

University of Nebraska - Lincoln

## DigitalCommons@University of Nebraska - Lincoln

---

Faculty Publications in Computer & Electronics Engineering (to 2015)    Electrical & Computer Engineering, Department of

---

12-2005

### An Ultra Wide Band Simulator Using MATLAB/Simulink

Peter Vial

*University of Wollongong*, [peter.v@uow.edu.au](mailto:peter.v@uow.edu.au)

Beata J. Wysocki

*University of Nebraska-Lincoln*, [bwysoki2@unl.edu](mailto:bwysoki2@unl.edu)

Tadeusz A. Wysocki

*University of Nebraska-Lincoln*, [wysoki@uow.edu.au](mailto:wysoki@uow.edu.au)

Follow this and additional works at: <https://digitalcommons.unl.edu/computerelectronicfacpub>



Part of the [Computer Engineering Commons](#)

---

Vial, Peter; Wysocki, Beata J.; and Wysocki, Tadeusz A., "An Ultra Wide Band Simulator Using MATLAB/Simulink" (2005). *Faculty Publications in Computer & Electronics Engineering (to 2015)*. 46.  
<https://digitalcommons.unl.edu/computerelectronicfacpub/46>

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications in Computer & Electronics Engineering (to 2015) by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

# An Ultra Wide Band Simulator Using MATLAB/Simulink

Peter James Vial, Beata Wysocki and Tad Wysocki  
University of Wollongong

*Abstract: Ultra Wide Band (UWB) is a promising technology for sensor networks, broadband wireless data access and location finding applications. This study outlines the development and validation of a single transmitter and receiver system across the multipath channel proposed by Saleh-Valenzuela. We have designed and tested a UWB simulator using MATLAB's Simulink in combination with the Real Time Workshop(RTW) Tool box for Simulink using the Fixed Step Discrete Solvers of RTW which is required to produce executable simulations on multiple computers (both Linux and Microsoft based devices). This paper outlines the basic design and modules chosen for the simulation and compares our results to those published in the literature. We found that our simulator provides similar performance to the reported results. We also found that an error floor occurs at high signal to noise ratio when the Saleh-Valenzuela channel is used and the assumption is made that the channel changes every symbol period. Using time traces from the simulation we show the mechanism behind this phenomenon.*

## I. INTRODUCTION

Ultra Wideband (UWB) is an area of interest to researchers interested in the areas of sensor networks and wireless broadband data access in particular. There are various forms of this proposed technology. Direct Sequence systems can be classified into Pulse Position Modulation (PPM) and Pulse Amplitude Modulation (PAM). Mathematical expressions have been provided in the published literature which include both types of systems in the same mathematical analysis, for example Yang & Giannakis does this in [1].

Our simulator is designed to consider only PPM so PAM systems will not be considered further here (see [1] for analysis). We also assume a system with perfect knowledge of the Channel State Information (CSI), where all the phases and gains of the channel are known. Since UWB signals are real, we only consider the real part or inphase component of the channel setting the imaginary or quadrature component to zero. We also limit the channel to 100 nano-seconds, truncating any further multipath that Saleh-Valenzuela channel may have generated to zero thus ignoring any possible Inter Symbol Interference (ISI). For the current version of the simulation we use only one Gaussian pulse to represent a symbol and the CSI changes for each such pulse. This is not typical in a real system where a channel is slow flat fading and changes approximately every 300 milliseconds. We do not implement any spreading or time hopping which will result in a very peaked and discrete power spectrum which would also not be implemented in a real system.

Section II describes the Gaussian pulse used and defines the signal and channel state information used in the Simulator. Section III describes how the channel state information was generated and what parameters were used in its generation. Section IV describes the developed Simulink model and how

MATLAB's Real Time Workshop (RTW) was used to generate monte carlo type simulations. Section V compares the results of our study to those found in Yang & Giannakis [1] and uses a time trace to explain the observed error floor inherent in high SNR values. Section VI provides conclusions and future work.

## II. UWB Signals and Channel

To develop a UWB simulator we first needed to define a pulse shape. In order to compare results we adopted the pulse shape seen at the receiver suggested in [1]. This pulse shape, called  $\omega(t)$ , was suggested by [1] to be the second derivative of the Gaussian function:

$$g(t) = \left( \sqrt{\frac{\tau^3}{3}} \right) \left( \frac{2}{\pi} \right)^{1/4} e^{(-t^2/\tau^2)} \quad (1)$$

Which we calculated to be:

$$\omega(t) = 2 \left( \sqrt{\frac{\tau^3}{3}} \right) \left( \frac{2}{\pi} \right)^{1/4} \left( \frac{1}{\tau^2} \right) \left[ \frac{2t^2}{\tau^2} e^{(-t^2/\tau^2)} - e^{(-t^2/\tau^2)} \right] \quad (2)$$

Here  $\tau = 0.1225 \text{ ns}$  which [1] points out gives a pulse width of 0.7ns. Figure 1 shows a plot of the squared value of the pulse, which was shown using Mathematica and MATLAB to have a unit area under the curve and hence unit energy (as suggested by [1]). Figure 2 shows a plot of the expression in Equation 2 which was the modeled received waveform  $\omega(t)$  [1].

We use the Saleh-Valenzuela channel model[2] which till relatively recently was accepted as a sound model for the UWB channel [3]. This was chosen to validate our simulator using results found in [1] where the Saleh-Valenzuela model was used.

In our model the channel multipath arrive in clusters within the observation window of 100ns. We have a cluster arrival

rate with Poissonian mean  $\Lambda$ , while within clusters we have a ray arrival rate with Poissonian mean  $\lambda$ . These values determine the time of arrival of each ray within the channel. The amplitude of each arriving ray is then chosen from a Rayleigh distributed random variable which is now dependent on which cluster the ray arrives in and then on another Rayleigh distributed variable within the cluster. That is, one Rayleigh variable is assigned which determines all the clusters amplitudes and the other is assigned on an individual basis within the arriving rays within the cluster. These are then denoted (in nano-seconds) as  $\Gamma$  for the cluster and  $\gamma$  for the individual rays within the cluster. These values determine the rate of decay of the amplitudes over time which conforms to results found in [2].

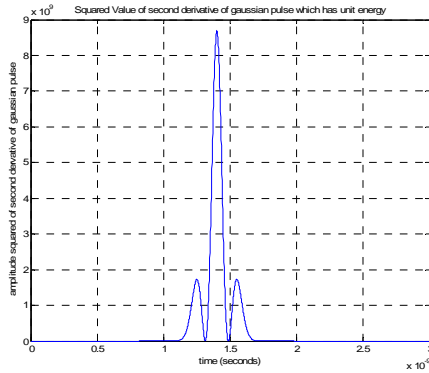


Figure 1: Squared value of second derivative of Gaussian pulse

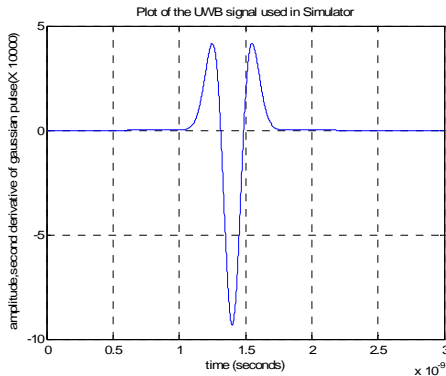


Figure 2: Time plot of second derivative of Gaussian pulse

MATLAB code sourced from [4] was modified so that it produced channel data which conformed to that for the Saleh-Valenzuela model suggested by [2] and used in [1]. We produced the real and imaginary components of the channel, but as UWB is a real signal we only used the real values of the channel as input to our simulink model. The

MATLAB program then generated many versions of the channel into an array which could be easily read at the start of the Simulink session and processed to create quickly the channel. This was done using embedded MATLAB. A snapshot of one channel is shown in Figure 3. To generate this data a threshold was selected which limited the number of possible multipath to between 100 and 300 paths. The number of multipath was further culled to 200 for our simulated channel data.

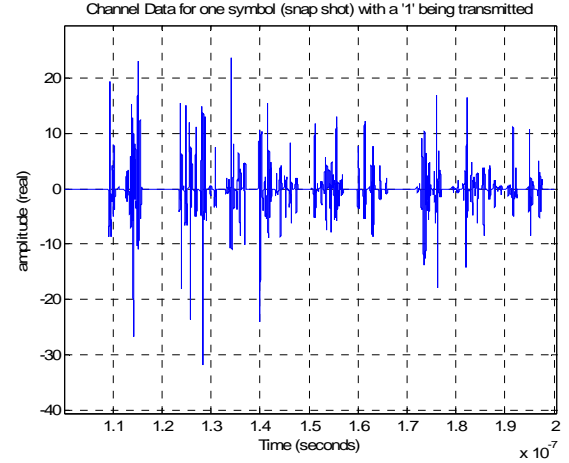


Figure 3: Snap shot of a single channel over 100ns

### III. Channel Definition and modeling

Modifying some MATLAB files from [4] we generated thousands of different channel state information sets, which were to ensure that the channel behaved as per the description in [1]. These channels were changed every symbol and there was no spreading or time hopping patterns used (simply peer to peer single user system was examined). This also means that the system was changing at a faster rate than would be expected in an actual UWB channel, where as a rule of thumb the channel changes every 300 milliseconds

Once the channel data set was generated and stored into an array with such information as time of multipath arrival with respect to the line of sight path and using a NLOS (Non Line of Sight) models as suggested in [1] and [2]. To generate this data we used the parameters  $\Gamma = 33 \text{ ns}$ ,  $\gamma = 5 \text{ ns}$ ,  $1/\Lambda = 2 \text{ ns}$  and  $1/\lambda = 0.5 \text{ ns}$ . These parameters were used by [1] for their results as per the description in [2]. The simulation of the channel was done with a sampling rate of  $1 \times 10^{12}$  samples per second so that one channel span of 100ns would have 100000 samples.

The array which was produced by the MATLAB code was then input via the Simulink block "Matrix Channel Data" which at the start of the simulation loads all of the matrix into Simulinks data space. This was followed by a Variable Selector and an embedded MATLAB function which

randomly chose a channel matrix to use for generating the channel. Figure 4 shows the technique employed to do this.

#### IV. Design of Simulink Model

The Simulink model uses the data from the channel matrix and embedded MATLAB to generate a sequence of pulses with different amplitudes but zero phase (for real UWB signals). Embedded MATLAB provides a cut down or reduced set of MATLAB functions. Using embedded MATLAB and RTW compilations it was possible to have multiple simulations with different seed values in a reasonable time frame. Also, we were able to access available clusters of computers to run multiple simulations in parallel without needing any MATLAB or Simulink licenses. This allowed the results reported to include 95% confidence intervals, as shown in Figure 6.

The Simulink Model itself used a section to generate the case where a '1' was sent and a section to generate the case

where a '0' was sent. In this simulation we used Orthogonal PPM with a spacing of 1ns to conform to the simulations in [1]. These were generated simultaneously even though only one of the two was transmitted. The energy of the signal and the energy of the noise as per the description in [1] was calculated. The Expected SNR was calculated based on the ratio of received signal strength (for all received paths) to the received noise power multiplied by channel gains. The system implemented a partial Rake using the  $L^{\text{th}}$  best arriving (useful) rays but rejecting rays that had multipath too close (within 1ns) of each other. A Maximal Ratio Combining (MRC) decision was made to decide for a '1' or a '0'.

In our model we assume that 100% of the power is detected by the receiver (in [1], it is assumed that 10% of the power is lost). We used the case where we had the template using '1' path ( $L=1$ ) and '4' paths ( $L=4$ ). Figure 5 shows some of the modules used in the 'correct value sent' communication link.

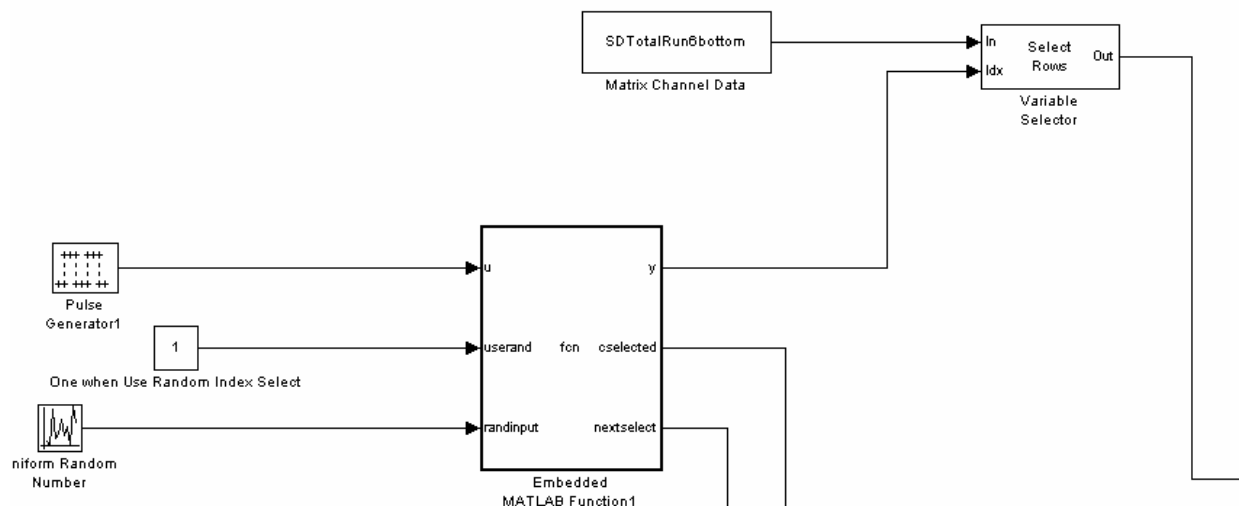


Figure 4: Selecting the Channel in Simulink

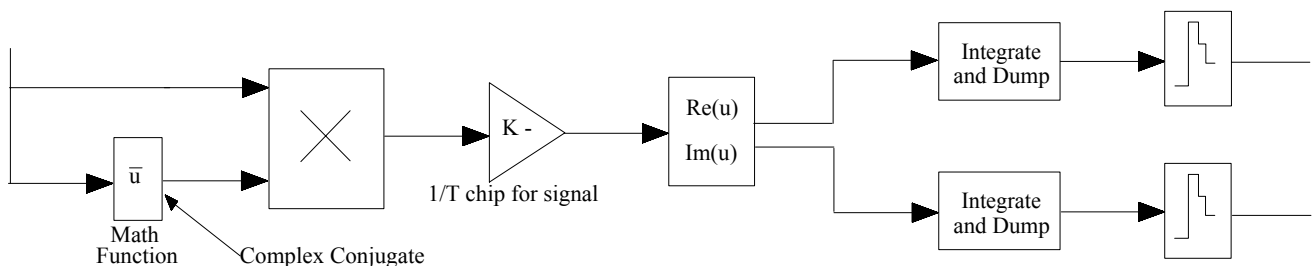


Figure 5: Main communication link in simulator

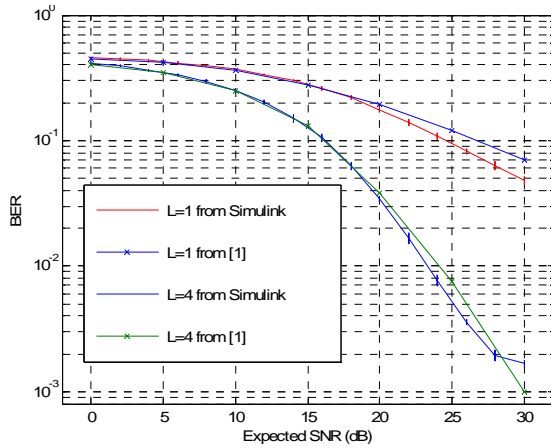


Figure 6: Comparison of measurements from Simulink model to estimates from Figures 7 and 8 from [1]

## V. Model Validation

Using the data found in [1] we obtained estimates of the simulation when orthogonal PPM with spacing between the '0' and '1' being 1 nanosecond was used for the case the number of paths in the template was '1' and the case where the number of paths was '4'. Figure 6 shows the data that the Simulink simulation / RTW produced compared to that from Figures 7 and 8 in [1]. It can be seen that our data is very close to that reported in [1] with BER's versus Expected

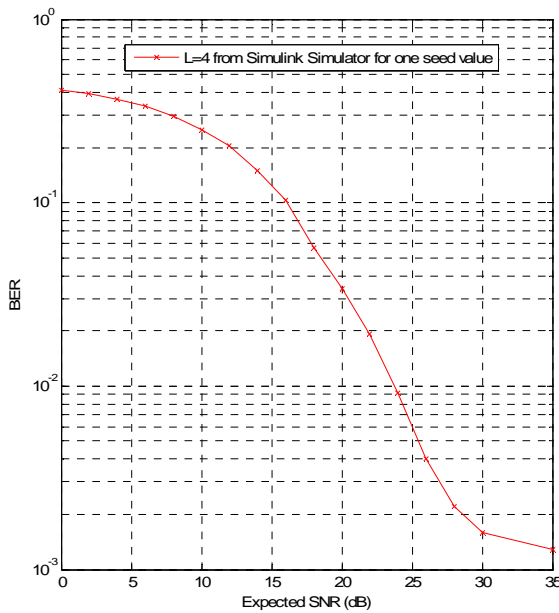


Figure 7: Plot of a single seed for L=4 showing an error floor appearing after 28dB in Simulink Simulation

SNR either within error bars or close to the estimated (by eye) data from [1], especially for Expected SNR less than 22dB. Above 22dB, for L=4, we found that our system had a slightly smaller BER than that reported in [1] until we were

above 28dB where we found an error floor to develop. All results for L=4 above 22dB were measured until 100 error events occurred to ensure that the data was statistically reliable (using the law of large numbers). Above 20dB for L=1 we found that our system experienced a smaller BER. The explanation for this is the conservative selection of the number of multipaths that were included, an example of which is shown in Figure 3. The error events used to generate the data in Figure 6 for L=1 had a minimum value of 200 errors (which was more than the minimum used of 100 error events chosen for L=4 measurements). Figure 8 and 9 show the results from the Simulink simulator for the case L=1 and L=4 respectively when the first set of arriving rays is used, a Partial Rake is implemented and a Selective Rake is implemented.

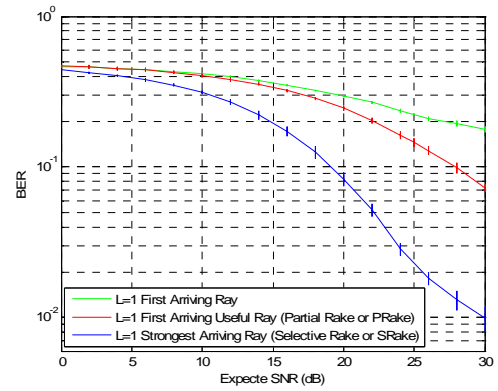


Figure 8: BER versus Expected SNR (dB) for first arriving ray, first useful (PRake) arriving ray and strongest arriving ray (SRake) for Simulink Simulation

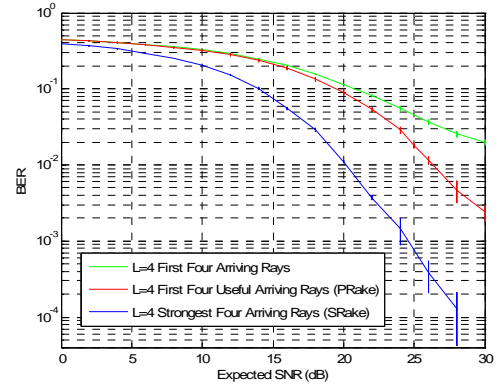


Figure 9: BER versus Expected SNR (dB) for the case first 4 arriving rays, first 4 useful (PRake) arriving rays and strongest 4 arriving rays (SRake) for Simulink Simulation

We plot in Figure 7 for a single seed value the observed data. It shows that the curve flattens out at high SNR. When we ran the simulator using the UWB channel model and no noise we found errors occurred. The Simulator with multipath was then run under no noise to locate where the first error occurred and a time trace was recorded. Figure 10

shows a snapshot of the results from probe signals inside the simulation. It shows the location of the pulse sent for a zero and then the location of a series of pulses close to each other which combine to 'appear' to be the correct symbol sent (using MRC). Table 1 shows the data that was used to form the channel at the location where the error occurred. The first column in Table 1 indicates the magnitude of the real part of the multipath at the time index appearing in column 2 of Table 1. The third column in Table 1 represents the time elapsed since the start of the current symbol's period (which is a total of 100ns). The fourth column of Table 1 is an internal simulation flag to indicate if a multipath is present or not (there is a maximum of 200 such paths, but in many instances there are less than 200). The fifth column of Table 1 is a flag to indicate if the multipath element satisfied the criteria used initially to indicate if the multipath was useful

in terms of magnitude and relative location to other significant multipath. From Table 1 the magnitude of the surrounding pulses to the main pulse were of opposite phase and comparable magnitude. More importantly the two multipath that were 0.4ns and 1ns away from the correct decision point (bottom two entries in Table 1) were nearly as large and had the same phase reinforcing each other and creating the error condition. Better choice of simulation data set would alleviate this problem, but in a real system this would correspond to ignoring a complete channel set or choosing pulses which had smaller magnitude decreasing the expected SNR of the system.

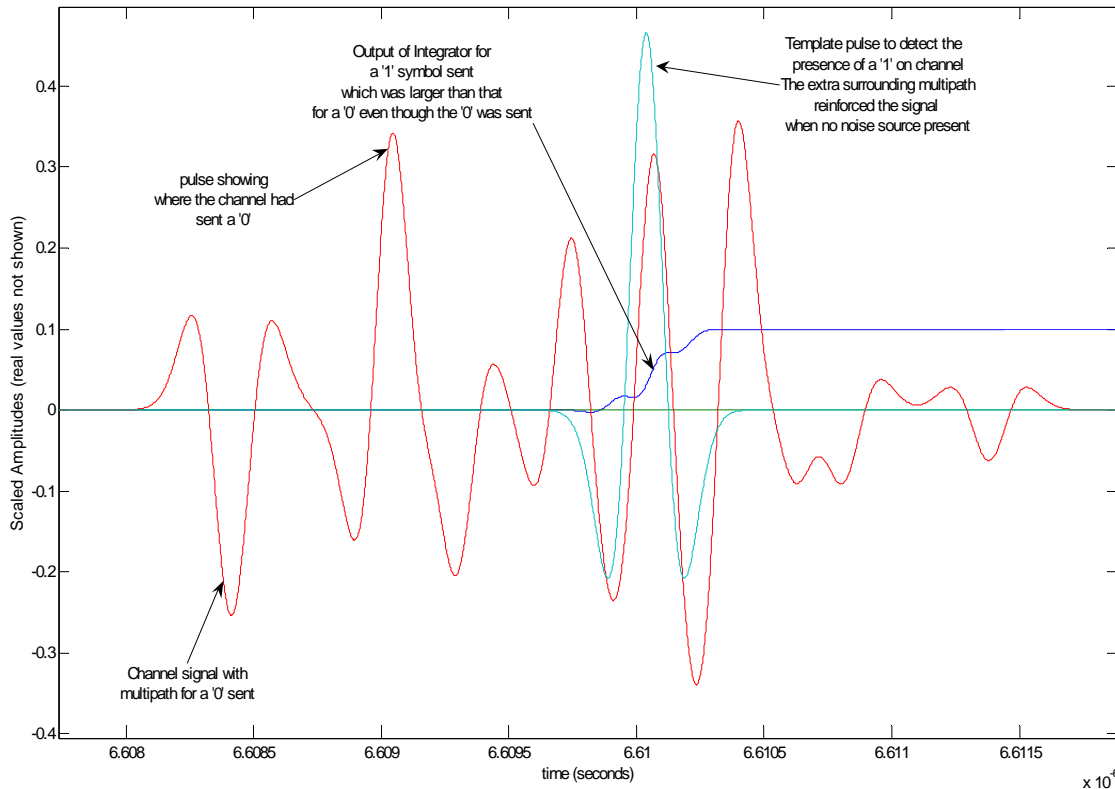


Figure 10: Snapshot of a small segment of channel signal at the point where error occurs without any noise.

Magnitude	Time Index	Time (secs)	Present	Useful Number
3.7774e-005	16889	1.6888e-008	1	0
-0.00035181	17647	1.7646e-008	1	1
8.5502e-005	17672	1.7671e-008	1	0
0.00022275	18112	1.8111e-008	1	0
4.9095e-005	18352	1.8351e-008	1	0

Table 1: Data used to form the channel around the location of the error.

## VI. CONCLUSIONS

We have developed a UWB simulation using Simulink which shows comparable results to those reported in [1]. We did this for a Saleh-Valenzuela channel using the same parameters as [1]. We also found that for our randomly generated data set with the channel changing every symbol sent an error floor became evident at about 28dB expected SNR. This was found to be a result of the multipath around the incorrect decision point.

We have developed a model which can be modified such that the rate of fading can be changed relatively easily. The shape of the received pulse can also be changed with some effort. We plan to add time hopping followed by direct sequence spreading to our simulator. It is expected that if the channel state information is changed less often that the simulation time will decrease as there will not be a requirement for the embedded MATLAB to be called as often to modify the channel state information. The developed model is flexible in that we can also use its framework to model other channel models by generating the channel state information separately and importing these into our simulinks channel data array (see Figure 4). This allows the simulator to be used for other datasets (that's is for other UWB channel models proposed in the literature [4]).

## References

1. Liuqing Yang, Georgios B Giannakis, "Analog Space-Time Coding for Multi-Antenna Ultra-Wideband Transmissions", IEEE Transactions on Communications, March 2004, pp. 507-517
2. Adel A M Saleh, Reinaldo A Valenzuela, "A Statistical Model for Indoor Multipath Propagation", IEEE Journal on selected Areas in Communications, vol SAC-5, No 2 Feb. 1987, pp. 128-137
3. Ian Oppermann, Matti Hamalainen, Jari Linatti, "UWB: Theory and Applications", Wiley Press, 2004, pp. 23-33.
4. Maria-Gabriella Di Benedetto, Guerino Giancola, "Understanding Ultra Wide Band Radio Fundamentals", Pearson, 2004