

# Characterising latency for AO optical sensors: an implementation

Thomas Dixon<sup>a</sup>, Francis Bennet<sup>b</sup>, Ian Price<sup>b</sup>, and Francois Rigaut<sup>b</sup>

<sup>a</sup>University of New South Wales, Sydney, Australia

<sup>b</sup>Research School of Astronomy and Astrophysics, Australian National University, Canberra, Australia

## ABSTRACT

The latency of electro-optical components is of high importance in the design of Adaptive Optics systems, as it limits the performance of the control loop. There exists a need for a latency measurement method that can be constructed with simple components found in most Adaptive Optics labs that still provides a measurement accurate to sub-microseconds. Through a combination of research and experimentation, potential methodologies were investigated with the aim of producing reliable latency measurements. This document will discuss one such method, involving coupling a LED pulse output and detected pulse input signals to the same clock for easy comparison. For this method, a proof-of-concept was developed using MATLAB and small analogue electronics, and the performance characterised. This characterisation showed that although there is some merit to the method, improvements are necessary to increase the precision of the measurement to a level usable in Adaptive Optics systems.

**Keywords:** Latency measurement, LED Pulse, Control System Latency, Adaptive Optics, Closed Loop

## 1. INTRODUCTION

The performance of control loops within Adaptive Optics (AO) systems is highly sensitive to the latency of their components.<sup>1</sup> For optical sensors employed in AO, delay between the end of an exposure and that exposure's data becoming available directly influence their effectiveness, particularly in cases where fast moving objects are being tracked, such as satellites and space debris.<sup>2</sup> The variability in this delay (jitter) is also of concern. However, for many Commercial Off-The-Shelf (COTS) detectors, this latency is not published with other technical specifications,<sup>3</sup> instead requiring direct measurement. No simple, clear method of measuring camera latency with simple elements is available, and as such the development of a generic latency measurement method able to be constructed with simple components is of merit.

This document discusses the development of a concept for a low-complexity latency measurement system, as part of a student project at the Research School of Astronomy and Astrophysics (RSAA) at The Australian National University. This system uses a LED controlled via PC-serial output to stimulate the sensor, the output of which is compared with LED-trigger time stamps to generate a latency estimate. Results of a successful proof-of-concept of this method, as well as an outline of refinements needed to improve the latency estimation conclude the body of this document.

Within the AO control loop, output from a Wavefront Sensor (WFS) is used to control a deformable mirror and correct for atmospheric aberrations. The effectiveness of this correction method is dependent on the speed and latency of the control loop, of which WFS latency is a factor. The RSAA are designing and building compact AO systems for space situational awareness including satellite imaging. Such AO systems have particular challenges requiring high WFS frame rate, making the closed loop latency critical to performance.<sup>4</sup> For these systems, it is necessary to estimate latency with a maximum error of 10-20 microseconds; this is deemed the 'upper threshold' of measurement uncertainty. The time stamp resolution for typical cameras used in AO WFS is on the order

---

Further author information: (Send correspondence to T.D.)

T.D.: E-mail: thomas.dixon@unsw.edu.au

F.B.: E-mail: francis.bennet@anu.edu.au

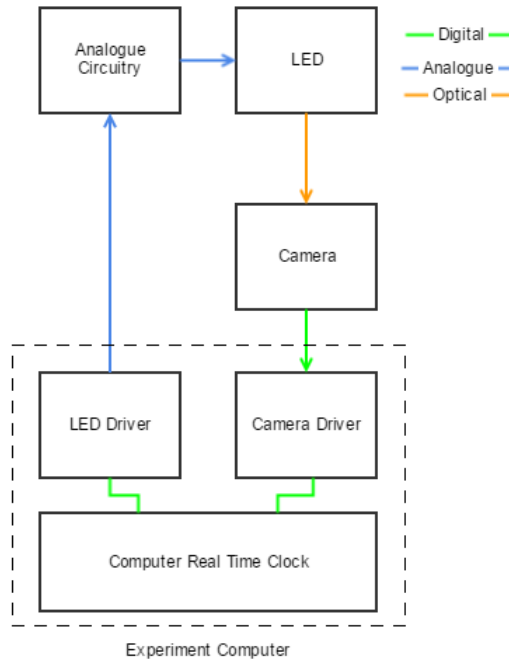


Figure 1. Method Apparatus Block Diagram

of one microsecond.<sup>3</sup> This is taken as a useful lower bound of delays-of-concern; any latency measurement uncertainty below this point is below the measurement threshold of the sensor itself, and could not be assumed to have an impact on the latency characterisation.

## 2. MEASUREMENT METHOD

This measurement method involves pulsing a LED using a computer, receiving the pulse through an optical device connected to the computer, and evaluating the latency through a comparison of timestamps. This section will outline the methodology in detail, demonstrating additional modifications that can improve the resolution of latency measurement. A proof-of-concept measurement apparatus was completed, and results gathered from this will be discussed.

### 2.1 Method Outline

The proposed method is a relatively simple arrangement of digital and analogue components. A computer sends a pulse command to the LED through a serial interface, which is conditioned through a small analogue circuit to drive the LED. The LED is positioned facing the camera to be characterised, which detects the pulse. A block diagram of the method apparatus can be seen in Figure 1. A camera driver within the PC continually samples input and stores the frame(s) within which the pulse is captured, timed using the same clock as the LED driver. Post-processing of metadata (timestamps of commands and of received frames) is used to determine the system latency.

#### LED Control

LED Control is achieved through the use of the Request To Send (RTS) control pin on a standard COM (serial) port. Standard linux drivers can pull this pin from -12V to 12V at a rate on the order of a single microsecond, making it an ideal method for the control of the LED. Fluctuations in serial line voltage (usually between +/- 9 and 12V)<sup>5</sup> require the RTS output signal to be conditioned before driving the LED.

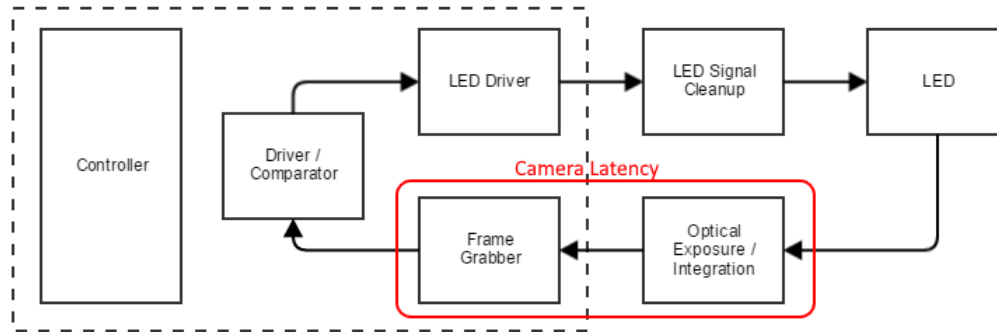


Figure 2. Refinement of the latency estimate can be achieved by accounting for the delay components of the measurement apparatus. The elements outside of the red box (which indicates the latencies that this technique is to measure) add inaccuracy to the measurement and should be characterised.

### Experiment Control

The controlling software for latency measurement is split into two functions; calibration and experimentation. In the calibration stage, the integrated intensity of an image with the LED on and off are compared and a midpoint evaluated. This midpoint is then used in further functions as a threshold value, beyond which the LED is said to be 'on'. This calibration stage was implemented to avoid issues with setup of apparatus. The threshold value can also be derived as a function of noise, where the 'LED off' input is measured for a series of exposures and standard deviation calculated. Setting a threshold value of  $5\sigma$  (lower than the on/off midpoint) will yield a more accurate measurement for latency as the calculated pulse position will be closer to the true pulse start point.

Within the experimental stage, exposures are taken through an image acquisition tool, while the LED is pulsed and a timestamp of that action recorded. Following the frame capture, the integrated intensity of each exposure is computed and the timestamp of the first frame to cross the threshold value compared to the LED pulse timestamp to generate a latency estimate.

### Improvements to Latency Determination

Comparing the timestamps of the 'pulse sent' and 'pulse received' events in the method generates a coarse estimation for latency on the order of one frame, seen below in Section 2.2. This estimate can be improved in two ways; accounting for the system delay and exploiting the bleed of the pulse across two frames to more accurately determine its position.

#### *System Delay*

The delays present in the measuring equipment itself are a factor that affects the accuracy of latency measurements. In this method, the system delay is evaluated piece-wise, for each of the core components of the apparatus (Figure 2). The cumulative system delay (which can be summarised as a delay between the 'LED on' command and the LED actually turning on) is then subtracted from the 'LED on' timestamp to improve the system latency estimation. Values for these delay parameters for the proof-of-concept setup are discussed in Section 2.2.

#### *Integration Ratio (Pulse Bleed) Pinpointing*

The timing of pulse reception can be improved by identifying where in the exposure the pulse initiates. If the LED pulse is timed to occur with a duration of more than one frame, it will track across two exposures. The pulse's true position within the two frames can be computed using relative integrated intensity; in essence, depending on where the pulse actually occurred, the measured cumulative intensity across the two frames will be different.

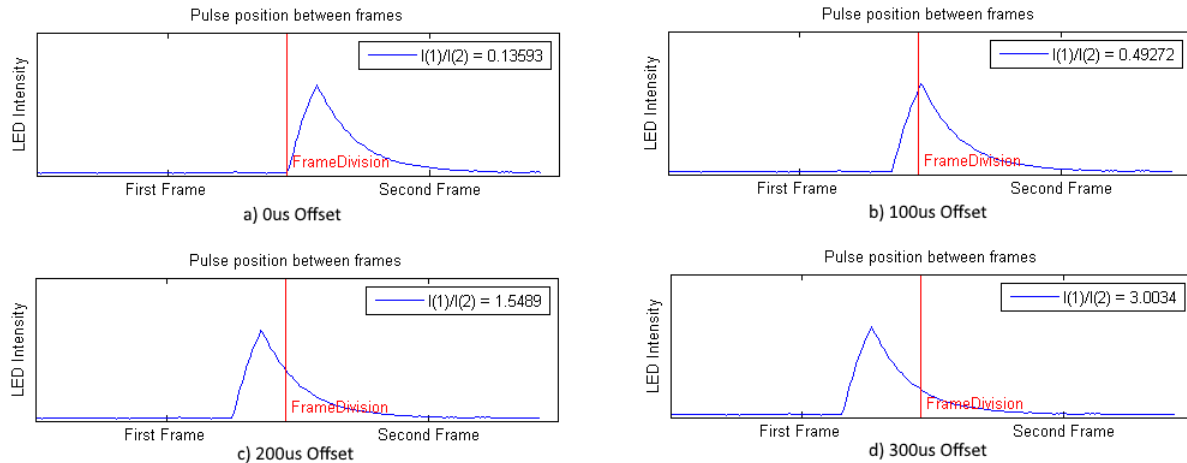


Figure 3. By simulating the expected counts across two adjacent frames as a function of pulse position, the ratio of integrated intensities for a captured pulse can be calculated, which can later be used to determine true pulse position.

Firstly, comparative intensities as a function of pulse position are to be calculated. Figure 3 shows a series of these comparisons for a LED characterised in an older apparatus configuration, with integration ratios in the top-right corner. While the exponentially rising and decaying pulse provides adequate differentiation between different positions, a square pulse with high rise and fall times would be preferable. This characterisation should occur separately from the latency measurement setup, using a high-frequency optic sensor to generate a good quality characterisation of the LED pulse apparatus.

Following this characterisation (which need occur only once, prior to placing the LED pulse apparatus in the latency measurement loop), the measurement is then run as normal, with the frames containing the pulse undergoing further analysis to compute the ratio of intensities. This value is compared to the previously calculated ratios and pulse position determined. The true pulse position is then used to further refine the latency estimate.

## 2.2 Proof-of-concept

To demonstrate the validity of the method in determining optical device latency, a proof-of-concept apparatus was set up and utilized to determine the latency of two webcams. Control software was written in MATLAB; although other methods would yield a lower system delay, MATLAB's ease of operation and ability to interface with both the serial-LED controller and cameras available made it ideal for use in a demonstrative experiment. It is of note that MATLAB's image processing toolbox itself adds a latency that cannot be separated from camera latency; measurements will include the time taken for the MATLAB frame grabber to retrieve and store frame data in an available dataspace.

### System Delay Estimates

The System Delay parameters (Figure 2) were estimated as follows;

- *LED Driver* - The delay of the LED driver code in MATLAB was evaluated by using the 'tic' and 'toc' commands, placed at the start and end of the LED driving code and run for 100 iterations. The measured delay was 0.035s +/- 0.01s (one sigma).
- *LED Interface Circuit* - Evaluated using an oscilloscope across two channels; one reading the voltage across the LED, and one reading the driving voltage of the interface circuit. No discernible delay was measured.
- *LED* - A plot of LED response to a 200 microsecond square wave pulse can be seen in Figure 4. As can be seen from the figure, although the LED response is not a square wave, the initiation of rise and of decay correspond with the edges of the square wave, meaning that a small correction to the integration ratio pinpointing can remove this delay component entirely.

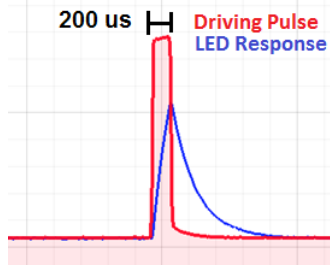


Figure 4. For pulses in the duration of interest, LED response to square wave input is distorted. However, the start and end of the pulse rise correspond to the rising and falling edge of the square wave, meaning that a small calibration correction can remove the delay effect of the LED on latency measurement.

Table 1. Measured latency distribution for each device, and mean frame duration in each mode.

Camera	Integration Time (Frame Duration)	Latency Mean	Latency Standard Deviation
Camera1 (inbuilt camera) @ 30fps	60.6ms	137ms	7.25ms
Camera2 (external camera) @ 30fps	45.5ms	86.9ms	9.43ms
Camera2 (external camera) @ 15fps	53.2ms	81.1ms	13.4ms

## Results

Data from 100 iterations of latency measurements for two cameras can be seen in Figure 5, each fit to a normal probability distribution function. MATLAB timing uncertainty ( $\pm 0.01s$ , Section 2.2) is visualised through error bars. A distinct mean latency was observed for the different devices independently of frame rate, with 'Camera1' (an inbuilt webcam) possessing a mean latency of 137ms, while 'Camera2' displayed a mean between 81 and 87ms, depending on acquisition rate. These mean latency values, standard deviations, as well as integration times for each configuration can be seen in Table 1.

The distribution of results show a rather high variation in measured latency compared to the 10-20 microsecond limit discussed in Section 1, with standard deviations between 7 and 14ms. As stated above (Sec 2.2) it is not possible to decouple camera variability with uncertainties induced by the measurement apparatus; the image acquisition tools in MATLAB and the use of a USB-Serial as opposed to recommended COM port will have increased the standard deviation of these results.

However, these results still permit comparative analysis of camera latency. From Figure 5 and Table 1, it is clear that Camera1 has a higher latency than Camera2, despite the uncertainties inherent in the measurement setup. Similarly, the 30fps and 15fps settings for Camera2 exhibit different latency profiles, although any direct comparison is still within the margin of error.

## 2.3 Further Work

Following the proof of concept, further work is required to improve the accuracy of this measurement technique. Primarily, this work concerns the implementation of integration ratio pinpointing; this refinement technique was not able to be employed in the proof of concept method.

### Pulse Bit Standardization

Utilization of integration ratio pulse location requires a pulse duration that is consistently on the order of one exposure. In the proof-of-concept, the pulses generated lasted more than 10 exposures due to the slow execution speed of the computer and software (MATLAB) used. While full-scale utilization of this technique should employ faster software and a better hardware connection (COM ports as opposed to the USB-Serial that was available), it is foreseeable that a purely software based LED control may generate a pulse length too great or too variable

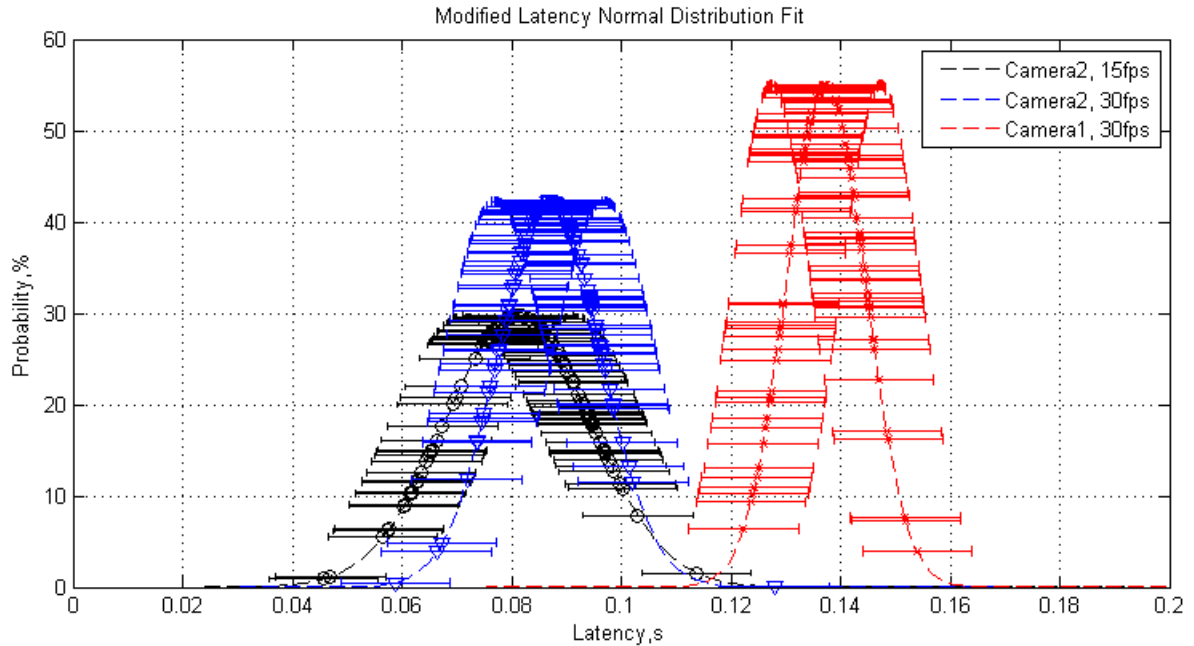


Figure 5. Normal probability fit of Latency Distribution for two cameras. The substantial majority of values are within one frame's duration for each camera - this proof of concept method is at least viable for a 'number of frames' delay. Error bars account for the uncertainty in the LED driver.

for the effective characterisation of a particular sensor. A hardware solution for standardizing the pulse bit is recommended; a system whereby the pulse is 'cut off' a set amount of time after rising, either through use of a separate, small time scale timing chip or analogue components with a known delay.

### Integration Ratio Pin-Pointing Resolution

The resolution of the integration ratio technique used to estimate true pulse position can be easily computed. For a square pulse  $\Delta t_{5\sigma}$  - the  $5\sigma$  resolution (necessary movement in pulse position to be distinguishable from a  $5\sigma$  noise deviation) can be derived as follows;

$$N_{mean} * t_e + I_{Pn} * \Delta t_{5\sigma} = (N_{mean} + 5\sigma_N) * t_e$$

$$\Delta t_{5\sigma} = \frac{5\sigma_N * t_e}{I_{Pn}}$$

Where  $N_{mean}$  is the mean noise intensity,

$I_{Pn}$  is the intensity of the pulse above the noise mean, and

$t_e$  is the exposure time.

For a non-square pulse, such as the one characterised in Figure 3, the resolution of the pulse will differ along its length; the  $\Delta t$  required to achieve an integrated intensity change greater than  $5\sigma * t_e$  will vary depending on the value of  $I_{Pn}$ . In this case, the resolution must be characterised numerically. For the pulse in Figure 3, the minimum resolution (occurring at the trailing edge, where the gradient of intensity is small) was found to be 190ms, while the maximum resolution (occurring at the leading edge, where the gradient is quite steep) was found to be 50ms. This variation in resolution highlights the need for a square LED pulse when generating latency estimates.

### 3. CONCLUSION

The method outlined above has demonstrated viability as a simple solution for measuring the latency of optical devices. By using the same clock to control both LED pulse output and sensor input, this method avoids the need for synchronization of different components and offers latency measurements at a reduced complexity.

A proof-of-concept characterisation was performed on two non-scientific cameras, with results clearly showing one device to have a lower mean latency than the other. Variability in this latency (jitter) was similar across devices. Comparisons between different acquisition speeds on the same device did not yield highly differentiating results, as uncertainties inherent in the measurement apparatus and methodology overshadowed any discernible difference in latency. Proposed improvements to the latency measurement method would increase the accuracy of the result, allowing a more characterisation and comparison. One such proposed improvement is the utilization of integration ratio pinpointing to distinguish the true position of the LED pulse within the exposure(s) that it illuminates.

### REFERENCES

- [1] "Impact of latency and jitter on the performance of adaptive optics systems for elts." [Online] <https://www.eso.org/sci/meetings/2012/RTCWorkshop/12fedrigo.pdf> (2012).
- [2] Bennet, F., DOrgeville, C., Gao, Y., Gardhouse, W., Paulin, N., Price, I., Rigaut, F., Ritchie, I. T., Smith, C. H., Uhlendorf, K., and Wang, Y., "Adaptive optics for space debris tracking," *Proc. SPIE* **9148**, 91481F–91481F–9 (2014).
- [3] PCO, "Pco edge 4.2 product specifications." Online: <https://www.pco.de/support/manuals/scmos-cameras/pcoedge-42/>.
- [4] Tyson, R., [*Adaptive optics engineering handbook*], vol. 67, CRC Press (1999).
- [5] Strangio, C., "The rs232 standard." Online: <http://www.camiresearch.com/Data Com Basics/RS232 standard.html> (1993).