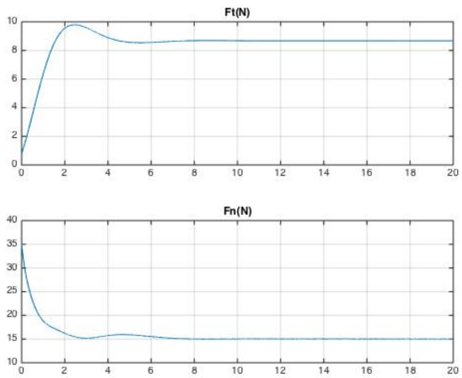

Fig. 6. F Plot

Figure 5 plots the four elements of the state. The time-span is set as 20s. For l and θ, they will go to $5m$ and $\frac{\pi}{6}$ respectively. And the other two will go to zeros.

Figure 6 gives the plot of force imposed to the rope.They will finally go to a steady value as set in the code. To test the accuracy of the control system, there gives the tolerance test of the steady state.

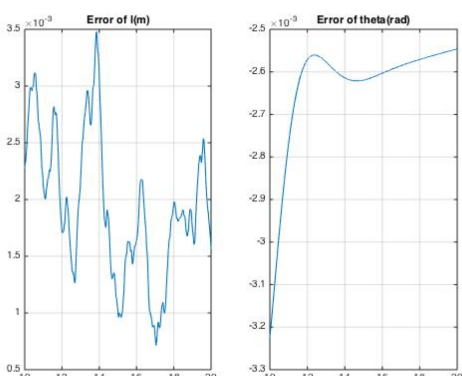

Fig. 7. Error Plot

Figure 7 gives the error of l and θ after 10 seconds. The absolute error of l is around $10^{-3}(m)$ and the error of θ is also in this regime. However, there gives a tendency that the error the l is fluctuating around $2 * 10^{-3}(m)$, while the error of θ is keeping decrease. Maybe given a larger time-span, the error of θ can be control within a better regime.

B. Variance Simulation

To get more detail information of the model, in the other word, finding a better control parameter K, variance analysis is necessary and inevitable.

1) Changing K_{pt}: In previous section, the value of K_{pt} is set as -15, in this part, it is ranged from -20 to -10, the span is 10.

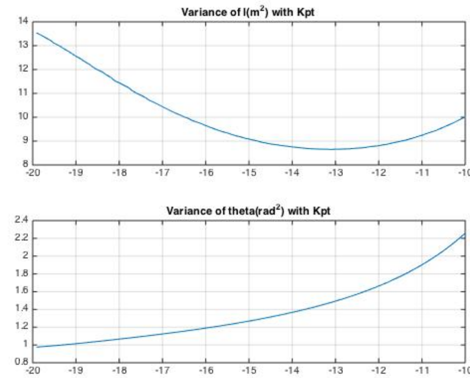

Fig. 8. Variance Analysis with K_{pt}

From Figure 8, the variance of l has a valley when the value of K_{pt} is located near -13, while the variance of θ keeps increasing as a higher value of K_{pt}.

2) Changing K_{pn}: In previous section, the value of K_{pn} is set as 10, in this part, it is ranged from 5 to 15, the span is 10.

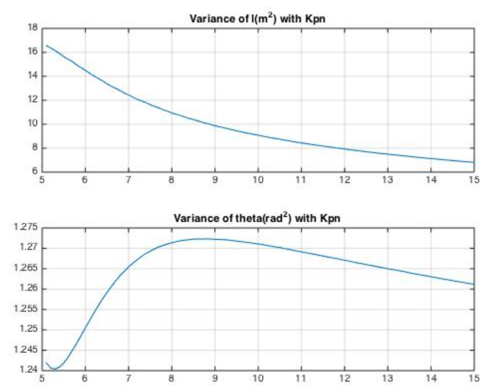

Fig. 9. Variance Analysis with K_{pn}

From Figure 9, the variance of l keeps decreasing as the value of K_{pn} is increasing, while the variance of θ is a peculiar curve. That curve, to some extent, like a cubic parametric curve. There shows a mountain when the value of K_{pn} is around 8.5, however, in the same plot, there shows a trough at $K_{pn} = 5.3$.

V. Conclusion

This article has several highlights.

First and foremost, the model of the scene is original, which has been modified for several times so that it can fit the intuition and common sense. And the model still remained improvable to have a better veracity of the simulation.

Besides, this article gives two calculation strategies, non-linear version and linearized version. Both of them can be taken into practice in program simulation. The linearized model will cost less time but the result will become less accurate. It's like a trade off to make which strategy to implement the model.

Last but not least, the article gives the analysis of variance with the change of some control parameters. From the simulation results, it will have a better performance in accuracy if K_{pt} is set as -13 and K_{pn} is 5.3.