

# 1 Introduction

There have been a succession of clocks in Trinity Great Court since 1610 [1], with the latest one approaching its 100th anniversary next year. This current mechanism is a beautiful piece of engineering; it can still keep time quite accurately, to within 1 second per month, powered by falling weights and regulated by a pendulum. It is unusual in that it strikes the hours twice (according to legend, once for Trinity and once for St John's); it makes an appearance in William Wordsworth's poetry<sup>1</sup>, and it is famous as the timekeeper for the "Great Court Run."

Although the clock is capable of keeping time quite accurately, it does need adjustment. However, knowing instantaneously what adjustment is required is a bit of a challenge: even a relatively large error in the rate of the clock of, say, 1 second per day amounts to only an extra  $35\ \mu\text{s}$  in every swing of the pendulum. Since this is rather small, traditionally the clock keeper would wait until the end of the week, when the error will have built up enough to be measurable by comparing the clock with a radio-controlled wall clock. Then the clock can be adjusted by adding or removing small weights on the pendulum.

It would be preferable if the rate of the clock could be measured, and corrected, without waiting for the error to accumulate to noticeable levels, and this is the problem which past work on this project has aimed to solve. A previous 4th year project developed a system to measure the "going" (the rate of the clock, i.e. seconds gained per second) from both the sound of the ticks and an optical sensor on the pendulum [2]. This project builds on that work, aiming to produce a finished monitoring system which is accurate enough to measure all the variations in the clock's running, and reliable enough to run continuously for months on end.

The ability to instantaneously measure the "going" of the clock will make keeping the clock on time much easier. Using this ability, we can make a set of calibrated weights to

---

<sup>1</sup>William Wordsworth, *Residence at Cambridge*, The Prelude (1850), Book Third.

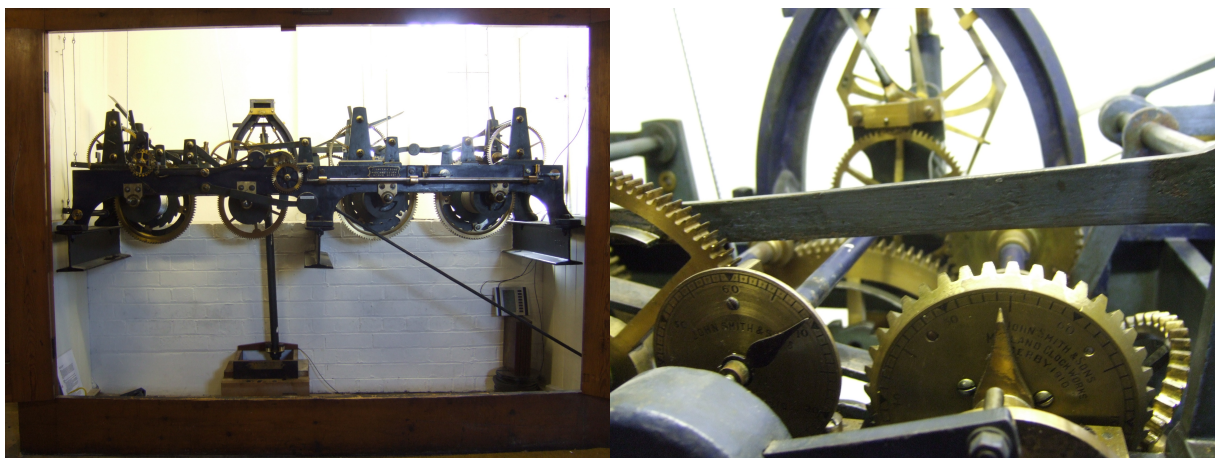


FIGURE 1.1: The Clock

replace the assorted washers currently in use; with these two developments, if the clock is running slow or fast the exact correction can be applied immediately.

But there is a whole other side to the project: when one starts to look at the resulting data, there are a multitude of interesting physical effects happening. Changes in going can be attributed to variations in pendulum amplitude, air density, temperature, air currents... even the gravitational pull of the Moon, as in the title of the project. To borrow a phrase from Nelson and Olsson [3], we can explore ‘rich physics from a simple system.’ The aim is to make a fairly comprehensive analysis of the types of behaviour arising from these effects. We can then draw on the recorded data from the clock to demonstrate these behaviours in the real world.

Also, a means of organising the fairly large quantities (one reading per tick) of data is needed. The aim here is to produce a website where the data can be browsed online, and selected parts downloaded. This will be useful not only to those studying or looking after the clock, who will be able to check its status from anywhere, but also will open up the rich physics of the pendulum to anyone in the world, turning the Trinity Clock into a public ‘pendulum laboratory.’

In summary, the aims of the project are to:

1. Finalise the instrumentation and techniques for measuring the clock, resulting in a reliable set-up which can monitor the clock indefinitely;
2. Make the data freely available on a website for investigation;
3. Aid the clock keeper’s work by, for example, producing calibrated regulation weights;
4. Model the behaviour of the clock in response to environmental variations, and compare against the real-world data.

## 2 Measurement Apparatus

### 2.1 Reference time source

Throughout their history there have always been methods of checking the running of clocks, ranging from sundials to modern atomic oscillators. For this project, the accuracy of the measuring system must be quite high: for example, a typical situation would be measuring the clock over the period of [ideally] uninterrupted running between the start and end of Daylight Savings Time in March and October. Over these 7 months, we might expect the clock to stay within 10 seconds of the true time, which represents an accuracy of 0.5 parts per million (ppm). The reference time source must, of course, be better than this.

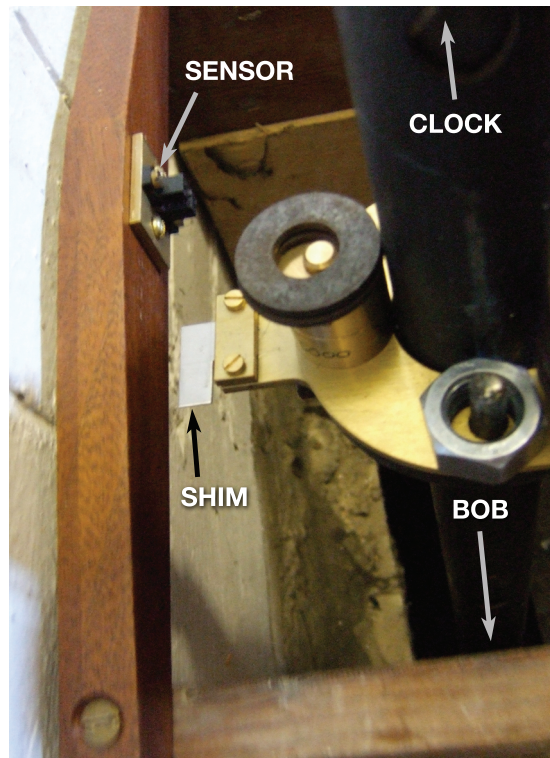


FIGURE 2.1: Photo of optical sensor and shim on pendulum.

A simple quartz oscillator was tested, but was not nearly good enough (it ran  $\sim 5$  seconds per day, or 60 ppm, too fast). A better time reference is the Global Positioning System (GPS). The GPS satellites have very accurate clocks, required for position fixing, and the receiver used<sup>1</sup> gives an pulse output with an accuracy of  $\pm 1$  microsecond [4]. Although  $\pm 1 \mu\text{s}$  is only  $\pm 1$  ppm, unlike the quartz oscillator it is within 1 ppm of true time *every second*. Over 7 months the accuracy will be  $10^{-7}$  ppm! At least 3 satellites must be visible before the time is known accurately, so the receiver is placed on the roof or in a window with a good view of the sky. As well as the pulse per second (PPS) output for timing, the serial connection to the GPS is used for configuration of the receiver and monitoring of the number of satellites visible.

## 2.2 Sensor

The next part of the apparatus is the sensor. Acoustic, optical, magnetic and inductive sensors have all been used to measure clocks [2, 5], but here we have continued to use the infrared (IR) optical sensor used previously.

The sensor is arranged such that a shim on the pendulum breaks the beam twice per period (see Figure 2.1), giving a signal with 4 edges per period; a typical recording of the input signals is shown in the upper part of Figure 2.2. To calculate the amplitude some

---

<sup>1</sup>a Garmin GPS 18x LVC

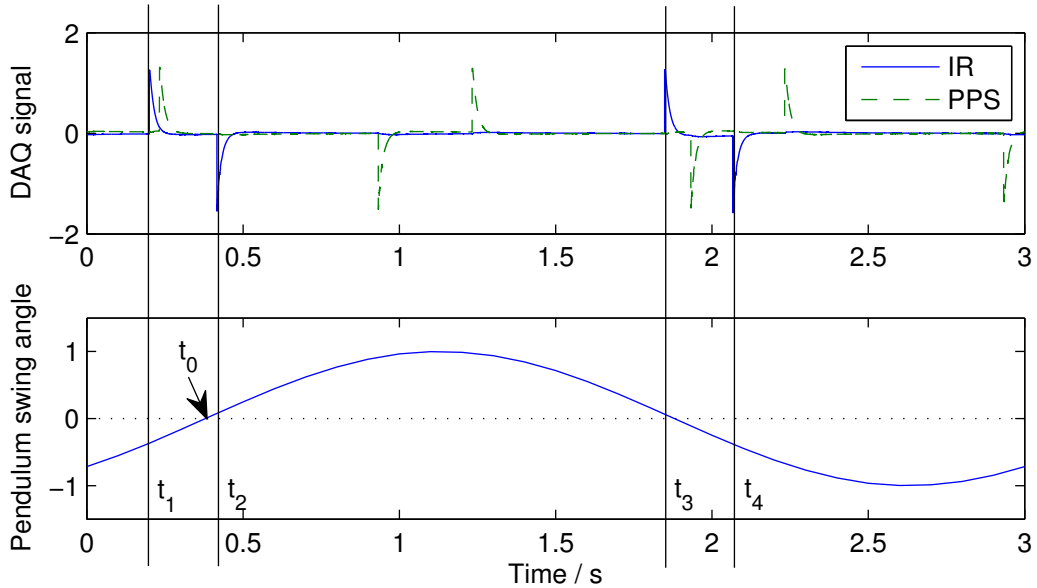


FIGURE 2.2: Typical input signals from IR sensor and GPS (high-pass filtered, above), and corresponding pendulum position (below).

asymmetry is needed in the position of the edges. As the sensor is located at the centre of the pendulum's swing, the shim must be mounted off-centre — the resulting IR edges are related to the pendulum position is shown in the lower part of Figure 2.2.

### 2.2.1 Optimisation of sensor / shim location

The size and location of the shim determines the points in the swing at which the IR sensor is triggered, and these times are used to calculate the amplitude of the pendulum (using Equations (2.1) & (2.2) below).

There will be some uncertainty in the trigger positions (due, for example, to variable ambient light levels), and the sensitivity of the amplitude calculation to these variations depends on the position of the trigger points in the swing, so there will be an optimum shim design. Equations (2.1) & (2.2) are not readily differentiable, so a Monte-Carlo analysis was used to find the optimum design. The resulting contours of error sensitivity are shown in Figure 2.3.

This shows that maximum accuracy is achieved for wide and highly asymmetric shims. The best design would be one which blocks the beam for almost the entire half-swing, allowing a brief pulse through at the top end of the swing. However, the amplitude of the pendulum changes both during day-to-day running and, more significantly, whenever the clock is stopped, intentionally (for example, for an hour at the end of Daylight Saving Time in October) or not. We would like to be able to measure the characteristics of the decaying oscillations, so the system must be able to measure down to, say, 40% of the original amplitude. This means the shim must be smaller than the optimum, and this

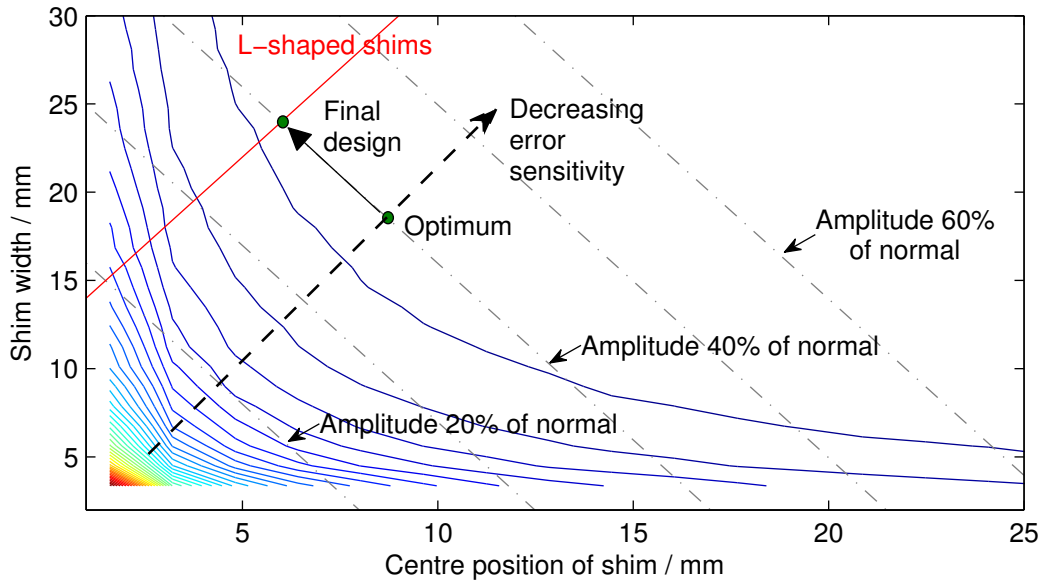


FIGURE 2.3: Shim design: contours of constant error sensitivity, with minimum measurable amplitudes (dashed lines). Final design: width 24mm, centre position 6mm.

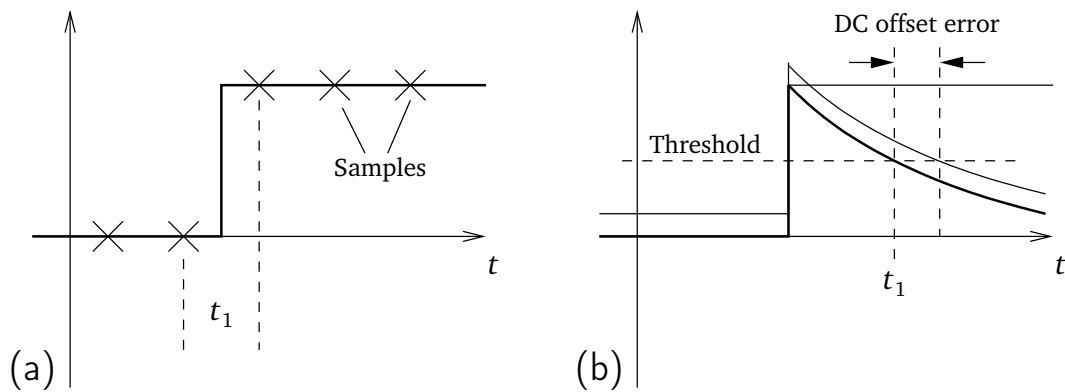


FIGURE 2.4: Locating edges: left, the edge of a square wave can only be found to a resolution of 1 sample. Right, fitting a curve to the exponential decay allows the edge to be found to sub-sample resolution, but any DC offset in the signal must be removed to avoid bias.

constraint is shown by the dashed lines in Figure 2.3.

The final design step was to decide that the simplicity of an L-shaped shim (rather than a more complicated offset rectangle) was worth the small reduction in accuracy shown in Figure 2.3.

### 2.3 Data acquisition

There are two signals to deal with: the pulse-per-second (PPS) reference from the GPS, and the 4 edges per tick from the IR sensor on the pendulum.

The processing of the input data starts with precisely identifying the times at which the edges in the PPS and IR signals occur. The edge of a square wave (such as these) can only

be found to an accuracy of 1 sample, as sketched in Figure 2.4(a). To attain better accuracy, the signals are high-pass filtered to produce a sharp edge with an exponentially decaying tail (Figure 2.4(b); for details of the interface electronics see Appendix A). A curve can then be fitted to this tail, and the time of the edge found by interpolation to sub-sample accuracy. However, the downside to this method is that it is susceptible to variations in DC offset, which cause a shift in the time identified for the edge; care is needed to remove the DC offset for each edge.

A PC datalogger (National Instruments NI-DAQ 6023E) is used to digitise the signals.

## 2.4 Software processing

The starting point for this project was the work of various previous students, resulting in a MATLAB program which acquires data through the PC's sound card and analyses it to give the current drift (clock time minus true time) and pendulum amplitude. Development this year has built on the same underlying algorithm, but the software has been rewritten entirely in C++, which is more reliable and uses fewer computer resources than MATLAB (not to mention avoiding the need for a MATLAB license). Reliability is much improved, and several problems with the original algorithm which created spurious results have been fixed. The move away from MATLAB also opens up the possibility of removing the PC and running the software on a stand-alone microprocessor. A further improvement was to log data from every tick (every 3 seconds, down from every 15 seconds).

### 2.4.1 Algorithm

1. Acquire  $\sim 3$  seconds of data. The datalogger driver records data continuously, and the length of chunk extracted from the acquisition buffer is varied slightly around 3 seconds to keep the chunk aligned with the clock's ticking.
2. Locate the edges in each channel (IR and PPS):
  - a) Locate the edges approximately by looking for crossings of a threshold;
  - b) For each edge, subtract the mean DC offset over a short period before the edge;
  - c) Refine the edge locations to sub-sample accuracy by fitting an exponential curve to the decaying signal. The edge location is taken where the fitted curve crosses a threshold.
3. Calculate drift:
  - a) For consistency, drift is always measured relative to the same point in the pendulum's swing: the first edge of the shim as the pendulum swings left-to-right. The correct IR edge can be determined because the gaps between edges are unequal

due to the asymmetry of the shim (see example edges in Figure 2.2, where the relevant edge is the first one);

- b) Find is the time difference between this reference edge and the previous PPS edge;
- c) This time difference must be ‘unwrapped’ to give drift (because there is a pulse every second, it is only possible to determine drift to modulo 1 second):
  - i. if there is a sudden decrease of more than 0.5 seconds relative to the drift from the previous tick, it is assumed than the time difference has wrapped and 1 second is added;
  - ii. vica versa for an *increase* of more than 0.5 seconds.

#### 4. Calculate amplitude:

- a) The period (and hence frequency  $\omega$ ) of the pendulum is found from the mean of the time differences between corresponding IR edges in the current and previous ticks.
- b) Using the times of the 4 IR edges  $t_1, \dots, t_4$ , work back to find the time when the pendulum crossed the centreline, using the following formula (derived by [2, Appendix 6.5], assuming the pendulum motion is sinusoidal):

$$t_0 = \frac{1}{\omega} \arctan \left( \frac{\sin \omega t_1 - \sin \omega t_2 + \sin \omega t_3 - \sin \omega t_4}{\cos \omega t_1 - \cos \omega t_2 + \cos \omega t_3 - \cos \omega t_4} \right) \quad (2.1)$$

- c) Then the amplitude  $A$  is given by:

$$A = \frac{W}{\sin \omega(t_1 - t_0) - \sin \omega(t_2 - t_0)} \quad (2.2)$$

where  $W$  is the angular width of the shim.

- 5. Write the drift and amplitude to the log file with a timestamp (the number of seconds since midnight on Jan 1, 1970).

## 2.5 Weather logging

Many effects on the clock are environmental, so it is interesting to measure the atmospheric conditions around the clock. The weather station<sup>2</sup> from the previous project was used, placed in the clock enclosure. A program runs once per minute and logs temperature, humidity and pressure.

The temperature, in particular, varies significantly in different locations around the clock, and depending on whether the sensor receives direct sunshine or not, so care is needed about exactly which temperature is being measured. The final location of the sensor is near the pendulum, out of direct sunlight.

<sup>2</sup>A HeavyWeather WS2300, using the Perl software library ‘Device::LaCrosse::WS23xx’.

### 3 Website interface

To ease the viewing and organisation of the large quantity of data collected, a website for the clock has been set up at <http://www.trin.cam.ac.uk/clock>. A screenshot of the main viewing page is shown in Figure 3.1 (although I would recommend visiting the website itself).

The web interface has been implemented in PHP. The main features available through the interface are:

- Browse graphs of data from one or more days;
- Download data, averaged to a selected sampling rate, with selected quantities included;
- View scatter plots, similar to those presented in Section 6;
- Add corrections to the recorded drift, e.g. when the clock is advanced or put back;
- Annotate interesting events with comments.

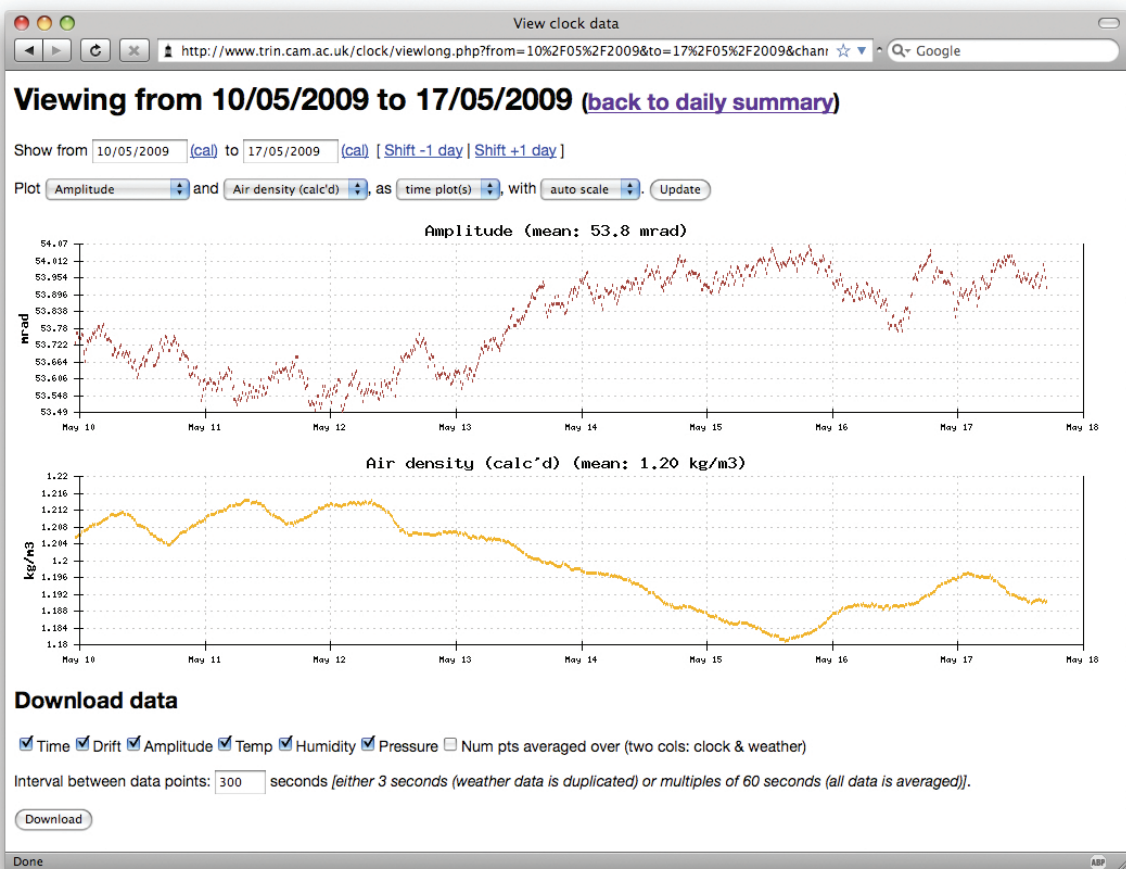


FIGURE 3.1: Screenshot of main web interface for viewing data.