Adaptive Delay Compensation Algorithms for the Optical Array Receiver

C.-W. Lau,[1] V. Vilnrotter,[1] and M. Srinivasan[1]

A delay compensation algorithm is developed for optical array receivers. This algorithm is designed to compensate for slowly varying delays due to turbulence and to combine the delay-compensated signals from all elements of the array. First, algorithm performance is evaluated through the use of simulated data. Next, the algorithm is used on experimental data, and it is shown that significant gains are possible by combining the signals from the two telescopes. Finally, the algorithm is validated using a combination of real and simulated data to determine performance limits.

I. Introduction

With optical arrays fast becoming a reality, finding efficient algorithms for combining and delay compensation becomes necessary. The premise behind an optical array lies in splitting a single aperture into a number of separate apertures, or telescopes, each with its own array of detectors in the focal plane. Each telescope detects a portion of the transmitted signal and sends the samples obtained from the detected signal to a central processing station. The central processing station then applies synchronization and detection algorithms to reconstruct the original signal.

The essential difference between a single-aperture optical communications receiver and an optical array receiver is that a single-aperture receiver focuses all of the light energy it collects onto the surface of an optical detector before detection, whereas an array receiver focuses portions of the total collected signal energy onto separate detectors, optically detects each fractional energy component, and then combines the electrical signals from the array of detector outputs to form the observable, or decision statistic, used to detect the signal [1,2].

A conceptual block diagram of an optical array receiver suitable for long-range free-space communication is shown in Fig. 1. Once the electrical signals from each array element have been collected at some central processing station, the signals can be recombined to recreate the original signal. In order to effectively recombine the signal, the central station must synchronize the individual signals, compensate for the delay between each array element, and perform demodulation and decoding on the reconstructed signal.

---

[1] Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

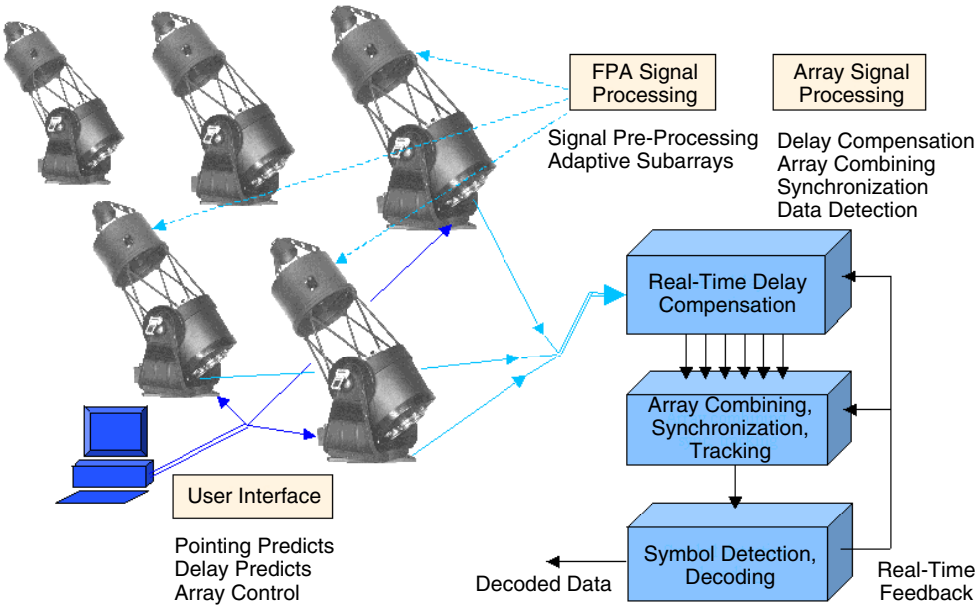

Fig. 1. A conceptual design of an optical array receiver, illustrating key functions required for optical array reception. The picture of the 25-inch telescopes is courtesy of JMItelescopes, Colorado.

The time synchronization of the signals from each of the array elements is performed in two stages. Large delays due to geometry can be calculated and compensated for ahead of time, leaving the residual slowly varying delay drifts between signals which occur on the order of a fraction of a slot duration (nanosecond level). These residual delays are tracked and removed by use of a delay compensation algorithm that requires slot boundaries and an estimate of the signal slot location. Depending upon the received signal strength, the initial slot clock may be derived from an individual antenna's signal, using either a narrow bandwidth closed-loop slot synchronizer or even an open-loop correlator if there are embedded tracking symbols inserted into the transmitted data. Once the slot boundaries are known along with an estimate of the signal slot position, the individual signals can be combined and the output combined signal sent to the receiver.

Following delay compensation and combining, the software receiver refines the slot synchronization using the combined signal and forms sufficient statistics from the received combined samples, which in this case reduce to sums of sample counts over each pulse-position modulation (PPM) slot. PPM symbol synchronization and code frame synchronization then are performed via the use of embedded tracking symbols and frame headers. Finally, the necessary slot statistics for each symbol are passed on to the decoder. In general, hard decisions on the PPM symbols may be needed as well, to provide signal-slot information to the combining algorithm. The optimum uncoded PPM detection algorithm compares the slot counts over each PPM symbol and selects the symbol corresponding to the greatest slot count [3].

A block diagram of the slot synchronizer and PPM detector portion of the receiver is shown in Fig. 2. The PPM signals, delayed by some timing error Δ, are passed to the top branch that forms the integrated slot statistics and selects the slot corresponding to the maximum value. In the lower branch, the signal is delayed by a symbol duration and is passed to the error detector, which multiplies it by a square wave at the slot frequency and integrates over the slot duration. This signal is then gated by the slot selector from the upper branch in order to create the timing error signal, which is filtered and then passed to the numerically controlled oscillator (NCO), which then outputs the slot clock. The PPM symbol decision is passed along with the slot clock back to the combiner, while the integrated slot statistics are passed on to the decoder. It is also conceivable that at high enough data rates PPM symbols fed back from the decoder might be used to improve the performance of the combining algorithm.

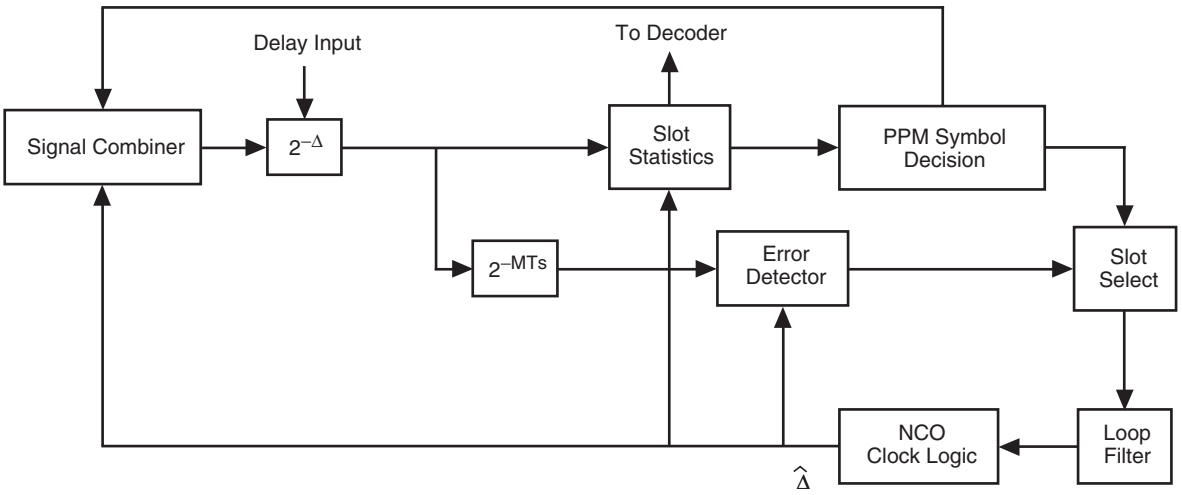

Fig. 2. Slot synchronizer and PPM detector.

The focus of this article is the delay compensation algorithm, by which the central processing station compensates for the individual signal delays and combines the individual signals from the array elements. In the following sections, the algorithm will be described, followed by a detailed performance analysis incorporating both simulated and experimentally obtained data.

II. Delay Compensation Algorithm

This algorithm was developed specifically to work with PPM signals detected with photon-counting detectors and is intended to compensate for slowly varying delay drift, ensuring that PPM pulses received from each telescope are properly aligned in time and synchronized with the receiver clock before combining. It is assumed that the slowly varying delay drift translates into delays of at most one slot length. The delay compensation algorithm takes as input the electrical signals from each of the array elements. In addition, other inputs required by the algorithm include the PPM order, the number of samples per PPM slot, the step size μ, an estimate of the average number of signal photons per PPM symbol per telescope, the average number of noise photons per PPM slot per telescope, and estimates of the PPM signal slot for each symbol.

The adapted delays for the ith telescope are formed according to an algorithm that resembles the well-known least mean square (LMS) and constant modulus algorithm (CMA) often employed to "phase up" conventional radio frequency arrays [1], except that here the delay adaptation is based entirely on the observed photon counts. Denoting the PPM slot number by n and the delay variable by τ, the delay update algorithm for the ith telescope is of the form

$$\tau_i(n+1) = \tau_i(n) - \mu \left(\sum_{j \in \text{slot}} y_j(n) - y_o \right)^2 \left[\sum_{j \in \text{slot}/2}^{\text{slot end}} s_{ji}(n) y_j(n) - \sum_{j \in \text{slot start}}^{\text{slot}/2} s_{ji}(n) y_j(n) \right] \tag{1}$$

where j is the index of the sample within the nth slot, $y_j(n)$ is the summed photon-count output of all telescopes over the jth sample of the nth slot, y_0 is the desired combined average photon count for perfectly combined PPM sequences (usually determined from predicted or measured received signal power levels), and μ is the step size that controls the rate of the algorithm's convergence and the degree of smoothing. The error signal is seen to be the difference of correlations of the received photon-count samples and the combined photon-count samples from the first and second halves of each slot. The cost

function is seen to be the difference between the summed photon-count output of all the telescopes over the nth slot and the desired combined average photon count. When the cost function approaches zero, the algorithm stops updating and continues to supply time-aligned combined samples to the receiver for further processing.

The combining algorithm implemented in software finds an estimate of the gross delay by averaging the delay estimate over a block (a user-defined number) of symbols. It then applies this estimate to the rest of the symbol stream and proceeds to estimate the delay again for the next block of symbols. This process is repeated until all the symbols have been processed.

A graphical user interface (GUI) display was developed to control and visually show the algorithm performance in real time. The algorithm GUI displays the input parameters, estimated and ideal delays, cost function, and the combined signal level for each signal slot. An example of the GUI display is shown in Fig. 3. Algorithm inputs are shown in the upper-left quadrant. The lower-left quadrant shows the ideal telescope delays as red circles and the algorithm estimates as blue crosses. The upper-right quadrant displays the cost function. Ideally, if all the telescope delays are estimated correctly, the cost function should be close to zero. The lower-right quadrant displays the signal level for each signal slot. This display starts low and should approach the level corresponding to the number of telescopes times the mean signal photons per slot per telescope.

Three metrics were defined to measure the performance of the delay compensation and combining algorithms. These metrics were the root mean square (rms) delay error, mean cost function, and rms combining error. The rms delay error measures the accuracy of the delay estimates and is calculated as follows:

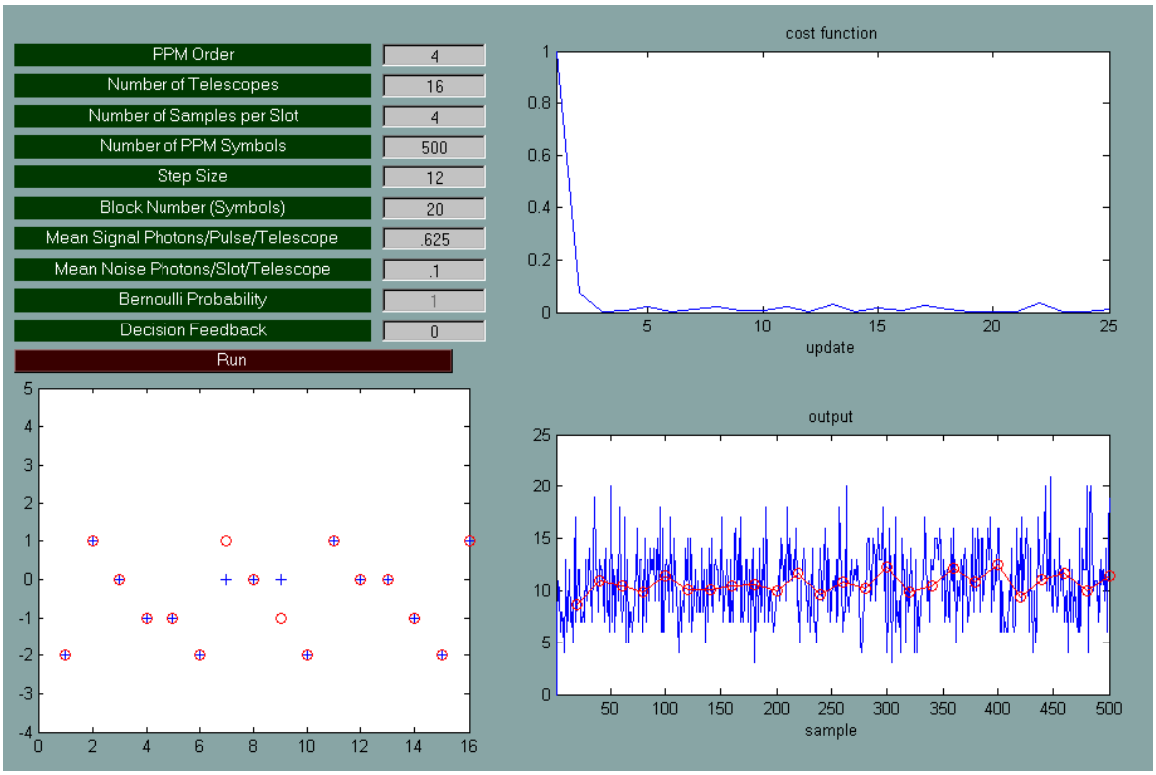

Fig. 3. Delay compensation GUI display, showing user-defined variables, true and estimated delays, cost function, and combined output for a hypothetical array of 16 telescopes.

$$\text{rms delay error} = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(d_i - \tau_i)^2} \tag{2}$$

There are L telescopes: i indexes the telescope number, τ_i is the estimated delay for telescope i, and d_i is the true delay for telescope i. Another metric of interest is the combining algorithm cost function, which indicates how quickly the algorithm converges and is obtained from Eq. (1) as

$$\text{mean cost function} = \left\langle \left(\sum_{j \in \text{slot}} y_j(n) - y_o \right)^2 \right\rangle \tag{3}$$

where $\langle - \rangle$ denotes time converged. The rms combining error measures how well the signal energy has been recombined and is defined as

$$\text{rms combining error} = \sqrt{\left| 1 - \frac{E(y|\text{signal slot})}{L\,(K_s + K_b)} \right|} \tag{4}$$

where K_s is the number of signal photons per pulse per telescope, K_b is the number of noise photons per slot per telescope, and y is the combined signal-slot output.

Algorithm performance was first tested using simulated data streams with artificially induced delays. A MATLAB program was first used to generate a random stream of PPM symbols. A copy of this symbol stream was delayed for each antenna in the array. The delay between telescopes was calculated based on a slot duration of 1 nanosecond, a linear array with elements 52 meters apart, a zenith angle of arrival of $\pi/3$ radians, and an azimuth angle of arrival of $\pi/10$ radians. These simulated symbol streams then were used as inputs to the delay compensation algorithm. The PPM order, number of telescopes, average number of signal photons per signal slot K_s, and average background photons per slot K_b were varied in order to measure the algorithm performance. Test results are summarized in Table 1.

The last 3 columns in Table 1 show the metrics defined in Eqs. (2) through (4), and the rest of the table lists the algorithm input parameters. It can be seen that the algorithm performs well for high values of input signal photons and no background photons. As the signal photons decrease, or noise photons are introduced, the performance degrades. The delay error behaves exactly as expected: as the input signal photons decrease, the delay estimates become less accurate and the delay error increases. The input noise photons K_b have a similar effect: as K_b increases from 0 to 0.1, the delay error correspondingly increases. The cost function also increases as the input K_s decreases or K_b increases. Table 1 shows the average cost function increasing inversely with K_s and proportionally with K_b. The combiner signal error also increases as a function of decreasing K_s or increasing K_b. Figure 4 visually shows the change in metric performance, as K_s and K_b are verified.

Figure 4(a) depicts the case with $K_s = 1.25$ and $K_b = 0$; this is the ideal case and corresponds to test number 1 in Table 1. As the figure shows, the cost function is very close to zero, and all the delay estimates match the ideal values. Figure 4(b) depicts the case when K_s decreases and corresponds to test 4 in Table 1. For this test, the input K_s level was decreased to 0.25 while keeping $K_b = 0$. The average cost function is no longer zero, and the delay estimates do not all match the ideal case. The algorithm still performs well, with only 3 delays remaining in error, but no more than 1 sample away from the ideal. Figure 4(c) shows the impact of introducing noise. This figure corresponds to test 6. The cost function is only slightly above zero. The delay estimates match the ideal case for the majority of the telescopes. Only one estimate is in error by 1 sample. Even for the case of 1 photon into the array, the algorithm is able to compensate for the majority of the telescope delays.

Table 1. Algorithm performance results with simulated data.

Test number	Number of telescopes	Samples per slot	Total number of PPM symbols	K_s/pulse · telescope	K_b/slot · telescope	Signal photons to array	Mean combined signal	RMS delay error	Average cost function value	RMS combining error
1	8	4	500	1.25	0	10	9.6811	0.2678	0.0070	0.1786
2	8	4	500	1	0	8	7.7176	0.3132	0.0072	0.1879
3	8	4	500	0.625	0	5	4.7575	0.3890	0.0106	0.2202
4	8	4	500	0.25	0	2	1.7475	0.8179	0.0300	0.3553
5	8	4	500	0.125	0	1	0.8505	0.9432	0.0572	0.3867
6	8	4	500	1.25	0.1	10	10.4385	0.2960	0.0075	0.1901
7	8	4	500	1	0.1	8	8.3588	0.3603	0.0087	0.2348
8	8	4	500	0.625	0.1	5	5.3223	0.6171	0.0185	0.3091
9	8	4	500	0.25	0.1	2	2.3522	0.8414	0.0329	0.4732
10	8	4	500	0.125	0.1	1	1.5781	0.9745	0.0518	0.4711
11	16	4	500	0.625	0.1	10	11.1771	0.3648	0.0117	0.2056
12	16	4	500	0.4375	0.1	7	7.6559	0.4160	0.0164	0.3672
13	16	4	500	0.3125	0.1	5	6.0499	0.5461	0.0151	0.3317
14	16	4	500	0.25	0.1	4	4.9502	0.5557	0.0181	0.4031
15	16	4	500	0.125	0.1	2	3.0324	1.1799	0.0377	0.5327

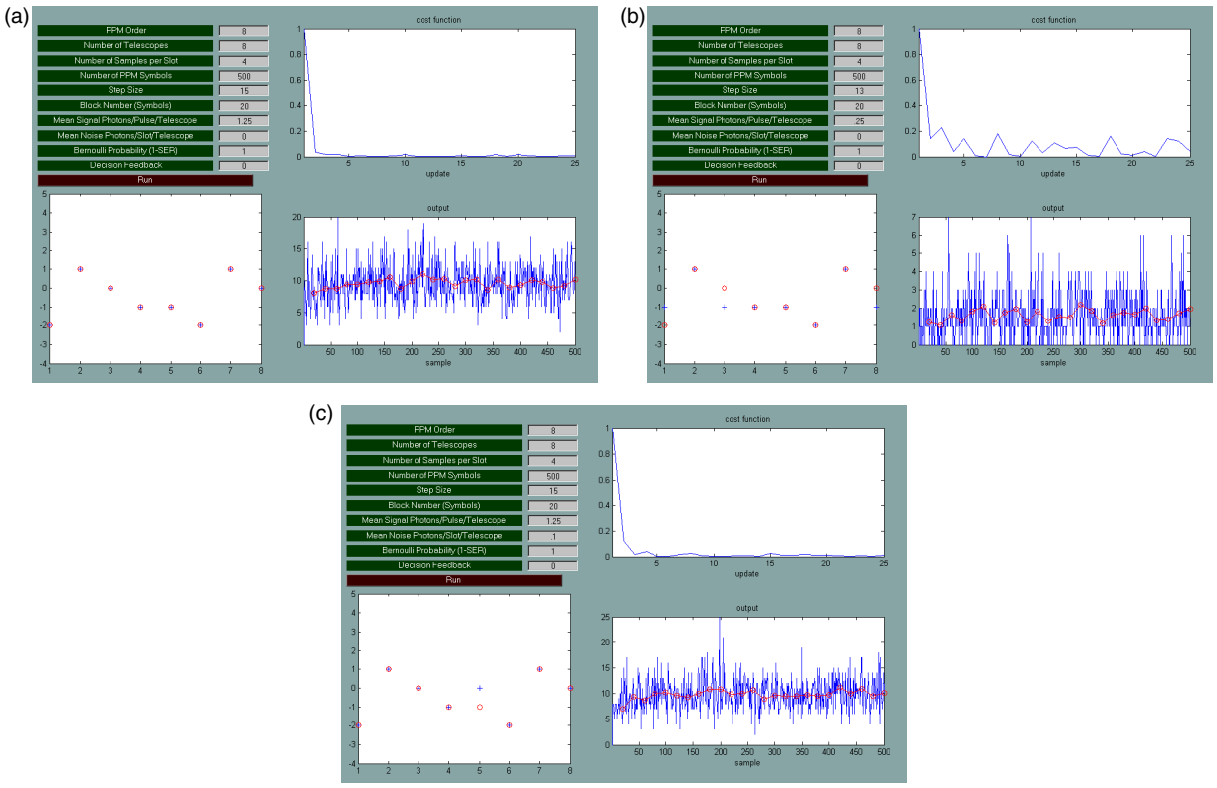

Fig. 4. GUI displays for several test cases: (a) $K_s = 1.25$ and $K_b = 0$, (b) $K_s = 0.25$ and $K_b = 0$, and (c) $K_s = 1.25$ and $K_b = 0.1$.

Once the algorithm performance was verified with simulated data, experimental test data were used to further evaluate performance. The data were taken in field experiments using two 25-inch (63.5-centimeter) computer-driven telescopes [4]. In the test field setup, both telescopes pointed towards a retro-reflector assembly approximately 64 meters away. The laser transmitter assembly was located on a table between the two telescopes and consisted of a 5-milliwatt (maximum power), 635-nanometer laser mounted in a fine-pointing bracket, and a pulse generator configured for 1-microsecond pulses repeating at 16-microsecond intervals. The laser was effectively gated "on" by the 1-microsecond pulses and "off" for the following 15 microseconds, simulating a simple repetitive 16-PPM symbol suitable for test purposes. Photons from the laser were detected by the telescopes, each equipped with a high-sensitivity geiger-mode avalanche photodiode (GAPD) detector array assembly custom built by aPeak, Inc. The streams of electrical pulses generated by each detector array were captured by high-speed logic analyzer assemblies manufactured by GoLogic, Inc., and stored as samples of 8-nanosecond binary data. Physical limitations on the experimental setup prevented delays between telescopes exceeding more than a few samples.

The experimental data were initially processed by a software receiver to obtain estimates of K_s and K_b. The receiver outputs then were used as inputs to the delay compensation algorithm. After combining, the combined signal was sent back to the receiver for symbol-error rate (SER) estimates. The GUI was modified to process experimental data. Since the ideal delay was unknown in the real data, the estimated and ideal delay portions of the GUI were removed; however, the algorithm itself remained unchanged.

Figure 5 shows the GUI display for one of the experimental data sets. Once again, the cost function rapidly approaches zero. For the combined channel output display, the green and magenta curves represent the individual telescope outputs, and the blue curve represents the combined output. The red curves represent their respective averages. Figure 5 shows that the blue curve is indeed the sum of the green and magenta curves, demonstrating that the algorithm correctly compensates for the delays and combines the signals from both telescopes without significant error. The average combined signal value was 20.96 photons. This is very close to the expected value of 21.07 photons based on the individual receiver outputs for each telescope. After the two symbol streams from each telescope were combined, the combined output was sent to the software receiver. In order to obtain non-zero symbol-error rates due to flooding of the detector arrays with photons, the laser output was attenuated using neutral density (ND) filters, and the process was repeated. Neutral density filters of values 0.3 and 1 were used.

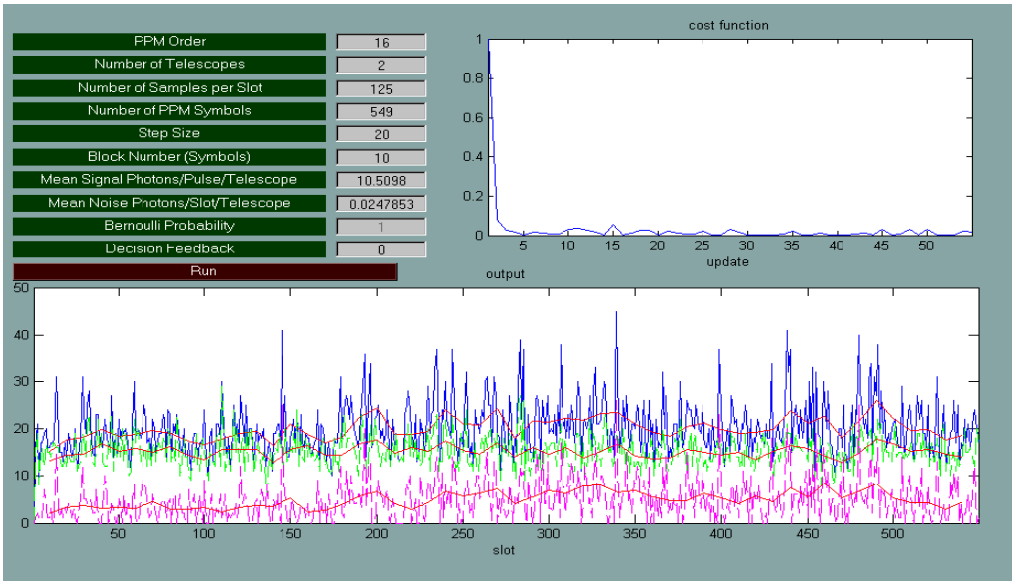

Fig. 5. Delay compensation algorithm using real data. This case represents two telescopes with $K_s = 10.51$ and $K_b = 0.02$.

The results of the experimental data are summarized in Table 2. Column 1 lists the ND filters used, and columns 2 through 7 list the statistics for each telescope as determined by the receiver. Column 9 lists the average cost function, and column 8 lists the symbol-error rate after the combined output was passed to the receiver. It can be seen that the cost function increases as the signal level decreases. It also can be seen that the combined output has a lower SER than either of the individual telescopes, as Fig. 6 depicts. Clearly, combining yields better performance than either of the telescopes alone.

Since delays of a significant fraction of a slot length could not be introduced in the field data, a combination of experimental and simulated data was used to measure the algorithm's delay compensation performance. Simulated input signals were generated and delayed similarly to the tests listed in Table 1. However, unlike the test cases in Table 1, the algorithm no longer had perfect knowledge of symbol slots. Errors in the symbol slot location were introduced artificially, so that a certain percentage of the time (user defined) the combining algorithm would use the "wrong" symbol slot to estimate the delay error. Table 3 summarizes the algorithm results for delay compensation and combining with non-zero symbol-error rates.

It can be seen that the delay error, cost function, and combiner signal error all increase as the SER increases. The mean combined signal steadily degrades as the SER increases. For SERs above 0.2, the algorithm cannot track at all, as is evident by the mean square error and cost function. Both values increase by an order of magnitude.

Table 2. Experimental algorithm results.

Test number	ND	Telescope 1 K_s	Telescope 2 K_s	Telescope 1 K_b	Telescope 2 K_b	Telescope 1 SER	Telescope 2 SER	Combined SER	Average cost function (algorithm)
1	0.0	1.1916	1.7757	0.1599	0.0458	0.7905	0.7359	0.5792	0.2137
2	0.3	0.4964	1.3782	0.0829	0.0444	0.8452	0.7996	0.6976	0.3106
3	1.0	0.0406	0.4085	0.0505	0.0432	0.9727	0.9271	0.9162	Cannot track

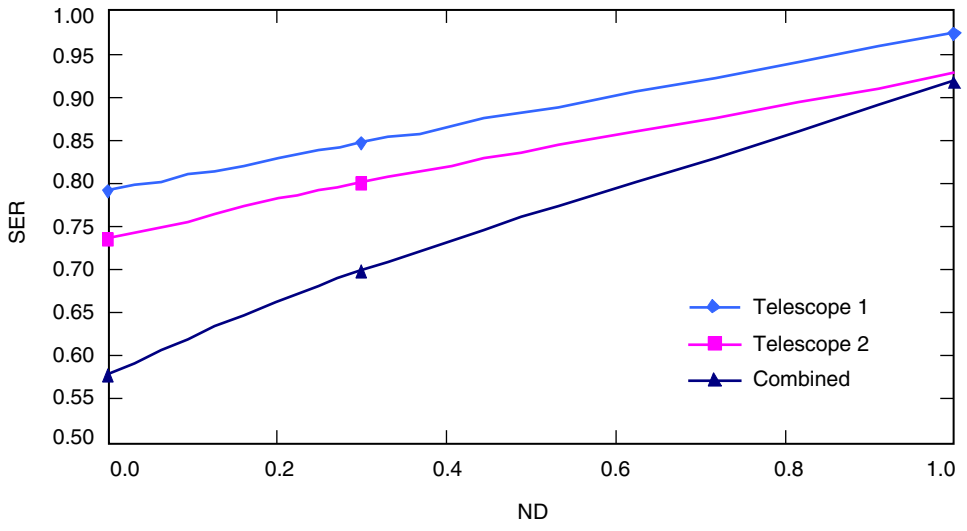

Fig. 6. Symbol-error rate performance as a function of ND filter value for experimental data.

Table 3. Simulated data results using symbol-error rates from experimental data.

Test number	Telescope number	Samples per slot	Total number of PPM symbols	K_s/pulse/ telescope	K_b/slot/ telescope	Signal photons to array	SER	Mean combined signal	RMS delay error	Average cost function value	RMS combining error
1	8	4	500	1	0.1	8	0	8.6844	0.3132	0.0072	0.1879
2	8	4	500	1	0.1	8	0.1	7.4784	0.4578	0.0268	0.3876
3	8	4	500	1	0.1	8	0.2	6.3887	0.4482	0.0930	0.5235
4	8	4	500	1	0.1	8	0.3	5.1395	1.5206	0.1930	0.6450
5	8	4	500	1	0.1	8	0.4	3.6678	1.7461	0.3351	0.7637
6	8	4	500	1	0.1	8	0.5	3.3223	2.0691	0.3860	0.7890

Figure 7 depicts the difference in GUI displays between the SER values of 0.2 and 0.3. Figure 7(a) depicts the case with an SER of 0.2, where it can be seen that the majority of the estimated delays match the ideal case. Figure 7(b) depicts the case where algorithm performance fails, with an SER value of 0.3. When the algorithm fails, the cost function becomes erratic, and the majority of the estimates are not close to the ideal values.

In summary, we found that the compensation algorithm developed for array detection of PPM pulses performs quite well both in the field and with simulated data. Performance gains in the field data have been shown in which the SERs of individual telescopes have been lowered through combining. Also, simulation data show the algorithm is able to handle an SER up to 0.3, although algorithm behavior in this region requires further investigation. Future plans involve acquiring more comprehensive experimental data sets to test the compensation and combining algorithms. In future versions, decision feedback also will be implemented to provide real-time symbol decisions for use by the array compensation algorithms. Finally, more comprehensive tests on the effects of symbol-error decisions on algorithm performance will be explored.

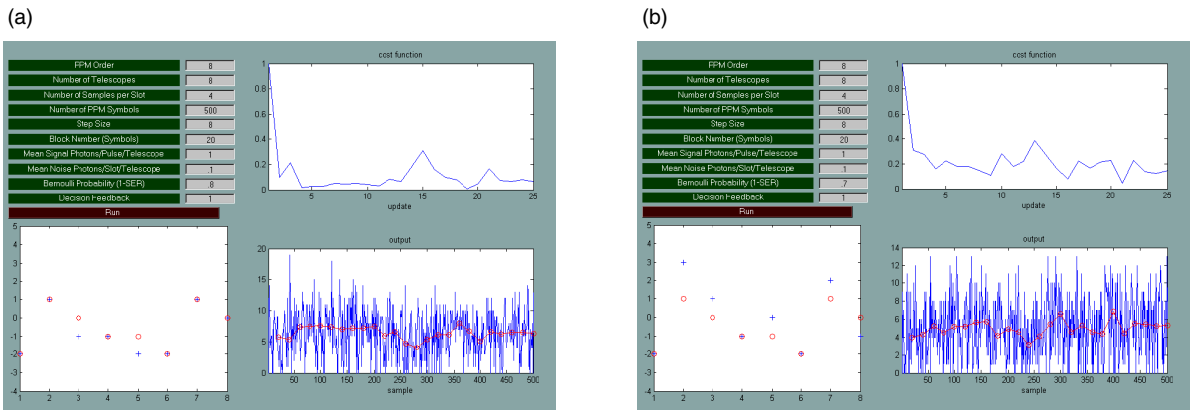

Fig. 7. Algorithm performance with (a) SER = 0.2 **and (b)** SER = 0.3**.**

III. Conclusions

In this article, the concept of a delay compensation algorithm has been described and its performance established. The algorithm performance has been tested with simulated, experimental, and a mix of data. The simulated data showed the algorithm performed as expected. Tests with experimental data show definite performance gains. Although more comprehensive tests with simulations and experimental data are needed, preliminary tests indicate the algorithm satisfies the needs of an optical array receiver.

References

[1] V. Vilnrotter, C.-W. Lau, M. Srinivasan, R. Mukai, and K. Andrews, "Optical Array Receiver for Communication through Atmospheric Turbulence," *IEEE Transactions on Lightwave Technology*, vol. 23, issue 4, pp. 1664–1675, April 2005.

[2] V. Vilnrotter, C.-W. Lau, M. Srinivasan, R. Mukai, and K. Andrews, "An Optical Array Receiver for Deep-Space Communication through Atmospheric Turbulence," *The Interplanetary Network Progress Report 42-154, April–June 2003*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–21, August 15, 2003. http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-154/154I.pdf

[3] V. Vilnrotter, C.-W. Lau, M. Srinivasan, R. Mukai, and K. Andrews, "Optical Array Receiver for Deep-Space Communications," *Proceedings of the SPIE*, vol. 5338, pp. 163–174, January 2004.

[4] V. Vilnrotter, C.-W. Lau, K. Andrews, and M. Srinivasan, "Two-Element Optical Array Receiver Concept Demonstration," *The Interplanetary Network Progress Report*, vol. 42-161, Jet Propulsion Laboratory, Pasadena, California, pp. 1–20, May 15, 2005. http://ipnpr/progress_report/42-161/161K.pdf