Image Processing for Data Acquisition and Machine Learning of Helicopter Flight Dynamics

Dan Tappan and Matt Hempleman
Department of Computer Science, Eastern Washington University, Cheney, WA, USA

Abstract - *Learning to fly a full-sized helicopter is a complex iterative process of mapping interdependent causes to effects via inputs to outputs in real time in a wildly dynamic, messy, and unforgiving environment. This work presents a prototype system for noninvasively acquiring otherwise inaccessible data from the controls, instruments, and flight dynamics of a Robinson R22 helicopter with an array of cameras and sensors and then processing these images with OpenCV-based solutions into corresponding numerical form for later use in a machine-learning project. It describes a hardware and software architecture for safely and successfully calibrating the system, running a breadth and depth of representative experiments, and qualitatively and quantitatively presenting and validating the results.*

Keywords: feature extraction, data acquisition, machine learning, aviation

1 Introduction

Autonomous aircraft, especially consumer drones, have become an $11 billion yearly industry [9]. Machines can learn to fly well for many mainstream purposes now, but most approaches are disconnected from the way human pilots learn to fly [6]. The computational models provide little insight into the learning processes of either group. A better understanding would advance the field of artificial intelligence and intelligent systems. It could also extend this capability to other environments where machine learning might be advantageous.

This paper addresses the first two objectives of a larger project: (1) to build an acquisition system for recording flight data from a full-sized helicopter; (2) to collect data from basic flight maneuvers as representative teaching examples of how to perform them; (3) to investigate data processing and fusion techniques to merge data from numerous repetitions of maneuvers done to account for variation and errors; (4) to build a rudimentary software flight-dynamics model based on the nature of the collected data; and (5) to investigate machine-learning techniques to allow the system to learn and explain how to perform the same actions as the human pilot (Tappan).

The key element is to acquire real-world data from a Robinson R22 two-place helicopter, which is the world's most popular trainer [15]. Its wide range of capabilities and relatively low operating cost make it convenient for such activities. However, its primitive instrumentation provides no capability to log flight data directly. This limitation is significant because the machine-learning project must have the same awareness that a human pilot has, namely visual perception of the outside world and an understanding of the internal state of the helicopter by visually observing its instrumentation. Outfitting the helicopter with a complex array of sensors, as is common in other work, would undoubtedly be more effective, but a human pilot does not learn to fly based on such unnatural stimuli [4].

Two aspects of this proof-of-concept solution are considered here: the architecture for visual data acquisition, and an OpenCV-based postprocessing system for converting these data into usable numerical form [13]. The primary requirements address safety and practicality (in no order): no interference (physical or electrical) with the helicopter; no attachments at all outside, and no substantive ones inside; ease of setup and tear down; minimal wiring; the fewest number of cameras in the least obtrusive places; and no distraction for the pilot. Section 3 covers the technical requirements.

2 Background

A helicopter exhibits six degrees of freedom in its physical state: it has a position in space on the x, y, and z axes and an orientation respectively in *roll*, *pitch*, and *yaw* (collectively known as attitude) about them. A seventh variable is *time*, which contributes to computing the speed (change in state) and acceleration (change in speed) of the other six. This work assumes the coordinate system in Figure 1a.

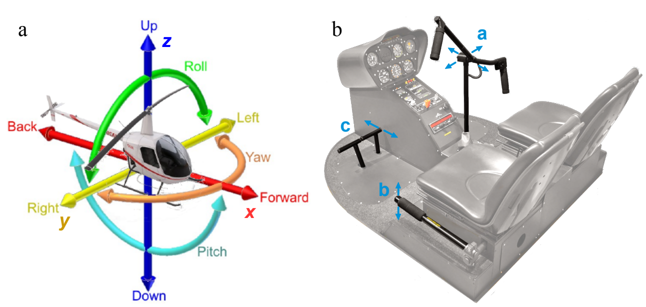

Figure 1: Degrees of freedom and cockpit controls [2,1]

2.1 Control inputs

In order to establish cause-and-effect relationships in flight, the machine-learning system must be able to connect changes in the inputs to their effects on the output state of the helicopter. To this end, the first part of image acquisition monitors the primary flight controls available to the pilot. Most aircraft have dual controls available to both pilots simultaneously, which is essential in a training environment. This work assumes only one pilot, sitting on the right.

The *cyclic pitch control* (usually called the “cyclic”) is in principle a joystick for the right hand with two degrees of freedom (x and y) such that forward/backward movement affects pitch, and sideways movement affects roll. However, the actual arrangement in Figure 1b (a), known as a *T-bar*, places the pivot in the middle of the cockpit. The downside is that the position of the pilot’s hand cannot be directly tracked to determine the corresponding inputs because the teetering nature of the bar allows vertical movement of roughly 30 centimeters without any actual changes to the input. Section 6.4 covers this issue further.

The *collective pitch control* (the “collective”) in Figure 1b (b) is a lever with a vertical arc of travel that changes the amount of thrust from the main rotor to affect the z position (altitude), and through more complex interactions, the x and y positions. While its range of motion is more regular than that of the cyclic, it is mostly obscured by the seats and the pilot’s left arm. On the end of the lever is the throttle, which is like a motorcycle twist grip. In some helicopters, the pilot manages this input manually, which would require corresponding data acquisition, but the R22 normally operates in automatic mode.

Finally, the *antitorque pedals* (the “pedals”) in Figure 1b (c) travel in a forward/backward arc to change the amount of thrust from the tail rotor to affect yaw. The pedals are linked in opposition, so pushing one forward moves the other backward correspondingly. Only one needs to be tracked.

2.2 Augmented outputs

Through complex flight dynamics far beyond the scope of this paper, every input affects the state of the helicopter in multiple interdependent ways. Unlike an airplane, which is always in motion in flight and must generally face its direction of travel, a helicopter is practically unlimited in its maneuverability. This capability offers great flexibility in use, but it has a significant downside for automated data acquisition because most of the fine state awareness is acquired visually by the pilot looking out the window. At least in small helicopters, instrumentation is sparse.

To mitigate this limitation, this work provides quantitative instrumentation in the form of small, inexpensive sensors. The CHR-UM7 integrated inertial measurement unit (IMU) and attitude heading reference system (AHRS) in Figure 2a records all six degrees of freedom [12]. It operates within a local frame of reference, meaning that it is aware of the state of the helicopter relative to itself only, not of its relationship to the world it operates in. In other words, it records changes only; it cannot establish absolute state data like latitude and longitude or altitude. For this purpose, the Parallax LS20031 GPS receiver in Figure 2b supplies x, y, z coordinates and compass heading for yaw, as well as real-world time, for the global frame of reference [14]. This instrumentation is essential for data acquisition in the larger project, but its role in this paper is limited to crosschecking the results from the image acquisition and processing.

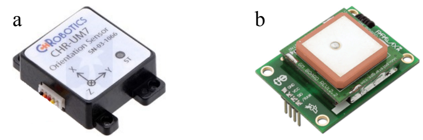

Figure 2: IMU/AHRS and GPS units [12,14]

2.3 Instrument outputs

The sensors do not interact with the helicopter beyond being simply attached to it internally. Despite the rich quantitative data they provide in native digital form, the overall picture is still incomplete. The visual data available to the pilot from the following cockpit instruments are also needed.

The *altimeter* in Figure 3a measures altitude above sea level in feet. It has three elements of interest: a long needle for hundreds, a short needle for thousands, and a triangle for tens of thousands. Converting them from individual needles into a single value for altitude is a straightforward equation, but it does require establishing their values separately.

The *vertical speed indicator* (VSI) in Figure 3b measures change in altitude in feet per minute. The GPS already provides the equivalent of the altimeter and VSI data. However, it is counter-intuitively *too good* in this role. The cockpit instruments have complex real-world behaviors and limitations that affect how the pilot interprets them, such as a lag in response time. For machine learning to function as a human does, it needs to deal with the same issues.

The *airspeed indicator* (ASI) in Figure 3c measures the speed of the helicopter through the air. The GPS receiver also appears to provide these data, but it actually measures the speed over the ground. Wind conditions almost always cause these two values to be different. The aircraft, and thus the pilot, react to airspeed, which the sensors inside the cockpit cannot measure.

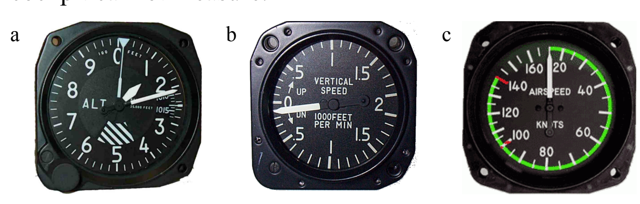

Figure 3: Altimeter, VSI, and ASI instruments

The *manifold pressure gauge* (MAP) in Figure 4a measures the amount of power being demanded from the engine, which varies according to the inputs from the pilot. The acceptable range is based on atmospheric conditions and determined from tables in the pilot's operating handbook.

The combined engine and main-rotor *RPM gauge* in Figure 4b measures the rotations per minute of each as a percent and indicates the acceptable operating range. This instrument is not considered here because of the automatic throttle management, but in other helicopters or more advanced experiments, it would be important. Section 7 covers its value for future work.

Finally, Figure 4c depicts the least high-tech instrument, the *yaw string*, which is a small piece of yarn attached to the front outside of the canopy. It indicates by wind deflection how the nose and tail of the helicopter (essentially the yaw) are aligned with respect to the direction of travel, known as coordinated flight. While this detail could actually be very useful in some contexts, for logistical reasons this output is not considered here. (And it can be derived reasonably well from the sensors.) Similarly, the compass, which also technically provides yaw, is not considered because in practice it is so unreliable as to be almost completely useless, even to a human.

Figure 4: MAP and RPM gauges and yaw string [3]

3 Architecture

The hardware architecture needed to support up to six cameras in simultaneous operation for complete coverage. The requirements were (in no order) that they be inexpensive, small, lightweight, relatively easy to mount, externally powered, have reasonable video quality, store to flash memory cards, and permit remote operation. The FlyCamOne eco V2 in Figure 5a satisfied all these needs remarkably well [11]. Designed to provide a pilot's view in small radio-controlled aircraft, its compact 15-gram package records 24 frames per second of 24-bit color at 720×480 resolution with three megapixels. The image quality from its tiny lens is acceptable, but not great.

Figure 5: FlyCamOne and BeagleBone [11,10]

A critical safety requirement in this work was not to distract the pilot from flying the helicopter. Each test flight generally took an hour and involved several dozen small experiments. The pilot could not afford to be manipulating the system in any complex way to start, run, and end each experiment. (The acquisition system occupied the other seat, so having an assistant was not an option.) The large number of experiments combined with the large number of cameras and sensors required simple one-button coordinated operation to advance to the next experiment.

To this end, the compact BeagleBone Black single-board computer in Figure 5b mapped this button to the appropriate actions [10]. Through serial and I^2C interfaces, it controlled the sensors and recorded their data. Controlling the cameras was similarly convenient because their intended use in radio-controlled aircraft provided a communication interface through standard pulse-width-modulated (PWM) servo signals. The camera data, however, were stored on the 8GB microSD memory cards in the cameras themselves. Transferring so much data over such a distance on lightweight unshielded cables to a relatively weak computer was not an option, so the BeagleBone could not manage the files itself. (In earlier proof-of-concept tests, even a high-powered laptop was unable to keep up with six comparable cameras connected via USB.)

This solution introduced a major problem with synchronizing the files across all the cameras because each camera generates a different filename with no timestamp when started. Therefore, after a flight, it was almost impossible to determine which file referred to which experiment. Conveniently, however, these cameras also record audio. Each time the BeagleBone instructed the cameras to start recording, it played a Morse code-like preamble identifying the automatically generated test number. While not particularly human-friendly, this code provided enough information to change the filenames by hand to something meaningful later. The BeagleBone also generated a second tone sequence every five seconds to ensure that the timing across all videos could be synchronized when startup delays occurred or the recording rates were not exactly the same.

4 Image processing

Image processing is the core of this work, but this paper is primarily about the architecture that facilitated it. It plays the role of postprocessing the videos into a series of values that correspond to the state of the controls and instruments. The processing itself is relatively straightforward and uses traditional approaches in OpenCV as intended. Hempleman [5] provides a very detailed description to supplement the summary here.

4.1 Controls

The controls come in three forms with related types of linear or angular motion, so the same image-processing approach could be applied to each. The most important aspect was being able to track a known object affixed to key points on the controls. This step entailed significant *what if* experimentation to find a satisfactory (but never ideal) solution. The requirements limited the object to being something small and unobtrusive, like a sticker. Selecting the size and color alone could be its own paper because image acquisition operated under such a wide range of environmental conditions. (See Section 6.4.) This part investigated dozens of combinations of swatches made of every conceivable colored tape and paper, as well as 19 small LEDs. Similarly, camera placement entailed many experiments to find reasonable compromises within the tiny, cramped cockpit. This section summarizes the overall approach of color-based blob detection, the details of which often varied depending on the particular goals and actual conditions, etc.

Although the lighting, contrast, and other uncontrollable dynamic factors varied wildly in the cockpit, nothing else consistently resembled the roughly 8mm reflective orange tape squares on a black background in Figure 6. Color, hue, and saturation isolation were usually able to find this object within the expected region. The controls do not normally move quickly, so tracking the position of a known object at 24 frames per second was reliable. However, different positions of both the controls and the helicopter itself changed the target color threshold frequently. To mitigate this variation, the tracking algorithm started with the exact color to find (or its components) and then relaxed the requirements iteratively until it found a strong match. If it could not, it ignored these frames until it could again. Further postprocessing into machine-learning data interpolated any missing frames, assuming that the missed motion was linear.

Figure 6: Pedal, collective, and cyclic tracking objects

With the target object isolated within the frame and the bounds of the calibrated range known (see Section 5), it was a straightforward algebra problem to translate the centroid of the object to its corresponding approximate coordinates.

4.2 Instruments

The instruments also share many commonalities in their presentation and behavior, so generally the same image-processing approach could be applied to each. However, the need for finer resolution combined with the presence of smaller features, more clutter, interference and distortion, and a lack of pilot-provided tracking objects, proved more challenging. Unlike the controls, attaching anything to the needles was not an option because they are in sealed glass cases. Even worse is that both the needles and the information on the instruments are usually presented in the same white on black. Due to space limitations, this section summarizes the general process that applied to all the instruments. Each instrument also had its own positive and negative aspects and idiosyncrasies to accommodate.

Interpreting the instruments first involved knowing where they were. The camera responsible for this perspective was mounted in the middle of the cockpit facing forward (see Figure 1b). The top and left edges of the instrument panel form a high-contrast fixed reference that helped automatically establish the exact scale and bounds of the instrument region, which then established the position of the instruments, as in the top row of Figure 7. Next came contrast normalization to improve the boundary between the needles and the background. This process involved redistributing the histogram representation of the colors in use over the entire range available, thereby spreading similar colors farther apart. The standard luminosity method then converted these new colors to grayscale. Tests showed that under normal conditions, the needles were (by design) almost always the most prominent feature. The primary color value of the needle thus became the binary threshold by which all pixels were finally converted into either pure black or white, as in the bottom row.

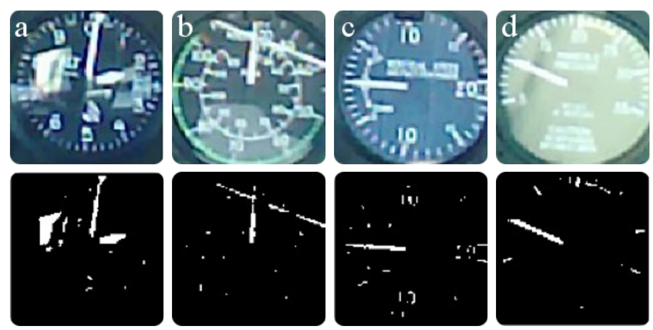

Figure 7: Original and binary-thresholded images

The needles are normally the most prominent linear features, called blobs. Glare can produce artifacts, but the shape does not normally lend itself to recognition as a line, as in Figure 7 (a). When it does, as in (b), the needle still tends to be larger, as well as in its expected position and a legal orientation. In the event that no line is found, two relaxation methods take over. The first incrementally erodes the image in an attempt to break up congealed features until the needle is present. If this attempt fails, then the opposite occurs to dilate the image to join separate features until they form a blob. If both fail, no instrument data are recorded for this frame.

Running a best-fit line approximation on the blob produces an angle, which maps to the predefined numerical scale on the instrument dial for the state value to record. In the case of the altimeter, there are two needles to isolate. (The triangle for tens of thousands of feet was not considered because no flight tests took place so high.) When the needles are far enough apart to differentiate, the process is identical. When they are fused, however, the centroid of this superblob still suffices. Later cleanup for machine learning could interpolate from the last known separate values, but in practice, this occlusion (which also happens to pilots) is not an issue. The altimeter is not precise enough anyway.

5 Calibration and experiments

For reliable, repeatable measurements of the controls and instruments, each in-flight testing session required an initial calibration stage to ensure that the same states mapped acceptably close to the same values. In fact, calibration was actually necessary before *and after* each session to verify that no changes occurred from vibration. This calibration qualified as experiments in its own right because it permitted comparison of the actual values to the expected.

5.1 Static experiments

It is extremely difficult to establish a set of ground-truth states during real flight maneuvers because the dynamic operating environment is so messy; i.e., the expected values are not precisely known. The sensors provided some capability for cross-checking, but their coverage was limited. To establish best-case baseline performance, the initial tests were static on a non-operating helicopter.

5.1.1 Controls

The pedals travel along a known arc with three natural calibration points: full forward, full backward, and half way, which is straightforward to determine because both pedals are adjacent. Similarly, the collective has full down and up positions. A vertical calibration jig with known angles printed on a poster board established intermediate points. The cyclic, however, was troublesome. Its range of two-dimensional motion is over a large horizontal plane whose limits exceeded the field of view of any single camera. Data collection in flight was not affected because the cyclic rarely reaches these limits; however, calibration did need them, or at least an equivalent. To this end, a similar jig with a rectangular internal cutout established the known limits for the center post, which also established the neutral center position. However, positioning the jig itself was tricky because there are few convenient fixed reference points in the cramped cockpit. This process looked ridiculous because it involved aligning small stick-on bubble levels and a lot of contortion, but it was actually effective.

5.1.2 Instruments

Static calibration of the instruments was far more limited because there is almost no access to their needles. Only the altimeter has a knob that changes the internal value, but its range is limited to a thousand feet or so. Instead, the simplest approach proved most effective: color printouts of the instruments to scale with known needle positions taped over the actual instruments. While this approach did not account for the visual disturbances covered in the next section, calibration would be inappropriate under such suboptimal conditions anyway.

5.2 Dynamic experiments

The dynamic experiments involved a breadth and depth of representative flight maneuvers. The purpose was to test the data acquisition system, not to collect actual data for machine learning. It was therefore not necessary to demonstrate more than one acceptable representative exemplar of each. Data for machine learning actually requires many such samples for filtering, smoothing, averaging, fusion, complex statistical analyses, etc. beyond the scope here.

The first set involved airborne maneuvers. They exhibited relatively large changes in the inputs and instruments:

- straight and level at a constant speed, accelerating, and decelerating
- straight with a shallow climb at a fixed climb rate
- straight with a steep climb
- straight with a shallow descent at a fixed speed
- straight with a steep descent
- straight with shallow sinusoidal climbs and descents
- right turn level
- right turn with a shallow climb
- right turn with a steep climb at a fixed speed
- right turn with a shallow descent
- right turn with steep descent at a fixed descent rate
- a left rectangular runway traffic pattern: taking off, climbing, leveling off, descending, and landing

The second set was near the ground with small changes:

- stationary hover
- pivot turn: stationary while rotating about the *z* axis
- square, circle, and figure 8: always facing forward, and always facing the center of the shape

6 Results and discussion

Evaluating results in terms of the agreement between actual and expected values is difficult when the former are messy and the latter are not definitively known. There was significant variation in the experiments caused by pilot error and uncontrollable conditions like wind, as well as measurement errors in the sensors themselves. This paper focuses on the raw image acquisition, not on their complex postprocessing into cleaner form, so the discussion of the results is mostly subjective. To mitigate biases, however, there were several complementary approaches.

6.1 Qualitative internal validation

Qualitative validation involved the pilot reviewing the results of each test in a form that consolidated the inputs and outputs into a meaningful representation. The instrument panel in Figure 8 shows the results derived from the image processing and the sensors [7]. Validation in this form is based on whether the instrument view internally from the pilot's local perspective in the virtual cockpit was consistent with performing the maneuvers correctly. Here the low-level data combine with the high-level knowledge and wisdom of the experienced human pilot to make informed interpretations and evaluations.

Figure 8: Virtual instrument panel and control indicator

Also available were the underlying raw values strategically plotted in Excel in Figure 9 to show relationships. The exact values are not so important as the trends and behaviors. For example, discontinuities and abrupt jerks inconsistent with regular flight can be attributed to acquisition errors because no exemplars with such events were used.

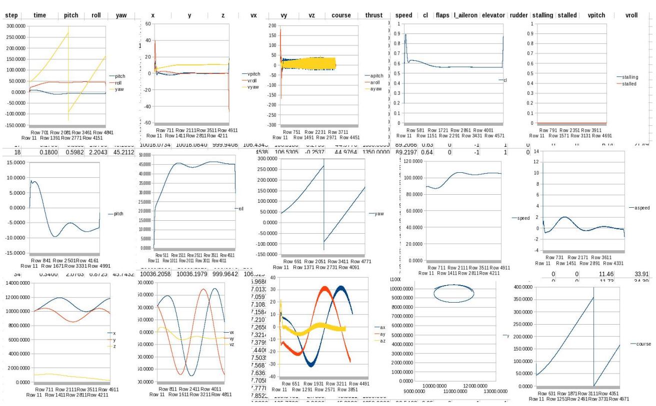

Figure 9: Excel data plots

6.2 Qualitative external validation

Qualitative external validation was also from the pilot's perspective, but from outside the cockpit. Plotting this global representation in two and three dimensions, along with helpful metadetails, as in Figure 10, provided another valuable consistency check [8]. However, unlike the internal view, a pilot is not ordinarily familiar with interpreting the cause-and-effect relationships of inputs to outputs this way. Still, the same kinds of discontinuities would be apparent.

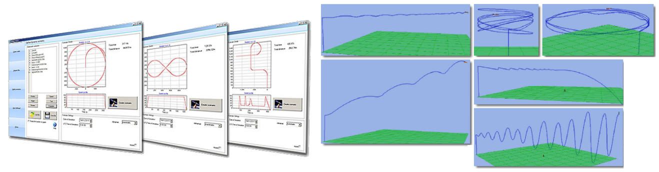

Figure 10: 2D and 3D visualization

Finally, for a richly integrated perspective, the position data exported directly to Google Earth, as in Figure 11.

Figure 11: Google Earth track

6.3 Quantitative validation

Quantitative validation was as objective as possible given the current limitations of this prototype system. The first set of tests was a variant on the calibration process, in which the image processing analyzed color printouts with known needle positions. These tests were static because there was no way to change the needle positions without substituting another image by hand. The second set involved dynamic tests in flight where the sensors provided a reasonable approximation of the expected values. However, over time these sensors had an unfortunate tendency to drift out of calibration. They could not be recalibrated in flight, so usually only the earlier tests in a session were reliable.

6.4 Observations

The weakest link in the image processing is the cameras. On the positive side, consistency in accurate positioning was not a factor because the postprocessing successfully isolated features and produced comparable results from any reasonably similar position and perspective. Likewise, the endless vibration inherent throughout all tests surprisingly played no significant role (except in gradually nudging cameras out of alignment at times). The frame rate was high enough to capture redundant images that effectively canceled it out. On the negative side, the dynamic range of the camera sensors is poor and tends to smear colors and especially wash them out toward the low and high ends of brightness, which degraded contrast. The small lens likely contributed to this problem, which means that higher resolution alone would probably not be an improvement.

Object tracking for the controls was very effective. Except in cases where the orange square was completely washed out, the postprocessing generated positions that were within a few percent of the believed expected values. Still, for the larger project, this resolution turns out to be problematic because much of flying a helicopter involves subtle control movements — very often just pressure, not even overt movement. Large movements are comparatively rare, especially in ground maneuvers, which are the ones of greatest interest. Similarly, manual inspection shows that there is noticeable backlash (slop) in the cyclic, especially vertically, which means that small movements do not always translate into actual inputs. The size of the tracking square also plays a role: larger area is easier to follow, but it requires more movement before the image processing perceives it. Poor contrast causes the edges to appear to change, which affects the centroid that translates to the position value. Averaging multiple frames helps stabilize the raw values by smoothing them, but it simultaneously smooths away desired movements and hides actual changes until they become larger.

Needle localization was also very effective in most nonpathological cases, isolating 92 out of 100 frames on average. Translating needle positions to absolute values was always within 2.9 degrees of the expected values, with 82% being within 1.5 degrees. This error is completely acceptable because the instruments themselves are not this precise. The only influences that could not be overcome were lens flare and glare from sunlight, but even the human pilot was rarely able to interpret such cases.

7 Future work

From the standpoint of technology, better cameras would improve the results. FlyCam now offers (at a much higher price) an HD 1080P version with a larger lens, which appears to be a drop-in replacement for the ones used here [11]. Even better would be to reduce the complexity of having many cameras and use one 4K high-resolution GoPro with a fisheye lens. This approach would require another stage of image processing to correct for the spherical distortion, which could introduce its own issues, but the idea seems promising. Likewise, using better sensors and more of them for error detection and correction would provide better baseline data for performance evaluation.

Future work on the methodology and testing will involve a much richer breadth and depth of experiments, as well as many repetitions of them. This effort could lead to improved results and a better mechanism for quantitatively evaluating accuracy and precision.

Finally, future projects could involve the yaw string for actual data about the aerodynamic behavior of the helicopter in flight, which the sensor-derived approach only approximates. Similarly, using the view of the outside world could contribute to machine learning of hovering based on visual references, which is the hardest part of learning to fly for a human. Finally, despite the R22's predominant role as a training helicopter, it has few features that help the student recognize when they are doing something wrong. In particular, the maximum manifold pressure is extremely easy to exceed when a student is focused/fixated on other activities. (This condition does not cause instantaneous destruction of the engine, but it does reduce its operational life over time.) A simple warning tone based on an automated observation of the gauge would be helpful. Other such conveniences are also likely possible.

8 Conclusion

This proof-of-concept work successfully showed that an array of inexpensive cameras can collect data from complex control inputs and instrumentation outputs. The architecture met all the safety and performance requirements, although the latter could use improvement from better cameras. The image processing was able to isolate features reliably and translate their states into numerical form for later use in machine learning.

9 References

[1] Adapted from oe-xam-simulators.blogspot.com.
[2] Adapted from wikipedia.org and turbosquid.com.
[3] Adapted from X-Plane flight simulator, x-plane.com, last accessed Mar. 12, 2016.
[4] Coates, A., P. Abbeel, and A. Ng. *Apprenticeship Learning for Helicopter Control*. CACM, vol. 52, no. 7, pp. 97–105, 2009.
[5] Hempleman, M. *Image Processing for Machine Learning of Helicopter Flight Dynamics*. Masters thesis, Eastern Washington University, 2015.
[6] Liem, R. *Surrogate Modeling for Large-Scale Black-Box Systems*. Masters Thesis, MIT, Sep. 2007.
[7] Tappan, D. and M. Hempleman. *Toward Introspective Human Versus Machine Learning of Simulated Airplane Flight Dynamics*. 25th Modern Artificial Intelligence and Cognitive Science Conference, Spokane, WA, Apr. 26, 2014.
[8] Tappan, D. *A Pedagogy-Oriented Modeling and Simulation Environment for AI Scenarios*. WorldComp International Conference on Artificial Intelligence, Las Vegas, NV, July 13–16, 2009.
[9] Walter, L. "Yearly UAV Market Exceeds $11B." Aviation Week & Space Technology, Aug. 18, 2013.
[10] www.beagleboard.org, last accessed Mar. 17, 2016.
[11] www.camonetec.com, last accessed Mar. 17, 2016.
[12] www.chrobotics.com, last accessed Mar. 17, 2016.
[13] www.opencv.org, last accessed Mar. 17, 2016.
[14] www.parallax.com, last accessed Mar. 17, 2016.
[15] www.robinsonheli.com, last accessed Mar. 17, 2016.