Active Camera Stabilization from High Altitude Balloons

Max A. Bowman,[1] Jeremy Seeman[1], Ken Walczak[2]
Adler Planetarium Chicago, Chicago, IL, 60605, United States

Many applications of High Altitude Ballooning (HAB) require maintaining a bearing in a non-inertial frame of reference, for example keeping a camera continuously pointed at the sun from a HAB payload in motion. Maintaining a consistent bearing is especially difficult on a HAB flight because it is hard to accurately measure and compensate the relative motion of the payload. In addition, friction, conservation of angular momentum, and errors in relative bearing measurement complicate the solution. This project aims to solve these issues using a combination of accelerometer data, gyro data, and a motor-driven platform. It was launched as part of the Far Horizons project at the Adler Planetarium, whose mission is bringing real space exploration down to Earth and into the hands of students, volunteers, and the public. The initial design was built for use in the 2017 Eclipse mission, but will also be used in missions such as mapping light pollution from a HAB platform. The system expands the capabilities of general HAB missions and can be open-sourced to provide the HAB community with a solution to a unique problem in High Altitude Ballooning.

Nomenclature

dt = time step

f = generic function

θ = generic angle (degrees)

t = time (seconds)

Subscripts

f = final

I. Introduction

With the advent of High Altitude Ballooning projects, the potential for a cost-effective way to gather targeted visual or sensor observations is becoming more necessary. Several of these observations require the maintenance of a constant bearing in order to gather useful data. Examples of such phenomena include observing the 2017 Great American solar eclipse, mapping urban light pollution, and other remote sensing applications. However, because of the unique conditions HAB flights operate in, collecting useful visual data presents a set of challenges. HAB payloads can rotate at speeds of up to one revolution per second [1] during flights, making active stabilization especially necessary for flights involving imaging. On a flight, accurately estimating and compensating for angular rotation with little noticeable jerk is especially challenging due to the problem of smoothing motor rotation. Although a stepper motor will work well for applications where applying post-production techniques to a video is possible [2], there are some applications where this method is infeasible. Currently, the Far Horizons program is preparing a mission to map light pollution in the Greater Chicagoland area from a HAB, where this technology will be utilized. Previous Far

[1] Far Horizons Lab Assistant, Adler Planetarium, 1300 S Lake Shore Dr, Chicago, IL
[2] Far Horizons Program Manager, Adler Planetarium, 1300 S Lake Shore Dr, Chicago, IL

Horizons HAB flights have indicated that non-stable photography of light pollution results in "streaking," in which an image become blurred from considerable motion during exposure (see Fig. 1). It then stands to reason a similar effect would be induced by the sharp jerks inherent in stepper motor operation. Additionally, cameras that need to be wired to other payload elements introduce the problem of wire-strain [3]. This paper explores the use of a motor, rotary slip, and gyro data to compensate azimuth rotation with minimal jerk and log x, y, and z-axis relative heading angles in order to maintain instrument bearing during HAB flights.

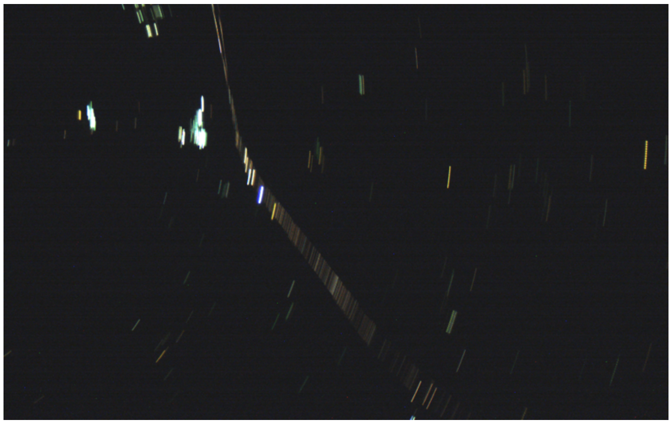

Fig. 1 "Streaking" caused by payload motion during a HAB flight

II. System Description

A. System Overview

The system utilizes a 9DoF Razor IMU, DC motor with encoder, and RTC to provide complete pointing knowledge and correct azimuth rotation. Different parts of the system use various kinds of serial protocols to communicate, including SPI, I2C, and UART. Fig. 2 shows a high-level block diagram of the downward-pointing system. The original downward-pointing system was very similar, but used a different IMU and did not log RTC or heading information to an SD-card.

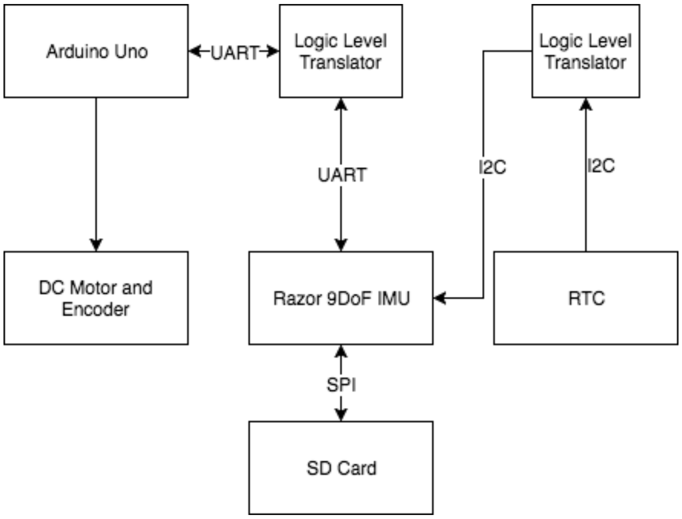

Fig. 2 System Block Diagram

B. Mechanical design

The system utilizes a DC motor with encoder, rotary slip, and beveled gears in order to permit continuous rotation around the azimuth axis (see fig. 3). The rotary slip prevents wiring used for camera (or sensor) control and external power from interfering with rotations or being damaged from excessive azimuth rotations in one direction, a problem encountered in other azimuth-correcting systems [3]. The older upward-pointing system uses a similar mechanical design, albeit with a stepper motor and helical gears.

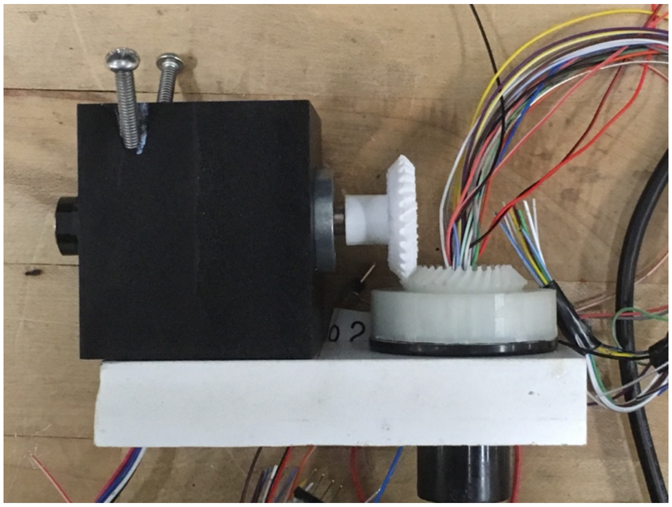

Fig. 3 The system uses a beveled gear to rotate the camera(s) around the rotary slip

A primary goal of the system is to be completely self-contained and fit into a single payload box. Inside the payload are trays that hold necessary electronics and mechanical elements. This includes gears, the DC motor, and the 9DoF IMU. The system was initially designed to be upward-facing (see Fig. 4) and this design is currently being adapted to be used for missions that require a downward-pointing camera. The cameras will be insulated inside the payload box, but holes in the bottom of the payload box will permit unobscured imaging.

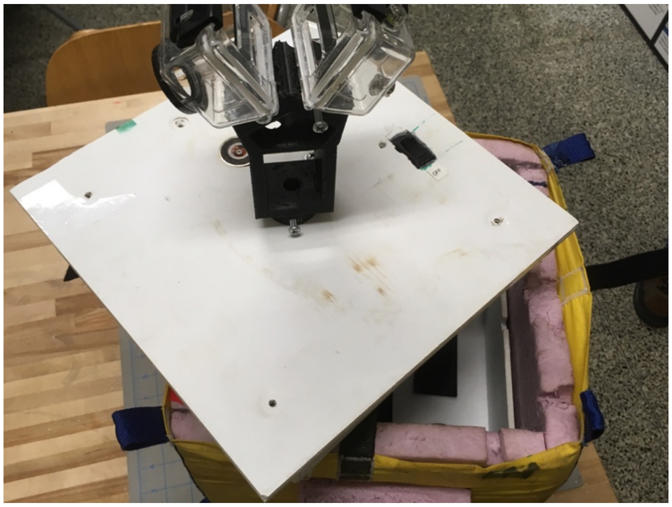

Fig. 4 System trays design with legacy upward-facing pointing system as top tray

C. Electrical design

The Razor IMU was chosen because its firmware is programmable and it has an onboard microSD slot. Programming the firmware to integrate the gyro over time takes a large burden off the Arduino Uno and allows it to more precisely control the motor, its primary job. Currently, the IMU only integrates the gyro over time with no sensor fusion with a magnetometer or accelerometer. Since the system compensates rotation only about the azimuth, the x, y, and z-axis headings are logged to a microSD card onboard the IMU for accurate pointing knowledge when analyzing flight data. This removes the need to add an SD card shield, and decreases the overall cost and complexity of the system. With each heading, a reading from an external RTC is included for time-sensitive data and for correlation with data from other onboard sensor logs. The original upward-pointing system had a similar electrical design, but the IMU did not do onboard integration (the Arduino Uno performed this task) and used stepper motor instead of a DC motor and encoder to make azimuth corrections (see Fig. 5).

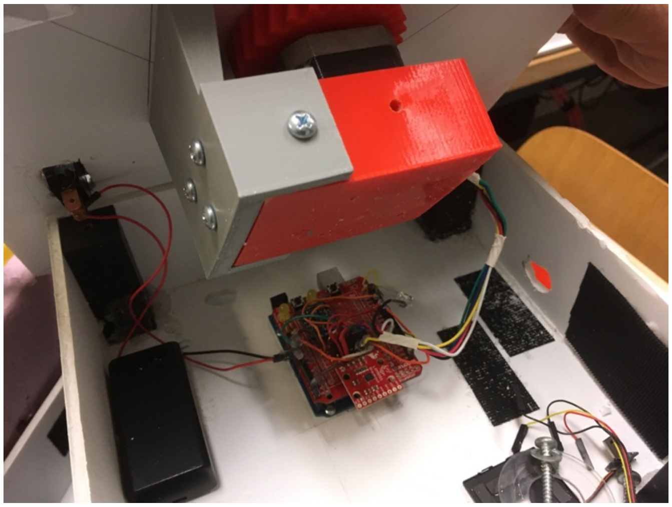

Fig. 5 The original upward-pointing system used a stepper, IMU, and Arduino UNO to maintain bearing

III. Motor Control

As a result of this system's intended use in imaging and video capture, a system priority is reducing jerk involved in rotating cameras. In order to prevent blurring the camera image, it is necessary to smoothly accelerate and then decelerate the motor. In mathematical terms, we are looking for some smooth function that describes desired motor rotation versus time given an angle measure θ_f to rotate. We will refer to this equation as $\omega(t \mid \theta_f)$ for now, but eventually we will need to add a constant k as a given to the equation. In order for this equation to behave correctly (that is, to rotate the proper degree measure), the following equation must be true where t_f is the end time of the rotation.

$$\int_0^{t_f} \omega\left(t \mid \theta_f\right) dt = \theta_f \qquad (1)$$

We hypothesize that using a beta distribution scaled along the x-axis will distribute the jerk over time such that it improves the quality of visual data. We chose a beta distribution where $\alpha = \beta = 5$ so that maximum acceleration occurs roughly one third and two thirds through the rotation. The area below a beta distribution from $x = 0$ to $x = 1$ is equal to one [4], so the following equation applies to our selected beta distribution.

$$\int_0^1 \frac{\Gamma(5+5)}{\Gamma(5)\Gamma(5)} (1-t)^4 t^4 \, dt = 1 \qquad (2)$$

We propose making the end time of the rotation t_f proportional to the desired angle rotation θ. We call $t_f = k\theta_f$ our "target time," as it is the time by which the rotation will have finished. In order to stretch the beta distribution to accommodate our rotation, we will substitute t with $\frac{t}{k\theta_f}$ to obtain the function $f(t) = \frac{\Gamma(5+5)}{\Gamma(5)\Gamma(5)}(1 - \frac{t}{k\theta_f})^4(\frac{t}{k\theta_f})^4$. We are looking to find a function of this form with an integral of θ_f on the limits of 0 to t_f. Therefore, we integrate $f(t)$ with respect to t from 0 to t_f using the substitution $u = \frac{t}{k\theta_f}$ and $du = \frac{1}{k\theta_f}dx$. This substitution yields the following equation.

$$k\theta_f \int_0^1 \frac{\Gamma(5+5)}{\Gamma(5)\Gamma(5)}(1-u)^4 u^4 \, du \tag{3}$$

By applying eq. (2) to the previous formula, we find that

$$k\theta_f \int_0^1 \frac{\Gamma(5+5)}{\Gamma(5)\Gamma(5)}(1-u)^4 u^4 \, du = k\theta_f = \int_0^{k\theta_f} f(t)\, dt \tag{4}$$

As eq. (4) demonstrates, $f(x)$ almost behaves as the function we are looking for. However, its integral with respect to x from 0 to $k\theta_f$ is only proportional to θ_f, not equal to θ_f. This is easily remedied by dividing k from both sides. We see that $\frac{f(t)}{k}$ matches the function we are looking for, so we determine that the following equation models desired motor rotation speed versus time.

$$\omega(t \mid \theta_f, k) = \frac{f(t)}{k} = \frac{1}{k}\frac{\Gamma(5+5)}{\Gamma(5)\Gamma(5)}(1 - \frac{t}{k\theta_f})^4(\frac{t}{k\theta_f})^4 = \frac{630}{k}(1 - \frac{t}{k\theta_f})^4(\frac{t}{k\theta_f})^4 \tag{5}$$

In application, this ideal function will only be sampled. As a result, the amount the motor rotates will be greater than θ. To correct for over-rotations, encoders can be used. We propose simply subtracting the amount over-rotated from the next rotation. Alternatively, a Riemann-sum can be kept of the rotation and the difference between the sum and θ can be subtracted to the next rotation. Fig. 6 demonstrates an ideal rotation curve for $k = 100$ and $\theta = 3^\circ$. For this rotation, sampling the curve on MATLAB with a Riemann-sum every 100 milliseconds yielded an over-rotation error of roughly 2.42%, while sampling it every 50 milliseconds yielded an error of just 0.042%.

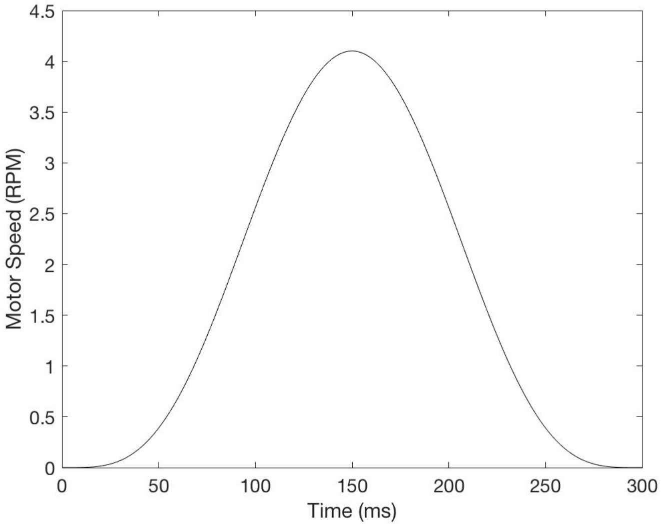

Fig. 6 A scaled rotation given $\theta = 3°$ and k = 100

IV. IMU Testing

In lab, we tested the efficacy of our IMU firmware. After mounting the IMU on a 180-degree servo, we rotated the servo in steps of 3 degrees and measured the change in angle the IMU calculated. Fig. 7 demonstrates the IMU's calculated change in angle measure versus actual change in angle. Over 50 data points, the IMU measured the 3-degree rotation to be 3.129 degrees on average with a variance of plus or minus 0.410 degrees. This large variance makes clear that we will need to use a magnetometer and accelerometer along with the gyro to correct for gyro errors. Additionally, we observed gyro drift on the order of single degrees per second in lab, indicating that sensor fusion of the gyro with an accelerometer and magnetometer is necessary for an accurate heading measurements.

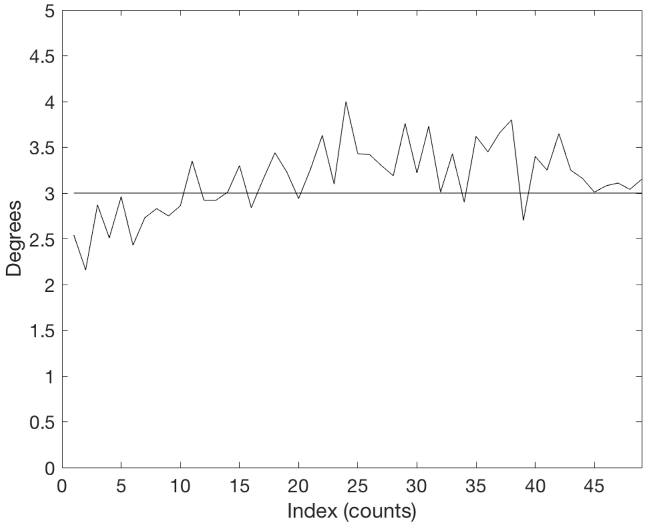

Fig. 7 Calculated rotation vs. actual rotation

In addition to this lab test, we used z-axis IMU data from a past flight in order to characterize azimuth rotation over the duration of a flight. Fig. 8 shows data taken from launch to burst, as this is where the majority of usable HAB flight data comes from. IMU data is useful in determining design constraints of active stabilization systems and providing accurate conditions for lab testing of these systems. We found a maximum azimuth rotation rate of 269.15 degrees per second, which supports prior research indicating a maximum rotation rate of roughly one revolution per second. The average rotation speed over the flight was approximately 46.22 degrees per second, with a standard deviation of 40.02 degrees per second.

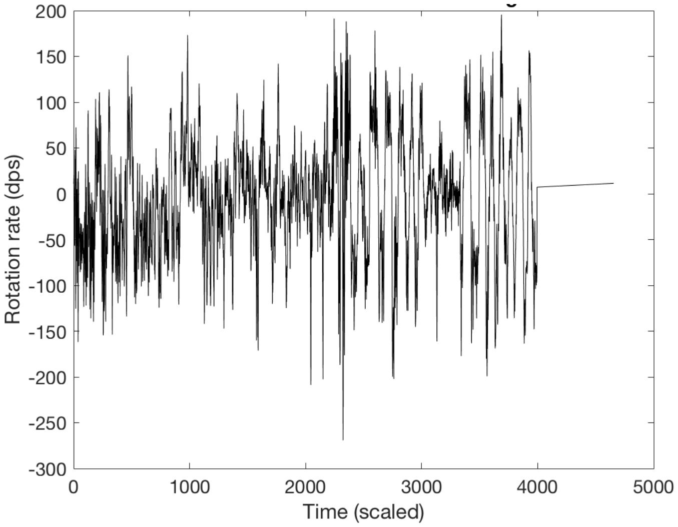

Fig. 8 Azimuth rotation profile over a HAB flight from launch to burst

V. Conclusion

Maintaining a constant bearing via active stabilization is a difficult process that requires many considerations and mechanical design elements. Through our design process, we have been able to identify preliminary success. However, there are many outstanding design challenges that need to be addressed. The first is creating a better relative heading determination system. Our current system relies solely on a gyro, but we are currently working to add sensor fusion to increase accuracy of the heading and reference system and allow for absolute pointing determination. Additionally, we are working to add passive payload stabilization elements in order to lower active system requirements and prevent the rotating platform from inducing noticeable payload counter rotations. Far Horizons is preparing for flight testing in upcoming months which will provide a gauge of this solution's efficacy. This project is currently being adapted in an effort to map the light pollution of the Chicago area in order to expand our understanding and quantification of light pollution for research.

Acknowledgments

I would like to thank Ken Walczak and Jeremy Seeman from the Adler Planetarium, Chicago for advising and mentoring me with this project. I would also like to thank the Adler Planetarium Far Horizons team and the Illinois Space Grant Consortium for making this research possible.

References

[1] Flaten, J., Gosch, C., and Habeck, J., "Techniques for Payload Stabilization for Improved Photography During Stratospheric Balloon Flights," *Proceedings of the 2015 Academic High Altitude Conference*, 2015.

[2] Kruger, A., Maksimowicz, R., Zaheer, M., Almaraz-vega, A., College, W. W., and Ave, N. N., *Active Heading Control Platform for Instruments Flown on High Altitude Balloons*, 2015.

[3] Plewa, M. I., and Scharlau, B. M., "Directional Camera Control on High Altitude Balloons," 2017, pp. 1–5.

[4] Weisstein, E., "Beta Distribution," *Mathworld.wolfram.com,* URL: http://mathworld.wolfram.com/BetaDistribution.html [retrieved 9 December 2017].