A Kite Simulator Based on Aerodynamic Model

Lei Kang1, JianZhe Liang2 & CaiLing Wang1

1Xi'an Shiyou University, Xi'an, Shaanxi Province. R. China
2 Tsinghua University, Beijing, P. R. China

Abstract. Kite simulator is important for training air borne wind energy controllers. A kite model based on aerodynamics and ordinary differential equation has been proposed to calculate kites' trajectory in the air. A constant problem solver has also been developed to study the system's stability and cross validate the simulators behaviour. Elementary evaluation shows a typical trajectory of a kite and verifies the system's stability.

1 Introduction

It has been a consensus that there will be an increasing demand in airborne wind energy for the next few decades. Power kites, compared with traditional horizontal axis wind turbines, are able to access wind energy at higher altitude, which is of better quality than near ground wind resource [1]. Plenty of elaborate kite controllers utilizing artificial intelligence have been proposed and implemented [2]. Effective kite simulators enable the training process of controllers to be finished with high efficiency and randomness, leading to shorter training time and better real world performance.

We present a kite simulator based on an aerodynamic model which is capable of emulating a kite's trajectory according to system's initial parameter and characteristic parameters and sensors' input. Our model is able to solve the trajectory with both tension and acceleration input of the kite's rope, the latter of which is more desirable in reality as the force on a rope can be usually less accurately measured. We have also developed a constant model and its solver to study the system’s stability and cross validate the resulting trajectory of our simulator.

2 Model Design

2.1 Description and Dynamic Characteristic Analysis

In our model a kite is a combination of a plane-like panel generating wind forces and a mass point providing gravity. A rope is attached to the mass point with another ending to the base station (origin in our coordinate system). Without loss of generality, we restrict all motion in a 2 dimensional plane. Therefore, both the kite’s panel and the rope can be represented by a line segment. We further assume that these segments are orthogonal to each other.

As is shown by figure 1, while the kite is in the air, it is acted upon by a lift force and a drag force caused by wind. Gravity and tension on the rope are also applied to the kite. The lift force $F_l$ and drag force $F_d$ are orthogonal to each other and they are calculated using the following formulae

\[ F_d = \frac{1}{2} \rho \nu_{rel}^2 C_d(\alpha) \]  \hspace{1cm} (1)

\[ F_l = \frac{1}{2} \rho \nu_{rel}^2 C_l(\alpha) \]  \hspace{1cm} (2)

where $A$ is the area of the panel, $\rho$ stands for the density of air (regarded as a constant throughout the simulating process), $\nu_{rel}$ represents wind's relative velocity to kite. $C_l$ and $C_d$ are characteristic functions of the kite itself about the angle, denoted by $\alpha$, between $\nu_{rel}$ and the panel.

![Figure 1. Dynamic characteristic of kite simulator](#)

2.2 Algorithm

The key point in deriving the trajectory of the kite is to identify its acceleration at time $t$, denoted by $\ddot{a}(t)$. In general, one may write

\[ \ddot{a}(t) = f(p, \dot{s}) \]  \hspace{1cm} (3)

where $p$ is the system's initial parameter, $\dot{s}$ stands for systems state at time $t$. Once we derive the relationship $f$, we could use standard numerical method for solving ordinary differential equation $\ddot{a}(t) = \frac{d^2\mathbf{r}(t)}{dt^2}$ to get an
approximated trajectory. We adapt the Euler method here to solve the ODE \([3]\). We first choose a small step interval \(\Delta t\); then we derive the acceleration at \(t_0\), we update the kite’s velocity by adding \(\vec{a}(t_0) \Delta t\) to initial speed \(\vec{v}_0\); next, we update the kite’s position by adding \(\vec{v}_0 \Delta t\) to initial position vector \(\vec{r}_0\). We work out the acceleration, velocity and position at \(t_0 + \Delta t\), \(t_0 + 2 \Delta t\), \(t_0 + 3 \Delta t\)... iteratively to get the whole trajectory.

The form of relationship \(f\) is determined by the type of input, i.e., whether we are working with a sequence of pulling force on the rope or its acceleration. When working with tension, the kite’s acceleration can be derived simply by utilizing Newton’s second law of motion:

\[
\vec{a}(t) = \frac{1}{m} \left( mg - T \frac{\vec{r}}{\| \vec{r} \|} + F_d + F_i \right)
\]

(4)

Note that the input tension is simply a scalar, the tension force vector is in opposite direction with the rope. Thus the kite can achieve a balanced state so long as the sum of all forces but tension is parallel to the rope. Without solving ODEs, though it involves non-linear equations, as we shall see. To better understand the characteristic of the system’s stability and cross validate our simulator, we also developed a constant solver to get the kite’s balanced point by solving non-linear equations.

When the kite is at a balanced state without velocity, \(V_{rel}\) will now be equal to \(V_{wind}\). Also note that the tension on the rope is always parallel to the direction of rope. The most interesting state of kite occurs when it stops at certain position in the air. At this point, the kite is acted upon by a group of balanced force and its velocity is 0.

\[
\tan(\phi) = \frac{C_i(\frac{\pi}{2} - \phi) - 2mg}{C_d(\frac{\pi}{2} - \phi) - ApV_{wind}^2}C_d(\frac{\pi}{2} - \phi)
\]

(7)

where \(\phi\) is the angle between the rope and horizon.

We perform the standard bisection method to derive \(\phi\) at satisfying precision with precise knowledge of functions \(C_i\) and \(C_d\). Note that the difference of the two sides of equation \(7\) (we call it target function in the sequel) may have the same signs at 0 and \(\pi\). Thus before entering the bisection loop, our algorithm detects the value at \(0, \phi, 2 \phi, \ldots, \pi - \phi, \pi\), where \(\Delta \phi\) is user-defined detection step. The algorithm uses the first pair of neighboring points having values of opposite signs as the endings for initial bisection interval. It then performs the standard method to approximate the zero of the target function.

The overall procedure of our algorithm can be visualized using the flow chart in figure 2.

![Figure 1. Flow chart of simulation algorithm](image)

3 Evaluation

We have implemented the simulator and constant problem solver both in python. We have also developed a visualized demo to show the kite’s behavior when it is in the air. We place the kite at (-10, 0) with wind blowing from \(+\infty\) at the speed of 5 m/s. We use the input of rope's acceleration to calculate the kite’s trajectory. We set the acceleration to 0 throughout simulation, i.e., the length of the rope stays unchanged. We design the characteristic functions of the kite’s panel \(C_i\) and \(C_d\) to be:

\[
C_i(\alpha) = \frac{8}{\pi} \alpha (\pi - \alpha)
\]

(8)

\[
C_d(\alpha) = 0.01 + 0.6 \sin(\alpha)
\]

(9)
so that $C_l$ is monotonically increasing in $[0, \frac{\pi}{2}]$ while decreasing in $[\frac{\pi}{2}, \pi]$ as the wind is blowing from an opposite direction relative to the panel. The same qualitative analysis also applies to $C_d$.

As is depicted in figure 4, the kite lifts to the right top of origin after launching, and is slowly bellowed to a lower position. It finally becomes stable at that position with velocity gradually vanishing to 0.

We can also derive the stable position directly using the constant problem solver with the same initial and characteristic system parameters. The resulting tangent value of $\phi$ is 10.537 calculated by the constant solver (with precision set to be 0.01) and 10.531 after 300 times of iteration in simulator. This shows that the balanced position of kite is also its stable point and the stable point itself is an attractor, i.e., the system itself is stable.

4 Conclusion

We have developed an aerodynamic model with a constant problem solver to compute the system's balanced position directly by solving non-linear equations. Simulation, based on specifically chosen initial system parameter and cautiously designed characteristic function, of the launching process is performed to observe the kite's pattern of behavior when taking off.

The simulated process demonstrates that the kite is able to reach a stable point, after experiencing a rapidly attenuated oscillation. Comparison of the position of dynamic stable point and the result of the constant problem under the same condition shows no significant difference, implicating the system's balanced position is an "attractive" stable point.

References