# A Landing Site Prediction System for High-Altitude Balloons

Uri Pierre Burmester

November 27, 2017

AITC Summer Internship
Supervisor: Dr James Gilbert

# Contents

# 1 Introduction

High-altitude balloons (HABs) are unmanned, near-space balloons which are released from the Earth's surface. They rise (typically to the strastosphere), before bursting and falling back down to the ground. These balloons are constructed for the purposes of gathering scientific data (such as temperature, pressure and wind speed) or just to take photos from a high altitude. Variations in flight characteristics like weather conditions, balloon design and obstacles can render a HAB's landing site unpredictable. This in turn makes the process of retrieving the balloon and accessing its data difficult. The purpose of this project is to develop software to predict the HAB landing site using GPS coordinate data gathered by an onboard Raspberry Pi and transmitted to a computer on the ground.

# 2 Project Scope and Requirements

Existing landing prediction software runs on the balloon's main flight computer. The main process running on this computer is the tracker, which handles GPS reception, telemetry, and radio communications with the ground. Though the landing prediction system is separate to the tracker, it does interface with it - GPS data is an input, and a periodic prediction of the payload landing site is an output. Tracker-related tasks, such as driving GPS hardware or handling radio communications, is not within the prediction system scope. This project will not consider the software driving transmission of the signal from the balloon nor receipt of the signal on the ground. That is, sources of interference, incomplete data transmission and the radio transmissions protocols will not be considered.

## 2.1 Interfaces

The system interfaces are as follows:

| Tracking system (GPS log) | |
|---|---|
| Type | Data |
| Description | Recent and historical latitude, longitude and altitude data. |

| Tracking system (data transmission) | |
|---|---|
| Type | Data |
| Description | The latest landing site prediction (latitude, longitude) to be transmitted by the tracking system's radio transceiver(s). |

## 2.2 Model and Assumptions

A moment-by-moment prediction of the landing site will be modelled by considering what would happen if the balloon burst at this particular instant. Air flow will be modelled as a series of concentric rings of equal wind speed through which the balloon will fall during its descent. The parachute is assumed to deploy instantly after the balloon burst and it will be modelled as a drag force on the payload during its descent. This, along with the wind , position and velocity information are sufficient to pose this as a kinematics problem. Flight data from actual balloon launches (available online) will be taken as a basis to test the accuracy of the prediction.

## 2.3 Requirements

| HAB-LPR-R01 – Purpose and inputs | |
|---|---|
| Requirement | The landing site prediction system shall estimate the landing site of the payload based only on recent and historical GPS coordinates and altitudes. |
| Rationale | A log of GPS data is the primary source of information from the balloon payload. |

| HAB-LPR-R02 – System host | |
|---|---|
| Requirement | The landing site prediction system shall run on the same computer as the tracking system. This computer will be a Raspberry Pi Zero or Raspberry Pi A+. |
| Rationale | Ease of data I/O and reduction of payload mass (existing computer, power supply etc.) |

| HAB-LPR-R03 – Start on boot | |
|---|---|
| Requirement | The landing site prediction system shall be capable of starting when the flight computer is booted, without external input from other systems or users. |
| Rationale | Simplicity upon launch; the computer will not have user interfaces, and reliance on other systems or people increases the chance of error before balloon release. |

| HAB-LPR-R04 – Input format | |
|---|---|
| Requirement | The landing site prediction system shall obtain input data from a text file at a customisable path on the flight computer. The data format is comma-separated values (CSV), using ASCII characters and terminated by a newline character. There may be up to twenty fields per line. The column numbers of the input fields (latitude, longitude, altitude) shall be customisable in the code. Numbers may be zero-padded. An example string is: $\$\$YERRA, 698, 00 : 23 : 30, -35.32110, 149.00710, 00747, 11, 16, 9, 50.7, 31.2, 932, 31.1 * 16A8$ |
| Rationale | An industry standard method of storing data that is compatible with the existing tracker software. |

| | |
|---|---|
| HAB-LPR-R05 – Units | |
| Requirement | All input and output units shall be in decimal degrees (±90 lat, ± 180 lon) for coordinates, and metres for altitudes. |
| Rationale | Consistency with existing tracker software. |

| | |
|---|---|
| HAB-LPR-R06 – Output format | |
| Requirement | The landing site prediction system shall write output data to a text file at a customisable path on the flight computer. The data format shall be comma-separated values (CSV), using ASCII characters and terminated by a newline character. The column order shall be customisable in the code. The values shall be numbers only. |
| Rationale | An industry standard method of storing data that is compatible with the existing tracker software. |

| | |
|---|---|
| HAB-LPR-R07 – Logging | |
| Requirement | The landing site prediction system shall write all prediction outputs to a log in the form of an ASCII text file. All logs shall include a UNIX timestamp (nearest second or better), and any other relevant parameters, such as the latest input data. |
| Rationale | A historical record of the system's output will be useful for improving the system. |

| | |
|---|---|
| HAB-LPR-R08 – Internet connection | |
| Requirement | The landing site prediction system shall not require an internet connection to operate. |
| Rationale | A driving requirement of this system is that it functions without an internet connection. |

| | |
|---|---|
| HAB-LPR-R09 – Valid prediction timings | |
| Requirement | The landing site prediction system shall begin outputting valid prediction data within one minute after balloon burst (payload decent), or earlier. Prediction during ascent (assuming imminent burst) is desirable but optional. |
| Rationale | Balloon burst is the point at which all ascent data should exist and landing site prediction becomes particularly important. |

| | |
|---|---|
| HAB-LPR-R10 – Prediction update interval | |
| Requirement | The landing site prediction system shall output a new prediction at intervals of two minutes or less, from when the first prediction is made. |
| Rationale | Refining predictions based on the latest data should increase prediction accuracy, and frequent updates will benefit ground logistics to approach the predicted landing site. |

| HAB-LPR-R11 – Error handling | |
|---|---|
| Requirement | The landing site prediction system shall at no point jeopardise the continued operation of the flight computer and tracker system. All errors shall be handled in a way that, at worst, stops operation of the prediction software only. The system shall be fail-safe. |
| Rationale | Affecting the tracker system could lead to a loss of communications and a lost flight. |

# 3 Project Milestones and Deliverables

The code will include several iterations (”versions”):

1. VERSION 1: The code reads a static .txt file, identifies the relevant wind speeds in the data and stores them in a data format.
2. VERSION 2: The code predicts a single landing site as a set of GPS coordinates based on the static .txt file
3. VERSION 3: The code updates its prediction of the landing site as new data is added to the file.

## 3.1 Deliverables

In this VERSION 4, the code that takes a continuously updating .txt file in the format produced an actual Raspberry Pi and outputs a predicted landing site on a map.

# 4 Project Timeline

An expected timeline is shown in Table 1.

| Date | Week | Event | Comments |
|---|---|---|---|
| Fr. 01/12/2017 | 2 | Literature review complete, project plan set | - |
| Fr. 08/12/2017 | 3 | File I/O working in telemetry format (static) | - |
| Fr. 15/12/2017 | 4 | VERSION 1 Complete | - |
| Fr. 22/12/2017 | 5 | File I/O working in telemetry format (updating) | Raspberry Pi should be obtained before break for testing purposes. Timeline to be reviewed |
| Fr. 05/01/2018 | 6 | VERSION 3 complete | - |
| Fr. 12/01/2018 | 7 | Presentation of results | Practice time allocated for this week |
| Th. 18/01/2018 | 8 | VERSION 4 complete | - |
| Fr. 19/01/2018 | 8 | Project Report complete | Examination of further work included in report |

Table 1: Expected project timeline