**Simulation Process**

Our group decided to use Simulink as the CAN Bus tools were readily available in the Vehicle Network Toolbox. It was decided that as opposed to using a CAN DBC file or ARXML file we would create an input signal with a “***Signal From Workspace***” block. This was decided because we also wanted to implement a BPSK (Binary Phase Shift Keying) block as a method of modulation for our CAN Bus data messages. To create a signal that is a 1:1 replica of a BPSK signal we would need to modulate the signal we created with a sinusoid block. This process proved to be impossible, however, due to the datatype restriction on the input of the CAN Pack block, which was UINT 8. Due to this restriction we decided to skip this process and proceed to incorporate a binary signal that changes with respect to sim time. To add realism, we opted for cyclic repetition so the model would be able to run indefinitely with the code being transmitted in a loop. We named the different signals according to commonly checked functions such as A/C, steering, ABS, TPMS, Smart Key, Airbag and Speed. A manual switching system was then created that provided the ability to change test parameters during model runtime. It is set up in a ladder style meaning that only one block is allowed to propagate its data after being switched. The associated signal is then propagated into a “***Sample and Hold***” block that samples the data at a period of five seconds per bit, which then gets converted into a UINT8 signal from a double. After the data passes through the UINT8 block, it is passed to the “***CAN Pack***” block. This block associates a message name, identifier type (Standard 11-bit), CAN Identifier and length field for the input signal. Once processed, we pass the signal to a “***CAN Transmit***” block which sets a channel for the data to be sent at a rate of 0.025s with a corresponding bus speed of 500 kbps. The message is then received and unpacked to be passed to the “***BPSK Modulate***” block. The phase offset parameter was set to (-pi/5) to account for both the in-phase an quadrature component portions of our input signal. After the modulation has been performed it is now time to transmit the signal wirelessly. Once again, we wanted to replicate a real-world scenario as closely as possible, therefore the implementation of the “***AWGN***” block (Additive White Gaussian Noise) was used to add an element of ambient disturbance. The signal is then sent to the wireless receiver block and then processed by the “***BPSK Demodulator***” block which has a phase offset of (pi/5) to correct the phase shift. It should be of not that Simulink blocks handle several details that wouldn’t be handled in real life. We would have needed to implement a high-pass filter to account for the introduced noise as well as an ADC to account for the transition from analog to digital data transmission to the receiving hardware.

