Simulation of Raising & Lowering

Andrew J. Instone-Cowie

Published in the Ringing World, No. 5494 12 August 2016

In a recent article (RW2016 p453), Richard Major suggests alternative approaches to simulator sensing using accelerometers, taking as his starting point the limitations of conventional simulators in teaching raising and lowering.

I had also considered the use of accelerometers in simulator sensing, but concluded fairly quickly that the practical and computational challenges made this a problem for a different day; although I believe that the Belfree project (unfortunately not Open Source) was possibly working along similar lines, and an update from them would be welcome. The article did however prompt me to think again about ways that the teaching of raising and lowering using simulators might be improved by developing existing technologies.

This article summarises the conventional sensor approaches, defines some basic requirements for improvements, suggests an alternative solution, and describes a prototype implementation. All the code and other artefacts referenced are available under Open Source licences as part of the Liverpool Ringing Simulator Project.

Simulators are usually used in the tower in one of two ways: For silent practice, where all bells are rung by ringers and the simulator provides only the sound of the bells; or with a simulated band, where a single bell is rung by a ringer, and the simulator provides the correct timing of the remaining bells as well as the sound. None of the main simulator software packages currently appears to support raising or lowering with a simulated band, and this article is therefore confined to silent practice use.

Before considering the difficulties faced in the simulation of raising and lowering, a brief recap of current typical sensor arrangements may be useful. Installations generally follow one of two patterns. In the first, a fixed sensor with a single detector is activated by two triggers mounted on the bell wheel. These triggers are positioned so that one operates at the point at which the open bell would strike at backstroke, and the other at handstroke. This is illustrated in Figure 1, which shows a dual trigger installation with the bell at the point of striking a backstroke blow (and on its way up to the handstroke position).

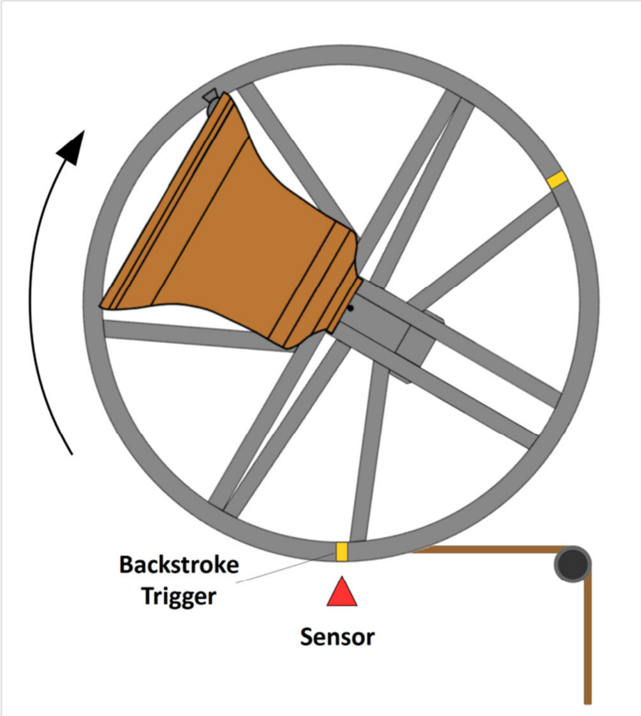

Figure 1

The twin trigger approach has some practical disadvantages which are not directly relevant here, and the more common alternative arrangement has a single trigger which activates the sensor detector as the bell passes through the bottom dead centre (BDC) position. A fixed delay is then applied either by the simulator software or by a sensor signal aggregation device such as a multi-bell interface, after which the simulated sound is generated. This approach depends on the observation (validated by John Norris) that for a given bell, variations in BDC-to-strike interval during normal full circle ringing are too small to matter. This arrangement is illustrated in Figure 2, which shows a single trigger installation with the bell again at the point of striking a backstroke blow.

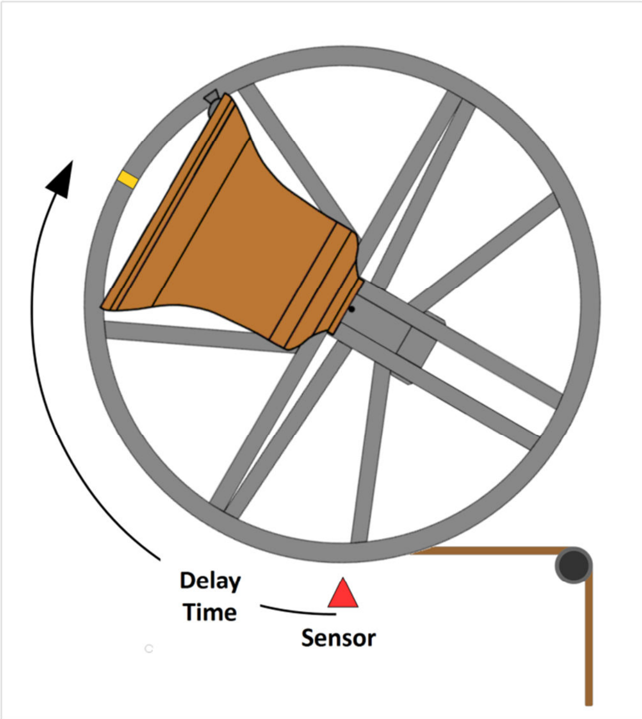

Figure 2

Sensors typically may use phototransistors or magneto-resistive detectors, with reflective tape or rare earth magnets as triggers, but other variations are possible.

The behaviour of a bell being raised or lowered differs from that of a bell being rung full circle in two key ways. The behaviour is asymmetric, so that handstroke and backstroke behave differently, and the behaviour depends on how “high” the bell is, or more properly on the current amplitude of the swing. For example, one of the most obvious manifestations of these differences is that the backstroke ceases to sound at some point during a lower, while the handstroke continues to sound.

It will be immediately apparent that neither of the conventional sensor arrangements can differentiate between handstroke and backstroke, and neither conveys any useful data about the amplitude of the swing. Furthermore, current simulator software packages and interface hardware such as multi-bell interfaces lack the capability to communicate or interpret such data.

The precise behaviour of a real bell during raising or lowering would be complex to model, but can be simplified for the purposes of simulation. The object is to provide simulator behaviour which approximates to that of a typical idealised bell, adequate for teaching purposes, and not to produce a millisecond-accurate emulation of the precise behaviour of any particular real bell, even the bell hosting the simulator sensor.

Clearly an approach requiring the complete re-engineering of all simulator installation and software would be a major undertaking. A simpler approach has therefore been devised, with the following basic requirements:

- The ability to differentiate between handstroke and backstroke,
- The ability to detect changes in the amplitude of the swing of the bell,

- The capability to modify the signals sent to the simulator, based on identification of stroke and amplitude of swing,
- Electrical and logical compatibility with existing multi-bell interfaces and simulator software,
- The ability to provide a reasonable simulation of the principal features of raising and lowering, in particular the suppression of the backstroke strike.
- The ability to adjust the timing of the strike of the simulated bell based on the amplitude of the swing.

The solution designed to meet these requirements consists of an “intelligent” sensor with two detectors positioned close together. These are arranged symmetrically around a single trigger when the bell is at BDC. This arrangement is illustrated in Figure 3, which shows a bell approaching BDC during the handstroke.

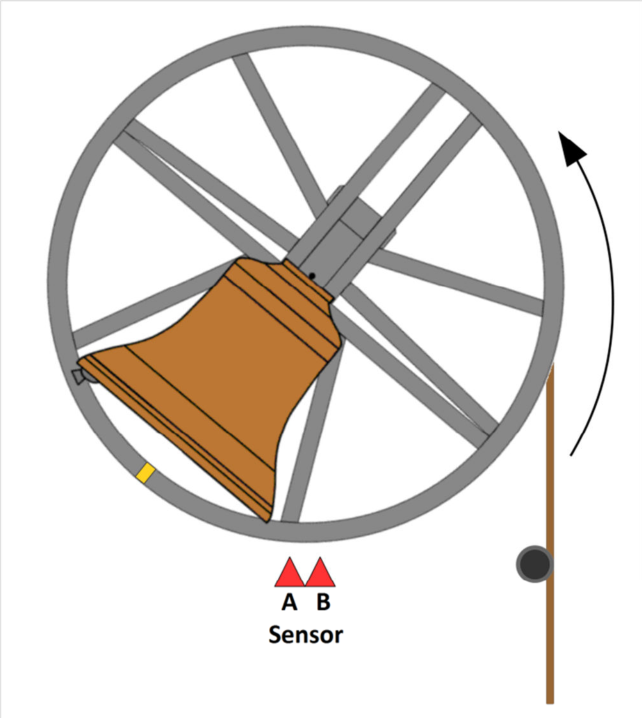

Figure 3

This arrangement allows the sensor to differentiate between handstroke and backstroke, by means of the sequence in which the two detectors are triggered. In the diagram, detector A will be triggered shortly before detector B during handstroke, and vice-versa during backstroke.

Detecting and tracking the amplitude of the swing of the bell directly in real time would be mechanically and computationally complex, so the sensor relies on the fact that the peak angular speed of the bell as it passes through BDC decreases as the bell is lowered. This change in speed can be detected by measuring the delta time interval between the triggering of the two detectors, and used as a proxy for the amplitude of the swing.

Figure 4 illustrates the increase in the delta time as a 10cwt bell is rung down, twice with minimal consistent checking, and once with more vigorous checking towards the end of the lower as would typically happen when ringing down in peal.

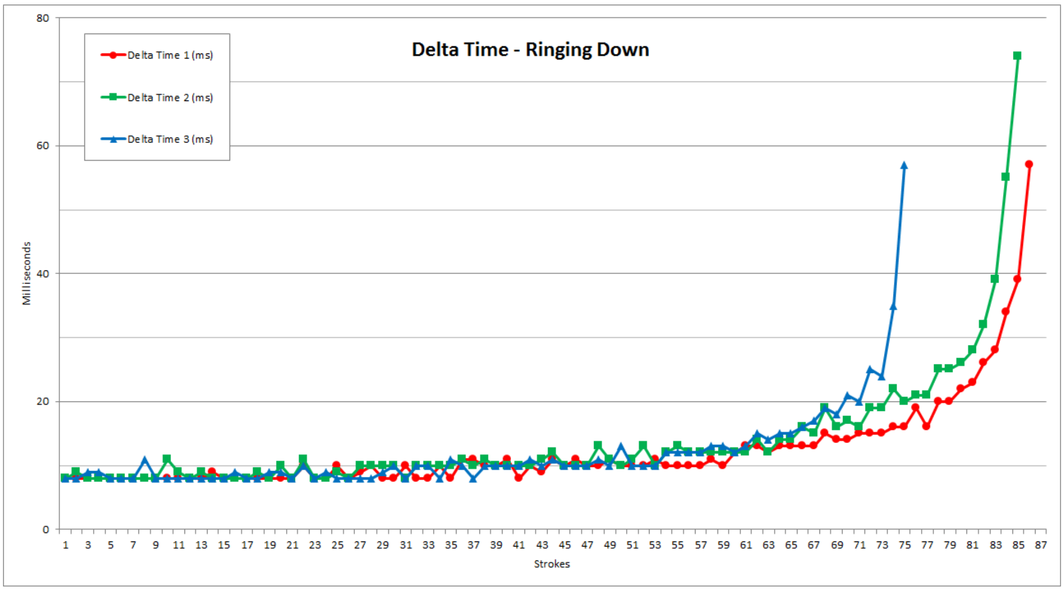

Figure 4

The converse is observed when a bell is raised. Figure 5 shows the decrease in delta time during three raises of the same bell.

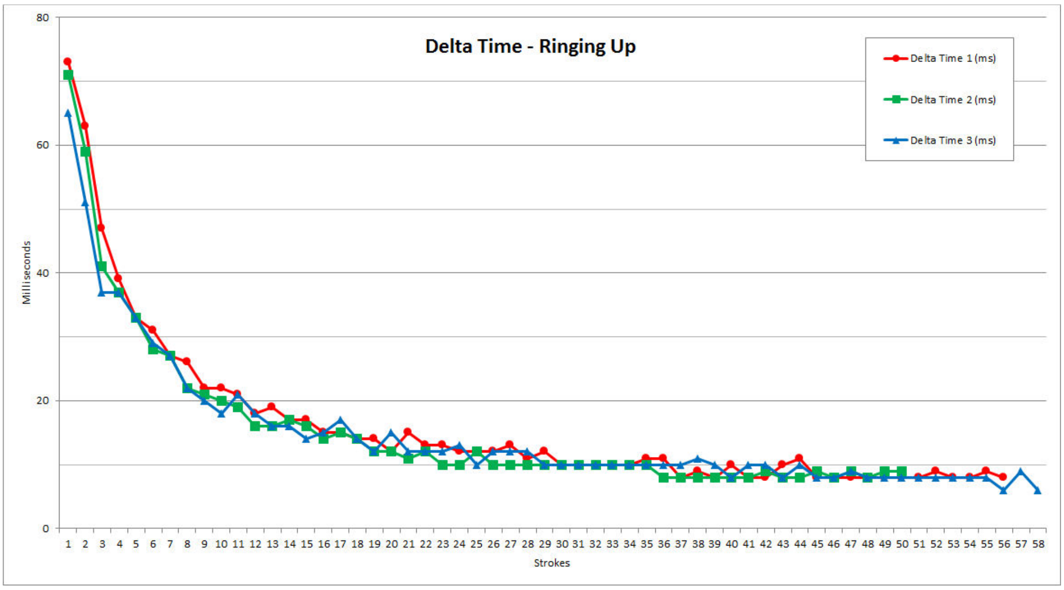

Figure 5

In practice, the absolute speed of the bell does not matter, and it is sufficient to detect the relative change in peak speed, compared to the maximum observed during full circle ringing. The fact that the detectors do not trigger exactly at BDC is also unimportant, provided that the arrangement of detectors is symmetrical around the trigger at BDC.

A prototype sensor has been developed to test this approach. The sensor uses the Honeywell magneto-resistive sensors pioneered by Aiden Hedley, triggered by a rare earth magnet fixed to the bell wheel. The separation of the detectors is approximately 55mm, the maximum that could be achieved in the standard sensor enclosure adopted by the Project. A small microcontroller provides the necessary processing power, and the sensor has a conventional 3-wire power and signal connection to the multi-bell interface. Note that this sensor is not suitable for direct connection to a PC serial port.

The prototype sensor circuit board is shown in Figure 6. The magneto-resistive detectors are mounted on the reverse of the board.

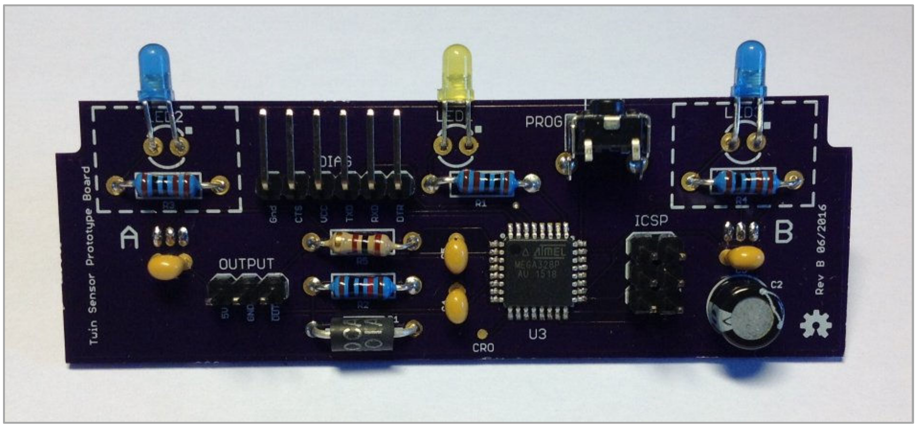

Figure 6

The microcontroller used is the AVR ATmega382P, allowing the use of the popular Arduino tool chain for software development, consistent with the rest of the Liverpool Ringing Simulator Project prototypes.

The firmware on the microcontroller identifies the direction of each swing from the sequence of triggers, and measures the time delta between the triggering of the two detectors. The current version of the firmware allows for the backstroke to be supressed while the time delta exceeds a defined threshold, thus simulating the most obvious feature of raising and lowering a real bell. A threshold of approximately two times the full circle minimum delta time seems to be a useful starting point.

The configuration of the sensor depends on two parameters specific to the installation on a particular bell; the direction of the handstroke relative to the sensor, and the maximum peak speed of the bell during full circle ringing. A single programming button allows these to be configured in situ once the sensor has been installed.

The sensor firmware applies its own delay timer (currently 100 milliseconds) to the outgoing signal sent to the simulator. The delay configured in the simulator or multi-bell interface is reduced by the same amount, resulting in an unchanged overall delay during full circle ringing. The firmware could therefore also adjust the BDC-to-strike interval based on the amplitude of the swing, although it does not yet do this pending collection of more data and development of a suitable algorithm.

As a development prototype, the sensor is deliberately highly configurable and allows the collection of statistics and debugging information via a diagnostic serial port. A simpler construction, avoiding the use of surface mount components, is envisaged in future once the behaviour of the sensor is better understood and the firmware further refined.

Figure 7 shows the prototype sensor installed (the trigger magnet had not yet been fitted to the wheel).

Figure 7

Initial results are promising. The simulated bell behaves much more convincingly during raising and lowering than when fitted with a conventional sensor, and this could be further improved by implementing some adjustment to the BDC-to-strike interval in the firmware. It would seem a worthwhile exercise to build a number of prototype sensors to assess the usefulness of this approach during raising and lowering in peal as part of a simulator silent practice.

Although useful, the solution is certainly not perfect: Although the bell will behave more or less correctly in response to a correct chiming action, it cannot require that the correct technique is used; it is impossible to ring the bell up wrong, but the knack of chiming is arguably something best taught with an open bell. The sensor certainly would not handle a Devon-style silent swinging start. Perhaps accelerometers might have a future role in addressing these shortcomings.

Further information on the prototype sensor can be found in the development branch of the Liverpool Ringing Simulator Project's GitHub repository (https://github.com/Simulators), where all code and artefacts are released under GPL or Creative Commons Open Source licences.

The author would like to thank John Norris for permission to re-use and adapt his bell diagrams, Frank King for interesting theoretical discussions, and the Liverpool Cathedral Guild of Change Ringers for their continuing patience.