Satellite-to-ground coincidence matching under Doppler effects

October 15, 2019

1 Introduction

Entanglement-based quantum key distribution (QKD) relies on a steady source of photon pairs. The first step in establishing a key in such a scheme is the assignment of photodetection events to entangled photon pairs. Ho et al. (2009) [1] introduced an algorithm for coincidence matching under a constant time offset (ΔT) and a constant frequency difference (Δu). However, for satellite-based photon pair sources, an element of Doppler shift causes the relative clock frequency between the satellite and groundstation to vary in time ($\Delta T(t), \Delta u(t)$). The complexity from Doppler shift rates at different angles of elevation can cause this cross correlation to be spread out over thousands of time bins. This note introduces two methods for correcting Doppler effects in satellite-to-ground coincidence matching.

1.1 Time stamping a satellite SPDC source

We consider a setup where a spontaneous parametric down conversion (SPDC) source on a satellite is time-stamped by two individual time-stamp cards with a relative clock drift of 165μs/s (Fig 1).

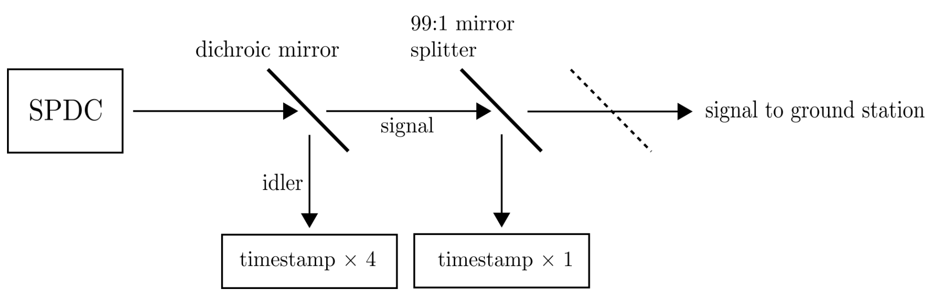
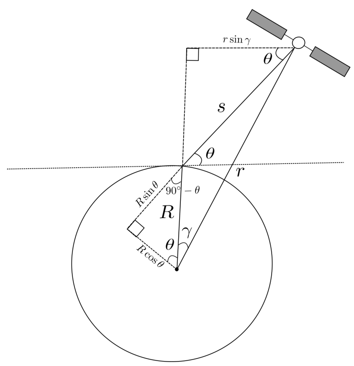


Figure 1: Idlers and 1% of signal photons are time-stamped on the satellite. The rest of the signal photons are transmitted to ground.

1.2 Propagation delay

We introduce a propagation delay for each time-stamp based on pair generation time. When the elevation angle is at its maximum 90°, we define that t is 0 seconds. Assuming that the satellite is in a circular orbit at altitude $h = 500$km, a simplified expression for the propagation delay can be given by:



$$V = \sqrt{\frac{R^2}{r} \cdot g}, \tag{1}$$

$$\gamma = \frac{V \cdot t}{r}, \tag{2}$$

$$\theta = \arccos\left(\frac{r \sin\gamma}{s}\right), \tag{3}$$

$$s = \sqrt{R^2 + r^2 - 2Rr\cos(\gamma)}, \tag{4}$$

$$\tag{5}$$

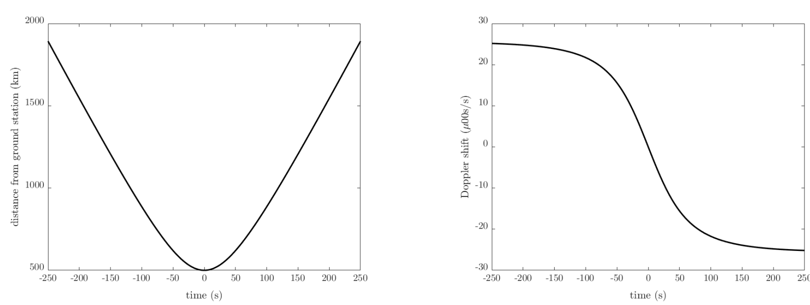
Figure 2: A simplified schematic of the satellite's orbit.

where R is the Earth's radius, $r = R + h$, g is gravitational acceleration, and c is the speed of light. Equation (1) is the velocity of the satellite; (2) is its orbital phase angle γ; (3) is its angle of elevation θ; and (4) is its distance from the ground station s.

Using this model, we get the delay on each timestamp due to Doppler shift (Fig 3),

$$\Delta t = \frac{s(t)}{c}, \tag{6}$$

where distance is calculated for each pair generation time in orbit.



(a) (b)

Figure 3: (a) Delay due to Doppler shift and (b) rate of change of delay.

1.3 Doppler shift

Since the stream of time stamps $\{t_i\}$ and $\{t_j\}$ on each side has no intrinsic time structure, it is difficult to distinguish a signal event from a background event in the frequency-shifted $\{t'_j\}$ (Fig 4(c)). The Doppler shift is of the order of 0.2 nanoseconds per nanosecond and causes the cross correlation to spread out thousands of time bins, making it difficult to distinguish a significant peak.

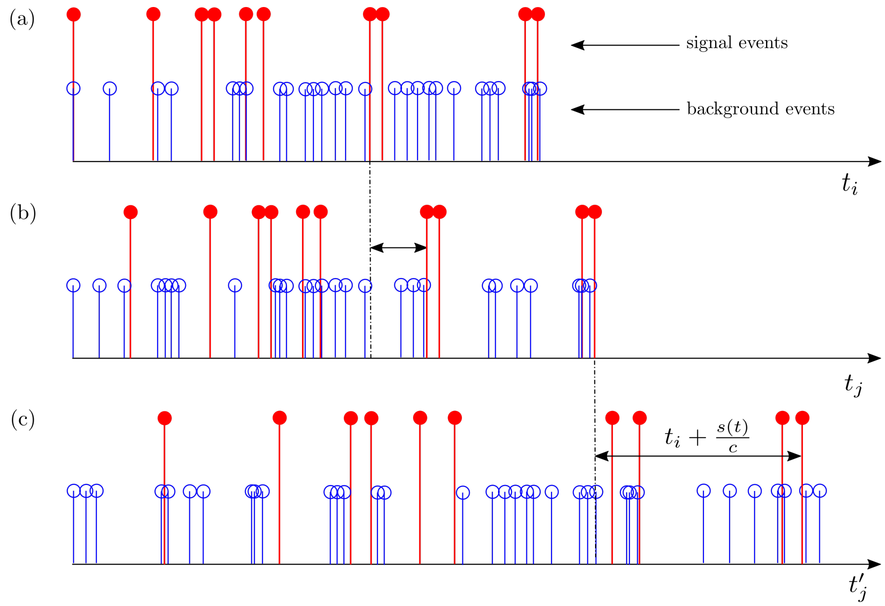


Figure 4: Effect of Doppler shift and Doppler shift rate on photoevent sets. Trace (a) represents the event set $\{t_i\}$ on side A, trace (b) an event set $\{t_j\}$ on side B with a time offset, and trace (c) with an additional Doppler shift $\frac{s(t)}{c}$. Image adapted from [1].

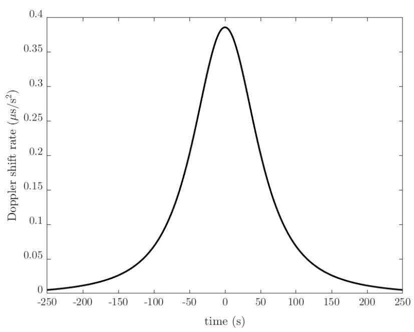


Figure 5: The change in Doppler shift.

To correct for this, we introduce a second order term to our time shift equation:

$$\Delta t = \frac{s(t)}{c} + \delta \cdot t, \tag{7}$$

where the Doppler shift δ is defined by (cf. eq. **??**)

$$\delta = \frac{V}{c} \cos\theta. \tag{8}$$

2 Correction using timing beacon on satellite

The timing beacon correction method installs a beacon with regular pulses on the satellite. We time-stamp the beacon signal both on the satellite and at the ground station (Fig 6). By introducing structure, we can warp the time-stamps until the beacon pulses match up, and then apply the same transformation to the idler and signal photoevents.

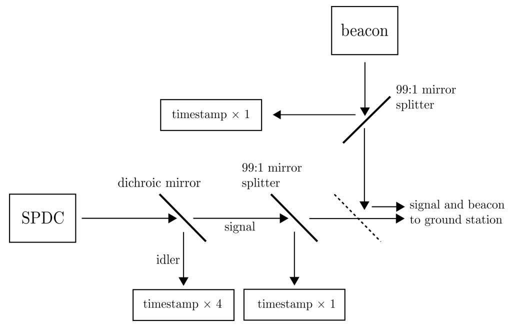


Figure 6: The beacon setup.

The higher order correction calculates the clock drift for each point and uses a second order polynomial fit to determine the clock drift at each point. For the case of 90 degrees elevation, this is shown below:

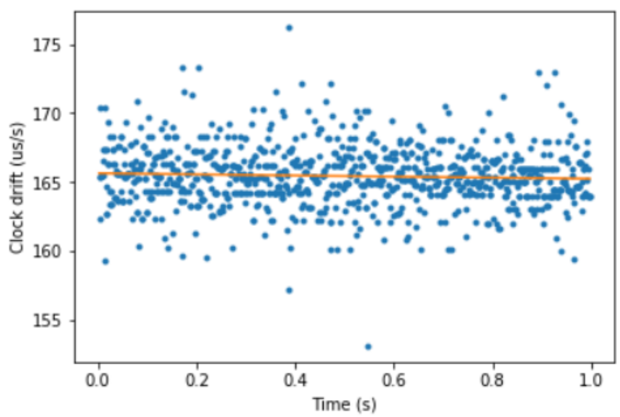


Figure 7: Estimation of the second-order correction at each point.

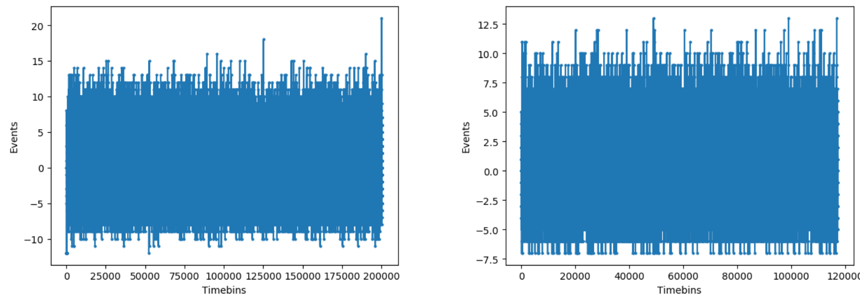
We find that even a high jitter beacon with poor detection efficiency is sufficient to correct for all clock drifts including the change of the Doppler shift for high elevation angles.

However, the beacon introduces additional requirement into the satellite's payload. By using information from the TLE and methods like gradient ascent, we can recover the shift without the additional structure of beacon pulses.

3 Correction with entanglement

3.1 Experimental setup

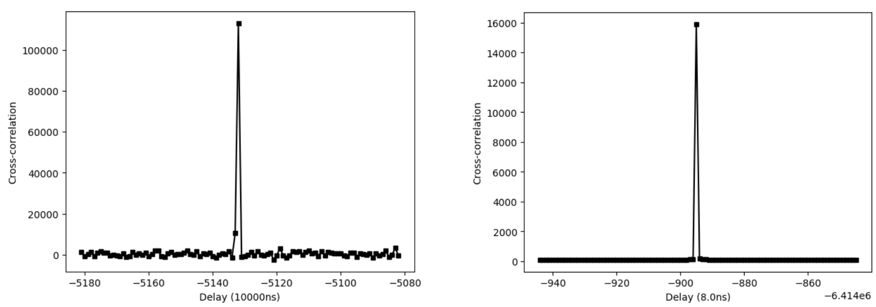
We begin with timestamps from a tabletop experiment, where Alice transmits photons to Bob. There is no Doppler shift here, but Bob's signal picks up some noise and losses.



(a) (b)

Figure 8: Timebins of 10000ns on (a) Alice tabletop atomic clock, (b) Bob tabletop atomic clock

To efficiently perform cross-correlation on two-million point dataset, we first use the Fast Fourier Transform with coarse timebins of 10000ns (Fig 9(a)). Having identified a coarse peak, we then zoom in on a window around it and repeat the process with finer timebins (Fig 9(b)). Since the resolution of the timestamps is 2.0ns, we decided on a timebin size of 8.0ns would be precise enough to be useful, yet still robust enough against statistical noise.



(a) (b)

Figure 9: Cross-correlation of Alice's and Bob's timestamps, using (a) coarse timebins and FFT, followed by (b) fine timebins with `pycorrelate`.

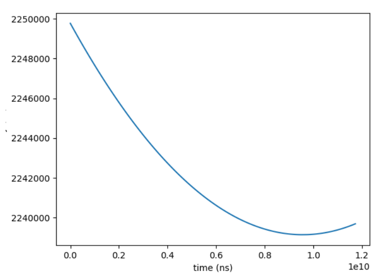
3.2 Artificial Doppler shift

We introduce an artificial Doppler shift on Bob's timestamps. This is done by setting the timestamp to be in a time range for a satellite pass, and calculating the real Doppler shift for each timestamp. We used the `GALASSIA` satellite (Fig. 10) and the `pyephem` package for this.

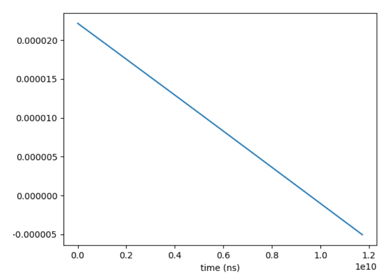
GALASSIA
1 41170U 15077E 18291.47886069 .00002035 00000-0 73075-4 0 9998
2 41170 14.9881 191.7979 0013088 334.9563 25.0122 15.13422614157130
Latitude, longitude, elevation: 1.2954752^o, 103.7800079^o, 100^o
Time and date: 2018/10/19 04:57:11

Figure 10: GALASSIA TLE and saved pass

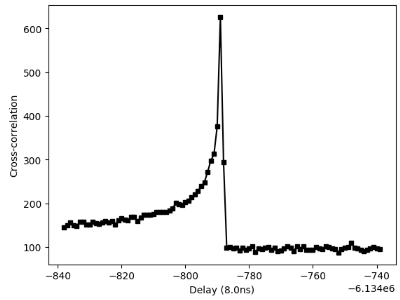
As expected, the Doppler shift introduces an approximately quadratic delay on the timestamps (Fig. 11(a)), and it is happening at the steepest rate (11(b)). After being shifted, the cross-correlation peak spreads out over several timebins (11(c)) and has to be corrected to recover a sharp peak (11(d)).



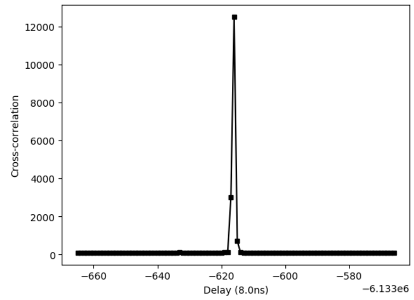
(a)



(b)



(c)



(d)

Figure 11: (a) Introduced Doppler shift on each timestamp for the chosen satellite, and (b) rate of change of Doppler shift. (cf. Fig. 3). Cross-correlation of signals with (c) propagation delay and (d) after correction and gradient ascent.

3.3 Gradient ascent on Doppler shift

To recover Fig 11(d) from Fig 11(c), we use gradient ascent on a guessed Doppler shift function. Our gradient ascent method is as follows:

1. Using TLE data and the time range of the satellite pass, guess a Doppler shift as a polynomial function (e.g. quadratic):

$$f(x) = ax^2 + bx + c. \tag{9}$$

   Note that we cannot now replicate the exact artificial shift we had put in earlier, since we should not have access to Bob's unshifted timestamps. The point is therefore to recreate the shift as closely as possible using a polynomial fit.

2. Apply $f(x)$ to Bob's (shifted) timestamps, and perform cross-correlation with Alice's timestamps.

3. Define a gain function: here, we want to maximise the total distance between the cross-correlation peak and its adjacent points, i.e. get as sharp a peak as possible:

$$g = 2 * \texttt{max(xcorr)} - \texttt{xcorr[maxIdx - 1]} - \texttt{xcorr[maxIdx + 1]}. \tag{10}$$

   This is always positive by definition. We can also normalise the gain function $g' = g/\texttt{max(xcorr)}$. We expect a perfect cross-correlation to have $g' = 2.0$, with a peak at one point and zero at all others.

4. Use gradient ascent to update the polynomial coefficients, and maximise the sharpness of the cross-correlation peak:

$$\begin{aligned} a_i &= a_{i-1} + \alpha * \Delta_{a_{i-1}} \\ b_i &= b_{i-1} + \beta * \Delta_{b_{i-1}}, \end{aligned}$$

   where α, β are learning rates, and $\Delta_{a_i} = \frac{g_i - g_{i-1}}{a_i - a_{i-1}}, \Delta_{b_i} = \frac{g_i - g_{i-1}}{b_i - b_{i-1}}$ are the gradients which define the direction of the next step. This method treats each coefficient as independent of the others.

   For the initial iteration, where there is no previous set of parameters, we use a manual offset to kickstart the ascent. Note also that we ignore the constant offset, since the point of the cross-correlation is to capture that very delay.

5. Repeat until the gradient ascent no longer improves the result ($g_i < g_{i-1}$). In cases where the learning rate is too low, and our result improves at an ever-slowing rate, we define an ϵ to terminate the ascent below a certain rate ($\texttt{STOP if } g_i - g_{i-1} < \epsilon$).

4 Results and discussion

`insert graph of gain function over iterations for different degrees and learning rates`

4.1 Results

- g
- time taken to run

4.2 Validation

- validate using timestamps from real passes
- show recovery of key, QBER etc. (or include reference to paper / code which can do it)

4.3 Limitations and future directions

- main bottleneck is cross-correlation: sidestep with caching / windowing?
- adaptive learning rates
- gradient descent on other fits besides polynomial (e.g. spline?)

References

[1] Caleb Ho, Antia Lamas-Linares, and Christian Kurtsiefer. Clock synchronization by remote detection of correlated photon pairs. *New Journal of Physics*, 11(4):045011, 2009.