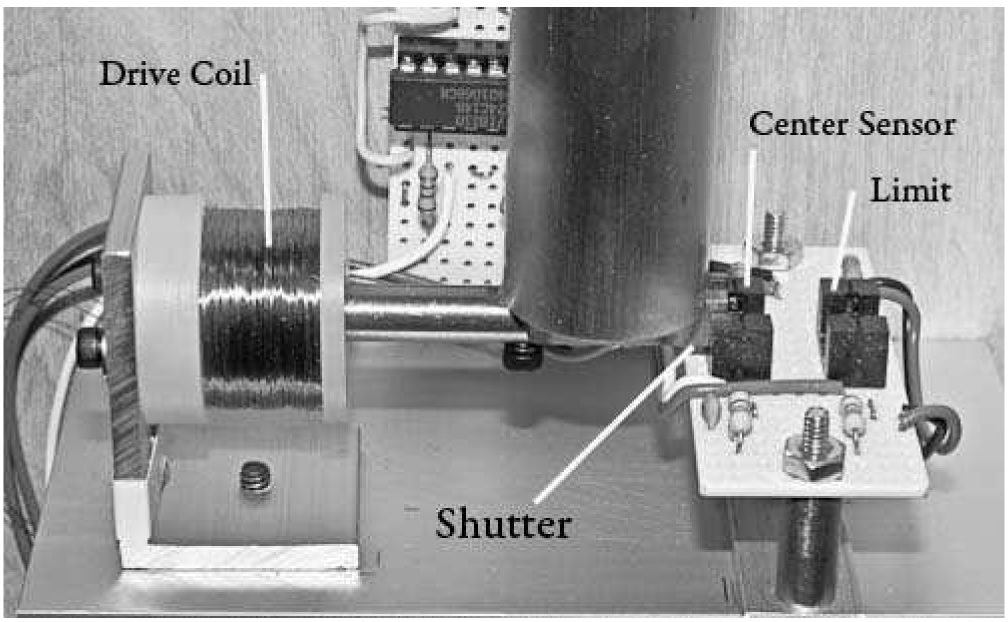
Results of Proportional Digital Drive with #8 Pendulum

by Robert L. Belleville

Pendulum #8, which has been described before in these pages, is driven by a brief impulse from a solenoid pulling a non-contacting steel core. Amplitude is controlled by two IR switches. A single chip microprocessor is used to orchestrate the drive. Since this processor can be very easily reprogrammed in place, I can change the method used on the fly without even stopping the pendulum. Over the last months I've been experimenting with the system.

Here I'll describe the system and show how the parts work. I'll give detailed information on the performance of the IR sensors I use. I'll show circular error and I'll show how circular error can be held essentially constant. Finally I'll show the performance of this clock in terms to temperature error and finally, temperature corrected rate.

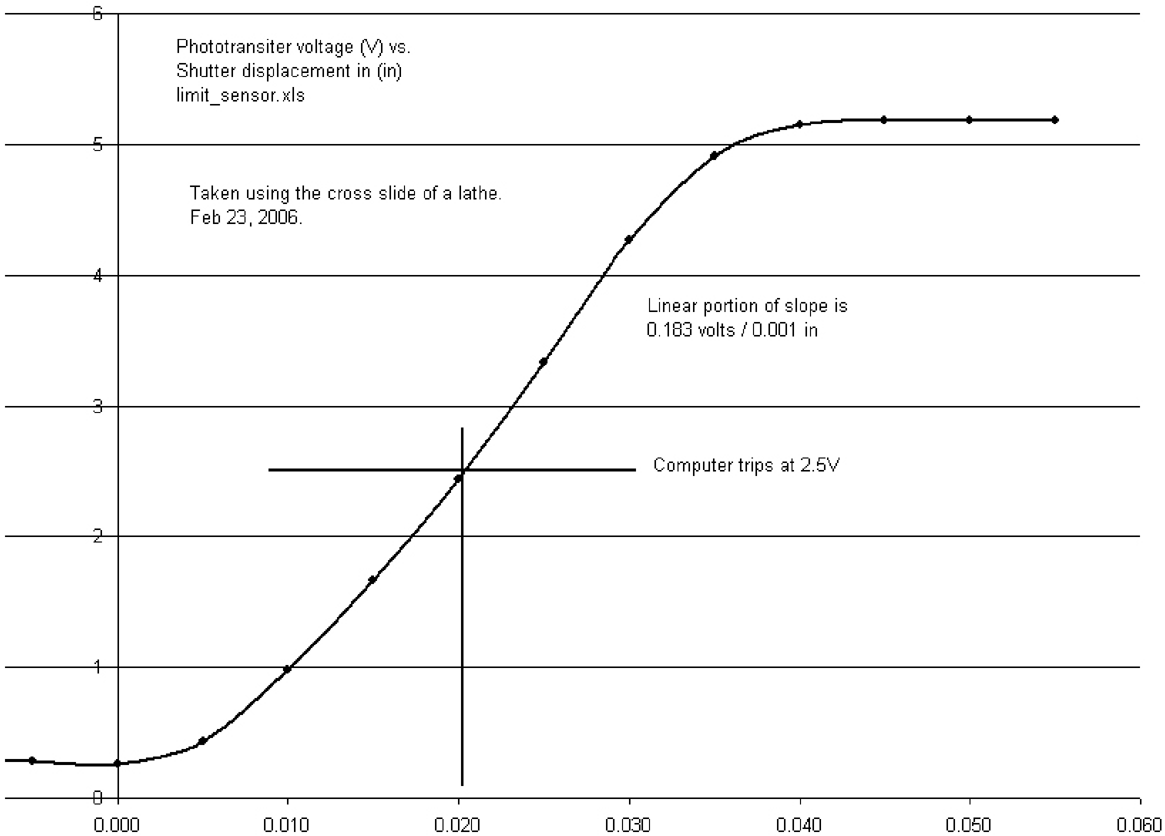


The electro-magnetic drive is simple. The microprocessor energizes the coil for a few milliseconds as the pendulum just passes the center sensor going to the left. Drive can range from 60 ms to about 2.0 ms depending on the experiment. Opposite the solenoid armature, seen clearly above, is the shutter to interrupt the IR sensors.

The sensors are nothing special. These are easily available for $1 or less. An IR LED, in the back in this photo, illuminates an IR sensitive phototransistor opposite. The gap is about 0.2 inch. When not blocked by the metal shutter, photons flood the base of the phototransistor and cause it to saturate at about 0.4 vdc. When completely blocked, the transistor turns off and is pulled up by a 10 Kohm resistor. It is what happens as the shutter swings through the center of the beam that is actually interesting.

More details: http://en.wikipedia.org/wiki/Phototransister

To create a voltage/position plot, I mounted the sensor in the jaws of my little machine lathe and fastened a metal shutter to the cross slide. (Since IR goes right through paper and plastic, metal is needed.) Starting with the shutter outside the sensor as zero, I made measurements of the voltage every 0.005 inches until the transistor was completely off. As the chart shows, there is a nearly linear voltage to displacement relationship for several thousands of an inch near the 'center' of the path.



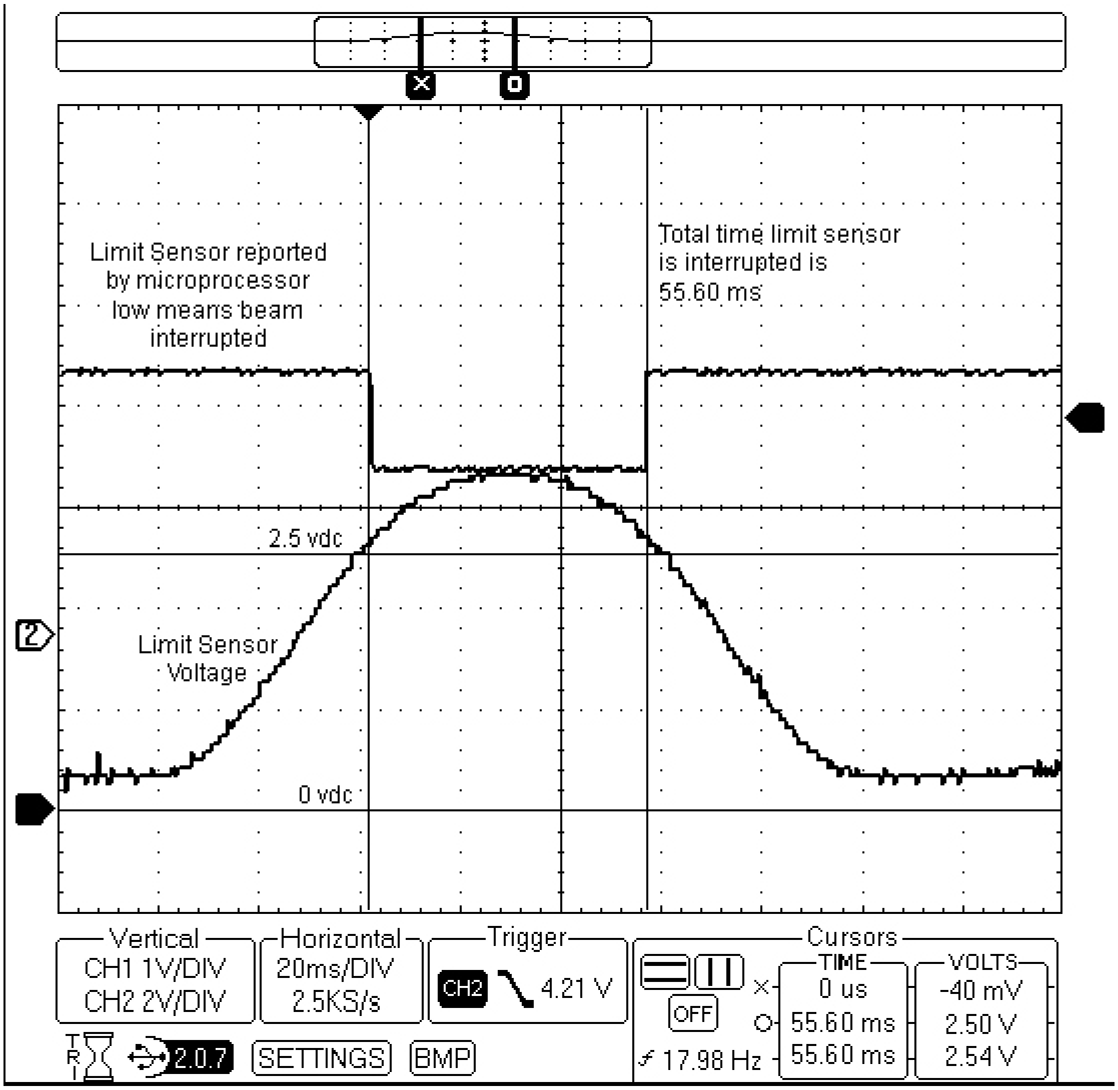
For about 0.030 inch the output of the phototransistor is about 0.183 volts/0.001 inch. Circuitry on the input pin of the microprocessor treats voltages greater than about 2.5 volts to be a '1' and those below as '0'. In this way I can measure the time that the pendulum passes a particular point to within a few thousands of an inch in a simple and inexpensive way. Uncertainty about when/where the transistor is on or off is eliminated.

Two sensors, placed exactly 1/2 inch apart, are used. With the pendulum as still as it gets, I adjust the center sensor to just switch. This means that the limit sensor voltage will rise to 2.5 volts starting 1/2 inch to the right of center. Flexure to the sensor distance is 854mm, the amplitude just at the limit sensor is 0.0148701 radians or 0.852degrees. Note also for our purposes here the period of the pendulum is 1.506 seconds. (It is just a 7/8 inch brass rod.)

A nifty attachment to any portable computer is a USB oscilloscope. I got one several years ago for about $125. At the slow speeds of pendulums it works great. The figure below show how the computer sees the shutter partially covering the limit sensor at the end of the swing.

(www.parallax.com, item code 28014, Parallax USB Oscilloscope, the price went up a little)

Notice how close to the 2.5 volt point the computer switches for both the leading and trailing edges. Also note that the beam was interrupted for 55.6ms. It is this time interval which will give us the exact amplitude of the swing in each cycle.



Each cycle of pendulum goes as follows:

1) Microprocessor (mpu) sits idle waiting for the center sensor to go high, that is just going right of center. Also this ends the timer period and starts the next. This mpu runs at about 1 million instructions per second so the pendulum looks very very slow to it.

2) Next the mpu waits for the limit sensor voltage to rise above 2.5 volts. This might not actually happen if the pendulum is not swinging widely enough, say after an earthquake. In this case the mpu gives strong drive pulses, about 60ms, to get it going again.

3) Once in the limit sensor, the mpu measures the time the limit sensor is blocked. This time is directly related to amplitude. In the mpu amplitude is counted and stored in an 8 bit number.

4) After this time has been measured, the mpu waits for the pendulum's center sensor to go off, that is just left of center.

5) Using the amplitude, a drive pulse is selected (or perhaps skipped if the amplitude is still large.) See below for more on this process.

6) This drive pulse is given via the coil.

7) A report of the cycle is created and transmitted to my desktop computer where it is logged into a file. Also recorded are environmental data for temperature, barometric pressure, and relative humidity. Pendulum period is measured and compared to the 1 second signal from a GPS. (Garmin GPS18-LVC) Next repeat from step 1.

In the past I have chosen a drive pulse length long enough to keep the amplitude above some value for several cycles, usually 10 to 20 seconds. In this model there is a drive pulse and a number of completely free, coasting cycles - like a Synchronome. Once the amplitude drops below a set number another drive pulse is given and the cycle of drive and coast continues.

Data in this paper shows a different method. Using amplitude as feedback, I drive each cycle inversely proportional to amplitude. I can turn on the drive coil in increments of 50 μsec. To decide how many drive increments to use, I subtract the amplitude count from a arbitrary number called capN. For this data capN is 173. So for an amplitude of 129 I give 173-129 = 40 times 50 gives 2000 μsec or 2 ms. As amplitude increases I give less drive power, as amplitude decreases I give more drive power. The system stabilizes at some amplitude related to capN. For this run that average number was 127.23 resulting in a average drive of just over 2.1 ms. The range, or variation in amplitude, was only 1.19 counts.

This count in the limit sensor is converted to milliseconds by dividing by 1.65. So amplitude can be computed by:

amplitude (a) (radians) = 0.0148701 / cosine(π * time in limit(ms)/1506(ms))

During this run average amplitude was 0.015065 radians or 0.863 degrees. Variation was $4e^{-6}$ radians or 0.00023 degrees.

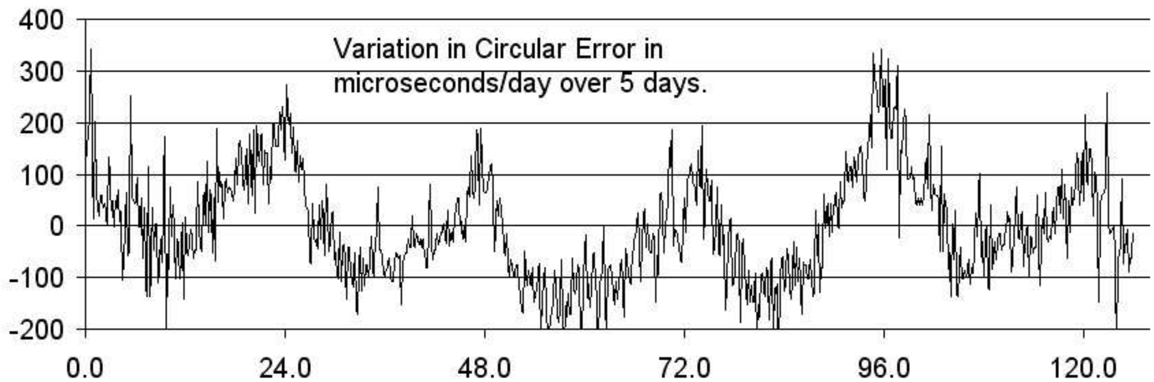
This is because we know that amplitude 1/2 inch is just 0 ms in limit sensor. (Follows method in HSN 2009-2 pages 35 to 36 by Peter R. Hastings. I separately checked this result and get the same equation.)

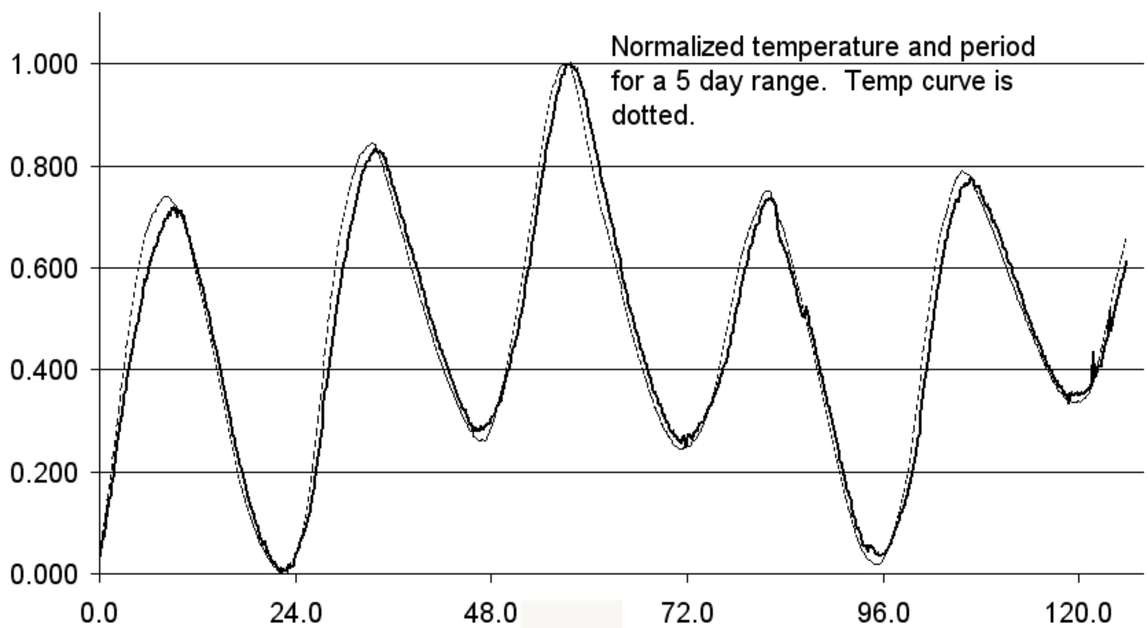
I'll use the usual first order approximation for circular error - $5400a^2$ seconds/day. There are about 57371 cycles per day with a nominal period of 1.506 seconds. So the per cycle circular error is 5400/57371 times a^2 ($0.0941250a^2$.) Here circular error is 1.23 seconds/day but the variation was only 600 microseconds/day. (Two seconds per year. But see below.)

As an aside, since the pendulum weighs 2.7175 kg and its center of mass is half the length (0.427 m) and since we now know the amplitude in radians it is easy to compute the potential energy at the end of each cycle as 2.7175 kg * 9.8 m/s^2 * 0.427 m * (1 - cos(a)).

Over 300,000 cycles of experimental data was taken starting on the 8th of June 2009. To make this manageable, I averaged all the data over 10 minutes - about 398 cycles.

Returning to circular error, it varied only 600 μsec/day for the entire run - essentially zero since the variation is cyclical with temperature as the plot shows. Average value is zero to at least 6 decimal places. (X axis is total hours.)

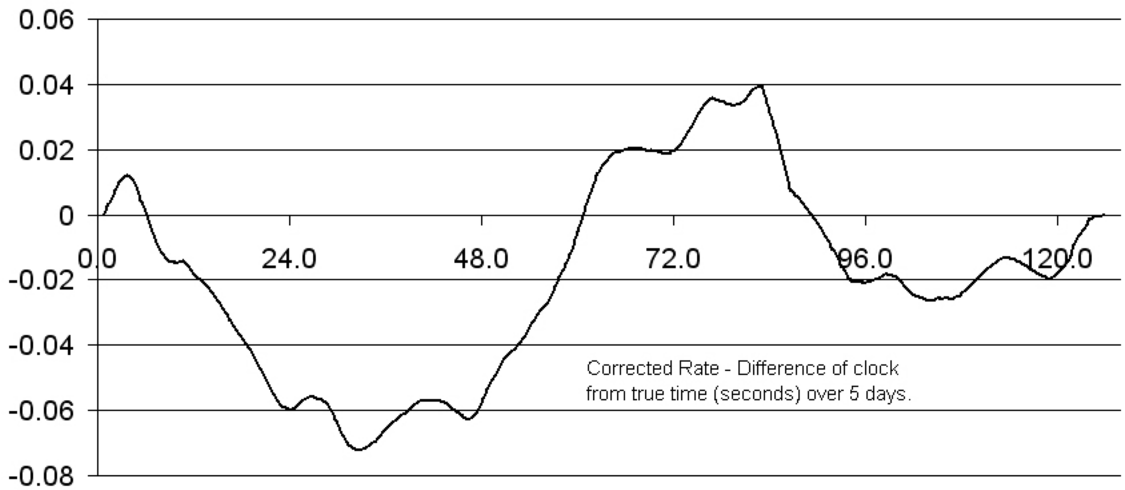




Buoyancy error for this run had a range of only 2.48 μsec while to total range of period variation was 78 μsec. None the less, I corrected for buoyancy error. What remains is almost all temperature error from length change in the brass bar. In fact this error is about 12 μsec/°C,

which is close the the theoretically value of generic brass. The plot above shows temperature and period together with a amplitude scale of 1.00 = 6.23 °C and 1.00 = 78 μsec.
Period lags temperature by about 40-50 minutes. Adjusting for thermal lag, the correlation is very close. (Specifically $R^2 = 0.994$.)

Finally, making a correction for the tempco, using a delay of 50 minutes, and converting period to rate by integrating we have the following plot.



This clock is compared to an atomic clock (via GPS.) It shows no more than 0.07 seconds slow or 0.04 second fast during the 5 day run.

Final notes:

IR sensors. As long at the pendulum stays in a single plane these sensor are cheap and accurate. If there is any front to back motion jitter of several microseconds is introduced into the timing.

Thermal correlation. A single tempco works well as long as the temperature is not changing slowly as it does near daily highs and lows. The bulk of the small rate variation is from this source. The variation tends to look like the first derivative of temperature however estimating this derivative from digital and noisy data so far eludes me. Larger variations include slight error in the phase relationship and other factors.

Flexure. This data was obtained only after the pendulum was operational for several years. My experience continues to be that beryllium-copper springs, even if run at 4000 psi, take many many months to settle. During that time the pendulum continues to speed up at an unpredictable rate making long term measurements impossible.

Next. I replaced the brass rod pendulum #8 with one of Invar 36 rod and a lead bob. Tempco of #9 should be about 0.7 microseconds per degree C. Perhaps in a year or so I can report a rate curve 10 times better than the one above. For now the system has to just settle. As this was written it is speeding up about 40 nanoseconds every 10 minutes.

Los Altos, California July 2009