NANOGrav Summer Research Memo

Investigating the Impact of Slicing on Fitted Timing Model Parameters

Philip Andrews

with Michael Lam and Tim Dolch

August 2020

1 Introduction

There have been some larger, perhaps unexpected changes in pulsar parameters after adding additional years of observation into the data set, for example moving from the 11 year to 12.5 year NANOGrav data sets. The goal of this project is to investigate how pulsar model parameters change as more observation data is compiled over time. By simulating pulsars, data sets of greater length than is available from real world observations can be analyzed in order to answer questions such as whether model parameters approach the true pulsar values over time and how quickly they approach those values. Simulation also allows for the isolation of certain effects, such as the type of noise in pulsar data. The hope is that this analysis provides some guide to expectations on how and by how much, model parameters should change as more data is added to NANOGrav data sets.

2 Simulating TOAs

In order to examine datasets of varying lengths of years, the first step of this project involved simulating the Times of Arrival (TOAs) of a pulsar. Using libstempo which is a library for accessing TEMPO2 routines in python, TOAs of various observation length can be simulated. In this project, libstempo's toasim package allowed for simple TOA simulation. Simulation follows a process that is very much the reverse of typical pulsar timing. Rather than start with observed data and work towards a pulsar timing model made up of parameters, simulation takes the parameters as a starting point and works backwards to generate TOAs.

The first step in simulation was to create a parameter file for a pulsar. Rather than make up new pulsar parameters, the parameters for the pulsar J0740+6620 were used. Rather than include the more than four hundred parameters that make up a typical pulsar timing model, this project focused on only seven of the main parameters: Ecliptic Longitude, Ecliptic Latitude, Proper Motion of Ecliptic Longitude, Proper Motion of Ecliptic Latitude, Spin Frequency, Derivative of Spin Frequency, and Parallax. Below is the .par file used:

```
PSR J0000+0000
PEPOCH 50000.0
ELONG 103.7591360662907 1
ELAT 44.1024846763988 1
PMELONG -2.7479 1
PMELAT -32.4337 1
F0 346.5319964932128300 1
F1 -1.463885178981e-15 1
PX 0.5376 1
```

Next, the following function in libstempo.toasim created an idealized simulated pulsar from a given .par file with an observation length specified by the starting and ending MDJ. The observation period was set to every 14 ± 1 days and the error bars on each TOA was set to a common error of $10ns$. After injecting noise, the .tim file could be saved for the simulated pulsar.

Simulate Pulsar

```
psr = LT.fakepulsar(parfile=basename+".par", obstimes=np.arange(start_MJD,end_MJD,
    obs_period)+np.random.normal(0,1), toaerr)
```

Three different categories of noise were injected into the simulated pulsar, white noise, red noise, and gravitational wave background (GWB). The amplitude for red noise was $1.5e-15$ and $5e-16$ for GWB. By specifying different starting and ending MJDs, data sets ranging from one to thirty years of length in each noise regime could be created. In order to ensure that the injected noise remained the same in years shared by two data sets, a consistent seed value was used for each type of noise. This ensured that as more years of observation were added, the previous years remained unchanged.

Inject Noise

```
    if white_noise:
        LT.add_efac(psr,efac=1.0,seed=seed*7)
    if red_noise:
        LT.add_rednoise(psr,A=1.5e-15,gamma=3,seed=seed*7)
    if gwb:
        LT.add_gwb(psr,flow=1e-10,gwAmp=5e-16,seed=seed*6)
```

To visualize each type of noise and confirm that simulation was working, the residuals were plotted, examples of which can be seen in Figure 1.

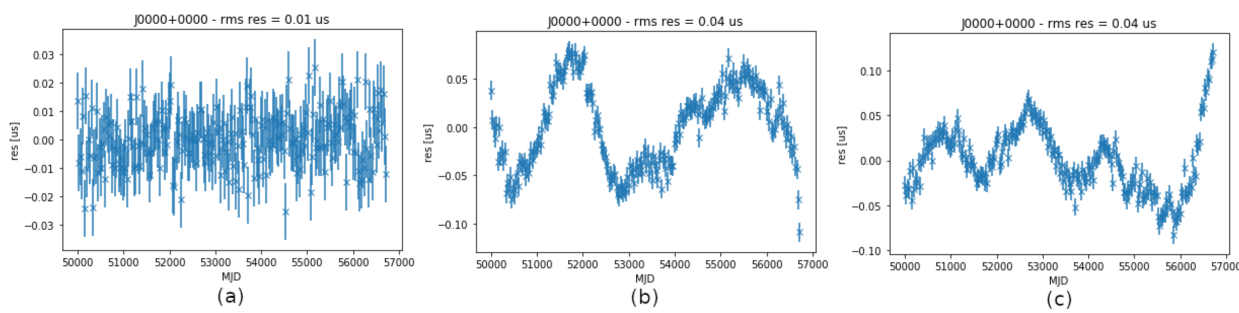

Figure 1: Examples of simulated residuals (a) White Noise (b) Red Noise (c) GWB

3 Calculating Fit Parameters

Armed with a process for creating TOAs of different observation length and noise, the simulated data sets were treated as if they were real observations. For each length of observation, the TOAS were modeled using TEMPO2's fit routine to obtain best fit values for each of the seven parameters. Saving the parameter values each time a new year of data is added to the TOAs allowed for the observation of how each parameter value changes over time. The hypothesis was that as the observation length increase, each parameter value should approach the known value, which specified in the .par file used to generate the simulated data (akin to some physically true value of a real world pulsar). This hypothesis was confirmed, and Figure 2 illustrates how one of the parameters does indeed approach the true value over time.

Note that fit values for the one year are excluded due to their highly sporadic nature interfering with visualizing the rest of the observation lengths. For all analysis from this point onward, only observation lengths ranging from two to thirty years were considered. Plots such as this were generated for each parameter in each of the three noise regimes and these full results can be found in the appendix.

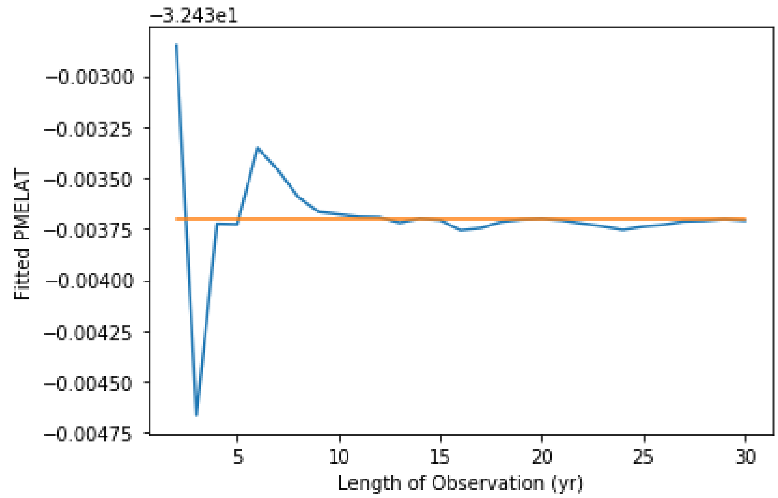

Figure 2: Example fit parameter approaching the known value

4 Standard Deviations of Parameters

The next step in this analysis involved investigating how consistently the fitted parameters behave when the noise and the residuals are different. Again for observation lengths from two to thirty years, thirty samples were computed, each with a different noise seed. Then the standard deviation of each parameter at each time slice was computed. The hope was that the standard deviations decreased over time which would indicate that there is less variance in the fitted parameter values when more data is available to fit. These standard deviations can be thought of as giving a sense of how the confidence interval of each fit parameter changes as more years of observed data become available. Almost always the standard deviations did decrease over time and an example plot is included in Figure 3. As expected, some parameters such as the spin frequency have a much lower absolute variance than parameters like parallax. Additionally, some parameters consistently approach the true pulsar value more rapidly than others which is indicated by the sharpness of the exponential trend.

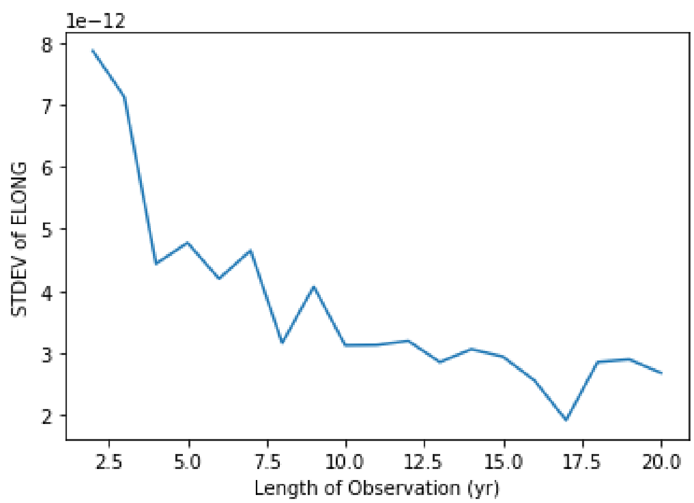

Figure 3: Example standard deviation over time

5 Curve Fitting

Beyond just visually examining the trends, each standard deviation was fitted with an exponential function of the form:

$$f(T) = Ae^{\frac{T}{\tau}} + C$$

Figure 4 illustrates one such fit. Note that because of the small amplitudes involved in the standard deviations of some parameters, the standard deviations were normalized before fitting for computational ease. The true amplitude of each fit function was saved, despite each plot displaying the normalized version. The characteristic time τ gives some sense of the timescale of how quickly the precision of each parameter increases as more observing years are added and allows for some numerical comparisons between fit parameters and noise regimes.

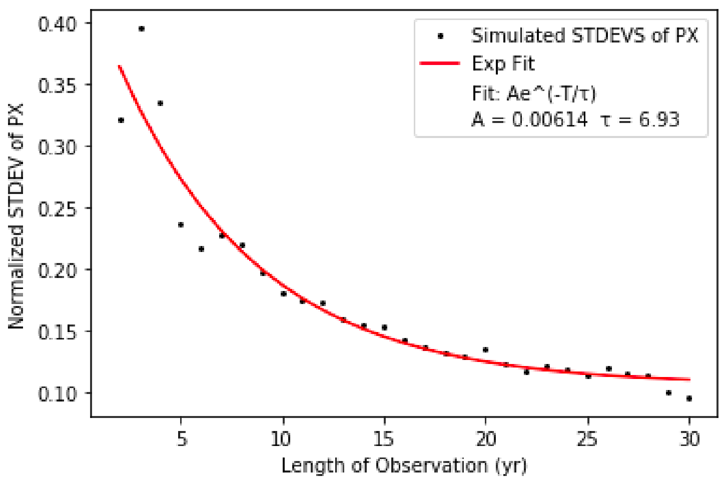

Figure 4: Example fitted standard deviation

6 Auto-correlation Functions

The traditional purpose of calculating the autocorrelation function of a signal is to detect non-randomness. Given measurements, $y_1, y_2, ..., y_n$ at time $x_1, x_2, ..., x_n$, autocorrelation function at lag k is defined as

$$A_k = \frac{1}{n} \frac{\sum_{i=1}^{N-k} (y_i - \bar{y})(y_{i+k} - \bar{y})}{\sigma^2}$$

In this application, the question of interest was whether each parameter approaches the true value smoothly from one direction or oscillates above and below the value. For example, if a fit parameter started above the true value and and asymptoted towards the true value without ever dipping below the true value, then the manner in which the fit parameter approaches the true value has a very low randomness. On the other hand, high randomness would mean that the fit parameter approaches the true parameter over time, but does not do so in a predictable way and bounces back and forth randomly. The autocorrelation hopefully provides some insight into this question for each parameter, so that jumps in model parameters from real world analysis has some quantitative context.

To achieve this, the ACF of each parameter function was calculated individually for each sample. Before calculating the ACF of each parameter function sample, each signal was normalized by the standard deviation fit function found earlier so as to remove the inherent greater strength of oscillations around the true parameter value at shorter observation lengths. Figure 5 shows the raw signals of one parameter, and after dividing each signal by the fit function, example normalized signals are shown in Figure 6.

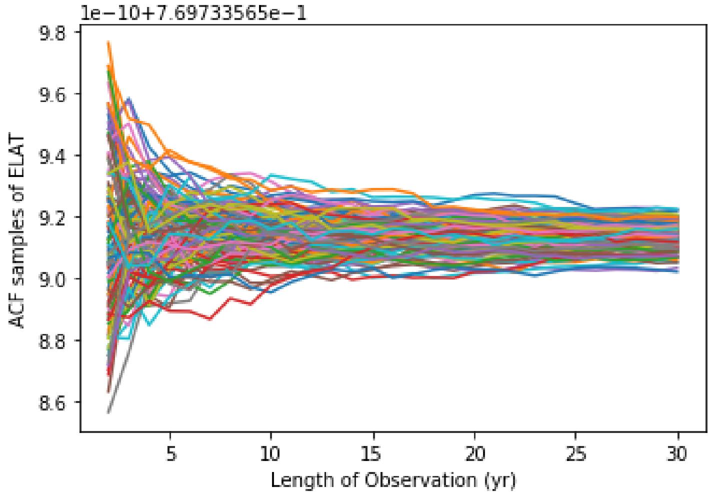

Figure 5: Example raw ACF signal samples

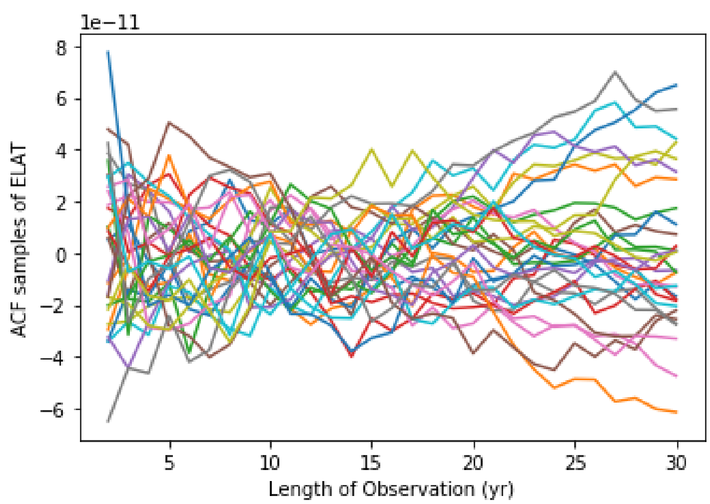

Figure 6: Example fit-normalized ACF signal samples

After calculating the ACF of each sample normalized signal (Figure 7), these results were averaged together to generate a representative ACF for each parameter in each noise regime (Figure 8).

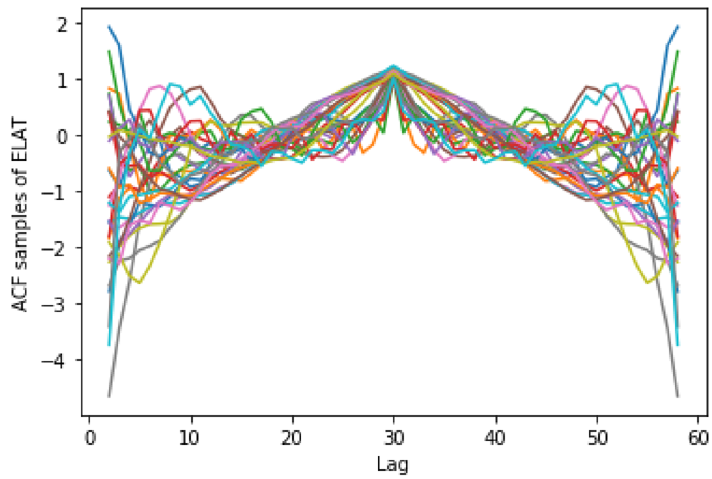

Figure 7: Example ACF samples

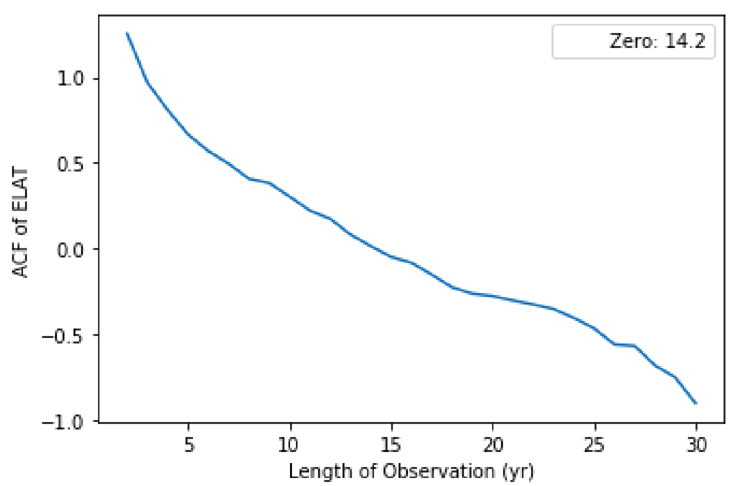

Figure 8: Example averaged ACF, ELAT with white noise

In the full results, the zero crossings of each ACF are included as a metric of interest in characterizing the width.

7 Conclusions

Overall, these results suggest that as a whole model parameters generally tend to approach their true values. While we have certainly observed instances where a model parameter diverges from its true value, in most cases the fit does converge. This can be seen in both individual parameter plots as well as the standard deviation plots in the Appendix. Nearly all of the standard deviations of each parameter decrease as more data is added in which indicates a convergence of fit. For most parameters the fit appears to converge exponentially. This means that typically the standard deviations decrease rapidly in the first five years of adding new observation and slow later on.

Beyond this simple trend, we found a clear trends in the effect each noise regime has on fit parameter convergence. As expected, fit parameters converged the fastest with just white noise.

Red noise and GWB noise behaved fairly similarly, but with slight differences. Red Noise fits tended to converge slightly slower than GWB ones, but their convergence also tended to be slightly more smooth and stable. These observations can be seen by comparing the magnitude and shape of each standard deviation plots in the Appendix Figure 12. Also note that parameters such as spin frequency and spin derivative have much nicer behaviors than more troublesome parameters such as parallax. This aligns with current findings and understanding within NANOGrav that suggested that parallax by nature is a more difficult parameter to fit, whereas spin frequency can be determined much more precisely.

The implications of these results are important in guiding decisions within other research groups such as NANOGrav for several reasons. One common question is whether or not data from recently discovered pulsars such be included in updates of pulsar timing arrays. These results make clear that any pulsar timing model built on only one or two years of observation data is questionably accurate. Additionally, looking at the results included here can help guide expectations for how much the accuracy of a pulsar timing array should be expected to improve over time. The time constants τ calculated from standard deviation fit functions suggests that deviations should be cut by a factor of $\frac{1}{e}$ every τ years (generally around 2 years for a majority of parameters and about 10 years for parallax).

Future work on this project would involve simulation larger sample sizes of pulsars which would hopefully smooth many of the standard deviation curves. Additionally, in calculating the ACFs, more work is required to normalize each signal before finding the ACF. Using an exponential function to fit each signal under the current method is a naive assumption and often artificially injects power into parts of the normalized signal since the exponential fit can be systematically inaccurate. Additionally, more exploration is required to characterize the effect of the strength of injected noise. With this information, accurate noise limits could be set using the latest research on the strength of the gravitational wave background.

8 Appendix/Results

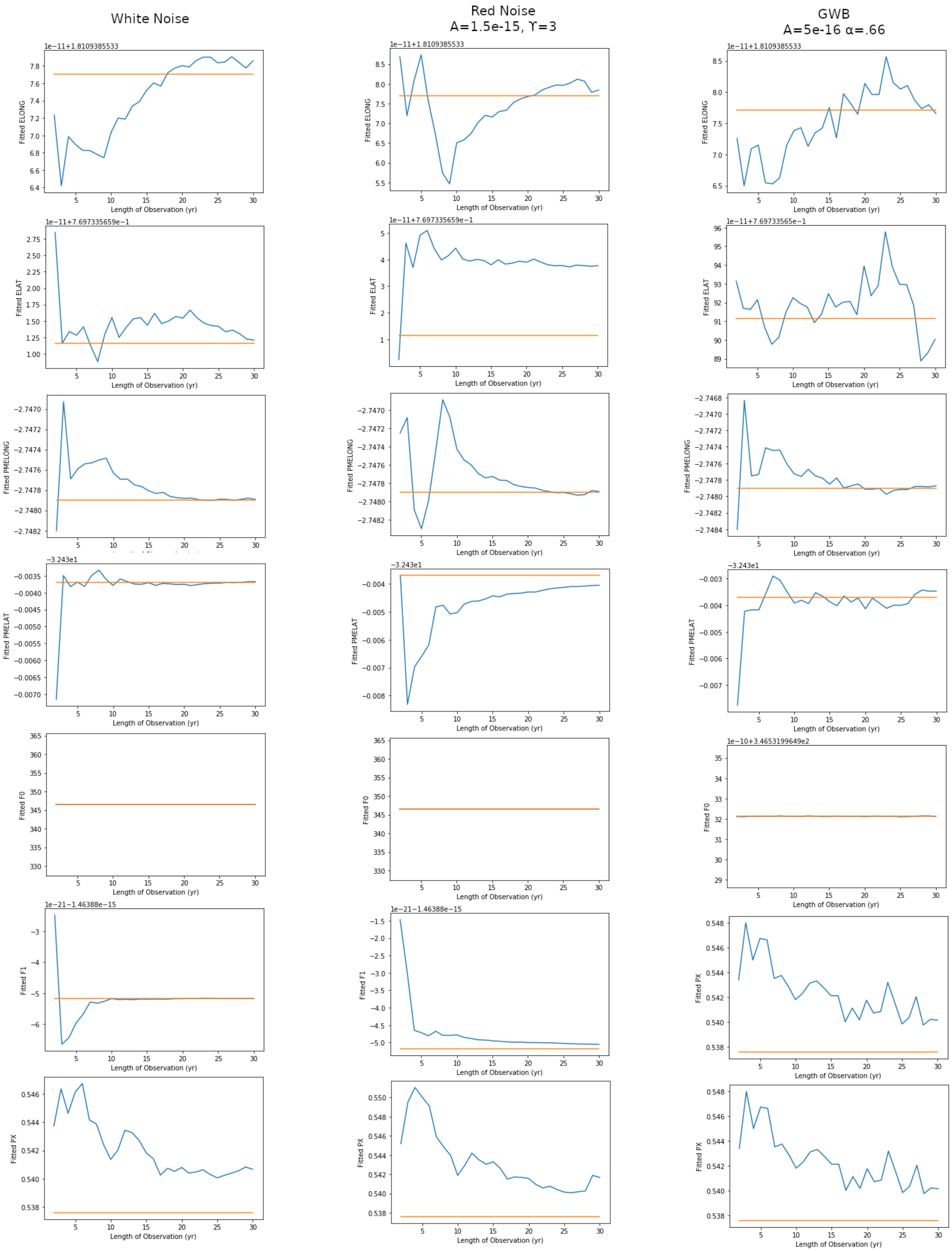

Figure 9: Parameter fits

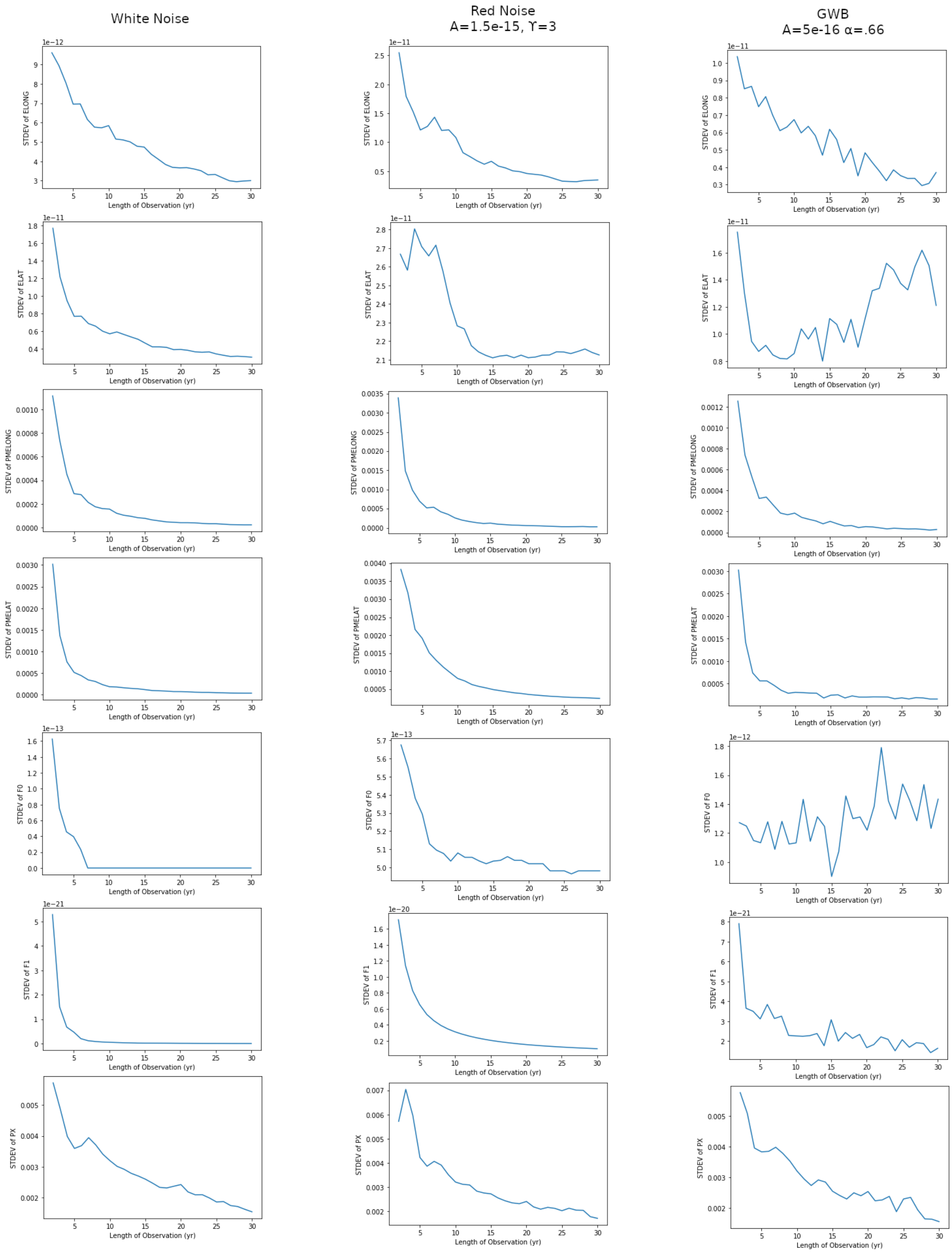

Figure 10: Standard Deviations

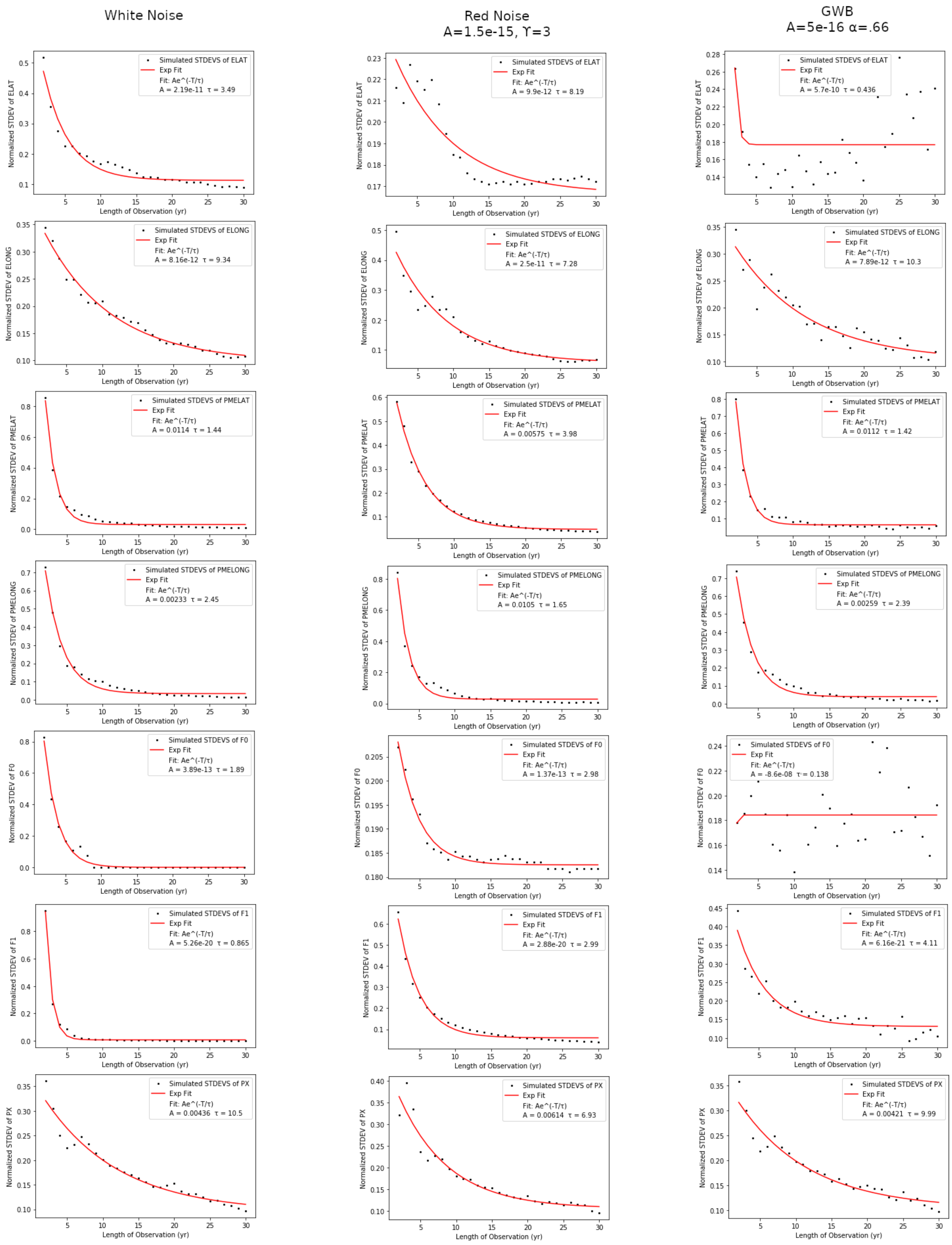

Figure 11: Fit Functions

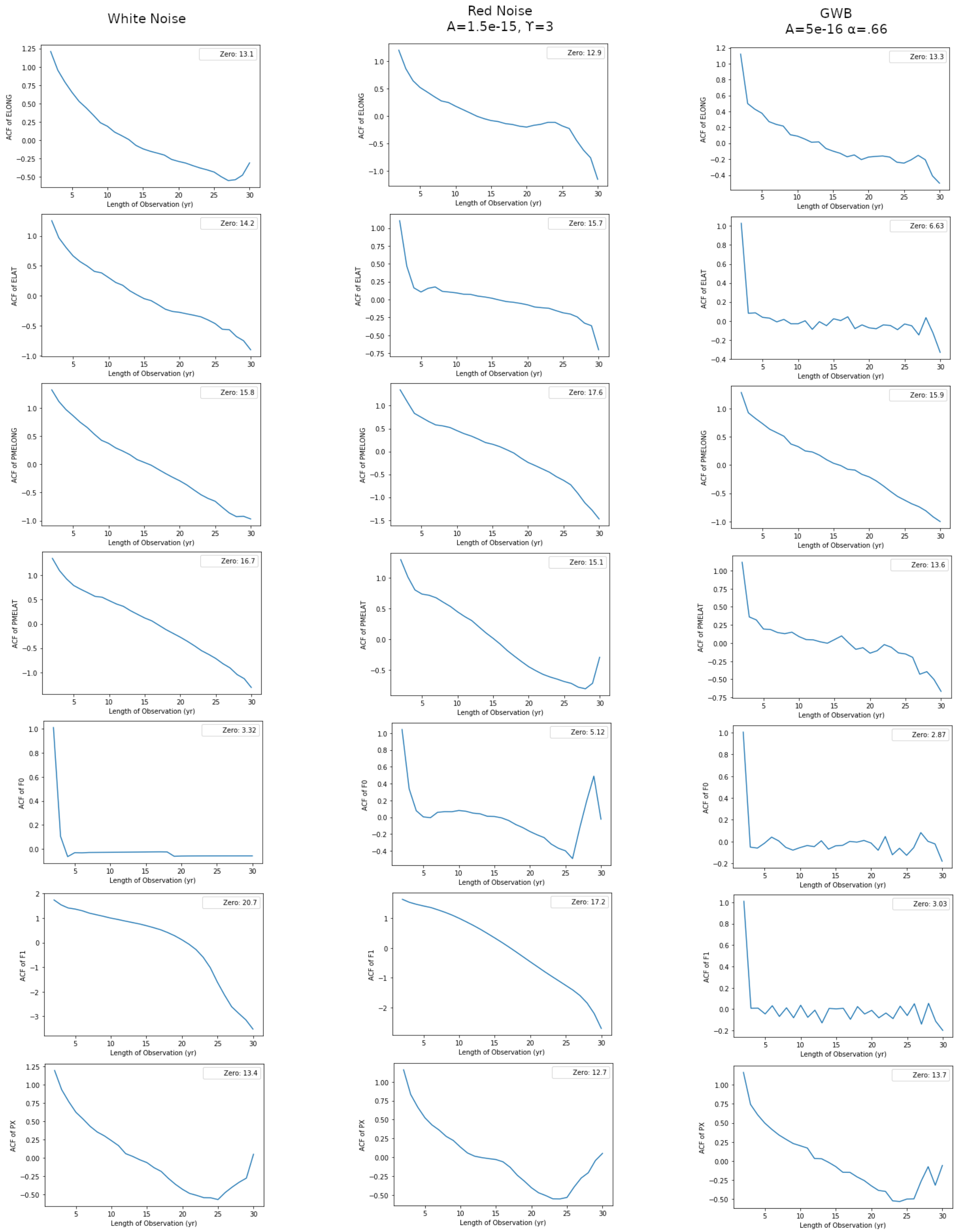

Figure 12: Autocorrelation Functions