摘 要

近年来空难事故的频繁发生，引起了人们的广泛关注，尤其是2014年3月8日失联的MH370，事故极其严重。但是事故已发生，239名遇难者的性命已无法挽回，至关重要的是如何从事故中得到教训。在多种空难事故研究方法中，通过得到飞机的残骸来辅助分析遇难时的情况已然成为研究热点。本文将在此背景下对空难事故研究方法展开以下研究，得到一个完整可靠的飞机空难事故仿真模型。

本文通过两个阶段建立总体仿真模型，第一阶段以飞机最后一次与地面通讯地点为基点，然后通过计算飞机的故障状态下的坠落轨迹，得到飞机的横向与纵向位移，第二阶段从坠机地点展开残骸漂移模拟，得到一定时间以后的飞机残骸位置。

在飞机坠落阶段，本文将从空气密度角度出发，计算飞机在坠落过程中的飞行阻力，飞行升力，再结合飞机的自身重力，进行力学分析，得到飞机的飞行受力情况，然后计算出飞机的加速度、速度和飞行轨迹，即可得到飞机的最终坠机地点。

在残骸漂移阶段，本文将在国际通用的风压差Leeway模型的基础上，再结合概率统计模型加以改进，得到本文所使用的残骸漂移模型。该模型将使用美国航空航天局（NASA）喷气推进实验室公布的风场与流场数据，计算每次更新时的残骸漂移速度，然后在概率模型的支持下合成出一个符合实际的漂移速度。最后根据仿真时间得到残骸的具体位置。

本文将采用OpenSceneGraph开源三维仿真引擎结合Qt界面开发框架，搭建出一个界面友好的海洋仿真环境，实时展示出上述的模型飞机在坠落阶段、残骸漂移阶段的具体情况。最后整合飞机在失联时的地点、坠机地点、残骸的最终地点，使用高斯投影反向计算得到残骸的最终经纬度。

关键词：MH370 飞行仿真 残骸漂移 Leeway OpenSceneGraph

**ABSTRACT**

　　The frequently occurring of air disasters in recent years has caught extensive attention. The missing MH370 in 2014-3-8 was especially serious. But 239 victims can't return to life now that disaster has happened. the most important is how to learn from the disasters. Among all the research methods used for air disasters, it has become a research hotspot to assist in the analysis of the wreckage of the plane. This paper will conduct the following research on the research methods of air crash accidents in this context to obtaine a complete and reliable simulation model of plane crash.

　　1. This paper establishes an overall simulation model through two stages. The first stage is based on the aircraft's last contact with the ground and then get the horizontal and longitudinal displacement of the aircraft by calculating the falling trajectory of the aircraft. In the second stage, wreckage drift simulation was carried out from the crash site, and the location of the wreckage was obtained after a certain time.

　　2. During the stage of falling, the paper calculates the resistance, lift from the Angle of air density, and then combines the gravity of the aircraft to carry out the mechanical analysis for obtaining the force of flight. Then the acceleration, speed and trajectory of the aircraft will be calculated and plane's final crash site can be obtained.

　　3. During the stage of wreckage drift, In this paper, the model of wreckage drift will be obtained which is improved by combining the probabilistic statistical model based on the international general Leeway model. The model will calculate the speed of wreckage drift at each update by using the wind and flow fields data that released by NASA's jet propulsion laboratory and synthesize an actual drift velocity with the support of the probabilistic model. Finally, the specific location of the wreckage is obtained according to the simulation time.

　　4. This paper will adopt the OpenSceneGraph open source 3d simulation engine and the Qt interface development framework to build an interface friendly ocean simulation environment and demonstrate the specific situation in real time of the above model aircraft in the falling stage and wreckage drift stage. Finally, the final latitude and longitude of the debris will be calculated by using gauss projection with the final location of the plane, the location of the crash site, and the final location of the wreckage.

**Keywords: MH370 Flight Simulation Wreckage Drift Leeway OpenSceneGraph**

目 录

[摘 要 1](#_Toc514607992)

[**ABSTRACT** 2](#_Toc514607993)

[第一章 绪论 6](#_Toc514607994)

[1.1 研究背景 6](#_Toc514607995)

[1.2 国内外研究现状 6](#_Toc514607996)

[1.1.1 国内研究现状 6](#_Toc514607997)

[1.1.2 国外研究现状 7](#_Toc514607998)

[1.3 研究内容 8](#_Toc514607999)

[1.4 论文结构 8](#_Toc514608000)

[第二章 相关理论原理 10](#_Toc514608001)

[2.1 飞机坠落过程相关理论 10](#_Toc514608002)

[2.1.1 飞机坠落过程建模 10](#_Toc514608003)

[2.1.2 空气密度理论及其计算方法 12](#_Toc514608004)

[2.1.3 飞机坠落时的受力分析 13](#_Toc514608005)

[2.2 残骸漂浮过程相关理论 13](#_Toc514608006)

[2.2.1 Leeway的引入 13](#_Toc514608007)

[2.2.2 基于leeway的动态漂移模型 16](#_Toc514608008)

[2.2.3 基于概率模型的Leeway模型 17](#_Toc514608009)

[第三章 系统结构设计 19](#_Toc514608010)

[3.1 系统开发环境介绍 19](#_Toc514608011)

[3.1.1 OSG引擎介绍 19](#_Toc514608012)

[3.1.2 Qt框架介绍 20](#_Toc514608013)

[3.2 系统可行性分析 21](#_Toc514608014)

[3.2.1 市场需求： 21](#_Toc514608015)

[3.2.2 技术支持： 21](#_Toc514608016)

[3.2.3 系统可靠： 21](#_Toc514608017)

[3.3 系统总体结构设计 22](#_Toc514608018)

[3.3.1 需求分析 22](#_Toc514608019)

[3.3.2 概要设计 22](#_Toc514608020)

[3.4 飞机坠落过程结构设计 24](#_Toc514608021)

[3.4.1 飞机坠落计算模块设计 24](#_Toc514608022)

[3.4.2 外部控制模块 26](#_Toc514608023)

[3.4.3 核心模块 27](#_Toc514608024)

[3.5 残骸漂浮过程结构设计 28](#_Toc514608025)

[3.5.1 残害漂流计算模块 28](#_Toc514608026)

[3.5.2 外部控制模块 32](#_Toc514608027)

[3.6 外围辅助系统设计 34](#_Toc514608028)

[3.6.1 下载模块 34](#_Toc514608029)

[3.6.2 分析模块 34](#_Toc514608030)

[3.6.3 位置合成模块 35](#_Toc514608031)

[第四章 系统实现 36](#_Toc514608032)

[4.1 仿真系统总体结构搭建 36](#_Toc514608033)

[4.2 飞机坠落过程实现 36](#_Toc514608034)

[4.3 残骸漂浮过程实现 37](#_Toc514608035)

[4.4 外围辅助系统实现 39](#_Toc514608036)

[4.4.1 下载模块实现 39](#_Toc514608037)

[4.4.2 分析模块实现 40](#_Toc514608038)

[4.4.3 位置合成模块 40](#_Toc514608039)

[4.5 系统展示 40](#_Toc514608040)

[4.6 系统功能测试 45](#_Toc514608041)

[第五章 结论与展望 48](#_Toc514608042)

[5.1 结论 48](#_Toc514608043)

[5.2 展望 48](#_Toc514608044)

[致谢 50](#_Toc514608045)

[参考文献 51](#_Toc514608046)

第一章 绪论

# 研究背景

近年来随着科学技术的不断发展以及机械制造航空航天等行业的不断向前推进，飞机等交通工具开始平民化走向大众的生活。但是飞机作为一种便民的交通工具，以具有高速、高空、高不可抗性为特点。虽然飞机的事故率远低于其他交通工具，但是其一旦升空，在空中是很难保证安全的，事故若是发生，乘客的生命财产安全基本不可能可以得到保障，机毁人毁。

2014年3月8日凌晨2点40分，马来西亚航空公司称一架载有227名乘客和12名机组人员的飞机（马航MH370），原定于从吉隆坡出发准备飞往我国首都北京，但是中途与管制中心失去联系。北京时间2014年3月24日晚，马来西亚总理宣布了MH370的消息称其在南印度洋坠毁，机上无一人生还。

上面的消息无疑是痛心的，所以基于避免此类事故再次发生的考虑，人们自然而然的会想到是要找到坠毁的飞机，分析飞机在坠毁时的情况，来找到飞行事故解决方案和飞机的改进办法，期望使愈来愈频繁使用的飞机变得更加安全，来保证乘客乘坐飞机的短短几个小时可以得到足够的保障。

目前主要的事故调查手段是找到黑匣子，黑匣子作为一种专用于飞机的飞行电子记录仪，它可以把飞机在失事前30分钟内飞行员之间的语音对话和两个小时以内的飞行参数记录下来。这些重要的数据可用于飞机失事后的事故分析之用。

但是黑匣子可以记录数据的时间是有限的，往往只有短短的30到40天时间，由于类似于MH370这样的空难事件多发于海上，黑匣子的掉落地点充满了各种干扰因素，而且深不见底的海水给黑匣子的寻找带来了极大的困难。所以黑匣子这种空难分析手段的具有一定的局限性。

考虑到上面的情况，飞行事故的发生原因只能求助于找到飞机的残骸，找到飞机的残骸就可以分析飞机的空难机械因素，这种分析手段在环球航空800号班机空难事故中得到应用。由于飞机的残骸往往具有漂浮性，所以寻找飞机的残骸就类似于分析海上漂浮物的漂流轨迹。

综合考虑以上的背景，本文将从飞机空难事故的残骸寻找手段出发，分析研究出一种实用的残骸搜寻模型。并且使用OSG三维仿真引擎和Qt界面框架结合搭建一个三维动态的海洋仿真环境，来实现对得到的搜寻模型的仿真验证。

# 国内外研究现状



## 国内研究现状

以往的SAR（搜索和救援）预测算法主要有两种，一种是GDOC算法，这是一种基于传统公式的算法，形成的是一种以沿着风向两端的最大风压角得到的直线中点为正中心，一定的半径形成的扇形漂流区域，这种模型得到的区域并没有考虑不确定性因素，并且错误的只考虑了风压作用，洋流压力等其他因素未纳入考虑。并且这种算法还有一点不足，海洋上的风与流是时刻变化的，但是这种算法却只考虑平均状况，这显然是不充分的，所以漂浮物位于计算出的区域中的概率不能得到保证。这是一种最原始的模型，另一种模型是CASP模型，这种模型克服了前一种模型未考虑不确定因素的缺点，使用了蒙特卡洛（Monte-Carlo）方法，但是这种算法仍旧没有考虑洋流的作用，只是简单的考虑了风压作用。

一种基于Monte-Carlo统计实验方法的海上漂流模型综合考虑了海流和风场等因素，并且克服了GDOC的只考虑平均情况的不足，其首要条件就是具备海洋上的实时数据，只有使用完善且准确的海洋数据去实时的更新模型状态，才能够保证模型的正确性[[[1]](#endnote-1)]。

陈达森等人采用等步差分网格分割法和追赶法求解ADI方法得到的离散方程组，模拟得到了湛江某海域的三种流场[[[2]](#endnote-2)]。

肖文军等人在上海沿海采用WRF和FVCOM以及基于Monte-Carlo理论的Leeway模型建立起长江口附近海域漂浮物预测模型系统[[[3]](#endnote-3)]。

另外一种基于FSA理论和贝叶斯网络的技术提出被应用于长江口附近航行安全性的提升[[[4]](#endnote-4)]。

但是以上的研究都是局限于近海模型建立，对于在远海海域飞机残骸的轨迹预测模型却很少涉及。

## 国外研究现状

　　国外对于海上漂浮物的研究比国内进行的更加的早且全面，最早且广泛使用的Leeway模型就是由Hodgins 和 Mak提出[[[5]](#endnote-5)]，Allen 和Plourde1改进，利用这个模型可以展开高精度的模拟。

　　但是Breivik和Allen在一份研究中表明高精度的模型并不能大幅度的提高轨迹预测的精准度[[[6]](#endnote-6)]。鉴于上述原因，Allen提出了Jibe的概念——漂浮物可能由于其自身的不对称轮廓改变其方向[[[7]](#endnote-7)]。

　　另外海洋环境对动态漂移的影响也是不确定的。一般情况下，风压、洋流、波浪对不同的物体的受力是不同的，由此基于观测和随机理论的方法得以被提出并广泛用于预测Leeway运动。例如， Brushett在一份研究中表明，对于双桨小船和乘坐一个人的小船，使用Leeway模型计算出的速度大约是风速的2.4-4.24%[[[8]](#endnote-8)]。此外，Breivik还发现了10m近地面风速和Leeway模型下的风速之间具有很强的线性关系[[[9]](#endnote-9)]。Gayer指出漂浮物的Leeway模型速度大致与其自身的浸润比（物体在海水之上的比例与物体在海水之下的比例之比）称反比[[[10]](#endnote-10)]，这一结论具有及其重要意义。另外一项有广泛应用价值的报告中总结 了63类不同物体的Leeway数据1。

　　除了上述两种因素带来的不确定性，风场、流场以及波浪数据会导致计算的不准确。考虑到这种情况，很多先进的方法已经被采用来提高这些数据在预测时的可信度。例如高精度雷达就可以用于收集更加精准的洋流数据[[[11]](#endnote-11)]。Özgökmen等人还考虑了流场的时空相关性[[[12]](#endnote-12)]。在这些方法的推动下，高精度的流场和风场等数据的获得就有了可能性。

　　在有效利用这些数据方面，相关的研究大都采用基于概率统计方法的Leeway模型，这种模型由于可以很好的描述物体的漂移轨迹的不确定性而被广泛采用。Davidson在他的一片文献中给出了一份概述[[[13]](#endnote-13)]。Özgökmen在使用Lagrangian方法做路径预测时，考虑到风场和流场数据由于掺杂着噪声数据（不可用数据），使用了卡尔曼滤波的方法进行了数据去噪12。Mínguez所展示的Monte-Carlo路径模拟算法中，上诉的不确定性因素都通过概率分布的方式所呈现出来，其中的参数都是使用漂浮物的测试数据[[[14]](#endnote-14)]。

　　基于上诉这些模型，本文使用就是一种基于Leeway模型的概率模型，这种模型首先由Jinfen Zhang等人提出[[[15]](#endnote-15)]，本文在该模型上对其空间相关性计算算法做了改进。

# 研究内容

本文所要研究实现的内容是一个海洋环境下的飞机坠落和轨迹预测的模型，该模型可以用于海洋环境下的失事飞机残骸搜寻。另外本文将基于该模型使用OpenSceneGraph和Qt界面框架搭建海上实时仿真环境，实现出一款预测模拟系统，该系统将使用NASA喷气推进实验室提供的风场与流场数据进行预测，得到残骸的漂移路径以及最终位置。因此研究内容包括以下几个方面：

给出本文使用的飞机坠落模型，解释其中所使用的计算算法，以及所使用的算法涉及到的相关理论原理。

对最基本的Leeway模型进行研究，然后提出本文所使用的模型。解释模型中所使用的算法涉及到的相关的理论原理。

对本文所需要实现的系统进行可行性分析及系统分析，介绍系统所使用的开发环境，然后对系统进行模块划分。接下来将对讨论系统的具体实现过程，并展示系统完成后的情况，最后将进行系统测试，使用具体的数据去测试系统的可靠性，并验证提出的模型的正确性。

# 论文结构

本文将对章节进行如下划分：

第一章主要阐述研究背景，介绍总结国内外在本领域的研究现状，最后说明本文的研究内容。

第二章将会介绍本文的研究所划分的两个基本模块所涉及到的基本原理和理论。对坠落模块将解释坠落轨迹的计算算法，并重点解释其中飞机受力所涉及到的空气密度与高度相关性的基本理论知识。对残骸轨迹预测模块，将会重点解释Leeway模型的基本原理，然后引入本文使用的基本模型的来源，并提出改进后的算法。

第三章将进入到系统涉及，首先会对本系统所使用的基本开发环境（三维引擎和界面框架）进行对比性的介绍，在此基础上进行系统的可行性分析，分析之余将进行系统的结构设计和模块划分，并着力解释各模块内部的设计。

第四章将讨论系统的实现细节情况，首先介绍海洋环境的内核搭建即如何将OpenSceneGraph（以下简称OSG）引擎嵌入到Qt界面框架中。然后阐述各个模块的具体实现，接下来将会展示完成后的系统，最后进行系统的可靠性和正确性测试。

第二章 相关理论原理



# 飞机坠落过程相关理论

飞机的坠落过程是一个极其复杂的过程，涉及到很多的领域，例如控制工程，飞行器涉及与仿真等等。但是考虑到本文的目标主要针对飞机失事后残骸的搜寻，并且飞机残骸在海洋表面漂浮的位移要显著的大于飞机坠落事产生的位移。所以在本论文中将简化飞机坠落时候的过程。尽量使这个过程更加的利于系统的设计和实现。所以本节将从以下几方面进行论述。



## 飞机坠落过程建模

飞机在坠落的时候，本文中在建模时将会默认将飞机按以下情形对待：

1. 将其当做一个简单质点，不会因外界情况发生改变而产生形变，继而影响到力的产生条件。
2. 不考虑飞机的俯仰角，即飞机在飞行的过程之中，一直保持着速度方向与海平面平行。
3. 不考虑科氏力，科氏力是一种源于物体的惯性而产生的力，与离心力一样都是虚拟的。在旋转体系中，做直线运动的一个质点由于惯性想继续保持原有的运动方向。但是运动体系本身是旋转的，所以一段时间后质点的位置在发生变化后，假如按照旋转体系的视角，质点的位置相对于沿直线运动发生了一定的偏移。本文中将不会考虑这种情况。

在考虑到上述的假设以后，首先对飞机所受的力进行讲解。飞机受力比较复杂，但是在简化模型下，可以大致分为下述几类：

1. 升力 升力主要是由飞机的机翼产生，产生原理可以参考伯努利原理。伯努利原理清晰的解释了飞机在天空飞行时克服掉重力的原因。首先当飞机在引擎的推动下加速前进时，飞机的两翼由于其自身的构造在机翼的上下会产生速度差。在机翼的上侧，由于隆起所以会导致空气在流动时从上侧流过的气流需要经过的路程会大于工下册流过的气流需要经过的路程。在时间相等的情况下，路程越长，速度越快，从而导致气压越低，所以机翼下侧将会产生大于上侧的气压，纵而使得机翼下侧受力大于上侧受力，升力也就从此产生。但是在后人的研究中发现升力的产生并不是可以简单的用伯努利原理可以解释清楚的，但是本文重点并不是探讨这一点。所以使用伯努利原理简要的解释飞机升力是可行的。飞机升力的计算公式如下所示：

其中 Ft 表示升力，Cl 表示升力系数，该系数一般与阻力系数成比例，ρ 表示空气密度，v 表示气流相对速度，Sc 表示参考面积，一般取机翼面积为参考面积。但是 ρ 是不一定的，它将会随着海拔、温度等因素的变化而变化，下一节将会对空气密度做更深入的阐释。

1. 重力，重力是比较容易理解的，由于地球的引力，所以任何处于地面上且具有重量的物体都会产生一个垂直向下的重力，在重力加速度一定的情况下，重量越大，飞机所受的重力也就越大。这份重力是使飞机坠落的主要来源。重力的计算公式如下：

其中 m 表示物体的重量，g 是重力加速度，虽然重力加速度也会随着海拔的变化而变化，但是由于这个加速度是由万有引力产生的，而飞机飞行的高度并没有突破大气层，海拔的变化值还是相对较小，所以重力加速度的变化不会有很大的变化，这里将会假设 g 是一个常量。

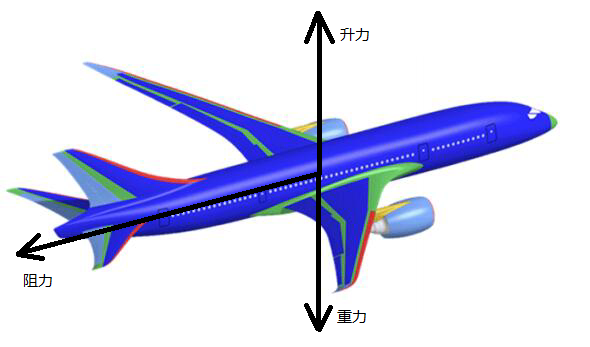
1. 阻力 飞机受到的阻力复杂程度是最大的。飞机在飞行过程中，其速度是十分大的，因此也会由于各种原因受到非常大的阻力，根据这些原因，大致可以将飞机受到的阻力分为摩擦阻力，压差阻力，诱导阻力，干扰阻力和激波阻力，由于篇幅原因，本文将对这些阻力做简要的介绍，具体细节不做深入讨论。
2. 摩擦阻力 摩擦阻力是在两个相对滑动的物体间产生的，当两个物体相对滑动时，就会产生一对相反的力，阻碍两个物体的相对运动。所以当飞机在空中飞行的时候，由于空气的粘性，气流通过飞机表面，空气微团与飞机表面发生了摩擦滞碍了气流的运动，因此也就产生了摩擦力。
3. 压差阻力。压差阻力产生的原因在于运动物体的前后存在压强差，这种由于压强差导致的阻力叫做压差阻力，其受到物体的迎风面积、形状和在气流中的位置影响。实验表明，最大迎风面积越大，压差阻力越大。
4. 诱导阻力。以上两种阻力在对飞行器的研究中合称为翼型阻力，飞机的机翼除了这两者之外，还存在一种诱导阻力，这种阻力是伴随着飞机的升力的产生而产生的。经由风洞实验发现，机翼在产生升力的同时，还产生了一个反作用力，这个作用力作用于气流上，导致气流从机翼的下侧绕过翼尖向机翼的上侧运动，从而产生了一个旋涡。在飞机向前飞行的同时，旋涡也随着向后移动，由此就产生了下洗速。旋涡的能量很大，将会带动周围的机翼下面的气流随着一起运动，因此就导致了来流方向朝向了斜下方，从而影响到升力的方向。原本升力应该垂直机翼向上，此时随着来流方向的改变，升力方向开始转变朝向了斜后方。如果将此时的升力分解，沿着飞机飞行方向的力就是诱导阻力，垂直向上的力才是真正的升力。
5. 干扰阻力。实验表明，飞机的各个部件在飞机飞行的过程中，所受到的阻力的总和并不等于飞机所受到的总阻力，而往往小于总阻力。其原因在于飞机的各个部件在飞行时，周围的气流相互干扰，因此产生了一种额外的阻力，这个额外的阻力称为干扰阻力。
6. 激波阻力，激波阻力当飞机由亚音速变化到超音速时产生的，飞机飞行的过程中，飞机前方的空气会压缩形成一种压力波，这种波在传播速度小于飞机时，后续压力和现有压力叠加致使空气遭到强烈干扰，阻力系数也会比0.8马赫时大四倍，激波也由此产生。但是民航飞机的速度一般低于音速，所以这种阻力不考虑。

综合上述的所有压力，下面给出压力的计算公式：

其中 Ff 表示飞机所受的总阻力，Cd 表示阻力系数，ρ 表示空气密度，v 表示气流相对速度，Sc表示参考面积，这里需要注意的是在计算阻力时仍然取机翼面积，而不是飞机的迎风面积。

1. 推动力，飞机的动力主要由引擎提供，飞机在巡航过程中，推动力朝向飞机飞行的方向。但是本文在做飞机坠落分析时，默认飞机引擎不再工作，失去动力，所以该力不再考虑。

在对飞机的各种受力情况做完大致介绍以后，下面给出飞机的受力情况：



## 空气密度理论及其计算方法

空气密度定义为单位体积内的地球大气的重量，其随着海拔的改变而改变，不仅如此，温度、湿度都对空气密度有显著的影响。

空气密度着一概念在很多的科学分支有着大量的应用，例如航空航天工程、重量分析、空调制造以及大气及气象研究。

根据测量工具以及上述的影响因素的的不同，空气密度会有不同的计算方法，大致可以分为以下几类：

1. 根据温度和压力进行计算，空气密度大致等于绝对压强与绝对温度和特定气体常数的乘积之比。公式如下：

其中表示空气密度，p 表示绝对压强，T 表示绝对温度，表示干燥空气的特定气体常数，也称为理想气体常数。

1. 潮湿空气的空气密度在计算时可以把它看作理想气体的混合。在这种情况下，水蒸气的分压被称作平衡蒸汽压，通过这种方法计算空气密度，当温度处于-10℃到50℃之间时，计算误差可以控制在0.2%之内。潮湿空气的密度计算公式如下：

其中 表示潮湿空气的空气密度， 表示干燥空气的分压，表示干燥空气的理想气体常数，T 表示温度， 表示水蒸气的气压， 表示水蒸气的理想气体常数， 表示干燥空气的摩尔质量， 表示水蒸气的摩尔质量，R 表示普适气体常数。

1. 由于空气密度会随着海拔的变化而发生变化，所以下面将给出使用海拔作为自由变量的计算公式。当然在得到最后的计算公式之前，需要几个额外的参数：

* p0 该参数表示海平面的标准大气压，值为101.325 kPa 。
* T0 该参数表示海平面的标准温度，值为288.15 K 。
* g 该参数表示地球表面的重力加速度，值为9.80665 m/s2 。
* L 该参数表示气温下降率，值为0.0065 K/m 。
* R 该参数表示理想气体常数，值为8.31447 J/( mol \* K ) 。
* M 该参数表示干燥空气的摩尔质量，值为0.289644 kg/mol 。

下面首先计算在海拔为h时的温度，在大气层内，计算公式大致可以用如下公式表示：

接下来计算海拔为h时的温度，计算公式如下：

最后根据理想气体定律给出空气密度的计算公式：

## 飞机坠落时的受力分析

本文在前面已经给出了飞机的受力情况，下面对飞机的受力进行分解。以飞机的重心为远点，向上为Y轴，飞机飞行的方向为X轴建立坐标系，具体可以参考前面章节中给出的图。

在水平方向上，飞机只受到阻力，所以有：

在垂直方向上，飞机受到重力和升力，飞机虽然失去动力，但是仍旧存在速度，所以升力还是存在，只不过升力大小小于重力，所以会下落。受力情况如下：

在分析完受力以后，就可以计算出飞机在各个时间点的速度，加速度，然后可以用积分法算出每个时间点飞机的具体高度。这样就可以实时的得到一条坠落轨迹，由这条轨迹线就可以模拟出飞机的坠落过程。具体细节在后面章节中详细讨论。

# 残骸漂浮过程相关理论



## Leeway的引入

现实生活中有很多海上的事故发生，这时就需要紧急的救援行动。例如，当一个海上船只由于某种原因而失去动力和通信信号，怎样去找到他们是一个很艰巨的任务，亦或是船只下沉，怎样去寻找救生筏上的遇险人员或者直接掉入海水里的人员？这些问题从二战时期就已经开始研究。

本文在研究时首先假设物体是漂流的，沉入海底的对象不做考虑。不同的漂流物体具有不同的漂流特征。这些影响漂流的性质主要有以下几方面：

1. 漂流物体的形状
2. 漂流物体的大小
3. 水面以上物体的比例
4. 水面以下物体的比例

海上物体的漂流主要由两中不同的力导致的，首先第一种主要来自于海流，这些海流主要包括以下几种：

1. 洋流
2. 风生流
3. 潮汐流
4. 其他海流

需要值得注意的是不同的海域有不同的海流，以下是说明：

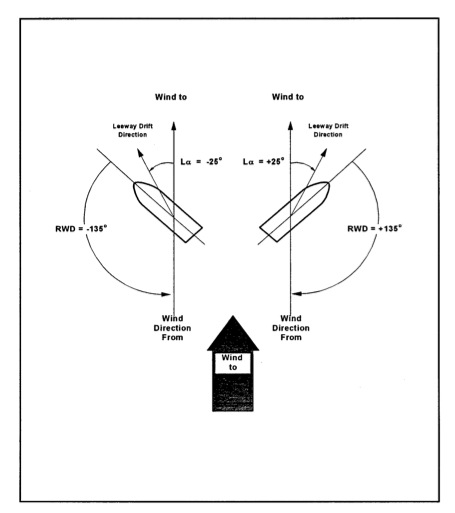
1. 潮汐流一般仅存在于离海岸线3海里以内的地方，特别是港湾等区域
2. 风生流主要存在于离海岸线20海里以内的地方，并且一般海水深度要大于30米
3. 洋流主要存在于离海岸线25海里以外的地方，海水深度超过100米

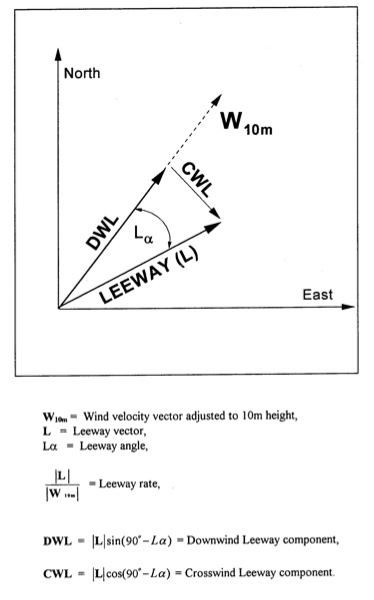
本文的主要研究的漂流对象是飞机残骸，这种情况应该对应于以上海流中的洋流。

另外一种受力是来自于Leeway，国际上对Leeway的定义比较多，其中维基百科给出的定义是由垂直物体运动方向的风失分量所引起的漂浮在水中的物体向一定方向运动的量。然而国际航空和海上搜救组织对Leeway的定义是由于吹向暴露于水面上的物体的风所引起的漂浮在水中的物体运动。虽然目前的研究结果表明水中物体的漂浮运动是由水流和风致作用导致的综合运动，在本文中也是如此考虑的。

在给出Leeway模型之前，首先给出Leeway相关的参数的定义，有关内容可以参考附图:

1. Leeway angle 这个角是指物体最终运动方向的角度减去风矢量的角度，如果物体向风矢量的右侧漂动，这个角的值就是正的，反之是负的。
2. Leeway Velocity 这个速率指的是Leeway速度的模值。Leeway速度一般指代的是物体漂流速度。
3. DownWind(DWL) and CrossWind(CWL) components of Leeway 其中DWL指的是物体漂流速度沿着风矢量方向的速度，CWL指的是垂直于风矢量方向的速度。
4. Leeway Rate 这个比例指的是物体漂流速率与风速率的比
5. Relative Wind Direction 相对风向指的是以漂流物体为参考原点并选取坐标轴来度量的风矢量方向





## 基于leeway的动态漂移模型

在上一节提到了物体漂移的受力主要是来自两方面的力，leeway和海流。下面提到的模型就是建立在这个基础之上。

当一个物体在海中由于风力和海流而漂浮的时候，在水线以下部分受到的是海流产生的力，在水线以上的部分受到的是风力。当然受到的力的大小和物体漂浮速度、风速、海流速度等有关系。但是由于很多研究都已经表明，物体在水中漂浮的时候，会很快的达到平衡，所以加速过程可以忽略不计。因此根据运动规律，当一个物体以一个相对稳定的速度漂动的时候，水线以下的海流产生的力和水线以上部分的风力应该大小相等，方向相反。在此基础上，动态漂移模型可以按如下方程表示出：

其中，CD 表示阻力系数，A、ρ 表示受力面积和密度，下标 W 和 C 分别表示 Wind 和 Current，即水线以上和水线以下部分。UW 表示风速，UB 表示物体漂流速度，UC 表示洋流速度。

以上的方程很显然与飞机坠落的阻力计算公式类似，即

在此基础上，假设 ，则以上公式可以简化为：

因此可以得到物体在稳定以后的速度为：

定义 ，很显然由上一节的介绍可以看出1-f的值就是leeway rate 。以上物体速度公式就可以表示为：

因此漂移物体在时间 K \* Δt 以后的位置可以表示为：

## 基于概率模型的Leeway模型

在上文的基础上，可以很清楚的发现漂浮物的漂移速度是和风速和流速有关的。假设风速和流速是不相关的，那么就有

所以漂移物的速度可以由风速和流速推导出，但是由于前文所述的动态模型会受到各类复杂的海洋环境影响，那么风速和流速的数据都是携带误差的，这些误差包含着随机性，因此轨迹预测需要以概率统计的方式进行。

在稳定的海洋环境中，海洋和大气环境都具有很强的空间相关性。所以在本文中，风场和流场都被假定的被描述为均匀随机场，也就是说两个位置的相关性只和他们的位置有关系。更进一步说就是风场和流场的平均值和方差都是定值而不和位置有关。Özgökmen提出的一种描述相关性的公式如下：12

其中Xi 和 Xj 是场中的两个点，则欧拉相关性可以描述为：

一个大的 R 值代表着对应的数据量在两个相邻的位置具有很高的空间相关性。

在拥有风场、流场以及空间相关性矩阵以后就可以对漂移物的速度做最优估计了，令为向量， 为漂浮物的随机位置， =是在t-1时刻围绕在平均值 周围的p-1个浮标的位置。必须注意的是 中的每一个值代表的位置相关性必须足够大。因此 的最优估计可以如下表示：

表示 的平均值， 是标准方差。

现在这个问题就可以看作从t-1到t更新漂浮物的位置。给出最优估计速度如下：

其中是物体的优化速度， 是平均速度， 是漂浮物体在t-1时刻速度的方差。

风速和流速都可以按如下公式表达：

浮物在时间t时候的速度可以被总结为：

由于在一个很短的时间里物体的速度可以看作一个常值，则动态模型也可以如下表达：

因此漂浮物在时间t时候的位置最优估计可以按如下公式计算：

第三章 系统结构设计



# 系统开发环境介绍

前面两章对本系统所需要的理论进行了简要的叙述，本章将会对本系统的结构设计过程进行完整的阐述。

首先是本系统的开发环境，在前文中已经有所提及，本系统是在WIN10操作系统中进行开发的，使用OSG引擎和Qt界面框架搭建的一款飞机坠落和残骸轨迹预测仿真系统。至于WIN10操作系统，不再赘述，下面主要针对OSG引擎和Qt开发框架简单介绍。



## OSG引擎介绍

1. OSG引擎介绍

OpenSceneGraph引擎是一款基于OpenGL库和标准C++实现的三维渲染引擎，由于其采用OpenThread库而具有了多线程渲染的特性。基于OSG引擎搭建的应用具有广泛的跨平台特性，可以运行在包括Windows、macOS、Linux、IRIX、Solaris以及FreeBSD等各类操作系统中，另外OSG引擎也有很强的移植性，可以针对例如IOS和Android等移动平台进行开发。

OSG引擎自从1998项目初始建立，在第二年就已开源并建立了自己的官方网站，第一个发行版本在2005年公布，后续的版本不断在前一版本的基础上更新补充。这款引擎现在已经很成熟,在科研、工业等众多领域例如可视化仿真、计算机游戏、虚拟现实、科学可视化以及建模等有着广泛的应用。这款引擎不仅具有良好的特性，这些特性将在下面具体介绍，另外这款引擎还提供了大量的例程、教学视频、参考数据、培训、论坛等技术支持，这对于一款引擎的广泛推广应用有着巨大的推动力。这两者促使了本系统采用这款引擎去搭建仿真界面。

1. OSG引擎的重要特性和框架
2. OSG引擎的重要特性

OSG引擎具有以下几点重要的特性：

**场景图** OSG引擎提供的场景图功能让众多的开发者从中受益，这种场景图十分方便图形图像的保存，也可以提供很高的性能，在仿真领域，对于性能的要求是很高的。场景图支持视图投影剔除、隐藏面剔除、小特性剔除、LOD、状态排序等。

**引用计数** OSG使用了智能指针，智能指针计数使得程序开发人员脱离了内存分配和回收过程，总所周知，C++的内存分配比其他语言更加复杂，所以OSG引擎提供的智能计数减少了内存泄露的情况发生。

**快速开发** OSG是基于OpenGL开发，所以包含了最新的OpenGL的底层功能，OSG提供了一整套补充程序库，使得程序开发人员可以集中精力去掌控这些补充功能来实现具体功能。

1. OSG引擎的框架

OSG引擎由以下几个重要的库组成：

**Core** 核心库是OSG引擎的基础，包括osg、OpenThread、osgUtil、osgDB。在核心库中包括了OSG引擎的基本类。例如向量、矩阵操作，相机，基本图元，图片等等。在其中实现了最核心的场景组织管理、图元操作以及外部导入导出等功能。

**osgViewer** osgViewer库实现了一个快捷的查看场景图的方案。并且它对各个操作系统平台是独立的。

**NodeKits** OSG还提供了一些工具库，其中包含了经常使用的高级3D应用程序组件和图形算法，这些库为一些共通的问题提供了解决方案。这些库有osgAnimation、osgFX、osgManipulator、osgParticle、osgQt、osgShadow、osgTerrain、osgText、osgVolume、osgWidget。

1. OSG引擎和其他常用的引擎的比较

目前经常使用的图形引擎有Unity、Unreal、CryEngine、OGRE。但是OGRE的使用频率近年来没有前三者高。下面对这几款引擎简单介绍一下

**Unity** Unity它是一个可以让玩家轻松创建自己想要的综合游戏开发工具。在三维视频游戏、建筑可视化、实时三维动画等方面有着广泛的应用。作为一个全面整合的利用交互图形化开发环境开发的专业游戏引擎，它的作品可以发布到Windows、Mac、Wii、iPhone、WebGL以及Android等平台。

**Unreal** 虚幻引擎是目前世界知名授权最广的游戏引擎之一，占据了国内外商用游戏引擎的80%的份额。虚幻引擎采用了实时光线追踪、高动态光照等最新技术，并且它的运算速度惊人，再与nVidia公司配合，这款引擎的渲染效果可以比肩电影的离线渲染效果。

**CryEngine** CryEngine是德国一家公司开发的基于DirectX的游戏引擎。CryEngine集成了实时动态光照、延迟光照、动态软阴影等技术，属于引擎中和Unreal一样的佼佼者。

**OGRE** OGRE是一款开源的面向场景的游戏引擎，其具有非常灵活的特点，使得开发人员更加方便的利用硬件加速的技术开发3D图形程序。这款引擎几乎拥有了商业3D渲染引擎的所有特点甚至有过之无不及。

介绍完以上四款其他的引擎，很容易知道这几款引擎的共同之处在于同属于游戏引擎，所以在仿真这一领域使用其他专用的渲染引擎将会不失为一个更好的选择。所以本系统选择了使用OSG引擎进行搭建。

## Qt框架介绍

Qt框架是国内外使用最频繁的软件开发框架之一，由于其基于C++且跨平台。所以自从面世以来广受用户喜爱。Qt很容易扩展，真正的做到了组件编程。Qt不仅提供了一套很好的界面开发方案，并且还提供了很多的其他的扩展功能，从数据结构到算法等等。方便用户可以从一而终的使用Qt开发应用。

同类型的界面框架还有GTK、MFC。前者虽然在很多方面做得很好，但是由于OSG的支持性并没有Qt良好，所以本系统并没有使用GTK来实现。另外OSG引擎虽然支持了MFC，但是由于MFC的整体架构、代码编写等已经大大落后于时代，选择MFC去开发本系统并不是一个很理智的选择。

# 系统可行性分析



## 市场需求：

自从飞机这一交通工具投入使用以来，广泛存在一种错误的认识：飞机的事故率极低，所以乘坐飞机要比其他的交通工具更加安全。但是尽管如此，极易忽视的一点是飞机的事故率纵然很低，但是由于飞机的特点，导致飞机事故的生还率相对其他的交通工具来说也是很低的。特别是当飞机在高空中以极快的速度飞行的时候。乘客几乎没有生还的机会。

根据厦航安全教育培训系列教程，该教程中给出了七大类的空航事故，并给出了详细的安全事故原因及遇难人数等。从该教程中可以获取到一个很重要的信息，即飞机这种交通工具并不是一定安全的，需要不断的改进飞机的机体性能，提高相关人员的乘机和管理飞机起降的素质。

既然飞机的性能、事故应变能力都需要提高，就牵扯到飞机性能从何处提升、如何提高飞机应对复杂情形下的处理能力。目前对于失事飞机的信息获取主要从黑匣子和残骸获取，但是由于黑匣子在海洋环境下不容易找到且数据保存时间有限，所以对于海上残骸的搜寻意义很大。具有很大的前景意义。

## 技术支持：

上文已经对本系统的搭建环境做了简要的介绍，OSG引擎和Qt的相互结合在仿真这一领域被很多同领域的公司与团队采用，因此具有很多案例可供参考。并且在海洋环境的搭建，OSG提供了开源的OsgOcean方案可以直接参考，因此开发人员只需要对OSG引擎和Qt框架有一定的了解，并且可以实现两者的嵌套就可以进行下一步系统内部核心工作，而不用耗费精力在海洋环境的实施方案上。所以选择OSG引擎和Qt界面框架进行本系统的开发是合适的。

## 系统可靠：

系统中采用的数据是来自于NASA的喷气推进实验室公布的数据，在系统完成后，可以提供数据的分析功能，系统的用户只需要将下载的数据使用本系统提供的功能进行解析，就可以得到可信的风场和流场数据，并利用这些数据进行相关的实验。因此本系统实验得到的结果具有一定的可信度。

# 系统总体结构设计



## 需求分析

1. 需求说明

根据前文的叙述，目前已经对本系统的业务有了大致了解，下面对本系统所需要的一些功能做一些说明:用户需要一款软件实现从飞机出现故障开始，通过一系列的步骤可以在一段时间后找到飞机的残骸，使用这些残骸去做飞机的故障分析以找到更好的飞机设计方案，避免飞行事故再次发生。其中这些步骤包括飞机坠入海中的过程，飞机残骸漂流过程。在这些过程完成后，需要有一个合成的步骤得到残骸的最终经纬度。这些步骤需要在一个三维的海洋环境中实时的展示出来。除此之外，为了方便用户的使用，需要对用户下载的数据进行分析合成，得到系统所支持的数据类型。

1. 需求导出

根据需求说明提供的信息，可以初步导出系统的需求情况如下:

1. 搭建一个三维的海洋仿真环境
2. 模拟实现飞机的坠海过程
3. 模拟实现飞机残骸在海洋中漂流的过程
4. 提供分析等其他辅助功能
5. 显示出全局轨迹和轨迹线

## 概要设计

1. 模块划分

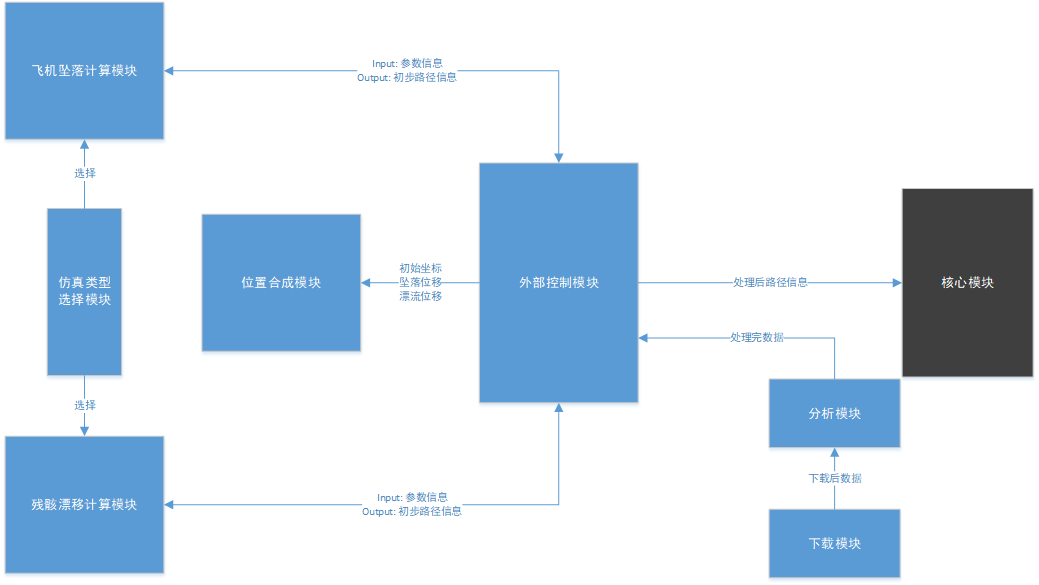
根据上面得到的需求，可以大致对系统划分为以下几个模块：

1. 核心模块，核心模块需要实现一个基本的系统仿真环境，即需要在其中实现一个海洋环境，另外需要向外部提供若干接口，用于外部与其进行数据交流，显示需要显示的内容等等。
2. 外部控制模块，这一模块主要对应于界面，当设计完界面以后，需要使用界面上的给个控件进行数据输入、数据交流、模块调用等等。
3. 仿真类型选择模块，这个模块可以选择坠海过程仿真还是残骸漂流仿真过程。
4. 飞机坠落计算模块，这个模块主要是利用外部控制模块输入的参数进行飞机坠落轨迹的计算，然后将轨迹使用核心模块进行显示。
5. 残害漂流计算模块，这个模块和坠落模块相似，都是利用参数进行内部计算，但是本模块最终得到的残害的漂流轨迹。
6. 下载模块，这个模块主要提供的是将用户提供的数据链接URL下载。
7. 分析模块，这个模块主要是将用户下载的数据进行解析，然后生成本系统所支持的数据格式，待到计算时利用。
8. 位置合成模块，这个模块主要是将用户计算好的数据进行综合计算，得到最终的残骸位置。
9. 接口设计

在完成对模块的划分以后，下面针对上述的模块分别进行详细的接口描述：

1. 核心模块 根据仿真类型选择模块的需要，核心模块需要提供坠海过程的场景、一个残害漂流的场景，这两个场景都需要在基础场景的基础上搭建，另外当两类计算模块完成计算以后会返回路径，核心模块需要接收这两种类型的路径，并让其中的节点显示出路径来。除了这两个接口外，由于创建HUD的需求以及显示轨迹线的需要，所以需要提供这两类接口。
2. 外部控制模块 外部控制模块需要提供很多的接口，因为有大量的参数需要通过主界面得到，事件的产生也是通过界面上的按钮进行，事件的响应也需要在该模块内部完成。所以总结一下有以下几种：向两类计算模块提供参数，响应仿真类型选择事件，响应各类控件的点击。
3. 仿真类型选择模块 该模块较易简单，通过产生点击事件与外部控制模块进行交流即可。
4. 飞机坠落计算模块 该模块是提供内部计算的一种，需要外部控制模块输入参数，这些参数输入同时就是接口，该模块最后也将提供一个接口返回在模块中根据这些参数计算产生的一条轨迹。
5. 残骸漂流计算模块 该模块是内部计算模块的第二部分，它也提供了相应的接口来接受外部模块传入的参数，然后进行内部的计算，最后通过接口返回一条轨迹线其中记录了残骸在大海中漂流的轨迹。
6. 下载模块 该模块是一个独立模块，通过外部控制模块调用出来，用于将用户输入的URL中的数据下载，供后期分析所用。这个模块的输入接口应该是一个控件，但是输出模块应该是输出下载好文件。参数合成为最终的经纬坐标。
7. 分析模块 该模块也是一个独立模块，通过将用户输入的数据分析出来供后期使用。模块的输入接口当然是用户在输入控件中输入的待分析的文件路径参数，输出接口是返回一个分析完以后的文件
8. 位置合成模块 该模块也是独立模块，输入接口是控件中输入的参数，输出接口是相应的控件中显示出计算好的经纬度。
9. 总体结构

下面是根据上述的结构设计和接口设计所得到的总体结构图，在图中可以看到本系统的总体结构划分，和个模块之间的数据沟通。在概要设计完成以后将会对系统各模块进行更加详细的设计，这些内容将在后面内容详细展开。



# 飞机坠落过程结构设计

飞机的坠落过程需要的模块主要包含以下几个模块：

1. 核心模块
2. 外部控制模块
3. 飞机坠落计算模块

下面针对这三个模块进行设计，当然核心模块和外部控制模块在残骸漂浮过程中也有重叠的部分，这里只对坠落过程中需要的内容进行论述。

首先由于这三个模块的数据流入虽然是从外部控制模块，即从输入参数开始，但是输入的参数类型都是按照坠落计算模块的需要进行设计的，所以首先对坠落计算模块进行设计。然后进行外部模块设计，最后进行核心模块设计。



## 飞机坠落计算模块设计

按照上一章对坠落过程的理论分析，可以很清楚的看出，坠落过程需要的参数有以下几种：

**初始位置** 初始位置的意义是飞机在失事时的坐标，该坐标是以笛卡尔三维坐标系为准，但是该坐标时常可以设置为X、Y坐标为0，只将Z坐标设置为飞机失事时的高度即可。

**初始速度** 初始速度是飞机失事时的速度，该速度也是以笛卡尔坐标系为准，由于飞机在失事时垂直方向的速度可以默认为0，所以该速度只需要设置一下X、Y坐标即可。

**飞机重量** 飞机的重量是一个重要的参数，用于计算飞机的重力和加速度。

**机翼面积** 机翼面积即计算阻力和升力时的参考面积，由于计算阻力时仍然以机翼为参考面积，所以不用另外设置飞机整体面积这一参数。

**升力阻力系数比** 这一参数用于升力和阻力计算时所用，当然阻力的计算只需要设置阻力系数，升力的计算需要将阻力系数乘以这一系数比才可以得到升力系数。

除了以上的几个重要的参数，在该模块中还需要有几个重要的常参数，这些参数在上述理论部分已经提及，即:

p0 该参数表示海平面的标准大气压

T0 该参数表示海平面的标准温度

g 该参数表示地球表面的重力加速度

L 该参数表示气温下降率

R 该参数表示理想气体常数

M 该参数表示干燥空气的摩尔质量

这些常参数的值已经在上文中给出。下面针对内部计算过程进行论述，首先若要得到一条轨迹，必须要得到这条轨迹线上的关键点。若要得到这些关键点避免不了的就是进行迭代计算，所以这些关键点中需要包含几个重要的信息，否则不能按照当前点计算下一个点的信息，这些信息是：速度、坐标、受力情况。只有知道当前的速度和坐标才能计算出当前的受力情况。相应的只有知道了受力情况，才可以计算出加速度，下一个关键点的速度和位置才能计算出来。

基于上述的论述，计算步骤可以这样设计：

1. 通过当前点(初始点)的速度和坐标计算出当前的受力情况
2. 通过当前的受力情况计算出(初始点)的加速度
3. 通过加速度计算出下一个关键点的速度
4. 通过下一个关键点的速度计算出下一个关键点的坐标

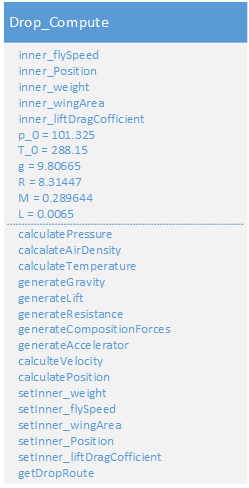
在前文的理论部分的基础上，可以将受力情况的计算步骤设计如下：

1. 通过输入当前点的高度，计算出当前点的气压P和气温T
2. 通过当前点的气压P和气温T计算出当前点的空气密度
3. 通过当前点的空气密度计算出当前点的阻力、升力
4. 结合阻力、升力和重力，计算出当前点的合力

基于上述的计算步骤，下面给出该模块需要提供的接口：

1. 计算气压和气温的接口
2. 计算空气密度的接口
3. 计算阻力、升力、重力的接口
4. 计算合力的接口
5. 计算加速度和速度的接口
6. 计算新的位置的接口
7. 返回轨迹的接口

综合上述的参数和计算步骤可以将该模块综合形成一个类，下面为该类的类图:



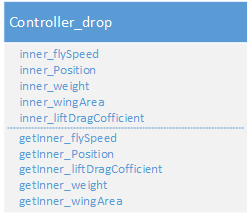
## 外部控制模块

外部控制模块在最终将会以一个界面形式展示，所以在其中针对飞机坠落过程必然嵌入了一个核心模块用于显示飞机的坠落过程，其嵌入方式在设计阶段不再阐述，细节将在实现时详细论述。

在仿真类型选择模块发送了一个坠落仿真的信息时，外部控制模块需要进入到坠落仿真阶段，然后将飞机坠落计算模块引入，此时需要衔接该计算模块需要的参数。所以在外部控制模块提供如下接口：

1. 飞机初始位置输入接口
2. 飞机初始速度输入接口
3. 飞机机翼面积输入接口
4. 飞机机身重量输入接口
5. 飞机升阻比系数输入接口

这些输入接口针对的是用户，即用户通过这些接口输入详细参数。现在将外部控制模块形成一个类，则对应飞机坠落计算模块这些接口就是内部参数了。所以该类的类图如下：



## 核心模块

核心模块是仿真显示最重要的模块，这个模块是将海洋环境展现出来的基础，这个模块将会嵌入到Qt界面中，直接以三维形式动态展现出来，因此这个模块中具有显示出这个海洋环境所需要的一切变量。另外由于这个模块需要根据外部控制模块提供的数据不断变化仿真的内容，所以需要有添加其他内容的接口。总结以上所有，归纳出这个模块的接口有如下：

1. 初始化基本海洋环境的接口

2. 初始化坠落环境的接口

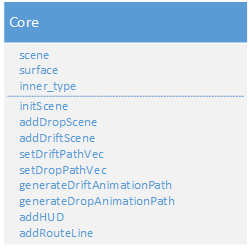
3. 初始化漂流环境的接口

4. 添加坠落轨迹的接口

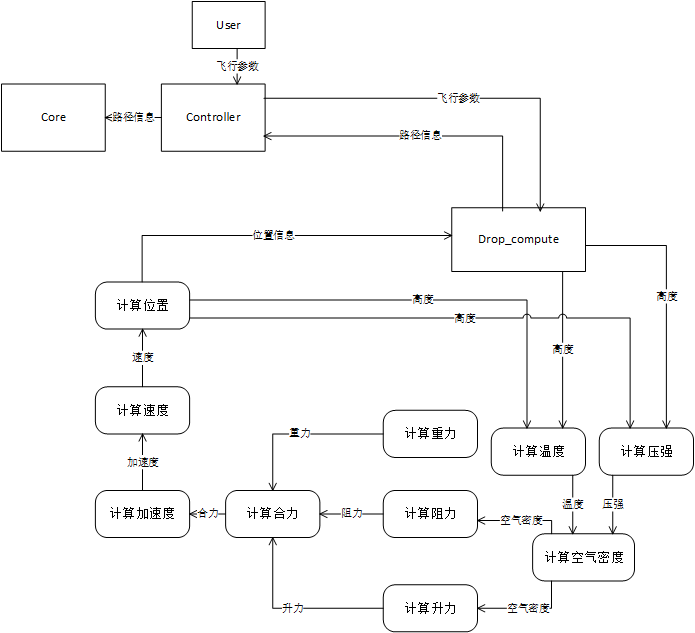
5. 添加漂流轨迹的接口

6. 其他接口

这一节将核心模块设计完成，下一节的重合部分不再详细设计。将这些内部参数和接口整合为一个类，所以该类的类图如下：



这一过程的数据流图如下所示：



# 残骸漂浮过程结构设计

残骸漂浮过程是本系统核心部分，该过程包括以下的几个模块：

1. 核心模块
2. 外部控制模块
3. 残害漂流计算模块

现在详细对这几个模块进行设计，由于前文已经说明该过程与前文的坠落过程的模块需求有重合，所以重合部分只针对本过程需要的内部参数和接口论述,和前一过程相同，本过程也从漂流计算模块进行设计。



## 残害漂流计算模块

本模块是所有模块中最复杂的一个模块，首先对系统内需要和外部控制模块进行数据沟通，对外需要和外部风流、洋流数据文件沟通。因此对这两部分进行划分。

下面先对内部进行设计，由上文所述的理论基础，本模块需要的内部变量有：

1. 初始位置 初始位置是飞机的坠机地点。该位置的坐标应按照笛卡尔坐标系为准，由于海平面的竖坐标基本为0，所以可以默认设置该坐标为二维坐标。
2. 初始时间 该变量表示飞机的坠毁日期，该变量可以用一个时间类型的变量表示出。
3. 初始风速和流速 初始风速和流速是在飞机坠机地点和时刻的风速和流速，该速度也默认为二维形式。
4. 风速方差和流速方差 这两个变量用于生成残骸附近点的随机风速和流速所用。这两个变量可以用百分比表示，用于表示风速可以在风速和流速的多大变化范围内。
5. leeway rate 这个变量的意义在前文中已经说明，该变量可以设置为一个小于1的小数。
6. 更新频率 该变量表示多久更新一下外部提供的数据。因为随着仿真的进行，所以数据也需要更新到最新的时间点上。该变量可以用一个整数表示用于表示多少天。
7. 仿真时间 该变量表示仿真所需要进行多少时间，可以使用一个整数表示进行的天数。
8. 虚拟浮标集合 这个变量中包含的是协助生成虚拟风速和流速的结构的集合，该集合可以用vector将这些结构整合在一起。
9. 虚拟浮标数量 该变量表示虚拟浮标集合的大小。可以通过一个整型表示出。
10. 空间相关性 该变量用于控制将没有很大的关系的虚拟浮标数据进行排除。该数据可以用一个小数表示。

下面对漂流计算的步骤进行简要的介绍。为了得到一条轨迹，和上文一样需要进行迭代计算。每一次迭代都需要在新的地点计算出该地点的残骸的位置和速度，用于下一个地点的计算。但是计算残骸速度需要计算残害所在地的风速和流速，然后通过理论中提供的公式计算出残骸的漂流速度。该步骤可以归纳如下：

1. 得到当前地点(初始点)的风速流速。
2. 通过得到的风速和流速，通过正态分布模拟，生成若干的虚拟浮标的风速流速。
3. 通过比较这些虚拟浮标的风速和流速的空间相关性，排除相关性较低的浮标数据，使用满足相关性的数据合成新的风速和流速。
4. 使用新的风速和流速合成漂浮物的漂浮速度。
5. 使用新的漂浮物速度计算经过一段时间后漂浮物的位置坐标。然后变换到这个坐标。

通过对过程的分析可以知道该模块需要提供接口有：

1. 得到风速和流速的接口
2. 合成新的风速和流速的接口
3. 合成漂浮物速度的接口
4. 计算新的位置的接口
5. 返回轨迹的接口

上述即为该模块面向内部的所有数据和接口,下面介绍面向外部的变量和接口。由于在海面上，所以可以看待为一个二维平面，如果要得到一个特定点的数据，势必要在一个二维网中进行。先将这个二维网定义为一个图结构，每当需要新的地点的数据时，就到这个图中进行插值查找，具体的查找算法实现过程中进行详细论述。在计算模块和这个图结构之间需要一个沟通的渠道，现将这个沟通的渠道定义为一个通道。每当需要新的数据时，通过这个通道到图中进行查找。所以将这个面向外部的部分构造为两个类。具体的变量和接口如下：

模块中需要定义的变量为一个通道对象

通道结构中需要的变量为一个图对象，需要的接口为图对象的setter接口和得到相应点的数据的接口

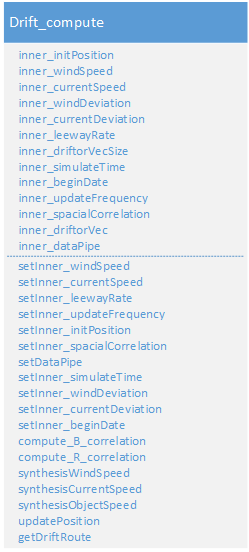
图结构中需要的变量如下：

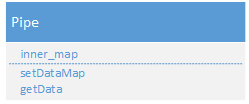
1. 可以使用的风场数据和流场数据文件集合 这两个集合中包含的是可以使用的数据文件的文件名集合，当需要使用时通过这些文件名结合下面的路径名进行查找然后读取。可以用列表包含一系列的字符串表示。
2. 数据文件的路径 包含数据文件的文件路径，可以用一个字符串表示。
3. 当前正在使用的风场数据文件名和流场数据文件名 这两个变量的意义很清楚，可以用两个字符串表示即可。
4. 包含风场数据节点和流场数据节点的集合 这些数据节点可以通过封装实现，每一个数据节点中包含的有这个节点坐标或这个节点的风场数据和流场数据。这个集合可以用vector表示。
5. 掩模长宽 掩模指的是一个mask，这个mask用于数据插值计算时缩小计算面积只用，当数据通道发送数据请求的时候，需要通过在图结构中插值计算出新请求点的风场和流场，为了使插值的范围更小，可以设置一个掩模，在这个掩模之内的数据节点才需要参与插值计算。这两个数据均只可以用整型表示即可。

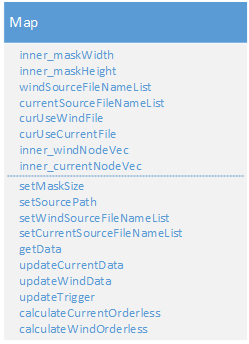
根据上述变量可以总结出图结构需要的接口如下：

1. 设置风场、流场数据文件集合的接口
2. 设置文件路径的接口
3. 设置掩模长宽的接口
4. 得到指定点的数据的接口
5. 更新风场、流场的数据接口
6. 其他接口

总结以上的变量和接口，可以将该模块设置为几个类，这些类的类图如下：







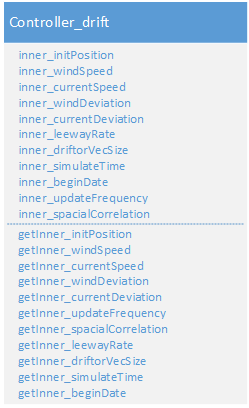
## 外部控制模块

上文已经对外部控制模块有了初步的介绍，本节接着从漂流轨迹过程的角度对该模块继续设计。

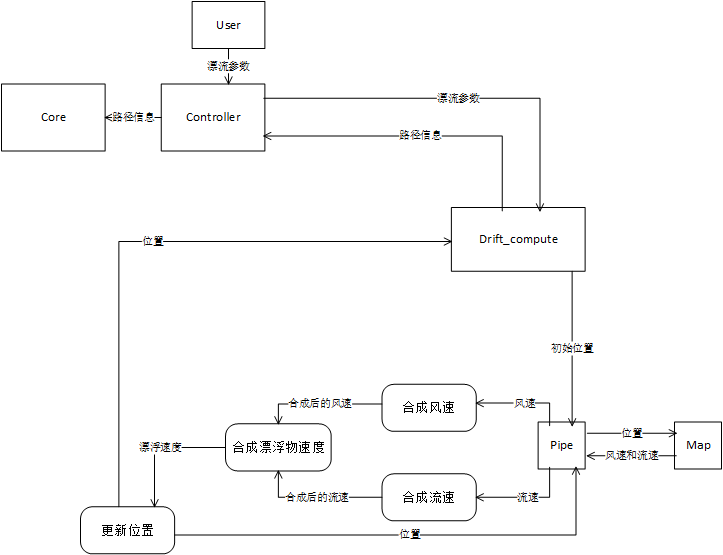
针对轨迹漂移计算模块提供的数据，外部控制模块需要提供以下的接口：

1. 初始位置输入接口
2. 初始风速和流速输入接口(针对简单的模拟过程)
3. leeway rate的输入接口
4. 仿真时间输入接口，在简单模拟中仿真时间对应秒，复杂模拟中对应天数。
5. 虚拟浮标数量输入接口
6. 风场和流场方差输入接口
7. 更新的频率输入接口
8. 空间相关性系数输入接口
9. 初始时间输入接口

结合上一节已经完成部分的外部控制模块的接口，下面给出外部控制模块的类图如下：



这一过程的数据流图如下所示：



# 外围辅助系统设计

外围辅助系统部分包括下载模块、分析模块、位置合成模块。下面对这三个模块进行详细设计。

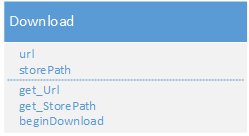


## 下载模块

下载模块提供的是供用户下载CDF文件的服务，所以很明显这个模块需要提供的接口有如下三点：

1. 存储位置的输入接口
2. 下载地址的输入接口
3. 下载接口

给出下载模块的类图如下：

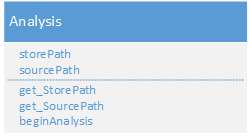


## 分析模块

分析模块提供给用户的是对CDF文件的分析服务，将内部的数据提取出，而不用用户自己去使用相应的SDK去解析。所以该模块需要提供的接口如下：

1. 待分析数据存储地址的输入接口
2. 分析结果存储地址的输入接口
3. 分析接口

给出分析模块的类图如下：

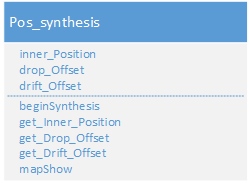


## 位置合成模块

位置模块是为用户方便得到最终残骸位置服务的，根据飞机失事的初始位置，飞机坠落位移、漂流位移计算出飞机最终位置。所以该模块需要提供的接口如下：

1. 初始位置的输入接口
2. 坠落位移的输入接口
3. 漂流位移的输入接口
4. 计算模块
5. 百度地图显示接口

下面是位置合成模块的类图：



第四章 系统实现



# 仿真系统总体结构搭建

本系统的搭建依赖于OSG和Qt，OSG中提供了OsgQt模块，这个模块中实现了如何将OSG的图形上下文嵌入到Qt中。

为了和Qt的特性相贴和，本系统采用实现一个类，继承自QtWidget和osgViewer，让这个类同时具有widget和viewer的特性，然后在这个类中实现海洋的初始环境。最后在Qt的界面中放入一个widget，利用Qt的提升 功能，将这个widget提升至构建的类。这样OSG的环境就可以无缝的衔接到Qt界面中，在需要调用这个类也就是上一章中提及的核心模块中的接口时，只需要使用界面ui提取即可。这种搭建方法即方便程序开发，又使核心 模块和外部控制模块实现了无缝链接。

# 飞机坠落过程实现

飞机坠落过程可以按照如下步骤实现：

1. 在外部控制模块中的接口中填入数据，这些接口可以用LineEdit实现。
2. 点击计算接口，这个接口使用Button实现，然后通过外部控制模块的LineEdit提取出相应的数据并转化为相应的格式。接下来将提取的数据通过飞机坠落计算模块的接收接口输入。
3. 开始计算，计算完毕后返回一个vector，其中包含的是坠落轨迹中的关键点。然后将这些关键点转化为OSG支持的三维数据格式，最后将这些数据通过核心模块的接收接口输入。
4. 在核心模块中，将上一步输入的关键点构造出一条动画路径，使飞机沿着这条动画路径坠落。

下面针对这些过程中较为复杂的部分详细介绍。

坠落轨迹计算算法 该算法将会利用输入的参数计算并返回坠落路径上的关键点。

**Algorithm:** Drop\_Route

**Input:** Initial Position, Initial Speed, Airplane Weight, Wing Area, Ratio of Drag and Lift.

**Output:** The Vector of Key Point.

**Drop\_Route**

Initialize a vector and append the Initial Position into the vector.

Pos ← Initial Position, Speed ← Initial Speed.

While Pos.z > 0 Do

　Generate Composition Forces Fc using Speed, Airplane Weight, Wing Area, Ratio of Drag and Lift.

　Generate Accelerator a using Fc and Weight

　Calculte Velocity v of next point using Speed and a, Speed ← v

　Calculate Position p of nect point using Pos and Speed, Pos ← p

　Append the Pos into Vector.

End

Return Vector

**Function:** Generate\_Composition\_Forces

**Input:** Position, Speed, Airplane Weight, Wing Area, Ratio of Drag and Lift.

**Output:** A Three Dimention Coordinates.

**Generate\_Composition\_Forces**

Compute the Air Density ρ using Position.z

Compute the Drag Force Fd using Speed, Wing Area, ρ Drag Cofficient

Compute the Lift Force Fl using Speed, Wing Area, ρ Drag Cofficient and Ratio of Drag and Lift

Compute the Gravity Fg using Weight

Compute and return the Composition Force Ff using Fd, Fl, Fg

# 残骸漂浮过程实现

在本系统中共实现了两种轨迹漂移形式，即简单模拟和复杂模拟，简单模拟用于短时间、风速和流速变动不大的情况下，复杂模拟适用于长时间、风速和流速会因时间变化而发生变化的情况下。下面分别对着两种情形的实现进行说明。

在系统中用于两种模拟形式变换是通过一个TabView实现的，两个TabBar分别对应简单和复杂模拟。

在简单模拟中只需要不需要设置风速流速的方差、不需要设置仿真的更新频率，不需要初始时间和位置。但是需要设置初始的风速和流速。实现步骤如下：

1. 在外部控制模块提供的接口中输入数据，这些数据通过LineEdit实现，风速和流速方向通过Dial实现。
2. 触发计算接口，这个接口时通过Button实现，当点击后需要将外部控制模块中的数据提取并转换为相应的格式。然后将转换后的数据输入到残骸漂流计算模块的输入接口中。
3. 当残骸漂流计算模块接收到外部数据后开始计算，然后返回一个包含了轨迹中关键点的vector给外部控制模块。外部控制模块完成转换，输入到核心模块的简单模拟的接口中。
4. 核心模块在接收到外部控制模块提供的轨迹数据后，生成一条动画路径，最后让残骸沿着这条轨迹动画漂移。

在具体步骤叙述完，接下来对简单模拟中较为复杂的算法进行讲解：

简单漂移轨迹计算算法 该算法将会利用输入的参数计算并返回漂移路径上的关键点。

**Algorithm:** Simple\_Drift\_Route

**Input:** Wind Speed, Current Speed, Simulation Time,Leeway Rate.

**Output:** The Vector of Key Point.

**Simple\_Drift\_Route**

Initialize a vector and append the Initial Position into the vector.

Synthesis Object Speed V using Wind Speed, Current Speed and Leeway Rate.

Compute the step using Simulation Time.

For i = 0, i < step. Do

　Compute a new Position p

　Append the p into the Vector

End

Return vector

在复杂模拟中不需要设置风速流速，所有的风速流速数据需要从数据文件中读取。但是其他的相关数据都需要填入。实现步骤如下：

1. 在外部控制模块的接口中填入相应数据，其中文件名、文件路径相关的接口通过FileDialog实现。文件名的显示通过TreeView实现。其余的数据接口使用LineEdit实现。
2. 当计算接口也就是Button被点击后，外部控制模块接口中的数据被提取出并完成转换，然后将这些转换后的数据输入到残骸轨迹计算模块提供的输入接口中。
3. 当残骸轨迹计算模块接收到数据后，不断从外部提取风场和流场数据然后进行计算并返回一条轨迹vector，外部控制模块将这个vector中的数据转换后输入到核心模块的复杂模拟接口中。
4. 当核心模块接收到外部控制模块的数据后将这些数据转换生成为动画路径，使残骸沿着动画路径漂移。

复杂模拟相对于简单模拟实现较为困难。其中的算法讲解如下：

复杂漂移轨迹计算算法 该算法将会利用输入的参数计算并返回漂移路径上的关键点。

**Algorithm:** Complex\_Drift\_Route

**Input:** Wind Data Source Path, Current Data Source Path, Wind Speed Deviation, Current Speed Deviation, Leeway Rate, Simulation Time, Update Frequency, Initial Position, Initial Date, Spacial Correlation Rate, Virtual Driftor Size.

**Output:** The Vector of Key Point.

**Complex\_Drift\_Route**

Initialize a vector and append the Initial Position into the vector, Pos ← Initial Position, Date ← Initial Date.

Compute the step using Simulation Time and Update Frequency.

For i = 0, i < step. Do

　Synthesis Object Speed V using Wind Data Source Path, Current Data Source Path, Wind Speed Deviation, Current Speed Deviation, Leeway Rate, Pos, Date.

　Compute the next position p and Pos ← p, append the Pos into vector.

　Date ← Date + Update Frequency.

End

Return vector

Algorithm: Synthesis\_Object\_Speed

Input: Wind Data Source Path, Current Data Source Path, Wind Speed Deviation, Current Speed Deviation, Leeway Rate, Pos, Date.

Output: Two Dimention Speed.

Synthesis\_Object\_Speed

Get wind speed data Vw and current speed data Vc from data Pipe using Wind Data Source Path, Current Data Source Path, Pos and Date.

Generate n virtual driftors, set each virtual driftor's wind speed and current speed which are generated by normal distribution and transformed by Box-Muller using Vw, Vc, Wind Speed Deviation, Current Speed Deviation.

Synthesis the wind speed Vw' and Vc' using the n virtual driftors.

Synthesis and return the object speed using Vw'、 Vc' and Leeway Rate.

**Algorithm:** Get\_Data

**Input:** Wind Data Source Path, Current Data Source Path, Pos, Date.

**Output:** A Struct include Wind Speed Data and Current Speed Data.

**Get\_Data**

If Date is out of date, Then

　Update the map data using Wind Data Source Path, Current Data Source Path and Data.

End If

Travel all the current and wind node, get all the node in the mask and the four node which is the nearest to the Pos in the four corner and the distance of the four node to the Pos.

Calculate the orderless of current and wind in the four corner.

Sysnthesis the factor using distance factor and the orderless factor.

Sysnthesis the final wind and current data using the data in the four corner nodes and factors.

Return the struct include wind and current data.

# 外围辅助系统实现



## 下载模块实现

本系统在实现下载模块时，采用了一个界面的形式。使用了QNetworkAccessManager实现了一个Http下载类，然后封装到下载模块中，使用FileDialog实现文件存储路径的选择。实现步骤如下：

1. 在下载模块中填入需要下载的CDF文件路径，这个接口通过LineEdit实现。通过ToolButton实现FileDialog的触发选择文件储存位置，并通过LineEdit显示。
2. 当下载接口触发，该接口通过Button实现，当按钮点击后，Http下载类使用填入的路径，下载文件。
3. 将下载后的文件储存在存储路径对应的位置。

## 分析模块实现

分析模块的实现同样是采用界面展现，使用TreeView将需要分析的文件名展示。下面给出分析模块的实现步骤：

1. 在下载模块的接口中填入待分析文件路径、分析完后的文件存储路径参数。这两个都通过FileDialog实现。
2. 当使用Button实现的分析接口点击后，分析模块从待分析文件路径中加载出可用文件。然后通过TreeView展示出来。
3. 用户通过单选或者多选的方式将需要分析的文件选择。
4. 底层使用CDF文件的SDK将需要分析的文件分析出。
5. 将分析结果存储在填入的存储位置。

## 位置合成模块

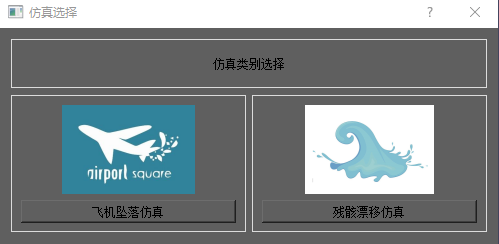
位置合成模块依旧采用界面展示。如下是该模块的实现步骤：

1. 在下载模块对应位置中填入飞机失事位置经纬度参数，飞机坠落位移参数、残骸漂流位移参数。这些都通过LineEdit实现。
2. 当使用Button实现的合成接口点击后，分析模块读取经纬度参数并做相应的转换，然后通过高斯正向投影，将经纬度转换为二维笛卡尔坐标。
3. 将转化后的二维坐标和分析模块读取的飞机坠落位移参数、残骸漂流位移参数相加得到最终的二维笛卡尔坐标系下飞机残骸位置。
4. 将最终的二维笛卡尔坐标系下飞机残骸位置通过高斯反向投影转化成经纬度。
5. 如果通过Button实现的地图显示接口点击后，将转化后的经纬度在百度地图上显示。

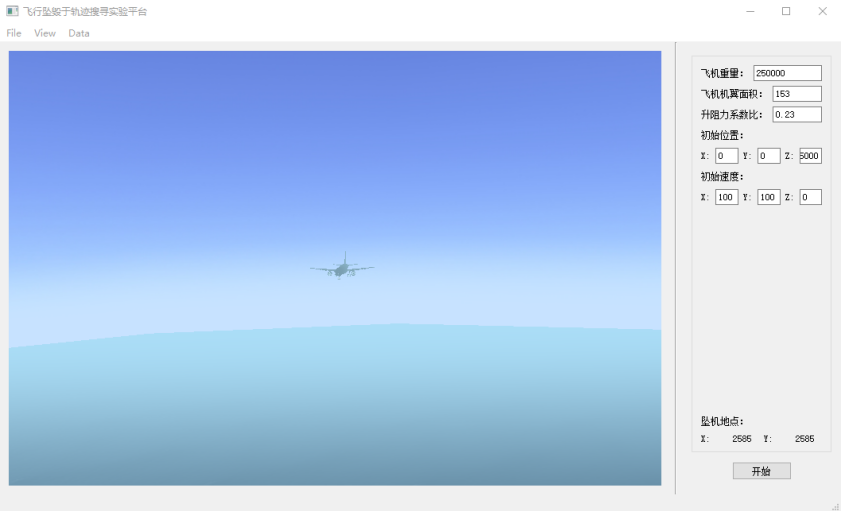
# 系统展示

由于本系统对整个仿真过程划分为三个步骤，所以以下将会按照大致的步骤顺序对系统进行展示。

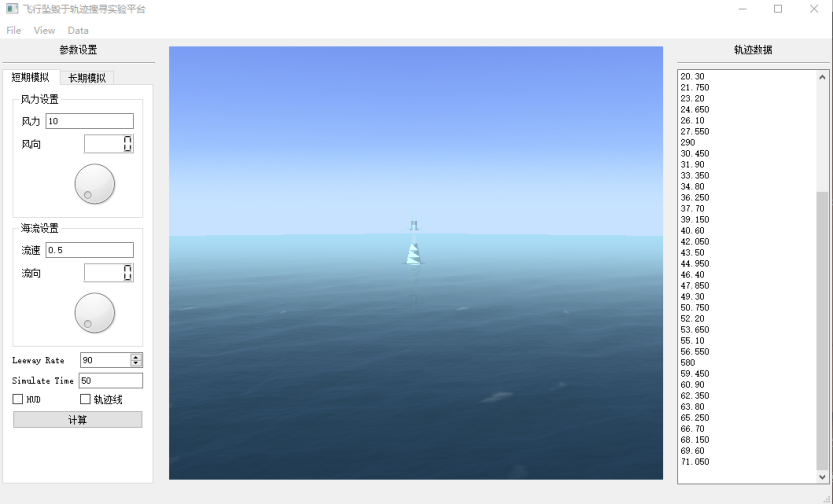
首先选择仿真类型：



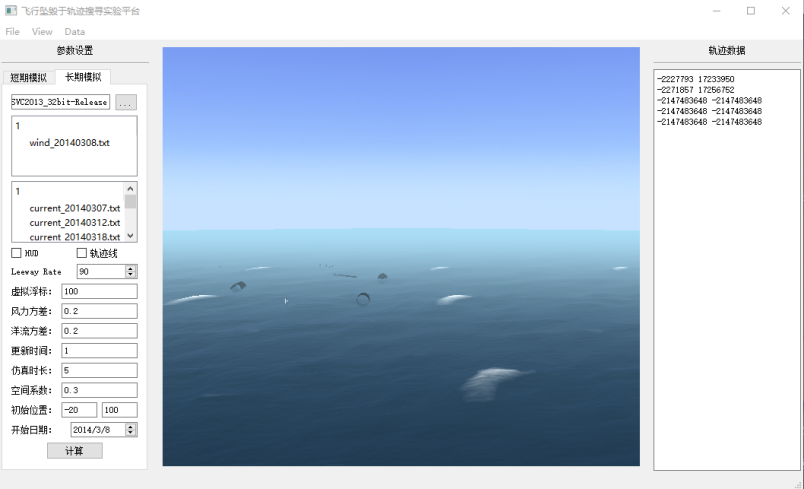
当选择飞机坠落仿真后，填好数据，飞机在坠落过程时候的情况如下：



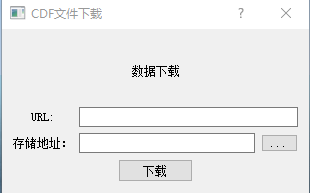
当选择残骸漂移仿真后，选择简单模拟，填选相关参数，简单的浮标漂移情况如下：



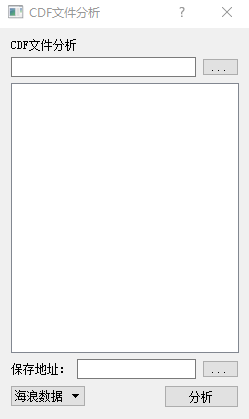
当选择复杂模拟，选择好数据文件，填好参数后，复杂的残骸轨迹漂移情况如下：



下面是下载界面的展示：



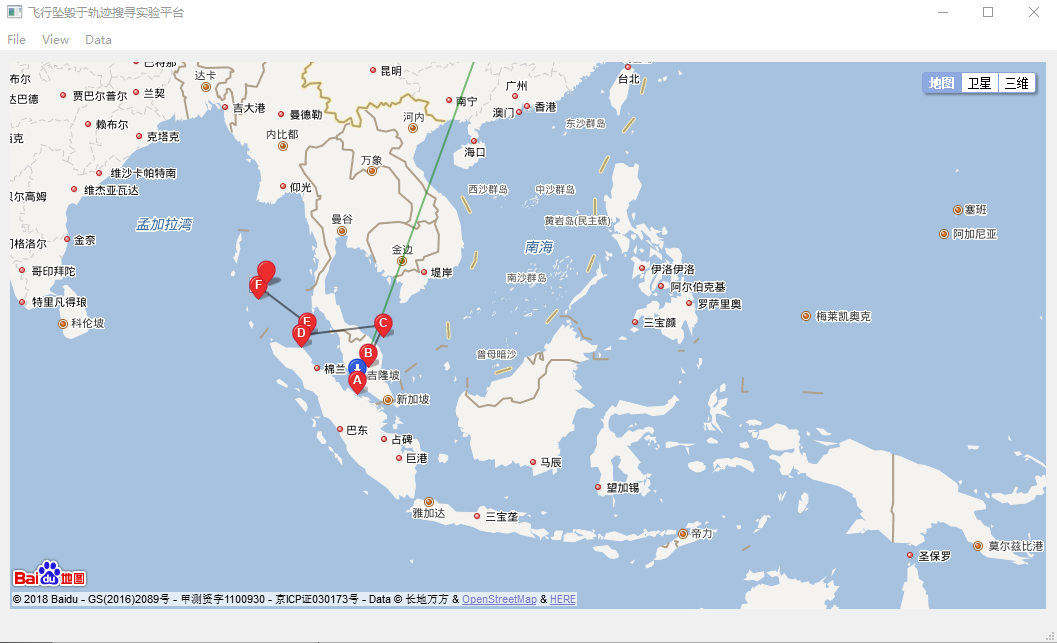
下面是文件分析界面的展示：



下面是位置合成界面的展示：



当位置合成完毕，点击地图显示，将会在百度地图中显示出来：



# 系统功能测试

对系统功能的测试，主要有对飞机坠落过程的测试和残骸漂移过程的测试以及附加功能的测试。

## 飞机坠落功能测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 预期结果 | 测试结果 | 测试状态 | 错误类型 |
| Boeing B727-200 Wing area:153 Lift-Drag ratio:16.4 Height:10000 | | | | | |
| 1 | Weight: 44330  Speed:(150,150) | (5400,5400) | (5417,5417) | 与预期结果相符 |  |
| 2 | Weight: 44330  Speed:(196,196) | (6900,6900) | (6908,6908) | 与预期结果相符 |  |
| 3 | Weight:78100  Speed:(150,150) | (5550,5550) | (5595,5595) | 与预期结果相符 |  |
| 4 | Weight:78100  Speed:(196,196) | (7100,7100) | (7149,7149) | 与预期结果相符 |  |
| Boeing 707-320 Wing area:283 Lift-Drag ratio:19.4 Height:10000 | | | | | |
| 1 | Weight: 67.495  Speed:(150,150) | (5300，5300) | (5355,5355) | 与预期结果相符 |  |
| 2 | Weight: 67.495  Speed:(198,198) | (6850,6850) | (6886,6886) | 与预期结果相符 |  |
| 3 | Weight: 151.381  Speed:(150,150) | (5600,5600) | (5610,5610) | 与预期结果相符 |  |
| 4 | Weight: 151.381  Speed:(198,198) | (7200,7200) | (7235,7235) | 与预期结果相符 |  |
| Boeing 747-100 Wing area:511 Lift-Drag ratio:17.7 Height:10000 | | | | | |
| 1 | Weight:162500  Speed:(150,150) | (5410,5410) | (5447,5447) | 与预期结果相符 |  |
| 2 | Weight:162500  Speed:(190,190) | (6700,6700) | (6756,6756) | 与预期结果相符 |  |
| 3 | Weight:322100  Speed:(150,150) | (5650,5650) | (5660,5660) | 与预期结果相符 |  |
| 4 | Weight:322100  Speed:(190,190) | (7000,7000) | (7035,7035) | 与预期结果相符 |  |
| Boeing 767-200 Wing area:283,3  Lift-Drag ratio:19.0 Height:10000 | | | | | |
| 1 | Weight: 80.286  Speed:(150,150) | (5400,5400) | (5410,5410) | 与预期结果相符 |  |
| 2 | Weight: 80.286  Speed:(167,167) | (5900,5900) | (5968,5968) | 与预期结果相符 |  |
| 3 | Weight: 136.078  Speed:(150,150) | (5550,5550) | (5576,5576) | 与预期结果相符 |  |
| 4 | Weight: 136.078  Speed:(167,167) | (6100,6100) | (6155,6155) | 与预期结果相符 |  |

## 残骸漂移功能测试

简易漂浮测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 预期结果 | 测试结果 | 测试状态 | 错误类型 |
|  | | | | | |
|  |  |  |  |  |  |

复杂漂浮测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 附加功能测试

附加功能有下载功能、数据分析功能和位置合成测试。前两者不需要进行测试，下面是针对位置合成功能的测试：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 测试用例 | 预期结果 | 测试结果 | 测试状态 | 错误类型 |
| 初始经纬度（0，0） | | | | | | |
| 1 | 坠落偏移：  (5000,5000)  漂浮偏移：  (-10000,-10000) | | (-0.428  ,-0.024) | (-0.428147  ,-0.0242566) | 与预期结果相符 |  |
| 2 | 坠落偏移：  (5000,5000)  漂浮偏移：  (10000,-10000) | | (-0.428,  0.0801) | (-0.428144,  0.0807697) | 与预期结果相符 |  |
| 3 | 坠落偏移：  (5000,5000)  漂浮偏移：  (10000,10000) | | (0.080,  0.081) | (0.0804415,  0.0807616) | 与预期结果相符 |  |
| 4 | 坠落偏移：  (5000,5000)  漂浮偏移：  (-10000,10000) | | (0.080,  -0.024) | (0.0804387,  -0.0242539) | 与预期结果相符 |  |
| 5 | 坠落偏移：  (-1000,-1000)  漂浮偏移：  (10000,10000) | | (0.045,  0.045) | (0.0450649,  0.0452584) | 与预期结果相符 |  |
| 6 | 坠落偏移：  (1000,-1000)  漂浮偏移：  (10000,10000) | | (0.045,  0.056) | (0.0450654,  0.0557603) | 与预期结果相符 |  |
| 7 | 坠落偏移：  (1000,1000)  漂浮偏移：  (10000,10000) | | (0.056,  0.056) | (0.0555237,  0.0557597) | 与预期结果相符 |  |
| 8 | 坠落偏移：  (-1000,-1000)  漂浮偏移：  (10000,10000) | | (0.056,  0.045) | (0.0555232,  0.0452579) | 与预期结果相符 |  |

第五章 结论与展望



# 结论

在飞机这种交通工具使用频率日趋频繁后，近年来空难事故的发生频率也开始增长。找到事故发生原因并透过原因切实的改进飞机开始成为了研究的热门。但是从黑匣子出手有时候会受到这种方式的约束。如何找到飞机的残骸利用残骸来还原事故发生场景变成一种可行的方法。

本文在前人的研究基础上，提出了使用三步走的方法来搜寻残骸，首先从飞机坠落这一角度出发找到飞机坠落点，然后通过海洋漂浮物理论进行飞机残骸轨迹计算。最后通过叠加计算找到飞机残骸的最终地点。

本文在第二章中详细的对飞机的坠落和残骸漂浮理论进行了讨论，通过计算空气密度得到飞机的升力、阻力等，然后通过受力分析得到飞机的各阶段的受力情况，算出飞机的加速度、速度和高度。最后得到飞机的坠毁轨迹和坠机点。另外又从多方面对海洋漂浮物理论进行了详细介绍，对比分析了多个现有模型，最后引申出本文所使用的模型并提出改进方法。

由于这些理论需要转化为一个仿真模型，第三章详细的对需要实现的这个系统进行了需求分析概要设计和详细的设计。在第四章中实现了所有的设计并对实现过程中复杂的算法进行了详细叙述。在实现基础上本文给出了完成后的系统的展示并做了测试案例。

# 展望

本文对飞机残骸的搜寻进行了研究，在前人的基础上提出了新的残骸定位方法。将残骸的定位分为三步，首先需要从飞机失事点开始，展开飞机的坠落研究，这对飞机的坠机点的确认有极大的意义。然后从飞机的坠机点开始展开飞机的残骸的轨迹模拟，在使用海上风场和流场数据的基础上，研究了leeway模型，再加上概率统计模型提出了一种新型的轨迹模型。最后综合飞机坠落、残骸漂移得到飞机的最终位置。然而在这个定位方法中，仍旧存在着很多的不足。以下几方面在未来的研究中需要加以考虑和改善。

1. 飞机的失事点确认，有一个更准确的飞机失事点对于飞机的残骸搜寻有很大帮助，假如有一个准确的失事点，那飞机的坠机点就可以更精确，残骸模拟初始情形就可以更精确。不同的坠机点的初始风场、流场存在着很大的差异。
2. 飞机的坠落模型需要改善，本文所使用的方法是最简单的抛物模型，未来的研究中对于飞机的受力分析可以更加完善。
3. 飞机的坠落原因在本文中只是考虑飞机自然坠落，也就是失去动力的情形。未来可以从飞机的机械因素、天气因素等多方面加以考虑。
4. 对于随机因素对残骸轨迹的影响需要加强，而不仅仅只从简单的正态分布进行随机因素研究。

致谢

　　时光飞逝，转眼从进入校门到现在已经过去将近四年，在这四年里，我成长了很多，从一个懵懂的小子，变成可以独当一面的成人。除了在心智上的发展，我在自己喜欢的领域也有很大的进步，对计算机有了总体的了解，但是时间是有限的，未来的发展还需要靠我自己，母校已经给了我太多太多。在这即将毕业的半年里，我综合所有我所学的完成了毕业设计，但是这份毕业设计不仅仅是我个人的，还凝结了众多老师、学长学姐的心血，因此我由衷的感谢每一个帮助和支持我的人。

　　首先感谢我的指导老师张菁教授，张老师为人很和蔼、开朗，对学生总是笑脸相迎，她不仅帮助和指导我顺利完成了毕业设计，还教会了我很多其他的生活、职场中需要的技能。没有张老师的支持，我是没办法独自走过从立题到答辩的艰难历程的，所以真诚的感谢张菁老师！

　　然后我要感谢学长学姐的帮助，他们在我的毕设历程中扮演了重要的角色，他们教会了我如何设计这款系统的功能，如何改进等等。我的毕设也是他们努力的成果！另外需要感谢我的父母和挚爱对我的陪伴，他们在毕业设计的这段路途中不断鼓励我，以及生病时及时的关怀。希望我的父母身体健康，我和她可以走得更远！

　　最后感谢母校的唐滨老师，在他的帮助下，我才有机会进入计算机学院的实验室学习，在实验室里成长！没有他的指导，我也没有机会接触到我所感兴趣的领域。希望唐滨老师工作顺利，事业成功！

参考文献

1. []Allen A A, Plourde J V. Review of Leeway: Field Experiments and Implementation[J]. Review of Leeway Field Experiments & Implementation, 1999. [↑](#endnote-ref-1)
2. [] Chen D S, Yan J H, Xiu-Ying B I. A flow field model and its application in search and rescue operations in adjoining sea area of Zhanjiang[J]. Journal of Tropical Oceanography, 2008, 27: 16–21 [↑](#endnote-ref-2)
3. [] Xiao W J, Du P J, Gong M X, et al. Xiao W J, Pan-Jun D U, Gong M X, et al. An operational search and rescue model system for Shanghai coast and adjacent seas[J]. Marine Forecasts, 2013, 30: 79–86 [↑](#endnote-ref-3)
4. []Zhang D, Yan X P, Yang Z L, et al. Incorporation of formal safety assessment and Bayesian network in navigational risk estimation of the Yangtze River[J]. Reliability Engineering & System Safety, 2013, 118(10):93-105. [↑](#endnote-ref-4)
5. [] Hodgins, D.O., Mak, R.Y. Leeway Dynamic Study, Phase I: Development and Verification of a Mathematical Drift Model for Four-person Liferafts. Seaconsult Marine Research Limited, 1995 [↑](#endnote-ref-5)
6. []Øyvind Breivik, Allen A A. An operational search and rescue model for the Norwegian Sea and the North Sea[J]. Journal of Marine Systems, 2011, 69(1):99-113. [↑](#endnote-ref-6)
7. [] Allen A A. Leeway Divergence[J]. Leeway Divergence, 2005. [↑](#endnote-ref-7)
8. []Brushett B A, Allen A A, Futch V C, et al. Determining the leeway drift characteristics of tropical Pacific island craft[J]. Applied Ocean Research, 2014, 44(1):92-101. [↑](#endnote-ref-8)
9. []Øyvind Breivik, Allen A A, Maisondieu C, et al. Wind-induced drift of objects at sea: The leeway field method[J]. Applied Ocean Research, 2011, 33(2):100-109. [↑](#endnote-ref-9)
10. [] Geyer W R. Field Calibration of Mixed-Layer Drifters[J]. Journal of Atmospheric & Oceanic Technology, 2009, 6(2):333-342. [↑](#endnote-ref-10)
11. [] Spaulding M L, Isaji T, Hall P, et al. A hierarchy of stochastic particle models for search and rescue (SAR): Application to predict surface drifter trajectories using HF radar current forcing[J]. Journal of Marine Environmental Engineering, 2006, 8(3):181-214. [↑](#endnote-ref-11)
12. [] Özgökmen, Tamay M, Griffa A, Mariano A J, et al. On the Predictability of Lagrangian Trajectories in the Ocean[J]. Journal of Atmospheric & Oceanic Technology, 2000, 17(3):366-383. [↑](#endnote-ref-12)
13. [] Davidson F J M, Allen A, Brassington G B, et al. Applications of GODAE Ocean Current Forecasts to Search and Rescue and Ship Routing[J]. Oceanography, 2009, 22(3):176-181. [↑](#endnote-ref-13)
14. [] Mínguez R, Abascal A J, Castanedo S, et al. Stochastic Lagrangian trajectory model for drifting objects in the ocean[J]. Stochastic Environmental Research & Risk Assessment, 2012, 26(8):1081-1093. [↑](#endnote-ref-14)
15. [] Zhang J, Ângelo P. Teixeira, Soares C G, et al. Probabilistic modelling of the drifting trajectory of an object under the effect of wind and current for maritime search and rescue[J]. Ocean Engineering, 2017, 129:253-264. [↑](#endnote-ref-15)