UNIVERSITY OF CALIFORNIA, SAN DIEGO

Normal Point Generation and First Photon Bias Correction in APOLLO Lunar Laser Ranging

A dissertation submitted in partial satisfaction of the requirements for the degree

Doctor of Philosophy

in

Physics

by

Eric Leonard Michelsen

Committee in charge:

Professor Thomas W. Murphy, Chair
Professor Duncan Agnew
Professor Kim Griest
Professor David Miller
Professor Hans Paar
Professor Kevin Quest

2010

The Dissertation of Eric Leonard Michelsen is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2010

DEDICATION

This is work is dedicated to my wife, Laura, my daughter, Sarah, and my son, Ethan, for their enduring patience during its creation. Because of them, everything in my life is made better.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

This work is only possible because of Tom Murphy, the creator of APOLLO, and my advisor. During my time at UCSD, I was also helped immeasurably by many fellow students, and especially by my friends Kevin Duggento, Sarah Garcia, Jason Leonard, and Yaniv Rosen.

VITA

1981 BSEE, Florida Institute of Technology

2005 Master of Science, Physics, University of California, San Diego

2010 Doctor of Philosophy, Physics, University of California, San Diego

ABSTRACT OF THE DISSERTATION

Normal Point Generation and First Photon Bias Correction in APOLLO Lunar Laser Ranging

by

Eric Leonard Michelsen

Doctor of Philosophy in Physics

University of California, San Diego, 2010

Professor Thomas W. Murphy, Chair

The APOLLO Lunar Laser Ranging (LLR) system studies gravity by tracing out the orbit of the moon to ~1 mm, over many years. LLR in general provides extensive tests of many aspects of gravity, including deviations from General Relativity (GR), and time rate-of-change of the gravitational constant, *G*. APOLLO's precision is approximately 10× better than previous LLR measurements, enabling about an order of magnitude improvement in tests of gravity over the coming years. APOLLO requires complex data reduction methods to extract the distance so precisely. There are currently three choices for determining the round-trip-time to the moon from the data: the correlation method, the Augmented Calculation method, and the PDF-fit method. The results here suggest the PDF-fit method as preferable, for minimum random uncertainty over the full operating range of conditions, and stable systematic error below ~1 mm. As a second topic, the APOLLO system includes a systematic error called "First Photon Bias," which causes time measurements to be skewed early. An algorithm is presented and simulated, showing that it is inherently capable of achieving < 1 mm systematic error under normal operating conditions. However, the final algorithm requires a correction table calibrated from a more accurate model of shot-to-shot intensity fluctuations. Such a table could be the subject of future investigations.

1 Introduction

1.1 What Is APOLLO?

APOLLO is the Apache Point Observatory Lunar Laser Ranging Operation. By measuring the distance to the moon over time, we trace its orbit around the earth to ~1 mm, as a test of gravity. Lunar Laser Ranging (LLR) in general provides extensive tests of many aspects of gravity, including deviations from General Relativity (GR), and time rate-of-change of the gravitational constant, *G* [Nor]. Since August 1969, right after the first lunar retroreflector was put down by Apollo 11 astronauts, earth bound ranging stations have taken laser ranging measurements. Figure 1 shows the locations of retroreflectors on the moon. (The Luna 17 rover, known as Lunokhod 1, was lost for 39 years. It was found in April, 2010, from photographs from the Lunar Reconnaissance Orbiter, and ranges from APOLLO. It is now located to better than 0.5 m, and after 4 decades of mystery, APOLLO ranges to it regularly.) Other stations, currently operating or in the past, include the Observatoire de Cote d'Azur in France, the University of Texas, McDonald Observatory, and stations in Hawaii and Italy.

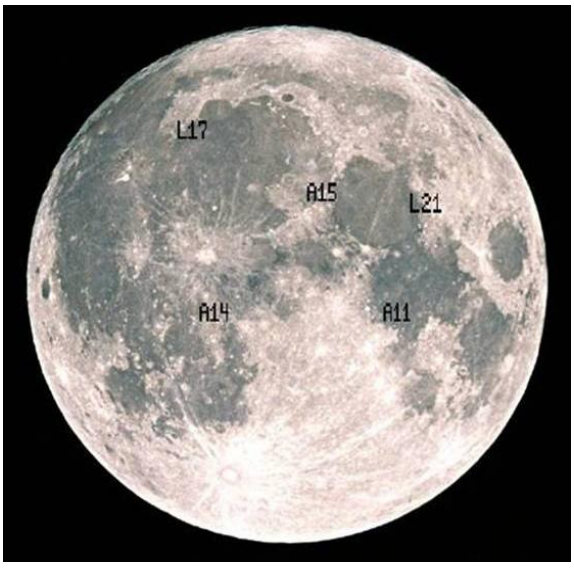


Figure 1: Retroreflector locations on the moon.

Note that “APOLLO” is the laser ranging operation, but “Apollo” is the series of lunar

missions started by NASA in the late 1960s.

Combining lunar ranges with other solar system observations, scientists at Jet Propulsion Laboratory (JPL) and others determine coefficients in the General Relativity (GR) equations of motion, by fitting ~150 parameters to a model of the entire solar system.

Prior systems have had an accuracy of ~200 down to 10 mm (exact figures are hard to determine, and change over time). APOLLO provides ~10× improvement by frequently achieving ~1mm precision over a night's session [Bat]. Over the coming years, this will improve experimental bounds on many gravitational parameters by an order of magnitude [Mur].

Why another lunar laser ranging (LLR) operation? Although every gravity experiment ever performed is consistent with General Relativity, our understanding of gravity is incomplete [Car p376]. It is well known that quantum mechanics and GR are incompatible, as they stand. Even more directly, there are unexplained phenomena which might reflect new gravitational physics, such as those related to the unexpected accelerating expansion of the universe and the associated dark energy, and the fact that the rotation of galaxies indicates that their gravity is stronger than the visible matter supports (dark matter). There are many proposals for gravity beyond GR, including things such as scalar fields and extra dimensions. All such new theories require experimental evidence. It is possible (albeit unlikely) that APOLLO could measure violations of General Relativity.

The most likely metric theories of gravity fall into the Parameterized Post Newtonian (PPN) framework, which defines 10 adjustable parameters that quantify various effects on the metric. While most of the details are not important here, for completeness the 10 parameters are called γ, β, ξ, α_1, α_2, α_3, ζ_1, ζ_2, ζ_3, ζ_4 [Wil ch 4]. Of these, two important ones are γ and β, because these two can vary while still conserving energy, momentum, and angular momentum, and having no preferred-location effects. Briefly, γ measures the factor by which space-space curvature is different from space-time curvature. Space-time curvature accounts for the gross aspects of gravity, while space-space curvature introduces corrections visible at relativistic speeds. It is possible that these two aspects of gravity could be distinct. In GR, there is no difference, and $\gamma = 1$. The PPN parameter β is a measure how gravity

interacts with itself: does $E = mc^2$ apply when the energy is gravitational potential energy? Another way to say this is that β measures the non-linearity of gravity: is gravitational potential energy itself a source of gravity? GR is about the simplest of metric theories, and has $\gamma = \beta = 1$, with all other PPN parameters zero. The JPL solar system model usually includes fits only for γ and β, but it can also test others. LLR constrains α_1 well, and also constrains α_2.

LLR can also put constraints on other gravitational parameters besides those in the PPN framework. For example, perhaps the gravitational "constant" isn't really constant (like Hubble's "constant" turned out to be a function of time in an expanding universe). "G-dot" is a measure of the rate of change of the gravitational constant with time. LLR currently constrains G-dot/G to $< \sim 10^{-12}$ /yr. Pre-APOLLO LLR has constrained the equivalence principle for the earth and moon to about 1 part in 10^{13}, the value of β to about 10^{-4}, and the time variation of G, as $G\text{-}dot/G$, to less than 10^{-12} [JW2]. APOLLO hopes to improve these by a factor of 10 (or actually measure a non-zero rate of change).

As an example of how LLR can detect properties of gravity, consider the earth-moon system in orbit around the sun. While we usually consider the moon to orbit the earth, and the earth-moon system to orbit the sun. However, the force of the sun on the moon is greater than that of the earth on the moon, so both earth and moon are in orbit around the sun. Potentially, if the equivalence principle is violated, the effect of the sun on the earth might be different than the effect of the sun on the moon. The earth and moon are made of different materials: mostly iron for the earth, vs. silica (or rock) for the moon. The have different fractions of their mass in gravitational binding energy: 4.6×10^{-10} for earth vs. 0.2×10^{-10} for the moon. Does $E = mc^2$ apply to gravitational binding energy? In other words, if we could drop the earth and moon together onto the sun, would they fall at the same rate? Since the earth and moon are both in orbit around the sun, this experiment is being performed every day.

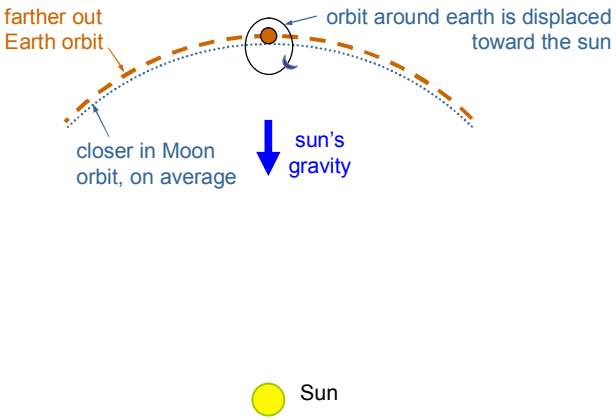


Figure 2: Earth-moon system as a test of the equivalence principle.

Suppose that the earth falls more slowly toward the sun than the moon (corresponding to a lower ratio of gravitational mass to inertial mass for the earth compared to the moon). Figure 2 shows the behavior of the orbits: the average earth orbit is farther out than the average lunar orbit around the sun. From the earth, we would measure this difference as a displacement toward the sun of the lunar orbit around the earth.

Why use the moon to measure gravity? The moon is a massive, and therefore stable, example of gravity. It is not influenced much by the irregular solar wind (though it is influenced by the radiation pressure of the sun). In contrast, satellites are buffeted by the solar wind and atmosphere (at low orbits) at a level that obscures any subtleties in gravity. Also, their much tighter gavitational binding to earth precludes any equivalence principle test. While there are significant satellite laser ranging (SLR) experiments in progress, their science goals are different, and they cannot serve as accurate measures of gravity. There are also sensitive laboratory measures of gravity [Kap], but they are constrained to small distances. There is a need to measure gravity at larger lengths, as well, which is well-served by LLR.

The primary goal of APOLLO is ranging to the moon, about 100 times per year, with data frequently at the ~1 mm level for a night. One mm one-way equals 6.7 ps of round-trip time (RTT). We will refer often to these two numbers in deciding what uncertainties are acceptable or negligible.

The goal of this dissertation's research is to bring the systematic errors in our measurement down to near the level of precision, so that our results are roughly as accurate as they are precise.

The following terms will be used frequently in this dissertation. See the Glossary for more.

normal point	a single hypothetical launch time, its associated round trip time (RTT), its uncertainty, and associated ancillary data. A normal point is computed by aggregating the RTTs measured on dozens to thousands of detected photons in a run.
run	a continuous sequence of laser shots and measurements. A run is typically 5000 to 20,000 shots, spanning ~4 to 18 minutes. To date, each usable run produces a single normal point.
session	a sequence of runs within a short time span, typically ~1-2 hours, allowing ~3-12 runs. We usually are allocated only one session in a night.
precision	random error
random error	errors that "average down" to a more accurate value with repeated measurements.
accuracy	systematic error
systematic error	errors that do not "average down" with more measurements. In APOLLO's case, there are significant systematic errors that vary from session to session, e.g. those due to background photons, lunar libration, return signal intensity, and others.

1.2 Overview of the APOLLO System

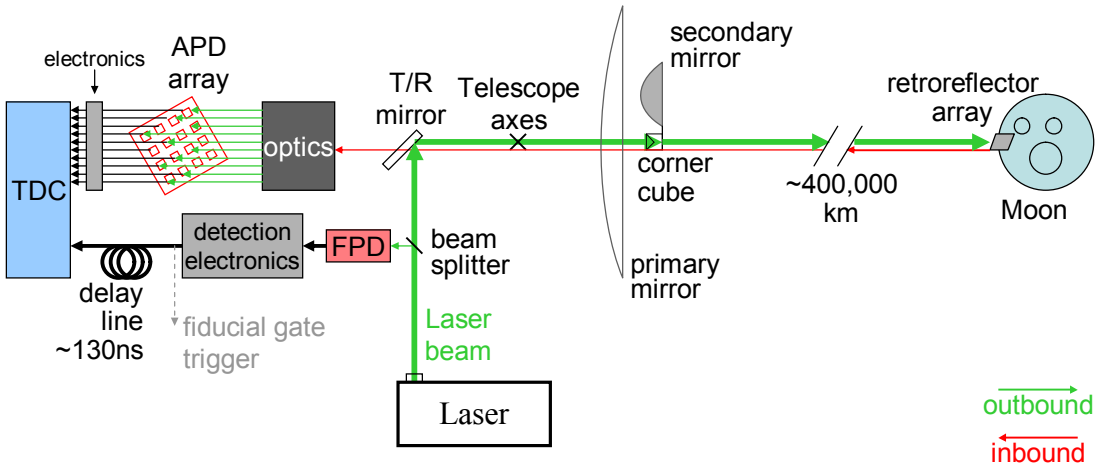


Figure 3: Schematic overview of APOLLO: Lunar Laser Ranging

To reach our 1 mm goal, we need to aggregate the RTT measurements of many photons. A single photon RTT may have an uncertainty anywhere from ~170 ps at low lunar libration, to ~400 ps at high libration. These require uncertainty reduction factors from "averaging" of ~25 to 60, respectively, corresponding to ~630 to 3600 photons. Nightly photon yields for APOLLO are regularly above this level, so we are generally meeting our precision goal.

Here is a simplified overview of the operation of the APOLLO system (see Figure 3) [Mur]. All uncertainties and variations are given as 1 σ, unless otherwise stated. A measurement starts with a pulse of green light (532 nm) from the laser. The pulse is ~200 ps, or about 60 mm long. On the way out, a small fraction of the beam is diverted by a beam splitter to a Fast Photo-Diode (FPD). The FPD provides the lowest-jitter measurement of the pulse launch time (~55 ps), and is the timing reference point for all other measurements.

The laser pulse is synchronized to a rotating optic (Transmit/Receive or T/R mirror) which has a mirrored section and a clear section. The laser pulse starts when the mirror is in the beam path, and the mirror diverts the pulse out the telescope. As the pulse goes past the secondary mirror (into the sky), a small corner cube on the edge of the secondary mirror reflects a small fraction of the outbound beam back into the measurement system. This reflected pulse is the **fiducial** signal, which

allows a (nearly) differential time measurement between the outbound beam and the return photons. For the fiducial pulse, the mirrored section of the T/R mirror is still in place, and attenuates the reflected pulse by ~10^6. Other attenuation reduces the intensity to the single-photon level, much like the lunar return signal will be. At this level, many fiducial measurements (FIDs) detect no photons at all, however the FPD timing anchor still provides a reference point for pulse launch time, and a measurement can still be made on the lunar return signal. We use the aggregate of all the fiducials in a run to calibrate the FPD-FID time, which ultimately results in a differential measurement of lunar-return (LUN) to FID time.

For technical reasons, the FIDs alternate between two types on each shot: the first type of FID is focused on a single spot of the APD. This is markedly different from the LUNs, which illuminate the APD uniformly across their active area. The focused FIDs are used for calibrations and comparisons to better understand our system. The second type of FID is used for our measurements: it is diffused across the face of the APD, much like the LUNs are, and is therefore more directly comparable to LUNs in detection time. There is typically a 10-20 ps difference in average detection times of the two types of FIDs. The APOLLO system does not record which type of FID is which in the data files. The two types are identified and separated during data reduction, and only the uniform FIDs are used for the RTT measurements.

The outbound pulse propagates to the moon, where it hits a retroreflector, which reflects it along the incident path, returning a fraction of the pulse back to the telescope. At the telescope, the T/R optic has had time to rotate to the clear section, so the weak return signal enters the receive system with no attenuation. The returned light is imaged onto an Avalanche Photo-Diode array of 16 detectors in a 4 × 4 pattern. Each diode can trigger independently on a single detected photon, so APOLLO could ideally detect from 0 to 16 photons per pulse. In practice, only ~11 detectors work, and we typically detect one or fewer photons per pulse; the best was 4.5 detections/shot for a brief (15 s) interval.

The detection electronics measures the arrival time of a photon at the APD with 25 ps

resolution, using an off-the-shelf Time-To-Digital converter (TDC). The TDC measures time as a 12-bit integer, between 0 and 4095, in units of 25 ps. This gives the TDC a maximum "window" of ~100 ns in which it can measure time. Due to this and other timing restrictions, APOLLO must know the return time of the pulse to within ~60 ns (~9 m one-way), and in practice, it is better to know it to within a few nanoseconds. APOLLO relies on an ephemeris program from JPL, which is based in part on previous LLR data, to predict the RTT of each pulse, given its launch time. At low elevations, the earth rotation projects to ~400 m/s, or 20 m/shot, so each shot's launch time is measured to 1 μs precision. This prediction of RTT is an integral part of both our data-taking operation, and our data reduction.

The RTT to the moon and back is ~2.5 s. The laser fires ~20 pulses/s, so at any given time, there are ~50 pulses in flight to the moon and back. The system synchronizes the firing of the laser to the predicted RTT to interleave the return pulses with the outbound pulses. Thus every ~25 ms, the system is either launching a pulse, and measuring the FIDs, or receiving a return signal and measuring the LUNs.

In the APOLLO system as of 6/2010, channels 3 and 5 (of the 16-channel TDC) are dead. Channels 8 and 14 are unreliable, with higher measurement deviations than the others. Channel 15 is the FPD channel. This leaves 11 properly functioning APD elements for measurement.

The fiducial APD times are about 5 ns *earlier* than the FPD, because the FPD has ~130 ns of deliberate cable delay, to make it arrive near the fiducials, so they all land close in TDC bins. This reduces errors due to TDC variation across its range. Standard deviations of FID-FPD offsets are about 120 ps for good channels, which is mostly from the deviation of the FIDs.

1.3 The Solar System Model

To get science information, our RTTs are fit to a model of the solar system [JW]. The model includes 33 dynamic variables: the 3D positions of all 8 planets, Pluto, and the moon, and the 3 rotation angles of the moon. The model uses Solar System Barycenter (SSB) coordinates, so the center

of mass of the solar system does not move. Therefore, the Sun is not tracked separately as a dynamic variable, since its location is determined by the locations and masses of the other bodies such that the center of mass of the solar system remains fixed.

The model includes all other effects needed for ~10 mm accuracy in the lunar range. Other effects include atmospheric delays, tectonic plate motion, earth tides (crustal deformation), the larger asteroids, and more. The model includes ~150 fit parameters. This may seem like a lot, but most of these are simple, such as the initial positions and velocities of all 33 dynamic variables, body masses, mass moments, 3D coordinates of the observing station and the lunar reflectors, etc. A much smaller number concern the subtleties of gravity, such as the PPN parameters. The model includes data from many measurement sources, including optical measurements, LLR, and others. The fit is done simultaneously to all data sources, taking into account the uncertainties of all measurements (though the method of accommodating uncertainty is a subject of much debate). There are over 16,000 LLR measurements alone, so the ~150 fit parameters are well constrained.

The model starts with a linearized (weak gravity) version of General Relativity (GR), and adds parameters that allow for deviations from GR, such as the PPN β and γ parameters. Assuming the equivalence principle leads to a many-body gravitational equation of motion stated in terms of acceleration; forces aren't needed because all bodies accelerate the same under gravity, regardless of their mass or composition. The EP equation of motion is

$$\ddot{\mathbf{r}}_{i\text{-point-mass}} = \sum_{j\neq i} \frac{\mu_j \hat{\mathbf{r}}_{ij}}{r_{ij}^2} \left\{ 1 \right.$$

$$\left. - \frac{2(\beta+\gamma)}{c^2} \sum_{k\neq i} \frac{\mu_k}{r_{ik}} - \frac{2\beta-1}{c^2} \sum_{k\neq j} \frac{\mu_k}{r_{jk}} + \gamma \frac{v_i^2}{c^2} + (1+\gamma) \frac{v_j^2}{c^2} - \frac{2(1+\gamma)}{c^2} \dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_j - \frac{3}{2c^2} \left(\hat{\mathbf{r}}_{ij} \cdot \dot{\mathbf{r}}_j \right)^2 + \frac{1}{2c^2} \mathbf{r}_{ij} \cdot \ddot{\mathbf{r}}_j \right\}$$

$$+ \frac{1}{c^2} \sum_{j\neq i} \frac{\mu_j}{r_{ij}^2} \hat{\mathbf{r}}_{ij} \cdot \left[(2+2\gamma) \dot{\mathbf{r}}_i - (1+2\gamma) \dot{\mathbf{r}}_j \right] \dot{\mathbf{r}}_{ij} + \frac{3+4\gamma}{2c^2} \sum_{j\neq i} \frac{\mu_j \ddot{\mathbf{r}}_j}{r_{ij}}$$

Other versions of the equation exist for fitting for EP violations. Those versions separate inertial mass from gravitational mass.

The earth and the moon are constantly moving, so a round trip time is only meaningful when

given for a precise "launch time." At the mm-level, the launch time must be given to at least microsecond accuracy, while the LLR data set spans over 40 years.

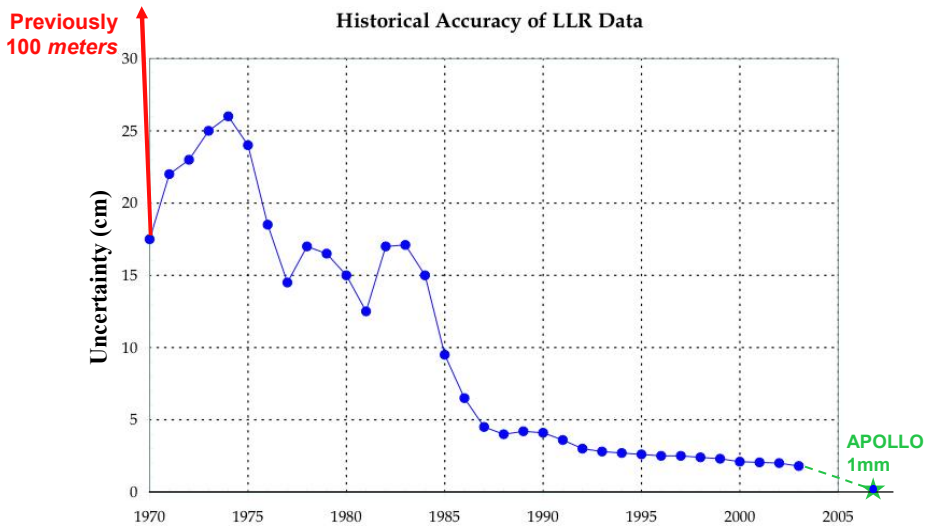


Figure 4: Fiducial and lunar detection histograms from real data, with model fit curves.

Figure 4 shows the improvements in LLR data over its history. APOLLO is clearly a big step forward. Consequently, the solar system model will need upgrading for mm-level accuracy. One example of a known omission in the model is solar radiation pressure on the moon, calculated to be about a 4 mm effect.

1.4 This Dissertation

My PhD dissertation comprises 2 subtopics. The first concerns the reduction of raw photon measurements into a "normal point," a virtual round-trip time estimate of low uncertainty. This is discussed in Chapter 2. The second topic concerns first photon bias reduction in APD single-photon detectors. This is discussed in Chapter 3. Chapter 4 concludes with results of reductions of real APOLLO data.

In this dissertation, I describe the successes, and some failures. It is important to document not only the things that worked, but also many of the things that didn't work. This illuminates the

kinds of things that were considered, and avoids needless future attempts at things already known to fail.

A separate document, "ResearchNotes.doc", augments this dissertation with simulation commands, and computer outputs, as well as other topics that are too detailed for this dissertation. ResearchNotes.doc approximately follows the same order of topics as this dissertation.

Notation: throughout this dissertation, references to computer programs or batch files are included in square brackets, so that future researchers can inspect and build on the existing work. Other readers may ignore them.

2 Generating Normal Points

2.1 Overview of Data Reduction

We briefly introduce several important issues here, and discuss each in much more detail later.

APOLLO, and most other LLR stations, generate "normal points." A normal point (NP) is a hypothetical launch time for a pulse of light, the round-trip-time (RTT) associated with that time, and its uncertainty. Normal points become the data points for a higher-level fit to solar system dynamics. Normal points also include ancillary data necessary for interpreting the RTT, such as temperature, relative humidity, and barometric pressure at the measurement station. APOLLO normal points include 10s to 1000s of individual photon measurements. APOLLO produces 1 NP per "run," typically of 5-20 min.

We need normal points because (1) higher level analysis, such as the JPL solar system model, does not need, and in fact cannot handle, thousands of individual photons. It is a computational burden. (2) The normal points include calculations and calibrations that are specific to APOLLO, and must be included for sensible results. Having APOLLO generate those normal points keeps that data processing with the experts who know it best. It is not feasible to train others to handle it, and others do not want to know.

APOLLO's goal is to minimize the long-term uncertainty of our normal points (measured over months and years), while controlling our systematic errors to below that level. Our experience and analysis lead us to believe that we can decompose the long-term uncertainty (σ_L) into two parts: (1) random uncertainty within a run (σ_r); and (2) run-to-run offsets, due to systematic errors that vary with observing conditions, but are fairly stable within a run (σ_s). Since we believe the random errors are not affected by observing conditions, these two are uncorrelated, so that

$$\sigma_L = \sqrt{\sigma_r^2 + \sigma_s^2}$$

It is σ_L we wish to minimize. In addition, besides minimizing σ_L, each NP must estimate its

own uncertainty, including both components, to properly weight the RTT.

Besides these two errors, there is also a fixed "range-bias" due to some mismeasure of the telescope geometry between the fiducial corner cube and the axes intersection. This range-bias is a fit parameter in the solar system model; it is a single number used as an offset for all APOLLO measurements. As such, a long-term fixed bias in our NPs doesn't really hurt anything: it is fitted out by the solar system model fit, and so is not a big concern for us. However, in general, we want to understand all aspects of our system, and therefore make some effort to understand and minimize range-bias. Of course, any offset in our measurement that drifts significantly with time would be a problem. As of 6/2010, our range-bias is a few mm. We do not discuss range-bias any further in this work.

To minimize σ_L, we must understand its components. Therefore, we examine both random uncertainties and systematic errors. Both effects enter symmetrically into our overall uncertainty, and neither is, a priori, more important than the other. Whichever one is bigger becomes the focus of our attention. An NP "uncertainty" is only meaningful in the context of both random (i.e., intra-run) uncertainties and systematic errors that are quantifiable and small.

Our data quality is very diverse. We encounter signal return rates that vary by ~two orders of magnitude, background rates that vary by a factor of ~5, retroreflector spreading that varies from a few dozen picoseconds to a few hundred picoseconds, and other variations. In general, one cannot optimize for all conditions with a single set of data reduction parameters. We choose here to optimize for the higher quality points of our data. In the big solar-system fit, the data are weighted according to the inverse square of the uncertainty. Therefore, the weak points don't count nearly as much as the strong ones. Making the strong data points better usually provides more overall benefit than making the weak ones a little better.

Measurement Challenges

APOLLO's significant challenges include (1) a target accuracy below our basic measurement resolution; (2) a diffusion "tail" in the distribution of detection times; (3) retroreflector tilt; and (4)

first photon bias. We introduce these challenges below, and discuss them all in much greater detail throughout the dissertation. By far the biggest challenges are the diffusion tail, and first photon bias (FPB).

APOLLO's fundamental time measurement is done with 25 ps resolution. We seek to "average" this down to a meaningful result of ~7 ps uncertainty. This can *only* be done in the presence of some kind of "noise" or variation that spreads the measurements over several or more 25-ps measurement "bins." Then, with sufficient measurements to average, one can approach the true value with reasonable confidence. In APOLLO's case, the variation comes from many sources, including the laser pulse width, lunar range changes with time, retroreflector tilt, and noise in the measurement chain.

A significant fraction of the APD detections are "late" due to a physical effect of the APD. We discuss this in more detail later, but briefly, some photons are absorbed below the shallow electric field. The resulting electrons are not accelerated toward the avalanche region, and wander randomly through the bulk. Some of them eventually wander into the electric field region, and finally trigger a detection. These detections can be many nanoseconds late.

We model the probability distribution function (PDF) of detection times as a gaussian "core" plus a "diffusion tail." (For a large number of detections, the PDF may be thought of as a "response function" of the system to each laser shot.) The core represents those photons absorbed in the high-field regions, which are detected quickly, but with a narrow, approximately gaussian time distribution. The tail of the PDF accounts for the fraction of detections that are late. The tail PDF varies like $\exp\left[-\sqrt{t/\tau}\right]$. This model is described in detail later.

Figure 5 shows a FID detection histogram, and a simulated LUN histogram of high libration (run 070103-064107). Because of lunar libration (changes in the lunar orientation), the retroreflector is generally not exactly normal to the incident laser pulse. This tilt spreads out the time distribution of returns. Therefore, the lunar profile is usually wider than the fiducial.

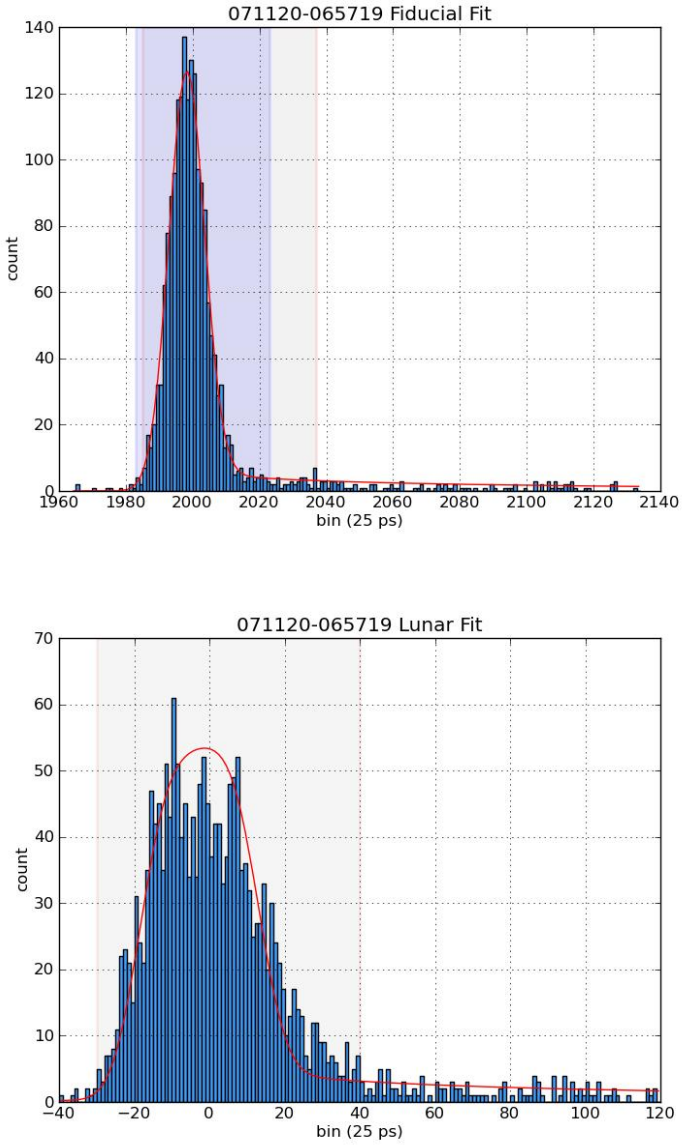


Figure 5: Fiducial and lunar detection histograms from real data, with model fit curves.

We can see from the detection profiles in Figure 5 that a simple average of detection times, including the tail, gives a bad result for two reasons. First, the uncertainty is high due to the wide width of the tail. Second, the tail must be truncated at some point, and the since the lunar return is generally a different shape than the fiducial, the difference in tail truncation might introduce a significant systematic error. Therefore, instead of a simple average, we try to compute a differential

measure of the RTT with low random and systematic errors.

Note that without reflector tilt, the tail would be much less of a problem. We could use a simple window of most any size for both FIDs and LUNs, and average within that window. If the windows were equivalently placed for FIDs and LUNs, the effect of tail truncation would exactly cancel. Background levels would still be a problem: they are much higher for LUNs than FIDs, but this is likely an easier problem to address. A uniform background has no preferred direction to pull the window (for most window positioning algorithms), and hence introduces only some variation, but no bias in the result.

Another significant effect is First Photon Bias (FPB). An APD triggers on the *first* detected photon, but we want a *representative* arrival time. This first-photon trigger results in a bias toward the early times when the detection rate is high enough that there is a significant chance of two or more detected photons in a single detector element on one shot.

The diffusion tail and First Photon Bias are challenges on their own, but they mix like drugs and alcohol: the problems don't just add; they multiply.

A final measurement challenge is how to calibrate and optimize the data reduction parameters. While someday we hope to use actual residuals from lunar ranging compared to the solar-system model, the model is currently only good to ~10 mm. Since we are tuning for 1 mm, we can only test our reduction methods by simulating the APOLLO system as realistically as possible with known range parameters, and comparing the reduction results with the known parameters.

Possible Areas for Future Study

There are several identified potential sources of error that this dissertation does not address.

TDC calibration: The Time-to-Digital Converter (TDC) has gains, nonlinearities, and offsets that change over time, and with temperature. Before and after each run, we use a known-pulse-width generator to calibrate the TDC. We fit quadratic curves to each of the 16 channels, and use those curves to derive calibrated time measurements. We do not address here the stability and accuracy of those calibrations. The simulator models the TDC as exactly 25 ps/bin.

Noise during FIDs: There is some evidence that electrical noise from the laser fire pulse causes the FID to FPD alignment to lose accuracy. We do not consider that here.

Illumination variation between FIDs and LUNs: No illumination mask on the APD causes the FIDs to overfill the APD compared to the LUNs. This delays the average FID, and biases the result toward shorter RTTs. The magnitude of this effect is not known, but is likely bounded by 20 ps, which about the difference between the diffuse and spot FID times.

Telescope temperature: we roughly estimate that seasonal variations in temperature cause a change in telescope mount height of about 3 mm. This introduces seasonal offsets in our data of about ±1.5 mm. We do not currently account for this effect.

Crosstalk: There is some small crosstalk between APDs. The rate is low, and there is most often a large time delay (> several nanoseconds) between a detection and its subsequent crosstalk-induced detection. Therefore, we believe crosstalk is insignificant, though it may contribute some to the "back porch" of the APD time profile (described later).

2.2 Research Plan

As much as possible, we seek to not only numerically evaluate our data reduction methods, but also to understand them physically and statistically. We therefore estimate the performance of various steps theoretically, and compare all our simulation results to the theoretical estimates. Our research plan for normal point generation comprises these points:

- Create Normal Point (NP) generation code for the two most promising methods of reduction.
- Simulate representative sets of data, and reduce them with the above methods.
- Optimize parameters of the code (e.g., window sizes, polynomial-fit orders, etc.) by comparing results to the known simulation parameters.
- Confirm that there is a region of stable behavior for all major data reduction parameters.
- Compare simulation results to theoretical estimates.
- Compare results between NP generation methods.

- Apply the best method to real data, and assess its effectiveness

2.3 Three Approaches to Normal Point Generation

Over time, we have considered three methods of normal point generation, each of which has many variants. We introduce them here, and describe them more fully later. In the following discussions, note that a **histogram** is made of actual measurements; a "probability distribution function" or **PDF** is a calculated function based on statistical theory and our empirical models. A histogram can be compared to a PDF, and the two can be tested for goodness of fit.

To get a measurement of RTT at all, APOLLO must predict the range to the moon for each shot to within ~10 m (~30 ns RTT). (We typically achieve ~0.3 m (~2 ns) prediction accuracy.) The range can change as much as ~20 m per shot. Therefore, all three methods start with the same steps: (1) subtract the prediction from each RTT, and (2) fit and remove a straight line from the prediction residuals. This leaves a set of lunar residuals which are "lined up" to the same time point, and can be aggregated in various ways.

All three methods then process these lunar residuals. Briefly, the three methods are:

- Correlation method: intended to be temporary until more accurate (and complicated) methods can be developed. It correlates the fiducial and lunar detection histograms, using the correlation peak to find the RTT. This method is not studied in this work.
- Augmented calculation: It averages the lunar residuals within a carefully chosen window, and with significant corrections.
- PDF-fit method: It fits the fiducial histogram for its parameters; from those and the retroreflector tilt, it predicts the lunar return PDF; fit the lunar histogram to the predicted PDF, for its location in time. The difference from FID location to LUN location is the RTT.

Before discussing these methods in more detail, we first describe the important aspects of the system that affect our timing measurement.

Where Data Reduction Stands Today

As of 6/2010, the current correlation method for generating NPs probably has several mm of systematic error, due to First Photon Bias (FPB) variation with signal intensity, and asymmetry effects of the APD tail, exacerbated by the reflector tilt. Our current methods of assessing our uncertainty can only compare multiple runs on a single reflector within a single night. These methods are necessarily blind to nightly systematic errors, which contribute directly to the long-term uncertainty of APOLLO NPs in the larger solar system fit.

2.4 APOLLO Timing Profile: The PDF Chain

We now describe the APOLLO timing chain in more detail, to better understand our measurements. Knowing the timing PDFs of fiducial and lunar returns allows some quantitative estimates of both systematic and random errors due to various impairments of the system. We perform several such calculations in later sections. Note that the difference in the timing chain between FIDs and LUNs is that the LUNs include the retroreflector on the moon. This is unavoidable, and prevents the RTT measurement from being exactly differential.

Figure 5 (above) shows a typical fiducial distribution, and on the same time scale (25 ps/bin $\rightarrow$ 4 ns), a lunar return during a time of high lunar libration. (Because the librations are roughly sinusoidal, there are more high libration times than low ones.)

These shapes are quite different, and demonstrate that a simple "differential" measurement of fiducial launch to lunar return time is not possible. For many calculations, and for some data reduction algorithms, it is important to have a model of the detection PDFs for both FIDs and LUNs. We describe the model later. First, we discuss the origin of these distributions. Figure 6 shows schematically the origin of the major contributors to the timing PDFs.

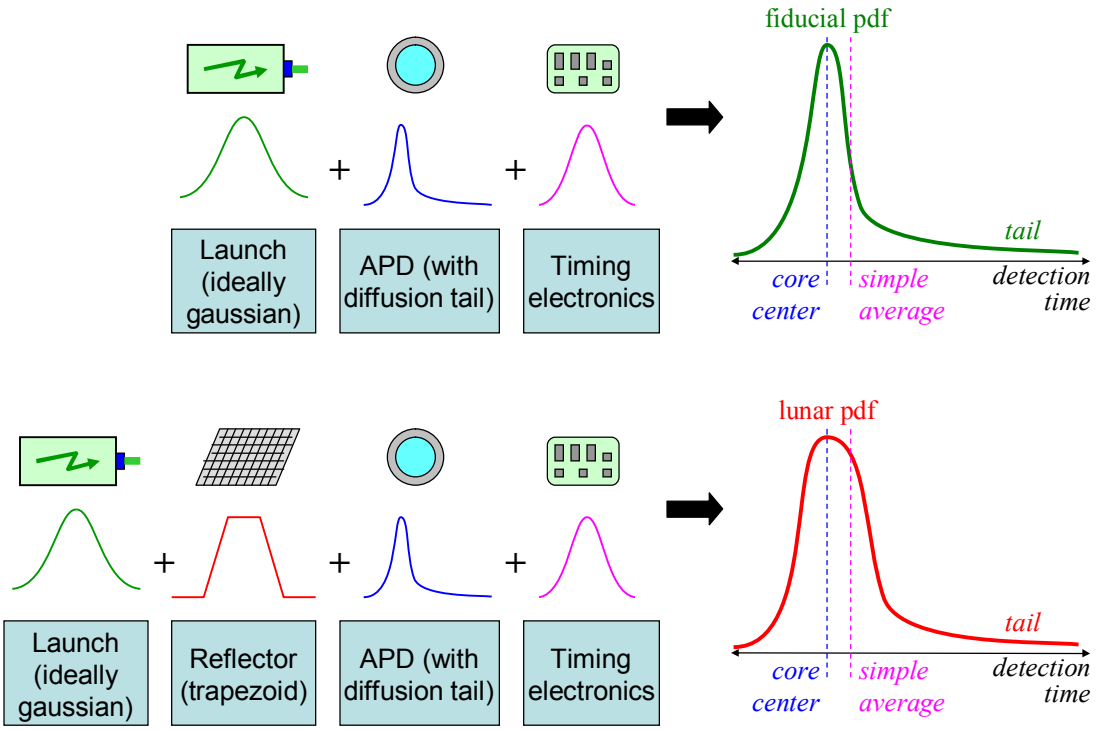


Figure 6: Main contributors to fiducial (top) and lunar (bottom) detection time distributions

Launch

The laser pulse is approximately gaussian. It is specified at FWHM of 95 ps, or $\sigma = 40$ ps, but our results show that it operates closer to $\sigma \approx 100$ ps. However, the launch profile includes the laser pulse temporal profile and likely some other things such as reflections, so is not exactly gaussian. The final PDF has a long APD tail. The atmosphere affects the spatial distribution of the laser pulse, but has no significant effect on the temporal distribution.

Reflector Tilt

The face of the moon visible from the earth varies ~9° east-west and ~6° north-south over time. There are 3 reasons for this: (1) the moon "wobbles" in its orbit (due to tidal forces acting on the moon's multipole mass moments), (2) the viewing angle changes as the moon progresses through its elliptical orbit, and (3) the viewing angle changes with the viewers position on earth (which varies with the earth's rotation). This variation in the apparent orientation of the moon is called **libration**.

The retroreflectors are aligned to nominally face the earth center at zero libration, but for any given observation, the reflectors are tilted with respect to the plane wave of laser light. This tilt spreads out the return pulse, over the time it takes the wave front to pass from the nearest reflector point to the farthest. For an idealized δ-function pulse, this results in a reflection intensity wave-form of a trapezoid; in other words, the retroreflector's PDF is a trapezoid, which is convolved with the temporal PDF of the incident pulse to produce the return pulse temporal PDF. For any given session, we can accurately predict the libration angles from prior lunar measurements.

Avalanche Photo-Diode (APD)

We now give an overview of the principles of operation of the APDs.

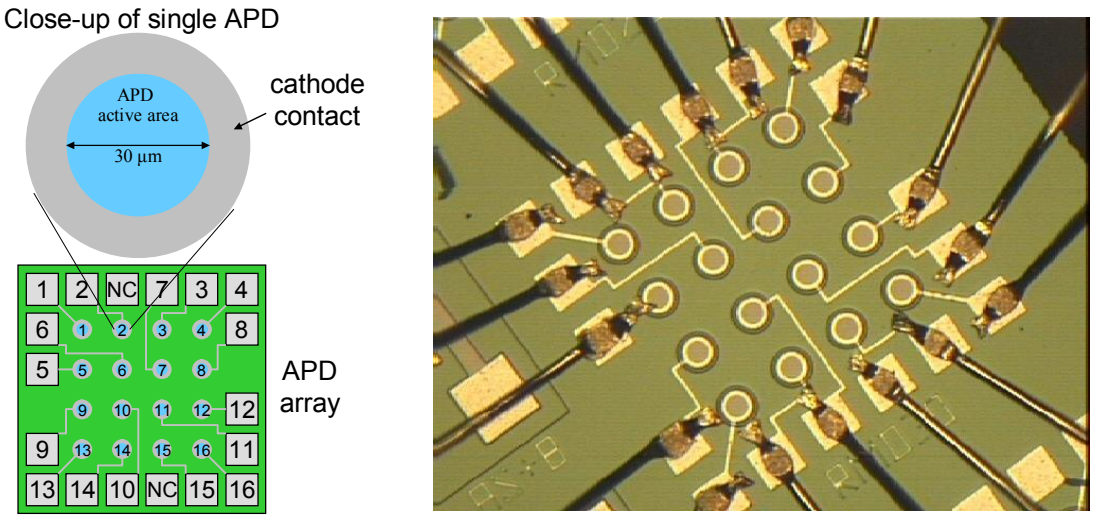


Figure 7: Avalanche Photo-Diode (APD) Detector Array

APOLLO uses an array of 16 APD detectors (only 12-13 APDs work right now; one works occasionally). The field of view of the moon on the APDs is 1.4 arcsec on a side. Each APD "pixel" is 0.35 arcsec on a side. Each APD is circular, with a 30 µm diameter active area (Figure 7). APDs are spaced 100 µm center-to-center. This geometry would produce an active area fraction of only 7%, i.e. we would lose 93% of our light into inactive space between the detectors. To recover this light, we place a lenslet array above the APD array. The lenslet array is a set of 100 µm square lenses which focus all the light entering them onto the active areas of the APD array.

The APDs are silicon diodes. They operate in reverse-bias above the breakdown voltage of the diode. This means that a single electron will quickly trigger an avalanche of current carriers, which produces a sudden current increase through the diode. Detection electronics measure the event time of the current increase.

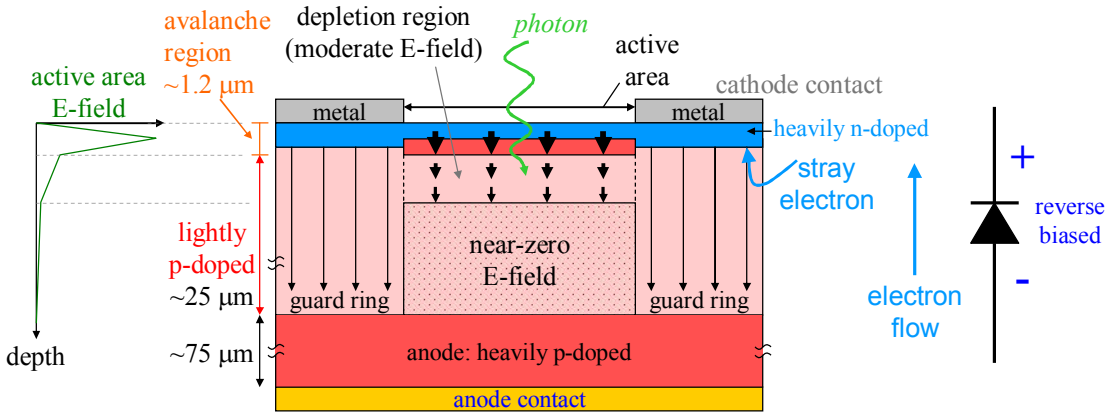


Figure 8: APD Structure: Cross-section of a reverse biased APD. Arrow thickness indicates electric field strength. A single photon triggers an avalanche in the high E-field region.

Figure 8 shows a cross section of an APD, under reverse bias for normal operation. The active area has a heavy n-doped surface, with a P-N junction to a heavy p-doped layer below. This junction is the "avalanche region," and is ~1.2 µm deep. The heavy doping of these layers results in a high electric field, but initially no current because the diode is reverse biased. Below the heavy doping is a lightly doped p-layer ~25 µm deep. It is depleted of holes by the reverse bias to a depth of several µm. This depletion region has a moderate electric field that quickly pushes any electrons up to the avalanche region.

Outside the active area is a guard ring which keeps stray electrons from wandering into the diode from the sides. The guard ring has a deep electric field that sweeps up electrons to be conducted out of the cathode without triggering an avalanche.

When a single photon is absorbed by the APD in the depletion region, it creates an electron-hole pair in the silicon. The hole drifts downward and is inert. (Note that in conduction theory, the term "drift" means the motion of an electron or hole being pushed by an electric field; it does *not* mean

random wander.) The electron drifts upward where it promptly enters the high field region. The high electric field accelerates the electron so much that it ionizes a lattice atom on impact, and creates *another* electron. Both go on again to create more electrons, in an increasing "avalanche" of electrons. These electrons conduct the current through the diode which is detected. The avalanche current stabilizes at a constant value, and continues until external electronics cut it off.

There is negligible crosstalk between APDs in the array; the detectors are independent.

The APD physics results in three important characteristics of its detection PDF:

(1) first photon bias (FPB);

(2) spatial dependence of detection time; and

(3) a tail of some late detections.

First, the APD avalanches on the first photo-electron. Electrons generated later can only join the avalanche in progress. We would like to measure the average arrival time, but are only sensitive to the first avalanche-producing photon. For example, if there are two detectable photons in a single shot, we will detect the first, resulting in a bias of half the difference in arrival times (Figure 9).

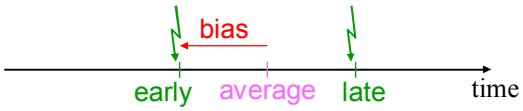


Figure 9: First photon bias: the APD responds to the first photon, not the average.

Second, the spatial dependence of detection time arises due to the growth characteristics of the avalanche "micro-plasma." Figure 10 shows how the micro-plasma grows laterally with approximately a fixed linear speed. The current through the diode is proportional to the area of the plasma region. For detections near the center of the active area, the plasma grows without obstruction, and therefore grows $\sim t^2$. When the plasma area is large enough, the current reaches the detection threshold of the electronics, and that is the measured time of the detection.

The plasma cannot grow outside the active area. For detections that originate near the edge of the active area, the growing plasma area hits the edge, and the plasma area increases more slowly

than the unobstructed t^2 growth (Figure 10). This delays the detection time by up to ~200 ps.

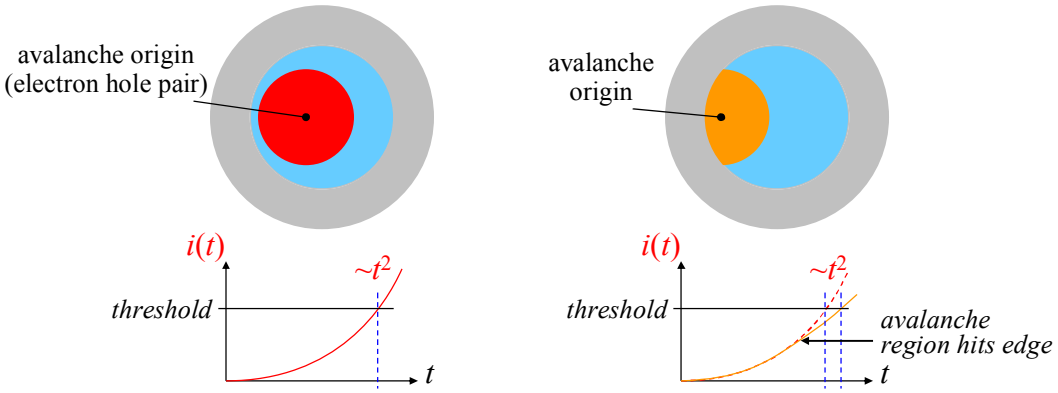


Figure 10: APD operation: Micro-plasma grows laterally at a near-constant linear rate. The current is proportional to the plasma area. (Left) When unobstructed by the edge, the current grows quadratically with time. (Right) When obstructed, current grows more slowly.

Lowering the electronic detection threshold reduces time variation with detection position, since a smaller fraction of plasmas will hit the edge before detection. However, this also increases the electronic sensitivity to noise, and can increase the random error of the time measurement.

Third, some photons will penetrate below the depletion region. The photo-electron is in zero E-field, and wanders randomly. Some fraction of these aimless electrons happen into the depletion region, where they are quickly accelerated to the avalanche region and trigger a detection. However, the wander time is quite long compared to other times, sometimes many nanoseconds. This effect produces the long tail on the late side of the main peak. One of our model parameters is the fraction of all detections that are in this tail of the distribution. The tail-fraction depends on the reverse bias used: higher bias widens the depletion region, which captures more photons in it, reducing the tail-fraction. Higher bias also increases the dark rate, since there is a larger volume in which thermal electrons can be produced that trigger a false detection.

Detection Electronics

The detection electronics record the time of arrival of a photo-electron. Each APD has a dedicated circuit that converts the APD current to voltage, and compares it to a fixed threshold. The comparator outputs feed a 16-channel Time to Digital Converter (TDC), which measures the time of

the transition as a 12-bit integer, to 25 ps resolution. The APOLLO control computer reads the detection times from the TDC.

2.5 Round-Trip Time Prediction

As mentioned earlier, for optimum APOLLO operation, the system must be able to predict the return time of a pulse to within a few nanoseconds. Before each session, operators run a stand-alone program, which uses the JPL ephemeris, to predict the RTT for hypothetical pulses launched at 5 minute intervals throughout the session duration. The session is typically scheduled for 1 hour, but the program generates predictions until "moon set," so there are ~40 predicted RTTs in the set. The program fits a polynomial of order 8-12 to those data, allowing a prediction of the RTT for any launch time in the interval, agreeing with the ephemeris to ~1 ps. The predictions include a number of factors, such as empirical earth orientation angles measured every few days. Usually, the predictions are accurate to a few nanoseconds. As described below, the prediction of RTT is an integral part of both our data-taking operation, and our data reduction.

2.6 Description of Data Reduction Methods

We now describe the three methods of data reduction. After all three are described, we analyze the performance of the two which are the focus of this dissertation.

2.7 Correlation Method Description

The first NP generation (aka "data reduction") method is the "correlation method," which has produced all APOLLO results so far. However, the correlation method is a temporary method, and is not analyzed here in detail. We will be brief, since this method is not a focus of our research. However, the FID alignment, fast photodiode offset, prediction subtraction, and linear fit and subtraction are all used in the later methods, as well.

Like all methods of reduction, there are a number of tunable parameters. We describe here the general method, without too much concern for the exact current values of all the parameters. This method likely has a significant systematic error for all reflectors, due to the libration spreading of the

LUNs compared to the FIDs, which makes the return histograms different and skewed. In addition, the offset for A15 is likely larger than A11 and A14 because of the larger reflector size skewing the LUNs even more.

The correlation method was originally motivated by desire to tolerate the laser firing double-pulses, about 2 ns apart. The laser had this problem for a few months (in 2006). The steps are these:

1. The FIDs over the different TDC channels are aligned by correlating their detection histograms. This means producing the correlation function, a function of time offset, and finding the time offset that maximizes the correlation. The correlation function is given by:

$$c(\Delta t) = \sum_{i=bins} a(t_i)b(t_i + \Delta t) \qquad where \quad \begin{cases} t_i \equiv \text{the TDC bin histogram times} \\ a(t),\, b(t) \text{ are the FID histograms for channels a \& b} \end{cases}$$

The correlation function is similar to a convolution, but with a "+" instead of a "–". Note that t_i and Δt have 25 ps resolution. Channel 1 is arbitrarily chosen as the reference channel, and all other channels' offsets from channel 1 are found from the correlation functions, as described above. Their histograms are time-shifted to align with channel 1, and summed to produce a single histogram. A second iteration use the combined histogram (instead of channel 1) as the reference, realigning each channel to produce a more robust "master" histogram. Then, for each channel, the method computes the total offset from the FIDs to the fast photo-diode (FPD). The FPD detects a launch time on (nearly) every shot, whereas the system detects a FID on only some shots. Therefore, the FPD is the time point to which FIDs and LUNs are referred (Figure 11). The average FID channel offsets are later applied to the LUNs, as well. Thus a FID detection is not required to match a LUN detection; only an FPD detection is required, and is almost always available.

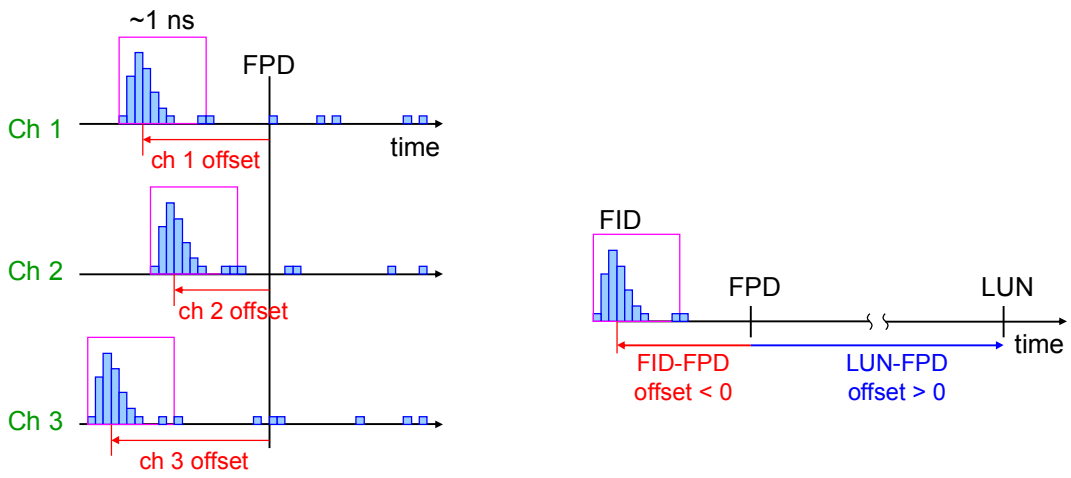


Figure 11: (Left) Individual channel offsets. (Right) FID and LUN timing offsets

In the system, the FPD detects the outgoing light first. The pulse proceeds to the secondary mirror, where a small corner cube reflects the fiducial pulse back into the system. The fiducial pulse goes through exactly the same receiver optics and electronics as the lunar returns, allowing a near-differential measurement. The FPD detection is much earlier than the APD detections of the fiducial, out of range of the TDC to cover them both. Therefore, a long cable delays the FPD signal so that it arrives at the TDC close to the fiducial APD detections. In fact, it actually arrives ~5 ns later than the fiducial pulse, even though it is detected much earlier (Figure 11). This has no impact on the measurement, but can be a little confusing.

It is possible to use other methods to align the channels with the FIDs, for example, aligning windowed averages could instead be done. In all three reduction methods, the method of aligning FIDs across detectors does not affect subsequent processing. We can separately evaluate FID alignment methods, and choose the best one.

2. The LUNs from all the channels are offset by the offset computed earlier in FID processing. RTTs are computed from LUN – FPD differences, minus the FID-FPD offset.

3. The RTTs are subtracted from the polynomial prediction, which over short times leaves a predominantly linear prediction error (Figure 12). This linear error is fitted, and removed. The linear fit uses a 2 ns window with the most points in it; without such windowing, background detections will

degrade the results. Now all the lunar returns can be combined into a single lunar histogram, similar to the master FID histogram.

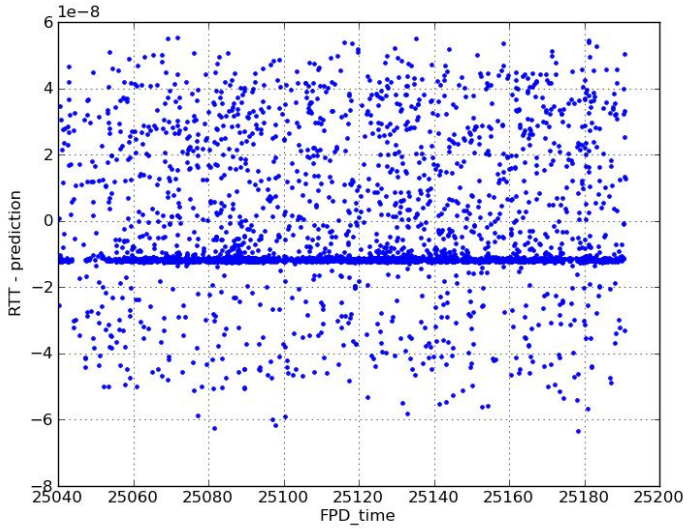


Figure 12: Lunar detection time differences from prediction (sec × 10^{-8}) vs. launch time (sec)

4. The FID and LUN histograms are correlated as a function of Δt to produce the correlation function histogram. (Note that on average, the correlation function histogram is the same as the histogram of directly computed RTTs, because the correlation function of two PDFs is the PDF of the difference of the random variables.) The correlation function has a clear peak at some value of Δt, and is roughly gaussian in that neighborhood.

5. A gaussian is fitted to the histogram bins with counts above 0.6 of the peak bin. The center of the gaussian fit is taken as the best estimate time difference. The deviation of the fit is the measurement population deviation, and dividing that by $\sqrt{\#\,\text{points}}$ gives the measurement uncertainty. The # points is estimated from the actual points within the linear-fit window (which is larger than that within the gaussian fit), minus the estimated background photon count.

6. The NP launch time is the average launch time of valid LUNs (of the linear fit). We round this time to the nearest 5 sec, to aid in comparing different reduction methods. The actual RTT as a function of time is found by adding back the linear fit and the polynomial prediction at the time chosen

for the normal point.

The correlation method has the following characteristics: It is comparatively simple. It is robust in the face of the double-pulsed laser, and background noise.

It likely has significant systematic errors from FPB and retroreflector tilt. To reduce FPB, we currently accept only those lunar detection gates in which a single detector triggered from a lunar photon. This generally selects the weaker intensity shots (weak from speckle), which have the least FPB. However, simulations show that there is still significant FPB even in these weak shots (see FPB chapter).

Retroreflector tilt makes the LUN histogram wider than the FID histogram. Each histogram is a sample of the population of histograms from the FID and LUN detection time PDFs. It is important to understand that correlation of the PDFs of two independent random variables is exactly the PDF of the difference of the random variables. Therefore, the correlation method, just like computing each RTT separately, does not improve the skew from retroreflector tilt. This results in a probably significant systematic error, toward longer RTTs, increasing with retroreflector tilt. However, this effect has not been quantified.

The correlation method has at least 3 impairments: Tails are included with same weight as core-photons, but have much larger variation, and so degrade overall uncertainty. Correlating FIDs to LUNs includes the APD variation of both, bypassing the potential of the FPD to improve on the fiducial (launch) time point, and thus improve overall uncertainty. Finally, the correlation function is a histogram which is fit, and therefore includes the general additional uncertainty incurred by fitting to histograms.

2.8 Augmented Calculation Description

The augmented calculation (AC) method originally started as fitting a polynomial to the measured RTTs of all the photons. The augmented calculation was hoped to give the best possible overall uncertainty (systematic + random) for each run, for clean (non-double-pulsed) data. It turned

out to be slightly suboptimal (described later), so the AC method was extended to be more like the others, i.e. to first subtract the prediction and a linear fit, before averaging the latter residuals.

This method uses some of our a priori information about the detection PDFs to help choose which detections are useful for estimating RTT.

Original (Preliminary) Version

This original version minimized the use of our RTT prediction, by using it only once in a non-critical way, to help find the valid LUNs. This addressed concerns voiced by several people about heavily using the prediction in our data reduction to compute NPs: was the prediction "steering" the NPs, so that the results were biased? Since this version of AC used the prediction only "lightly," it is easily defensible: one can readily see that the prediction does not steer the measurement significantly. This version is also fairly simple, which allows many people to quickly understand and be confident in it. However, this version of AC was later modified, because it produces a slightly larger variation than can be achieved by more quantitative use of the prediction. The original version processing follows these high-level steps (the final version retains these, and adds a few more; First Photon Bias (FPB) correction is omitted here for brevity):

1. Process each TDC channel's fiducials separately, computing an average and deviation of the offset between the fiducial APD returns and the fast photodiode (FPD). Filter out invalid fiducial APD detections.
2. Compute the round trip times (RTTs) from the lunar photons, again processing each channel separately. Invalid lunar APD hits are filtered out. Each shot's lunar photons are measured against their shot's FPD. Then the average fiducial to FPD offset for the LUN's channel is subtracted from the lunar to FPD measurement. This gives the lunar to fiducial differential measurement.
3. Fit a polynomial to the RTTs. To produce the normal point, evaluate this polynomial at the average launch time for all valid lunar returns. The uncertainty in the NP is the RMS residual error, reduced by the number of degrees of freedom of the fit.

Each step is detailed below. Tunable parameter values (allowed offsets, limits, etc.) may be optimized over time.

Fiducial Processing

The assessment of "valid" fiducial APD hits takes two passes, and each pass processes all 16 channels separately. In the first pass, any TDC measurement within ± 5 ns of the known APD-FPD offset of ~5 ns is accepted. These offsets are processed in TDC units of ~25 ps/bin. The method separates even and odd FIDs (because only one set matches the lunar APD illumination pattern), and chooses the uniformly illuminated ones by signal strength. We have found that the diffused APD illumination fiducials have a consistently stronger signal than the simply filtered point-illumination.

Within the chosen even or odd FIDs, the method then finds the ~1 ns window of offsets from the FPD with the highest number of hits. These are deemed "valid." The method computes the average and standard deviation (separately for each channel) within the "max hit window."

Lunar Processing

First fit: The method computes all potential RTTs by computing the LUN-FPD difference, and subtracting the average FID-FPD difference for that channel. At this point, all channel offsets are cancelled, and we have differential measurements. All the RTTs (from all valid channels) are combined, and the differences of them from the prediction are binned into a histogram with 25 ps bins. A sliding window of ~4 ns finds the maximum number of measurements within the window (see Figure 13). These measurements are used to fit a preliminary 4^{th}-order polynomial to the RTTs. This first fit is mostly to eliminate any dependence on the prediction.

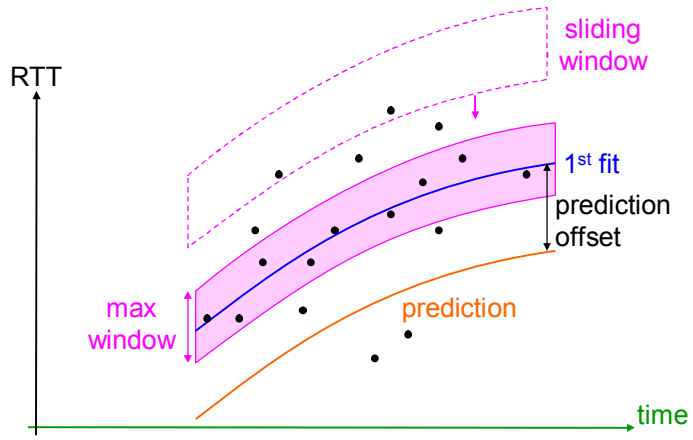


Figure 13: Choosing preliminary RTT data points for further processing

Second fit: The method then computes the fit errors to the first polynomial, and bins them into 25 ps bins. It finds the ~2 ns window containing the peak of the distribution (i.e., the most fit-error points). These are deemed "valid" RTT measurements. The method then fits a final 4th-order polynomial to these points (Figure 14).

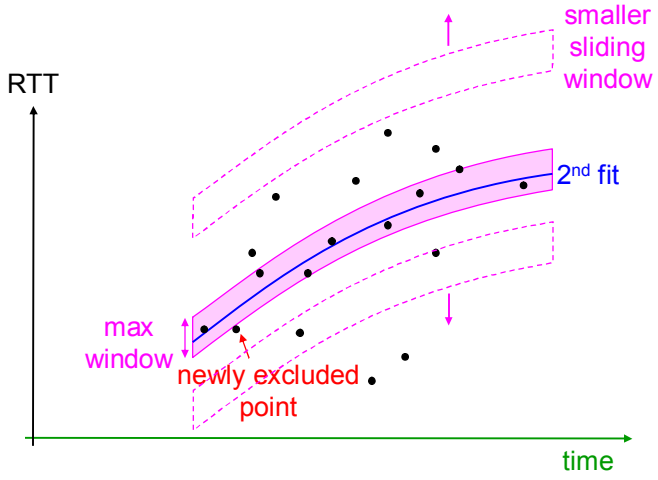


Figure 14: Choosing the final RTT data points for further processing

With no retroreflector tilt, it would be desirable that the method use the same window size for accepting lunar measurements as for fiducial measurements, so that the same amount of APD "tail" is included in both, and the averages of the two have the same tail-bias, which then cancels. In short, the

method would be insensitive to detector (or pulse) asymmetry. However, retroreflector tilt makes this simple scheme impossible, because it spreads out the lunar PDF tails by convolving the return PDF with the retroreflector PDF. Therefore, we must use other methods to eliminate the systematic error due to reflector tilt in the RTT calculation.

In addition, background detections skew the RTTs. The background detections are uniformly distributed across the final acceptance window. Therefore, when fitting a curve to the RTTs, these background detections "pull" the curve toward the middle of the window. For weak signal and/or bright background, this "pull" is significant, and must be corrected.

Normal Point: Range and Uncertainty

The NP launch time is the average launch time of valid LUNs in the final fit, rounded to the nearest 5 sec. The method then evaluates the final polynomial at the NP launch time to produce the NP RTT. The uncertainty is the RMS fit error (accounting for the degrees of freedom of the polynomial fit) divided by the square-root of the number of data points (Figure 15).

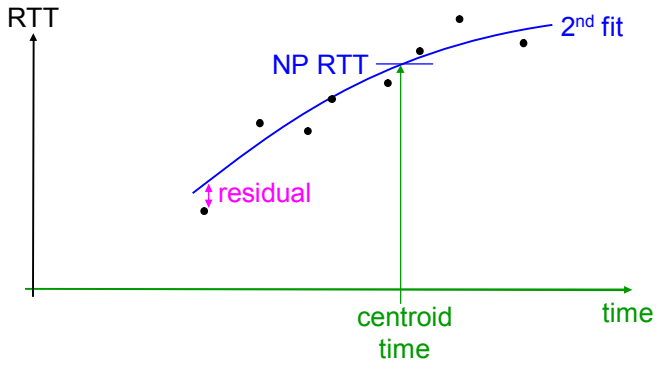


Figure 15: Schematic of 2nd polynomial fit to data. The curvature is exaggerated for illustration.

In the analysis section, we show that this method has higher uncertainty than the final method, because the higher order polynomial fit can follow random noise in the RTTs. Therefore, this method was changed to subtract the prediction, and use a linear fit to those residuals.

Final Augmented Calculation Method

By using more a priori information, i.e. using more of the prediction, we can get a tighter uncertainty on our normal point. Our predictions, based on the JPL ephemeris, typically have slope and offset errors from our measurements, but no higher order errors are significant over the time-scale of a run. This statement is justified by study of the ranging geometry and its time evolution. Therefore, by subtracting the prediction from our measurements, we reduce the problem of computing a normal point to that of finding the slope and offset of these resulting residuals. In other words, we fit a straight line to the prediction residuals. In principle, if there were, say, significant 2^{nd} order errors in the prediction, we could fit a quadratic to the residuals instead of a line.

The concern about this method is again the difference in the distribution of the lunar detection times (LUNs) compared to the fiducial detection times (FIDs), due to reflector tilt, and also their different background rates. These differences can lead to systematic errors that depend on the observing conditions, and skew the results. This method is constructed to reduce those errors to acceptable levels. Some steps are the same as the original version of this method:

1. Process each TDC channel's fiducials separately, computing an average and deviation of the offset between the fiducial APD returns and the fast photodiode (FPD). Invalid fiducial APD detections are filtered out.
2. Compute a single fiducial FPB based on the fiducial sigma, and the entire run's average fiducial return rate.
3. Compute a set of raw RTTs, one for each return photon detected, by subtracting that channel's average FID-FPD offset from the LUN-FPD difference. This gives the lunar to fiducial differential measurement. Invalid lunar APD hits are filtered out.
4. For FPB, window the histogram to the maximum counts within ~2 ns window. Compute residual uncertainty for lunar FPB correction. Adjust RTTs for lunar FPB.
5. Subtract the predicted RTT from each measured RTT.
6. Fit a line to the residuals. To produce the normal point, evaluate this line at the average

launch time for all valid lunar returns, and add back the prediction. The uncertainty in the NP is the RMS residual error divided by $\sqrt{N - \text{dof}}$, where N is the # of detections contributing to the fit, and dof is the number of degrees of freedom of the fit (= 2 for a line). First photon bias is the subject of a later chapter.

2.9 PDF-Fit Description

The PDF fit method is hoped to reduce systematic error, by being less sensitive to windowing choices, and therefore, in particular, to retroreflector tilt and tail asymmetry. As mentioned earlier, the retroreflector tilt spreads out the return signal in time, causing any data windowing to truncate different points on the retroreflector differently. This differential truncation can result in a systematic offset of the computed RTT that depends on retroreflector tilt. The PDF fit method does not depend explicitly on averages computed within windows, and the RTT PDF is predicted from the known reflector tilt angles. If our fitting model is good, the fit parameters, including the position in time, of the lunar returns is hoped to be largely insensitive to the fit method.

This method is more complex than the others. It includes the general increase in sigma when fitting to histograms. I am concerned about possible instability on low-signal returns.

As always, FIDs across channels can be aligned in one of several ways. I aligned them by averaging their windowed returns, because measurement theory says that is minimum deviation. The method of aligning fiducials across detectors does not affect subsequent processing. In particular, it does not directly affect the LUN to FID time comparison.

The PDF-fit steps are summarized below. We later provide more detail:

1. Process each TDC channel's fiducials separately, computing an average and deviation of the offset between the fiducial APD returns and the fast photodiode (FPD). Filter out invalid fiducial APD detections.
2. All the fiducial channels are aligned by their average FPD offset, into a unified fiducial histogram. We fit this histogram for FID model parameters: core-average, sigma, tail

fraction, and decay time.

3. Compute a single fiducial FPB based on the fiducial sigma, and the entire run's average fiducial return rate.
4. Compute a set of raw RTTs, one for each return photon detected, by subtracting that channel's average FID-FPD offset from the LUN-FPD difference. This gives the lunar to fiducial differential measurement. Invalid lunar APD detections are filtered out.
5. Compute the differences in RTT from the prediction, and bin them into a histogram.
6. For FPB, window the histogram to the maximum counts within ~1.5 ns window. Compute residual uncertainty for lunar FPB correction. Adjust RTTs for lunar FPB.
7. Subtract the predicted RTT from each FPB-adjusted RTT, and bin residuals into a histogram.
8. Window the residuals to the maximum counts in ~2 ns window. Fit a straight line to the residuals.
9. Subtract the linear fit to make new residuals, binned into a new histogram.
10. Compute the predicted RTT PDF from the fitted fiducial core_sigma, tail_fraction, and t_decay, the known reflector tilt, and an estimate of background rate.
11. Fit the RTT PDF to the RTT histogram for best location (core-average), using the fixed parameters of the previous step.
12. Use the fiducial core-average time, fiducial FPBias, and lunar core-average time to offset the straight-line fit.
13. Find the average launch time of the lunar returns to use as the Normal Point "launch" time. Finally,

    RTT = (lunar core-avg) - (fiducial core-avg) + fiducial FPB

    + straight-line fit + prediction.

First photon bias (FPB) is the subject of a later chapter.

2.10 Data Reduction General Concerns

Before examining the two data reduction methods separately, there are some general concerns relevant to many reduction methods.

Definition of Uncertainty: In some of our APOLLO runs, we get a small enough number of photons that our estimate of the uncertainty is itself somewhat uncertain. This makes the concept of a confidence limit relevant. For example, should we be 95% confident that our uncertainty is as good or better than what we put on the normal point? Jim Williams [private communication, 2007] says that our uncertainty isn't very critical, and we should use our best-estimate 1σ value.

Systematic errors as false signals: Some systematic errors depend on the observing conditions, which can cause a systematic error to mimic a science signal. For example, if the NP has a systematic dependence on lunar background, it may show an offset at full moon, when the moon is bright, as compared to new moon or waning crescent, when the moon behind the reflector is dark. This could mimic a cos D signal, which is exactly the signature of the equivalence principle, and other science signals of interest. Similarly, a systematic error that depends on retroreflector tilt would corrupt the lunar orientation fit parameters. Therefore, it is crucial that we minimize any such effects.

Systematic errors as increased uncertainty: APOLLO may have an absolute fixed range offset that does not vary over time. The JPL program fits for such a "range-bias" for our measuring station. Usually, the bias is considered fixed for all time, though JPL has run analysis experiments to look for a time drift in bias. In particular, our fiducial corner cube should stay fixed on the telescope, to keep range bias constant. However, its exact location is not critical, since the range-bias parameter includes that, and other (possibly unknown) sources of bias.

Given the above, in our own analysis, any absolutely fixed bias (fixed over all times and measurement conditions) will be "fitted out" with the range-bias parameter, and won't affect our results. However, some influences may affect our results on a session basis: one night the results are offset by one amount, and another night they may be offset by another. To the extent that these offsets are independent of lunar range, they don't bias any science results. Since the JPL fit spans years, these

"session offsets" are equivalent to random error in our measurement. Controlling session offsets to better than 1 mm is essential to achieving the 1mm goal of APOLLO.

Some possibly significant sources of session offsets are data reduction sensitivity to (1) overall signal intensity (through First Photon Bias); (2) retroreflector tilt, which changes the return PDF, and contributes to truncation offset; and (3) seeing, which causes varying intensity and spatial distribution of the return throughout a run.

Retroreflector tilt is only a problem because it interacts with the APD temporal PDF, which is asymmetric due to the trailing tail. We must be careful not to introduce undue error when estimating the "average" time of arrival. With no reflector tilt, the lunar returns would have tails similar to the fiducials (altered only by FPB), and there would be little need for concern: any window would truncate the tails the same on the fiducial as on the lunar signal. However, the tilted retroreflector spreads out the LUNs, both gaussian core and tail, so a simple window truncates the tail differently on the fiducial compared to lunar signal. Such truncation contributes to the condition-dependent systematic error, σ_s.

Are the tails worth extracting information from? Our tail model indicates the standard deviation for tail photons are an order of magnitude worse than core photons. If used, they should be weighted ~0.01 as much. The tail fraction is ~20% of measurements, which means they can theoretically contribute to reducing our variance by < ~0.2%, and uncertainty (sqrt(variance)) by half that. In other words, excluding the tail has no significant effect on our random uncertainty. However, we must still be careful that they do not introduce a significant systematic error.

Round off error: Polynomial fit coefficients have the potential for significant round-off error. When summing a lot of numbers, the precision in the sum is reduced by the number of digits in the sum. For example, with exactly 18 digits of machine precision, a sum of 10,000 numbers would include only the most significant 14 digits of the last few numbers added into the sum. The sum (and average of we divided by 10,000) would only be good to about 14 digits, 4 digits less than the machine precision. Therefore, the average of 20,000 RTT measurements have a round-off ~ 4-5 digits worse than the machine precision. When fitting higher order polynomial coefficients, we must sum powers

of the time variable, and can possibly lose even more precision. I use x86 "long double" precision for the statistical sums, which insures a minimum of 18 decimal digits precision, and up to 21, depending on the mantissa. Losing < 5 digits leaves at least 13 digits, which is sufficient for picosecond resolution at ~2.5 s RTT. However, as a test, I also computed fits with "twice-precision" sums, where each sum is spread across two long-double numbers, and hence has at least 36 digits of precision. The results were identical to simple long-double precision, so in general, I don't use the twice-precision summation method.

In addition, proper rounding of quantized time values is critical to avoiding system errors of ½ bin. Our TDC provides 25 ps bins, so half is 12.5 ps, which is large (almost 2 mm). The criticality of proper rounding is a general feature of any system that seeks to produce an aggregate result that is higher resolution than the raw measurements. In APOLLO's case, the raw measurements are 25 ps ≈ 3.5 mm, but we seek ~7 ps = 1 mm results.

Prediction steering the results: Because the range prediction is deeply embedded in the data reduction, several people have expressed concern that the prediction might inadvertently drive the reduction to agree with it. [I, Chris Stubbs, and Eric Adelberger voiced similar concerns at the JPL data reduction meeting 3/18-20/2007.] In APOLLO, some use of the prediction is required, because we can only find the signal at all if we're close enough to it to extract it from the background. In the end, both analysis and our residuals suggest that our results are only slightly affected by the prediction, and not significantly driven by it. Also, the ILRS recommended method of reduction for satellite laser ranging uses a prediction, and subtracts it from the measurements, similar to what APOLLO does. Nonetheless, it does take some explaining to newcomers to convince them that reduction using the prediction is valid. In short, it is valid because APOLLO results are much more precise than the original prediction, which can't predict down to the millimeter level. Also, it can be shown that higher order terms are from simple circular motions, such as the earth's rotation, which happen on time scales much faster than the lunar orbit.

Also note that in general, fitting to a function, versus subtracting a baseline, fitting, and

adding back the baseline, produce different results. This is because the fit coefficients are interdependent. This dependence is why stepwise regression produces suboptimal results, and techniques like backward elimination are used to improve stepwise regression performance. It is also why polynomial fits must solve simultaneous equations for all the coefficients at once.

Note also that there is a difference between using the prediction polynomial to *identify* legitimate lunar photons, and then fitting those as-is (without subtracting), versus using the polynomial both for identification and as a baseline function. This was the motivation for the original AC method, which did not subtract the prediction. However, I showed this to produce unnecessary additional uncertainty, which is avoided by subtracting the prediction first.

Use a priori information to maximum advantage: Note that concern over using the prediction too much is directly opposed to the concept of using as much a priori information as possible to get the best results. In general, using more information enables one to produce better results. One must be careful, though, because if our prior "knowledge" is wrong, it becomes "prejudice," and worsens the results. Such prejudice is often insidious and hard to detect, precisely because it conforms to our expectations.

Photons in the tail of the distribution: In general, including higher-variance points like the tail points can help if they are weighted inversely to their variance. Weighting them more than that degrades overall uncertainty. A simple correlation weights them the *same* as all other points, which degrades the within-run uncertainty, σ_r.

Statistically valid samples: This caveat might go without saying, but is important enough to be explicit: During algorithm development, it is tempting to focus on a single run or simulation, and make adjustments based on the result of analyzing that one run. However, there is significant variation from run to run, and making adjustments from a single run generally leads to biased results, since a single run is unlikely to reasonably represent the population of runs. One must resist this temptation, and make adjustments based on statistically valid samples of runs.

Direct Calculation vs. Fits to Histogram

A common method of computing measured results is to take a set of data, bin it into histograms, and fit a model of the PDF to the histogram by varying parameters of the model. For continuous measurements, the binning process discards some information. In our case, however, our fundamental TDC measurement is quantized to 25 ps, and that is the size of our histogram bins. (If we rebinned the data into larger bins, then we would be discarding more information.)

A second potential problem with fitting to find the location is that broad, flat parts of a histogram contribute nothing to the fit position, and are hence lost for locating FID time, LUN time, or RTT. It is the edges that carry position information in a histogram fit (or correlation); the steeper the edge, the more information it carries.

We must ask: to find the best estimate of detection time, is it better to average the TDC values, or to bin them, and fit a curve? Before examining this question in more detail, we must understand some general issues of fitting model PDFs to histograms.

What figure of merit to use? The standard answer is to use a χ^2 parameter of the model PDF, scaled to match the histogram amplitude, compared to the histogram bin counts as the data points of the χ^2 parameter. The fit parameters that minimize χ^2 are taken as "best." The uncertainty in the bin count, used to weight the error in the bin count, is the model bin count. This applies because the bin count is binomially distributed, and with a large number of bins, the variance is approximately the bin count. However, there are some problems with χ^2. The biggest problem, which is contrary to expectations and good results, is that the histogram fit is driven by the *tails* (more weight), than by the bulk of the data. But the tails are usually the worst modeled, and the noisiest (most affected by background)!

Another possible figure of merit (FOM) is simply the sum-squared-error (SSE). In this case, the squared-error in the bin count is not weighted (as it is with χ^2). In some cases, SSE yields a better result than χ^2.

We examine the FOM question more later.

Overestimating the low-count model: The χ^2 FOM is based on some idealizing assumptions about the model and data. The general formula for χ^2, for any type of fit, is:

$$\chi^2 \equiv \sum_{i=1}^{N} \frac{\left(meas_i - model_i\right)^2}{\sigma_i^2}$$

where $meas_i \equiv$ the measured value of the i^{th} data point

$\sigma_i^2 \equiv$ variance of the i^{th} measurement

$model_i \equiv$ the model value of the i^{th} data point

For a fit to a histogram, the measurements are the bin counts, and the σ_i^2 approximately equal the bin count. Then we have:

$$\chi^2 \equiv \sum_{i=1}^{N_{bins}} \frac{\left(count_i - model_i\right)^2}{model_i}$$

where $count_i \equiv$ the measured count in the i^{th} bin

$model_i \equiv$ the model average count in the i^{th} bin

The bin counts in a histogram are binomially distributed (not Poisson, as some authors state). The variance of a binomial random variable is exactly $np(1 - p)$, where np is the average bin count, and p is the probability that a measurement falls into this bin. With a reasonably large set of bins, p is small, and $(1 - p) \approx 1$. Therefore, the bin count $np \approx$ Var(bin count). Thus $model_i$ is used as the variance of the bin count, and it therefore appears in the denominator of the χ^2 sum. Notice that low bin-count bins carry *more* weight than high bin-count bins. Also, the bin counts of any sample are always integers, but the model counts, which are averages, are real numbers.

This formula assumes that the bin counts are large enough, and that all parameters have reasonably broad effect. If there are a lot of low-count bins in the histogram, they will carry most of the weight. You may find that the fit tends to overestimate the low-count bins, and underestimate the high-count bins (Figure 16). The best fit requires normalization: the sum of bin counts equals the sum of the model predicted counts, as in [Bev p273b]. In other words, the sum of overestimates and underestimates must be zero. But since low-count bins weigh more than high-count bins, and since an overestimated model reduces χ^2 (the model value $model_i$ appears in the denominator of each χ^2 term), the overall χ^2 is reduced if low-count bins are overestimated, and high-count bins are underestimated.

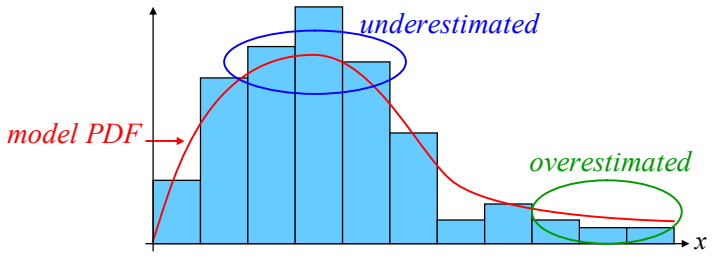


Figure 16: χ^2 is artificially reduced by overestimating low-count bins, and underestimating high-count bins.

This effect can only happen if your model has the freedom to "bend" in the way necessary: i.e., it can be a little high in the low-count regions, and simultaneously a little low in the high-count regions. Most realistic models have this freedom. If the model is reasonably accurate (where the ideal reduced-χ^2 would be 1), this effect can cause reduced-χ^2 to be consistently less than 1 (which should not happen consistently).

APOLLO models definitely violate the idealizing assumptions: the core-average and core-deviation parameters affect almost exclusively the main bump, and the tail-fraction and tail-decay constant affect almost exclusively the tail. Therefore, our fiducial model has exactly the problematic flexibility. We therefore expect this effect to overestimate the tail fraction and decay constant, and indeed, that is exactly what we find.

Overweighting the low-count bins: For model PDFs with rapidly decaying tails, such as exponentials or gaussians, it is possible to have model bin-counts that are tiny. The weight of a bin in the χ^2 sum is the inverse of the model bin-count, so tiny counts have huge weights. This can severely distort the results. I believe I have observed this in many fits, so I adopted a universal criterion that the maximum weight of any bin is 1, regardless of the actual model bin-count. This appears to have worked well, though I have no quantitative studies of the effect of imposing a maximum weight.

Scaling the model PDF: When fitting a PDF to a histogram, the amplitude of the PDF is usually normalized to some value, and it must be scaled to match the histogram bin counts. One could allow the scaling to be a fit parameter, but this tends to float the curve above the low-count bins, to

make their χ^2 contributions smaller. This is similar to the low-bin count effect previously described, but worse since there is *no* cost to overestimating (it doesn't increase errors elsewhere), and gives an incorrectly low χ^2.

In general, it is better to have a constrained normalization. One can sum the PDF over the fit region, and compute the scale factor that then matches the exact counts from the histogram. Constraining normalization demands that the sum of the positive errors exactly cancels the sum of the negative errors. This forbids the trial model curve from floating above the low-count bins, while still following the other parts of the histogram. This is similar to, but better than [Bev p273] (which normalizes the entire curve, including regions excluded from the fit), because we scale only the subset of the histogram and PDF that is being fit.

However, constrained normalization can introduce unexpected false minima. For example, when the trial core-average is far from the histogram peak, we are essentially fitting the model flat tail to the histogram peak. Constrained normalization allows the very flat tail to be "normalized" by a factor of millions, which ends up being a flat line through the middle of the histogram peak. The flatter the line, the better the "fit," even though reduced-χ^2 is in the hundreds, which actually drives the core-average parameter to worse values. To fix this, we recognize that any realistic fit will include more than 0.2 of the model PDF area, and will therefore have a scale factor < 5×(# points). Clamping the normalization factor at this upper value causes more distant tails to consistently have higher reduced-χ^2, and eliminates the false minimum. There are other ways to fix this false minimum, as well, such as constraining the core average to fall within the fit bounds of the histogram. The key point is that the fits are often not stable, and require lots of care and automated sanity checking to regularly converge to meaningful results.

Choosing how to window the histogram: Our detection time histograms have long, low tails which fade into the background detections. For the fiducial data, it is only sensible to fit a region of the histogram that is well above the background. I always start with a 4 ns window, and then narrow it. One way to choose the window is to start from the earliest bin with some minimum count

in it (fit_bound), and end at the latest bin with at least 'fit_bound' counts. One must then verify that the critical parameters are largely insensitive to the value of 'fit_bound'. Generally, 'fit_bound' must be small enough to keep the # bins high enough (ideally > ~30), and large enough to be above the background. Windowing also helps avoid being driven too much by the tails, which don't contain any significant information for our use. For low return runs, it may not be possible to keep the number of bins we would like.

When fitting to histograms, one must fix the data window of the histogram, i.e. that part used for the fit, for all variations of parameters. One cannot let the variational theoretical fit choose the histogram columns dynamically, as this alters the included data during parameter variation. The result is sudden discontinuities in χ^2, which cause many local minima, and highly unstable fits. The fits vary wildly depending on the starting values of the parameters.

I observed this instability very clearly when I tried using the variational fit function to truncate the data, instead of the actual data counts.

Simulation of Direct Calculation vs. Fit to Histogram

We now return to the question of whether there is a difference in results between a direct calculation of a parameter, and a PDF fit to a histogram of the same data. To examine this question, I simulated a simple approximation to our data: estimating the average and standard deviation of a gaussian population from samples of it. This is relevant because a big part of our data distribution is a central gaussian core of measurements.

I simulated multiple samples, each of a fixed size, from a gaussian distribution. For each sample, I first computed the average and standard deviation using the standard formulas:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N-1}}$$

Without quantized measurements, the average and variance are mathematically proven to be the best possible unbiased estimators (i.e., minimum variance). However, the measurement quantization (our TDC bins) may affect these statistics in a non-obvious way, and indeed my results

show that it does.

Secondly, I binned the sample into a histogram, using the TDC bin size of 25 ps, and fitted for minimum χ^2, varying both the average and standard deviation as the parameters of the model.

After simulating many such samples, I computed statistics on the parameters estimated from both the direct calculation and fit-to-histogram, to compare the two methods. I performed such simulations for a wide range of sample sizes, covering the range of N = 30 to 10,000, which is about the range of the number of photons in our normal points.

When performing such tests, it is important to vary the actual population average uniformly across a histogram bin, as happens with real data. Otherwise, the rounding of measurements to bin centers can artificially reduce variability, yielding meaningless results. For example, if the population average is at a bin center, and the population variation fits entirely in that bin, the results of binning will give a perfect average every time, which is clearly unrealistic. Also, to avoid biases, it is critically important to properly round all the results, both in binning the data to simulate the hardware, and in interpreting the bins, to analyze both real and simulated data. One must write the code very meticulously [degrade.c].

I ran the simulations with standard deviations of 0.16 ns, to represent typical fiducial-scale variations; and 0.25 ns, to represent typical lunar-scale variations. I then computed the factor by which uncertainty increased, and the "photon factor," or the ratio of photons needed when fitting to a histogram to equal the uncertainty of direct calculation in determining the position of the average (Figure 17). The photon factor is just the square of the uncertainty increase.

N	σ = 0.16		σ = 0.25	
	Uncertainty increase	Photon factor	Uncertainty increase	Photon factor
30	1.13	1.27	1.15	1.33
100	1.09	1.18	1.11	1.24
300	1.05	1.10	1.07	1.14
1000	1.04	1.08	1.05	1.10
3000	1.03	1.06	1.03	1.07
10,000	1.02	1.05	1.03	1.07

Figure 17: Uncertainty increase and "photon factor" for various sized runs, with different σ.

Each entry in the table used 10,000 simulated runs. This shows that the degradation from fitting to histograms is small for large photon counts, and somewhat significant at small counts. However, the small photon counts correspond to the most-uncertain normal points, which are the least weighted of the overall data set. Some earlier simulations had suggested a more significant difference, but these did hold up under further investigation.

Comparing the smaller σ to larger σ results: The larger deviation of the starting gaussian increases the number of bins, however the individual bin counts decrease. In terms of fit parameter uncertainty, the increased number of bins should exactly compensate for the decreased bin counts, but more bins effectively increases the binning resolution. The finer resolution decreases the quantization noise to sigma ratio, and I would expect the results to be slightly better. It is not clear why the larger σ results are consistently worse than the smaller σ.

2.11 Fitting the Fiducial PDF

Much of the analysis, and all of the simulations, require a model for the PDF of our detection times for both FIDs and LUNs. Here we consider the FID PDF. (We consider the LUNs later).

I have chosen my fit parameters to have a direct physical meaning, when possible, so that results from different tests can be compared directly, without the need to "transform" either set of parameters. I considered these 5 parameters:

1. Gaussian core average: the location of the peak of the gaussian component of the PDF.
2. Gaussian sigma: the standard deviation of the gaussian component.
3. Tail fraction: the fraction of the APD detections that are delayed by the "tail." This fraction is of the photons remaining after any back porch (below) is subtracted off
4. Tail decay: the time constant in the decay factor of the PDF tail.
5. Back porch fraction.

For FIDs, the background is insignificant, and ignored. In the end, only the first four parameters proved useful: the "back porch" parameter was highly degenerate with others, and obscured their values (described later).

I found the fit parameters by a numerical optimization in which the parameters evolve through parameter space finding progressively better fits (smaller figure of merit).

When fitting a model to data, all data must be truncated to some starting and ending points on the curve, such as where the signal (i.e., the modeled part) is cut off by measurement limitations, or the signal becomes small compared to unmodeled noise. Note that as with all histogram fitting, such data truncation must be based on the bin counts themselves, with no reference to the variational fitting function as it evolves over the parameter space. If the truncation depended on the variational fit function, there would be discontinuities in the figure-of-merit that would lead to multiple local minima and unstable fit parameters.

Numerical Fitting Functions In General

Even professional fitting functions (e.g., Numerical Recipes) don't protect against common failure modes, which arise frequently in processing noisy data. My optimization routine protects against a large number of failure modes, including invalid parameter values, search windows that collapse to zero width due to finite precision, unreasonable 'step' increments (which can result from collapsed windows), and convergence to 0. In the course of my research, I encountered all of these problems, and more.

Optimization algorithms iteratively evaluate a figure of merit function, *fom*(**p**), as a function of the list of parameters, **p**. The given trial function, *fom*( ), *must* protect itself against all unreasonable parameter values, since the higher-level general search function cannot know what is reasonable. *fom*( ) must return a large (guaranteed suboptimal) result for invalid inputs. It must also *slope the function toward valid values*, i.e. provide a "restoring force" to the invalid parameters toward the valid region (Figure 18). With multiple parameters $\mathbf{p} = \{p_i\}$, I used

$$guiding\_bad\_FOM = big\# + \sum_{i=1}^{N} |p_i - valid_i| \qquad where \quad valid_i \equiv a\ valid\ value\ for\ p_i$$

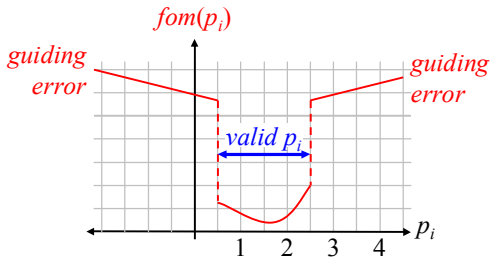


Figure 18: "Guiding errors" lead naturally to a valid solution.

Most multi-parameter optimization algorithms include a step which minimizes a function by varying only 1 parameter. For maximum stability, and reasonable performance, I use a binary search. This requires no continuity or other "niceness" of the FOM function. There are only 63 bits in the significand of long-double precision, so I limit the binary search to ~70 iterations. I clearly flag any failure to converge, to confirm that all my optimizations converge successfully.

The Fiducial Model PDF

The fiducial data taken by the APOLLO system during runs can be used to fit parameters for a fiducial detection time model PDF. Prior work of Jana Strasburg provided the starting point for my fiducial model PDF. Her model is this [Str p81]:

$$\text{pdf}_{APD}(t) = A\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right] + B\underbrace{\left[1-\exp\left(-\frac{t-(\mu+d)}{t_0}\right)\right]}_{\text{soft turn on}}\exp\left[-\sqrt{\frac{t-(\mu+d)}{t\_decay}}\right] + C$$

where $\mu \equiv$ gaussian core average

$\sigma \equiv$ gaussian core standard deviation

$d \equiv$ tail delay

$t_0 \equiv$ tail delay turn on constant

$t\_decay \equiv$ tail decay time constant

$A \equiv$ core amplitude

$B \equiv$ tail amplitude

$C \equiv$ flat background rate

Figure 19 shows schematically the main physical features of the model PDF.

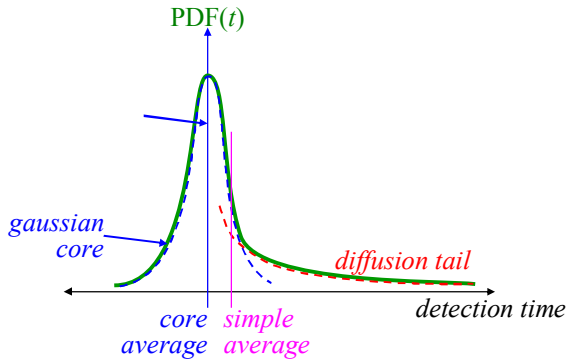


Figure 19: The detection time PDF includes a gaussian core and a long tail.

The detection model is dominated by the characteristics of the APD, but the parameters also include the effects of random errors in the timing electronics and TDC.

[Str] provided essential insight and verification of many aspects of the APD timing PDF. In particular, she showed that the PDF of the diffusion tail time follows very closely the model

$$diffusion(t) = N\exp\left(-\sqrt{\frac{t}{t\_decay}}\right)$$

However, the 2nd term in that model includes an ad-hoc "soft-turn on" factor (in brackets), which allows for a continuous transition from the gaussian core to the tail. This model had a small problem with lack-of-fit just after the gaussian core [Str p82, 84, 86, 89].

I found that the fit is better if the model is built directly from the physics. This also reduces

the number of parameters from 6 to 4. The tail of the distribution arises because some fraction of the electron-hole pairs are generated below the high-field depletion region, where they diffuse slowly. Some of these carriers enter the high-field region, where they are quickly swept to the avalanche region, and trigger a detection. These "late detections" create the long tail in the PDF. The time delay is simply the sum of the usual detection delays, plus the diffusion time. Thus, the detection time for "tail photons" is the sum of two random variables. The PDF for the sum of two random variables is the convolution of the PDFs of the two variables. Thus, the PDF of the tail photons is generated numerically by convolving the PDF for the slow diffusion, with the gaussian core PDF:

$$\mathrm{pdf}_{tail}(t) = gauss(t) * diffusion(t) = N \underbrace{\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right]}_{gaussian} * \underbrace{\exp\left[-\sqrt{\frac{t}{t\_decay}}\right]}_{diffusion}$$

where * ≡ convolution

N ≡ simple normalization factor

This automatically generates a smooth tail PDF by direct modeling of the physics. I do not write the tail PDF in analytic form, since it is constructed numerically by convolving the PDFs of the random variables that compose it. The final PDF, including both tail and core photons, is then given by the weighted sum of the PDF without diffusion and the PDF with diffusion, with the weights determined by the probability of a detection without a diffusion ($1 - t_{frac}$), and with (t_{frac}):

$$\mathrm{pdf}_{APD}(t) = \left(1 - t_{frac}\right)\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right] + t_{frac}\,\mathrm{pdf}_{tail}(t)$$

where μ ≡ gaussian core average

σ ≡ gaussian core standard deviation

t_frac ≡ fraction of detections in the tail (i.e., with diffusion)

t_decay ≡ tail decay time constant

The fit parameters are now μ, σ, t_frac, and t_decay. t_frac is the fraction of all detections that start below the depletion region, and diffuse a while before detection (essentially replaces the B parameter). μ, σ, and t_decay are as before. The d and t_0 parameters are not needed, since the physics models a soft turn-on automatically, implemented mathematically by the convolution. The previous

parameter *C* defines the flat background rate, and is negligible for FIDs. However, there is a need to account for the background detections in the LUNs.

Note that the fit to parameters is fully nonlinear, and we do not linearize it.

[Str p95] found that at 786 nm, the fit parameters for 30 μm detectors have t_decay = 1.67 ns; at 668 nm, t_decay = 1.27 ns. Due to the shallower penetration of photons at 532 nm, we expect t_decay to be even shorter, and indeed my fits yield t_decay ≈ 1 ns.

[Str p146] estimates t_frac (= P_{tail} / (P_{core} + P_{tail})) ~ 0.3 @ 668 nm. At 532 nm, due to higher absorption in silicon, we expect t_frac to be lower, and our data show that t_frac ≈ 0.2. However, [Str p155] estimates t_frac < 0.05 for 532 nm, at a bias of 3.5 - 4V over breakdown, but our actual results are not nearly that good. The reality is a bigger tail, which is consequently much harder to work with.

Figure 20 is a sample fit to real data, which shows very good fit, even in the "shoulder" where the prior model sometimes did not. The χ^2 = 1.22, reduced by the number of degrees of freedom. The improved fit just past the main peak may be due to the "shoulder" being washed out by the wider gaussian core in the real APOLLO system, compared to the laboratory setup previously used.

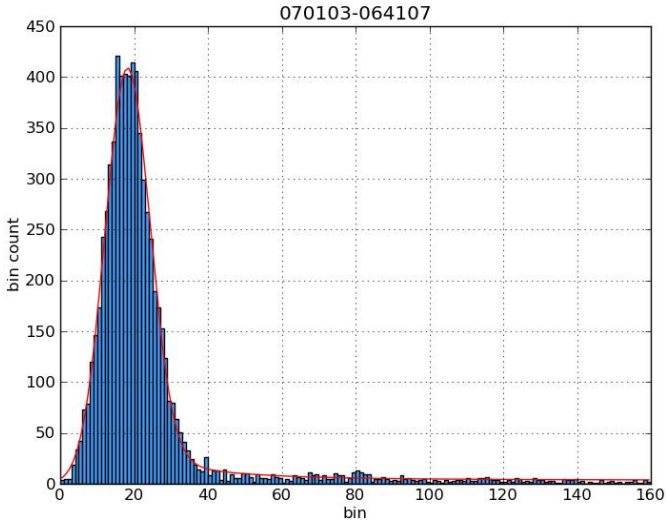


Figure 20: Sample fiducial histogram fit, 40 bins/ns. Reduced-χ^2 = 1.22. The fit is over bins 1 - 139, which includes bins between the most widely space pair with 5 or more counts.

Fiducial Fitting Issues

Generating the Model Histogram: To fit a model PDF to a histogram, we first create a model histogram (aka discrete PDF) from the model PDF, using trial values for the model parameters. In my case, I simply multiply the PDF at the bin-center by the bin-width. This is valid so long as the PDF changes slowly over a bin width.

When binning time measurements, proper rounding is *critical*. Failure to round the time to the *nearest* time-point introduces a ½-bin error = 12.5 ps ≈ 2 mm (large). Such failures were occasionally observed during development, and had to be corrected. The time-points of bins *always* refer to the bin center (Figure 21, left), a standard that helps meet the proper rounding requirement.

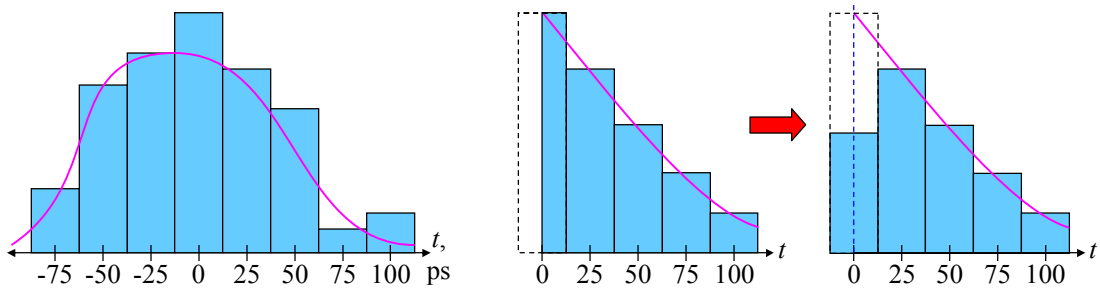


Figure 21: Bin centers define the time, but the $t = 0$ bin of the APD tail is special.

When simulating the APD tail, its discrete PDF (25 ps bins) requires a special treatment of the $t = 0$ bin, because the tail cuts off exactly at $t = 0$ (Figure 21, middle). Therefore, the number of counts in the bin is ½ of (25 ps) times $PDF_{tail}(0)$; in other words, the bin is effectively only 12.5 ps wide (Figure 21, right).

Impairments: Our fiducial data contains a number of impairments, some of which likely affect the model parameters. The first is background noise due to dark-counts (probably thermal electron-hole generation). The background is quite low for FIDs, and I ignored them for fitting to the model. (Background is important for LUNs, due to the frequently bright lunar surface.) Another impairment is "echoes" of the main peak in the histogram, at a smaller level, appearing later in time. These are likely due to reflections within the system, which occasionally are detected shortly after the main peak. One particularly prominent echo sometimes appears delayed by ~4 ns from the main peak.

Other echoes have been seen, and are also intermittent. The exact source of these is not known, but they do not appear in the lunar returns; it is possible that physical adjustment of the optics might make some of them come and go. Another impairment is that the laser is not completely stable. Its pulse characteristics vary, and our measure of its standard deviation is much wider than the manufacturer's claim. One laser adjustment (8/07) abruptly changed the core-σ from 160 ps to 120 ns. Also, for a few months in mid-2006, the laser had a severe problem of producing two strong pulses about 2 ns apart, rather than a single pulse. These runs cannot be analyzed by the methods considered in this dissertation, and are not used for any results herein. Another impairment is electrical noise. The laser produces a large EMI spike when it fires, and this is known to degrade our measurement σ by at least 20 ps. Shielding some of the laser electronics might reduce this impairment.

First photon bias: An impairment of all APD measurements is FPB. My simulations show that, in practice, this affects mostly the core-average (that is, the time measurement). The effect on other parameters appears to be offset by the tendency to overestimate the tail parameters. The net effect is that the fit parameters from fiducial data closely match the underlying PDF parameters. More on this later.

Degeneracy of the fit parameters: Overall, the model fit is good, but the individual fit parameters vary quite a bit from run to run, in large part due to significant degeneracy in the parameters. In particular, t_frac and t_decay are strongly correlated. Increasing either produces a more pronounced tail, but with indistinguishable χ^2. I have taken some measures to try to separate them (described below), but the result is somewhat arguable. However, in the end, the model fits the data with the chosen parameters, and that is what is most important. This degeneracy causes a large dependence of the t_frac and t_decay parameters on fitting method, but little change in reduced-χ^2. I give examples below.

Back porch: On some of the runs from 2007, there is a high-level trailing edge of the PDF after the main peak on some of the runs. This "back porch" completely obscures the tail of the APD detection time. On these runs, if unaccounted for in the model, the back porch in the data drives the

APD diffusion time parameter t_decay to very long times, and forces a low tail fraction, t_frac, to set the level. I tried adding an ad-hoc back-porch parameter, p_frac, which was a flat region (similar to a background) but appearing only *after* the main peak. (However, to avoid an unphysical step in the PDF, the back porch model included a 200 ps linear ramp "turn on," centered at the gaussian core average.) This parameter accommodates the anomalous flat region, but at the expense of t_frac and t_decay. In both the anomalous and "normal" runs, the tail fractions were unrealistic, even though the reduced-χ^2 is good. Therefore, for most runs, we *must not* include a p_frac fit parameter. The anomalous back porch appears in only a few runs, so I dropped it from the model. The physically-based, four-parameter model works the best for almost all runs.

The cause of the back porch for those runs remains uncertain, but recent wide-gate measurements suggest it might be crosstalk between APDs in the array.

Fitting criteria: There are several criteria one must choose to perform a fit. These are not fit parameters, but criteria describing *how* the fit is to be done. I use the term "criterion" to distinguish from a fit parameter. These criteria have a significant effect on the results. The first fit criterion is when to declare the fit complete. I use a fractional change in the figure of merit, usually reduced-χ^2: when the fractional change is below 'fit_toler' (~ .001), the fit is done. A second criterion is which bins to include in the fit. Bins with too few counts are mostly noise, and degrade the results. I fit from the first bin with at least 'fit_bound' (~5) counts, through the last bin with at least 'fit_bound' counts. A third possible criterion is for rebinning (though it turned out not to be used): one can combine small bin counts in adjacent bins to make wider bins with larger counts. Note that there is no statistical reason why the bins should be equal width. The criterion 'min_bin' specifies the minimum count a bin (combined or not) must have to include it. Bins with fewer counts are combined with the following bin, until 'min_bin' is satisfied. 'min_bin = 0' means that no rebinning takes place. Below, I describe the experiments used to choose these criteria.

Fiducial Fit Results

The main goals of characterizing our fiducial PDF is to provide the model used for

simulations and in the PDF-fit NP generation method. The core-average and core-σ are easily characterized, but t_frac and t_decay are much harder, in part due to their degeneracy, and partly due to their producing only a fraction of all the detections. I performed a number of fits in various ways to establish the stability and validity of the final fit parameters, as I now describe.

The "Anomalies in Data Reduction" appendix lists all the problematic data, the runs which were excluded, and why.

I fit the fiducials from all the runs of 2007 to the model. There were 10 or so clear outliers within a range of a few weeks. I removed those, leaving 191 runs, which produced the following fits [fidfit.bat]. In general, t_frac is clamped between 0 and 0.5, and t_decay is clamped between 0.05 and 4 ns. Figure 22 shows the results for each run of the t_frac and t_decay parameters.

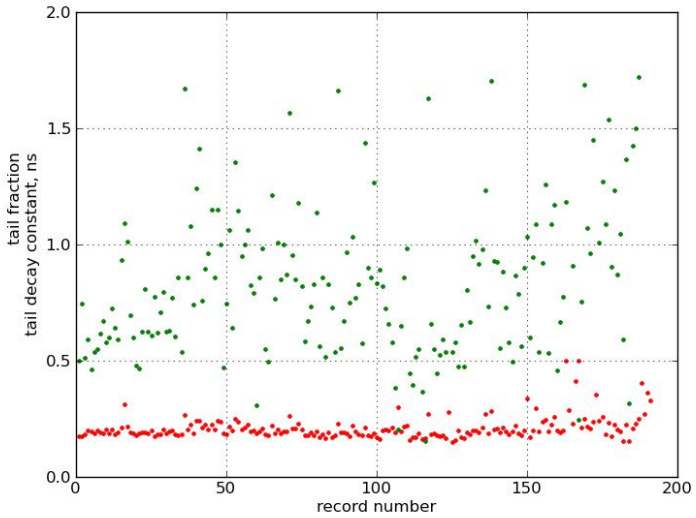


Figure 22: 2007 fiducial fits, with starting fit values of: 0., 0.16, .225, 1.
The red dots are tail fraction; the green dots are decay constant (ns)

Summarizing:

2007 Parameter	Avg	σ
t_frac	0.211 ± .004	0.051
t_decay, ns	0.92 ± .04	0.55
reduced-χ^2	1.39	0.36

Figure 23: Fiducial fits for 2007 data.

We see that all but a few of the t_frac and t_decay values are well within the allowed limits, and very few reach those limits. However, the σ on t_decay is quite large, and the plot shows a fairly wide distribution. Nonetheless, after many tests for validity, described below, I adopted values close to these as the standard values for simulation.

For these fits, I used the standard fit criteria of fit_toler = .001, fit_bound = 5, and min_bin = 0. I explored other values of these three criteria to find a reasonable set of choices that provide stable, reliable fit parameters. Using fit_toler = 10^{-4} yields slight increases in variation: the t_frac σ = .054, t_decay σ = .64, but χ^2 is no better, remaining at 1.39. Tightening further to 10^{-5} produces still larger t_decay σ = .69, and χ^2 remains unchanged at 1.39. Essentially, these tighter convergence criteria cause wander over the parameter space, finding insignificant improvements in χ^2. This increases the variation in the fit parameters, but to no benefit. With ν ~140, the standard deviation of χ^2 is ~12%; certainly fit_toler < .001 demands too much from the χ^2 parameter.

Setting fit_bound = 10 skews t_frac high, and t_decay low. It increases the t_frac σ to 0.085, and greatly changes t_decay to 0.77. This fit_bound is too high, and leaves off too much of the tail, though χ^2 remains good at 1.40. Changing the fit_bound from 5 to 2 improves χ^2 insignificantly, but radically changes the tail fit parameters. I believe this fit_bound is too low, and includes excess noise. The choice of fit_bound = 5 is based on fit simulations, described later.

To test for rebinning instead of histogram cut-off points, I set fit_bound = 0, and min_bin = 10, as recommended by [Bev p110]. The reduced χ^2 is significantly worse (1.84 vs. 1.39), and its variation is also much worse (0.76 vs. 0.36). This suggests that min_bin=10 is not a good choice. Since fit_bound = 5 and min_bin = 0 works reasonably well, I did not try other values for min_bin.

To test for sensitivity to starting values, I re-did the analyses but changed the starting fit values for t_frac and t_decay to be too small. This produced only tiny, insignificant changes in the parameter statistics, confirming that the fit is stable.

Since there is wide scatter in t_decay, lots of degeneracy between t_frac and t_decay, *and* the two are correlated with a nonlinear relationship, choosing the average value for both is likely to give a skewed result. Instead, since t_frac is more stable, I chose a typical value for t_frac (= 0.225), then re-fit for t_decay keeping t_frac fixed at this value. This yields a significantly tighter distribution, with t_decay = 1.16 ± .02, σ = 0.31 ns, though still disappointingly wide. Reduced-χ^2 is only slightly degraded, with χ^2 = 1.46, σ = 0.40.

For general interest, I also tried the reverse: fixing the t_decay, and fitting only for t_frac. The 2007 data with fixed t_decay = 1.0 ns gives tighter t_frac = 0.215, σ = 0.030, with reduced-χ^2 again only slightly degraded, at χ^2 = 1.44, σ = 0.39.

To test for consistency over time, I also fitted some 2008 data through March (nearly current at the time of the work). Removing two outliers leaves 78 runs (Figure 24).

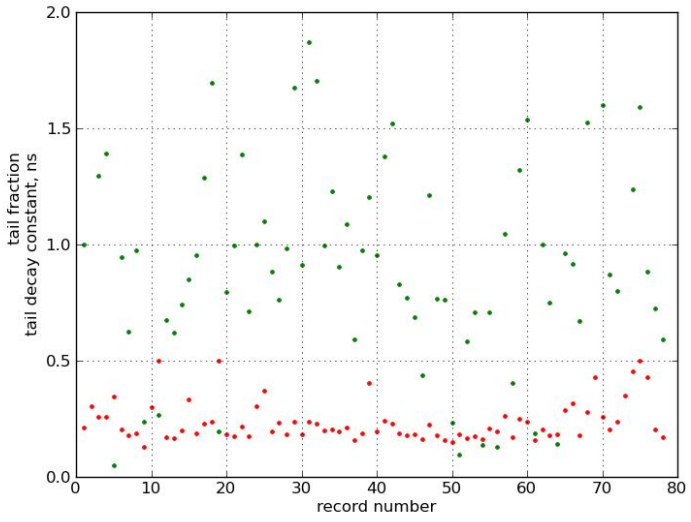


Figure 24: 2008 fiducial fits, t_frac (red) and t_decay (green).

Summarizing:

2008 Parameter	Avg	σ
t_frac	0.238 ± .010	0.086
t_decay, ns	1.00 ± .07	0.64
reduced-χ^2	1.30	0.34

Figure 25: Fiducial fits fo 2008 data.

These are quite consistent with the 2007 data.

Verification of FID Simulations

A number of upcoming tests rely on simulation. It is necessary to verify the validity of such simulations. The most direct test is to simulate runs with known PDF parameters, and then perform fits to them to see how well the recovered parameters match the known values.

Using t_frac = 0.212, and t_decay = 0.97 ns, I simulated 100 runs of 20,000 shots each. The simulations include a background of dark detections (e.g. thermal electron-hole pair generation), as in the real system. Then I fitted those simulations for the fiducial fit parameters, with a convergence criterion of 10^{-3} (Figure 26).

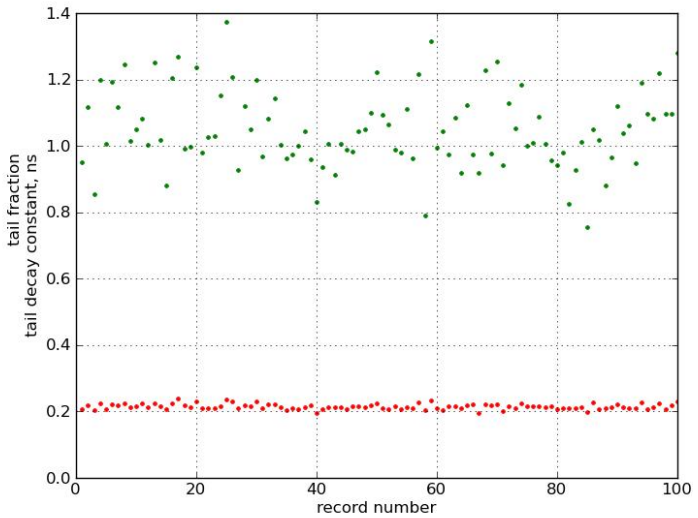


Figure 26: Fiducial fits to simulated runs: t_frac (red) and t_decay (green)

Summarizing:

Parameter	Simulated value	Fitted Avg	Fitted σ
t_frac	0.212	0.213 ± .001	0.008
t_decay, ns	0.97	1.02 ± .01	0.12
reduced-χ^2	1 (ideal)	0.98 ± .01	0.11

Figure 27: Fiducial fits to simulated runs.

The tail fraction is centered within the uncertainty of the simulated value, with σ ~5% of the average. The decay constant is also very close, high by ~5%, and σ ~10%. The χ^2 parameter is nearly perfect at 0.98, with σ ~10%. The fact that the simulations have tighter distributions than fits to the real system data, and better χ^2 values, indicates that the real system has small, but detectable, variations that are not modeled.

Effect of First Photon Bias on Fiducial Fit Parameters: The above simulations include the full effects of FPB on the parameters. Given that, it is perhaps surprising that the fit parameters agree so well with the simulated values. One must ask, what is the impact of FPB on the fit parameters? The simulation models the real system by first simulating the "underlying" time PDF, and then the detections are modified by triggering the APD on the first photon of any multiple-detectable photons in an APD. If anything, FPB should *decrease* the tail fraction, as some tail detections are masked by another, non-tail detection. As a test of the effects of FPB on the fiducial parameters, I simulated 100 runs with no FPB; the simulator insures no more than one photon per detector per shot. This eliminates ~4 ps of bias due to FPB, and ideally, should recover the simulated fiducial profile parameters. However, absent FPB, the tail parameters are overestimated: t_frac = 0.264 (25% higher than the simulated 0.212), and t_decay = 1.29 ns (33% higher than the simulated 0.97 ns). This is likely due to the small-count effect mentioned previously: the fit curve overestimates the low bin-count bins, at the expense of the high bin-count bins (supported later by minimum sum-squared error (MSSE) fits.) Coincidentally, this appears to be almost exactly offset by the FPB. In the end, these simulations show that the *net* effect of FPB within the given fitting procedure is to reasonably

determine the underlying tail parameters.

Convergence criterion, again: I experimented with a tolerance of 0.01, which produced even smaller variations. However, I initially discounted that tolerance because I had miscalculated the statistical significance of χ^2 from the run simulation. The corrected figure of ~12% (given earlier) shows that fit_toler = 0.01 may be reasonable, but needs further investigation. Therefore, the fit tolerance remained 0.001.

Rebinning test: Again testing for the effect of min_bin=10, I fit to simulations with t_frac=0.212 and t_decay = 0.97. The results are only slightly better than the min_bin=0 fits, with averages a little closer to the simulated values, and deviations ~ 30% smaller. Therefore, I kept the rebinning criterion that is best for the real-data, min_bin = 0.

Adoption of standard simulation parameters: At this point, I have adopted "standard" values of t_frac = .212, and t_decay = 0.97 ns, as the underlying simulation parameters. These are consistent with the fits taken from real data, with small adjustments made from simulation results. The exact values are somewhat subjective, depending on exactly which data runs one includes, and which are considered outliers.

Minimum Sum-Squared Error Fits: The use of reduced χ^2 assumes a somewhat ideal situation where the fit parameters' effects are spread uniformly over the graph. Our fit parameters are distinctly different from that assumption. The tail parameters affect almost exclusively the tail, and the core parameters affect almost exclusively the core peak. This causes reduced χ^2 to make the false trade-offs already mentioned, which result in overestimates of the tail parameters.

This understanding of why biases occur with χ^2 fitting prompts the question: would another figure-of-merit give less-biased fit results? An obvious possibility is a simple minimum sum-squared-error criterion (MSSE). Here, the fit residuals are not weighted, so there is no propensity to skew the curve high on low bin-count bins. With simulated tfrac=0.212 and tdecay=0.97 ns, MSSE fitting yielded low estimates for both t_frac (0.201) and t_decay (0.93 ns), consistent with the general trend for FPB-skewed results. The variation of t_frac is unchanged from χ^2 fitting, and the variation of

t_decay is slightly better. Figure 28 shows a sample fit to a simulated run. The tail is smooth and consistent with the simulation.

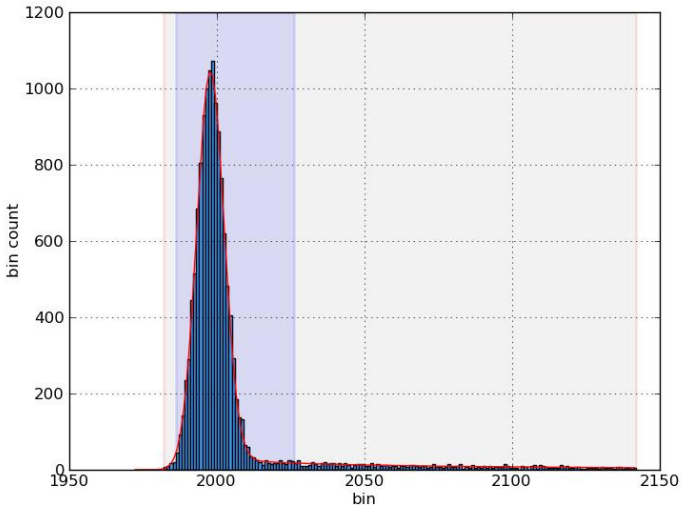


Figure 28: MSSE FID fit to a simulated run. Blue shading indicates the region actually fit. Gray shading is the maximum-count 160-bin window.

MSSE fitting to simulations with no FPB gives tail parameters very close to the simulated values: t_frac = 0.219, t_decay = 0.98 ns. This corroborates the explanation above that FPB offsets the bias in tail parameters from χ^2 fitting.

However, MSSE fitting to real data does not work. On year 2007 runs, t_decay is implausibly small at 0.27 ns. Year 2008 has a similar problem, t_decay = 0.37 ns, and has terrible uncertainty. A sample fiducial fit histogram shows some lack of fit on the trailing edge (Figure 29).

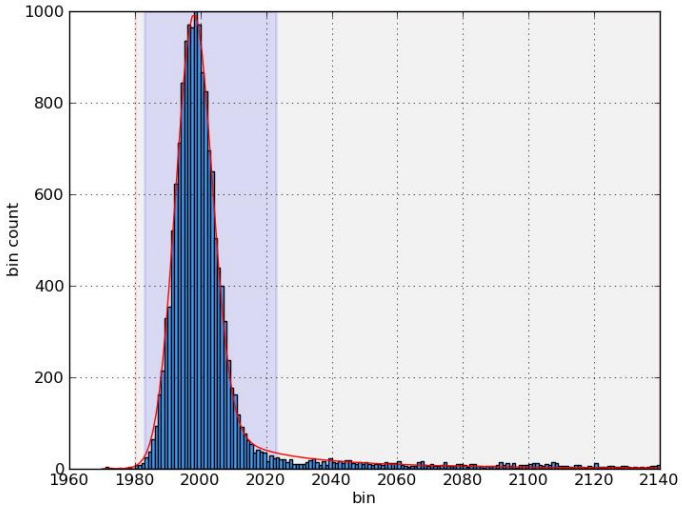


Figure 29: MSSE FID fit to a real run (070109-125848)

With MSSE fitting, the main peak is emphasized over the tail, since the tail errors are small. This causes the tail parameters to suffer. In the real run shown, the trailing edge seems to have either a deficiency of detections around bin 2030, or a small echo bump between 2030 and 2060. Compare to the χ^2 fit (Figure 30), which follows the trailing edge well, even through the trailing dip and bump in detections.

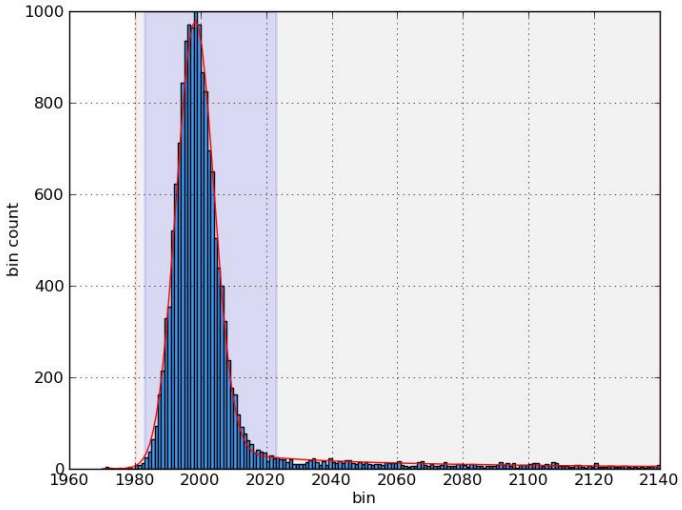


Figure 30: χ^2 FID fit to the same real run (070109-125848)

Ultimately, our goal is to analyze real data, so we use the χ^2 fitting criterion, which gives the best results on real data.

Fiducial Fit Position Stability

Up to this point, we've focused on the tail fit parameters, t_frac and t_decay. However, for the PDF-fit method, the ultimate measurement is round-trip-time (RTT), which comes from the core-average parameter. So we must ask: how stable is the fitted core-average? First, we examine the fit_bound parameter, which determines the left and right edges of the histogram to include in the fit. How does core-average vary with choice of fit_bound? To get started, I chose a run at random, and fitted it repeatedly with each value of fit_bound from 1 to 30 (Figure 31) [fitbound.bat].

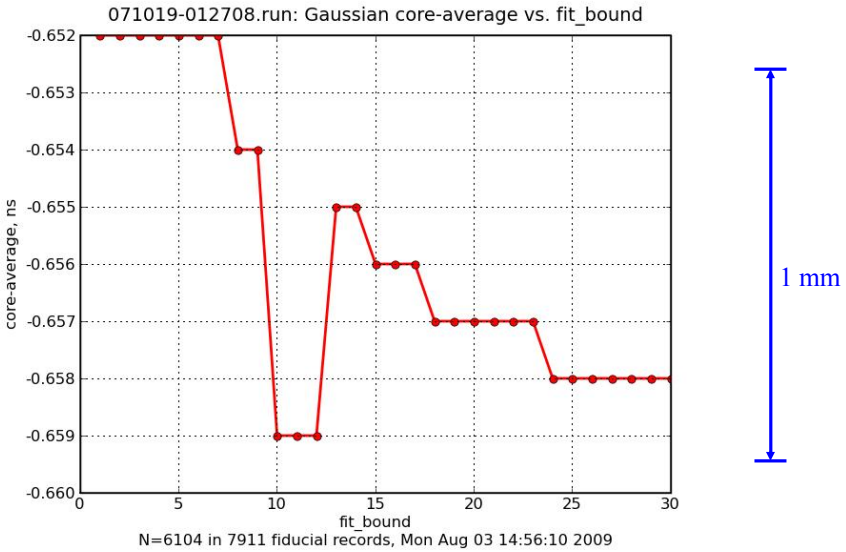


Figure 31: 071019-012708: Fit parameter core-average vs. fit_bound

Above fit_bound = 7, the time measurements are somewhat unstable: there is no clear choice. But fit_bound = 5 is perhaps in a flat zone. Another run from the same night, with far fewer FIDs, looks quite different (Figure 32, note the different vertical scale).

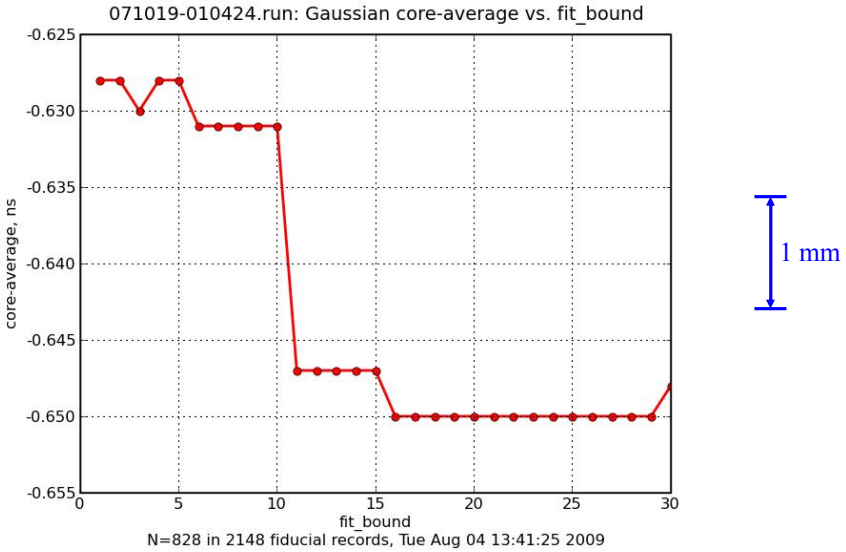


Figure 32: 071019-010424: Fit parameter core-average vs. fit_bound

Here again, though, fit_bound = 5 appears to be in a stable region. These results suggest the

core-average for larger fit_bound is sometimes unstable, and heavily dependent on the choice of 'fit_bound'. Such fits cannot be trusted, because we have no criterion for choosing a single value of 'fit_bound' from the upper range of choices.

To try for a better fit criterion, I performed another set of fits using bin combining instead of cropping: adjacent low-count bins were combined, so that each data point in the fit contained at least 'min_bin' counts. (This may truncate a small amount of data at the end of the tail where there were insufficient counts to meet the min_bin requirement.) The hope was that this would provide stable parameters across a reasonable range of 'min_bin' choices. Note that as with all histogram fitting, the data must be re-binned based on the data counts themselves, with no reference to the variational fitting function as it evolves over parameter space. If the re-binning depended on the variational fit function, there would be (again, as with truncation) discontinuities in the figure-of-merit that would lead to multiple local minima and unstable fit parameters. The fits to the runs shown previously are shown in Figure 33 and Figure 34 [min_bin.bat].

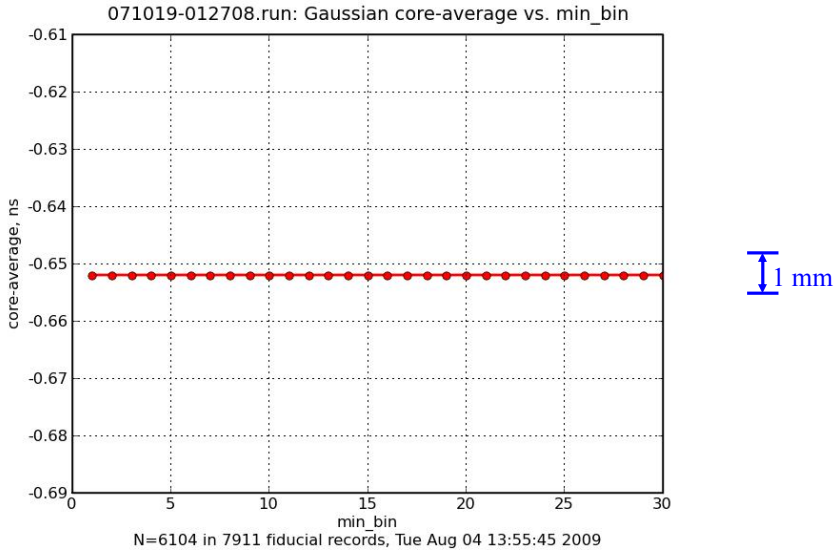


Figure 33: 071019-012708: core-average vs. min_bin

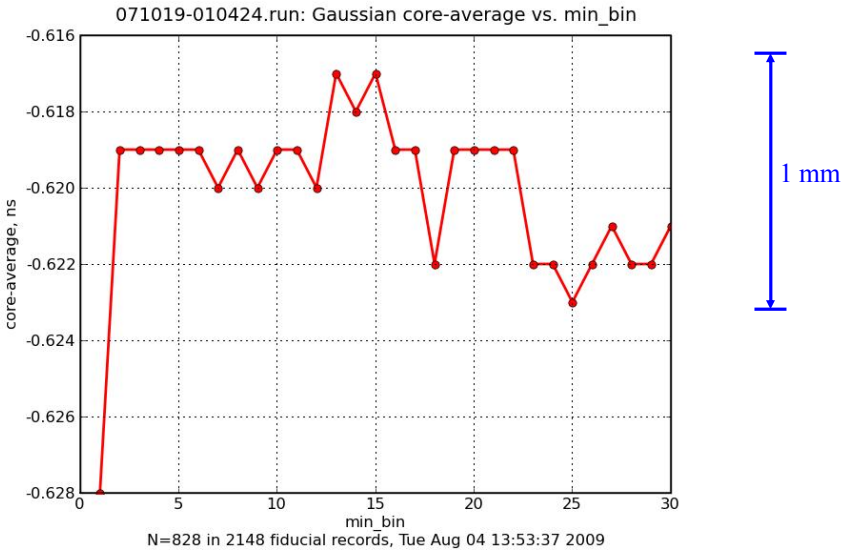


Figure 34: 071019-010424: core-average vs. min_bin

Figure 33 shows perfect consistency: every point is identical, to 1 ps. Even though the second run (Figure 34) has low fiducial detections, its graph is reasonably stable. For all min_bin > 1, the range of positions is 6 ps, and the standard deviation across all 30 min_bin values is 2.1 ps.

A concern with both fit_bound and min_bin as fitting criteria is that they are absolute numbers; runs with more returns, either from stronger signal or more shots, may have different fractions of the histogram tails included in the fit, even if they are otherwise identical to runs with fewer returns. In all cases, the main peak of the histogram will be included, and should dominate the result.

In the end, I chose to stick with fit_bound = 5, and min_bin = 0 (i.e., no rebinning), since the N = 828 run is about the fewest FIDs we'll ever use, and the results with these criteria appear stable below 1 mm, and so are consistent with the 1 mm overall goal. And as mentioned earlier, using min_bin = 10 causes problems with the tail parameters.

One could consider other determinations of the distribution position, perhaps averaging over some bins. This starts to look like the augmented calculation method. However, I have not investigated any other measures of position.

2.12 General Simulation Issues

The upcoming evaluation of NP generation methods relies heavily on simulations. To improve the validity of simulations and their analysis, there are several issues that must be addressed.

Simulation Calibration and Verification: I used calibration runs with specific characteristics to verify the implementations of both the simulator and the data reduction.

The first calibration is to simulate a simple gaussian distribution of FIDs and LUNs (no tails, noise, reflector tilt, etc.). To simulate such a distribution, I set all the impairments to zero, and artificially set the laser 1σ pulse-width to 120ps, which is our actual fiducial core-gaussian width. This results in a simple gaussian distribution, with only the simulated TDC quantization noise added (which is unavoidable with our data file format). I generated 100 runs, and analyzed them. The results were as expected (more details later).

Changing a parameter: Changing any simulation parameter usually changes the whole simulation set because the random number generator gets called a different number of times. In other words, you usually cannot precisely "add" or "remove" an effect from a given simulated run: changing anything effectively resamples the population of potential runs. Therefore, when comparing the effects of different simulation parameters on, say, the NP, the NP uncertainty must be small compared to the difference in NP times. Otherwise, the comparison is more about sampling noise than any real effect. Typically, I simulate 100 runs for each set of parameters, and compare the average over the set of runs. This reduces the analysis result σ by a factor of 10, which is usually sufficient.

Systematic round-off error: It is frequently useful to isolate a single impairment (e.g., the background detections) separately from all others (APD tails, retro-reflector tilt, etc.). This allows us to understand cause-and-effect for some kinds of impairments. Sometimes, we isolate an impairment by simulating just the one impairment in question, and set all others to zero. When doing so, there are many subtleties that can corrupt the results. For example, to verify that the normal point analysis code is correct, we simulate many runs with a purely gaussian distribution of RTTs, and no other variations. TDC time quantization to 25 ps then becomes very important. If the simulation TDC bins are

centered at 1 s boundaries, but the PDF-fit method uses a fit with bins with *left-edges* at 1 s boundaries, then the PDF-fit will consistently round differently by ½ bin than the simulation, resulting in a 12.5 ps bias in the result. This phenomenon was, in fact, observed, and had to be corrected. Real data would not suffer from either rounding choice in analysis. But if the simulation and analysis taken together have such a systematic error, our ability to understand our results *does* suffer.

Intentionally unrealistic conditions: The analysis code must also be robust against some unlikely idealisms, such as the retroreflector tilt being exactly 0 (orthogonal to the incident beam), which makes the "trapezoid" into a δ-function. Here again, simulations often use this idealization to help understand phenomena, and the reduction code must behave reasonably in these conditions.

Also, when analyzing simulated gaussian photon times, with no tail, the analyzer needs a sanity bound of t_decay ≥ ~0.05 ns, to avoid massive round off error in the model with 25 ps bins. The analyzer fit then cleanly drives t_frac to zero to fit the data.

Random Number Generator: The first random number generator was ported from Numerical Recipes, 2nd ed. It was a 32-bit generator. I ran many tests confirming the port is accurate, and the pseudo-random numbers are generated with the right distributions. However, a single simulation set of 100 runs of 20k shots each requires 400 M random numbers. This is 10% of the 32-bit (4×10^9) period of the pseudo-random generator. When using close to the period of a generator, undesirable correlations and uniformities start to appear. In reality, since I use the random numbers for a wide variety of purposes, these uniformities, at 10% of the period, probably would have no effect. However, a simple test bins 10^9 random numbers into 10^6 bins, for an average of 1000 hits per bin. The statistics of the bin counts should be very closely gaussian, with predictable deviation of $\sqrt{1000} = 31.6$. I tested the 32-bit generator and found $\sigma = 27.7$. This clearly shows a regularity in the numbers, which caused the variation in the bin counts to be less than theoretical. A test with $\sim 4 \times 10^9$ numbers would have covered the period of the generator, and driven the bin-count variation to nearly zero.

Numerical Recipes, 3rd ed., has a newer, 64-bit generator. (These are surprisingly hard to find in public source code.) I ran the same test as above, and the 64-bit generator produced $\sigma = 31.6$,

exactly the theoretical value. The runtime cost of the 64-bit generator *alone* compared to the 32-bit is < 20%, which is immeasurably small compared to other computations involved in simulation.

Other idealizations: We are not currently simulating any timing skew between detector channels. The analysis program, however, does not rely on this, and performs full skew alignment on every channel, the same way for simulations as for real data. The simulator also simulates each APD/TDC channel with identical time PDFs, though we know there is some variation in both detection efficiency and timing. I believe these omissions have no significant effect on the results. If there is concern, these might be areas for further study.

Choice of return rates: It is most helpful to choose realistic simulated return rates. The bulk of our return rates span roughly 2 orders of magnitude, from ~0.01 - 1 det/shot. The low rates have high uncertainty, so they contribute much less to the final solar system model fit parameters. Historical return rates covering 2006 through October 2008 have the distribution shown in Figure 35 and Figure 36 [return_rate.py]. Figure 36 shows more detail at the lower return rates (the highest bin includes all runs > 0.99.) A very few runs make 1 photon/shot. About 80% are 0.1 photon/shot or less. Choosing "representative" return rates is subjective. Considering that higher rates are weighted more heavily, I chose rates of 0.05, 0.5 and 1.0 photon/shot.

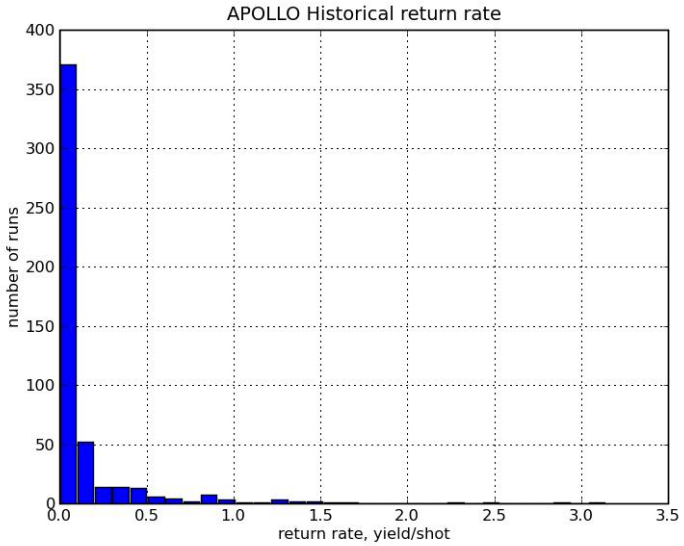


Figure 35: Historical lunar return rates

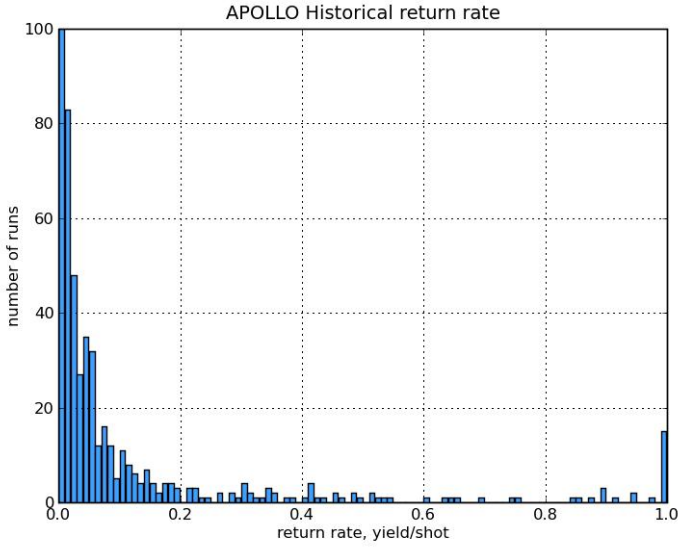


Figure 36: Detail of lower lunar return rates

Analyzer objectivity: In general, my analyzer does not "know" about the existence of a simulator, and does not distinguish between simulated or real data in any significant way. The one option the analyzer has, purely for convenience, is that operators can set it to use a fixed 25 ps/bin TDC calibration, which matches the simulator. One could construct TDC calibration files that do the

same thing, but this is tedious, and serves no purpose.

2.13 Augmented Calculation Analysis and Simulation

The algorithm for augmented calculation (AC) has already been described. We now consider that algorithm's performance, including some analysis and simulations. In general, having analytic formulae, even approximate ones, for various effects is superior to simulations alone, because it allows understanding where impairments come from, and which parameters are important to them. This often guides future improvements in directions that are most productive. Where feasible, I provide analysis of important effects. Simulations confirm the analysis, and may provide more accuracy for some approximate analyses.

The simulations and analysis were done with my own code, written in standard, portable C and C++. The execution speed of this code was essential to completing the research. It should be readily portable to Unix or other platforms. Though our final data reduction code will likely be in Python, it was not feasible to do the work using it. Even the plots, which were done with Python, were occasionally a bottleneck in completing a study.

All data reduction methods compute an NP by comparing in some way the FIDs for a run to the LUNs. Because FIDs and LUNs have different statistical distributions, and the LUN distribution in particular depends on observing geometry, a big concern for any data reduction method is any systematic error in estimating the time position of the LUNs compared to the estimate of the time position of the FIDs. In the AC method, there are four impairments that might offset the lunar returns differently than the fiducials:

- tail truncation may be different
- background level is different
- first photon bias is different due to different and more variable light intensity
- first photon bias may be different due to the trapezoidal PDF of retroreflector delay

What Order to Fit?

The main operation in the AC method is to fit a polynomial to data, either directly to RTTs, or to differences from the prediction.

I first attempted to fit directly to RTTs, in hopes of avoiding any biasing of our results based on our prior expectations, i.e. the RTT prediction from the JPL ephemeris APOLLO uses. While this method produces valid NPs, it leads to significantly increased uncertainties, as I now describe. In all, there are three polynomials involved:

1. the prediction polynomial, made before the data are taken from a JPL ephemeris
2. the "first" polynomial, fit to measured data chosen with the help of the prediction.
3. the "second" polynomial, fit to measured data chosen with the help of the first polynomial.

In general, we must use the JPL-based prediction to some extent, because our equipment can only measure detections within a 100 ns window, and we must know the approximate trajectory of the RTTs over the duration of the run to find the signal photons. The AC method uses a wide (~4 ns) "first-window" to select the detections deemed "mostly valid." The first window locates the signal by sliding across a histogram of differences from the original prediction. The window of bins with the maximum number of detections is deemed to contain the signal. The method then fits a first polynomial to those points. The purpose of the first fit is specifically to decouple the final result from the original prediction. Differences in RTT from this first polynomial, and a smaller window, are then used to select points for the 2^{nd}, hopefully better, fit.

The original AC method fit the first polynomial directly to the RTTs chosen by the first window. Because of the complexities of the motion of the earth and moon, the RTT curve as a function of time over the duration of a run requires a 4^{th} to 8^{th} order polynomial to fit well. However, these order polynomials allow unphysical "wiggles" in the curve that follow random background noise. For example, Figure 37 shows a 4^{th} order fit to an actual run [fitorder.bat]. The fit is directly to RTTs, but we commonly *plot* the differences from the prediction, to keep the vertical scale small enough to

see details.

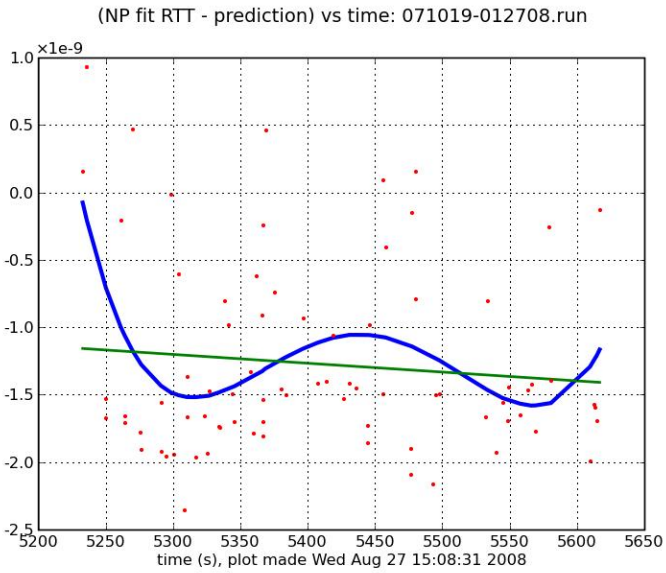


Figure 37: Example of 4th order fit (blue) to small yield run. The wiggles are unphysical. The red dots are individual RTTs. The green line is a linear fit to prediction differences.

Reducing the fit order, in hopes of reducing the wiggles, doesn't work. A 2nd order fit to the same data shows similar problems (Figure 38). Note that even in the 2nd order fit, the *prediction difference* curve is clearly at least 3rd order, because it is the difference between the 2nd order fit and the 8th order prediction polynomial.

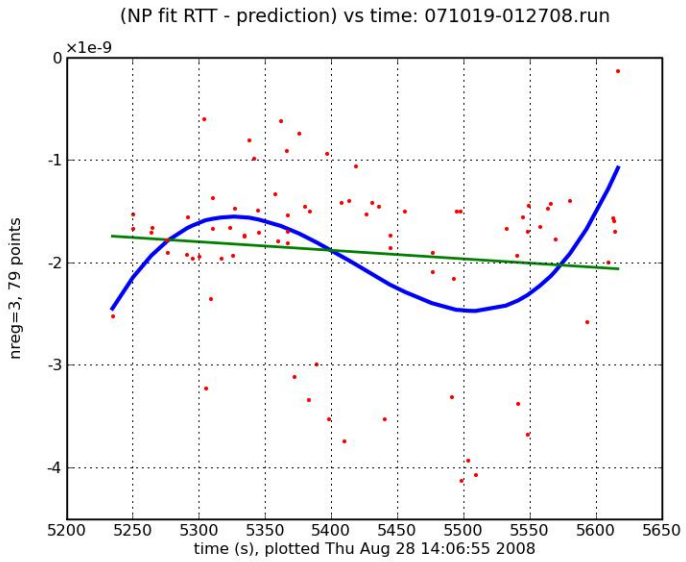


Figure 38: Example of 2nd order fit (blue) to the same small yield run. The wiggles are unphysical. The red dots are individual RTTs. The green line is a linear fit to prediction differences.

Also, the linear fit to prediction difference (green line) is different in the above two cases because the first polynomial selects which points to include. The 4th order polynomial selects different points than the 2nd order.

To quantify this effect, I ran 100 simulations of 4th order vs. straight line fits. These are worst-case simulations, with brightest background and large retroreflector tilt, but lots of data points. Again, FPB is omitted, to focus on the issue of fit order. The first half of Figure 39 plots the NP difference from prediction for the 100 runs fit to a straight line; the second half plots the *same* 100 runs fit to a 4th order polynomial.

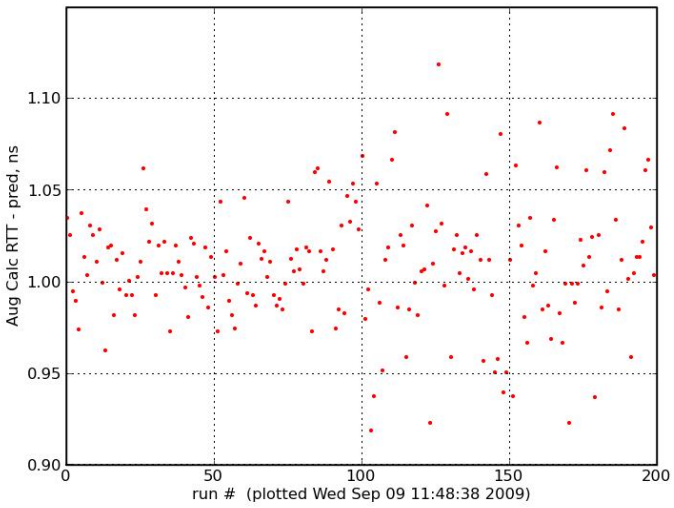


Figure 39: Comparison of straight line fit (runs 1-100) to 4th order fit (runs 101-200).

Visually, the uncertainty is clearly smaller with straight line fits. The computed standard deviation of the normal points from the 100 straight line fits is 7.5 ps; from the 100 4th order fits is 11.2 ps.

We can estimate the difference in uncertainty with the following simple argument: a 4th order polynomial has up to 4 zero crossings, which divides the time axis into 5 intervals. If we divide our data into 5 subsets, each with ~1/5 the data points, the uncertainty in the average of each group increases by ~√5. The 4th order polynomial has the freedom to follow these subgroup average variations, as seen in the earlier plots above. In contrast, a straight line has 1 zero crossing, dividing the data into 2 groups. Each group's uncertainty in the average increases by ~√2, and the line can only follow those two fluctuations. Roughly then, we expect the 4th order polynomial to be worse by a factor √(5/2) = 1.6. The actual ratio of the uncertainties in the above simulations is 11.2 / 7.5 = 1.5, which agrees very well with the computed ratio.

Without using any prediction, even 2nd order fits have measurably more variation. Using the data alone for the first fit "wastes" our prior knowledge, and degrades our uncertainty somewhat. The net result is that while higher order effects of motion are very significant, they are known well from

the prediction polynomial, which is fit to a high accuracy JPL ephemeris. Therefore, I chose a linear fit to the (RTT – prediction) difference.

Note, though, that these are small yield simulations, with NP uncertainties much larger than the 3.7 ps difference in NP variation. It may be that this effect is ultimately negligible, and at worst, it is small (~0.5 mm). As an option, my code retains the ability to perform fits directly to the RTTs.

Possible future enhancements: Since the beginnings of real runs are often devoid of signal while the operator searches for the reflector, the early detections are often mostly noise. When fitting a line, early detections carry a lot of weight (their "lever arm" is large). We could consider fitting a 2nd order polynomial to the prediction differences, which weakens the strength of the early detections (the lever arm is floppy). Another possibility is to use a statistical significance test on the early detections to find where the signal was likely acquired, and ignore detections before then. This would eliminate the early detections that are mostly noise.

Tail Truncation

We present here some statistics of the tail model that we will soon need. The tail adds some time delay to the average photon detection. As described earlier, our model tail time PDF is this:

$$\mathrm{PDF}_{tail}(t) = \frac{1}{2\tau}\exp\left(-\sqrt{t/\tau}\right) \qquad \textit{where} \quad 1/2\tau \text{ is the normalization}$$

The average and standard deviation are (derivations in the Appendix):

$$\langle t \rangle = \int_0^\infty t\ \mathrm{PDF}_{tail}(t)\ dt = 6\tau, \qquad Var(t) = \langle t^2 \rangle - \langle t \rangle^2 = 84\tau^2, \qquad Dev(t) = \sqrt{Var(t)} = 9.165\tau\ .$$

The $\mathrm{CDF}_{\mathrm{tail}}$ exists in closed form:

$$\mathrm{CDF}_{tail}(T) = \frac{1}{2\tau}\int_0^T dt\ \exp\left(-(t/\tau)^{1/2}\right) = 1 - \left(\sqrt{T/\tau}+1\right)\exp\left(-\sqrt{T/\tau}\right),$$

which is useful for numerically generating simulated detection times. I verified all of the above numerically with simulations.

τ is a simple scaling of the time-variable t, so $\langle t \rangle$ and Dev(t) are proportional to τ. In APOLLO, $\tau \approx 1$ ns, so the deviation of the whole, infinitely long tail is ~ 9 ns, compared to our 1 mm

goal of 0.007 ns. Even though only a fraction (~ 0.2) of detections are from the tail, the AC method clearly must truncate the tail somehow before averaging. At most, we are limited by the 100 ns TDC window, but in practice, to avoid background photons, we truncate to a much smaller window, of a few nanoseconds.

If we could include the infinite tails in both FIDs and LUNs, the two tails would exactly cancel, regardless of retroreflector spreading. This is because the tail delay is an additive random variable, and so is the retroreflector delay. The average of the sum of two random variables is the sum of the averages. Explicitly, let:

F be a random variable for fiducial core-photon detection time, and

T is a random variable for the average tail delay time,
L is a random variable for lunar core-photon detection time
RR is a random variable for reflector delay
Then the average measurement is the sum of the average contributions

<FID> = <F> + <T> and <LUN> = <L> + <RR> + <T>

When subtracting <FID> from <LUN>, the tail contributions cancel exactly. This is true for *any* underlying PDF of fiducials or retroreflector tilted lunar returns.

However, all measurements must be windowed (truncated) to some finite range. Without retroreflector tilt, the FIDs and LUNs could be truncated identically, and no systematic error would result. However, with reflector tilt, the tail of the LUNs is also spread by the reflector. There is no well-defined tail truncation length. Reflector tilt together with data truncation might then introduce a systematic error from skewing the lunar detection times differently than the fiducial times. We can estimate the systematic error for a given window by starting with the average of a truncated distribution:

Defiune $w \equiv$ window width of truncated tail, in units of decay constant τ

Then $$avg_{trunc}(w) = \frac{\int_0^w t\ \mathrm{PDF}_{tail}(t)\ dt}{\int_0^w \mathrm{PDF}_{tail}(t)\ dt} = \frac{\int_0^w t\ \mathrm{PDF}_{tail}(t)\ dt}{\mathrm{CDF}_{tail}(t)} \quad where \quad t \text{ is in units of } \tau$$

Therefore $$\lim_{w\to\infty} a(w) = \langle tail\text{-}delay \rangle = 6$$

I made numerical tests, using 100 points per τ, to compare truncation windows of $t / \tau = 25$, 100, and ∞ (no truncation). I compared the resulting average and deviations of the detection times, as shown below [apsimlib.c]:

	Truncated to $t/\tau = 25$	**Truncated to $t/\tau = 100$**	**Model, $t/\tau \rightarrow \infty$ (no truncation)**
$<t / \tau>$	4.60	5.94	6 (exact)
Dev(t / τ)	5.19	8.76	9.165

Figure 40: Detection time statistics for truncated APD PDF tails.

In our system, these averages are downweighted by t_frac ~ 0.2. Even so, we see that keeping the tail out to 25τ ≈ 25 ns, which is a very long time, still causes significant shifts in the average detection time (~1.4 ns), about 200 times larger than our accuracy goal of .007 ns. Also, such big windows lead to large uncertainties: the tail photons have huge standard deviations.

Without reflector tilt or background, we could choose a single, small window for both FID and LUN: the tail truncation would be the same for both, and would cancel. Without background, but with reflector tilt, we could in principle choose a single, large window for both FID and LUN, and the difference in tail truncation from the reflector-spread lunar return could be made negligible. But with tilt and a background, we must examine the window more carefully, to avoid systematic truncation error, and to allow reasonable background immunity.

Since the core PDF has some width, a fixed window on the data truncates the leading edge of the core distribution at a larger tail time than the trailing edge (Figure 41).

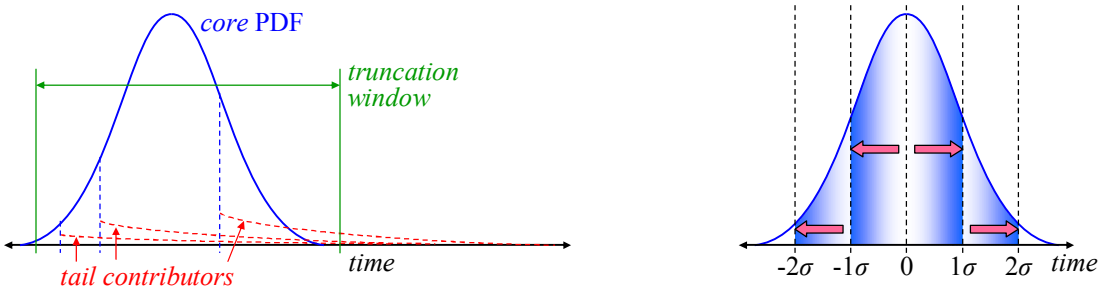


Figure 41: (Left) Varying tail truncation due to core distribution.
(Right) Upper bound skew determined from discrete sum.

When truncating the data within a time window, different core PDFs (different between FIDs and LUNs due to reflector tilt) result in different fractions of the tail distribution being included in the average. This difference induces a shift in the computed average round-trip time. Is this shift significant? The shift is proportional to the tail-fraction (fraction of detections which have a tail delay), and the average tail delay, and therefore the decay time-constant τ. The shift also depends on the core PDF.

The tails originating from the leading edge of the core-PDF are truncated at longer times than the tails originating from the trailing edge of the core-PDF. The core-PDF is symmetric (it is the tails themselves that make the *final* PDF asymmetric). The effects of longer truncation from the leading edge, and shorter truncation from the trailing edge, partially cancel, and it is only the difference from exact cancellation that systematically skews the FID from the LUN positions. Such a skew is a systematic error in the final NP. We can quantify the skew as follows:

$$NPskew = f_{tail}\tau\left[\underbrace{\int avg_{trunc}(t/\tau)\,\text{core-pdf}_{LUN}(t/\tau)\,dt}_{LUN\ offset} - \underbrace{\int avg_{trunc}(t/\tau)\,\text{core-pdf}_{FID}(t/\tau)\,dt}_{FID\ offset}\right]$$

$$where \quad avg_{trunc}(t/\tau) \equiv \text{average detection time of just the tail}$$

$$\text{over the interval 0 to } t/\tau \text{ time constants} \quad = \int_0^t (t')\,PDF_{tail}(t')\,dt'$$

We can make an upper-bound numerical estimate of the LUN offset, regardless of reflector tilt, by approximating the integral as a sum of two terms. The two terms put 2/3 of the photons at the $\pm1\sigma$ times, and the rest at $\pm2\sigma$, for the largest realistic σ (Figure 41, right). The largest LUN $\sigma \sim 0.3$ ns, and an upper bound on the window width is 2 ns (± 1 ns from center), which we approximate here as symmetrically placed. The truncated tail width for the core center is 1 ns. The truncated tail width for -1σ is 1.3 ns, and for $+1\sigma$ is then 0.7 ns. The skew for the $\pm1\sigma$ photons is just the average of the two tail averages truncated for -1σ and for $+1\sigma$, minus the truncated average for the core center. Similarly for the $\pm2\sigma$ photons. All together, we find the upper bound LUN offset:

$$f_{tail} = 0.21, \quad \tau = 0.97\ ns, \quad \sigma \approx 0.3\ ns$$

$$LUN\ offset \approx (0.21)0.97\left[\frac{1}{3}\underbrace{avg_{trunc}(1.3)}_{-1\sigma} + \frac{1}{3}\underbrace{avg_{trunc}(0.7)}_{+1\sigma} + \frac{1}{6}\underbrace{avg_{trunc}(1.6)}_{-2\sigma} + \frac{1}{6}\underbrace{avg_{trunc}(0.4)}_{+2\sigma} - \underbrace{avg_{trunc}(1)}_{mid-window}\right]$$

$$= (0.21)0.97\left[\frac{1}{3}0.549 + \frac{1}{3}0.312 + \frac{1}{6}0.662 + \frac{1}{6}0.185 - .433\right] = -1\ ps$$

The value is slightly negative, as we expect from the following: tails originating from detections after mid-window are truncated more than those partially compensating from before mid-window. But the tails decrease with time, so the loss of detections from tails originating after mid-window is larger than the gain from the longer tails originating before mid-window, but actually being detected after mid-window.

A similar calculation for a same-sized FID window (but no spreading from the retroreflector tilt) yields:

$$f_{tail} = 0.21, \quad \tau = 0.97\ ns, \quad \sigma \approx 0.12\ ns \qquad \rightarrow \qquad FID\ offset \approx -0.2\ ps$$

The upper bound to the LUN offset is already negligible, and is partly cancelled by the FID offset. This shows that, when the windows are reasonably centered, the *spread* in the LUN distribution on the tail, induced by reflector tilt, introduces no noticeable RTT bias.

A separate issue is the matching of the FID and LUN window *widths*, discussed later.

Note that quantifying the tail probability distribution function, including the model parameters, is important for assessing the RTT skew it introduces.

Window Sizes

An important consideration for a "window-and-average" algorithm is "What size window to use?" Clearly, if one uses a very small window, the choice of window placement will dominate the average computed within the window, and also one will get an unrealistically small uncertainty within that window. On the other hand, the uniform background of detections has unbounded standard deviation with increasing window-size. Hence, too wide a window yields an NP with unrealistically large uncertainty (and other problems). To have maximum confidence in the final uncertainty claim,

there should be a region of window sizes in which the uncertainty is fairly constant, and hence reflects the actual signal. Figure 42 sets a lower bound for window size: it plots uncertainty vs. window size for a best-case high-rate (~0.5 detections/shot), no background, simulation [lunwindow.bat]. The blue line is the Apollo 15 reflector with no tilt; the red line has maximum tilt. The uncertainty increases with window width due to the ever-present tail at all reasonable window sizes.

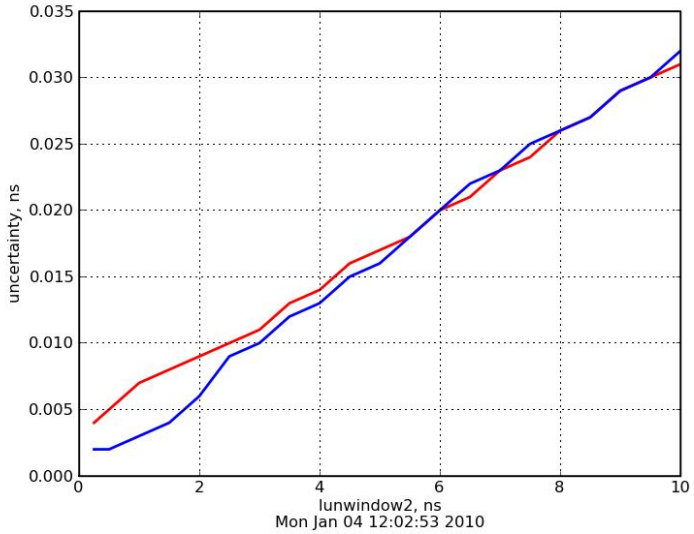


Figure 42: Uncertainty vs. window size. (Blue) No tilt. (Red) Maximum tilt.

For no tilt (blue), the expected optimum window is ~0.5 ns, and possibly there is some flat region between 0.25 ns and 0.5 ns, but it is not clear. For maximum tilt, there is no clear flat spot. Real runs, with background noise, also show no clear flat spot (Figure 43).

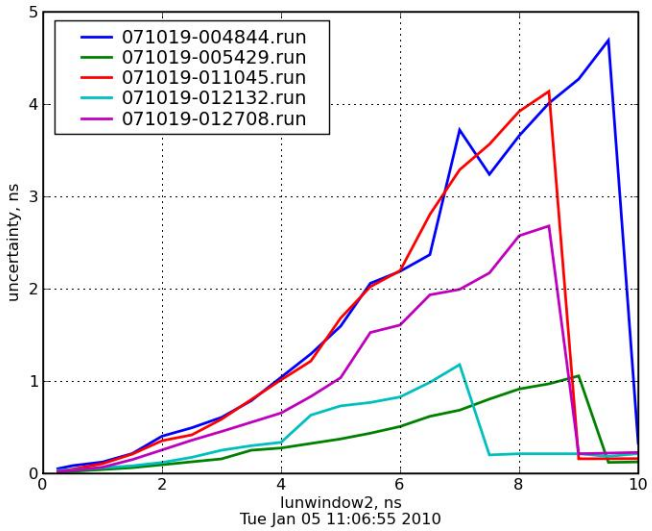


Figure 43: Uncertainty vs. window size for real data.

However, analysis of many simulations shows that the RMS residual from the final fits, with windows derived from detection deviation and reflector tilt, is reasonably consistent with the actual variation in NPs across the simulated runs. Therefore, we accept RMS residual (divided by $\sqrt{N}$) as the estimate of uncertainty.

Effect of reflector tilt: The varying reflector tilts from day to day induce significant differences in the spread of lunar returns. For a maximum tilt A15 simulation, Figure 44 shows the return time spread [lunwindow.bat].

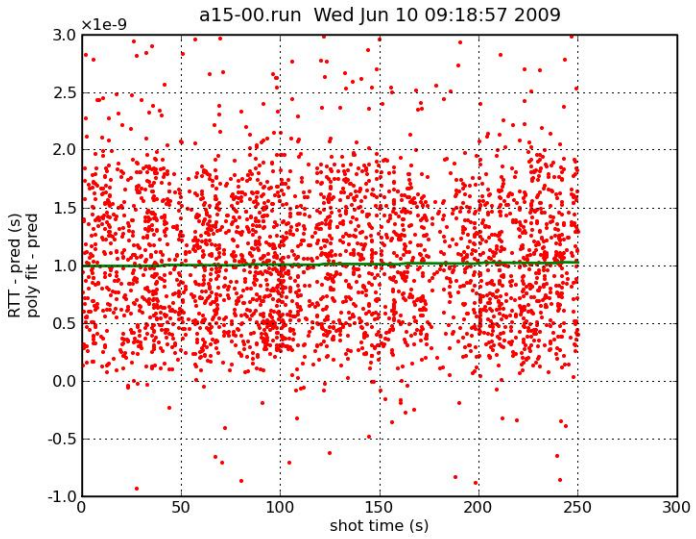


Figure 44: Simulated detections from maximum tilt A15 reflector.

To capture the valid signal here, and also to have a reasonable measure of uncertainty, requires a lunwindow of ~2 ns. The green line shows the best fit into a 2 ns window. However, with no reflector tilt, the return signal is much narrower (Figure 45).

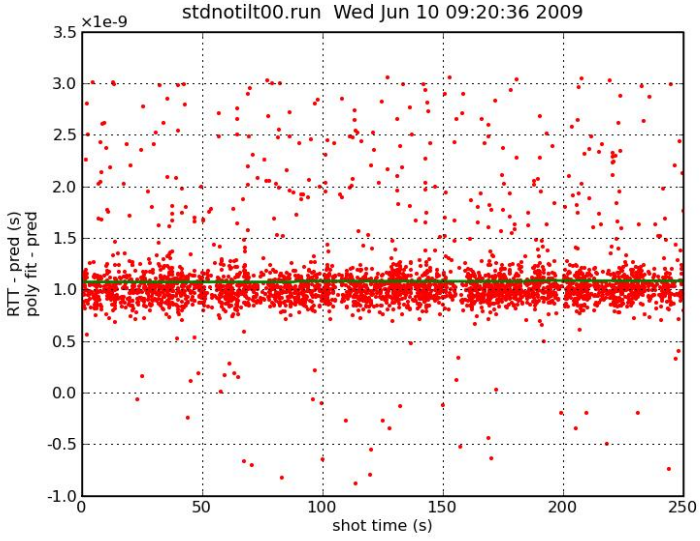


Figure 45: Simulated detections from no tilt A15 reflector.

The green line is again the best fit into a 2 ns window. This fit is clearly not through the

middle of the data points, however this is not necessarily a problem. So long as the tail truncation for both FIDs and LUNs is the same, neither of them need be represented by the middle of the peak of detection times. The problem with a single, fixed window size for all tilts is slightly more subtle.

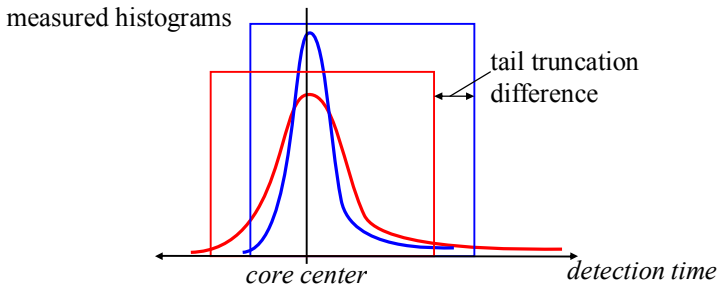


Figure 46: Effect of return spread on tail truncation.

Figure 46 shows two equal-sized windows for two return distributions, aligned to the same core -average for comparison. Each window is positioned to capture the maximum number of data points. The tail truncation is the time difference between the core-average and the right-hand window edge. The red window, representing the wider LUN distribution, is moved left by approximately half the width-difference of the distributions. Therefore, its tail truncation is shorter than the blue (FID) window, and would produce a systematic error in the LUN minus FID distribution. Therefore, a single, fixed window size produces bias, which turns out to be unacceptably large.

To quantify the effect of fixed window size, I ran simulations, and analyzed them with varying window widths, matching the FID and LUN windows to each other [lunwindow.bat]. The results show that for large reflector tilts, the AC method produces consistently biased NPs, of ~24 ps too short (~3-4 mm). This bias is stable for window sizes above 2 ns. This is consistent with the above qualitative analysis.

Also, for low-tilt returns, a fixed, large LUN window would include lots of high-uncertainty tail detections, and more background noise, than a window matched to the actual return width. (In contrast, the FID signal-to-noise ratio is very high, so increasing the FID window introduces little additional uncertainty.)

Because of these two effects, a variable window, matched to the signal width, is required for reducing both systematic error and uncertainty.

Adaptive Window

Instead of a fixed-size window, we use an adaptive window. The data analyzer knows the tilt angles (from prior knowledge of lunar orientation), and the outgoing pulse time spread (from the fiducials), so it computes a window size that is appropriate for each run. (This is the same information used in the PDF-fit method to construct the expected lunar return PDF.) Ideally, the window should be wide enough that it includes most of the valid returns, but narrow enough that the tail and background do not significantly degrade the uncertainty.

One can get a lower bound for the necessary window size by considering how the window size and window placement affect the average of a simple gaussian. By analytically computing the derivative of the gaussian CDF, one gets the sensitivity of the average of a sample to the window placement, as a function of window size (see Derivations appendix, and Figure 47):

$$\frac{d\bar{x}}{dp} = \frac{w \exp\left(-w^2/8\right)}{\operatorname{erf}\left(\frac{w}{2\sqrt{2}}\right)} \qquad \textit{where} \quad \operatorname{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x du\, e^{-u^2}, \text{ and } p \equiv \text{placement error}$$

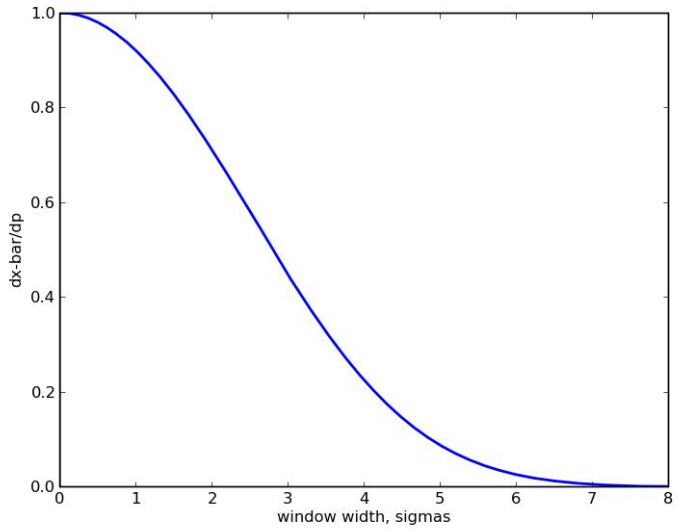


Figure 47: Sensitivity of gaussian average to window placement for various window sizes (normalized to σ = 1).

The vertical is $d\bar{x}/dp$, the rate of change of the sample average within the window divided by the window placement offset from perfect center. The horizontal is the window width in standard deviations. Clearly, a tiny window has an average that simply follows the window location, and $d\bar{x}/dp = 1$. A wide window includes nearly the entire gaussian distribution, and $d\bar{x}/dp \rightarrow 0$. If we assume (without any actual measures to date) that the window placement in a data reduction algorithm might occasionally be off by 1 histogram bin (25 ps ≈ 0.2σ), then a 6σ window width will give a < 1 ps (negligible) average error. This suggests ~6σ as a minimum width in APOLLO. (The fiducial window of 1 ns is ~6x the older FID core σ of 0.16 ns.) Note that the window could be off in either direction, and if equally likely, this effect will not introduce a systematic error, but will increase uncertainty.

The presence of the tail, however, makes the average somewhat more sensitive to window placement. Using the PDF of the tail, knowing that it is small at the window edge, we approximate

$$\frac{d\bar{x}}{dp} = \mathrm{pdf}_{tail}\left(\frac{w/2}{\tau}\right) t_{frac}\left(bin\_width\right)$$

For a worst-case (widest) window ~2 ns, a 1-bin error gives < 2 ps, also very small. The two effects are additive, and together are < 3 ps offset for a 1-bin window placement error.

My analyzer includes an adaptive window-size algorithm. The algorithm uses the σ of the FIDs, and the max-width of the trapezoidal PDF from the retroreflector, measured from foot to foot (wider than FWHM). The window width is set to 6σ + max-width of the trapezoid. However, to keep symmetry with the fiducials, the window width is not allowed to be smaller than the FID window (typically ~1 ns).

Background Photons

The background rate of detections, some thermally generated in the APDs, and some stray lunar photons from the bright moon, has an impact on both the RTT we compute, and its uncertainty.

RTT: To compute the average arrival time of signal photons, we must choose a window containing most of the valid measured points, and suffer the degradation of the background detections within that window. Some common algorithms, including that recommended by ILRS, iteratively shrink the window by excluding points outside, say, 3σ. (Note that usually you cannot do the reverse: *expand* the window based on standard deviation, because a constant background has infinite σ, and thus expands without bound.) Therefore, the window reduction algorithm works only if the starting window is small enough that the noise background is small compared to the actual data.

However, we avoid iterating window sizes because of our prior knowledge. The AC method uses two window sizes, both chosen independently of the data. Hence background or data problems will not affect the window sizes (but can affect window placement).

When we compute the average detection time within our window, the fiducial or lunar average is shifted by the background. The computed average is the weighted average of signal plus background, weighted by the number of signal, and background, photons in the window. The lunar background is strongly correlated to lunar phase, so without correction, the shift would introduce a highly deleterious systematic error, because it imitates a gravity signature.

If the window happens to be centered at the signal average time, then the background-shift is

0. The shift increases linearly with difference between window center and signal average. At our low background rate, background photons have an essentially flat PDF, across the window. Therefore, across runs, they average very precisely to the window midpoint. Correcting for this shift in the NP algorithm eliminates the *systematic* error due to background.

However, individual runs' averages vary by the standard error of the background mean, so this may degrade our NP uncertainty (examined shortly).

To form our time estimate of the signal, we estimate the background level, and subtract its "window-center" effect off the average detection time, using the following equation, relating average detection time <*det*>, to the average signal time <*sig*>, and average background time <*bkg*>:

$$\langle det \rangle = \frac{n_{signal}\langle sig \rangle + n_{background}\langle bkg \rangle}{n_{signal} + n_{background}} \qquad where \quad \langle det \rangle \equiv avg\left(signal\ and\ background\right)$$

We then solve for the average signal, since all other values are known or can be estimated:

$$(1) \qquad \langle sig \rangle = \frac{\left(n_{signal} + n_{background}\right)\langle det \rangle - n_{background}\langle bkg \rangle}{n_{signal}}$$

$$= \langle det \rangle + \underbrace{\frac{n_{background}}{n_{signal}}\left(\langle det \rangle - \langle bkg \rangle\right)}_{correction}$$

$$where \quad \langle det \rangle - \langle bkg \rangle \equiv \text{"window center offset"}$$

The average detection time is calculated from the data. The average background is the window center. We estimate the photon counts from the background rate. We estimate the background rate by counting detections outside the return window (Figure 49). (It is also possible to estimate by fitting the LUN histogram, but counting is more reliable.)

To see when the shifts might be significant, Figure 48 gives upper bounds to the "average shifts," for return rates of interest. The inputs are the background rate, the window size, the offset of the window center from the signal average, the signal rate, the overall detection standard deviation (signal + noise), and the number of shots in the run. The table includes mostly values for worst-case shifts: background rate = 0.75/gate, window size = 2 ns, window center offset = 0.037 ns, and number

of shots = 5000, but includes a low-background row for comparison:

	signal rate, /shot	window center offset, ns	bkgnd rate, /gate	detection sigma, ns	nshots	**average shift, ns**	Dev(avg detections), ns	Dev(avg signal), ns
max return	1.00	0.037	0.75	0.21	5000	< 0.001	0.003	0.005
strong return	0.50	0.037	0.75	0.21	5000	0.001	0.004	0.008
@ high libration	0.50	0.037	0.75	0.30	5000	0.001	0.006	0.011
typical return	0.05	0.037	0.75	0.21	5000	0.011	0.012	0.069
@ low background	0.05	0.037	0.15	0.21	5000	0.002	0.013	0.021
fiducials	1.80	0.25	0.15	0.21	5000	0.0004	0.002	0.002

Figure 48: Upper bounds on NP shift, and uncertainty degradation, due to background.

For typical return rates (~0.05 det/shot), these shifts are significant (up to 11 ps), and require correction. For strong returns (~0.5 det/shot), the correction is insignificant (~1 ps). The fiducial shift uses a larger window-center-offset, since it does not have a polynomial fit to better center it. This shows that the FID background effect is less than 1 ps, so can be ignored.

The last two columns, concerning uncertainty, are described next.

Uncertainty: Besides the systematic shift in average, the NP uncertainty is significantly affected by the background, and its correction. The simple estimate:

$$\sigma_{NP} = \frac{RMS\_residual}{\sqrt{n}} \qquad where \quad n \equiv \#\text{ detections in window} \qquad \text{(doesn't work)}$$

is seriously in error under most conditions.

We now compute the uncertainty of a signal average from a distribution raised by a constant background (Figure 49). Besides providing insight into our uncertainty, the AC method needs a way to compute the uncertainty of a normal point. This derivation is valid for any signal PDF, of which ours is one example.

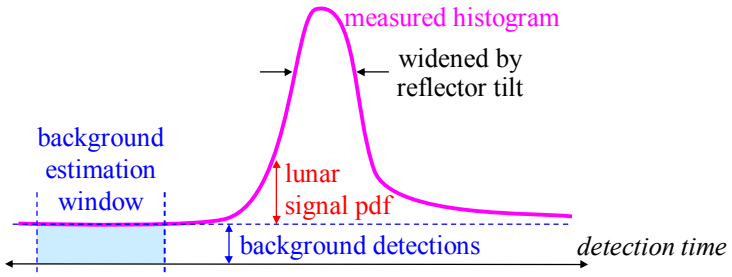


Figure 49: Components of lunar detection PDF

Without the background, we'd have for the uncertainty of our average signal estimate: Dev(<*sig*>) = Dev(<*det*>). The uncertainty in the background average time and background count degrades our Dev(<*sig*>). Propagating the uncertainties is tedious, requires some approximation, and recognizing the negative correlation between the number of signal and background detections. We outline the calculation here, because it illustrates the attention to detail required to compute an uncertainty for an NP with this method. Even then, the result is somewhat unsatisfactory, and if implemented, would require more work to produce the necessary uncertainties.

Define $\oplus \equiv$ *quadrature sum, i.e.* $a \oplus b \equiv \sqrt{a^2 + b^2}$

$n \equiv$ the number of detections in the window, signal + background

$n_b \equiv$ estimated $n_{background}$

$n_s \equiv n - n_b$ is the estimated n_{signal}

From equation (1) above, we compute the uncertainty:

(2) $$Dev\left(\langle sig \rangle\right) \approx Dev\left(\langle det \rangle\right) \oplus \underbrace{Dev\left(\frac{n_b}{n_s}\left(\langle det \rangle - \langle bkg \rangle\right)\right)}_{background\ correction}$$

where $Dev\left(\langle det \rangle\right) = Dev\left(det\right) / \sqrt{n_s + n_b}$

is the sample standard error of the mean of the detections

The uncertainty is approximate because the average detection time, <*det*>, appears in both terms of the correction equation given earlier. Therefore, the two terms are correlated. The strength of the correlation is weighted by n_b/n_s, so when the signal is larger than the background, i.e. when $(n_b/n_s)^2 < \sim 0.2$, the effect of the correlation on the uncertainty is small. We could rewrite the equation

so that <*det*> appears only once, but then n_b/n_s appears in each term, and the terms are still correlated (and other issues are actually worse).

In the absence of signal, n_b would be binomial up to the maximum detection opportunities. However, with a known fixed number of signal+background detections, the number that are background is only approximately binomial. Within that approximation, n_b has deviation:

$$Dev(n_b) = \sqrt{n_{opp} p (1-p)}$$

where $p \equiv$ the probability of a background detection, within the window, per detector, per shot, and n_{opp} is the number of such detection opportunities. $p << 1$, which means the binomial approximation is good, and also $(1-p) \approx 1$. Then,

$$Dev(n_b) = \sqrt{n_{opp} p (1-p)} = \sqrt{n_b}$$

We compute the deviation of the correction, the last term in (2) above, from the product rule:

$$(3) \qquad Dev\left(\frac{n_b}{n_s}\left(\langle det\rangle - \langle bkg\rangle\right)\right) = \frac{n_b}{n_s}\left(Dev\left(\langle det\rangle\right) \oplus Dev\left(\langle bkg\rangle\right)\right) + \left|\langle det\rangle - \langle bkg\rangle\right| Dev\left(\frac{n_b}{n_s}\right)$$

$$and \qquad Dev\left(\langle bkg\rangle\right) = \frac{\text{window\_size}}{\sqrt{12 n_b}}$$

Further, n_b and n_s are perfectly negatively correlated, since n is measured exactly, and each count that is background is one less count that is signal, shown by $n_s = n - n_b$. Therefore (see derivations appendix),

$$(4) \qquad Dev\left(\frac{n_b}{n_s}\right) = \left(\frac{d}{dx}\left[\frac{n_b + x}{n_s - x}\right]_{x=0}\right) Dev(n_b) = \frac{1}{n_s}\left(1 + \frac{n_b}{n_s}\right)\sqrt{n_b}$$

Equations (3) and (4) plug into equation (2) to give the final uncertainty of the lunar signal arrival time in terms of measured values. The last column of Figure 48 lists <*sig*> uncertainties for some typical values of our runs. For comparison, the preceding column lists the uncertainty of the core alone (without any background). In the worst-case (typical signal, high background) the uncertainty is degraded from 11 to 69 ps, due to background. For these conditions, this background degradation is unavoidable. Most of the high-background detections are from lunar stray photons, so

an improved APD, with lower thermal detections, would make only a small improvement.

This process illustrates an important fact: removing a global systematic error increases the variability of the final RTT; that is the inescapable cost of removing a statistically variable offset. Whether we should perform such a correction depends on the magnitudes of the effects. For example, if the systematic error is small (say, ~2 ps), and the uncertainty cost is large (say, ~20 ps), then it is not worth correcting. In the case of background shift, however, the systematic error is sometimes significant, so the uncertainty cost is unavoidable. (We will see later with First Photon Bias correction that *locally* removing an offset can actually *decrease* the uncertainty of the RTT.)

However, even without any bias, the uncertainty is substantially degraded by the presence of background. Note that simply subtracting the background count, and computing

$$\sigma_{NP} = \frac{RMS\_residual}{\sqrt{n_s}} \qquad where \quad n_s \equiv \text{estimated \# signal detections in window (doesn't work)}$$

still greatly underestimates the uncertainty.

Summarizing: At high background, for a single run of 5000 shots, with low libration, this method meets 1 mm *on that single run* only with high return rates > 0.5 photon/shot. For high libration, it requires averaging over 2 runs of 5000 shots, with return rates > 0.5 photon/shot, to reach 1 mm. A typical run of 0.05 det/shot, 5000 shots, even with low libration has about 10 mm uncertainty.

Simulation Results for Correction for Background Photons

I ran simulations to compare to the above calculations. Each set of simulations was 100 runs of identical parameters, giving 100 samples of background noise. Since the focus here is on backgrounds, the simulations omit any First Photon Bias, and the analysis omits FPB correction. For a worst-case test, the simulation is with maximum background, plus other realistic parameters. The results, in general, are too small to measure with confidence. Only one set of conditions, 0.05 photons/shot, has significant average shift. With no background correction, the RTT was too long by only 3.5 ± 2.6 ps. The background corrected RTT was too short by 2.5 ± 2.2 ps. These numbers are

consistent with the calculations, but the measurement uncertainties are fairly large.

The uncorrected uncertainty, which is scientifically meaningless since it assumes all the detections are signal, was 15 ps. The correction computes an uncertainty of 42 ps, however the actual standard deviation of the 100 runs, with correction, was 22 ps. The reason for the larger computed uncertainty is not known.

At higher return rates, 0.5 photon/shot and 1.0 photon/shot, the corrections and uncertainty degradations ~1 ps, which is immeasurably small.

Simulations at even lower rates, 0.01 photon/shot, where the background correction was larger, had a small number of outliers, likely caused by noise overwhelming the signal, causing the analysis to choose the wrong window of detections. Removing the outliers shows good correction, reducing a 28 ± 10 ps offset to only 7 ± 9 ps.

Augmented Calculation Summary

A summary of the AC performance to simulations (without FPB in the simulation or analysis) is given below.

LUN rate (photon/shot)	Other conditions	Bias, ps	Actual NP σ, ps
.05	no tilt	2 ± 2	20
.05	A11, 10^o x 2^o tilt	1 ± 2	23
0.5	A11, 10^o x 2^o tilt	4 ± 0.7	7
0.5	A15, 9^o x 6^o tilt	7 ± 1	11

Figure 50: Simulated performance of Augmented Calculation method for various conditions.

There is some significant systematic error with libration, up to ~1 mm worst case (A15, maximum tilt). Some of the original speculations about the AC method are not true. In particular, it is not especially simple, and with a variable-sized window matched to the known reflector tilt, there is not a problem with bias from tail truncation.

2.14 PDF Fit Analysis and Simulation

The PDF-fit method of NP generation was described previously. In this section, I quantify the

performance of the PDF-fit method of computing normal points, against various realistic parameters.

Theoretical analysis of the PDF-fit method is difficult: how fit parameters respond to impairments is often analytically intractable. We must rely more on simulations.

All of the earlier discussion about fitting to histograms, used to extract APD model parameters, applies as well to the PDF-fit method of calculating NPs. For example, all my PDF fits impose a maximum weight for any bin of 1, regardless of how small the model bin-count is. I did not re-examine the fit options for a sum-squared error figure of merit, or changing the fit bounds or rebinning parameters. I used the same settings as for the fiducial fits. Reuse of these preceding results makes this section misleadingly short: there are still many issues involved in the PDF-fit method, but they need not be repeated here.

In addition, a few other points are of practical importance. For example, when creating the trapezoidal PDF for constructing the overall LUN PDF, it is critical that it be perfectly symmetric. Any asymmetry in the trapezoid will translate to a bias in the resulting PDF, and final NP. Numerically, then, one generates only one half of it, and reflects the model bin-counts exactly across the zero point. In practice, this is more of an issue for simulations, where exact numbers sometimes appear, that can lead to unexpected truncation errors.

A second practical issue is setting the time-value of a histogram bin: it is important to define the bin time to be the center of the bin. This convention allows a Kronecker delta function to be defined: The $t = 0$ bin is set to 1, and all others are zero. This is important for some operations, such as the retroreflector tilt PDF at zero tilt: it must be a Kronecker delta function. In contrast, defining $t = 0$ to be *between* bins does not allow a Kronecker delta function to be defined.

Optimization of PDF-Fit Parameters

First, as a test of the code implementing the method, I created 100 calibration simulations. These are pure gaussian, $\sigma = 120$ ps, with no APD tail, no noise, and no FPB. Each run is 5000 shots. The results (in the table below) are good. The "ideal" NP σ is for a perfect measurement of the gaussian time points, with no quantization, and no fit to histograms. The actual σ shows degradation

consistent with those two effects.

LUN rate (photon/shot)	Bias, ps	Actual NP σ, ps	Ideal NP σ, ps
.05	< 1 ± 1	10	7.6
0.5	0 ± 1	4	2.4

Figure 51: Simulated performance of PDF-fit method for various conditions.

The bias was insignificant in both cases, at less than 1 ps. This confirms many, but not all, features of the implementation, and establishes a general expectation that the method is feasible.

I then proceeded to more realistic simulations, and fit criteria optimization. As with the AC method, I ran specific tests of the PDF-fit method on 100 simulated runs of identical parameters, including identical noise *statistics*, but different actual random noise [pdftest.bat]. Also as before, I here omitted FPB from the simulations and analysis, to focus on the baseline performance (FPB is considered later). This determines the noise performance of the PDF-fit method. I then varied some criteria to find good-performing choices.

PDF-Fit Window Size

When performing any fit, we must choose a window within the data over which to fit. For the fiducials, we choose a fairly wide time window (~ 4 ns), to get better results for the distribution tail parameters. However, for the lunar returns, we fit only for the core-average parameter. The following simulations show that fitting over the tail provides no benefit.

The widest distribution, from a maximum A15 reflector tilt, fits the entire main peak within a 2ns window. I therefore compared PDF-fit windows of 1, 2, 3, and 4 ns (the FID window size). In general, this fit window may be reduced, however, by the fit_bound parameter. As expected, at high libration, the 1ns window cuts out too much of the main peak of the return histogram, and fails completely. The performance of 2, 3, and 4 ns windows were *identical* under all conditions tested (to <1 ps). The following table lists the results for the 2 ns window [pdftest.bat]:

LUN rate (photon/shot)	Other conditions	Bias, ps	Actual NP σ, ps	Estimated uncertainty, ps
.05	A11, 9° x 6° tilt	< 1 ± 2	21	**16**
0.5	A11, 9° x 6° tilt	< 2 ± 0.6	6	**5**
0.5	A15, 9° x 6° tilt	–5 ± 1	9	**9**

Figure 52: PDF-fit performance under various conditions, for a 2 ns fit window.

The "estimated uncertainty" column is described later.

Because a 2 ns fit window gives identical results to 3 or 4 ns windows, even for the A15 reflector at maximum libration (9° longitude, 6° latitude), I chose the 2 ns window. It is less susceptible to potential disturbances in the tail. At the same time, it is probably not worth implementing a variable-sized window based on the known libration. This would add complexity, with no identifiable benefit. Finally, looking ahead, the First Photon Bias correction algorithm is possibly simpler and/or more effective with a fixed-size window.

The A15 reflector at maximum tilt shows ~6 ps bias toward shorter ranges. It's not clear why. A typical plot of the simulated lunar histogram (without FPB) looks reasonable (Figure 53), though it is not possible to "eyeball" a 6 ps bias (1/4 bin).

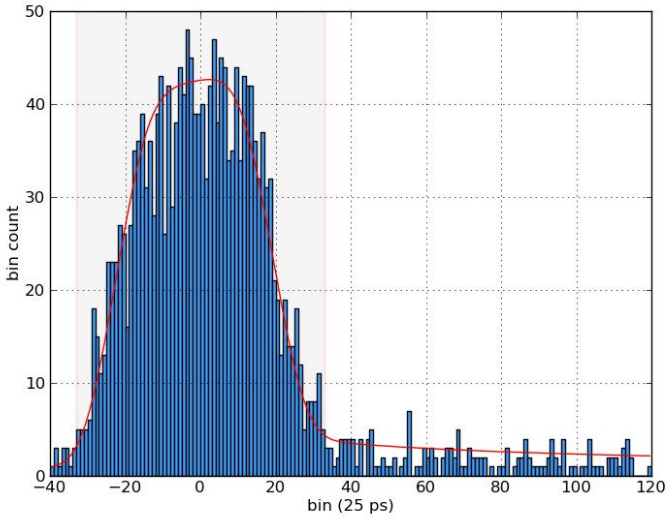


Figure 53: Typical Apollo 15 simulation, with high libration, and ~6 ps bias.

Counted vs. fit background

Some lunar returns have a background detection level due to lunar illumination from the sun, in addition to the "dark rate" of the APOLLO detection system itself. We must estimate the total background level, and include it in the computed PDF that we fit to the LUN histogram. Two ways of estimating this level are (1) to count background detections from a separate time window before the main peak (thus avoiding the tail), and (2) fit for background level as a parameter in the histogram fit. The counted background rate was described in the AC method (Figure 49), and uses a 20 ns window. With our small PDF-fit window (~ 2 ns), the fitted estimate of background rate from the main peak is necessarily worse than that from a 20 ns separate window. Also, the fit problems with low bin-counts (outside the main peak) hit the background estimate quite hard. Within the peak, background may be overwhelmed by signal. Therefore, we expect that counted background is better. I ran simulations of the fit-background method, and compared with the counted-background results given above:

LUN rate (photon/shot)	Other conditions	Bias, ps	Actual NP σ, ps	Estimated uncertainty, ps
.05	A11, 9^o x 6^o tilt counted background	< 1 ± 2	21	15
	fit background	3 ± 3	26	17
0.5	A11, 9^o x 6^o tilt counted background	2 ± 0.6	6	5
	fit background	2 ± 0.5	6	5
0.5	A15, 9^o x 6^o tilt counted background	5 ± 1	9	9
	fit background	4 ± 1	9	9

Figure 54: Comparison of counted background vs. fit background, under various conditions.

Not surprisingly, the low-signal returns are affected more than high-signal, and are noticeably degraded. Also, in general (as with choosing a narrower fit window earlier), reducing the demands of the fit (eliminating the background fit parameter) reduces susceptibility to unforeseen problems. It is therefore best to use the counted background. The lunar fits then have only 1 free parameter: the core-average location.

Uncertainty Calculation

As a simplified measure, I created an "estimated uncertainty" by quadrature summing the fiducial core-deviation with the standard deviation of the reflector trapezoid PDF, then accounting for the number of measurement points, and finally including a rough factor for the increase due to fitting to a histogram:

$$\text{estimated uncertainty} = 1.05\sqrt{\frac{\sigma_{fiducial}^2 + \sigma_{trapezoid}^2}{\#\ detections}}$$

This ignores the effect of background, and some small spreading from the fraction of the tail that is included in the main peak.

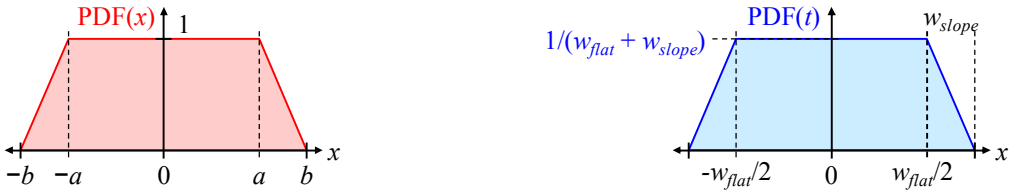


Figure 55: (Left) General trapezoid. (Right) Retroreflector trapezoid.

The PDF for the retroreflector array is given in terms of the width of the flat region, w_{flat}, and the width of each of the sloped regions, w_{slope} (Figure 55). The trapezoid variance is given by (see Derivations appendix):

$$Let \qquad a \equiv w_{flat}/2, \qquad b \equiv w_{flat}/2 + w_{slope}$$

$$\sigma^2 = \frac{1}{w_{flat} + w_{slope}}\left[\frac{2b^3}{3} - \frac{1}{b-a}\left(\frac{b^4 - a^4}{2} - \frac{2a\left(b^3 - a^3\right)}{3}\right)\right]$$

As shown in the last columns of Figure 52 and Figure 54, this estimate of uncertainty is too low at low return rates, consistent with the effect of background seen in the AC method. Note that a realistic uncertainty calculation for the PDF-fit method cannot be based solely on the formal errors of the LUN fit parameters; it must also include the effect of background, and the uncertainty of the FID parameters that are fixed in the LUN fit. The formal error of the LUN core-average reflects the *fixed* background noise sample of that run, but does *not* include the *population* of noise. It's somewhat like

fitting a line to non-collinear points with tight uncertainties. The formal error is small, but as an experimental measurement, the uncertainty is larger (indicated by a large χ^2 parameter). I did not investigate the details of a better uncertainty estimate, though I expect a reasonable estimate would not be hard to develop.

PDF-fit Method Summary

In most systems, one cannot optimize for all signal conditions simultaneously (this is true for almost any optimization effort). As mentioned earlier, we lean toward optimizing for the best cases, even though there are fewer of them, since they contribute far more to the final solar-system fit, weighted as $1/\sigma^2$.

Accurate libration angles are essential: when the PDF doesn't match the histogram, it shifts over to maximize the "overlap." With the asymmetric distribution, this will result in a biased NP. Fortunately, accurate libration angles are easy to predict.

A good choice for parameters of the method are to use a 2 ns fit-window (maximum, subject to fit_bound), and to estimate the background for the LUN fit by counting in a time window preceding the main peak.

A reasonable estimate of uncertainty requires including the effects of FID fit parameter uncertainties, and the effect of the population of potential background detections. A specific algorithm would need to be developed, which I expect would not be hard.

Possible Future Study of PDF-Fit Method

Besides developing a specific uncertainty measure, it might be helpful to investigate the following:

Systematic error in high-libration A15 runs: Clearly, it would be helpful to understand and eliminate the ~ –6 ps systematic error in high-libration A15 results. In general, the high libration runs have uncertainties larger than the bias, so the impact of the bias is somewhat reduced. But over time, given a sufficiently good model, the uncertainties of the high-libration NPs might be "averaged out;" the same is not true of the bias. In particular, the window size formula for the AC method was

chosen somewhat ad-hoc. A more quantitatively derived choice, based on the analysis here of windowing and tail truncation, might reduce the systematic error.

Figure of merit: Besides χ^2 and MSSE, there is also a "g-parameter" for goodness of fit. The g-parameter is based on log-likelihood functions, and is claimed by some to work better on histograms with small bin counts.

Rebinning: I did not re-examine the fit criteria for the fit bounds or rebinning parameters. I used the same settings as for the fiducial fits. The background level for LUNs is sometimes much higher than for FIDs, due to the bright lunar illumination. The fit criteria could be reconsidered for the LUN fit.

Sensitity to tail parameters: It may be helpful to determine the sensitivity of the PDF-fit method to errors in t_frac and t_decay, since these parameters come from the FID fit, and have some uncertainty. This effect is included in the general simulations, but it is aggregated with all other impairments, and is not separately identified. We might consider completely fixing these parameters ahead of time, and not fitting for them from the fiducial histogram. There is some risk, however, since the tail parameters depend on the reverse bias voltage of the APDs, and could change over time.

2.15 Comparisons of Reduction Methods and Conclusions

The following table compares the two methods for several sets of conditions.

LUN rate (photon/shot)	Other conditions	Bias, ps	Actual NP σ, ps
.05	no tilt augmented calculation	2 ± 2	20
	PDF-fit	< 1 ± 1	12
.05	A11, 10^o x 2^o tilt augmented calculation	1 ± 2	23
	PDF-fit	1 ± 2	21
0.5	A11, 10^o x 2^o tilt augmented calculation	4 ± 0.7	7
	PDF-fit	2 ± 0.6	6
0.5	A15, 9^o x 6^o tilt augmented calculation	7 ± 1	11
	PDF-fit	-5 ± 1	9.

Figure 56: Comparison of Augmented Calculation and PDF-Fit methods.

1. The PDF-fit method is currently a modestly better choice. It's uncertainty is slightly better for weak signals. Despite the concern about its positional information coming only from the edges, it does not appear to suffer compared to the AC method. The higher than expected uncertainty of the AC method is almost certainly due to the background detections, and the uncertainty in correcting for them, though the extent of the degradation is still puzzling to me.

2. The possibility that Augmented Calculation would be simpler and more direct than PDF-fit doesn't hold up. As described previously, Augmented Calculation uses the same prediction-subtraction, and removal of the straight-line component of prediction residual, that the PDF-fit method uses (though the benefit from that is small). Also, the corrections for background, and the resulting careful window choices, increase the complexity of the AC method compared to a simple window-and-average.

3. Real data reduction results are probably somewhat worse than simulations, due to varying laser conditions, echo bumps, etc. For example, the end of year 2007 shows a clear degradation in the consistency of fiducial fit parameters. Also, year 2008 has statistically significantly different FID fit parameters than 2007, although the differences are not large on an absolute scale. Even though we fit

each run's own fiducial PDF, such degradations might include other effects, such as possible deviations from the fit model. Also, there is some apparent variation in the per-channel offsets between FIDs and LUNs, which causes noticeable lack-of-fit in the LUN PDF.

2.16 Possible Future Enhancements to Data Reduction

1. Excluding the first returns of a run: The first detections in a run are often just noise while we hunt for the signal. It is likely a benefit to exclude these first lunar returns from the data, until the time in a run when a reliable signal is found. We might use a statistical test of photon rate in the predicted window to determine when valid lunar photons started being detected. We must then also decide if we should eliminate the early fiducials, to keep our measurement more accurately a differential between fiducial and lunar measurements. Given the much higher fiducial return rate, eliminating some of them (so long as we retain a thousand or so) won't significantly impair them compared to the LUNs, so this is likely a good choice.

2. Both AC and PDF-fit methods have some significant systematic errors under some conditions. They occur under high libration angles, when the NPs are not as heavily weighted. Nonetheless, understanding and eliminating these errors might improve the overall APOLLO results.

3. The Cramer-Rao bound is a very general lower bound on the uncertainty of any estimated parameter, given information about the signal and noise. Estimating the arrival time of a signal is a common application for the Cramer-Rao bound, so it is applicable to APOLLO. The bound uses frequency domain information from both signal and noise. It is often helpful because if a given algorithm performs close to the bound, there is little benefit to investigating methods of improvement. It would be interesting to investigate the Cramer-Rao bound for APOLLO, probably separately for different conditions, and perhaps computing separate bounds for steps of the computation, such as FID time and LUN time.

3 First Photon Bias

3.1 Introduction to First Photon Bias (FPB)

Recall that the APD array comprises 16 individual APDs, with 10-12 of them working properly in the current APOLLO hardware. On a given shot, each APD either registers a detection, or it does not. The system knows how many and which APDs detected something in each shot. There are often zero detections, and in principle, there could be as many detections as there are working APDs in the array. An APD can make only one detection in a gate. The APDs are "reset" before each fiducial event, and each lunar event, allowing all of them to make another detection. The timing electronics measures the time of detection with ~25 ps (3.7 mm) resolution.

High photon rates have a significant chance of 2 (or more) photons hitting a *single* APD (one element of the APD array) in a shot. The APD fires on the first arrival, ignoring all others, resulting in an early bias for these detections. An individual APD gives no indication of how many photons may have hit it. Figure 57 gives the detection time PDF for different numbers of detectable photons received, for a hypothetical gaussian detection time PDF. Note that FPB increases with the number of detectable photons received in an APD, i.e. the average detection time moves earlier.

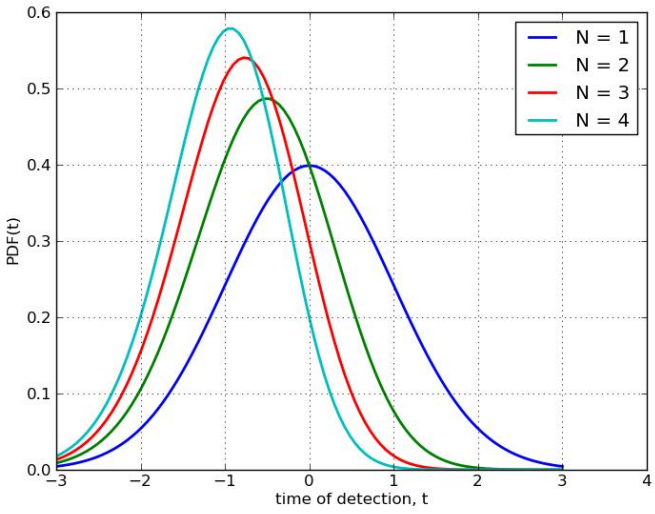


Figure 57: First photon bias increases with the number of detectable photons received.

The figure shows that for even a small arrival σ of 120 ps, the average arrival time for two detectable photons in a single detector is biased by ~60 ps, for three the bias is ~100 ps, and for 4 the bias is ~110 ps. In practice, the occurrence of so many photons in a single detector is rare enough that the *average* bias is usually smaller (much more on this later).

We can readily derive the detection time PDF from the single-photon detection time as follows: Assume the APD receives N detectable photons, each of which has a detection time probability distribution given by $pdf_{det}(t)$, and cumulative distribution given by $cdf_{det}(t)$. The APD responds to the first photon. The probability of a *given* one of the photons being detected first at time t is $pdf_{det}(t)$ times the probability that the $(N-1)$ other photons would have been detected later:

$$pdf_{given}(t) = pdf_{\det}(t)\left(1 - cdf_{\det}(t)\right)^{N-1}$$

There are N photons, each of which might be detected first, which are N distinct events, so their probabilities add. Therefore, the final (biased) detection probability is:

$$pdf_{biased}(t) = N\ pdf_{\det}(t)\left(1 - cdf_{\det}(t)\right)^{N-1}$$

We note that this is necessarily normalized, because

$$\int_{-\infty}^{\infty} N\ pdf_{\det}(t)\left(1 - cdf_{\det}(t)\right)^{N-1}\, dt = -N\int_1^0 U^{N-1}\, dU = \left[-U^N\right]_1^0 = 1$$

$$where \quad U \equiv \left(1 - cdf_{\det}(t)\right), \quad dU = -pdf_{\det}(t)\, dt, \quad U(-\infty) = 1, \quad U(\infty) = 0$$

First photon bias affects both fiducial and lunar photons. Early bias in FIDs increases the measured round-trip time (RTT); early bias in LUNs decreases the measured RTT. Typically, the fiducial detections are at a reasonably constant intensity within a run, and even across most runs. Operators manually adjust the fiducial detection rate during runs to keep it within about 20% of 1.5 detections per shot (across all 10-12 working APDs). However, lunar detections are quite different than fiducial detections. Lunar return rates often vary widely over a run. While the *average* rate over a run is almost always lower for LUNs than FIDs, short term rates are sometimes higher. This leads to a somewhat complicated variation in lunar FPB, where the bias varies throughout the run.

Speckle: Fluctuations in the atmosphere cause small-scale lensing due to air pockets with

varying conditions (such as temperature). The interference between lensing elements breaks up a coherent image into moving "dots" ~0.04 arcsec in size, at 532 nm and ~2.5 m aperture (the telescope is underfilled). The speckle dots have highly variable intensity from the varying interference from the multiple "lenses." The pattern of dots varies rapidly, becoming nearly uncorrelated on a scale of ~50 ms, which is our shot period. Speckle occurs on both the outbound laser pulse, and the return image of the reflector. For APOLLO, the dominant effect is the highly variable illumination of the retroreflector from the speckle image of the outbound pulse on the moon. The reflectors are much smaller than a speckle dot, so their illumination varies widely from shot to shot. Therefore, the return intensity varies widely, as well. The return image of the reflector on the detector array is also speckled. However, APOLLO pixels are ~0.35 arcsec on a side, so there are ~64 speckle dots per pixel. The detectors receive the sum of these dot intensities, which significantly suppresses the variation due to receive-side speckle, such that it is negligible.

With speckle, the first photon bias is significant at *average* photon levels > ~0.1 photons detected per shot. With the current speckle model, FPB is typically ~1 to 100 ps (~0 to 14 mm).

First photon bias increases with increasing time *spread* of received photons (Figure 58). This includes all system variations up to the APD detection, including retroreflector tilt. It does *not* include variation introduced by the timing electronics downstream of the APDs. Since lunar photons have a larger and more variable (from run to run) time-spread than fiducials (due to retroreflector tilt), this is a very important characteristic of FPB. Standard deviation for FIDs (before the timing electronic variation) is ~95 ps; for LUNs is anywhere from ~95 - 320 ps.

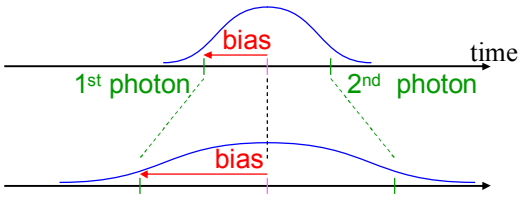


Figure 58: First photon bias is directly proportional to the time-spread of photon arrival.

Clearly, FPB depends on the shot-to-shot average intensity, with higher intensity having

higher bias. However, the FPB *also* depends on the shot-to-shot intensity *distribution*. One can see this intuitively by considering two extreme cases. First, consider that the shot-to-shot intensity is essentially uniform (as it is with FIDs). This results in some bias. Now consider the same *average* detection rate, but with half the shots receiving no detections at all, and half receiving twice the average. The FPB is determined only by those shots which have detections, and all the shots that have detections are at twice the average rate. Therefore, the FPB in this case is larger than the same average rate with uniform shot-to-shot intensity.

Recall that for any given shot, our APD detectors are essentially independent. Work of Adam Orin's in the UCSD lab shows that the crosstalk rate for detections is low, and almost all crosstalk induced detections are outside the time window used for the normal point (NP).

Summarizing: There are three dependencies of FPB: (1) The bias varies proportionally to the spread of arrival times. (2) The bias depends on the intensity: more intensity means more chance of multiple-photons hitting a detector, and therefore more bias. (3) The bias depends on the shot-to-shot *distribution* of intensity, even at fixed *average* detection rate: the more top-heavy the distribution, the greater the bias (discussed more below).

Note also:

- The number of *detections* in a shot is always less than or equal to the number of *detectable photons* in the shot.
- On any given shot (or run), the best we can do is *estimate* the bias, since we can never know it exactly. Our estimate has variation, which contributes *directly* to the variation in each photon's RTT (round trip time). Thus this variation is aggregated with all other variations into the final NP.
- Multiple detectors are essential for estimating photon flux in a shot. The APD array has up to 15 usable detectors, with only 11 fully functioning as of 6/2010. One more detector is intermittent. Each detector is (nearly) independent of the others, i.e. crosstalk is negligible.

  FPB is not limited to avalanche photo-diodes. Often, two or more photons arrive within 1 ns

of each other. Any detector (and its associated electronics) that cannot resolve this distinction will suffer some amount of FPB, even detectors with proportional response such as a photomultiplier with constant fraction discrimination.

Difficulties In Correcting FPB

The work in this chapter was done with reasonable models, believed at the time to be representative of our actual data. In particular, this work uses a simulated speckle model, and an assumed minimum return signal spot size of ~1 arcsec full-width half-maximum (FWHM). Chapter 4 compares our actual data to the results here.

Correcting FPB is made more difficult by several factors. First, as noted above, the correction algorithm must estimate the arrival time standard deviation. It must start with the detection statistics, which includes contributions from the downstream electronics that do *not* affect FPB. The algorithm must compute the detection deviation, and "back out" contributions from the electronics, to estimate the actual photon arrival time deviation. We must determine these downstream contributions from separate tests, outside of our usual run data. We rely on these deviations being reasonably stable. The downstream deviations are the (Fast Photo-Diode + detection electronics) ≈ 55 ps, the APD core deviation ≈ 25 ps (including position dependent delays), and the TDC deviation ≈ 13 ps. Contributors to the true photon arrival time standard deviation are the laser pulse temporal profile ~ 120 ps, and retroreflector tilt ~ 0 - 320 ps. In the end, our estimate of the photon arrival deviation includes some uncertainty, which then propagates into uncertainty in the FPB correction.

Another significant impediment to FPB correction is that the lunar flux varies significantly over time, even within a single run. This is due to varying conditions, including clouds, telescope pointing, seeing, and possibly other atmospheric disturbances. We rely on the these variations being significant only on time scales above several seconds. Our estimate of flux on a given shot comes from counting the number of detections in that shot, and nearby shots. These counts are noisy: a sample of at most 13 working detectors is fairly small. Uncertainty in the flux estimate propagates into uncertainty in the final NP.

Another possible impairment is that the spatial distribution of intensity varies over the APD array, i.e., the APDs are not uniformly illuminated. My investigation showed that down to < 1 arcsec, this is not a significant impairment (details later).

Each APD detector has its own background rate and gain (detection sensitivity). These are not necessarily stable over time: some detectors flake in and out over a span of days. It is possible that these variations affect FPB correction, however I did not study this effect.

The correction for first photon bias needs to know how many detectors are functioning. The estimate of the photon flux on the shot includes the number of detections in that shot (as well as the number of detections in nearby shots). Varying numbers of functioning detectors changes the meaning of a detection count, and makes first photon bias correction more complicated and less reliable. I did not investigate this in detail. The current implementation adjusts the detection rate to the equivalent for 13 working detectors.

When trying to correct for a short-term offset, it is well established in statistical quality control that making too frequent adjustments, i.e. with insufficient statistical information, makes the results *more* variable, rather than less variable. (See Appendix for a simple example of this.) Therefore, when we compensate for first photon bias, we must quantitatively consider the statistical basis for our correction, and include its variations in our overall analysis.

Precise Definition of First Photon Bias Is Difficult

For "compact" probability distributions (e.g., gaussian), which rapidly approach zero outside of a few standard deviations, a precise definition of FPB is simple. For example, for rectangular, triangular, or gaussian distributions, the shift in average arrival time is well defined, and can be defined as the first photon bias. Such a shift is easily simulated, and can be numerically evaluated.

For "wide" distributions, such as our actual APD distribution with a long tail, the strict average is not a useful measure. The average of the whole APD distribution is shifted far to the right by the tail, and the standard deviation is much too large for our purposes. Therefore, we specifically design our data reduction algorithms to focus on the core bump of the distribution, and to be largely

insensitive to the tail. This focus dramatically changes the FPB reflected in the normal point. Therefore, a practical working definition of FPB is "how much our algorithm shifts the normal point due to multiple photons in a single detection event." In principle, this makes the FPB dependent on the reduction algorithm, but my simulations show that in practice, both the augmented calculation and the PDF-fit methods agree on the FPB to within a few picoseconds.

FPB Research Questions

In my research, I sought to answer the following questions:

- How big is the problem of FPB? Certainly, the magnitude of the problem varies with observing conditions.
- Can FPB be corrected?
- How well can it be corrected, in principle?
- By what algorithm can we reasonably correct FPB, and to what level?

I used a combination of analysis and simulation to examine these questions. I tested the resulting algorithm against my simulations of the full APOLLO system with known parameters. It is important that we seek the *average* bias (which is shown to be representative), and *not* the maximum likelihood bias. We are interested in unbiased NPs, so that the long-term aggregate of NPs "averages out" to the true RTT. In general, maximum likelihood estimators for asymmetric distributions are biased.

When considering all of the following results, keep in mind that even on the highest intensity returns, the overwhelming majority of detections occur in shots with only a few detections or less. Therefore, in the end, we will likely include in the normal point only such shots, and discard shots with higher detection counts, because the uncertainty of their FPB probably outweighs their benefit to the NP.

As of 6/2010, all APOLLO NPs have attempted to reduce FPB by including in the NP generation only those shots with a single detection, discarding shots with two or more detections. This selection includes those shots most likely to be of low intensity, and with low FPB. This reduces FPB

to < ~ 1mm for most conditions (as shown below), and is a very big improvement in bias over including all shots, at some cost in detection number and therefore random uncertainty.

3.2 General Consideration for Correcting First Photon Bias

We must distinguish two important terms: the **detectable photon rate**, and the **detection rate**. The detectable photon rate, α, is the rate at which a detectable photon hits the detector, and is given as a number per detector per shot (i.e., per detection-attempt). However, since the detector cannot distinguish between one or more photons, the detection rate, μ, is always less than the detectable photon rate. In short,

α average *detectable photon* rate, per detector

μ_1 average *detection* rate per detector (always < α)

In principle, α is unbounded for arbitrarily intense light. μ_1 is in [0, 1]. We can relate the two as follows, noting that a single return photon can be spread over many square km. A detectable photon is a small probability out of a huge number of chances: we detect a few photons out of thousands that return detectably. This is the defining characteristic of a Poisson distribution. Therefore, the number of detectable photons per shot follows that distribution:

$$\Pr(n \text{ detectable}) = \frac{\alpha^n}{n!} e^{-\alpha} \quad \textit{where} \quad \alpha \text{ is the average \# detectable photons}$$

The average number of *detections* per shot is just 1 minus the number of shots with 0 detectable photons:

$$\text{detection-rate} \equiv \mu_1 = 1 - \Pr(n = 0) = 1 - e^{-\alpha}$$

For small α,

$$\mu_1 = 1 - e^{-\alpha} \approx 1 - (1 - \alpha) = \alpha \qquad (\alpha \textit{ small})$$

as expected. We can measure μ_1 directly, but we sometimes need α for theoretical analysis:

$$\alpha = -\ln(1 - \mu_1)$$

The above is exact for a uniform shot-to-shot intensity. However, the difference between α

and μ_1 grows with α. Illumination of the reflector with a speckle intensity pattern leads to a distribution of lunar intensities, referred to here as a **speckle distribution**. As a result of speckle, α varies from shot to shot. This variation causes the *average* α and *average* μ_1 to differ significantly from the above relations, and especially from the $\alpha \approx \mu_1$ relation. Once again, simulation must provide the working relationship across a range of shots with speckle.

I considered several approaches to correct for FPB. Here are some approaches, with brief comments. Each is described in more detail following.

- For each shot, use the number of detections on that shot alone to estimate the bias. This gives poor results, and does not eliminate the offset due to the underlying intensity (see below).
- Use both the detections on each shot, and the average detection rate for the run. This improves over the above, but suffers from large variations in underlying return signal intensity within a run.
- Use the detections on each shot, and the nearby average detection rate. This gives the best results, and is only slightly more work than using the global average detection rate for the run.

Use only the number of detections on that shot to compensate for FPB

Each observing night has, in general, a different average lunar return intensity. Some nights give higher returns than others. The underlying intensities are important for FPB correction, because different underlying intensities significantly change the bias-per-number-of-detections relationship. That is, the bias in a shot depends not only on the number of detections in that shot, but significantly on the average illumination intensity underlying the number of detections. For example, 3 detections could mean a bias of 4 ps (for low underlying intensity), or 13 ps (for high underlying intensity). If FPB correction considered only the number of detections on each shot (separately), a given night's underlying intensity would continue to bias the NPs for that night. Since the return rate partially depends on libration angles and lunar phase, these biases could mimic motion signatures, and result in significant systematic errors in science signals.

Use both the detections on each shot, and the run-average detection rate

Including the run-average detection rate information to estimate FPB improves over the above, but suffers from large variations in underlying return signal intensity *during* a run. For example, it often takes significant time to find the signal, and then more time to optimize it. This causes the rate to vary from zero, to some initial rate before optimizing, to some optimized rate significantly higher. Additionally, conditions vary during the run, including pointing drift (the earth rotates, and the moon revolves), clouds, and air turbulence (seeing). Also, some nights we acquire the signal quickly, and some nights it takes longer. This means sometimes we have a strong signal right from the beginning, and other times it's weak at first, and become strong later. Therefore, there is still substantial remaining bias with this correction approach.

Use the detections on each shot, and the nearby average detection rate

The algorithm I chose uses multiple nearby shots to give a reliable estimate of the underlying photon rate *at the time of each shot*. This gives the best results, and is only slightly more work than using the run-average detection rate. I use a rolling average of ~81 nearby shots to estimate the underlying average intensity (~40 before, the current shot, and ~40 after). This is a time-window of ~4 s. Error analysis (presented later) shows that the illumination random uncertainty in this case is negligible. The time-scale of intra-run underlying intensity variations (i.e., non-speckle) is believed to be mostly on the order of many seconds, which is slower than this average shot intensity tracking. Therefore, this method produces the best results, and does not have any significant remaining bias.

3.3 First Photon Bias for Compact Distributions

Before studying the more complicated case of our APD detection distribution (with its long tail), it is informative to first examine the simpler cases of compact distributions. By "compact," I mean that the distribution drops rapidly to zero within a few standard deviations of the average. I considered 3 time distributions: a gaussian (the tightest), a triangular (intermediate "tightness"), and a rectangular (the broadest).

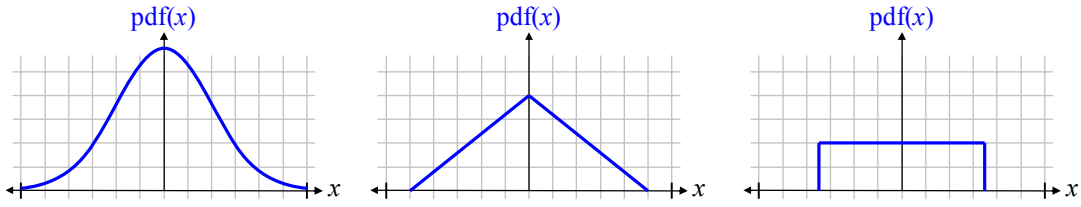


Figure 59: Compact distributions that were studied: gaussian, triangle, and uniform, to scale with equal σ.

The Fiducial Case: (Nearly) Constant Illumination

For fiducial detections, the laser shots are very uniform shot-to-shot. This is the simplest case for first-photon-bias (FPB). Each shot is independent of the others, and as always, each detector is independent of the others. (I did not consider the case of an unstable laser, whose intensity fluctuates within a single run, because it is rare.) The intensity of fiducials is uniform across the detectors (i.e., spatially uniform).

Why simulate these simple cases? Can we not numerically evaluate the time PDF? In principle, for the simple distributions, and simple shot-to-shot intensity profiles, we could numerically evaluate the time PDF. However, in our actual data reduction, some of our processing does not lend itself to mathematical analysis, such as the windowing to the central bump of the PDF. Also, the lunar returns have the more complicated speckle distribution of shot-to-shot intensity. For these realistic cases, mathematical analysis fails, and simulation is a viable alternative. These simple simulations are subsets of the simulations to come, and establish the tools and methods for the realistic cases.

Note that "random" or sampling error in the simulations becomes systematic error in the system when the simulation results are included in the lookup tables for the FPB correction algorithm. Therefore, we need a very large number of simulated detections to control these systematic errors. Thus for computation speed, these simulations focus on the APD behavior, and do not simulate the entire system. They are the starting point for developing the algorithms, and establishing the final lookup tables used to correct for FPB.

The simulations below use a typical detection σ = 120 ps, uniform shot-to-shot illumination,

and 1 million shots. Note that the width of each of the gaussian, triangle, and uniform distributions is chosen to produce the desired σ. For the fiducial case of uniform shot-to-shot intensity, each detection is independent of the others, so there is no information to be gained from the number of detections in a single shot. Therefore, the simulations produce bias as a function only of overall illumination intensity. There is a separate curve for each time distribution (gaussian, triangle, uniform). Figure 60 shows the three curves, which are so similar they are barely distinguishable. Note that we cannot know the number of detectable photons; our only measure of intensity is the detection rate over the 11-13 working APDs. There is only a fraction of a picosecond difference between the distributions in the most extreme case. This shows that the FPB, under these conditions, is largely insensitive to the time distribution, when σ is fixed. This motivates the study of FPB for our actual detections, with the expectation that the bias is largely insensitive to the exact detection distribution, and therefore APD parameters and reflector tilt. We confirm that expectation later.

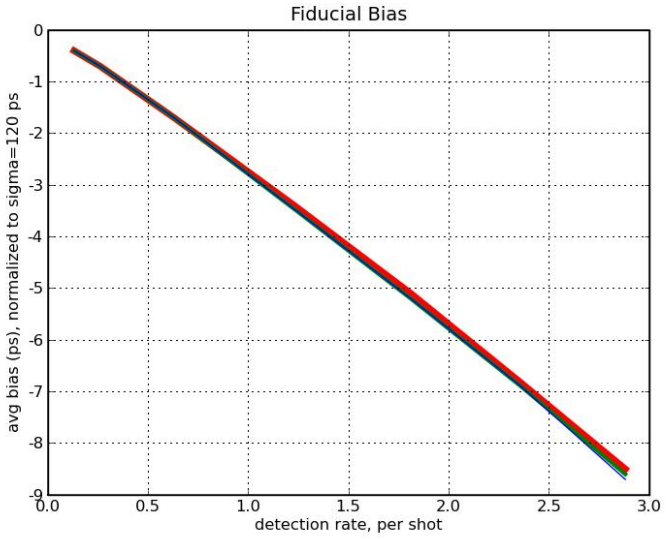


Figure 60: First Photon Bias as a function of detection rate over 13 detectors, for three different distributions, fiducial case: (red) gaussian; (green) triangle; (blue) uniform.

The Lunar Case: Speckle Illumination

The phenomenon of speckle causes the light intensity to vary greatly, and randomly, from shot to shot. I call the distribution of these intensities a "speckle distribution." The number of

detections in a shot is correlated to the intensity of that shot. Therefore, the FPB for a shot becomes a strong function of the number of detections in that shot, in contrast to the fiducial case. As an example, at various *average* detection rates, but with speckle illumination statistics from shot to shot, Figure 61 shows the bias as a function of the number of detections in the shot. This simulation used 10 million shots, gaussian distribution, and detectable photon rates per detector (α) of .01, .02, .05, .10, .15, .20, .25 photons/detector/shot. As always, though, the FPB correction cannot know the detectable photon rate; it only knows the detection rate. From the α's, the simulation measures detection rates (over 13 detectors). The α = .01 (blue) curve is invalid above 5 detections, due to the small sample size that results from such a low photon rate.

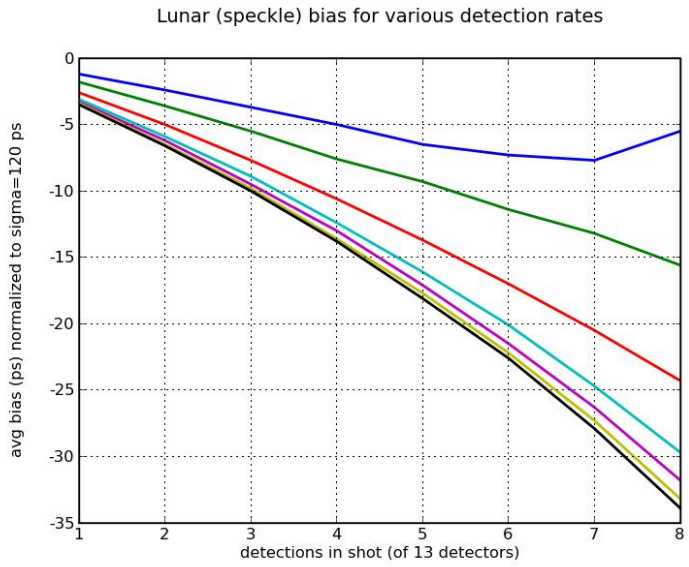


Figure 61: First Photon Bias vs. number of detections in the shot, for average detection rates of 0.13 (blue), 0.25 (green), 0.57 (red), 1.03 (cyan), 1.41 (magenta), 1.74 (yellow), and 2.03 (black) det/shot.. The blue curve is invalid above 5 detections.

These curves show that the lunar FPB is a strong function of both the number of detections in a shot, and the average detections per shot. For example, with 5 detections in a shot, the bias could be as low as 7 ps for an average detection rate of 0.13 det/shot, and as high as 18 ps for an average rate of 2.03 det/shot.

However, since the FPB is largely independent of the temporal PDF at all reasonable

intensities (as shown earlier), then FPB should also be largely independent of time PDF with speckle intensity variations, because each shot is at *some* intensity. To confirm that the FPB remains independent of the time PDF with speckle illumination variations, I simulated those conditions. In these discussions, it is important to distinguish between the photon arrival time PDF, and the shot-to-shot illumination PDF.

If there are any differences in bias between the different time distributions, they will be most evident at the highest photon rate. Therefore, I compared the three different distributions (gaussian, triangle, and uniform) at α = .25 photons/shot, which gives a very high average detection rate of 2.0 detections/shot. Figure 62 shows the three simulated curves (1 million shots/curve).

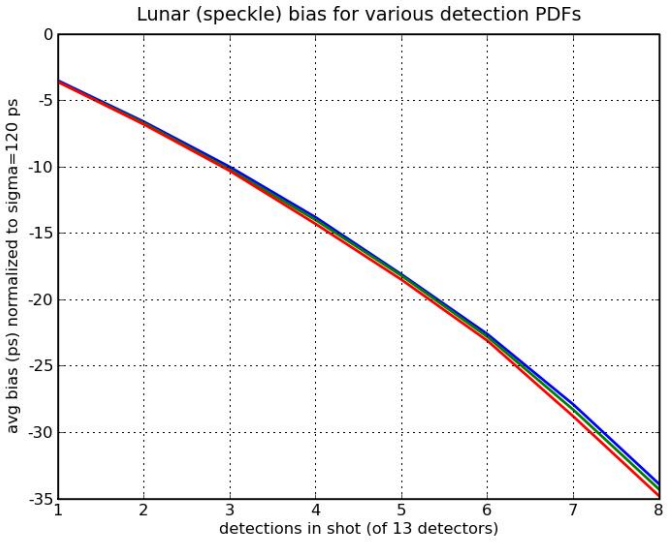


Figure 62: First Photon Bias for 3 compact arrival time PDFs,
for very high average detection rate of 2.0 det/shot.

We see that even at the highest detection count, there is at most only slightly over 1 ps difference between the distributions. As expected, the difference is insignificant.

3.4 First Photon Bias for APD Distributions

We now consider FPB for our actual system, including the long APD tails.

FPB Windowing

When doing the FPB analysis, the acceptance window size matters because of the broad distribution: the full tail drags out σ to 9 ns! If all the tail data points were used for estimating RTT, then this huge σ would probably contribute to FPB. However, when computing NPs, we deliberately avoid responding to the tails, which means the algorithm for FPB correction must similarly not respond to unrepresentative photons. Thus the FPB window should be close to the effective "window", i.e. should closely contain the group of raw measurements that contribute significantly to the final normal point. Although different window sizes produce nearly the same table of FPB when normalized to σ = 120 ps, bigger windows result in bigger sigmas, which result in bigger FPB corrections.

In the simulation to determine the FPB correction table, I therefore window the detectable photons to a range appropriate for FIDs, and another for LUNs. Only detectable photons within this window are included in the bias, and only final detections within the window are included in the bias statistics. Varying this window size produces fairly small changes in the estimated FPB, which means the exact choice is not critical. However, it must at least roughly match the actual detection peak width for reasonable results.

FPB for Fiducial Detections

The FPB is directly proportional to the standard deviation of the photons' arrival time. Therefore, for FIDs, only the laser pulse width part of the fiducial deviation counts; the rest of the electronics variation are downstream of the APDs, and so don't contribute to increasing the FPB. The algorithm estimates the photon arrival deviation by subtracting in quadrature an estimate of the downstream variations from the total fiducial detection deviation.

Our fiducials have a quite uniform shot-to-shot intensity, and the detectors are essentially independent. This is supported by measured fiducial detections, which closely follow binomial statistics [TMurphy, private communication]. Therefore, the number of detections per shot is of no significance; it simply reflects the random binomial statistics of 13 detectors, each of which may or

may not detect a photon on a given shot. While the FID illumination is quite stable within a run (when the laser is working well), it changes over longer times with laser power, and operator adjustments. The algorithm allows for these variations, since it uses the entire run's average fiducial detection rate to estimate a constant fiducial intensity for that run.

The fiducial simulation used 1 million shots over 13 detectors to measure FPB, using the full APD detection model, with a core σ of 120 ps, limiting detections to within a 1 ns window, starting at -3σ of the photon arrival times. The 1 ns easily covers our fiducial distribution peak. As noted, much larger windows yield unrealistic values of bias.

Since the bias scales linearly with photon arrival deviation, and our fiducial deviation sometimes changes over long times, for example due to laser drift, the correction algorithm estimates the arrival deviation, and scales the bias value accordingly (more below). Figure 63 shows the fiducial bias vs. detection rate [fidbias.bat]:

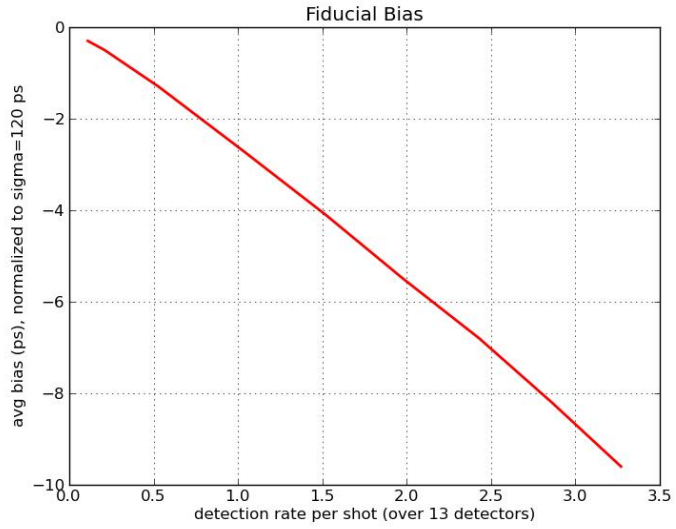


Figure 63: Fiducial bias for various average detection rates, normalized to arrival σ = 120 ps.

We typically operate at a *total* fiducial detection rate around 1.5 detections/shot. However, recall that FIDs are alternately (a) diffused and (b) the corner-cube imaged, onto the APD array. We use only the diffuse fiducials, because they mimic the spatial distribution of lunar detections on the

APD elements. The diffuse FIDs are stronger, at around 2.2 det/shot, with a bias around 6 ps. With well-controlled FID detection rates, we could assume this bias is always 6 ps. However, there are variations in laser brightness, and possibly other unexpected variations. It is easy to measure the diffuse FID rate, so we use the measured rate for each run.

The bias in a detection has some uncertainty, which adds in quadrature to the uncertainty of a single RTT measurement. The bias, deviations in bias, and uncertainties of the average bias $\left(= deviation / \sqrt{n}\right)$ are given in the following table, from a simulation of 10^6 shots [fidbias.bat]:

detectable photon rate (α)	**detection rate (= $n\mu$ = 13μ)**	**bias (ps)**	**bias deviation (ps)**	**uncertainty of avg (ps)**
0.01	0.13	-0.3	5.7	.016
0.02	0.26	-0.6	8.7	.017
0.05	0.63	-1.4	13.4	.017
0.10	1.2	-2.7	18.8	.017
0.15	1.8	-4.1	22.9	.017
0.20	2.4	-5.4	26.3	.017
0.25	2.9	-6.8	29.3	.017

Figure 64: First Photon Bias simulation results.

These show that even at the highest rates, the variations added per measurement are small (< 30 ps), across the overly wide range of 1 to 8 detections/shot. Given the FID detection standard deviation of ~120 ps, this is already small, and ultimately made even less significant when the lunar deviation is included. The uncertainty in the *average* bias from the simulation, which would produce a persistent systematic error, is totally negligible (<< 1 ps). Note that at higher detection rates, the *individual* uncertainty on a simulated detection time increases, due to the more-variable number of detectable photons. However, the uncertainty of the *average* compensates this increased individual uncertainty by having more detections to average over. This coincidentally results in a nearly constant uncertainty of the average fiducial bias.

While we currently have no quantitative data on our speckle model compared to actual data, I think it is likely that speckle modeling errors vastly outweigh the 17 femtosecond random error of the simulated average bias, especially at higher detection rates.

Fiducial FPB Correction Algorithm

While a straight-line fit to the above graph would yield sub-ps accuracy, I used a table-lookup with linear interpolation, which is very flexible. This is similar to the algorithm for the lunar FPB, with negligible implementation cost. The algorithm comprises these steps:

1. Compute the standard deviation of the fiducial detections, relative to the Fast Photo-Diode (FPD).
2. Subtract in quadrature the known deviations of the downstream devices that add variation to the fiducial detections: the FPD itself, the APD, the TDC. This yields the arrival time deviation, σ.
3. Compute the fiducial detection rate for the diffuse FIDs (which are used in the NP). Use the number of working detectors on this run to scale the rate to the equivalent for 13 working detectors.
4. Look up the bias as a function of detection-rate for a 120ps deviation in the table (given above), with linear interpolation of the detection rate.
5. Scale the table value to the estimated fiducial deviation, by multiplying by $\sigma/120$ ps.

To test the general principle, I used two sets of simulations: first, where FPB was not allowed in the simulation, and second, where only FID FPB was included. Rather than compare analyses to the known RTT, which may be affected systematically by the whole gamut of analysis impairments, I examined the *change* in RTT due to FID FPB, and how much of that was restored by the FID FPB correction.

Over 100 runs of 20k shots each, with no background, FID FPB increased the RTT by 5 and 4 ps, for AC and PDF-fit methods, respectively. In both cases, the correction algorithm restored the average RTT to within 1 ps of the unbiased simulation. The random error over 100 runs is ~0.3 ps,

but the systematic errors in either data reduction method are ~ 3 ps.

3.5 The Lunar Case: Notably Harder

First photon bias correction is potentially harder for the lunar returns due to four effects: (1) speckle intensity fluctuation; (2) wider time distribution due to reflector tilt, (3) spatial variation of intensity (variation across detectors), and (4) more background noise. We now examine these issues, starting with the most significant: speckle.

Speckle Intensity Variations and FPB

As described earlier, with speckle, the return intensity varies widely from shot to shot. Also, the first photon bias is significantly dependent on the intensity distribution. Simulation confirms these dependencies.

Our model of speckle shot-shot intensity distribution comes from a simulation from Tom Murphy [private communication]. The simulation produces a series of far-field diffraction patterns. A sample of the central areas of these patterns yields a list of 32490 intensity values. The probability of occurrence drops off rapidly with intensity. When normalized to unit intensity:

$$\sigma = \frac{\sigma_{unnormalized}}{\langle I \rangle} = \frac{647.145}{322.996} = 2.00$$

This ratio of σ to average looks suspiciously like a simulation artifact. It could be that the simulation algorithm somehow produces this near-integer ratio. This is a question for possible future study.

We have already seen that with speckle intensity fluctuations, the FPB varies widely from shot to shot, and we must use information about the number of detections in the shot to make an effective correction. In addition, the average intensity underlying the speckle is also important to interpreting the number of detections in a shot, and correcting for FPB.

As discussed earlier, if you include a lot of the tail in the FPB estimate (> a few ns), it skews the average detection time and bias tremendously, and the tail has an unrealistic impact on the FPB. To accurately compute lunar FPB, we must window to some arrival time interval, and compute FPB

only for detections within the window. This provides better results than using a gaussian distribution to approximate our detections, but requires the choice of a window. However, the FPB, *when normalized to* σ = 120 ps, is fairly independent of our acceptance window. Changing the window from 2.0 ns to 1.5 ns does not measurably change the FPB (only 1 ps at the highest rate of 2.0 det/shot, and 8 detections in the shot). This confirms that the FPB is largely independent of the shape of the distribution, even for the realistic APD PDF. The reason a larger window increases the bias estimate is because it increases the arrival time σ. We use a window of 1.5 ns, which covers the lunar spread during maximum reflector tilt. Figure 65 shows the FPB vs. number of detections in the shot [lunbias.bat]. This simulation used 10 million shots, and detectable photon rates per detector (α) of .01, .02, .05, .10, .15, .20, .25 photons/detector/shot. The α = .01 (blue) curve is invalid above 5 detections, due to the small sample size that results from such a low photon rate.

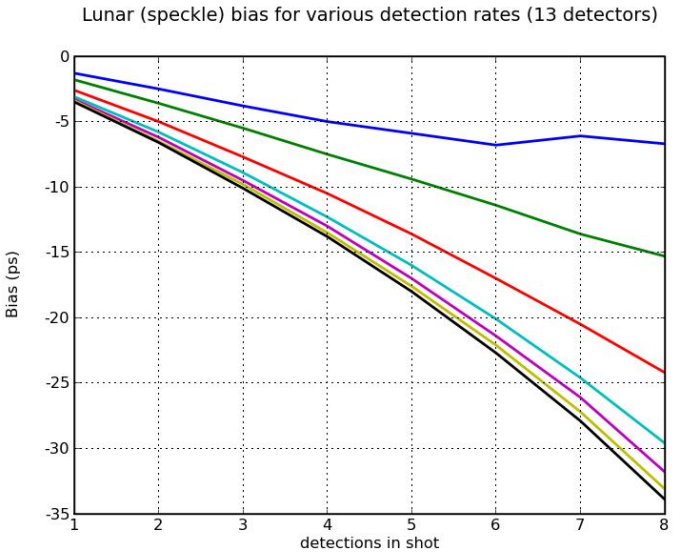


Figure 65: Lunar First Photon Bias vs. # detections in shot, normalized to σ = 120 ps, for average detection rates of 0.11 (blue), 0.21 (green), 0.49 (red), 0.90 (cyan), 1.24 (magenta), 1.55 (yellow), and 1.82 (black) det/shot, 1 arcsec FWHM spot size. The blue curve is invalid above 5 detections (see text).

There is some evidence that our speckle model under-estimates the number of high-detection-count shots (see chapter 4). I suspect that for data reduction, we will want to limit the included shots to only those with less than 3 or 4 detections, to control uncertainties at higher intensities. Such a

restriction still includes the vast majority of photons in any run.

Estimating Photon Flux From Recent History

In APOLLO, the underlying lunar signal intensity changes slowly, on the order of many seconds, due to changes in observing conditions such as clouds, seeing, and telescope tracking drift. Therefore, for FPB correction, we make an estimate of the underlying intensity *for each shot* based on the number of detections in shots nearby in time (both past and future). The correction algorithm uses this underlying average detection rate for the shot, along with the number of detections in the shot, to estimate the FPB for that shot. The two-dimensional FPB table, indexed by average detection rate, and number of detections in the shot, was generated by simulation at a set of fixed average detection rates. The correction algorithm interpolates between the simulated detection rates to provide a bias for any rate.

We now show that a simple flat average of the surrounding 4-second interval provides a good estimate of average intensity. We believe that 4 s is a reasonably short time compared to the timescale of intensity drift. Simulation shows that at a detection rate of 1.05 det/shot (over 13 detectors, center weighted spot shape), the standard deviation of the number of detections per shot is σ = 1.9 det/shot. (This is not binomial because the speckle intensity fluctuations cause correlations across detectors.) Averaging detection counts over 81 shots (4 s) has a standard deviation of:

$$Dev(rate) \equiv \sigma_{\mu} = \frac{1.9}{\sqrt{81}} = 0.21 \text{ dets/shot}$$

We propagate this uncertainty to the FPB correction table, using the standard formula for error propagation of an arbitrary function:

$$Dev\left(f(x)\right) = \left|f'(x)\right| Dev\left(x\right)$$

The FPB is a function of two variables: detection rate over the working detectors, μ, and number of detections in the shot, *ndet*. Bias is in picoseconds, and μ in det/shot, so the derivative $db/d\mu$ has somewhat odd-looking units of ps/(det/shot). At an average of 1.05 det/shot, for several different number of detections ($b \equiv$ bias), we compute the uncertainty due to estimation error in the

local detection rate:

$$\sigma_{bias} \approx \frac{db}{d\mu}\sigma_{\mu} = \frac{db}{d\mu}\left(0.21 \text{ det/shot}\right)$$

$$ndet = 1 \text{ or } 2: \quad \frac{db}{d\mu} \approx 1 \text{ ps/(det/shot)} \quad \Rightarrow \quad \sigma_{bias} = 0.2 \text{ ps}$$

$$ndet = 3 \text{ or } 4: \quad \frac{db}{d\mu} \approx 2 \text{ ps/(det/shot)} \quad \Rightarrow \quad \sigma_{bias} = 0.4 \text{ ps}$$

$$ndet = 5 \text{ or } 6: \quad \frac{db}{d\mu} \approx 3 \text{ ps/(det/shot)} \quad \Rightarrow \quad \sigma_{bias} = 0.6 \text{ ps}$$

Higher detection counts occur too infrequently to matter. Thus the uncertainty in the detection rate for FPB correction is completely negligible compared to other uncertainties in the system.

Retroreflector Tilt and FPB

As described earlier, reflector tilt from lunar libration spreads out the return pulse over time. The worst-case tilt effect is from the largest array, Apollo 15, ~1 m wide east-west. At maximum libration of ~9^o × 6^o (east-west × north-south, called x × y from now on), the reflector adds in quadrature about 320 ps of deviation to the return times. Since the first photon bias is directly proportional to the time-spread of the arriving photons, and libration spreads the arriving photons in time, the lunar photons have a bias that increases with the reflector tilt angle. Therefore, to estimate the lunar first photon bias, we must know the arrival time deviation. One way to estimate this is directly from the RTT fit residuals. (Another method might include the known reflector tilt angle, and incorporate its effect into the time-distribution of lunar photons, similar to the information used to generate the lunar model time PDF.)

The preliminary investigation of compact distributions found that the time PDF shape has a negligible effect on the FPB, even over radically different PDF shapes. Therefore, we expect that the details of the reflector tilt will not be important to FPB, especially since the PDF shape is always similar, comprising a main peak with smaller tails on the edges. Therefore, we can estimate of the lunar detection σ from the overall RTT residuals, without regard for their origin or precise shape, and

subtract off the known downstream variations.

Spatial Variation of Illumination and FPB

For lunar returns, the detectors are not perfectly uniformly illuminated: the image of the retroreflector array is approximately a gaussian spot, i.e. the intensity at radius r is proportional to $\exp(-r^2/2\sigma^2)$. The square APD array captures about 1.4 arcsec on a side of the lunar image. "Seeing," the long-time (seconds) working resolution of the telescope through the atmosphere, at Apache Point Observatory (APO) is typically 1-2 arcsec, full-width half-maximum (FWHM). The instantaneous seeing is somewhat smaller. The seeing recorded for APOLLO's highest returns was 1.2 arcsec long-time, and we estimate about 30% smaller instantaneous seeing. APOLLO's acceptance window for signal is ~2 ns, so instantaneous seeing is relevant. We convert between the radial σ and FWHM using:

$$\exp\left(-\frac{(FWHM/2)^2}{2\sigma^2}\right)=\frac{1}{2} \qquad \Rightarrow \qquad FWHM=\sqrt{8\ln 2}\,\sigma=2.355\sigma$$

Does this spatial variation affect our FPB correction? The simulations below show that this is not a significant consideration in FPB correction.

I simulated the APD array, with its (at most) 13 working detectors. I compared 3 different lunar return spot sizes, covering cases believed to be worst at the time: (1) uniform illumination; (2) 1 arcsec, centered between the 4 central APDs in the array; and (3) 1 arcsec, centered on APD element 11, which has the most working elements around it [lunbias.bat]. The results indicate that even across this wide range, the spot shape is not a significant factor. More realistic conditions, such as limiting to 4 det/shot, and the significantly more limited range of spot behavior, will yield even better results than simulated. In particular, perfectly uniform illumination doesn't happen, even during bad seeing. And even during the best seeing, the center of the spot moves around the lenslet array randomly from shot to shot, and is essentially never perfectly centered in one element. I also used a higher detection rate than for the rest of this dissertation, again to provide a wider margin of confidence. The simulation results are these:

spot shape	detectable photon rate (α)	detection rate (= $n\mu$ = 13μ)	avg bias (ps) (σ = 120 ps)
uniform	0.1	0.93	-9.4
centered on array	0.1	0.91	-10.4
centered on APD 11	0.1	0.9	-11.2
uniform	0.25	1.90	-19.7
centered on array	0.25	1.84	-20.6
centered on APD 11	0.25	1.79	-21.7

Figure 66: First Photon Bias results for various spot conditions.

These are normalized to 120 ps arrival σ. With maximum reflector tilt, these would be multiplied by about 3. For the highest rate of almost 2 det/shot, there is a difference of only ± 3 ps from the centered spot.

Background Detections and FPB

Our highest background rate is ~0.75 det/shot, uniformly distributed over the entire 100 ns TDC detection range. Within the accepted ~2 ns window, this is only 0.015 det/shot. Using the slopes of the bias vs. det-rate curves, as we did for computing the uncertainty due to intensity fluctuations, we find that even at ndet = 6, the slope is 3 ps/(det/shot), the worst-case background contributes less than 0.05 ps of bias, which is completely negligible.

3.6 Lunar FPB Correction Algorithm

The FPB correction algorithm corrects each RTT measurement (i.e., each valid lunar detection) with the average correction (specifically *not* most-likely). This makes the correction algorithm essentially independent of the data reduction algorithm, which is desirable, because the data reduction algorithm may change occasionally.

The algorithm to estimate and correct for FPB comprises these high-level steps (more detail follows):

1. Compute s_{at}, the estimate of the *arrival time σ* (excludes APD and downstream contributions).

2. For *each shot*, estimate the average detection rate (in det/shot) over a few-second interval around the shot, and scaled to the reference level of 13 working detectors.
3. For each lunar detection, look up in the table for its number of detections, and interpolate for its intensity, to find the bias at σ = 120 ps.
4. Scale the correction to s_{at}, the estimated arrival-time σ.

We detail each step below.

Estimate the arrival time σ

The data analyzer finds a reasonable window around the prediction, and estimates a first-fit to the data (typically a straight-line fit), to adjust for prediction errors. After such adjustment, the FPB algorithm computes s_{res}, the residual error estimate from the tentative fit. This residual is computed within an "FPB-window," which is chosen based on the retroreflector tilt and the fiducial deviation, to include those detections which are most significant in determining the NP. This determination is fairly insensitive to the NP generation algorithm, since any effective algorithm will use largely the same set of detections. I chose the same window size for FPB as that used for the Augmented Calculation (AC) NP algorithm, since the windows serve similar purposes. (Reiterating, the window width is set to $6\sigma_{fiducial}$ + foot-to-foot width of the reflector trapezoid.) Finally, the FPB algorithm finds the estimated arrival-time σ by quadrature subtracting the downstream-variations due to fast photodiode (FPD, ~55 ps), APD variations (core σ + parabola + tail), and TDC variations (~13 ps):

$$s_{at} = \sqrt{s_{res}^{\,2} - s_{FPD}^{\,2} - s_{APD}^{\,2} - s_{TDC}^{\,2}}$$

Note that the FPB window is used only for estimating s_{at}, but then all detections, in or out of the window, are corrected for FPB.

Estimate the speckle-averaged intensity

For each shot, the algorithm computes a rolling average of the # detections in nearby shots, using a window of n = 81 shots (± 40 shots). For shot j,

$$I_j = \frac{1}{n} \sum_{i=-(n-1)/2}^{+(n-1)/2} ndet_{j+i} \qquad where \quad ndet_i \equiv \#\text{ detections in shot } i$$

Near the beginning and end of the run, the window is truncated to the available shots.

Look up FPB correction in table

The FPB correction table is a 2D table indexed by average detection rate (intensity) and # detections in this shot. The algorithm uses the estimated intensity to linearly interpolate between the intensity values in the table. The table is normalized to an arrival σ = 120 ps.

The table is generated from the APD simulations, with a 1 arcsec spot centered in the APD array, a 1.5 ns acceptance window, and a fixed tilt of half of the maximum (4.5° in *x*, and 3° in *y*). It is normalized to σ = 120 ps, and looks similar to this:

det-rate μ	ndet=1	2	3	4	5	6	7	8
0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.13	-1.0	-2.1	-3.3	-4.1	-6.2	-6.5	-7.7	-9.0
0.58	-2.2	-4.3	-6.6	-9.1	-12.0	-15.4	-18.4	-22.4
1.05	-2.6	-5.1	-7.7	-10.6	-13.9	-17.5	-21.4	-25.9
1.45	-2.8	-5.4	-8.2	-11.3	-14.7	-18.5	-23.1	-27.9
1.80	-2.9	-5.5	-8.4	-11.7	-15.2	-19.2	-23.4	-28.9
2.10	-3.0	-5.7	-8.5	-11.8	-15.4	-19.6	-24.2	-29.5

Figure 67: Sample First Photon Bias correction table.

The correction table varies slightly depending on tilt, and the choice of half-maximum tilt is a reasonable compromise. In general, high tilts have higher arrival deviation, and so errors in the table are magnified more than those at low tilt. Also, higher tilts are more common than low tilts, since the librations are roughly sinusoidal, and spend more time near their extrema. These factors favor the FPB table be made with some tilt, for the best overall results.

Scale to the arrival-time σ

The table of FPB biases is scaled to an arrival sigma of 120 ps. FPB is linear with arrival-

time sigma (which was estimated earlier), therefore the algorithm multiplies the table value of FPB by (s_{at} / 120 ps):

$$estimated\text{-}bias = (table\text{-}entry)\frac{s_{at}}{120\ ps}$$

For each shot, the algorithm subtracts the estimated FPB from each detection in the shot. There is no way to distinguish between detections within a shot, so they all get the same adjustment. Note that all FPBs are negative, so this subtraction of a negative number makes the arrival time *later*.

Uncertainty in First Photon Bias Correction

At the rarely-seen high average-detection rate of ~2 det/shot, the standard deviation in FPB for any of 1 through 7 dets/shot <~ 20 ps (for arrival σ = 120 ps, or no reflector tilt), which is small compared to most other uncertainties in the system. For example:

TDC: ~13 ps

FPB: ~20 ps

APD: ~25 ps

FPD: ~55 ps

Laser pulse: ~95 ps (should be ~ 40 ps, according to specification)

Reflector tilt: 0 - 321 ps

In the case of no retroreflector tilt, and an ideal-performing laser, the uncertainty in FPB degrades the overall uncertainty per shot negligibly, from 74 ps to 76 ps. At higher reflector tilts, the FPB contribution to uncertainty increases to ~60 ps, but the return distribution is also wider by roughly the same factor. Therefore, the uncertainty degradation is still negligible. More realistically, with fewer detections per shot (less FPB), and our actual laser performance, the degradation is even less.

It is fortunate that FPB correction has negligible effect on the overall uncertainty of our data. This is not a priori insured. With different system parameters, it might have been that FPB was a significant detriment to our uncertainty.

3.7 Full System Simulation of FPB

To measure the performance of the lunar FPB correction algorithm, as with the fiducial FPB algorithm, we create two simulation sets and perform three analyses, for each set of conditions under test. First, we create a set of simulations with no FPB, and analyze them with no FPB correction, to establish a baseline performance (average NP over the runs). This baseline isolates the FPB performance from other analysis effects, which might confuse the FPB issue. Second, we create a set of simulations with only lunar FPB, but analyze it *without* FPB correction to see the effect of FPB on the NPs. Finally, we analyze the FPB simulations *with* FPB correction, to see how close it gets to the original baseline.

As usual, we use 100 simulations for each set of conditions. The simulations use 5000 shots (250 s), a high detection rate of ~1 det/shot, with an 8.64 ns/day slip relative to the prediction, starting with a 1 ns prediction offset. The simulations also include a slow, random drift in signal intensity over time, with a time scale of several seconds.

Our first sets of conditions exclude background, as verification of the method and implementation of FPB correction (calibration simulations). We use three different reflector tilt settings: no tilt, medium tilt (tiltx = 4.5^o, tilty = 3^o), and maximum tilt (tiltx = 9^o, tilty = 6^o). If these work acceptably, then intermediate tilts are likely to work, as well. The biases given below are the average of the 100 simulations under the given conditions. The table lists the biases, corrections, and residual bias for each set of conditions, and NP generation method.

	AC (ps)			PDF-fit (ps)		
Conditions	**Avg FPB**	**Avg Correction**	**Avg residual FPB**	**Avg FPB**	**Avg Correction**	**Avg residual FPB**
no tilt, no background	12	10	2	11	11	0
half-tilt, no background	22	18	4	22	17	5
max tilt, no background	40	33	7	34	32	2
no tilt, full background	13	10	3	11	10	1
half-tilt, full background	22	16	6	22	17	5
max tilt, full background	38	30	8	33	30	3

Figure 68: First Photon Bias correction performance on simulated runs.

The FPBs and corrections in the table are magnitudes, as all FPBs are negative. All random uncertainties in the table are < 1 ps. The AC residual FPB increases monotonically with bias itself, whereas PDF-fit residual FPB does not. All the corrections are slightly too small, which is consistent with a general indication that the auto-window size may be slightly too small.

The table above also shows reasonable consistency, at medium tilt and below, between the FPB of both NP generation methods. It also shows that there is no significant effect of background detections (thermal and full-moon stray photons), and 1% prepulses.

For the AC method, the residual bias is slightly over 1 mm at maximum tilt, which is also where the AC method has its largest systematic error, in general (as noted in chapter 0). Recall, though, that the testing method separates the FPB correction from other errors, so any systematic errors from other causes should not significantly affect the FPB measurement.

At smaller tilts, the AC residual biases are below 1 mm.

For the PDF-fit method, all the residual biases are below 1 mm.

Not shown in the table is that there is a very small *decrease* in the NP variation with FPB correction. The decrease is < 1 ps, which is insignificant on one batch of 100 runs, but fairly

consistent across the 12 sets of conditions and methods. At first, this may seem surprising. The FPB bias correction, for any given shot, is only an estimate of the bias for that shot, and it includes some uncertainty. If the corrections were uncorrelated to the individual RTTs, the FPB correction could only increase the variability of the NP results. However, there is temporal locality to the brightness of the return signal, and the FPB algorithm makes use of this. Essentially, it knows when a run is bright, and preferentially corrects those shots that most need correcting. I suspect that the decreased variation is due to the analysis' knowledge of "bright" vs. "dim" periods in a run, and overall brighter vs. dimmer runs, returning the bright detections closer to the actual RTT value, and thus reducing the overall RTT range.

3.8 FPB Conclusions

FPB correction works well up to 8 detections in a shot, subject to the validity of the speckle model. Residual bias is under 1 mm for all conditions of 5000 shot runs. Different run sizes should not affect the residual bias, but longer runs should yield better estimates of s_{at} that reduce uncertainty in FPB correction. Realistic simulations indicated that residual bias with the given FPB algorithm, for PDF-fit method, is < 0.5 mm at small and large tilts, and somewhat larger at intermediate tilts. There is no increase in NP uncertainty compared to an idealized system with no FPB. In fact, with realistically slow intensity drift, FPB correction appears to slightly *reduce* the uncorrected NP uncertainty, by adjusting upward those RTTs which are most likely to be biased low (the bright shots). A more physically based choice of window size might improve the results, as the current choice is somewhat ad hoc.

As of 6/2010, JPL claims APOLLO residuals to their solar system model are 16 mm, which indicates that further improvements to FPB correction, if needed at all, are not useful until this residual error problem is resolved.

As the number of detections in a shot increases, the quality of our speckle intensity model becomes more critical. Chapter 4 compares some real runs to our model. The NP generation

algorithms implement a "threshold" of detections in a shot, above which we discard the shot, to limit biases. Chapter 4 discusses this, as well.

3.9 Possible Areas for Future Study of First Photon Bias Correction

Acceptance window: It appears that a slightly larger acceptance window would be better. The current window size algorithm is somewhat ad hoc. It could be more physically based, by including a quadrature summed term representing the spread of the PDF tail. This term would need "tuning," as there is no obvious way to measure it exactly. A wider window would increase the estimated arrival σ, and increase the correction magnitudes. The real issue, though, is would it smooth out the residual bias, to make it more consistent across reflector tilt?

Number of APDs working: Is it worth correcting not only the detection rate, μ, for number of working APDs (compared to the simulation of 13), but also scaling the number of detections for each shot, and interpolating that value from the FPB table, as well? This might also eliminate the slight under-correction of the current algorithm. Alternatively, we could easily include a separate table for each of 10, 11, or 12 working APDs.

Speckle model: Investigate the speckle model for high-detection shots, by comparing with real data. See if we can improve the model, and therefore the FPB correction. Also, see if we can simplify the implementation by forming a mathematical model that facilitates simulation, without the need for a massive simulation data list. Chapter 4 discusses some early attempts at this.

Effect of "Threshold:" Excluding shots of high detection count, say > ~ 4 detections, may avoid some errors due to a mismatch between our speckle model and actual data for high brightness shots.

Fiducial prepulses: We do not have a direct measure of the fiducial pre-pulse rate. It may be high, causing us to underestimate the effective fiducial detection rate. A higher intensity increases the FID FPB for those shots that survive pre-pulses. This may cause our FID FPB correction to be systematically too small. To the extent that the pre-pulses are consistent, this systematic error would

be a constant offset to RTT, and would be fitted out in the solar system model fit.

Speckle-averaged intensity: The current intensity estimate for each shot is an unweighted rolling average of detections in nearby shots. An enhancement might be to use a *weighted* average of nearby shots, to reflect the lesser correlation of shots further away in time. In other words, use a finite impulse response (FIR) filter to estimate the underlying intensity:

$$I_j = \frac{\sum_{i=-(n-1)/2}^{+(n-1)/2} w_i ndet_{j+i}}{\sum_i w_i}$$

We might determine the filter coefficients from the autocorrelation function of the number of detections per lunar shot, i.e. the correlation coefficients between the number of detections at shots separated by *i*. We could probably find an average set of weights based on actual data.

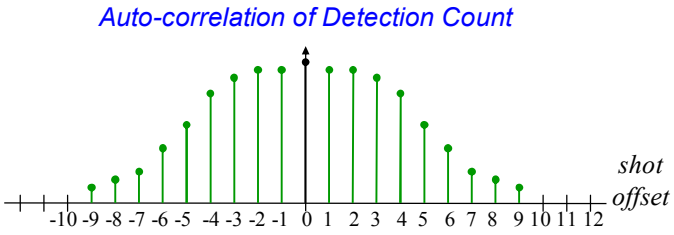


Figure 69: Hypothetical auto-correlation function of detection counts across a run

A relevant question is "How much does the auto-correlation function vary from night to night (e.g., due to weather)?"

Investigate alternative PDF approach: Is there a benefit to creating an FPB-modified PDF numerically, by stepping through the histogram bins, and incrementally reducing subsequent bins by the probability they are preempted by an earlier photon? Note that current real data shows noticeable lack-of-fit in the lunar histogram, which may be due to per-channel timing skews between fiducial and lunar detections. This lack-of-fit may need to be overcome before a more refined FPB-modified lunar PDF can be attempted..

4 APOLLO Data Reduction Conclusions

This chapter summarizes the final results, conclusions, and recommendations of my research on APOLLO. It includes the following:

- Results of reduction of real data from 2006-2009
- Data Reduction Methods
- First Photon Bias, speckle model, and limit on number of detections in a shot
- Spot size, and telescope focus recommendation
- Possible areas of future work, including some anomalies in our results
- Discussion of APOLLO data and solar-system model error analysis

4.1 Reduction of Real Data

All APOLLO data to date have been reduced with the correlation method (described briefly in Chapter 2). I refer to these reductions as "existing" normal points (NPs). For comparison, I have used the newer methods of the previous chapters to reduce our real data from the beginning (mid-2006) through mid-2009. Using default parameters for all reductions, but forcing the NP launch time to match the existing times, the PDF-fit method produced 498 NPs which were directly comparable to existing NPs. My PDF-fit method cannot currently reduce double-pulse laser shots (though it can be adapted to them with the addition of "echo" parameters), so some of the existing NPs around the middle of 2006 are excluded.

We can compare the results of the PDF-fit data reduction method to the existing NPs. JPL provided us with the residuals of our existing reductions to their model, after they fit their model parameters to our data, simultaneously with many other measurements. We can compute residuals to the PDF-fit reductions by offsetting the JPL-supplied residuals by the difference in NP RTTs between our existing results and the new PDF-fit reductions. This overstates the new residuals since JPL did not fit to the new reductions. Figure 70 shows the existing and new residuals.

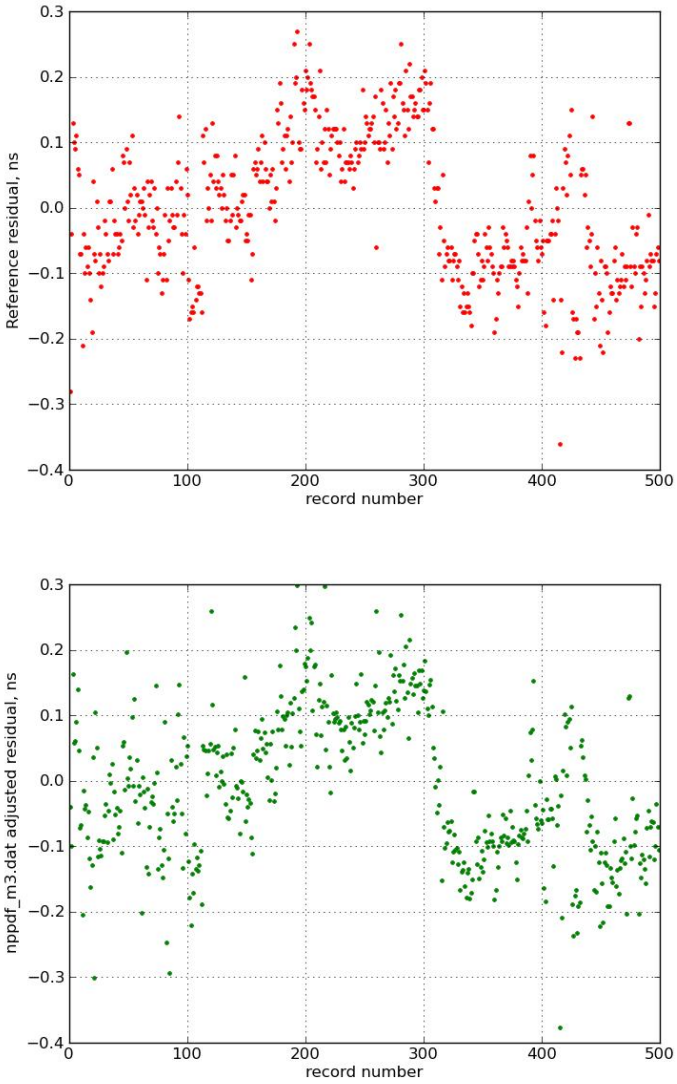


Figure 70: Existing residuals (red) and PDF-fit residuals (green).

The new data points follow the same general pattern as the old. The 642 original NPs have an unweighted average uncertainty of 27 ps. The residuals from JPL have an unweighted average of –1 ps, which is quite small since JPL fits for a "range bias" which forces the residuals to be nearly zero. The standard deviation of the residuals is 108 ps.

The new (but not newly-fit) NPs have an average uncertainty of 20 ps, an average adjusted-residual of –10 ps, with a standard deviation of 113 ps, essentially the same as the original. I suspect

the old NPs are biased somewhat late, because the correlation method does not account for the wider spread of lunar returns (due to reflector tilt), and this causes the maximum correlation point of the lunar histogram to be pushed outward, where the main-peak overlaps the fiducial tail.

Figure 71 shows the reduced-χ^2 parameter for the lunar fits. Some of the χ^2 indicate serious fit errors. These can probably be mostly corrected with a slightly improved algorithm for choosing the initial fit-parameter guesses. For example, the initial average and standard deviation can be computed from the sample of RTTs that are being fit using the standard closed-form formulas.

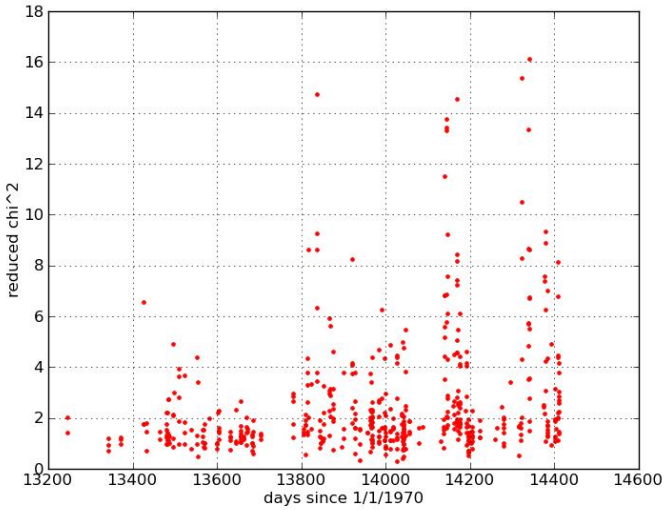


Figure 71: Reduced-χ^2 parameter for the lunar fits of real data.

Figure 72, Figure 73, and Figure 74 show example fits to lunar returns for rates of ~0.05, ~0.51, and ~1.0 det/shot (note the different vertical scales).

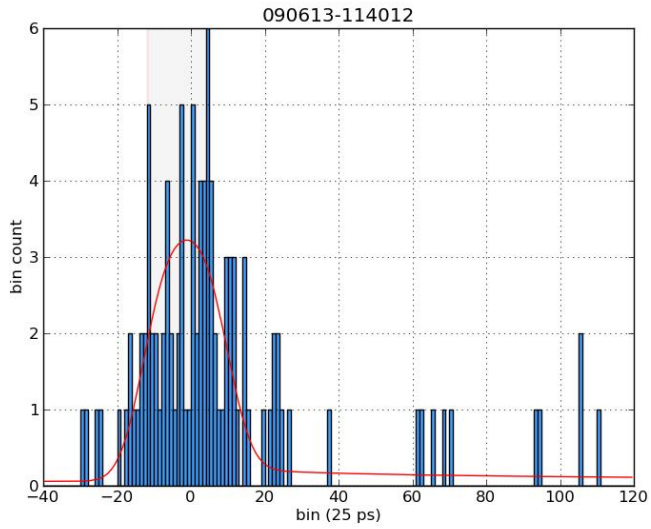


Figure 72: Sample lunar return fit for a return rate of ~0.05 det/shot.

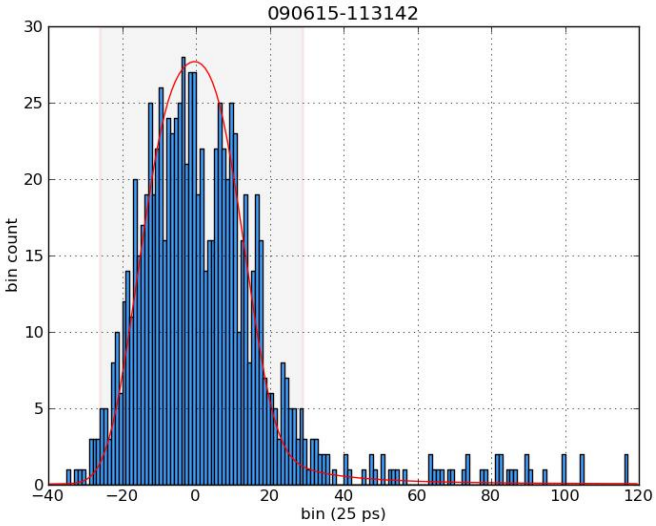


Figure 73: Sample lunar return fit for a return rate of ~0.51 det/shot.

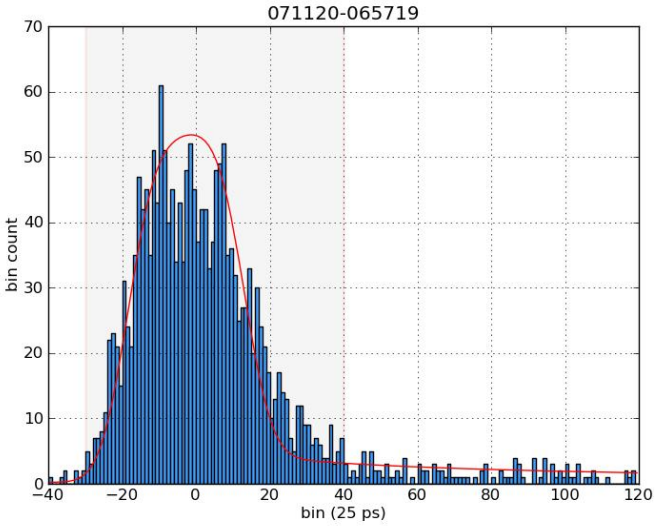


Figure 74: Sample lunar return fit for a return rate of ~1.0 det/shot.

The lack of fit in the high-rate (last) run may be due to some spreading of the calibration times of the fiducial channels, possibly due to electrical noise from the laser firing. The per-channel spreading is known to occur, but requires further study.

4.2 Data Reduction Methods

The data reduction results are already covered in chapter 2. Reviewing: the best performing data reduction method is the PDF-fit method, by a little bit. The random errors are quite similar between PDF-fit and AC. The main reason for eliminating the AC method is an unexplained bias in the case of high libration.

Does APOLLO meet the 1 mm goal? On many good nights, our *random* error does meet 1 mm (1 σ) when aggregated over the runs of the night. Neglecting the GPS uncertainty of 10 ps that we typically add in quadrature to our NP uncertainty, even some individual runs have random error of 1 mm. However, our systematic errors are dominated by first photon bias, so getting our systematic errors down to 1 mm requires first getting FPB correction to that level (described below).

4.3 First Photon Bias

Chapter 3 describes a workable FPB correction algorithm. The algorithm is shown to work to within 1 mm with a plausible speckle model, as confirmed by simulations using that model. It is flexible, and can easily adapt to other models by changing the lookup table. It is also easily extended to have a separate table for each number of working APDs (10, 11, 12, 13), which is useful because the number of working detectors varies somewhat from night to night.

However, analysis of actual data shows that the intensity variations experienced by APOLLO are different than the current speckle model. This is evident by comparing the histogram of the number of shots with a given number of detections from a high-rate run, with a simulated run. Figure 75 compares the histogram from a run with ~2.16 det/shot, with a simulation having the same average detection rate.

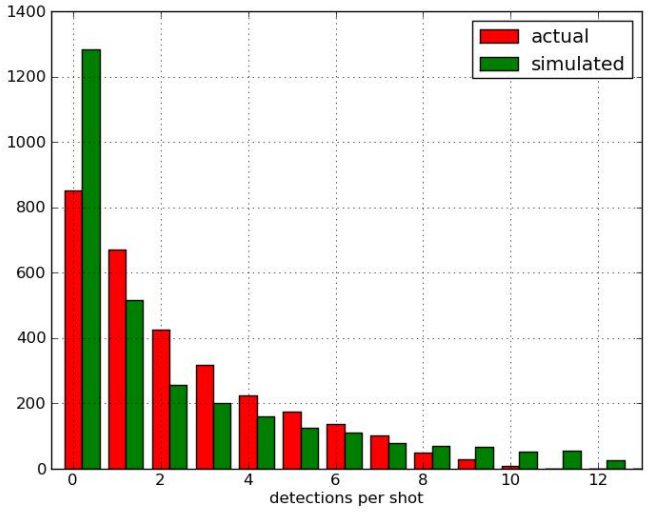


Figure 75: Actual vs. simulated number of shots with given number of detections, showing discrepancies at both the low and high ends.

As discussed below, the FPB correction is too small for our actual data, and the correction algorithm requires a table generated by a speckle model matching the real system. To explore FPB some in our real data, the FPB algorithm was extended beyond that analyzed in Chapter 3. The APOLLO system has a somewhat variable number of working APDs, and the number is generally

different than the 12 used in the earlier run simulations. To analyze real data, the algorithm was extended to interpolate the number of detections in each shot to the equivalent for the number of working detectors used to generate the table (13). In other words,

$$ndet_{effective} = ndet_{actual} \frac{13}{n_{APD}} \qquad where \quad n_{APD} \equiv \text{number of working APDs}$$

and 13 is the number of simulated APDs used to generate the FPB lookup table.

Even with the correction for number of working APDs, the FPB table (from the current speckle model) still under-corrects the actual FPB. We can get an estimate of what the actual FPB is by comparing uncorrected analyses that include shots with multiple detections, against uncorrected analyses that include only shots with exactly one detection. These latter shots have minimal FPB, and are therefore an estimate of the true round-trip time (RTT), probably good to 1 or 2 mm (see FPB graphs in Chapter 3). Figure 76 shows the all-shot (maxdet = 9) FPB corrected results, compared to the uncorrected 1-detection-only shots (maxdet = 1), as a function of data rate.

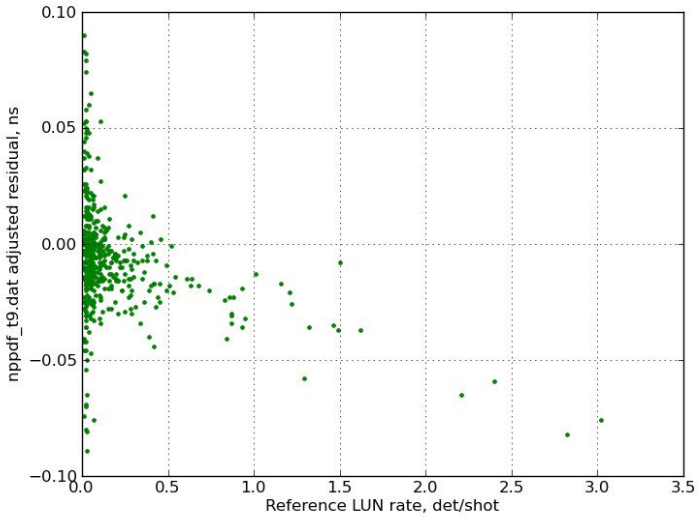


Figure 76: Discrepancies between uncorrected maxdet = 1, and FPB corrected using all shots.

Between 1 and 1.5 det/shot, the bias seems under-corrected by about 25 ps. For the few runs between 2 and 3 det/shot, when including all detections, the FPB appears to be under-corrected by

~60-80 ps.

Note that if FPB correction is working well, we expect a slight *increase* in the RTT residual with detection rate, because even the shots with 1 detection have some bias, and it increases with detection rate. However, with the current speckle model, at the highest rates, this is between 5 and 15 ps (~1 - 2 mm), depending on libration. The real speckle would likely be similar, but Figure 76 shows a much larger residual decrease (bias magnitude increase) with data rate, indicating the FPB correction is not making big enough adjustments.

Detection Per Shot Limit

The high-detection-count (and thus high-bias) shots are certaintly more demanding on the speckle model. These high bias shots also have higher bias-uncertainty as well. Therefore, we must ask: What is the optimum number of detections per shot that we should accept in computing our normal point? Figure 77 shows a plot of *random* uncertainty vs. the "threshold" (aka "maxdet") parameter, averaged over all the runs of our real data.

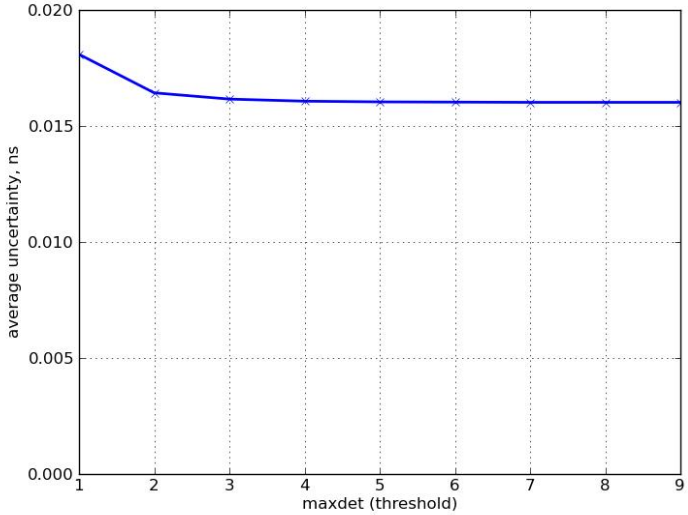


Figure 77: Random uncertainty vs. maxdet, averaged over all runs.

On average, there's not much difference between maxdet=1 and maxdet=3 (~17 ps vs. ~18 ps). However, for our highest return runs, random uncertainty at maxdet=3 is ~2.5 ps, and goes down

to ~1.5 ps when all shots are included. Even 2.5 ps is already well below our 10 ps minimum uncertainty that we currently add in quadrature to account for GPS clock uncertainties. Though in the future, better understanding of the clock may reduce this particular uncertainty contributor, we have many other uncertainties at the several picoseconds level, including atmospheric models, thermal expansion of the telescope mount, and FPB correction systematics. It is unlikely that increasing maxdet beyond 3 will ever produce a noticeable improvement in uncertainty. Therefore, if FPB is well-controlled, maxdet = 3 is probably best.

Applying this limit does indeed improve FPB correction. Figure 78 shows the FPB corrected results, limited to shots with 3 or fewer detections, compared to the uncorrected 1-detection-only shots, as a function of data rate.

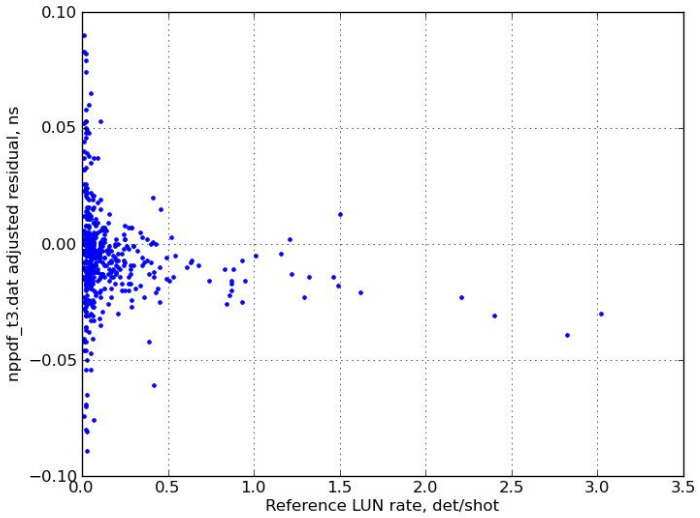


Figure 78: Discrepancies between uncorrected maxdet = 1, and FPB corrected with maxdet=3.

With maxdet = 3, between 1 and 1.5 det/shot, the correction is much better, probably within ~10 ps of the lower-rate runs. For the few runs between 2 and 3 det/shot, the FPB appears to be under-corrected by ~25-30 ps. This is much better than the all-detection NPs, but still something that APOLLO would like to overcome.

A Quick Attempt at a More Accurate Speckle Model

In the end, to improve FPB correction performance, we need a better-matched speckle model. However, we cannot measure speckle directly. The only measure we have of speckle is the histogram of detections per shot. This histogram is an amalgam of all the variations in seeing, clouds, pointing, etc. during the run. The histogram of number-of-shots with 'ndet' detections for the extremely bright real data differs markedly from that of the speckle simulation model. This suggests the speckle model as a likely reason for the discrepancy between the actual data and the simulations.

I made a quick try at a better model by guessing a functional form with adjustable parameters, simulating its detections, and adjusting the parameters to fit the measured detection histogram. My trial "brightness profile" has four adjustable parameters, α (alpha), b, c, and d:

$$brightness \equiv I(u) = \alpha N\left(bu^c + u^d\right) \qquad where \quad c < 1 \; and \; d > 1, \quad and$$

$$normalization \equiv N = \left(\int_0^1 \left(bu^c + u^d\right) du\right)^{-1} = \left(\frac{b}{c+1} + \frac{1}{d+1}\right)^{-1}$$

As is common with generating pseudo-random numbers, the profile takes a uniform random variable u on (0, 1), and produces a random value from the desired distribution by computing $I(u)$. $I(u)$ is the inverse cdf of the desired distribution. Alpha (α) is the usual average number of detectable photons per detector per shot, as before.

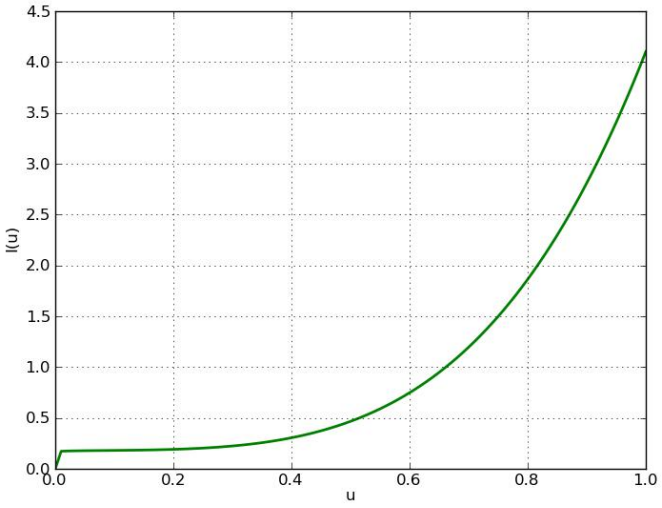


Figure 79: Sample plot of fitted $I(u)$ profile.

The fit profile $I(u)$ is shown in Figure 79. The first term allows for a curve up from zero, and leveling off. The second term provides a rise near the right-hand limit of $u = 1$, which produces a number of very bright shots. The coefficient b allows for a variable weight between the two terms. The normalization factor is calculated to keep the model average exactly 1, with α providing the usual scaling for number of detectable photons per detector per shot.

I enhanced the FPB simulator to allowing fitting speckle model parameters to an actual 'ndet' histogram. The best fits had a reduced-χ^2 of ~1.6, so the fit was reasonably good. Figure 80 shows the best fit.

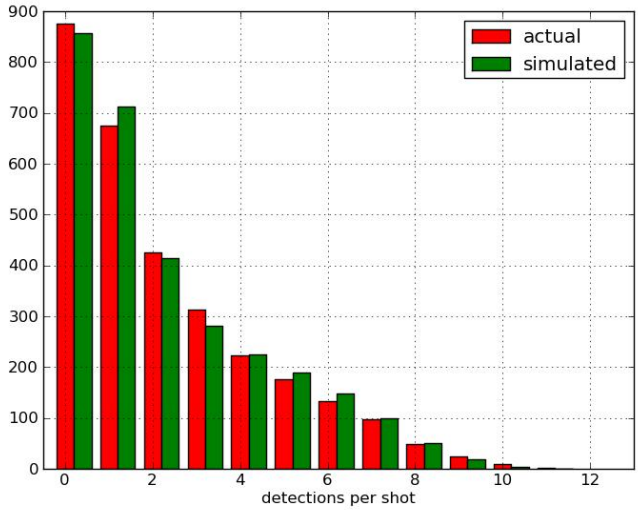


Figure 80: Sample histogram of fit speckle model vs. actual number of detections per shot.

However, at present, the bias vs. number-of-detections-in-shot resulting from this fit speckle profile is unexpectedly small, and it therefore does not provide proper FPB correction. Further investigation is necessary.

When Focus is Too Good

As noted above, the real data show that the actual FPB is larger than that simulated by ~60 ps in the strongest runs of 2-3 det/shot.

Spot sizes of 1 arcsec or bigger have virtually identical FPB response, to < 1 ps (< 3 ps at full reflector tilt). Spot size on the detector array is determined by "instantaneous" seeing, not the long-exposure (~15 sec) seeing. The instantaneous seeing is believed to be ~ 0.7 of the long-exposure seeing, though this varies depending on conditions. Spot size is a significant question, because spot sizes < ~0.85 arcsec have different FPB behavior than larger spot sizes. Correcting FPB for such shots requires knowing the spot size, which can only be imperfectly estimated. The corrections get noticeably larger as spot size shrinks, ~50% bigger at 0.5 arcsec compared to 1 arcsec. Errors in any spot size estimate propagate directly into uncertainty in the arrival times, and on into the final NP. The brightest returns likely have small spot sizes (which focus a lot of light on the detector), and

therefore large FPB.

Therefore, starting immediately, defocusing to an instantaneous spot size of ~1 arcsec is a good idea on extremely high-return runs (> ~1 det/shot). This has two effects: First, it spreads out the spot on our detector, reducing the bias in every APD, and also allowing more usable detections for the NP. Second, defocusing widens the spot on the moon, reducing the number of reflected photons (which is not necessarily desirable).

There are multiple problems associated with too tight a spot, which essentially defeats the benefit of multiple APDs in our detector array.

With too tight a spot, two photons are more likely to end up in the same detector, resulting in only one detection, and some first photon bias. If instead, those two photons are spread across two detectors, there is a double benefit: we get two detections, and they're both unbiased. This spatial spreading of photons can cause some defocusing to actually *increase* the number of detections, as well as the overall quality of those detections. Perhaps some real-time approximate measure of spot-size from ATUI or housctl (the hardware control computer program) could help in choosing when and how much to defocus.

Ideally, we'd like to recover all the photons we can, but spread them out better over the APD array. But defocusing also reduces the intensity at the reflector, and therefore our return photon count. Longer term, if the number of overly-tight-focus observing nights warrants it, we could add the ability to defocus an element in the receive path, thus preserving the outbound focus, and therefore the brightness on the moon, and number of detectable returned photons. The receive-path-only defocusing would then improve results (as above) by spreading the photons across more detectors, reducing bias and increasing usable detections.

Another problem with too small a spot is that it increases the error in estimating the overall brightness. The only measure we have of brightness is number of detections per shot. With all the photons collapsed into a small number of detectors, multiple photons cannot be counted by multiple detections, thus hiding the brightness.

We have considered the possibility of estimating spot size from the APD detections, possibly to adapt the FPB correction to spot size. However, it is much preferable to defocus to ~1 arcsec (instantaneous) spot size, above which the FPB is essentially independent of spot size. Also, estimating small sizes accurately is difficult. The difference in gain (detection efficiency) of the APDs in the array may become significant. If one detector has a higher gain than another, it will skew the measure of spot size, and also spot location if that is being estimated. Since even a tight spot jumps around from shot to shot, there is no easy way to account for varying gains. Also, the uncertainties in detection counts get worse (by about $\sqrt{11} \approx 3.3$) if we need to estimate the brightness of each detector individually. These uncertainties propagate into the spot-size estimate, and then into any spot-size-based FPB estimate. I believe the complexity and uncertainties involved will make a spot-size based approach to FPB correction infeasible. As mentioned, I believe some form of defocusing is much preferable.

Possible Improvement to FPB Correction

As a short-term, limited improvement to FPB correction, we could perhaps estimate the correction table for FPB from our actual data, using graphs such as Figure 76 (above). By comparing the uncorrected results, limited to no more than 3 detections per shot, with the uncorrected results limited to 1-detection shots, as a function of data rate, we could perhaps scale the simulated table of corrections to give the result indicated by Figure 76. While this would likely have more uncertainty than a more detailed speckle model would provide, it is almost certainty an improvement over what we have now.

Anomalies In Simulation Analysis

Anecdotally, there are times when either NP generation method produces unexpected anomalies: big jumps (~1 mm) from one value to another when a seemingly innocuous simulation parameter changes. Some of the jumps are not easily reproducible, and their origin is unclear. They generally do not appear in the systematic tests for stability, because *reproducible* jumps were able to be analyzed, understood, and eliminated. The net result, though, is that some anomalies may be currently

neither easily reproducible, understood, nor controlled.

4.4 On System Models and Data Uncertainty

There has been a claim that APOLLO data is not as reliable as our uncertainty claims. One argument given for this claim is that the reduced-χ^2 parameter that results from fitting our data to an existing model (built on 10 - 20 mm data) is significantly greater than 1. The argument continues by showing that by degrading our uncertainty claims, the χ^2 parameter can be made to be 1.

This argument can be mathematically proven invalid. From the definition of reduced-χ^2,

$$\text{reduced-}\chi^2 \equiv \frac{1}{\#\,dof}\sum_{i=1}^{N_{bins}}\frac{\left(meas_i - model_i\right)^2}{\sigma_i^2} \quad where \quad meas_i \equiv \text{the measured value of the } i^{th} \text{ data point}$$

$$\sigma_i \equiv \text{uncertainty of the } i^{th} \text{ measurement}$$

$$model_i \equiv \text{the model value of the } i^{th} \text{ data point}$$

it is easy to show that *any* process which uniformly increases the uncertainties will monotonically decrease χ^2. Therefore, for an *arbitrary* model, and *arbitrary* data with arbitrary uncertainties, if the initial $\chi^2 > 1$, there exists an uncertainty degradation process which will make χ^2 continuously smaller, and arbitrarily close to 0. In particular, there exists a process which will make $\chi^2 = 1$. This is an existence theorem. Since the theorem is true for arbitrary models and data, it is mathematically impossible for the execution of such a procedure, i.e. making $\chi^2 = 1$, by itself to provide *any* information about the correctness of the model, or the reliability of the uncertainty claims.

A second point concerns the correctness of the model. Given a model of sufficient complexity that it allows for a bounded error over the fitting interval which is of the same order as the data uncertainties, one cannot establish model data points significantly more reliable than the data used to fit the parameters. In other words, in a complex model allowing fluctuating errors, you can't separate the model errors from the data errors. You cannot "average out" the data errors over time, because the model is so complex (e.g. chaotic) that it has seemingly random errors of its own. Even more simply: you can't get a 1 mm model from 10 mm data. I believe the current solar system model, to which our data are fit, satisfies this condition of complexity and error bounds over the fitting interval. Further,

the model has deliberately avoided putting in physics whose results are estimated significantly below 10 mm. For example, the radiation pressure of sunlight on the moon is estimate to be ~ 4 mm effect, and is not yet included. Therefore, when applying such a model to APOLLO data, the reduced-χ^2 is *expected* to be > 1.

In fact, the archetypal definition of χ^2 is "goodness of fit;" it is a test of the correctness of a model, not of the credibility of uncertainty claims. More realistically, where experimental data are themselves subject to a healthy scientific skepticism, experimenters must devise measures beyond a single, simple χ^2 parameter to substantiate uncertainty claims. Such procedures are outside the scope of this dissertation.

5 Appendices

5.1 Contributions to APOLLO

While some results could be derived numerically by convolving known PDFs, the whole system is sufficiently complicated that direct numerical calculation of the final result is prohibitively complicated and error-prone. In contrast, simulation of the system, using known physics for the subsystems, provides more confidence in the result, and is more maintainable and flexible.

I developed and characterized an algorithm for first photon bias that significantly reduces the bias to below the mm level. This is complicated by the long tails of the detection time probability distribution function (PDF). In the tails, a 5 e-fold reduction takes 25 t_decay times, because of the square-root in the PDF. This gives long tails of about 40 ns.

During my time on APOLLO, I have made a number of contributions to the project. Some of them are the subject of this dissertation, and some are outside its scope.

Among those relating to this dissertation are the simplified APD detection time model, with better performance and fewer parameters; characterization of APD parameters for the actual 532 nm laser in operation, necessary for both data reduction and FPB correction; a complete, independent data reduction implementation, for validation of existing normal points; a data reduction software system that is easily modified to perform tests of reduction methods, and reduction parameter optimizations, used to find optima of several parameters; a first photon bias simulator for more in-depth analysis, including a variety of spatial and temporal illumination profiles; and a full system simulator that produces standard run files that can be reduced using standard reduction software. The full-system simulator is described in another Appendix. As a part of data reduction, I also wrote a TDCFIT program, again to independently confirm the results of the existing TDCFIT program.

Software libraries: I also created portable, maintainable, software libraries and utilities that are both used in the work of this dissertation, and also generally useful for many types of future work. These software modules include a keyword parameters library, that allows seamless integration of new keyword parameters to existing code; a C-library routines for reading and writing data files and

normal point files;

To produce most of my graphics, including those for this dissertation, I wrote a set of Python graphing scripts. They select data from flexibly formatted files, and produce graphs of various kinds, with many command line options. This allows very rapid visualization of data with minimal preparation time. These scripts rely on SciPy v0.7.2 and NumPy v1.4.1.

Other contributions include suggesting the position-variable attenuator on the quadrant diffuser for control of fiducial intensity during runs; and the Houston control software, that continuously runs the entire system.

Houston control software: During observing, "housctl" coordinates and controls all the subsystems in real-time, with a 25 ms event period. It also operates heating and cooling subsystems to maintain the environment of multiple air spaces, both when observing and when idle. I contributed to the extensible data and log file formats, including justification for various choices, and to the alarm and event notification systems. I also provided detailed operating and design documentation for housctl. For both housctl and other general purpose utilities, I created, extended, and documented reusable driver libraries for several devices, including the chiller, TCP sockets, serial ports, general asynchronous interfaces, and the Continuum laser.

To control the laser, I reverse engineered the Continuum laser control interface, to allow our Houston control computer to directly operate the laser. I fully documented the interface both in a manual, and in the interface code, and wrote the library for automated computerized control of the laser.

Instant prediction polynomials: Currently, APOLLO fits polynomials to RTTs generated by an ephemeris program. These polynomials are then used to predict the range to the moon during observing, which is necessary to place the lunar return within our measurement window. This fitting process is somewhat cumbersome. I demonstrated the feasibility of eliminating the separate fits to the predicted ranges. Instead, we can simply feed the ephemeris-predicted ranges to housctl, and it can easily compute a 4^{th}-order polynomial that is as good to a few picoseconds as the current polynomials.

The instant polynomial is valid for over an hour, which is far longer than any run we do.

5.2 The APOLLO Simulator

I developed an APOLLO run simulator, with accurate simulation of all significant effects. The data simulator creates complete run-files with known noise parameters, as a tool for verifying data analysis software, and investigating effects such as first photon bias. The run-files can be analyzed with standard APOLLO data analysis software, just like real run-files. We can compare data reduction results with the known simulator inputs to characterize our reduction for bias and uncertainty.

All the significant parameters are configurable, to allow flexible testing of individual effects. The simulations include (roughly following a laser pulse from beginning to detection):

- Laser fire time delay standard deviation
- Laser pulse width std dev
- Alternating FIDs, both uniform illumination and point illumination
- definable average photon rate per shot
- Slower (multi-second scale) random intensity drift simulates drifting observing conditions
- Random “speckle” variation (shot-to-shot) following simulated speckle PDF
- Retroreflector tilt in X and Y axes (relative to reflector)
- prepulses for both FIDs and LUNs
- First photon bias
- spatially dependent illumination of 16 APDs.
- Any number of working APDs
- Background APD triggers for dark (thermal) electron-hole pair generation period
- Background APD triggers for stray lunar photons
- APD time distribution (PDF), including radial dependent APD detector delay, flat to 8 μm radius, parabolic (to 100 ps) at 15 μm radius
- TDC standard deviation

Features:

- Produces standard data files with known impairment & noise parameters
- Can selectively turn off any and all noise, except for TDC quantization. This is helpful for developing analysis algorithms, debugging analysis software, and isolating cause and effect.

Known Limitations

The X and Y axes are defined as aligned with the retroreflector array.

The TDC is modeled as exactly 25 ps/bin.

There is no distinction between fiducial prepulse probability and lunar

No crosstalk between APDs

APOLLO Run Simulator Parameters

The simulation includes the following settable parameters (defaults in parentheses):

- It can turn off all noise, except for TDC quantization. This is helpful for debugging analysis software. Otherwise:
- Laser fire time delay standard deviation (1 μs)
- Laser pulse width std dev (40 ps)
- Fiducial average photon rate per APD in the center 4 elements (0.1)
- Prepulse probability (0.01)
- Thermal generation period (50 μs)
- APD standard deviation (25 ps)
- TDC standard deviation (13 ps)
- Radial dependent APD detector delay, flat to 8 μm radius, parabolic to 'detparab' at 15 μm radius
- Retroreflector tilt in X and Y axes (relative to APD) (10, 2.5 deg)
- Lunar average photon rate per APD in the center 4 elements (0.1)
    - Lunar detectors outside the center 4 elements have ½ the illumination of the center

4.

- Stray lunar photon period (50 μs)
- Controllable first photon bias (FPB) separately for fiducial and lunar detections

The follow features are currently fixed, and cannot be configured:

- Fixed prepulse rate for both FIDs and LUNs

These are the configurable parameters, and their defaults, as of 6/3/2009:

```
usage: apsim [keyword=value ...]
       simulator=1               1=this is a simulated runfile
          simopt=0               simulation option bit flags
          runpat="\temp\apsim%2#.run" output file name pattern
         comment=""              runfile comment

specklefile="\travel_briefcase\EricSchool\research\speckle2.dat;\temp\sp
eckle2.dat" speckle sample file
             var=std             {none, std, fpb}
            nsim=1               # runfiles to simulate
           nruns=20000           # shots to simulate
            seed=1               random # seed, 0 = take time of day
   fidphotonavg1=0.1             avg photons/odd-shot/APD
   fidphotonavg2=0.17            avg photons/even-shot/APD
      fidphotmax=99              fiducial FPB limit
       cc2_space=0.              2nd fiducial CC space, m
    lunphotonavg=0.05            avg photons/shot/APD
      lunphotmax=99              lunar FPB limit
       spotshape=center          spatial illumination shape of APDs
          seeing=1.              spot-size, FWHM arc-sec
           profi=speckle         speckle intensity profile
          idrift=0               enable slow intensity drift
        lasersd=9.5e-11          laser pulse std dev, s
          fpdsd=5.5e-11          FPD std dev, s
          detsd=2.5e-11          APD std dev, s
       detparab=1.e-10           APD parabola height, s
          tfrac=0.212            APD diffusion tail fraction
         tdecay=9.7e-10          APD diffusion tail decay, s
          tdcsd=1.3e-11          TDC std dev, s
          tiltx=10.              reflector tilt in x, deg
          tilty=2.5              reflector tilt in y, deg
     prprepulse=0.01             probability of prepulse
       thermrate=115000.         thermal electron generation rate, 1/s
       strayrate=462000.         lunar stray photon rate, 1/s
          firesd=1.e-06          fire time std dev, s
           dskew=-41.            (ns) t_lun = t_FPD + dskew + RTT
              t0=0.              polynomial start time
              tf=1.1             polynomial end time
             pdm=8.64e-09        prediction error slope, s/day
             pdb=1.e-09          prediction error offset, s
     tdc_target=2000             target TDC of lunar prediction
          motrps=20.             T/R speed setpt, rps
       reflector=0               Lunar reflector: 0=Ap11, 1=Lunokhod 1,
2=Ap14, 3=Ap15, 4=Lunokhod 2
```

Using the Data Simulator

Example: **apsim datafile="xyz.run" novar=1**

creates “xyz.run” with no noise (novar=1). Common parameters are:

nruns # shots to simulate

var specifies the type of variation to include: 0 disables all variation (noise), except TDC quantization; 1 provides “standard” impairments; 2 gives “standard” FPB calibration impairments

datafile output file name

tdcsd TDC std dev, s

fidphotonavg avg photons/shot/APD

fidphotmax max # detectable photons/detection (limits FPB)

lunphotonavg avg photons/shot/APD

lunphotmax max # detectable photons/detection (limits FPB)

lasersd laser pulse std dev, s

firesd fire time std dev, s

tiltx reflector tilt in x, deg

tilty reflector tilt in y, deg

prprepulse probability of prepulse

thermtime avg thermal electron generation time

straytime avg lunar stray photon time

detsd APD detector std dev

detparab APD parabola delay at edge

There are many other parameters described in the code.

All command-line numbers to apsim can have suffixes that specify the units. The suffixes follow the conventions of the SPICE electrical analysis program:

G = giga M = mega k = kilo m = milli u = micro

n = nano p = pico

For example, the following two commands are equivalent:

```
apsim nruns=20000 runpat=\temp\gauss120.run var=0 lasersd=120e-12
apsim nruns=20k runpat=\temp\gauss120.run var=0 lasersd=120p
```

Useful Test Settings

See [dosim.bat] for all the simulations used for analysis.

Standard simulation, with realistic parameters:

```
apsim nruns=20000 runpat=\temp\std20k.run
```

Simulation with no noise or quantization error:

```
apsim runpat=\temp\novar.run var=0
```

Quantization error only, fixed RTT, etc.

```
apsim runpat=\temp\gauss120.run var=0 lasersd=120e-12
```

This moves the shot time and return times around, but keeps the RTT fixed.

First photon bias and quantization error only:

```
apsim runpat=\temp\fpbtest.run var=0 lasersd=120e-12 fidphotmax=99
   lunphotmax=99
```

Note that first photon bias requires some variation in photon arrival time: if they all arrive at the same time, there can't be any bias.

Run File Statistics

At the end of each file are some statistics on how many of each type of event came out of the simulation. For example,

```
rem Lunar signal-dets/shot: 0.4985.  photons/shot/det avg dev: 0.0497849 0.112991
rem Fid types: photons, thermals, strays, prepulses, nodets: 32005 3107    0    0
224888
rem Lun types: photons, thermals, strays, prepulses, nodets:  9970 2995 20462    0
226573
rem Fid photons/detection counts: 227245 30521 2120 110 4 0 0 0 0 0 0 0 0 0 0 0
rem Fid detection (nhit) counts:      0  3278 6111 5489 , 3201 1336 451 119 13 2 0
0 0 0 0 0 0
rem Lun photons/detection counts: 248591 10075 1093 194 36 10 1 0 0 0 0 0 0 0 0 0 0
rem Lunar dets/shot avg dev: 1.6707 1.45512
rem Lun detection (nhit) counts:   4228  6282 4964 2530 , 1132 445 214 114 58 19 10
4 0 0 0 0 0
rem First photon biases avg dev: FID: -1.58289e-12 7.36877e-12, LUN: -1.19934e-11,
4.05426e-11
rem Elapsed time (ms) 1572.000
```

We describe these in turn:

```
rem Lunar signal-dets/shot: 0.4985.  photons/shot/det avg dev: 0.0497849 0.112991
```

The average number of lunar signal photon detections per shot; you can compare this to your analysis software results, for consistency. Followed by the average number of detectable photons per shot, with its standard deviation. This number is related to first photon bias.

```
rem Fid types: photons, thermals, strays, prepulses, nodets: 32005 3107
0    0 224888
rem Lun types: photons, thermals, strays, prepulses, nodets:  9970 2995
20462    0 226573
```

The sum of the 4 columns is the total number of possible APD events: (20000 shots) x 13 APD elements = 260,000 possible APD events.

```
rem Fid photons/detection counts: 227245 30521 2120 110 4 0 0 0 0 0 0 0
0 0 0 0 0
rem Lun photons/detection counts: 248591 10075 1093 194 36 10 1 0 0 0 0
0 0 0 0 0 0
```

The number of APD detections with 0, 1, 2, ... detectable photons in them.

```
rem Fid detection (nhit) counts:      0  3278 6111 5489 , 3201 1336 451
119 13 2 0 0 0 0 0 0 0
rem Lun detection (nhit) counts:   4228  6282 4964 2530 , 1132 445 214
114 58 19 10 4 0 0 0 0 0
```

The number of shots with 0, 1, 2, ... APD elements firing.

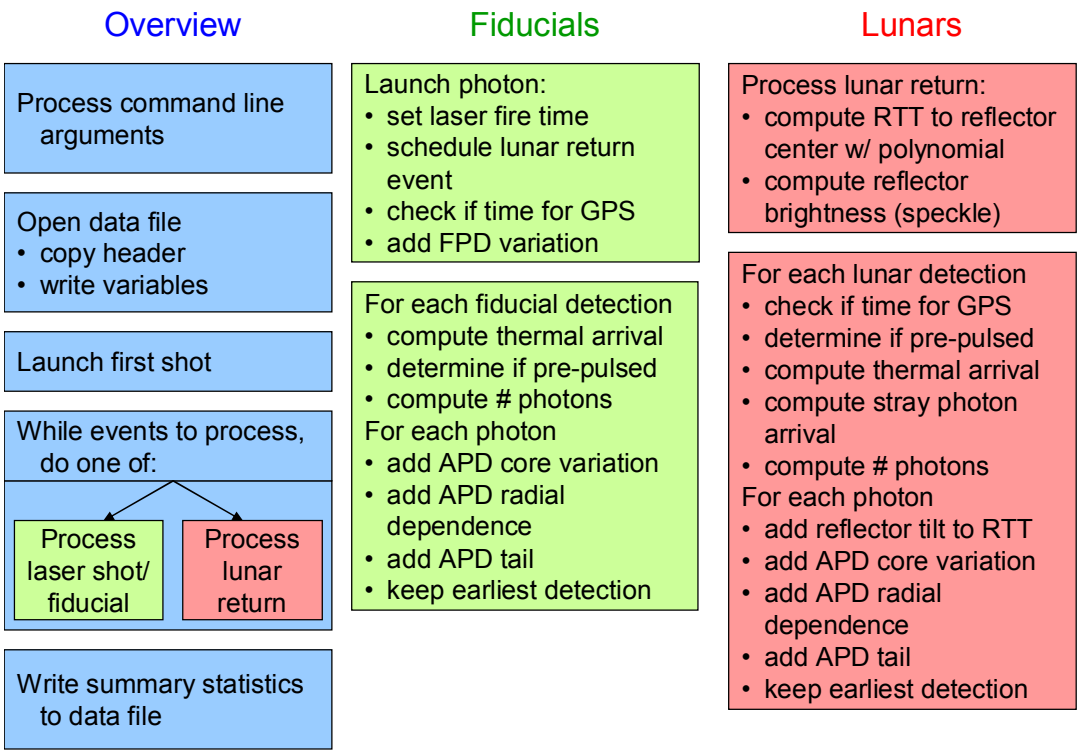
```
rem First photon biases avg dev: FID: -1.58289e-12 7.36877e-12, LUN: -
1.19934e-11, 4.05426e-11
```

Computed fiducial and lunar first photon biases, and their standard deviations.

```
rem Elapsed time (ms) 1572.000
```

The computer time it took to generate this simulation.

Simulator Design



Some Details of the Processing

Define u as a uniform random variable on (0,1).

The laser fire time average delay is 3 times deviation. Variation is gaussian.

FPD variation is gaussian.

Number of detectable photons in a detector is Poisson.

APD prepulse probability is coded as $u <$ Pr(prepulse):

```
if(rand_uniform() < prprepulse)
```

Thermal arrival is exponential PDF, using $\text{cdf}^{-1}(u)$, where u is uniform on (0,1):

```
// See if thermal electron or lunar stray beats out the earliest photon
r = -thermtime*log(rand_uniform());   // time to next thermal electron
```

Lunar stray photon arrival is exponential PDF, using $\text{cdf}^{-1}(u)$, where u is uniform on (0,1):

```
r = -straytime*log(rand_uniform());   // time to next lunar stray
```

APD core variation is gaussian.

APD radial dependence of detection time:

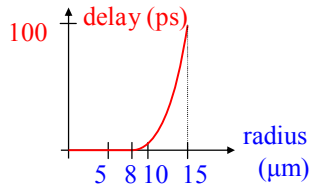


Figure 81: Radial dependence of APD detection time.

The radius is for uniform areal probability, using $\text{cdf}^{-1}(u)$, where u is uniform on (0,1):

$$cdf(r) = \frac{\pi r^2}{\pi R^2} = \left(\frac{r}{R}\right)^2 \qquad \Rightarrow \qquad \frac{r}{R} = cdf^{-1}(u) = \sqrt{u}, \qquad r = R\sqrt{u}$$

Coded as

```
r = 15.*sqrt(rand_uniform()) - 8.;    // microns
```

APD radial dependence is flat to "cut-in" radius, and parabolic to edge (15 um):

```
if(r > 0.)  time += detparab * r * r / (7. * 7.);  // 7um = 15 - 8
```

APD tail has a fixed probability, and is exponential in detection time:

```
if(rand_uniform() < ft)           // tail
t +=  -tau*log(rand_uniform() );
```

Reflector is A11/A14 type. Independent tilt in X and Y:

```
// We assume X & Y angles are small, and therefore independent
```

```
rrslopex = sin(tiltx*(M_PI/180.)) * .046 / PHYS_C; // meters/corner-cube
rrslopey = sin(tilty*(M_PI/180.)) * .046 / PHYS_C; // meters/corner-cube
cubeid = (int32)(100*rand_uniform());              // which cube photon
hits
return ((cubeid % 10) - 4.5)*rrslopex + ((cubeid / 10) - 4.5) * rrslopey;
```

RTT is from a real polynomial.

Speckle brightness determines average photons/detector per the simulations from Tom.

TDC Quantization

It might seem trivial, but for consistent results in zero-noise simulations, the TDC quantization needs to be *centered* at "nice" time values, i.e. multiples of 20 ns (our system clock period). For example:

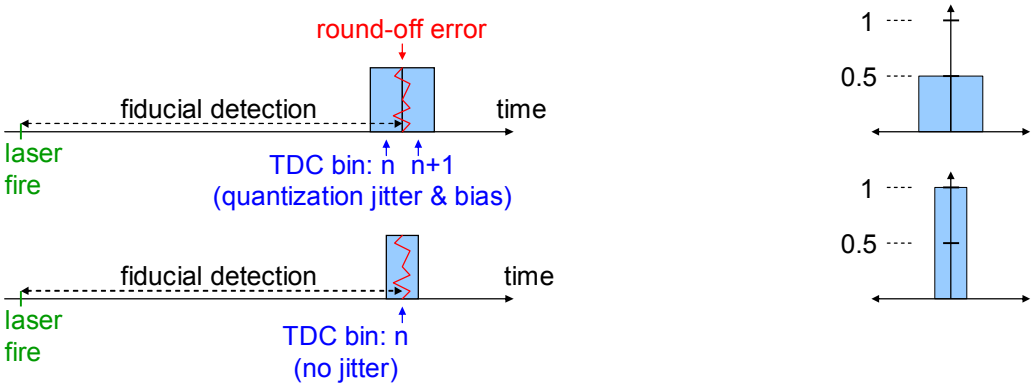


Figure 82: (Top) Bad bin boundary choice causes unpredictable, and often biased, rounding variation between bins. (Bottom) Good bin boundary choice eliminates round-off error. (Right) Putting zero between bins disallows a Kronecker delta function. Centering zero in a bin allows a true Kronecker delta.

For "nice" fiducial detection times, say 30.6 ns, we need there to be no TDC jitter due to round-off error. This allows no-noise simulations to have exact, reproducible results. The simulation therefore offsets the time by ½ bin before quantizing to a 25 ps TDC bin. Note that the real TDC offset is about 12 ns, roughly 1000× bigger than the simulation. In real data reduction, since all the TDC offsets are subtracted out, the average magnitude of the offset is not important, and quantization jitter is small compared to other variations.

A second important reason for defining the reference time to be the center of the bin is that it allows a Kronecker delta function to be defined. The $t = 0$ bin is set to 1, and all others are zero. This

is important for some operations, such as the retroreflector tilt PDF at zero tilt: it must be a Kronecker delta function. In contrast, defining $t = 0$ to be *between* bins does not allow a Kronecker delta function to be defined.

Thirdly, if we convolve two PDFs (or other quantized functions) which both have zero *between* bins, the result is a PDF with zero *centered* in a bin. This important shift must be tracked and exactly accounted for; again, in APOLLO, a half-bin error is huge. This problem does *not* occur with zero centered in a bin: the convolution results in a zero still centered in a bin. The difference arises because the zero points of the original functions add to produce the zero point of the convolved result. A zero between bins is a ½-bin value, and the sum of two ½-bin values is an integer bin, thus changing the convention. However, with zeros centered in bins, the sum of two integer bins is another integer bin, and the zero point remains consistently centered for all functions.

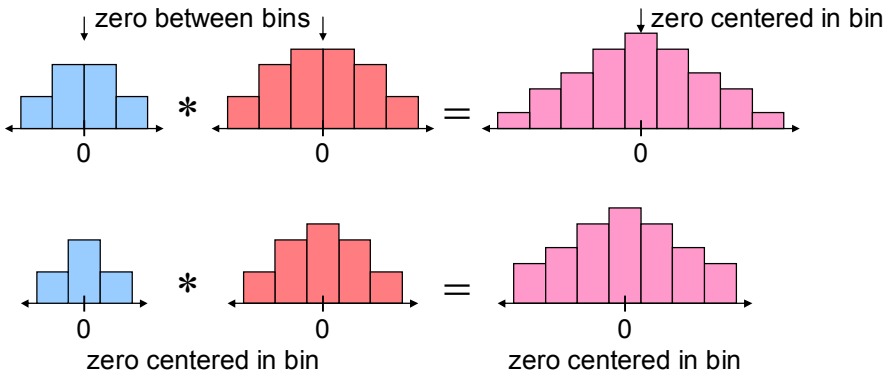


Figure 83: (Top) Putting zero between bins cannot be maintained when convolving functions. (Bottom) Centering zero in a bin is preserved in the convolved result.

Generating Pseudo-Random Tail Times On The Computer

The "standard" method for generating arbitrary distributions on a computer is generate a uniform pseudo-random variable, and apply the inverse of the CDF to that:

$t = \mathrm{cdf}^{-1}(u)$ *where* u is a uniform random variable

However, the tail CDF is not analytically invertible. For a long time, I generated pseudo-random tail times for simulation using the elimination method. This method generates a random 2D point within a rectangle, and tests whether the point falls below the PDF curve in that rectangle. If

not, the point is rejected, and a new point is generated, until the point falls under the PDF. That point's t value is the generated random value. Because the tail is so long and low, it is inefficient to use the simple elimination method, and it also requires a truncation choice (since the tail goes to infinity). The longer the tail included (i.e., the less truncation error), the less efficient the algorithm becomes. Perhaps a piecewise optimization of the method would be better, but the iterative solutions below are significantly superior on all counts.

In the case of the tail CDF, an iterative solution works to invert it:

$$t = \mathrm{cdf}^{-1}(u) \qquad \textit{where}$$

$$u \equiv \mathrm{cdf}_{tail}(t) = 1 - \exp\left(-(t/\tau)^{1/2}\right)\left((t/\tau)^{1/2} + 1\right)$$

$$= 1 - \exp(-y)(y+1) \qquad \textit{where} \quad y \equiv (t/\tau)^{1/2}$$

$$y = \mathrm{cdf}^{-1}(u) \qquad \textit{satisfies} \qquad \exp(-y)(y+1) = 1 - u$$

But u is uniform on $(0, 1)$, so $1-u$ is also. Then $1-u \to u$. Taking logarithms:

$$-y + \ln(y+1) = \ln u \qquad \textit{and} \qquad y_{n+1} = \ln(y_n + 1) - \ln u$$

The slope of the RHS for y is small, so this iteration equation converges fairly quickly. Convergence is faster for large y (RHS slope decreases), therefore a good starting point for iteration is to solve approximately for small y:

$$\ln(y+1) \approx y - \frac{y^2}{2} \qquad \Rightarrow \qquad y_0 = y_0 - \frac{y_0^{\,2}}{2} - \ln u, \qquad y_0 = \sqrt{-2\ln u}$$

Finally,

$$t = \tau y^2$$

Numerical testing shows virtually no truncation error, and several times faster execution times than the elimination method. The "direct" iterative solution was later replaced with a Newton-Raphson iterative solution, that converges even faster.

Simulations for Data Reduction

I typically simulate a fixed prediction RTT of exactly 2.475 s, with an actual RTT of 2.475 s + 1 ns + 8.64 ns/day drift. For a 5000 shot run, this results in a normal point RTT of

$$RTT_{NP} = 2.475\,000\,001\,000 + \left(8.64\ ns / 86400 s\right)\frac{1}{2}\left(\frac{5000\ shots}{20\ shots / s}\right) = 2.475\,000\,001\,012\ s$$

It is very important that this result in a uniform distribution of photon measurements across a full TDC bin of 25 ps. Otherwise, the quantization error will introduce significant bias in the quantized measurements, and prevent an accurate normal point calculation. This means that all simulated runs must be a multiple of 5000 shots, or must change the range drift rate for uniform TDC coverage.

Background rates: a dark rate of .15 detections/gate, and a bright rate of 0.75 detections/gate. Divided by 13 APDs equals 115,000 /s/apd, plus a stray photon rate of 462,000 /s/apd.

For "worst case" simulations, I use the maximum background illumination, and maximum libration angles of 6° in *y* and 9° in *x*.

5.3 Derivations

Here are derivations of some of the statements in the main text.

Demonstration That Working Too Hard Gives Worse Results

Consider a hypothetical random process with

- an average near 100
- Standard deviation of 10
- Daily drift in the average, with standard deviation of 1

Suppose we try to "correct" for the daily drift, as follows

- Each morning, take 4 measurements
- Compute the average
- Subtract that average from all future measurements that day

The error in this estimate of the average is 5, much *greater* than the deviation of the average itself. Such a procedure gives *worse* results than nothing at all.

Tail normalization constant:

$$\mathrm{pdf}_{tail}(t) = N\exp\left(-\sqrt{t/\tau}\right) \qquad \textit{where} \quad N \text{ is the normalization}$$

$$\frac{1}{N} = \int_0^\infty dt\,\exp\left(-(t/\tau)^{1/2}\right) \qquad \textit{Let} \qquad y = (t/\tau)^{1/2}, \qquad t = \tau y^2,\ dt = 2\tau y\,dy$$

$$= \int_0^\infty 2\tau y\,dy\,\exp(-y) = 2\tau\,\Gamma(2) = 2\tau \qquad \Rightarrow \qquad N = \frac{1}{2\tau}$$

The average and standard deviation are:

$$\langle t\rangle = \int_0^\infty dt\,t\,\mathrm{pdf}_{tail}(t) \qquad \textit{Let} \qquad y = (t/\tau)^{1/2}, \qquad t = \tau y^2,\ dt = 2\tau y\,dy$$

$$= \frac{1}{2\tau}\int_0^\infty 2\tau y\,dy\,\tau y^2\exp(-y) = \tau\int_0^\infty dy\,y^3\exp(-y) = \tau\,\Gamma(4) = 6\tau$$

$$\langle t^2\rangle = \int_0^\infty dt\,t^2\,\mathrm{pdf}_{tail}(t) \qquad \textit{Let} \qquad y = (t/\tau)^{1/2}, \qquad t = \tau y^2,\ dt = 2\tau y\,dy$$

$$= \frac{1}{2\tau}\int_0^\infty 2\tau y\,dy\,\tau^2 y^4\exp(-y) = \tau^2\int_0^\infty dy\,y^5\exp(-y) = \tau^2\,\Gamma(6) = 120\tau^2$$

$$Var(t) = \langle t^2\rangle - \langle t\rangle^2 = 120\tau^2 - 36\tau^2 = 84\tau^2$$

$$Dev(t) = \sqrt{Var(t)} = 9.165\tau$$

The $\mathrm{CDF}_{\mathrm{tail}}$ exists in closed form:

$$\mathrm{CDF}_{tail}(T) = \frac{1}{2\tau}\int_0^T dt\,\exp\left(-(t/\tau)^{1/2}\right) \qquad \textit{Let} \qquad y = (t/\tau)^{1/2}, \qquad t = \tau y^2,\ dt = 2\tau y\,dy$$

$$= \frac{1}{2\tau}\int_0^{(T/\tau)^{1/2}} 2\tau y\,dy\,\exp(-y) = \left[e^{-y}(-y-1)\right]_0^{(T/\tau)^{1/2}} = 1-\exp\left(-(T/\tau)^{1/2}\right)\left((T/\tau)^{1/2}+1\right)$$

Effect of Background Photons

For the uncertainty calculation, we need to find:

$$Dev\left(\frac{n_b}{n_s}\right) = Dev\left(\frac{n_b}{n-n_b}\right) \qquad \textit{where} \quad n_s = n - n_b$$

Therefore, if n_b varies by x, n_s varies by $-x$. So we have:

$$Dev\left(\frac{n_b}{n_s}\right) = \frac{d}{dx}\left[\frac{n_b+x}{n_s-x}\right]_{x=0} Dev(n_b) \qquad \textit{and} \qquad Dev(n_b) = \sqrt{n_b}$$

$$\frac{d}{dx}\left[\frac{n_b+x}{n_s-x}\right]_{x=0} = \left.\frac{(n_s-x)+(n_b+x)}{(n_s-x)^2}\right|_{x=0} = \frac{n_s+n_b}{{n_s}^2} = \frac{1}{n_s}\left(1+\frac{n_b}{n_s}\right) \qquad \Rightarrow$$

$$Dev\left(\frac{n_b}{n_s}\right) = \frac{1}{n_s}\left(1+\frac{n_b}{n_s}\right)\sqrt{n_b}$$

Variance of Trapezoid PDF

We derive the variance of the retroreflector trapezoid PDF by starting with the mean-squared value of the general trapezoid (**Error! Reference source not found.**), and scaling it to the normalized reflector PDF.

$$MS = 2\int_0^a dx\, x^2 + 2\int_a^b dx\, x^2\left(1-\frac{x-a}{b-a}\right)$$
$$= \frac{2x^3}{3}\Big|_0^a + 2\left(\frac{x^3}{3} - \frac{x^4}{4(b-a)} + \frac{ax^3}{3(b-a)}\right)\Big|_a^b = \frac{2a^3}{3} + 2\left(\frac{b^3-a^3}{3} - \frac{b^4-a^4}{4(b-a)} + \frac{a\left(b^3-a^3\right)}{3(b-a)}\right)$$
$$= \frac{2b^3}{3} - \frac{1}{b-a}\left(\frac{b^4-a^4}{2} - \frac{2a\left(b^3-a^3\right)}{3}\right)$$

The PDF for the retroreflector array is given in terms of the width of the flat region, w_{flat}, and the width of each of the sloped regions, w_{slope}. Properly normalized, we have:

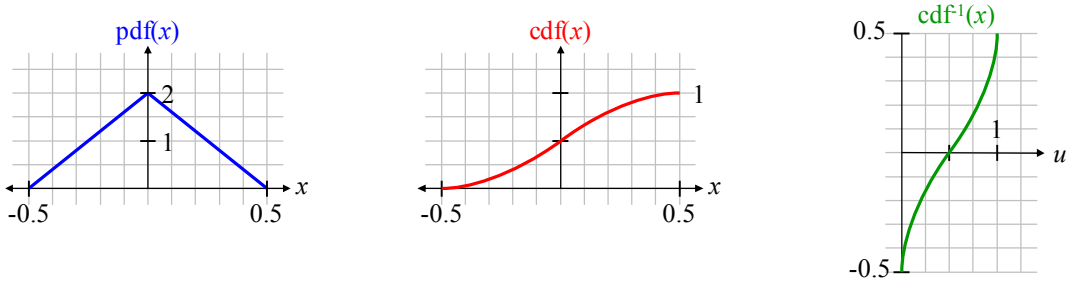
$$PDF(t) = \frac{1}{w_{flat}+w_{slope}}\left(PDF_{trapezoid}\right), \qquad a = w_{flat}/2, \qquad b = w_{flat}/2 + w_{slope}$$
$$\sigma^2 = \frac{1}{w_{flat}+w_{slope}}\left[\frac{2b^3}{3} - \frac{1}{b-a}\left(\frac{b^4-a^4}{2} - \frac{2a\left(b^3-a^3\right)}{3}\right)\right]$$

I verified this result numerically.

First Photon Bias Derivations

Triangle PDF in Time

The other extreme for a retroreflector is a triangle PDF, symmetric about 0:



$$\text{pdf}(x) = \begin{cases} 4(x+0.5) & -0.5 < x < 0 \\ 4(0.5-x) & 0 \le x < +0.5 \end{cases}$$

$$\sigma^2 = 2\int_0^{+0.5} dx\, 4(0.5-x)(x-0)^2 = 8\int_0^{+0.5} dx \left(\frac{x^2}{2} - x^3\right) = 8\left[\frac{x^3}{6} - \frac{x^4}{4}\right]_0^{0.5} = \frac{1}{6} - \frac{1}{8} = \frac{1}{24}$$

$$\sigma = \frac{1}{\sqrt{24}}$$

To generate the triangle PDF on the computer, we use the general form $x = \text{CDF}^{-1}(u)$, where u is uniform on (0, 1). To find $\text{CDF}^{-1}(u)$, we first find CDF(x). Consider first the negative side of the above PDF, $x < 0$, and note that cdf(0) = 0.5:

$$\text{cdf}(x) = \int_{-0.5}^{x} dx'\, 4(x'+0.5) = \left[2(x'+0.5)^2\right]_{-0.5}^{x} = 2(x+0.5)^2 \equiv u(x) \qquad \Rightarrow$$

$$x(u) = \text{cdf}^{-1}(u) = \sqrt{\frac{u}{2}} - 0.5, \qquad u \le 0.5$$

On the positive side of the PDF, $x > 0$, we have

$$\text{cdf}(x) = 0.5 + \int_0^x dx'\, \text{pdf}(x') = 0.5 + \int_0^x dx'\, 4(0.5-x') = 0.5 - \left[2(0.5-x')^2\right]_0^x$$

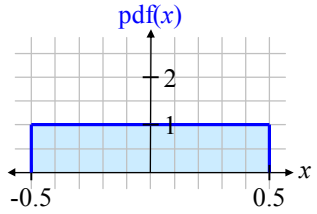
$$= 1 - 2(0.5-x)^2 \equiv u(x) \qquad \Rightarrow$$

$$1 - u = 2(0.5-x)^2, \qquad x(u) = 0.5 - \sqrt{\frac{1-u}{2}}, \qquad u \ge 0.5$$

Note that since u is restricted, u >= 0.5, we *cannot* use the common simplification that if u is uniform on (0, 1) then so is $(1 - u)$, so $(1 - u) \to u$.

Rectangular (Uniform) PDF in Time

The worst case (i.e., least gaussian) retroreflector profile is a rectangular PDF. Using a uniform PDF over the standard interval [–0.5, +0.5], PDF(x) = 1, we have



$$\sigma^2 = \int_{-0.5}^{0.5} dx\,(x-0)^2 = \frac{1}{3}x^3\Big|_{-0.5}^{+0.5} = \frac{1}{12}, \qquad \sigma = \frac{1}{\sqrt{12}}$$

Dependence of Average on Window Placement

The average of a gaussian distribution within a window of width *w*, centered on the origin, is zero, by symmetry. For an arbitrary window position, starting at *l* and ending at *r*, we have:

$$\bar{x} = \frac{\int_l^r dx\; x\,\mathrm{pdf}_{gauss}(x)}{\int_l^r dx\;\mathrm{pdf}_{gauss}(x)} \qquad where \quad l, r \text{ are the window left and right edges}$$

How much does the average depend on the window position? Let w ≡ window size, in sigmas. If the window is displaced from 0 by a small amount to the right, *dp*, the numerator loses on the left pdf(–*w*/2) (–*w*/2) *dp*, and gains the same magnitude on the right. At the origin, the denominator does not change to first order (its first derivative is zero). Therefore,

$$\frac{d\bar{x}}{dp} = \frac{2(w/2)\,\mathrm{pdf}_{gauss}(w/2)}{\int_{-w/2}^{w/2} dx\;\mathrm{pdf}_{gauss}(x)} = \frac{w\exp\left(-w^2/8\right)}{\mathrm{erf}\left(\frac{w}{2\sqrt{2}}\right)} \qquad where \quad \mathrm{erf}(x) \equiv \frac{2}{\sqrt{\pi}}\int_0^x du\; e^{-u^2}$$

Python script:

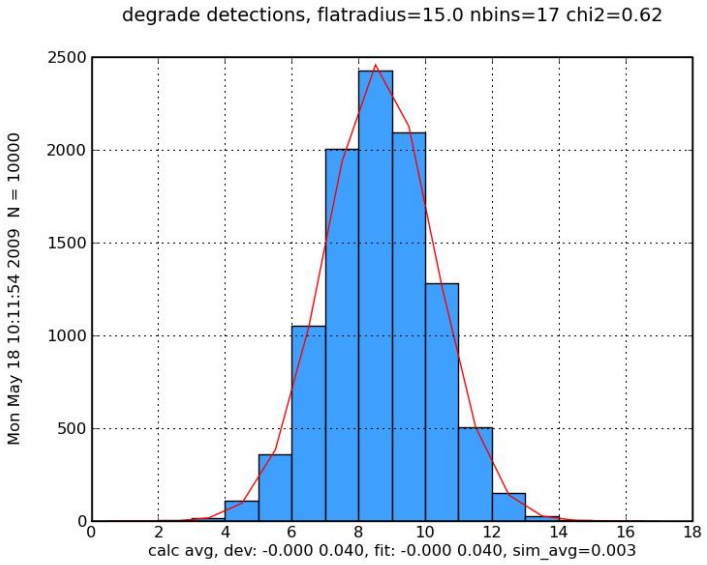
```
from pylab import *             # matplotlib
from scipy.special import *     # erf()

def g(z):
    return erf(z/sqrt(2))

def d(w):
    return w*exp(-w*w/8)/sqrt(2*pi)/g(w/2)

x=arange(.01, 8, .01)
xlabel("window width, sigmas")
ylabel("dx-bar/dp")
plot(x, d(x), linewidth=2)
show()
```

5.4 APD Spatial Dependence

The intra-APD spatial dependence introduces a noticeable ~27 ps delay, on average, over the active surface. It also slightly deforms the time PDF from gaussian, but the examination below shows that this slight deviation is negligible. We therefore model the APD as a gaussian core, plus a tail.

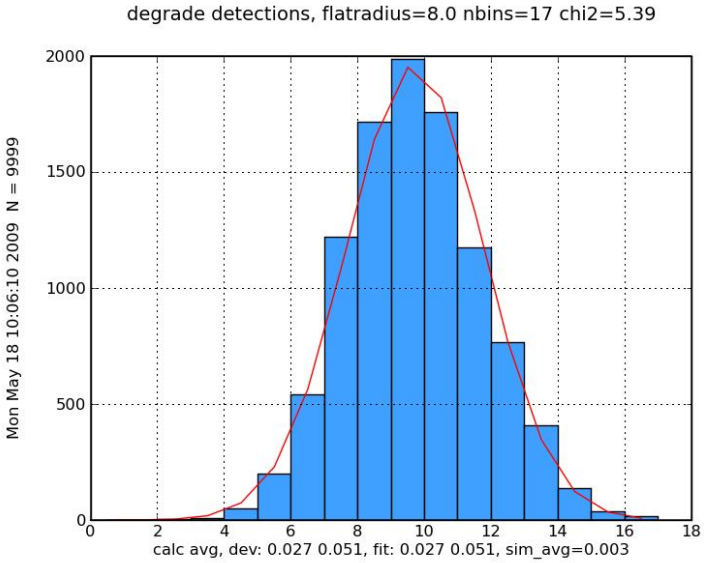
First, a calibration simulation verifies the integrity of the simulation an fitting function

[degrade.c]. I started with a laser gaussian with σ = 40 ps, and no spatial APD dependence:

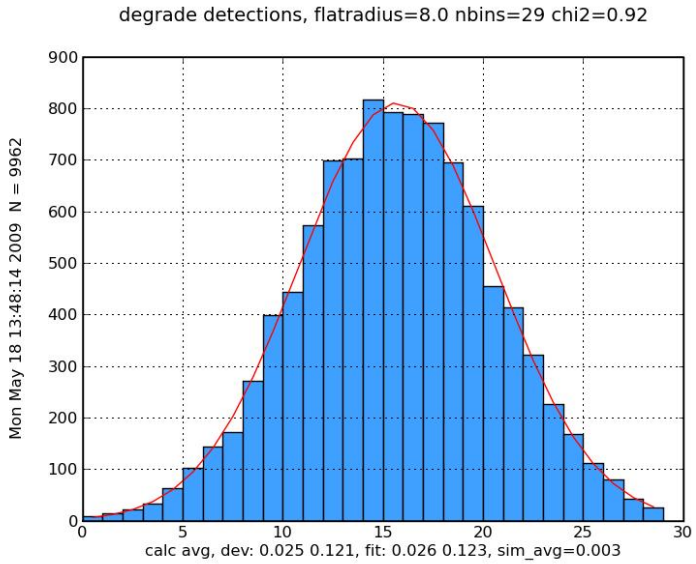


The reduced χ^2 is 0.62, which is surprisingly small. This may be due to the non-gaussian (binomial) distribution of counts in a histogram bin. At low bin counts, the discrete counts can sometimes cause the histogram to turn out "better than" a continuous gaussian distribution. Another factor decreasing χ^2 is that we limit the "theoretical" bin count to a minimum of 1. The inverse of the bin count is the weight of the bin in the χ^2 parameter, and tiny bin counts turn into unrealistically large weights. This lower limit also has the effect of slightly lowering χ^2. I removed this limit as a test, but it only increased χ^2 to 0.65, a negligible change. Another slight discrepancy is that the theoretical model is fit only at the bin centers, which tacitly assumes the bin center is a reasonable representative of the bin average. In regions of high curvature, compared to the bin width, this may not be a good assumption. I did not investigate the consequences of this, since as shown below, the final impact of the APD parabola is negligible.

Now, adding the APD spatial dependence, which is roughly flat out to a radius of 8 μm, and then increases parabolically to 100 ps at 15μm (the outside edge), the fit detects a noticeable non-gaussian form:



While visually, the gaussian fit looks quite good, the reduced χ^2 parameter is significantly higher, at 5.39. In practice, approximating even this distribution as gaussian would probably be fine. However, the real situation is even better. The APD detection PDF also has a diffusion tail, and is cascaded with other measurements of the FPD and the TDC, both of which obscure the slight deviation from the APD spatial dependence even further. Considering just the FPD and TDC, we get our fiducial gaussian core σ of 120 ps. A simulation of these effects with and without the APD spatial dependence reveals an insignificant difference, and a reduced χ^2 of 0.92:



Lunar detections are wider, still, and therefore are not shown. A core σ of 250 ps yields a reduced χ^2 of 0.96.

5.5 Anomalies in Data Reduction

The following runs of actual data gave anomalous results, which were often excluded from the final analysis in this dissertation. This list will enable future examination of these anomalies. These are only the anomalies I noticed; there are certainly more.

- From at least June through August 2006 the laser output had double (and sometimes triple: 060812-120912) pulses. My algorithms, both PDF-fit and AC, don't accommodate this. Double pulses come in all forms: sometimes both about equal strength, sometimes the first much stronger, sometimes the second.
- The year 2008 produces only about 80 runs that successfully analyze with 'apalyze'. The reason for apalyze rejecting so many is not known. 2007 produces about 200 successful analyses.
- The six runs on 071019 have back porches on fiducial profiles, that prevented good fits. It is not known if any other runs also have this feature.
- In 2007, starting around run number 150 (of ~180), the t_frac fit parameter gets noticeably more variable. See fiducial fit plot in text.
- Runs from 070619-025340 through 070622-025015 have bad t_frac ~0.5, and associated bad t_decay. Note that the first 2 runs of 070619 seem normal (070619-023218 and 070619-024307).
- 070709-110533.run, 070709-112540.run, 080118-065038.run and 080330-120150.run have bad fiducial fits.
- 2008/080101/080101-134634 doesn't get a good PDF-fit for LUNs.

5.6 Data Reduction Program: Apalyze

Short for "APOLLO analyze," apalyze.exe is a C++ program that analyzes a single run file of

data. apalyze.cpp (APOLLO analyzer) is a C++ program, relying on several libraries, to read a *.run file, and produce a normal point from it. Apalyze supports a large set of command line options, tests, and debugging features. A sample invocation is:

```
>apalyze 080101-180101.run opt=0x80
```

Apalyze can be invoked by the EXECUTE utility to batch-analyze any number of run files with one command:

```
>execute apalyze [r:2008\*\*.run] opt=0x80
```

where the square brackets surround the file name pattern that EXECUTE expands. For each file found, EXECUTE invokes apalyze with that file name in place of the bracketed pattern. In other words, EXECUTE performs the "globbing," so apalyze can be simpler. EXECUTE provides a rich set of file-selection capabilities, including limiting the dates, and file sizes, selecting multiple file patterns in one command, and excluding files matching a separate pattern. The output is similar to this, as of 5/20/2009:

```
---- APOLLO Analyzer May 18 2009 13:56:26.  ****** Uncalibrated TDC
******
Reading run file r:sim\fivek.run
44 parameters recognized
FPB: Chosen FID rate, bias: 2.01  -0.007 ns, applied
Working unified FID fit, relative to maxwin:
  fid_core_avg (rel to REF0) = -0.072
0 prediction differences more than 1, 0 of 1
671 backgrounds in 20 ns -> 559166 /s/apd
FPB Max-window from [ 0.275 to  2.275] relative to prediction
FPB # points = 2766
FPB correction stats, applied:
  n=6388, avg dev [min, max]: -1.583e-11  1.425e-11 [-8.051e-11, 0]
672 backgrounds in 20 ns -> 560000 /s/apd
664 backgrounds in 20 ns -> 553333 /s/apd
Trimmed FPB correction stats,  applied:
  n=2298, avg dev [min, max]: -2.309e-11  1.697e-11 [-8.051e-11, -
5.854e-12]
# pts in NP / maxfid = rate: 2298 / 5000 = 0.46
NP: time(h)    time(s)    RTT(s)  uncert(ns) npts, pred-dif zwidth:
 0.034722222   125.000  2.475000001016 0.006 2298,   1.016 3.7

PDF-fit method:
672 backgrounds in 20 ns -> 560000 /s/apd
pf_np = 2769
671 backgrounds in 20 ns -> 559166 /s/apd
NP: time(h)    time(s)    RTT(s)  uncert(ns) npts, pred-dif
 0.034722222   125.000  2.475000000968 9.999 2769,   0.968
Error flags: 0, Option flags: 1. Allowed dets/shot: 99
End of Wed May 20 13:04:00 2009: r:sim\fivek.run
```

TDCFIT Program

The tdcfit program creates TDC calibration files from the *.cal data files. 'housctl' creates

*.cal data files before and after each run, to provide calibration data for the TDC. We fit a parabola to the calibration data, and use that parabola to interpolate the actual times from the TDC values.

'tdcfit' can process multiple runs from multiple directories in a single command (in other words, it has built-in "globbing." Therefore, tdcfit does not need EXECUTE.):

```
>tdcfit ...*.run
  No cal file for C:\LoBackup\EricSchool\Research\data\2006\060620\060620-
  003728.run
  ...
  Created C:\LoBackup\EricSchool\Research\data\2006\061129\061129-
  002708.tdc
  Created C:\LoBackup\EricSchool\Research\data\2006\061129\061129-
  003232.tdc
  ...
  Created C:\LoBackup\EricSchool\Research\data\2007\071019\071019-
  012132.tdc
  Created C:\LoBackup\EricSchool\Research\data\2007\071019\071019-
  012708.tdc
    40 file(s) fit.
  23 errors in processing
```

5.7 Glossary and Acronyms

accuracy — systematic error

APD — Avalanche Photo-Diode: a single-photon detector.

APO — Apache Point Observatory

APOLLO — The Apache Point Lunar Laser-ranging Operation. Note distinction from "Apollo."

Apollo — The NASA mission to put men on the moon. Note the distinction from "APOLLO."

CDF — cumulative distribution function: gives the probability that a random variable X is less than a given value, x: CDF(x) = Pr($X < x$). Compare to PDF.

Dev(X) — standard deviation of random variable X. $Dev(X) = \sqrt{Var(X)}$.

drift — the motion of an electron or hole being pushed by an electric field. It is *not* random wandering.

FID — fiducial photon detection.

FPB — First Photon Bias: the time-offset of any single-photon detector from the average ("true") time for all detectable photons in the measurement interval, due to the detector triggering on the first of multiple photons. FPB is always negative, i.e.

	always skewed toward earlier times.
FPD	Fast Photo-Diode: the fast diode that measures the time of the outgoing laser pulse with highest precision.
FIR	Finite Impulse Response: a digital filter whose output is a weighted average of adjacent input values. It's response to an input impulse is of *finite* length. Contrast with IIR.
fit criterion	a number or other indication that describes *how* the fit is to be done.
FOM	figure of merit: a measure of the error of a curve fit (or other process), which is optimized (minimized) by varying parameters in the fit process.
histogram	is made of actual measurements; see also PDF.
IIR	Infinite Impulse Response: a digital filter whose output is a weighted average of adjacent input values, plus some prior output values. It's response to an input impulse is of infinite length (neglecting roundoff). Contrast with FIR.
ILRS	International Laser Ranging Service, http://ilrs.gsfc.nasa.gov/ .
LUN	lunar (signal) photon detection, after the photon has gone to the moon and back.
normal point	a single hypothetical launch time, its associated round trip time (RTT), its uncertainty, and associated ancillary data. A normal point is computed by aggregating the RTTs measured on dozens to thousands of detected photons in a run.
PDF	probability distribution function: gives the probability that a random variable lies in the range (x, $x + dx$). A histogram can be compared to a PDF, and tested for goodness of fit. Our detection PDF is a calculated function based on statistical theory and empirical models.
precision	random error: high precision means small random error.
random error	errors that "average down" to a more accurate value with repeated measurements.
run	a continuous sequence of laser shots and measurements. A run is typically 5000 to 20,000 shots, spanning ~4 to 18 minutes. To date, each usable run produces a single

normal point.

session — a sequence of runs within a short time span, typically ~1-2 hours, allowing ~3-12 runs. We usually are allocated only one session in a night.

significand — the multiplier bits in a floating point number stored in a computer, which determines the precision. In IEEE floating point, the significand has a value from 1.0 to just under 2.0.

systematic error — errors that do not "average down" with more measurements. In APOLLO's case, there are significant systematic effects that vary from session to session, which must be carefully controlled to avoid systematic error.

TDC — Time-to-Digital Converter: the highest-resolution interval timer in APOLLO, with ~25 ps resolution.

Var(X) — variance of random variable X. $Dev(X) = \sqrt{Var(X)}$

5.8 References

[Bat] Battat, J. B. R., T. W. Murphy, Jr., E. G. Adelberger, B. Gillespie, C. D. Hoyle, R. J. Mcmillan, E. L. Michelsen, K. Nordtvedt, A. E. Orin, C. W. Stubbs, and H. E. Swanson, "The Apache Point Observatory Lunar Laser-ranging Operation (APOLLO): Two Years of Millimeter-Precision Measurements of the Earth-Moon Range," Publications Of The Astronomical Society Of The Pacific, 121:29–40, 2009 January.

[Bev] Bevington, Philip R., and D. Keith Robinson, *Data Reduction and Error Analysis*, 3rd Ed., McGraw-Hill, 2003.

[Car] Carroll, Sean, *Spacetime and Geometry: An Introduction to General Relativity*, Benjamin Cummings, September 28, 2003.

[JW] Williams, J. G., X. X. Newhall, and J. O. Dickey, "Relativity Parameters Determined from Lunar Laser Ranging," Phys. Rev. D 53, 6730-6739 (1996).

[JW2] Williams, James G., Slava G. Turyshev, Dale H. Boggs, "Progress in Lunar Laser Ranging Tests of Relativistic Gravity," Phys.Rev.Lett.93:261101,2004. DOI: 10.1103/PhysRevLett.93.261101, arXiv:gr-qc/0411113v2.

[Kap] Kapner, D. J., T. S. Cook, E. G. Adelberger, J.H. Gundlach, B. R. Heckel, C. D. Hoyle, and H. E. Swanson, "Tests of the Gravitational Inverse-Square Law below the Dark-Energy Length Scale," Physical Review Letters, 98, 021101 (2007). DOI:

10.1103/PhysRevLett.98.021101.

[Mur] T. W. Murphy, Jr., E. G. Adelberger, J. B. R. Battat, L. N. Carey, C. D. Hoyle, P. LeBlanc, E. L. Michelsen, K. Nordtvedt, A. E. Orin, J. D. Strasburg, C. W. Stubbs, H. E. Swanson, and E. Williams, "The Apache Point Observatory Lunar Laser-ranging Operation: Instrument Description and First Detections," Publications of the Astronomical Society of the Pacific, 120:20–37 (2008). DOI: 10.1086/526428.

[Nor] Nordtvedt, Kenneth, "Lunar Laser Ranging - a comprehensive probe of post-Newtonian gravity," arXiv:gr-qc/0301024v1 (2003).

[NRC] Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, Numerical Recipes in C, The Art of Scientific Computing, 2nd Ed., Cambridge University Press, 1992.

[Str] Strasburg, Jana Dee, Characterization of Avalanche Photodiode Arrays for Temporally Resolved Photon Counting, PhD dissertation, University of Washington, 2004.

[Wil] Will, Clifford M., *Theory and Experiment in Gravitational Physics*, Cambridge University Press, 1993.