

Chapter 5

Verification and validation of the communication sub-system

In this chapter the focus will be on verification and validation of the developed in-house communication system. This chapter will commence by providing accurate definitions for each of these terms, where after each of these terms will be discussed. Then a detailed test and evaluation plan will be developed to address both verification and validation. Lastly the test and evaluation plans will be performed.

5.1 Introduction

Countless definitions exist for verification and validation. It was decided to use the following definitions which originate from a systems engineering perspective.

*“Verification confirms, by providing **objective** evidence, that **specified requirements** have been fulfilled. It determines whether products of a **given phase** satisfy the conditions imposed at the previous **phase**.” [63]*

*“Validation confirms, by providing **objective** evidence, that the **requirements** for a **specific application** have been fulfilled.” [63]*

Therefore it can be understood that verification and validation is the method used of establishing whether [63];

1. The requirements of the communication system are complete and correct.
2. The component finished at the current phase adheres to the *specifications* of the previous phase.
3. The communication system adheres to the end user’s operational, functional, performance and interface *requirements*.

Table 5-1: Key differences between validation and verification testing

Key differences		
Test and evaluation plan for verification	Test and evaluation plan for validation	Explanation
Controlled by the developer	Controlled by the developer	Both of the test plans will be conducted by the communication system developer.
Lower level of assembly	Higher level and lower level of assembly	The necessity for verification at the lower level assembly (component level) is to ensure that the specifications are maintained firmer.
Controlled environment	Realistic environments	Verification will be conducted in a controlled environment, where validation will be conducted in the ADES intended environment. For example, a very noisy environment.
Operations exceeding design limits	Operations as specified by user requirements	Verification will be conducted at exceeding design limits to aid in disclosing marginal design features.
Test to specifications	Test to requirements	Verification is done to determine whether specifications are met. Validation is done to determine that requirements are met.
Measure to determine performance specifications.	Measure of effectiveness.	<p>Validation will measure effectiveness. Examples of questions that determine effectiveness are: Does the communication system perform as intended? Is the communication system satisfactory for use in an AMB field? Does the communication system contribute to an effective AMB system?</p> <p>On the other hand verification will measure whether the performance specifications will be achieved, by making use of simulations and analysis.</p>

5.2 Test and evaluation plans

Testing means to, measure, analyse, evaluate and interpret results. Although various test and evaluation wisdoms exist, perhaps the best suited is: *“One test is worth ten thousand expert opinions”*. By conducting tests the subjective understanding is reduced by supplying quantitative data that can determine the performance and operational capability of the system. This is one step in the right direction when verifying and validating, keeping in mind that both the definitions state that objective evidence must be provided [63,64].

In this section two different test and evaluation plans will be developed to aid in verifying and validating the developed communication system. In Table 5-1 the main differences between how the verification testing and validation testing will be conducted are emphasized and explained.

5.3 Test and evaluation plan for verification

As mentioned previously, verification will be conducted at the lower levels (component level) of assembly in a *controlled* environment. In Chapter 4 the ADES digiComm protocol was designed and the various components were completed. At this stage it becomes necessary to determine whether the developed components satisfy the specifications of the previous phases.

The test and evaluation plan will begin by verifying each of the components in the developed communication system. These components will be divided into the two specified protocol layers. In Table 5-2 the different components that need to be verified in the physical layer are listed and in Table 5-8 the different components that need to be verified in the data link layer are listed.

5.4 Physical layer verification

The specifications that need to be verified are listed in Table 5-2.

Table 5-2: Physical layer verification

Specification to verify	Verified
1. Maximum bit rate	√
2. Maximum distance	√
3. Termination method	√
4. Biasing circuit	√

5.5 Electrical standard - and cable selection

The electrical standard selection and the cable selection will be verified by conducting two experimental tests. The two tests will be conducted at the maximum distance specification of 15 m, but at two different bit rates as tabulated in Table 5-3.

Table 5-3: Verification test

Test	Distance specification	Baud rate specification
1	15 m	2.24 Mbps
2	15 m	8.31 Mbps

Expandability is one of the key requirements of the ADES. The maximum bit rate required when there is no room for any expandability is 2.24 Mbps. In Chapter 3 the power amplifiers bit rate is calculated as 11.2 Mbps. However, this bit rate is decreased by a factor of 5, because point to point connections will be implemented. Therefore both the ISensorboard and the power amplifiers must reach a bit rate of 2.24 Mbps. The maximum reachable bit rate is selected as 8.31 Mbps. This value is obtained by considering the following;

- The maximum internal clock of the PowerPC which generates the synchronization signal to enable all the communication controllers is set to 133 MHz.
- The oversampling ratio is set to 16.

When these values are divided a bit rate of 8.31 Mbps is obtained. This value can be pushed to further limits by lowering the oversampling ratio to 8, however this will most certainly compromise the reliability of the system. Raising the bit rate will also not be necessary keeping in mind that timing is already four times higher than necessary.

To interpret whether the specifications can be achieved when using the RS 485 standard and the BLDN 9841 cable the following experiment will be conducted. Waveforms will be constructed for both of these tests to determine the two key characteristics of any serial data transmission system. These two characteristics are the rise time and the bit rate. The rise time must be no more *than 30% of the unit interval* (reciprocal of the **bit rate**) to aim for error free data transmission. Eye diagrams will also be constructed to study the openness of the eye which will indicate whether the physical layer will be sufficient for the ADES.

The experimental setup is shown in Figure 5-1 and Figure 5-2 . Data will be transmitted from the main controller to the slave and the physical layer will be analysed.

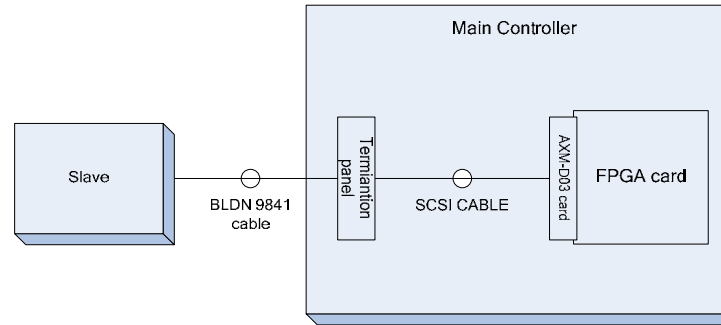


Figure 5-1: Experiment setup

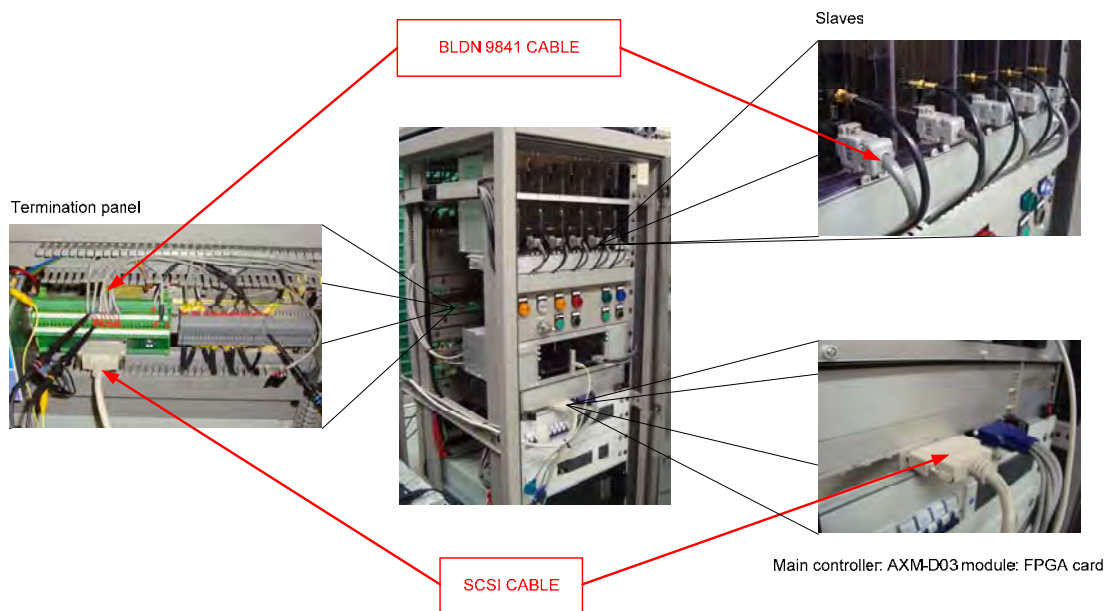


Figure 5-2: Experimental setup in the laboratory

5.5.1 Waveform

The waveforms will be obtained by transmitting one data frame from the transmitter to the receiver at the required bit rate. Furthermore the BLDN-9841 transmission line will be terminated in parallel on both sides of the transmission line. A termination resistor of 120Ω will be used. The waveforms will be constructed by using a real time oscilloscope. One waveform will be constructed at the transmitter end and the other at the receiver end. The rise times t_r will be determined by measuring the time from 20% of the logic zero amplitude to 80% of the logic one amplitude as shown in Figure 5-3.

5.5.1.1 Test 1:

The total voltage swing of the differential signal is 4 V, from -2 V to 2 V. To measure from 20% of the logic zero amplitude to 80% of the logic one amplitude the measurements will be taken from -1.2 V to 1.2V. In Figure 5-3 the receiver waveform is shown and in Figure 5-4 the transmitter waveform is shown. The rise time (t_r) obtained from measuring between these two values are listed in Table 5-4 as well as the percentage of the unit interval ($\%t_{UI}$) the rise time of the signal takes.

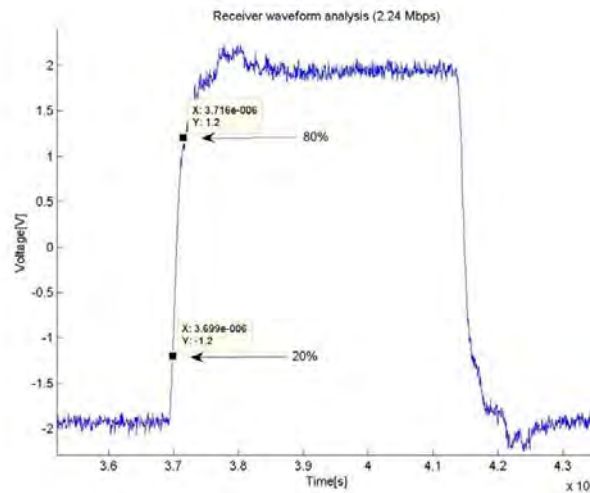


Figure 5-3: Receiver waveform analysis (2.24 Mbps)

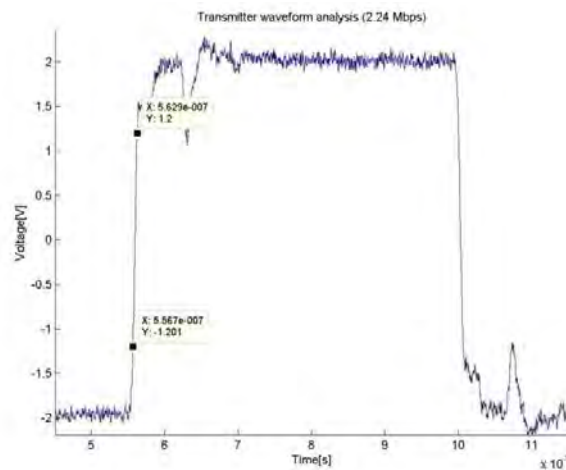


Figure 5-4: Transmitter waveform analysis (2.24 Mbps)

Table 5-4: Rise time analysis

Measure point	Rise time (t_r)	Unit interval (UI)	$t_r = \%t_{UI}$
Receiver	17 ns	446 ns	3.8%
Transmitter	6.2 ns	446 ns	1.4%

Both of the rise times are no more *than 30% of the unit interval* (reciprocal of the **bit rate**) therefore the physical layer will be able to attain the specified bit rate [65].

5.5.1.2 Test 2:

Test 2 is conducted exactly the same as test 1 to obtain the two different rise times. The measured rise times (t_r) are listed in Table 5-5 as well as the percentage of the unit interval ($\%t_{UI}$) the rise time of the signal takes. These rise times were obtained from Figure 5-5 and Figure 5-6.

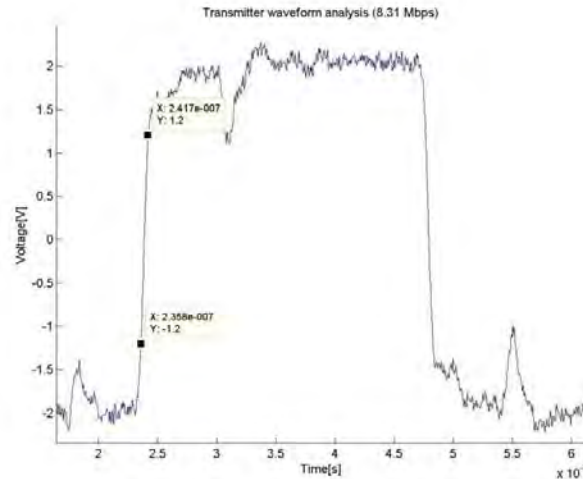


Figure 5-5: Transmitter waveform analysis (8.31 Mbps)

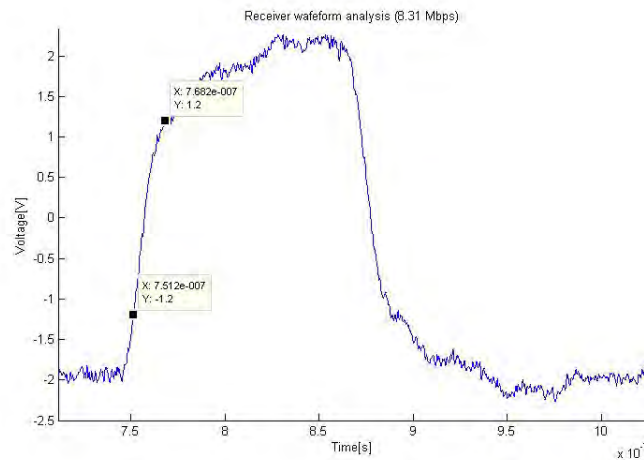


Figure 5-6: Receiver waveform analysis (8.31 Mbps)

Table 5-5: Rise time analysis

Measure point	Rise time (t_r)	Unit interval (UI)	$t_r = \%t_{UI}$
Receiver	17ns	120 ns	14.2%
Transmitter	5.9ns	120 ns	4.9%

Once again both of the rise times are no more *than 30% of the unit interval* (reciprocal of the **bit rate**) therefore the physical layer will be able to attain the specified bit rate which incorporates a level of expandability.

5.5.2 Eye diagrams

For the next experiment eye diagrams will be constructed allowing more complex verification of the selected physical layer. An eye diagram is a plot of voltage versus time (reciprocal of the bit rate). This wave form is constructed by overlaying waveform segments. By overlaying waveforms the cumulative performance of a data signal can be analysed.

For this specific section eye diagrams will be rendered by using a digital oscilloscope which incorporates advanced triggering and sampling technology. The necessity for advanced triggering is to enable repetitive triggering together with persistency plotting. The digital oscilloscope that will be used for constructing these eye diagrams is the LeCroy WaveRunner 104 MXi-A. An eye diagram will be constructed at the receiver end and at the transmitter end for both of the bit rates specified in Section 5.5.

When it comes to verification of an electrical standard the viewpoint that really counts is whether the receiver can take the data signal received and turn that data signal into clean ones and zeros. In order to do this, two key requirements must be studied. The first requirement is that there must be a *good amplitude difference* between the logic ones and logic zeros. This is of utmost importance bearing in mind that noise can influence the amplitude difference to such an extent that the receiver will not be able to differentiate between logic ones and zeros [42] .

The second requirement is that the receiver must receive a low jitter signal. A low jitter signal is considered to be between 5%-10% jitter of the unit interval (UI)[66]. By studying these two requirements, it is possible to know that the receiver will always be able to differentiate between ones and zeros when sampling at the mid-point of the eye.

Eye diagrams will be constructed at the transmitter and at the receiver end. Furthermore the BLDN-9841 transmission line will be terminated in parallel on both sides for *this experiment*. A termination resistor of 120 Ω will be used.

5.5.2.1 Visual inspection:

The first constructed eye diagram is shown in Figure 5-7. This eye diagram was constructed at the receiver end while data was received at a bit rate of 2.24 Mbps over a cable length of 15 m. This eye diagram will be analyzed by using visual inspection. Visual inspection is frequently used in the R&D environment to quickly study the quality of a signal [42].

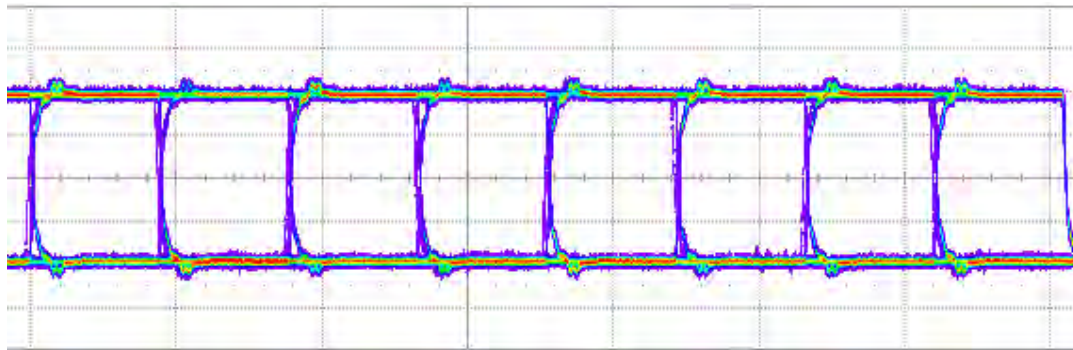


Figure 5-7: Eye diagram at the receiver end

After visually inspecting the eye diagram constructed in Figure 5-7 the following was observed:

- Good amplitude separation is present.
- Jitter present is less than 5% of the UI.

The second constructed eye diagram is shown in Figure 5-8. This eye diagram was constructed at the transmitter end while data was transmitted at a bit rate of 2.24 Mbps over a cable length of 15 m. This eye diagram will also be analyzed by using visual inspection.

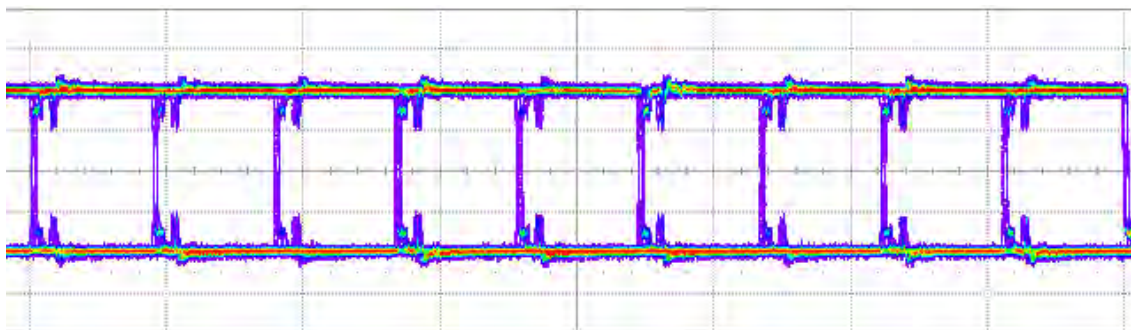


Figure 5-8: Eye diagram at the transmitter end

Visual inspection showed:

- Good amplitude separation.
- Jitter present is less than 5% of the UI.
- This eye diagram also shows a small ringing effect. This is caused by impedance mismatch giving rise to signal reflection.

5.5.2.2 *Statistical method*

The next two eye diagrams will be analysed by using a more advanced method than visual inspection. As mentioned briefly in the Chapter 2 this method involves acquiring data in an infinite persistence mode. This involves recording data points in a three dimensional database by using a digital oscilloscope. What this three dimensional database does in principle is it constructs various histograms at different cross sections. By using these histograms, maximum and minimum

signal values can be obtained. This method allows for more accurate results. For these calculations the distribution is assumed to be Gaussian [42]. These histograms are obtained by slicing horizontally and vertically through the eye diagram. Horizontal slicing provides statistical timing data and vertical slicing is done to determine statistical noise data. Slicing is shown in Figure 5-9.

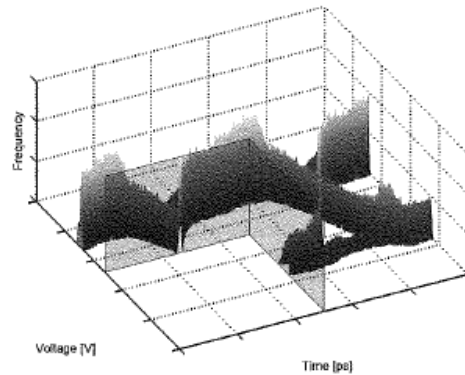


Figure 5-9: Cross section for jitter and noise measurement [65]

The key statistical parameters used in eye diagram measurements are shown in Figure 5-10. When vertical slices are taken through eye diagrams the *top* and *base* values are obtained. When horizontal slices are taken through the eye diagram *cross* values are obtained. Each of these values is characterized by a mean, μ and a standard deviation, σ component. “Most of the eye diagram measurements use 3σ distance from the mean as a statistical threshold that relates to the relatively high bit error rate.” [65]

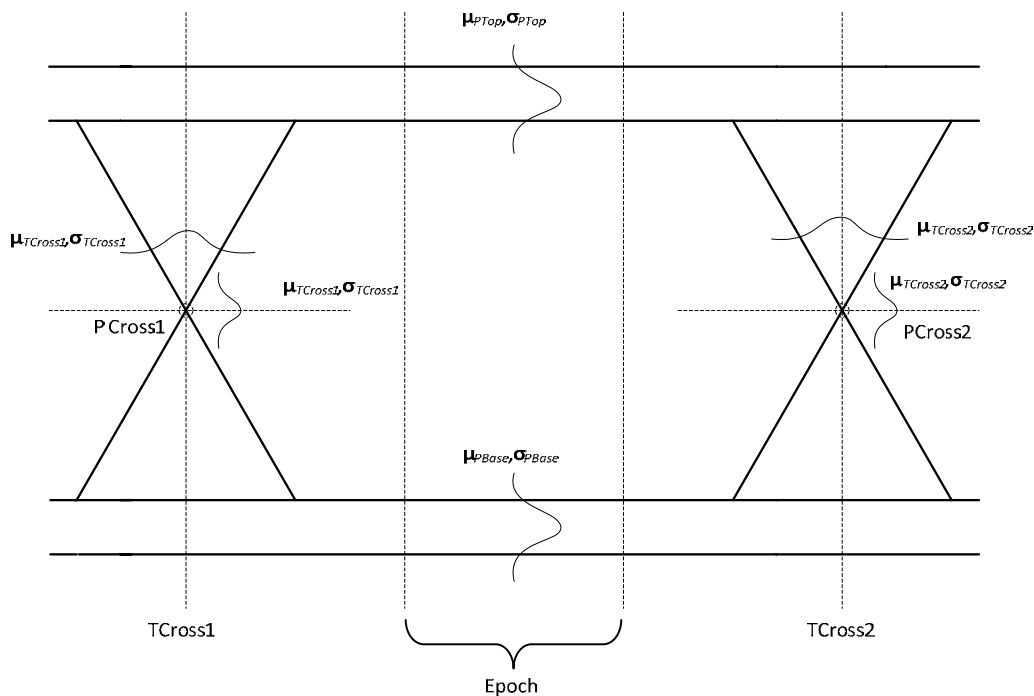


Figure 5-10: Statistical properties of eye diagram [42]

In this section eye diagrams will again be constructed at the transmitter and receiver end, however for this test the bit rate will be increased to 8.31 Mbps.

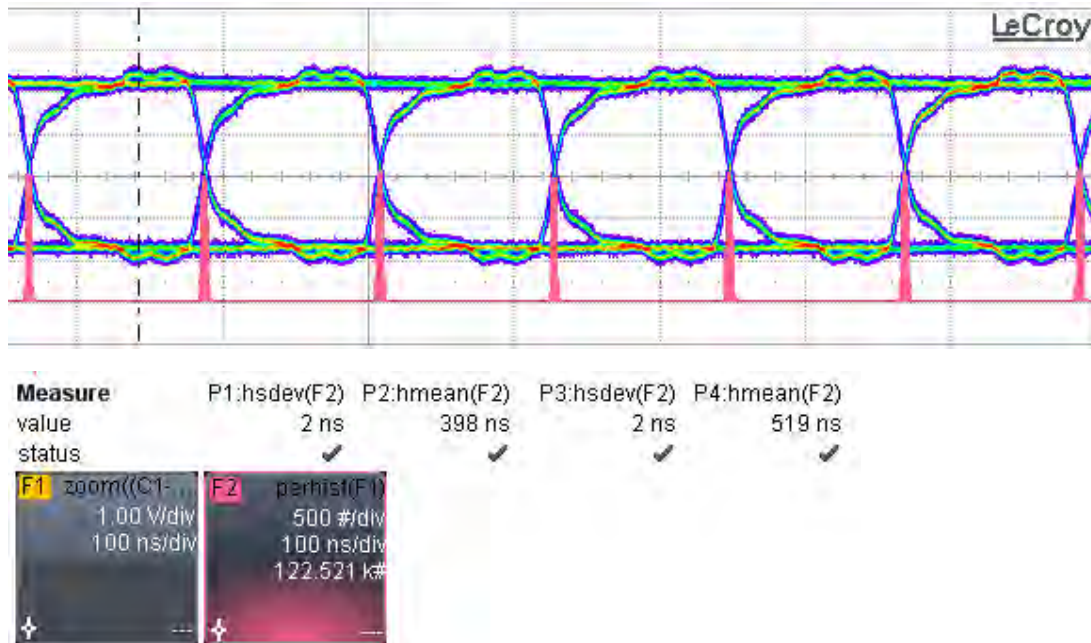


Figure 5-11: Eye diagram at the receiver end

Figure 5-11 shows ample information regarding the physical layer. Apart from just showing the eye diagram various histograms (as shown in pink) are also shown at the crossing sections. These histograms were obtained by horizontal slicing.

By using statistical analysis, the jitter percentage can be obtained by using (5.1) and (5.2).

6 sigma jitter can be written as:

$$6 \text{ sigma jitter} = 6 \sigma_{T_{cross1}} \quad (5.1)$$

where $\sigma_{T_{cross1}}$ is the standard deviation at T_{cross1} as shown in Figure 5-10.

Jitter percentage is determined by dividing the jitter (peak-to-peak jitter measured in time) by the unit interval as shown in Figure 5-12. Assuming that the jitter has a Gaussian distribution, 6 sigma jitter can be used which accounts for 99.999% of all jitter [67].

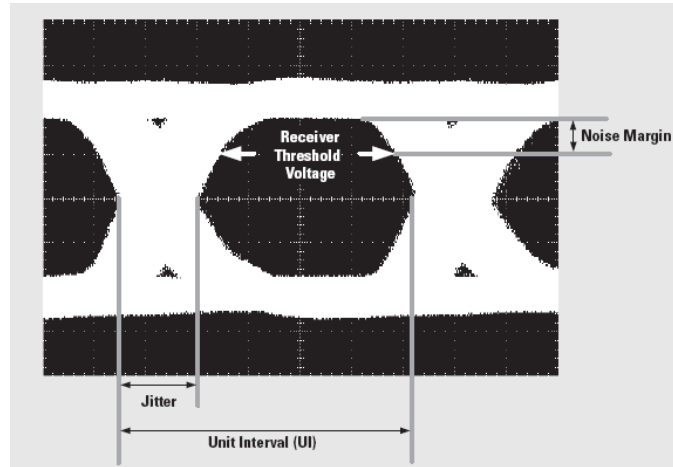


Figure 5-12: Percentage jitter measurement [68]

Jitter percentage can be written as:

$$\text{Percentage Jitter} = \frac{\text{Percentage Crossing Skew}}{\text{Unit Interval}} \times 100 \quad (5.2)$$

where the threshold crossing is given by peak-to-peak jitter and the unit interval is given by the reciprocal of the bit rate.

The necessary histogram values are shown just beneath the constructed eye diagram shown in Figure 5-11. These values were obtained by using the *Phistogram* function of the WaveRunner scope and are numbered as P1: hsdev for the standard deviations and P2: hmean for the means. In Table 5-6 the values are listed.

Table 5-6: Statistical eye diagram measurements on the receiver end

Parameter	Standard deviation, σ	Mean, μ
T_{cross1}	2 ns	398 ns
T_{cross2}	2 ns	519 ns

$$\begin{aligned} 6 \text{ sigma jitter} &= 6(2) \\ &= 12 \text{ ns} \end{aligned} \quad (5.3)$$

$$\begin{aligned} \text{Unit interval (UI)} &= \frac{1}{\text{bit rate}} \\ &= \frac{1}{8.31 \times 10^3} \\ &= 120 \text{ ns} \end{aligned} \quad (5.4)$$

$$\begin{aligned} \text{Percentage Jitter} &= \frac{12}{120} \times 100 \\ &= 10\% \end{aligned} \quad (5.5)$$

The jitter percentage is still considered good, keeping in mind that the receiver samples in an eye window which typically is the central 20% of the eye diagram [42].

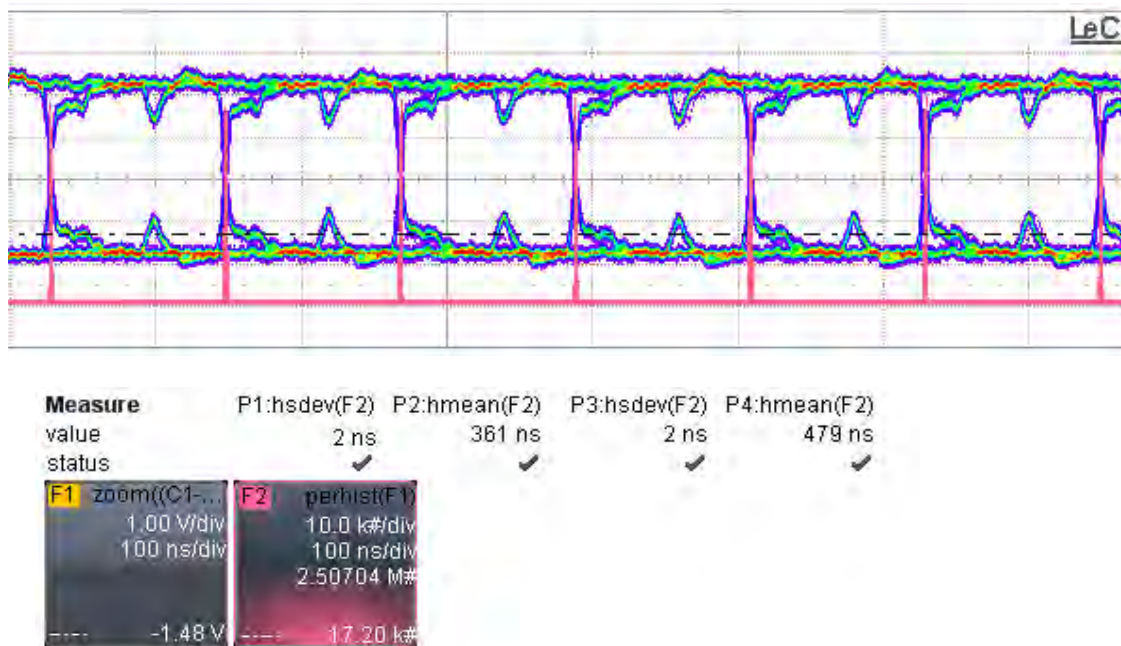


Figure 5-13: Eye diagram transmitter end

Figure 5-13 illustrates the eye diagram constructed at the transmitter end. One phenomenon which is once again observed by just visual inspection is the ringing effect. The ringing effect is caused by signal reflection due to impedance mismatch. Although this causes the amplitude separation to decrease it will not result in bit errors because the amplitude separation is still sufficient. However this effect will be discussed in the validation section. In Table 5-7 the statistical eye diagram measurements on the transmitter end is listed.

Table 5-7: Statistical eye diagram measurements on the transmitter end

Parameter	Standard deviation, σ	Mean, μ
Tcross1	2 ns	361 ns
Tcross2	2 ns	479 ns

$$\begin{aligned} 6 \text{ sigma jitter} &= 6(2) \\ &= 12 \text{ ns} \end{aligned} \quad (5.6)$$

$$\begin{aligned} \text{Unit interval (UI)} &= \frac{1}{\text{bit rate}} \\ &= \frac{1}{8.31 \times 10^3} \\ &= 120 \text{ ns} \end{aligned} \quad (5.7)$$

$$\begin{aligned} \text{Percentage Jitter} &= \frac{12}{120} \times 100 \\ &= 10 \% \end{aligned} \quad (5.8)$$

The noise floor will also be studied to compare the noise floor during verification with the noise floor during validation. This is assumed to be higher during validation due to the tremendous noise caused by the ADES operating environment. Noise peak to peak is determined by using (5.9). In Figure 5-14 the noise floor was determined by measuring the max (Pbase) and min (Pbase).

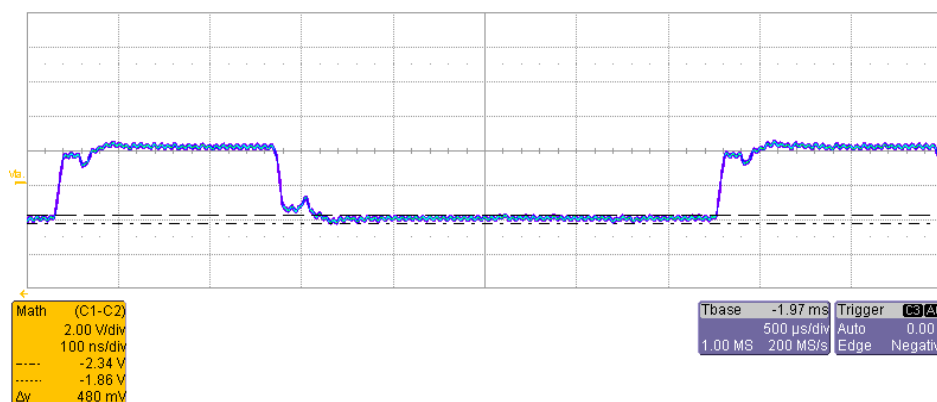


Figure 5-14: Noise peak to peak whilst the system was off

$$\begin{aligned} \text{Noise peak to peak} &= \begin{cases} \max(\text{Ptop}) - \min(\text{Ptop}), \text{ or} \\ \max(\text{Pbase}) - \min(\text{Pbase}) \end{cases} \\ &= 480 \text{ mV} \end{aligned} \quad (5.9)$$

The increase of the jitter percentage is due to the increase of the data transmission speed and the cable length [69]. Although the jitter percentage is higher it is still acceptable. Furthermore it must be remembered that these are worst case scenario conditions and the jitter percentage will certainly be better in the system where the optimum speed and cable length will be used.

5.5.3 Characteristic impedance and termination

Although it seems obvious to terminate the selected transmission lines, there exists a test to determine whether termination is an absolute necessity. This test involves determining whether a transmission line needs to be considered as a distributed parameter module or a lumped module. This can be determined by studying the relationship between the signalling rate at the driver

output, t_t , and the propagation time, t_{pd} , of the selected cable. If the relationship of $2t_{pd} \geq \frac{t_t}{5}$ is true then the selected transmission line must be regarded as a distributed parameter model and terminated accordingly [66].

As already mentioned in section 5.5.1.1, the transition times may not be more than 0.3 times of the specified UI [66]. The maximum specified transition time is:

$$\begin{aligned} t_t &\leq 0.3 \times UI \\ t_t &\leq 0.3 \times 120 \text{ ns} \\ t_t &\leq 36 \text{ ns} \end{aligned} \quad (5.10)$$

The propagation delay of the Belden 9841 cable is 1.6 ns/ft. The maximum transmission line length is 15 m, which is 49.21 ft. The propagation delay of the cable is therefore

$$\begin{aligned} t_{pd} &= 49 \times 1.6 \\ &= 79 \text{ ns} \end{aligned} \quad (5.11)$$

The final relationship can be established as follows:

$$2(79) \geq \frac{36}{5} \quad (5.12)$$

$$158 \geq 4.5 \quad (5.13)$$

The relationship is *true*, therefore the transmission lines will be treated as distributed parameter models and will be terminated accordingly. For interest sake the parallel termination was removed in the previous experiments and an eye diagram was constructed. This eye diagram is shown in Figure 5-15.

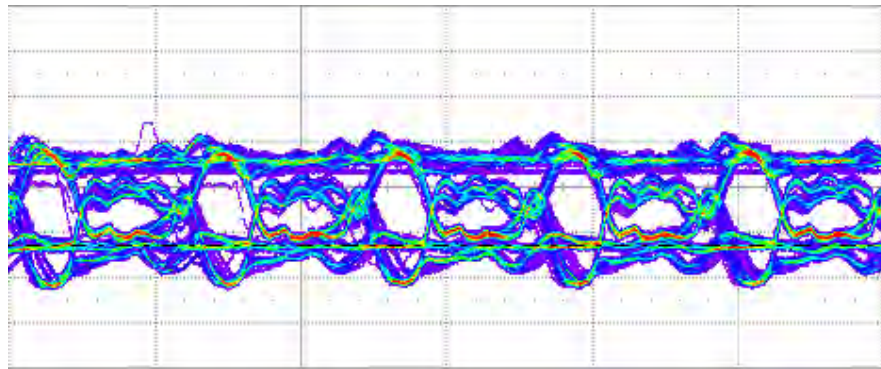


Figure 5-15: Eye diagram of un-terminated transmission lines

Visual inspection of the eye diagram shows severe undershoot, overshoot and ringing which totally corrupts the eye diagram. This is the reason to terminate the transmission lines.

The termination technique that will be used is parallel termination in conjunction with failsafe biasing. The termination resistor was selected as 120 Ω ; this selection was made because the data sheet of the Belden 9841 cable specified a nominal characteristic impedance of 120 Ω .

5.6 Data link layer verification

By predicting and verifying the physical layer performance, it does not ensure that the higher layers in the ADES digiComm protocol performs as specified. Therefore it is necessary to verify the data link layer which is the higher layer of the ADES digiComm protocol [64].

Verification of the data link layer will commence by providing certain stimuli and monitoring the data link layer responses. This will be done by using a simulation program called Modelsim[®]. As mentioned in Chapter 2, Modelsim[®] is a program that provides advanced code coverage capabilities which in return is a very valuable metric for *systematic verification*. One of the coverage types supported that will be used is Finite State Machine (FSM) coverage which indicates states and state transitions.

The various data link layer specifications that need verification are listed in Table 5-8.

Table 5-8: Data link layer verification

Data link layer specifications	Component to verify	Method of verification
Transmission parameters	UART top-level (consisting of UART TX, UART RX, FIFO TX and FIFO RX components)	Modelsim [®] simulations
State machines	Communication controllers and UART controllers	
Error modules	CRC controller and parity	
Error conditions	UART controllers	
Communication timing and Synchronisation	Communication controllers	

NB: Simulations will be done to indicate that each of these components is verified. The waveforms will be shown in figures and the most important aspects that need to be observed are numbered where after each of the numbers will be discussed in relevant tables.

5.6.1 Transmission parameters

The transmission pattern specifications for the ADES digiComm protocol are given in Chapter 4. These specifications are summarized in Table 5-9. In this section the transmission parameters, over-sampling ratio and the state machines of UART TX, UART RX and UART top-level component will be verified.

Table 5-9: Transmission parameters specifications

Transmission parameters specifications	Specification
Number of start bits	1
Number of data bits	16, LSB first
Number of parity bits	1
Stop bit	1
Start bit default	'0' (low)
Stop bit default	'1' (high)
Parity type	even
Over-sampling -ratio	16
Over-sampling -procedure	As specified in Section 4.9.1
Data transmission topology	Half-duplex

5.6.1.1 UART TX

The UART TX component and the transmission parameters are verified by creating a test bench program in Modelsim® SE 6.5 and instantiating the UART TX component. Communication will only be simulated in one direction. The test data that will be transmitted will be five randomly selected 16 bit HEX values; 91DB, CA55, BA19, 0000 and C788. The system clock that will be used is 133 MHz.

The UART TX component's waveform is shown in Figure 5-16 and the observations are shown in Table 5-10. The transmission parameters verification waveform is shown in Figure 5-17 and the observations are shown in Table 5-11.

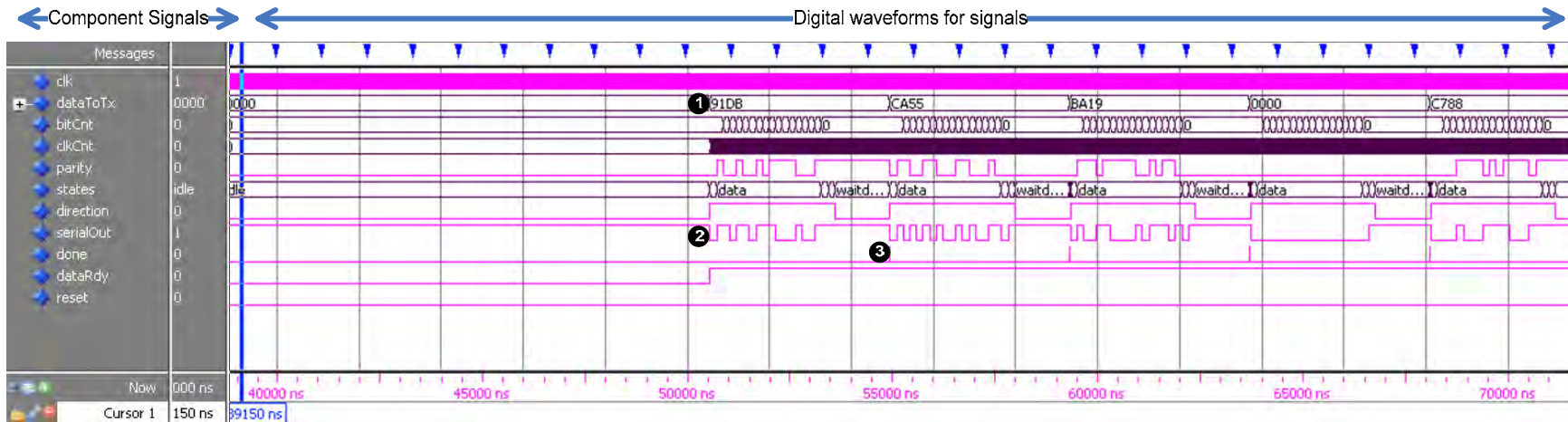


Figure 5-16: UART transmit simulation 1

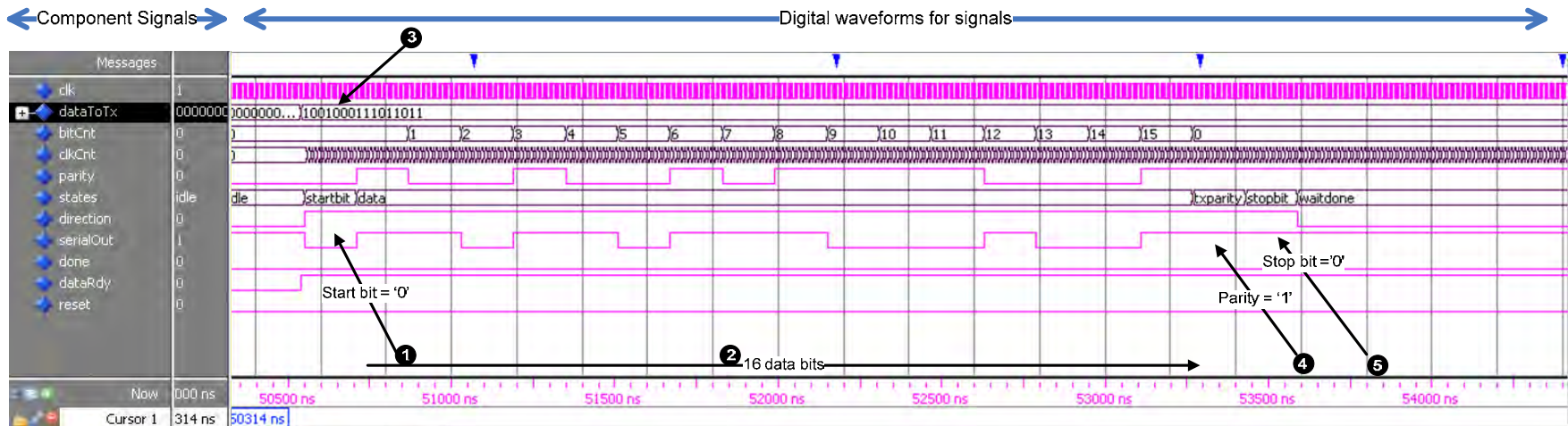


Figure 5-17: UART transmit simulation 2

Table 5-10: Observations concerning the UART TX component

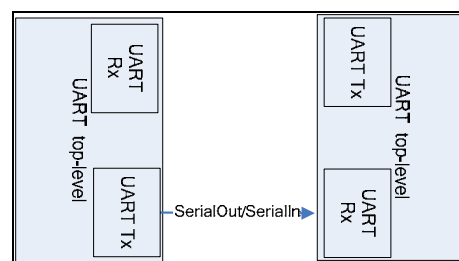
Simulation 1	Observations
1	The test bench correctly supplies five HEX values by continuously writing the values into the dataToTx signal.
2	Each of these values is then transmitted, bit by bit over the serialOut line.
3	After each of these values is transmitted the done strobe is flagged high, indicating the end of one transmission, after which a new value is written into the dataToTx signal as specified by the state machine in Appendix C.2.

Table 5-11: Observations concerning the transmission parameters

Simulation 2	Observations
1	One start bit was issued before the transmission of the data bits commenced. The start bit was low.
2	The 16 data bits followed after the start bit was issued.
3	The HEX value 91DB which corresponds to the binary value 1001000111011011 was transmitted indicating that the least significant bit was transmitted first on the serialOut signal.
4	A parity bit was transmitted. The parity bit equalled '1' indicating that even parity was implemented correctly, since an odd number of ones is present in the binary value 1001000111011011.
5	One stop bit was issued. The stop bit was a high. The state machine as specified executed correctly as shown in Appendix C.2.

5.6.1.2 UART RX and UART top-level

In this section the UART RX component, the over-sampling procedure and the UART top-level component will be verified as stipulated in Section 4.9.1. This will be done by creating a test bench program and instantiating both the UART RX - and UART TX component. These two modules will be connected as shown in Figure 5-18. Communication will only be simulated in one direction and the HEX values 91DB, CA55, BA19, 0000 and C788 will be transmitted.

**Figure 5-18: Test bench setup**

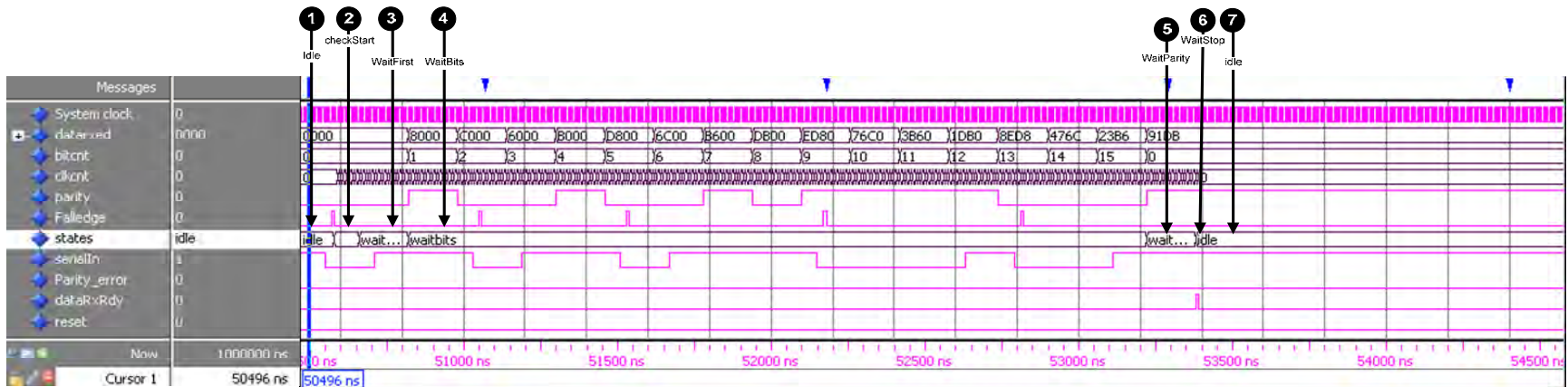


Figure 5-19: Simulation 1 UART RX

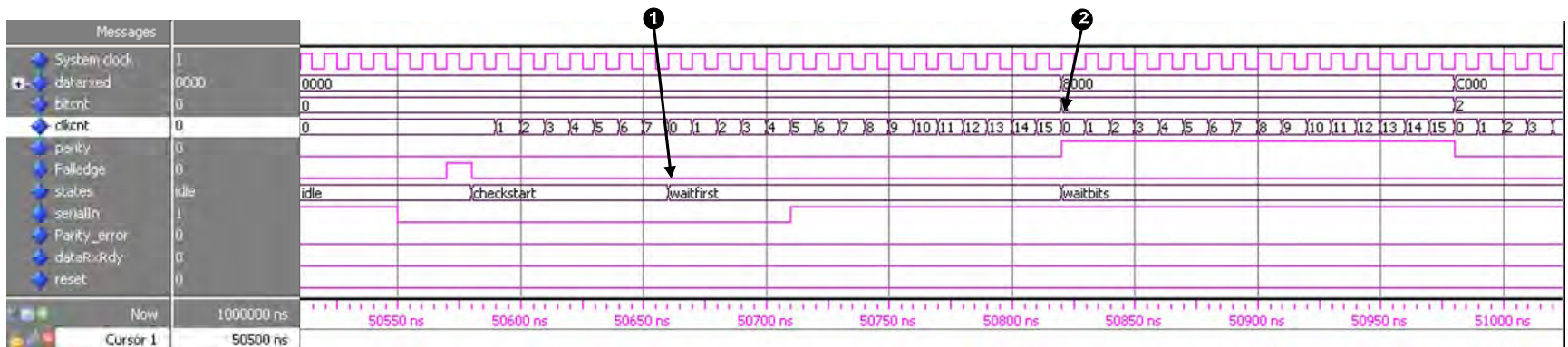


Figure 5-20: Simulation 2 UART RX

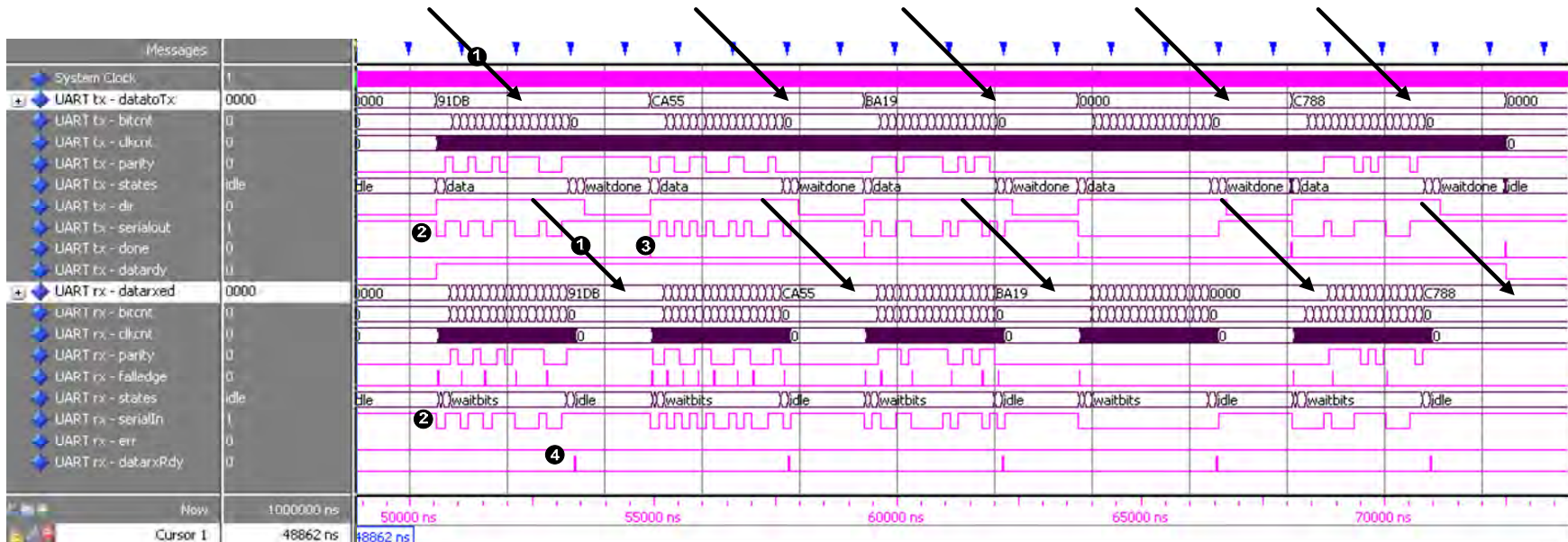


Figure 5-21: Simulation 3 UART top-level

Figure 5-19, Figure 5-20 and Figure 5-21 are the waveforms obtained when simulating the test bench. The observations for each of these figures are shown in Table 5-12, Table 5-13 and Table 5-14 respectively.

Table 5-12: Observations concerning the UART RX component

Simulation 1	Observations
1-7	Numbers 1-7 indicate that the state machine of the UART RX component executes as specified in Appendix C.1.

Table 5-13: Observations concerning the oversampling procedure

Simulation 2	Observations
1	The UART RX has to sample 16 times faster to reach the midpoint of each of the incoming data bits. In Figure 5-20 it can be noted that the system clock triggers a counter clkcnt to increment on the rising edge of the system clock. During the checkstart state the clkcnt increments to 7 to reach the midpoint of the start bit
2	The counter is cleared and a state transition occurs. In the waitFirst state the counter increments 16 times until it reaches the midpoint of the first data bit. This process repeated until the 16 data bits are received during the waitBits state.

Table 5-14: Observations concerning the UART top-level

Simulation 3	Observations
1	The correct data was received as transmitted by the UART TX component. This can be observed by monitoring the dataToTx signal and the dataRxd signal.
2	Data is transmitted out bit by bit on the serialOut signal and data is received bit by bit on the serialIn signal.
3	Once the 16 bit data is transmitted the done strobe flags high and new data is transmitted again.
4	Once all the bits have been received the dataRxRdy signal is flagged high and the receiver waits for new data to become available on the serialIn signal.

After studying the UART Tx – and the UART Rx component, the following transmission parameter specifications were verified as shown in Table 5-15 as well as the components listed in Table 5-16.

Table 5-15: Transmission parameters verified

Transmission parameters specifications	Specification	Verified	Not verified
Number of start bits	1	√	
Number of data bits	16, LSB first	√,√	
Number of parity bits	1	√	
Stop bits	1	√	
Start bit default	'0' (low)	√	
Stop bit default	'1' (high)	√	
Parity type	even	√	
Over-sampling ratio	16	√	
Over-sampling procedure	Specified in Section 4.9.1	√	
Data transmission topology	Half-duplex		X

Table 5-16: Components verified

Component	Verified
UART – TX	√
UART – RX	√
UART – top-level	√
FIFO RX	X
FIFO TX	X

5.6.2 Error modules

In this section the focus will only be on verifying the CRC controller, because the parity module has already been verified in Section 5.6.1. The following methods were used to verify the CRC controller module developed for the ADES digiComm protocol:

1. The code written in VHDL was simulated in Modelsim® by developing a test bench program.
2. The test bench program supplied a specified message to the CRC controller module and the output was monitored.
3. The same message and polynomial used in the Modelsim® test bench was supplied to the Communications Toolbox™ developed in MATLAB®

- The output obtained from the Modelsim® simulation was compared against the output generated by the MATLAB® simulation.

The message, CRC and initial values used are tabulated in Table 5-17. The MATLAB® and Modelsim® outputs are shown in Figure 5-22 and Figure 5-23 respectively.

Table 5-17: Variables for 32 bit message test

Variable	Hexadecimal	Binary	Decimal
Message	0x91DBBB12	1001000111 0110111011101100010010	2447096594
CRC	0xC599	1100010110011001	50585
Initial value	0x0000	0000000000000000	0

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> clear all;
h = crc.generator([1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1]);
msg = reshape(de2bi(2447096594, 32, 'left-msb'), 32, 1);
encoded = generate(h, msg);
showencodedmsg = reshape(encoded(1:32), 1, []);
showCRC = reshape(encoded(33:47), 1, [])

h =
    Type: CRC Generator
    Polynomial: [1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1]
    InitialState: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
    ReflectInput: false
    ReflectRemainder: false
    FinalXOR: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

showencodedmsg =
    Columns 1 through 26
    1 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1
    Columns 27 through 32
    0 1 0 0 1 0

showCRC =
    1 0 1 0 1 0 1 0 0 1 1 0 1 0 0
    
```

Figure 5-22: MATLAB® program calculating the CRC for the 32 bit input message

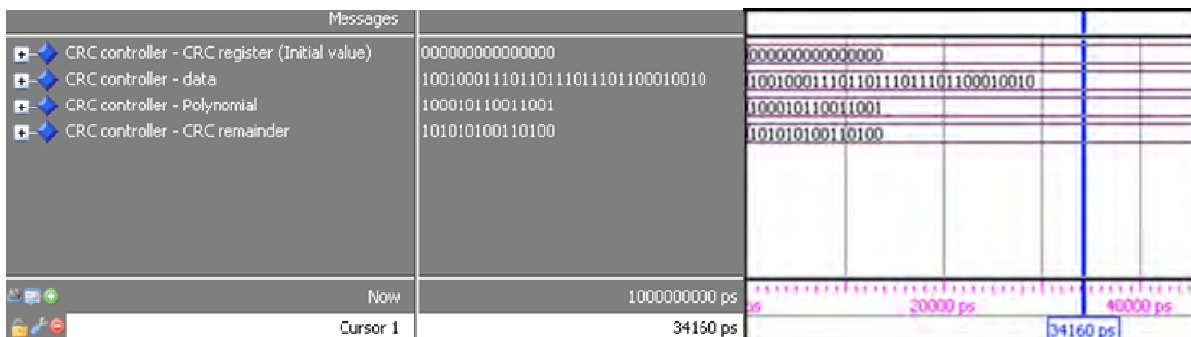


Figure 5-23: ModelSim® simulation of the CRC calculation for the 32 bit input message

A similar test was conducted with a *zero* initial value and a longer message, keeping in mind that data up to 64 bits will be transmitted. The message used is shown in Table 5-18. The MATLAB® and Modelsim® outputs are shown in Figure 5-24 and Figure 5-25 respectively.

Table 5-18: Variables for 64 bit message test

Variable	Hexadecimal	Decimal
Message	0x01DBBB12CA550000	133906303315935232
CRC	0xC599	50585
Initial value	0x0000	0

```

Command Window
New to MATLAB? Watch this video, see Terms, or read Getting Started.
>> clear all;
h = crc.generator([1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1]);
msg = reshape(dec2bi(133906303315935232, 64, 'intc-mb'), 64, 1);
encoded = generate(h, msg);
showencodedmsg = reshape(encoded(1:64),1,[]);
showCRC = reshape(encoded(65:79),1,[]);

h =
    Type: CRC Generator
    Polynomial: [1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1]
    InitialState: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
    ReflectInput: false
    ReflectRemainder: false
    FinalXOR: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

showencodedmsg =
    Columns 1 through 26
    0 0 0 0 0 0 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1
    Columns 27 through 52
    0 1 0 0 1 0 1 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0
    Columns 53 through 64
    0 0 0 0 0 0 0 0 0 0 0 0
    showCRC =
    1 0 0 0 0 0 1 1 1 0 1 0 0 1 1
    
```

Figure 5-24: MATLAB® program calculating the CRC for the 32 bit input message

Messages		
CRC controller - CRC register (initial value)	0000000000000000	0000000000000000
CRC controller - message	01DBBB12CA550000	01DBBB12CA550000
CRC controller - Polynomial	100010110011001	100010110011001
CRC controller - crc remainder	100000111010011	100000111010011
Now	1000000000 ps	999999200 ps
Cursor 1	0 ps	

Figure 5-25: Modelsim® simulation of the CRC calculation for the 32 bit input message

Table 5-19: Results for CRC tests

Test	MATLAB®	Modelsim®
1	0x5534	0x5534
2	0x41D3	0x41D3

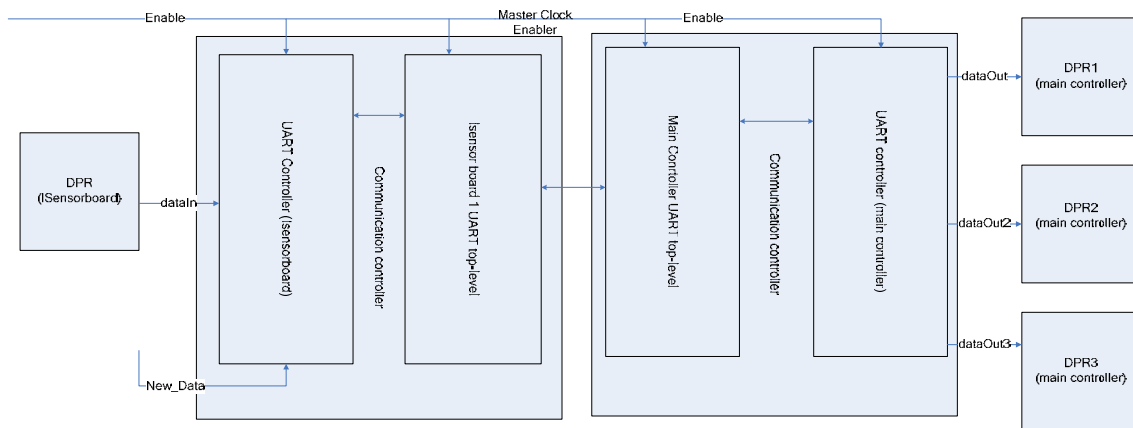
The results are listed and compared in Table 5-19. In both of the tests conducted the CRC remainders generated matched. Thus the communication controller functions correctly for the component specifications as listed in Table 5-20.

Table 5-20: CRC controller specifications

Parameter	Specification	Verified
Polynomial	CAN 15 CRC	√
Initial value	0x0000	√
Data length	Up to 64 bits	√

5.6.3 ISensorboard and main controller interconnection

In this section the ISensorboard - and the main controller communication controllers will be verified. This will be done by creating a test bench program and instantiating both communication controllers and connecting the two communication controllers as shown in Figure 5-26. The first simulation will monitor normal operating conditions and the second simulation will monitor error conditions. The X, Y and Z position values that will be read out of the DPR situated on the ISensorboard are; 91DB, CA55 and BB12 respectively.

**Figure 5-26: Test bench setup to verify communication controllers**

5.6.3.1 Simulation 1 of ISensorboard communication controller (normal condition)

Figure 5-27 is the waveform obtained when simulating the test bench shown in Figure 5-26 and studying the ISensorboard communication controller response. The observations concerning this simulation are shown in Table 5-21. Refer to the state machines in Section 4.13.

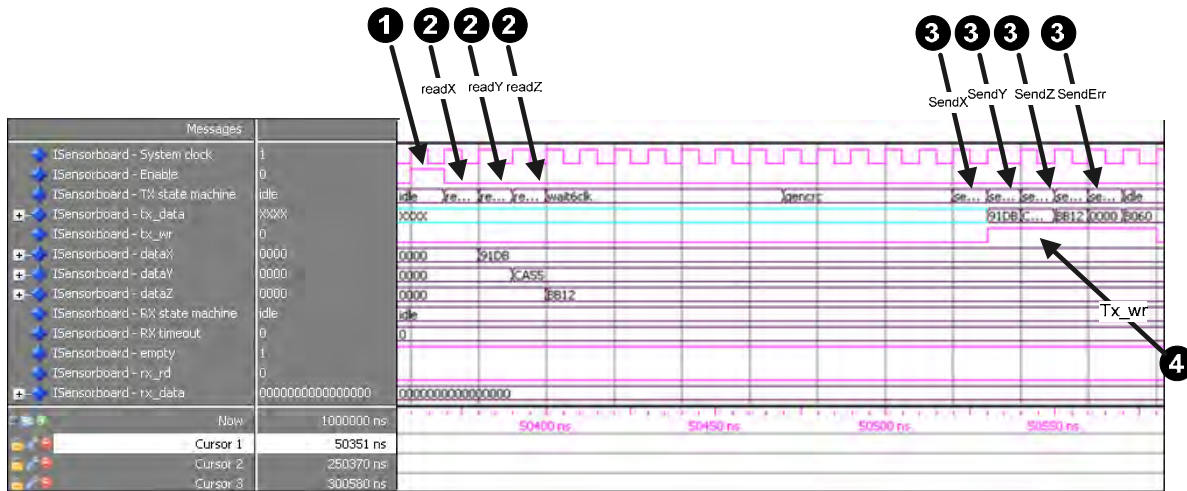


Figure 5-27: UART controller ISensorboard verification

Table 5-21: Observation concerning the ISensorboard communication controller

Simulation 1	Observations
1	The ISensorboard and the main controller is enabled by a sync signal.
2	After the ISensorboard communication controller is enabled the X, Y and Z positions are read out of the DPR ¹⁷ , where after the CRC are calculated.
3	These values are then transmitted back to the main controller as specified by the state machines. Figure 5-27 illustrates the state transition in the UARTcontroller.
4	The FIFO TX component can also be verified by studying this simulation. Once the ISensorboard is ready to transmit the tx_wr strobe is flagged high (transmit control signal), and the values in the tx_data signal is written into the FIFO. Once the FIFO is full, the UART TX is informed that data is available in the FIFO to transmit. Data is then transmitted to the main controller.

¹⁷ When a VHDL command is issued it is scheduled to execute on the next clock cycle. That explains why the state transitions to *readX* occurs, but the value is only read out of the DPR on the next clock cycle.

5.6.3.2 Simulation 1 of main controller communication controller (normal condition)

Figure 5-28 is the waveform obtained when simulating the test bench shown in Figure 5-26 and studying the main controller's communication controller. The observations concerning this simulation are shown in Table 5-22. Refer to the state machine in Section 4.14.

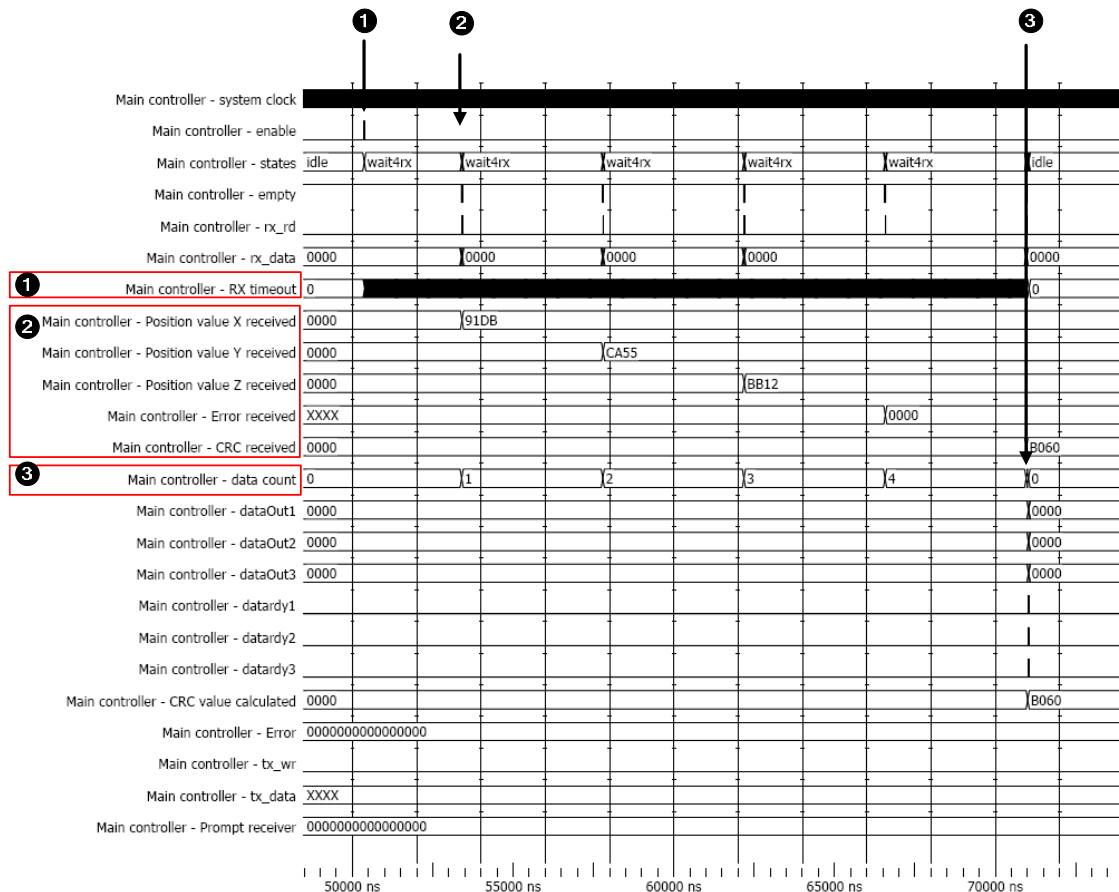


Figure 5-28: UARTcontroller main controller verification

Table 5-22: Observations concerning the main controller's communication controller

Simulation 2	Observations
1	Once the main controller's communication controller is enabled state transitions occur to the wait4rx state. At the same time the RX (receiver) timeout counter starts incrementing. This can be seen in Figure 5-28.
2	Once the UART controller is in the wait4rx state it waits for the empty signal to become '0'. Indicating that the FIFO RX component has new data available for the UART controller. State transition occurs to the getdata state and the rx_rd ¹⁸ strobe (read control signal connected to the FIFO) is flagged high and

¹⁸ The read control signal and the remove command are scheduled in that particular clock cycle, but execute on the next clock cycle.

the value present in the **rx_data** buffer is removed from the FIFO and written into a signal that holds the X position value received. State transition occurs to **wait1clk**¹⁹ state and then to the **wait4rx** state. The UART controller once again waits for the FIFO RX to indicate when new data is available to be processed by the UART controller. Figure 5-29 zooms into the region marked as 2 in Figure 5-28 to show the FIFO RX operating principle.

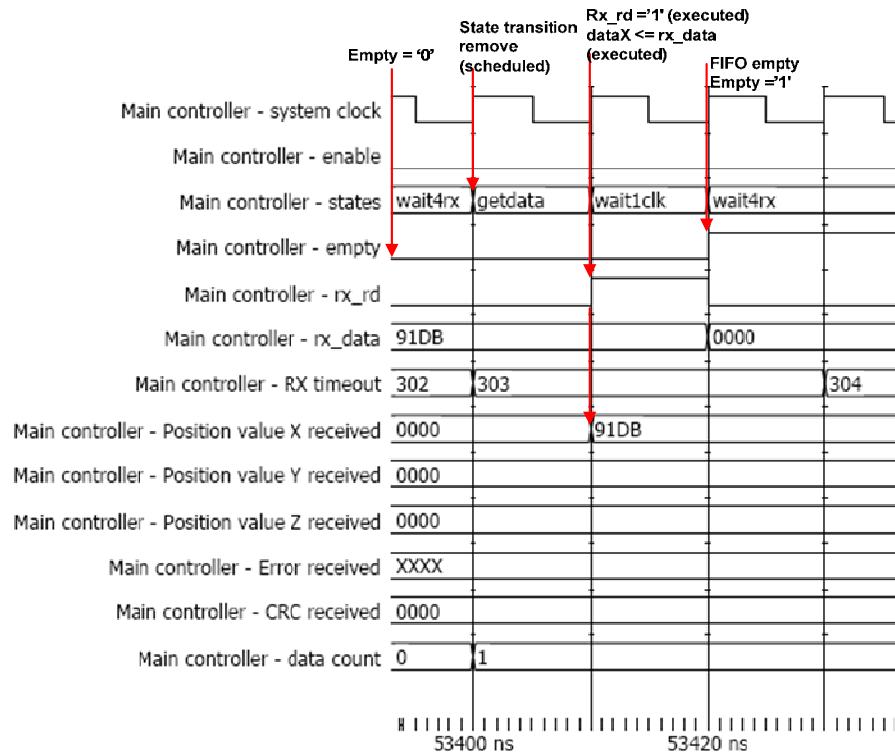
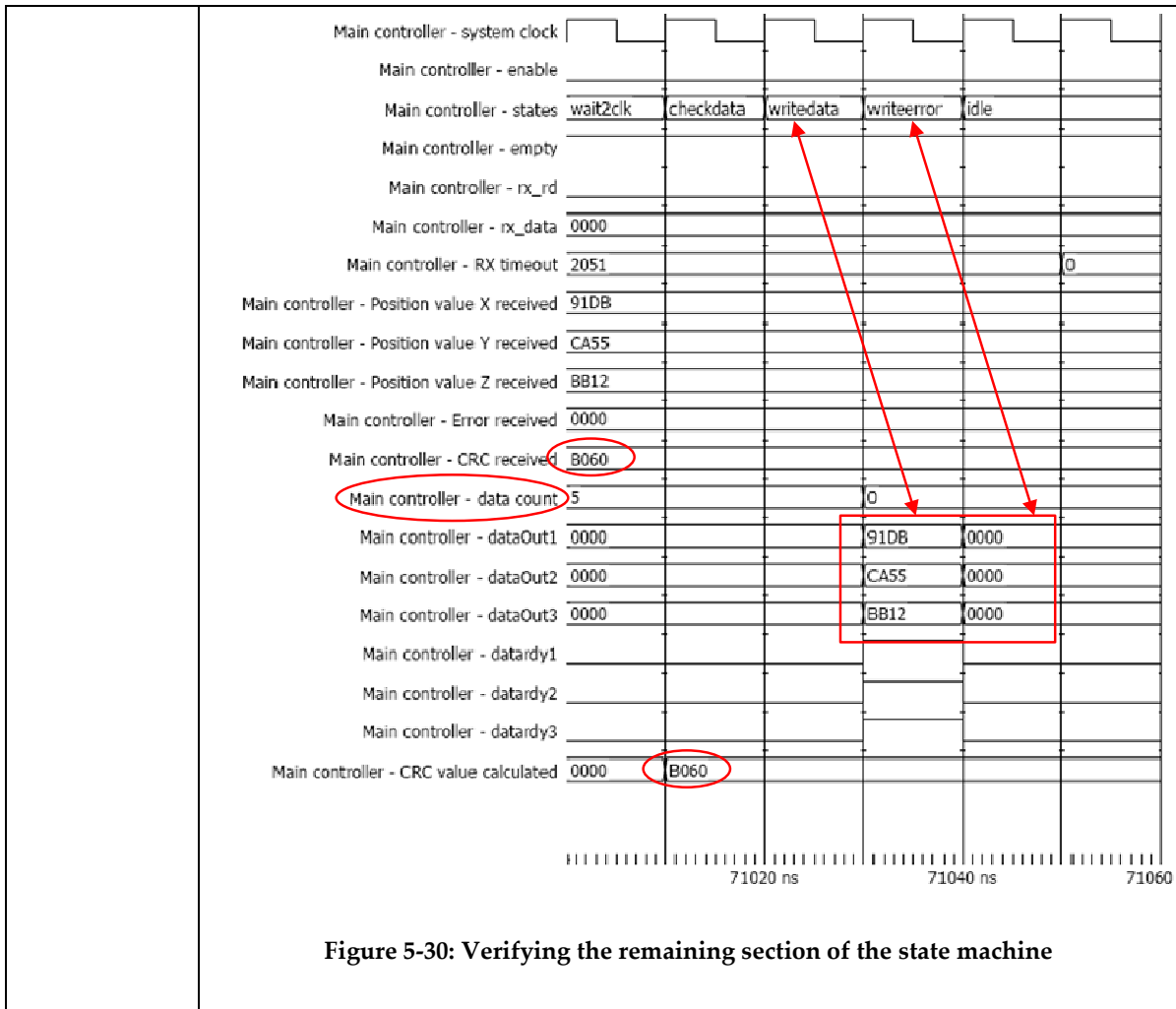


Figure 5-29: Verifying the FIFO RX component

- 3 After each value has been received the **data count** signal increments. Once the **data count** signal equals 5 all the data has been received and state transition occurs. In **checkData** the CRC is calculated. If the CRC received and the CRC calculated matches no error is flagged and the position values and the error values are written into DPRs. Figure 5-30 zooms into the region marked 3 in Figure 5-28 to show that the remaining part of the state machine executes correctly.

¹⁹ The `wait1clk` state is implemented to compensate for the off by one timing error as mentioned previously.



5.6.4 Error conditions

In these simulations errors will be induced and the response of the UART controllers will be monitored. This will be done to determine whether the UART controllers react correctly and flag the listed error conditions. Refer to Section 4.12.3 and Section 4.12.4 for an explanation of the error conditions. Figure 5-31 is the simulated waveform obtained when the synchronization signal is lost; the observations are listed in Table 5-23. Figure 5-32 is the waveform obtained when a CRC mismatch occurred. The related observations are listed in Table 5-24.

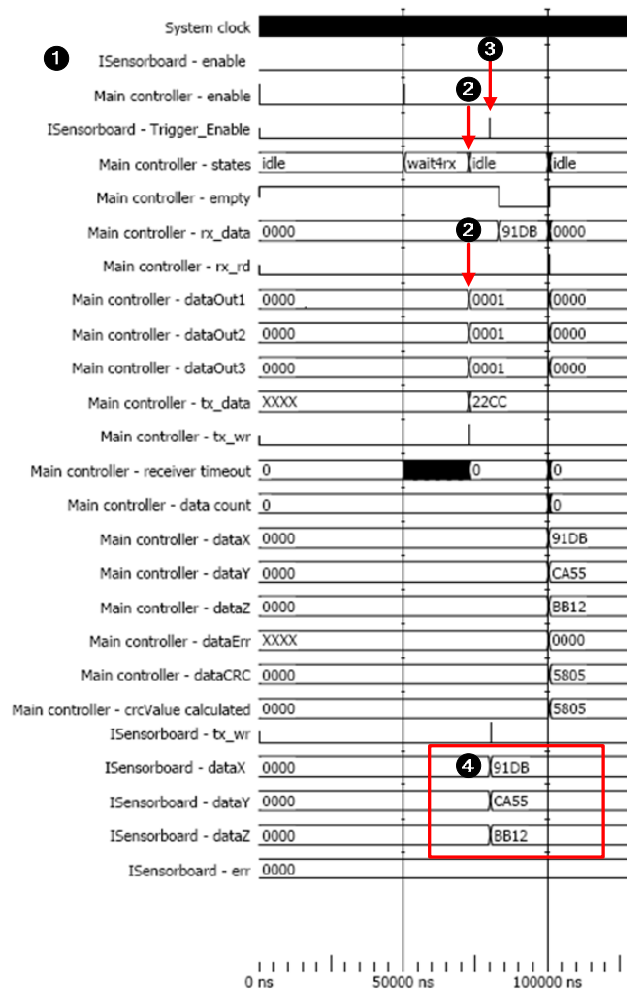


Figure 5-31: Error response to loss of synchronization signal

Table 5-23: Observations concerning the loss of the synchronization signal

Simulation 1	Observation
1	The sync signal connected to the ISensorboard UART controller was removed which causes a receiver timeout on the main controller UART controller.
2	This forced the main controller to flag an error (error condition 1) and prompt the ISensorboard.
3	After receiving the prompt command the trigger enable goes high and enables the receive process.
4	After the ISensorboard was enabled data is transmitted.

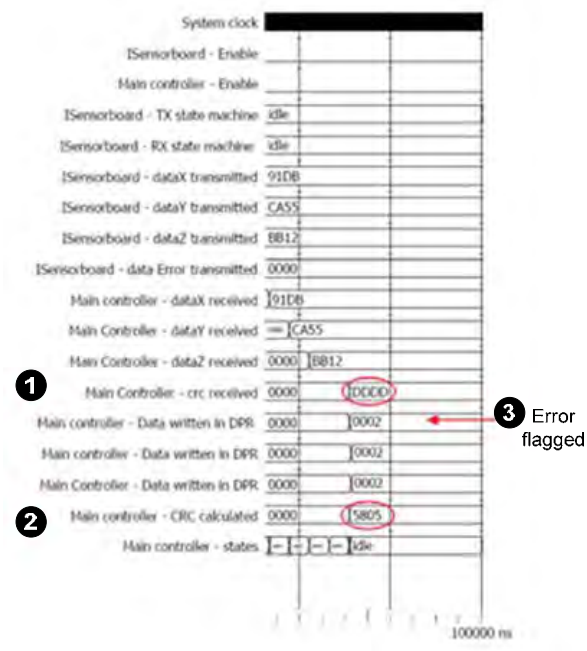


Figure 5-32: Error response to CRC mismatch

Table 5-24: Observations concerning a CRC mismatch

Simulation 2	Observations
1	A hard coded CRC value DDDD hex is transmitted from the I-Sensorboard.
2	This causes a CRC mismatch.
3	The X, Y and Z position values are written into the DPR, where after the error 02 is flagged indicating that a CRC mismatch has occurred.

The other error conditions were tested in the same manner; however it was not documented to prevent the dissertation from becoming too long. The units listed in Table 5-25 were verified successfully against the listed specifications.

Table 5-25: I-Sensorboard and main controller interconnection verification

Specification to verify	Verified
UART controller I-Sensorboard	√
UART controller Main controller	√
Error condition 1	√
Error condition 2	√

5.6.5 Communication timing & synchronization

One of the most important specifications of any communication system is timing and synchronisation. The synchronisation and communication timing were already explained in detail. In Table 5-26 the crucial specifications that need to be verified are listed.

Table 5-26: Communication timing & Synchronizations specifications

Component	Specification
Synchronization	Every 50 μ s or at a rate of 20 kHz all the functional units must be enabled by the sync signal.
Timing	The communication timeline between the ISensorboard and the main controller may not exceed 50 μ s. The same specification apply for the interconnection between the power amplifiers and the main controller

5.6.5.1 Enable

The ISensorboard and Main controller are enabled every 50 μ s as specified by the communication timing diagram. The cursor indicates the first enabled edge and the second enable edge. These two edges are 50000 ns apart, which correspond to a time interval of 50 μ s. Once the communication controllers are enabled state transitions occur and serial transmission start.

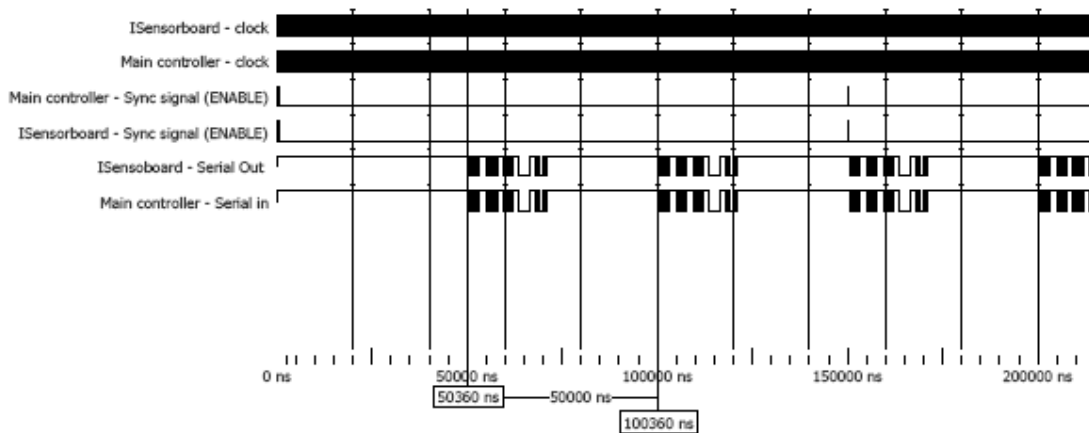


Figure 5-33: Synchronization verification

5.6.5.2 Communication timing (normal operation)

To determine whether the communication timing specification is achieved, the UART controller on the ISensorboard executes as well as the UART controller on the main controller. The time it takes for the receiver (main controller) state machine to execute once is measured, keeping in mind that

this represents one communication cycle. This is measured as 20660 ns (as shown in Figure 5-34) which corresponds to a frequency of 48.40 kHz. This deviates from the timing calculated in Chapter 3 due to the execution of the state machine which adds clock cycles.

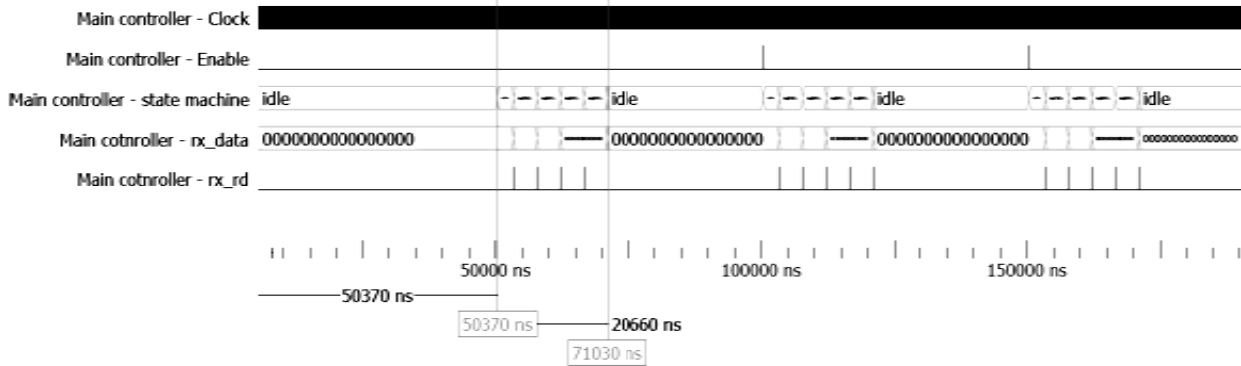


Figure 5-34: Communication timing during normal operation

5.6.5.3 Communication timing (error operation)

The frequency obtained in the previous simulation will decrease drastically when the main controller is forced to prompt the ISensorboard to transmit. This can be seen by studying Figure 5-35.

Once the main controller state machine transitions to the wait4rx state, a receiver timeout occurs, because no data is received from the ISensorboard due to the loss of the synchronization signal. An error is flagged and the prompt command is transmitted. The time it takes for the main controller to prompt, is measured as 15 000 ns. The rest of the state machine executes and the ISensorboard transmits back. This takes another 28050 ns. Thus the total time the communication cycle takes when prompting and receiving is 43050 ns which correspond to a frequency of 23.23 kHz.

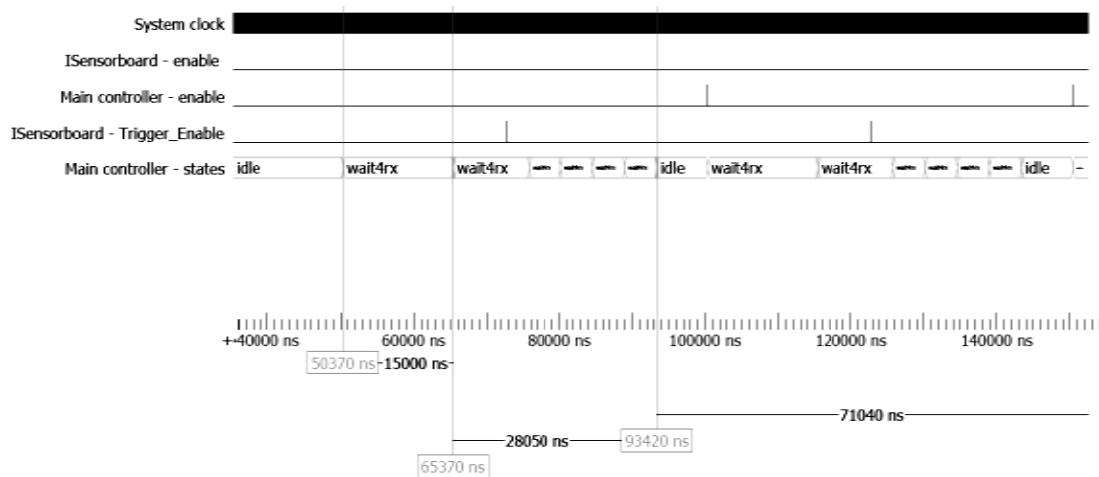


Figure 5-35: Communication timing during error conditions.

The synchronization and the communication timing has been simulated and recognized to be within the specifications.

5.7 Test and evaluation plan for validation

The test and evaluation plan for validation will be conducted in a *realistic* environment. This implies that the complete ADES (*higher and lower level of assembly*) will be tested in an AMB environment at *operational* conditions. Subsequently it will be possible to determine whether the *requirements* of the system were met. The test and evaluation plan will commence by validating whether the physical layer achieves the main requirements. Some of the requirements that will be validated are listed in

Table 5-27.

Table 5-27: Requirements to validate

Requirement	Method of validation
1. To what extent is the system considered to be noise immune?	Noise estimation
2. How reliable is the system?	Jitter calculation Value estimation Eye diagram construction
3. Is the system expandable?	Eye diagram construction

5.8 Physical layer validation

The performance of the developed physical layer of the ADES digiComm protocol can be validated by simply constructing eye diagrams and determining the statistical properties for eye measurement [65]. The *openness* of the eye is an indication of the *quality* of the signal as well as an indication of the noise and the distortion tolerance of the designed physical system. Therefore by constructing eye diagrams three of the main requirements of the ADES can be validated which are; reliability, robustness and expandability [70].

5.8.1 Eye diagram construction

The physical layer will be validated by constructing an eye diagram and measuring the statistical properties of the eye diagram as described in 5.6.2. The following will be calculated where after the results will be analysed. All the equations are obtained from “*Digital communication test and measurement for high speed physical layer characterization*” [65].

1. Eye height
2. Eye width
3. Jitter pp
4. Jitter RMS
5. Noise RMS
6. Noise pp
7. Signal to noise ratio

5.8.2 Eye diagram construction and measurements

The eye diagram was constructed at the main controller end. The highest bit rate specified was used (8.3 Mbps). The ADES was *fully operational* and the rotor was *suspending*, implicating that immense noise was present in the system. Some of the noise was caused by power amplifier switching.

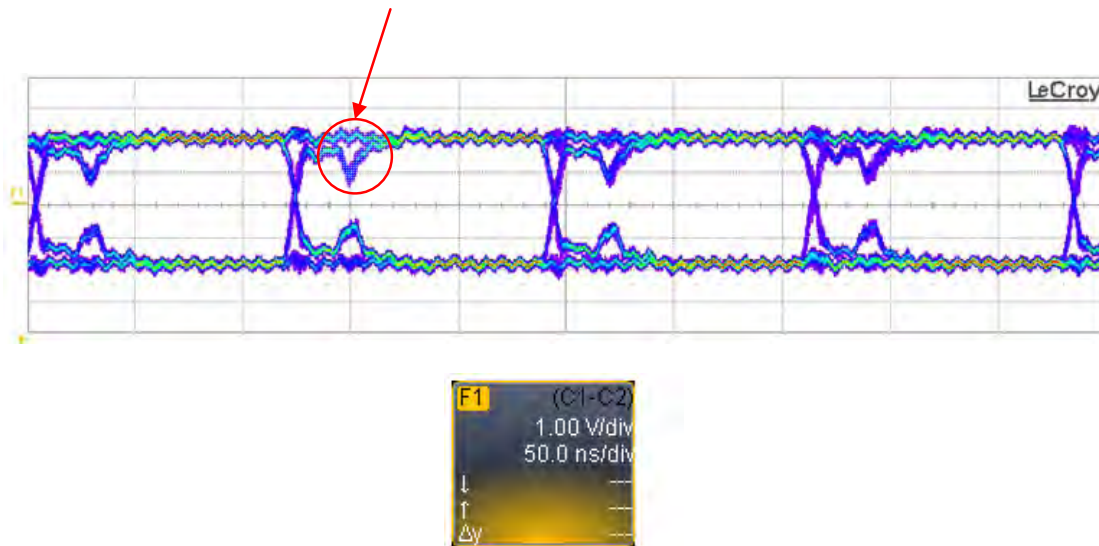


Figure 5-36: Eye diagram constructed whilst levitating

By visually inspecting the eye diagram in Figure 5-36 ringing is once again observed - as indicated by the circle - as well as a greater noise floor. However a good amplitude separation and low jitter is still observed which will not result in bit errors.

The aim however is to validate the communication system under *worst case conditions* concerning noise. This will be done by not just levitating the rotor, but *spinning* the rotor which will add even more noise to the system. Extra noise will be present due to the motor drive, which is an immense noise source present in the system.

As mentioned previously some advanced digital oscilloscopes have the ability to continuously save eye diagram data in a 3 dimensional database. This 3 dimensional database consists of values determined by constructing histograms at various different cross sections. An eye diagram was

constructed whilst the ADES was fully operational (*rotor was spinning*) in 3D just to illustrate exactly how the data is represented in the oscilloscope. Figure 5-37 shows the 3D representation.

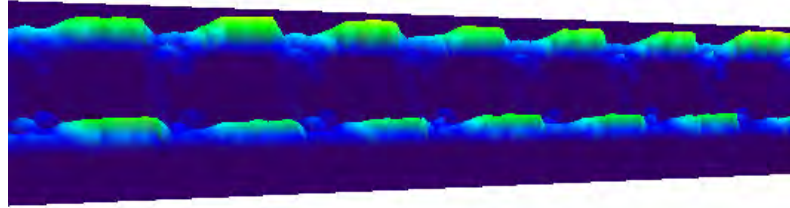


Figure 5-37: 3D representation of the eye diagram whilst the ADES was fully operational

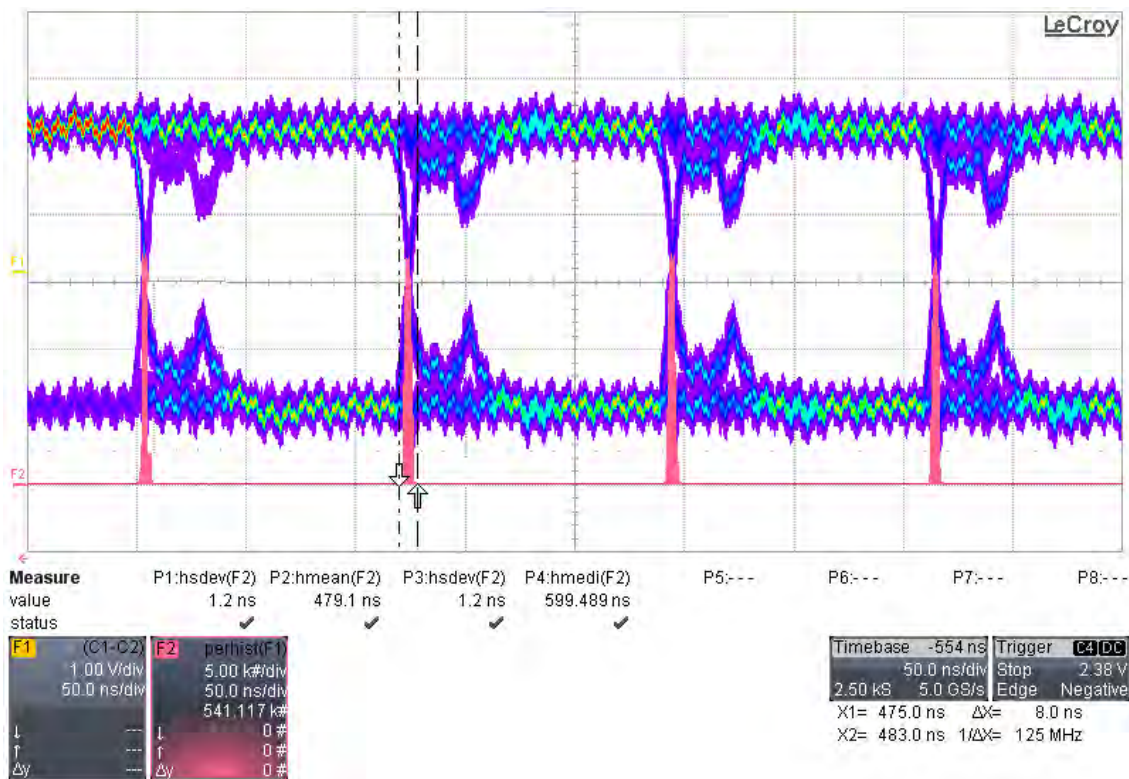


Figure 5-38: Eye diagram construction and a persistence histogram

Persistence based statistics were used to validate the physical layer under *worst case conditions*. In Figure 5-38 the eye diagram is constructed and a persistence histogram (*Phistogram function*) is used to analyse the timing jitter. F1 is the eye diagram construction of the differential signal and F2

is the persistence histogram constructed by specifying a *narrow horizontal slice* through the eye diagram and acquiring the data point in that specific range.

The horizontal slice was centred at 0 V which is precisely in the middle of the eye diagram and the slice width was selected as 1.40 V. Four histograms are shown in Figure 5-38, but only the two left handed histograms will be used to determine the statistical properties of T_{cross1} and T_{cross2} as shown in Figure 5-10. The measured statistical properties are listed just under F1 and F2 and are shown in Table 5-28.

Table 5-28: Statistical properties obtained from horizontal slicing.

P1:hsdev(F2)	σ	T_{cross1}	1.2 ns
P2:hmean(F2)	μ	P_{cross1}	479.1 ns
P3:hsdev(F2)	σ	T_{cross2}	1.2 ns
P4:hmean(F2)	μ	P_{cross2}	5.99 ns

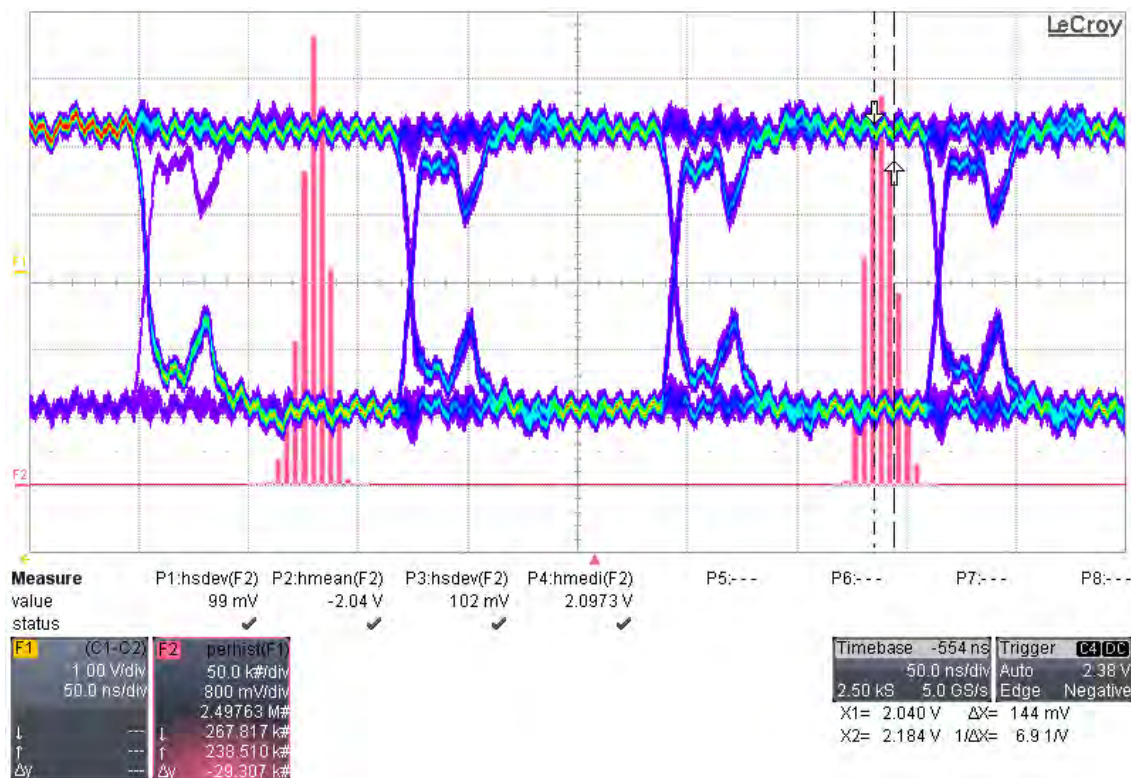


Figure 5-39: Eye diagram constructed as well as vertical persistence histogram

In Figure 5-39 a second type of persistency histogram is shown to study other properties of the eye diagram. F1 is once again the eye diagram of the differential signal and F2 is the *persistency histogram* obtained by vertical slicing. These histograms are constructed by specifying a narrow

vertical slice through the eye diagram and acquiring the data point in that specific range. The properties are computed by subtracting the right handed histogram (logic level '1' properties) from the left handed histogram (logic level '0' properties)

The vertical slice was centred at the mid-point of the eye diagram, and the slice width was 33 ns. This is an eye sampling window of approximately 28% of the unit interval. The right most histogram provides the *top* values and the left most histogram provides the *base* values as shown in Figure 5-10. The measured statistical properties are listed just under the F1 and F2 and are shown in Table 5-29. Once again the distributions are assumed to be Gaussian due to small sample sizes.

Table 5-29: Statistical properties obtained from vertical slicing

P1:hsdev(F2)	σ	P_{top}	102 mV
P2:hmean(F2)	μ	P_{top}	2.09 V
P3:hsdev(F2)	σ	P_{base}	99 mV
P4:hmean(F2)	μ	P_{base}	-2.04 V
<hr/>			
P1:hsdev(F2)	σ	T_{cross1}	1.2 ns
P2:hmean(F2)	μ	P_{cross1}	479.1 ns
P3:hsdev(F2)	σ	T_{cross2}	1.2 ns
P4:hmean(F2)	μ	P_{cross2}	599.59 ns

5.8.2.1 Eye height can be written as:

$$\text{Eye height} = (\mu_{P_{top}} - 3\sigma_{P_{top}}) - (\mu_{P_{base}} + 3\sigma_{P_{base}}) \quad (5.14)$$

where $\mu_{P_{top}}$ and $\mu_{P_{base}}$ are the mean values of P_{top} and P_{base} respectively. $\sigma_{P_{top}}$ and $\sigma_{P_{base}}$ is the standard deviation from the mean values of P_{top} and P_{base} respectively.

These values are obtained from Table 5-29 and substituted into (5.14) to obtain the eye height.

$$\begin{aligned} \text{Eye height} &= (2.09 - 3(0.102)) - (-2.04 + 3(0.099)) \\ &= 3.53 \text{ V} \end{aligned}$$

5.8.2.2 Eye width can be written as:

$$\text{Eye height} = (\mu_{T_{cross2}} - 3\sigma_{T_{cross2}}) - (\mu_{T_{cross1}} + 3\sigma_{T_{cross1}}) \quad (5.15)$$

where $\mu_{T_{Cross2}}$ and $\mu_{T_{Cross1}}$ are the mean values of T_{cross1} and T_{cross2} respectively. $\sigma_{T_{Cross2}}$ and $\sigma_{T_{Cross1}}$ is the standard deviation from the mean values of P_{top} and P_{base} respectively.

$$\begin{aligned}\text{Eye height} &= (599.59 - 3(1.2)) - (479.1 + 3(1.2)) \\ &= 113.29 \text{ ns}\end{aligned}$$

The next values that will be determined by using the eye diagrams statistical properties are peak to peak jitter, RMS jitter and 6 sigma jitter. These values are related. 6 sigma jitter accounts for 99.999% of all jitter (which is peak-to-peak jitter) and RMS jitter accounts for 1 sigma jitter as shown in (5.16) [67].

$$6 \text{ sigma jitter} = \text{peak to peak jitter} = 6 \times \text{RMS jitter} \quad (5.16)$$

5.8.2.3 Jitter RMS is written as:

$$\text{Jitter RMS} = \sigma_{T_{cross1}} \quad (5.17)$$

where, $\sigma_{T_{cross1}}$ is the standard deviation at T_{cross1} .

The standard deviation of T_{cross1} is obtained from Table 5-29 and jitter RMS is calculated by using (5.17).

$$\begin{aligned}\text{Jitter RMS} &= \sigma_{T_{cross1}} \\ &= 1.2 \text{ ns}\end{aligned}$$

5.8.2.4 Jitter 6 sigma can be written as:

$$6 \text{ sigma jitter} = 6\sigma_{T_{Cross1}} \quad (5.18)$$

where, $\sigma_{T_{cross1}}$ is the standard deviation at T_{cross1} . The standard deviation of T_{cross1} is obtained from Table 5-29 and 6 sigma jitter is calculated by using (5.18).

$$\begin{aligned}6 \text{ sigma jitter} &= 6(1.2) \\ &= 7.2 \text{ ns}\end{aligned}$$

5.8.2.5 Jitter percentage was explained in detail in Section 5.5.2.2 and can be written as:

$$\text{Percentage Jitter} = \frac{\text{Percentage Crossing Skew}}{\text{Unit Interval}} \times 100 \quad (5.19)$$

The percentage jitter is calculated by using (5.20) in conjunction with the relevant values in Table 5-29.

$$\begin{aligned}\text{Percentage Jitter} &= \frac{7.2}{120} \times 100 \\ &= 6\%\end{aligned}$$

5.8.2.6 Noise RMS can be written as:

$$\text{Noise RMS} = \sigma_{P_{top}}, \text{ or } \sigma_{P_{base}} \quad (5.20)$$

where $\sigma_{P_{top}}$, or $\sigma_{P_{base}}$ are the standard deviations of P_{top} and P_{base} respectively. Noise RMS is calculated by using (5.21) and the standard deviations of P_{top} and P_{base} can be obtained from Table 5-29.

$$\text{Noise RMS} = 102 \text{ mV or } 99 \text{ mV}$$

5.8.2.7 Noise peak to peak can be written as:

$$\text{Noise peak to peak} = \begin{cases} \max(P_{top}) - \min(P_{top}), \text{ or} \\ \max(P_{base}) - \min(P_{base}) \end{cases} \quad (5.21)$$

where $\max(P_{top})$ and $\max(P_{base})$ are the maximum values of P_{top} and P_{base} respectively. $\min(P_{top})$ and $\min(P_{base})$ are the minimum values of P_{top} and P_{base} respectively.

Noise peak to peak can be calculated by using (5.22) and Figure 5-40.

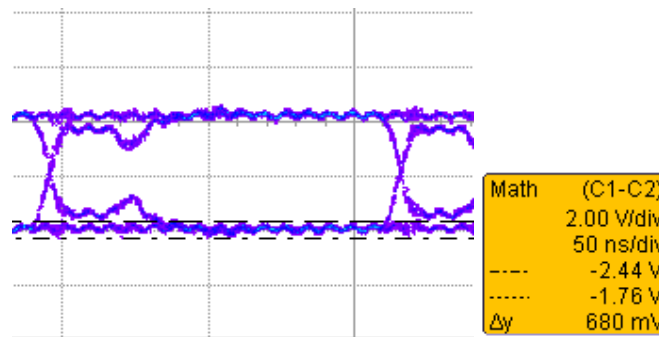


Figure 5-40: Noise peak to peak (fully operational)

$$\text{Noise peak to peak} = 680 \text{ mV}$$

5.8.2.8 The signal to noise ratio (SNR) can be written as:

$$\text{SNR} = \frac{\mu_{P_{Top}} - \mu_{P_{Base}}}{\sigma_{P_{Top}} + \sigma_{P_{Base}}} \quad (5.22)$$

where $\mu_{P_{Top}}$ is the mean value of P_{top} and $\mu_{P_{Base}}$ is the mean value of P_{base} . $\sigma_{P_{Top}}$ and $\sigma_{P_{Base}}$ are the standard deviations from the means of P_{top} and P_{base} respectively.

The SNR is calculated by using (5.23) and the values are obtained from Table 5-29.

$$\begin{aligned} \text{SNR} &= \frac{2.09 - (-2.04)}{0.102 + 0.099} \\ &= 20.55 \end{aligned}$$

The dB value of the SNR can be calculated by using:

$$\begin{aligned} &20\log_{10} SNR \\ &20\log_{10} 20.55 = 26.25 \text{ dB} \end{aligned} \quad (5.23)$$

5.8.3 Eye measurement conclusion

When interpreting the results, the following is noted: The eye's openness is considered to be good, with an eye width of 113.29 ns and an eye height of 3.54 V. This is good considering that the ideal eye for this application will have an eye width of 120 ns and an eye height of 4 V. A very low percentage of jitter was calculated, keeping in mind that a conservative guide for low jitter signals is 5% of the unit interval (UI). Therefore the assumption can be made that timing jitter will not cause bit errors in the ADES.

Although the noise floor observed is greater than the noise floor observed when the system is not operational, the physical layer does a very good job in shielding the digital transmission from the immense amount of noise present in the ADES. The SNR can be related directly to the bit error ratio (BER) in digital networks. Fewer bit errors occur when the SNR is higher, increasing the data throughput. The calculated SNR during validation is 26.26 dB which barely has discernible effect on the channel; however with a SNR of 10 dB the opening of the eye would have been barely noticeable.

Figure 5-41 is an illustration of an eye diagram which shows where the decision points are for the communication system – where the receiver will make the decision whether it is a logic '1' or logic '0'. It can be seen that samples are taken in the region called the eye window which is typically the centre 20% of the bit period [65]. By comparing this figure to the eye diagram constructed when the system is operational it can be seen that a very *large noise margin* exists. Thus it can also be assumed that the noise will not cause bit errors in the ADES.

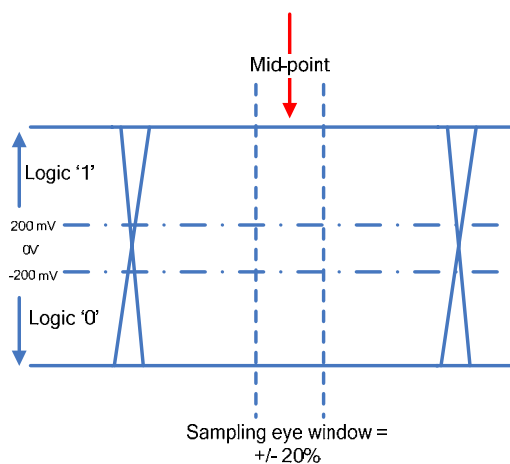


Figure 5-41: Eye diagram's sampling window

The last question that remains is will the ringing in the system result in bit errors? As mentioned previously the communication system over-samples 16 times to ensure that the data is sampled in the midpoint of the eye. The unit interval is 120 ns which is the reciprocal of the 8.31 MHz, implying that each of the bits is divided into 16 sections which are 7.5 ns apart. The receiver can be off by at most $\frac{1}{16}$ from the mid point of the eye, thus the sampling (decision) point will be 60 ns \pm 7.5 ns which is not in the ringing section. In the rare case that the sampling point drifts toward the region where the ringing occurs, it will still not result in bit errors. This is motivated by measuring from the crossing point to the ringing effect and obtaining amplitude separation of ± 900 mV which is still sufficient to register as a logic '1' or logic '0'.

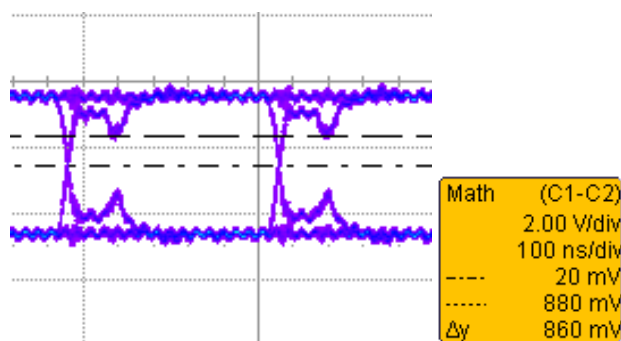


Figure 5-42: Amplitude separation measured at ringing effect

5.8.4 Impedance mismatch

The various eye diagrams showed undershoot. The cause of the problem was determined to be impedance mismatches that cause reflections which can appear as overshoot, undershoot or ringing in eye diagrams [71].

When reviewing the hardware setup as shown in Figure 5-43 of the internal communication system the following was noted: The digital I/O mezzanine card situated on the main controller and the communication controller is connected to the internal functional units by a 2 m SCSI cable which consists of 34 twisted pairs. This SCSI cable slots into the termination panel and from here the BLDN 9841 cables connect to the functional units. When measuring the resistance of the SCSI cable, it is noted that the cable is un-terminated. Furthermore it is also known that there is no termination resistors located on the digital I/O mezzanine card, which consists of the transceivers.

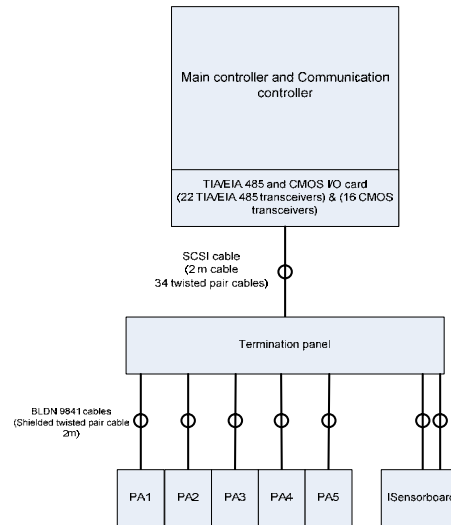


Figure 5-43: Overview of internal communication system

The termination method implemented between the internal functional unit and the termination panel is simple: Parallel termination is implemented to avoid reflection on the BLDN 9841 shielded twisted pair cables. Therefore these cables are terminated at the functional unit end with 120Ω resistors which equal the nominal characteristic impedance of the BLDN 9841 cable and on the termination panel also with 120Ω resistors.

This presents the following issue: Although the BLDN 9841 cable is properly terminated the SCSI cable is left un-terminated. This causes an impedance mismatch between the two cables. During the design it was considered that the termination depended exclusively on the characteristic impedance of the BLDN 9841 cable as shown in Figure 5-44. However the situation at hand is more similar to Figure 5-45.

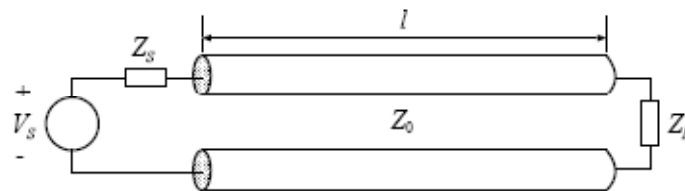


Figure 5-44: Termination of the BLDN cable

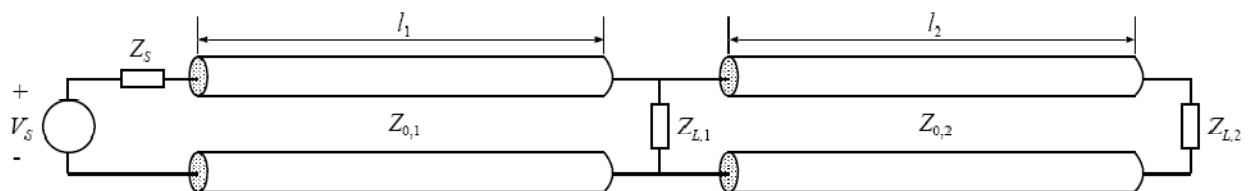


Figure 5-45: Termination of the BLDN cable and the SCSI cable

The only way to eliminate this problem is to implement an impedance matching network.

5.9 Data link layer – Lower level assembly

The physical layer only delivers a “raw” bit stream with no guarantees that it is the correct data. The data link layer has the ability to ensure that the correct data is transmitted and to detect errors. In this section the performance of the developed data link layer of the ADES digiComm protocol can be validated by monitoring the data link layer whilst the ADES is fully operational under different operating conditions.

5.9.1 Validating framing format

Framing is validated whilst the ADES is fully operational and the rotor of the induction machine is spinning, probes are connected over the differential communication channel connecting the ISensorboard with the main controller. Data will be saved on a digital oscilloscope to analyse whether the framing is correct. Figure 5-46 is an illustration of a data frame.

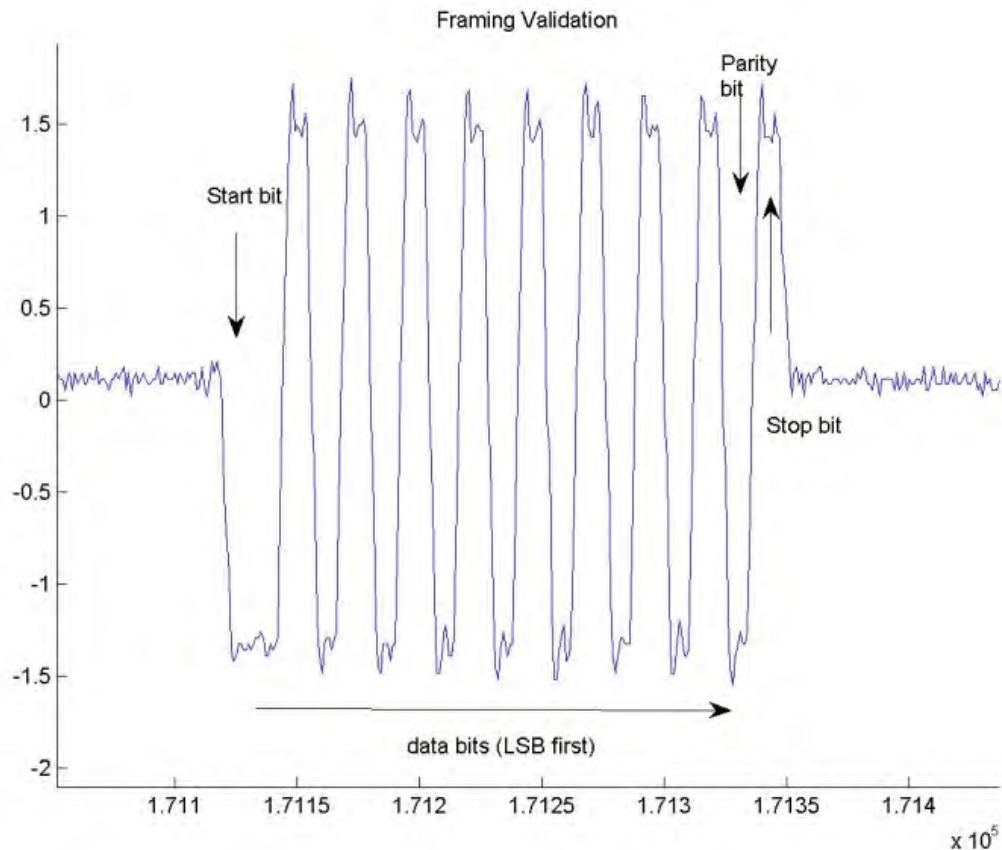


Figure 5-46: Framing validation

The following is observed when studying Figure 5-46. A start bit is issued which is a transition to '0'. Then the data value follows, which is AAAA in hex which corresponds to 1010101010101010 in binary. The least significant bit (LSB) is transmitted first. After the data is transmitted the parity bit is transmitted which is a logic '0' indicating even parity. Lastly a stop bit, which is a transition to '1', is transmitted. Then the communication system goes into a default idle state.

The framing is validated as well as the parity calculation. The conclusion can be made that the VHDL code simulated during the verification phase, synthesised the correct physical circuits on the FPGAs to produce the correct framing.

5.9.2 Validating data frames

The ADES will only function if the correct values are communicated between the functional units. In this section the data frames communicated between the functional units will be monitored while the ADES is *operational*.

5.9.2.1 ISensorboard

Figure 5-47 is an illustration of the data available between the ISensorboard and the Main controller under *normal operating* conditions. The following is observed when studying the waveform:

1. The first four frames transmitted are data frames that consist of the X, Y, Z position values and the error condition as determined by the communication controller situated on the ISensorboard.
2. The last frame is an error frame which consists of the CRC calculated from the four data frames.
3. The ISensorboard is continuously transmitting these values to the main controller.
4. No error condition is present on the communication controller situated on the ISensorboard. This is seen when studying the error value data frame. This frame consists of all logic zeros except for the last bit which is a logic one that corresponds to the default values of the stop bit of the framing structure.
5. After each of the frames is transmitted, the communication controller goes back into default *idle state*.

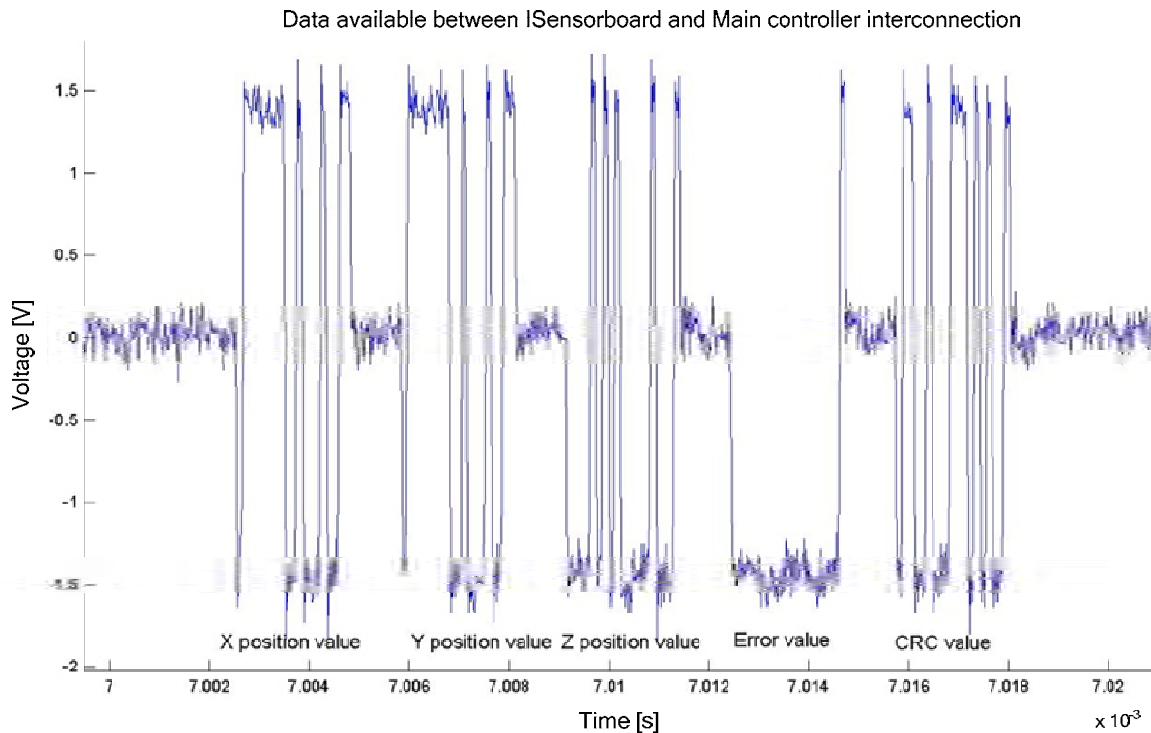


Figure 5-47: Data communicated between the ISensorboard and the main controller

The same is observed on the second communication channel of the ISensorboard.

5.9.2.2 Power amplifiers

Figure 5-48 is an illustration of the data communicated between the main controller and a power amplifier unit under *normal operating conditions*. The following is observed when studying this particular waveform.

1. The main controller continuously communicates the following data to the power amplifier board;
 - The main controller starts off by transmitting the current reference value as determined by using the position values received from the ISensorboard. This is a data frame.
 - Next the power amplifier unit command is transmitted. This command is either an off or on command. This is also a data frame.
 - The next value transmitted is the CRC calculated from the two data frames, which is the error frame.

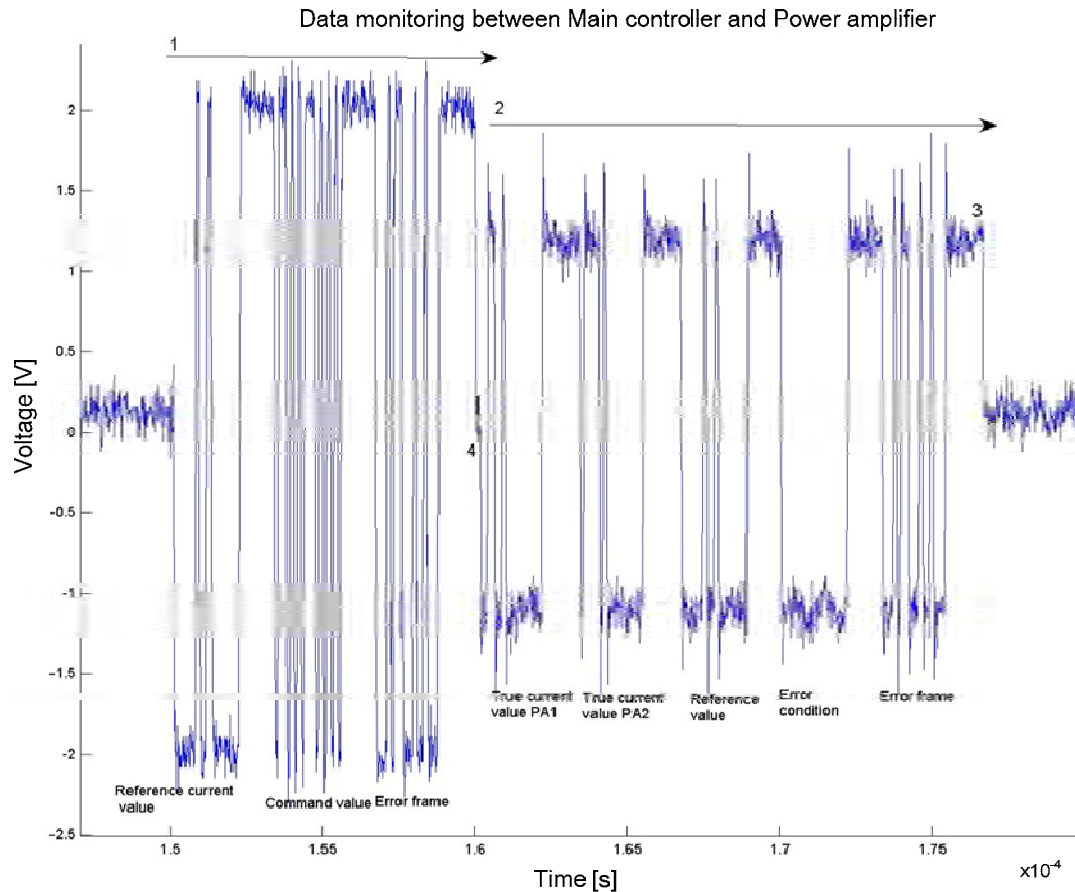


Figure 5-48: Data communicated between power amplifier unit and main controller

2. After the power amplifier unit has received these data values, the power amplifier unit immediately starts off by transmitting data values back to the main controller. These data values include;
 - Both of the true current values.
 - The received reference values are transmitted back. The reason for this will be discussed in 5.10.4. This is also a data frame.
 - Next the error condition as determined by the communication controller situated on the power amplifier unit is transmitted. This value is zero, indicating no errors. This is also a data frame.
 - Lastly the error frame is transmitted back, which consists of the CRC calculated from all the data values transmitted to the main controller.
3. After each of the frames is transmitted the serial communication line does not immediately go to the default idle state. The reason for this is to compensate for the delay on the power amplifier drivers.
4. Lastly it can also be seen that the direction is controlled correctly, because there is no data collision between the two different devices that communicate over the same transmission line.

5.9.3 Validating the CRC controller

One of the most important functionalities of the data link layer is the ability to detect more than one bit inversion by implementing a CRC controller in the various communication controllers. The CRC controller was discussed in detail in Chapter 4. Until now two questions concerning the CRC controller has not been answered. Will the CRC controller be synthesisable on the FPGAs? Will the execution time of the CRC controller cause a delay that can jeopardise the performance of the communication controllers?

In this section the CRC controller functionality will be validated while the ADES is *operational*. The interface that will be used to validate the CRC controller is the interface between the ISensorboard and the main controller. Probes will be connected over the differential communication channel that connects the ISensorboard and the main controller. The data communicated between these two functional units will be obtained by using a digital oscilloscope, where after the data will be analysed by using MATLAB®.

The data will be analysed by writing a program in MATLAB® that can decode each of the frames considering that the scope serial decoder cannot analyse 16 bit data frames. For the ISensorboard the X, Y, Z position values, error value and CRC value will be plotted in decimal values. The X, Y, Z position values and error values will be randomly sampled and these values will be supplied to the Modelsim® simulation used in Section 5.6.2. The CRC values calculated by the Modelsim® simulation will then be compared to the CRC values obtained by the CRC controller implemented on the FPGAs.

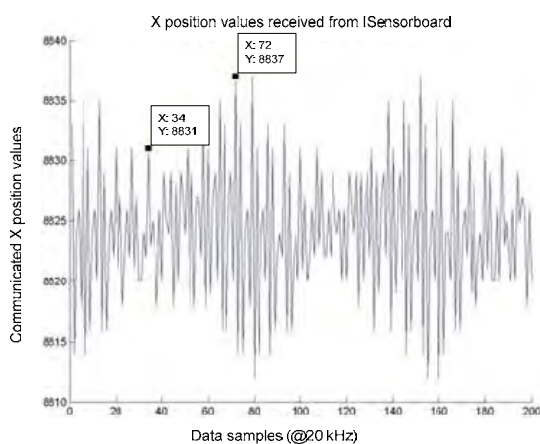


Figure 5-49: Decoded X values

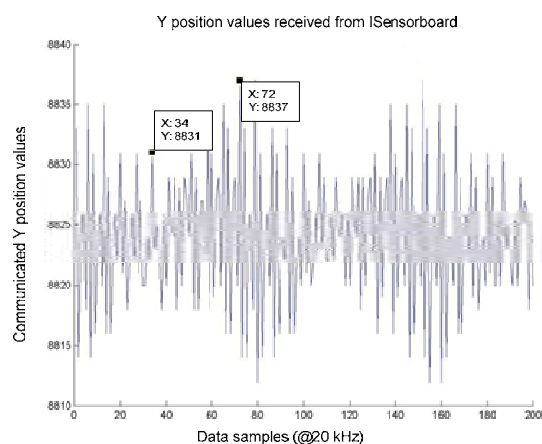


Figure 5-50: Decoded Y values

Figure 5-49 shows 200 decoded X position values and Figure 5-50 shows 200 decoded Y position values. The first random sample was taken at 34 and the second random sample was taken at 72.

Figure 5-51 shows 200 decoded Z position values and Figure 5-52 shows 200 decoded error values. The first random sample was taken at 34 and the second random sample was taken at 72.

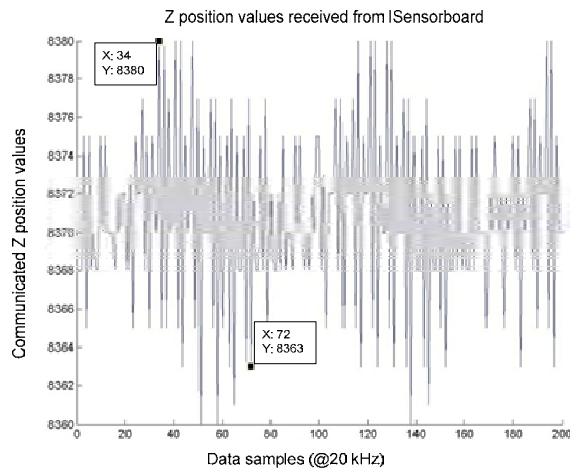


Figure 5-51: Decoded Z values

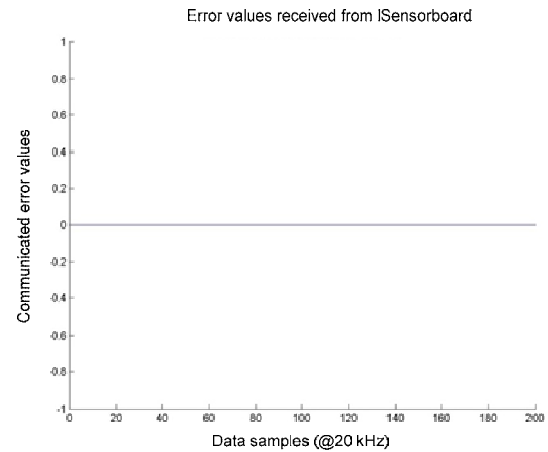


Figure 5-52: Decoded error values

Table 5-30: Values obtained from random samples

Random Sample	Frame	Decimal value	Hexadecimal
34	X position value	8831	227F
	Y position value	8831	227F
	Z position value	8380	206C
	Error	0	0000
72	X position value	8837	2285
	Y position value	8837	2285
	Z position value	8363	20AB
	Error	0	0000

The values obtained from sample 34 were concatenated and supplied to the Modelsim® simulation described in 5.7.2. The simulated result obtained is shown in Figure 5-53.

Data (Position values X,Y,Z and error)	227F227F20BC0000
CRC register	0000000000000000
CRC remainder (unsigned)	32594

Figure 5-53: Simulation results obtained from sample data 34

The values obtained from sample 72 were concatenated and also supplied to the Modelsim® simulation. The simulated result obtained is shown in Figure 5-54.

Data (Position values X,Y,Z and error)	2285228520AB0000
CRC register	0000000000000000
CRC remainder	62C8

Figure 5-54: Simulation results obtained from sample data 72

The next step is to compare the CRC obtained by the simulation against the CRC generated by the CRC controller situated on the FPGA. Figure 5-55 shows 200 decoded CRC values received from the ISensorboard. The CRC values are sampled at 34 and 72 where after the results are compared in Table 5-31. Figure 5-56 shows the precise value of the CRC sampled at 72 and not the rounded value, which can cause confusion in the results.

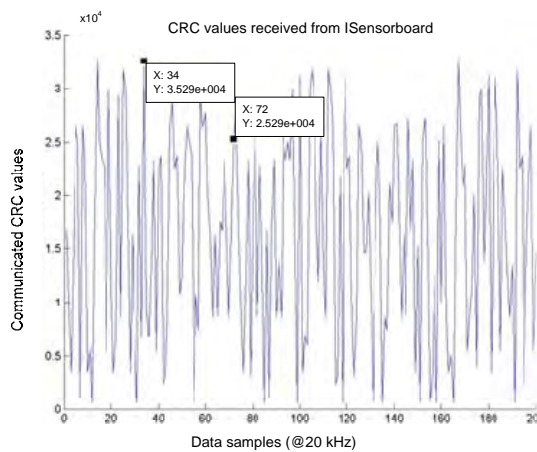


Figure 5-55: Decoded Error frame

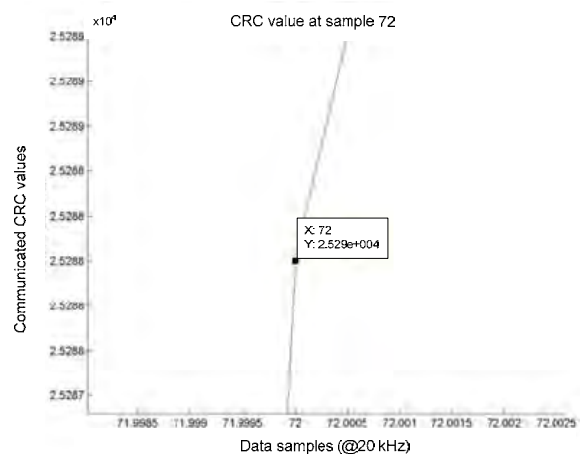


Figure 5-56: Accurate value

Table 5-31: Comparing results

Random sample	Simulation		CRC controller	
	Hexadecimal	Decimal	Hexadecimal	Decimal
34	7F52	32594	7F52	32594
72	62C8	25288	62C8	25288

The conclusion can be made that the CRC controller is fully synthesisable on the FPGAs and functions as stipulated in chapter 4. The question concerning the timing will be answered during the validation of the timing section.

5.9.4 Validation under error conditions

The data link layer has different responses to different error conditions to ensure reliability of the ADES. In this section the communication controllers that interface between the ISensorboard and the main controller will be subjected to error conditions and the data link layer response will be monitored. The power amplifier interconnection error was test on precisely the same method; it however is not documented to avoid the dissertation from becoming to drawn out.

5.9.4.1 Loss of synchronization signal (Main controller – Error 1)

During normal operation the sync signal (C3 signal) enables both of the communication controllers on the rising edge to establish communication between the ISensorboard and the main controller. Immediately following the enable signal the ISensorboard starts to transmit the required values (math signal) to the main controller as shown in Figure 5-57.

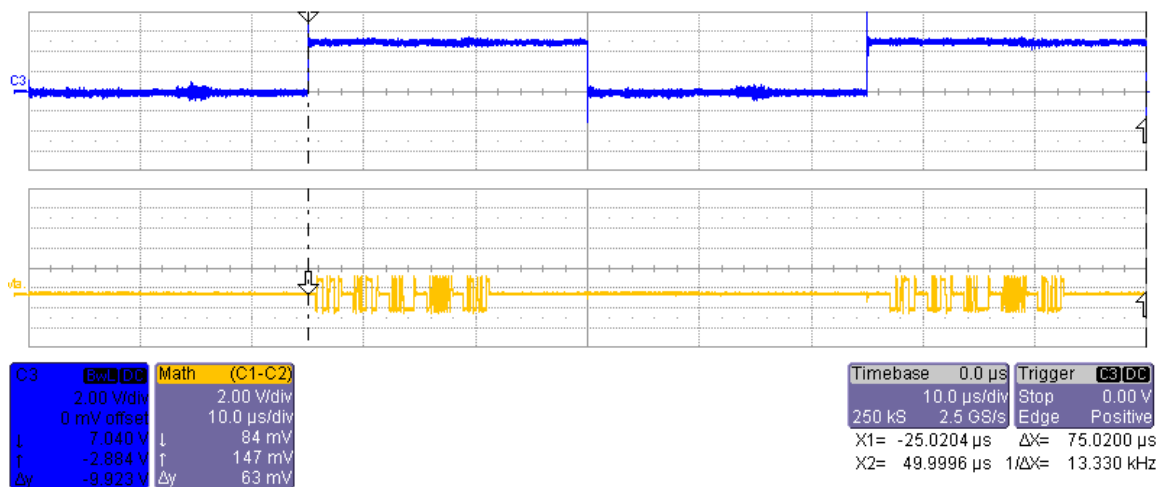


Figure 5-57: Normal operation

The first error that will be invoked is the loss of the sync signal between the ISensorboard and the main controller. In Figure 5-58 the sync signal (C3) is disconnected and the response of the data link layer (math) is monitored while the rotor is suspended.

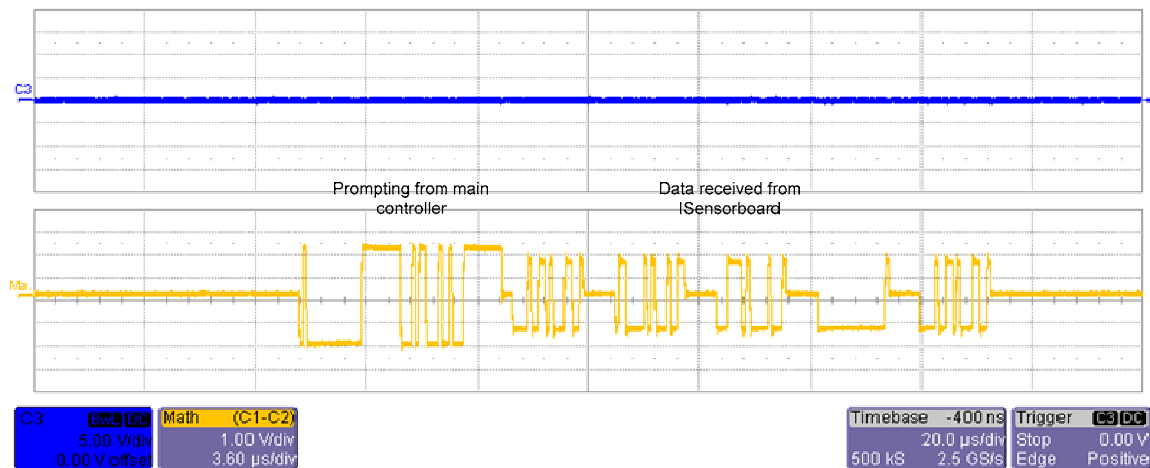


Figure 5-58: Response to loss of sync signal

The data link layer responds by writing error condition 1 into the DPR on the main controller as well as prompting the ISensorboard to transmit the data required for the control procedure. The prompting value received from the main controller is 1, where after the CRC is sent. This enables the ISensorboard to respond by sending the X, Y, Z position values as well as the error condition where after the CRC follows. Throughout this induced error the ADES was levitating without any glitches.

Whether the correct error is flagged in the DPR will be determined by analyzing the address where the error data is saved. This will be done by writing the DPR error data into the DDR and monitoring it. In Figure 5-60 it can be seen that the communication controller flagged the correct error.

```
Address Read : fffff68
Value read : 1
Error: Sync connection lost
```

Figure 5-59: Command line indicating that the sync error was flagged.

5.9.4.2 CRC mismatch (Main controller – Error 2)

The next test will involve the hard coding of the CRC value on the main controller communication controller, and monitoring the data link layer response. The data link layer must respond by

flagging an error condition 2 and writing the error condition in the DPR, indicating that data corruption has occurred. It is then left to the main controller decide when to invoke a critical stop.

In order to analyse whether an error was written into the DPR, the address that points to the error data is analysed. This will be done by writing the DPR error data into the DDR and monitoring it. In Figure 5-60 it can be seen that the communication controller responded correctly.

```
Address Read : fffff68
Value read  : 2
Error: CRC mismatch
```

Figure 5-60: Command line indicating that the CRC mismatch error was flagged

5.9.4.3 CRC mismatch (ISensorboard– Error 1)

The necessity for this test is too determined, whether an erroneous prompt command was received from the main controller. This test will involve hard coding the CRC value on the main controller and monitoring the response of the communication controller on the ISensorboard. The data link layer must respond by transmitting error value 1 to the main controller.

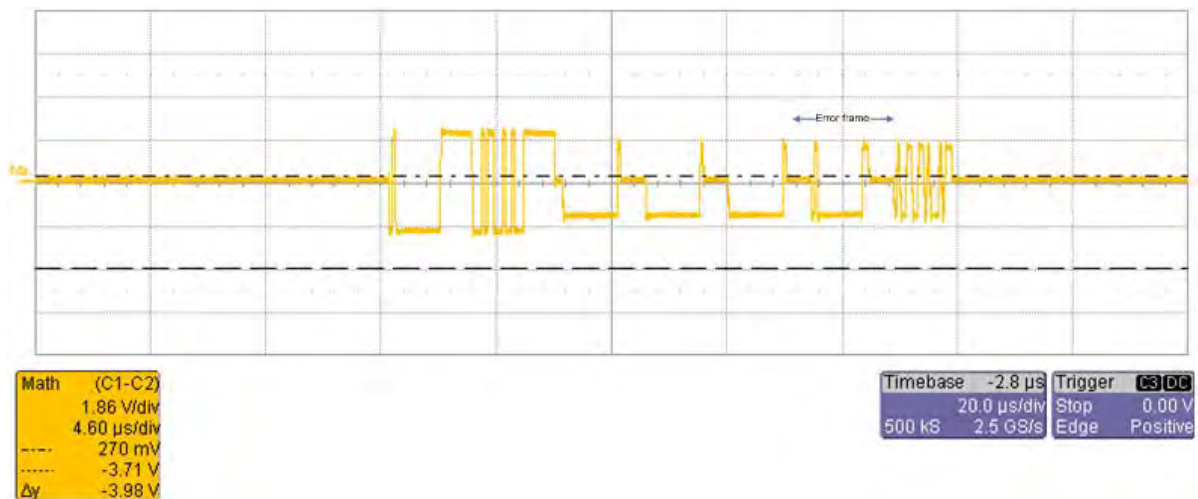


Figure 5-61: Data received from main controller was corrupted

As can be seen once the CRC values mismatch the error frame transmits a 1 indicating data corruption.

5.9.5 Validating timing and synchronization

Two of the *most important aspects* of a communication system that operates at such low levels of assembly is the *timing requirements and the synchronization requirements*. As explained in detail in chapter 4, the ADES will not be able to function if the timing of the system does not execute

precisely as stipulated in Figure 4-9. The most important requirement of the timing is that each of the communication cycles may not exceed more than $50 \mu\text{s}$. Furthermore the necessity for synchronization is to ensure that the system is not transmitting whilst the A/Ds are sampling which will cause noise that can possibly corrupt the data. Lastly the synchronization signal enables the communication controller when new data is available to be communicated. During normal operation the data that will be communicated between the ISensorboard and the main controller takes $16.95 \mu\text{s}$ as shown in Figure 5-62.

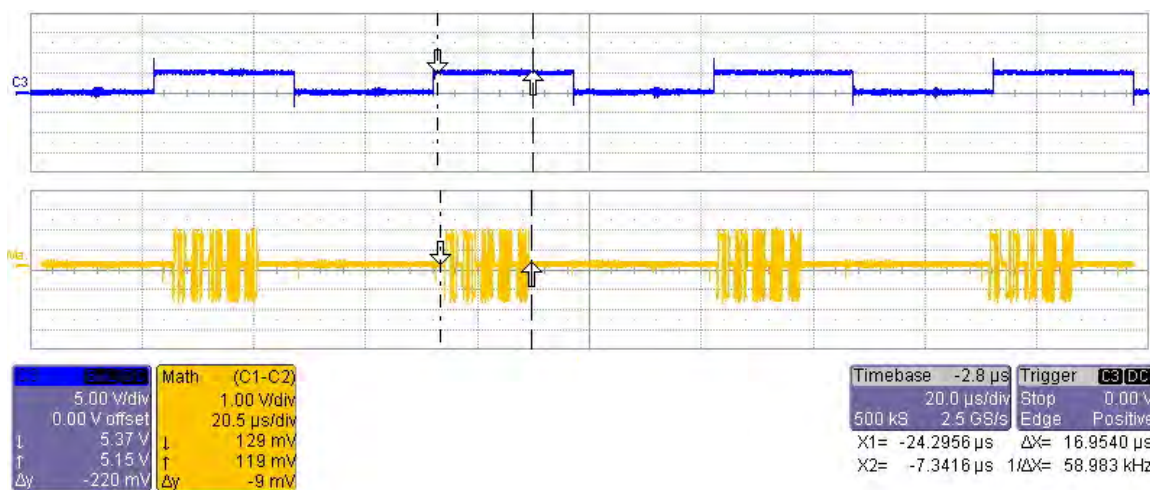


Figure 5-62: ISensorboard timing during normal operation

The highest amount of data that is transmitted at a time between the ISensorboard and the main controller is during the prompting phase. During the prompting phase the communication controllers will still be synchronized, because the internal sync signal generated on the FPGA is not lost, only the external sync signal cable malfunctions due to a possible connection break. Figure 5-63 shows that during an error operation the data communicated between the ISensorboard and the main controller take $25.13 \mu\text{s}$ which is almost half the specification. This is faster than obtained during verification, because the receiver timeout was set to an optimum value.

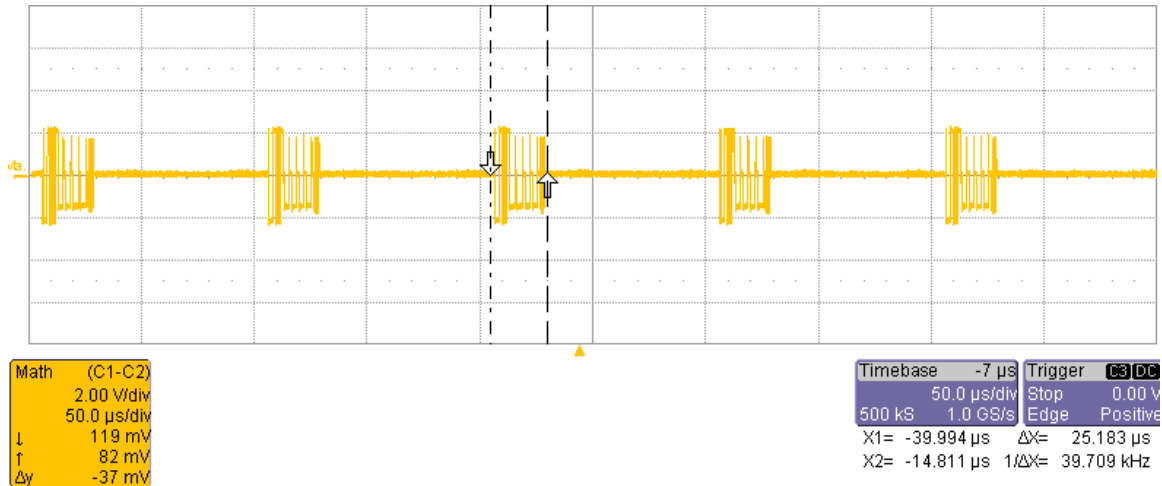


Figure 5-63: Communication timing between the ISensorboard and the main controller

The communication controllers that enable the communication between the power amplifiers and the main controller promptly react to the sync signal as shown in Figure 5-65. The highest amount of data that will be communicated between the power amplifiers and the main controller at a time are shown in Figure 5-65. The reference current value, the command value and the CRC are transmitted from the main controller to the power amplifiers. Then the power amplifiers respond by sending back the two true current values, the error condition and the CRC. To illustrate expandability the current reference value is also sent back and still the power amplifier communication cycle executes in 27.02 μ s. This is once again almost half the specified limit.

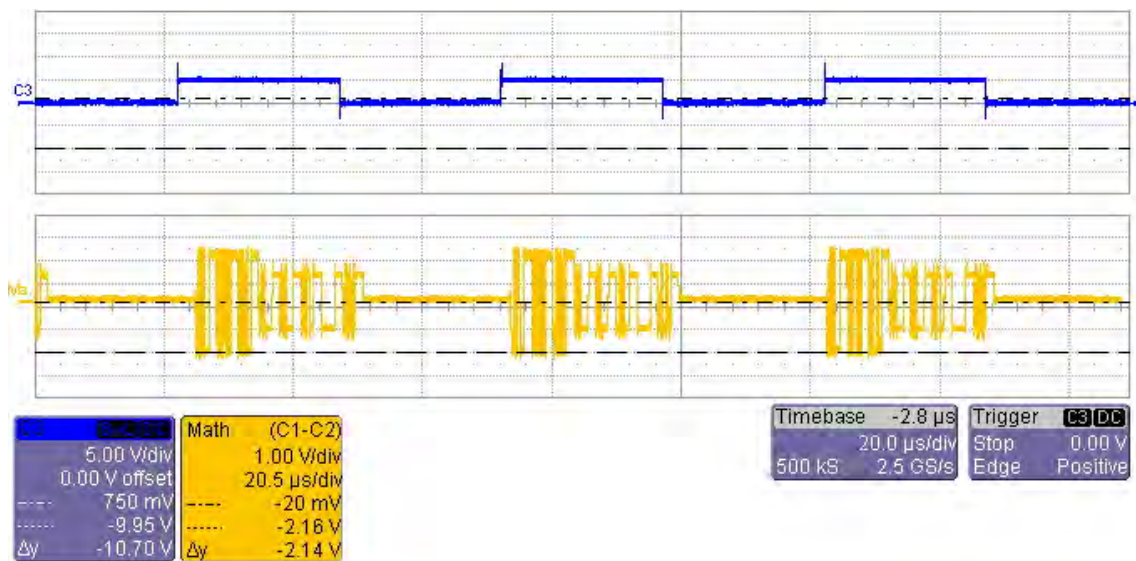


Figure 5-64: Power amplifier response to the sync signal

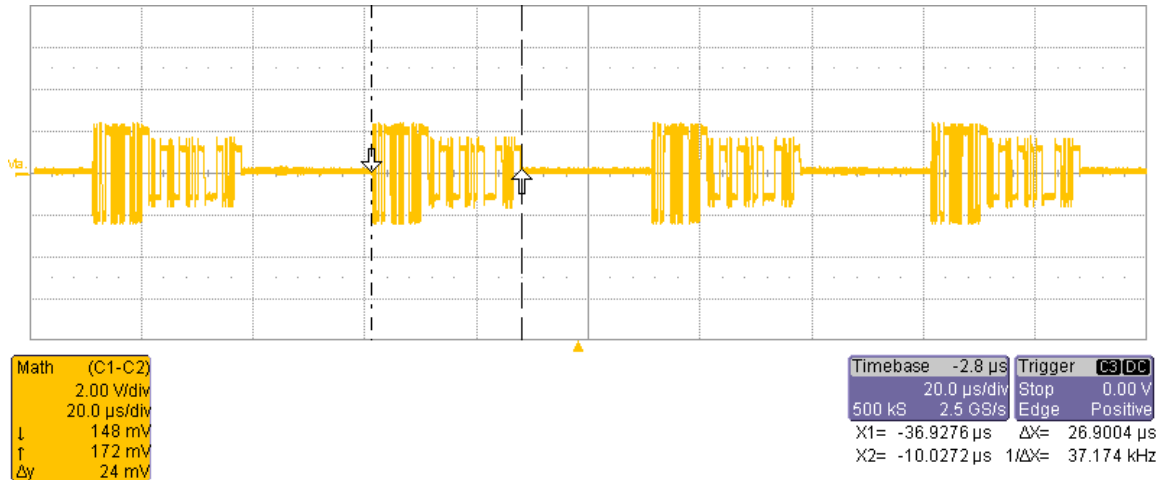


Figure 5-65: Communication timing between the Power amplifiers and the Main controller

Thus the conclusion can be made that the timing of the internal functional units are well within the system requirements.

5.10 Communication controllers and validation

5.10.1 Data Value estimation

Another method of validating whether the communication controllers function as required is by value estimation. Value estimation involves determining whether the values transmitted and received by the functional units are correct. For example the main controller generates a current reference value which is between -6 A and 6 A to be transmitted to the power amplifiers. These values are then scaled and written into the dual port rams (DPRs). The communication controller is signalled that new data is available in the DPRs to be communicated. These data values are then handled as discussed in detail in Chapter 4. After the data is processed the frames are placed on the physical layer to be communicated.

Value estimation will be done by reading data values saved into the DPRs by the communication controller out of the DPRs and into the double data rate synchronous dynamic random access memory (DDR) which is accessed by the PCI bus and transmitted to the single board computer (SBC) while the *ADES* is fully operational (rotor is suspending). These values are then saved to be analysed by MATLAB®. By doing this it will give a clear indication whether the communication controllers function correctly by reading and writing the correct data at the correct times into and out of the DPRs to be accessed by the processors. The data will also be studied for possible outliers, which can indicate possible data corruption.

5.10.1.1 ISensorboard value estimation

In this section the values transmitted by the ISensorboard will be analysed by saving the communicated values in the DDR while the ADES fully operational. The data communicated from the ISensorboard to the main controller is the un-scaled raw demodulated A/D values. These values transmitted for the X, Y and Z positions are shown in Figure 5-66. This however does not give any relevant information regarding whether the correct values are communicated.

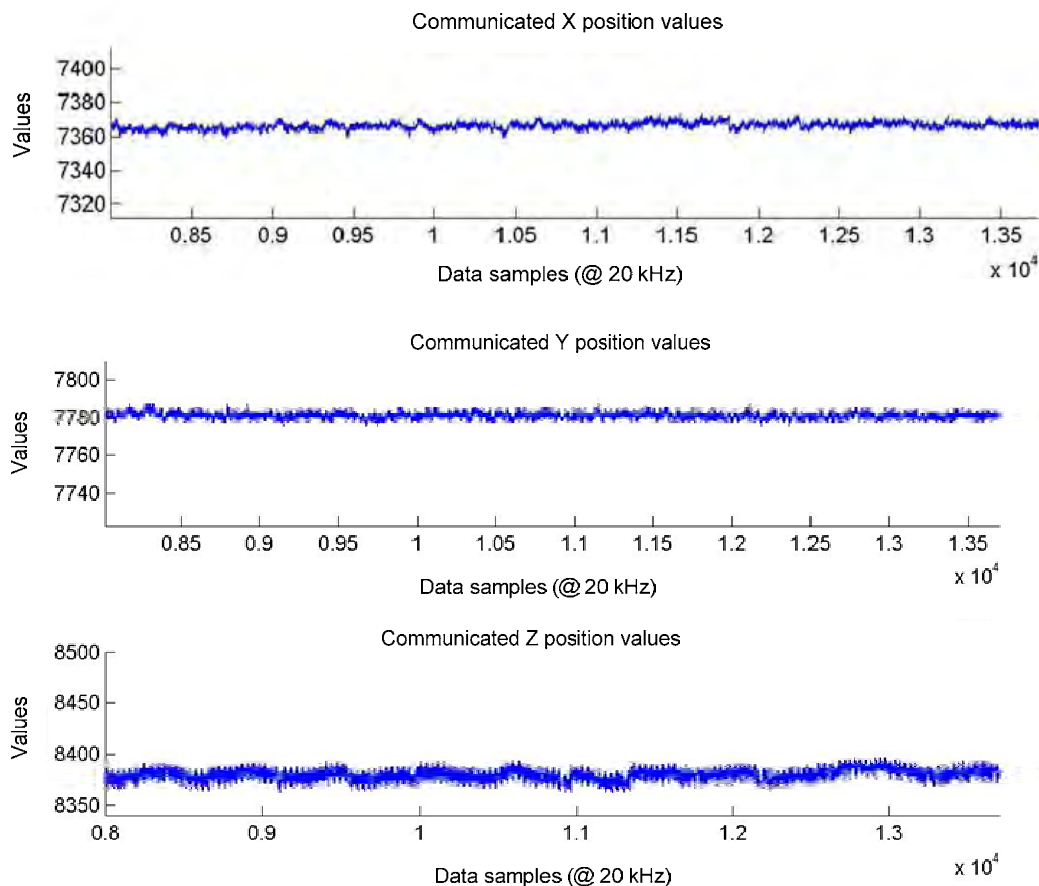


Figure 5-66: Communicated ISensorboard values

In Figure 5-67 the un-scaled values transmitted are scaled. Now it is clear to see that there are no outliers. The X, Y and Z position varies slightly due to noise picked up by the sensors in the bearing module; however a constant position value for all the position values are maintained throughout the time the data was logged.

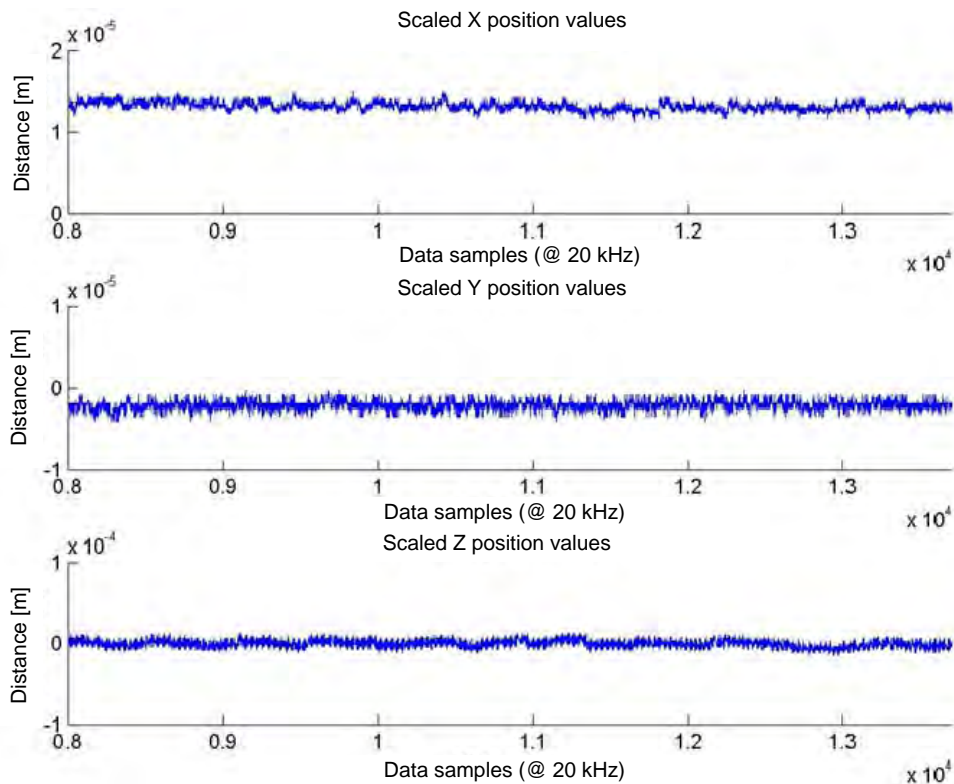


Figure 5-67: Scaled communicated values received from ISensorboard

The correct values are transmitted between the ISensorboard and the main controller, because it is in the exact range as specified by the ISensorboard, furthermore no outliers are observed.

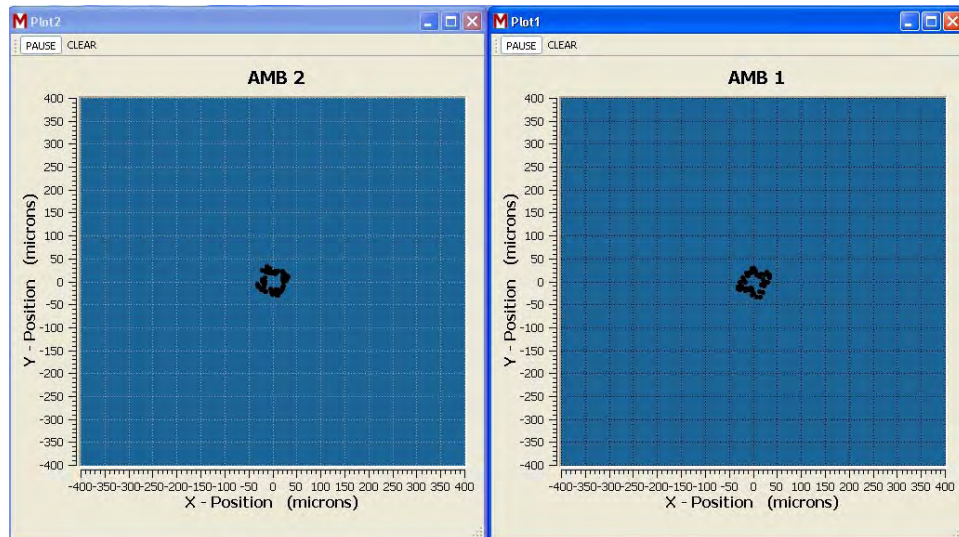


Figure 5-68: User interface displaying the position values

In Figure 5-68 the user interface is shown which constantly updates the position values received back from the ISensorboard. The user interface was programmed in Microsoft Visual Studio. The values are obtained by reading all the DPR values in which the communicated values are written into the DDR, the PCI bus accesses the DDR and transmits the values to the SBC, where the program continuously plots the data to enable system monitoring. It can also be seen that the rotor is levitated in the centre.

5.10.1.2 True current value estimation

In this section the values transmitted from the power amplifier to the main controller will be analysed by saving the data written into the DPRs into the DDR while the ADES is fully operational. Once again the data communicated from the power amplifiers to the main controller is un-scaled A/D values as shown in Figure 5-69, which does not give any relevant information whether the correct values are communicated.

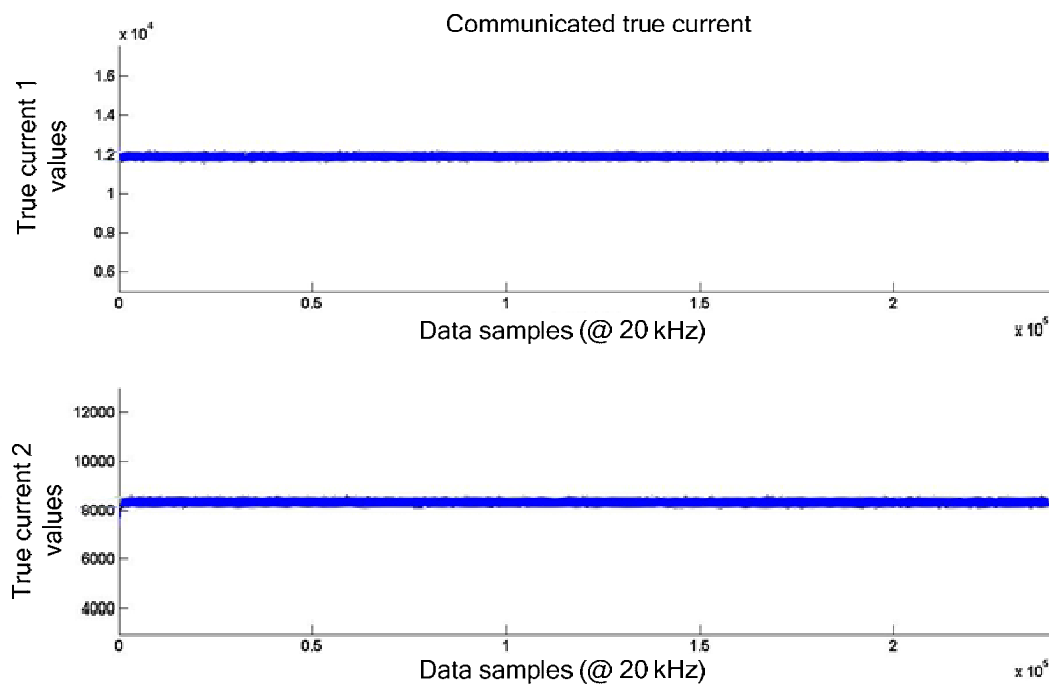


Figure 5-69: Raw unscaled true current values received from power amplifier board 1

The true current values communicated must be between 0 A to 17 A. Furthermore the power amplifiers are biased at 5.5 A. Therefore the data must range just above the region of ± 5.5 A and just below the region of 5.5 A for the two power amplifiers. The communicated true current values are scaled and this provides the values as shown in Figure 5-70.

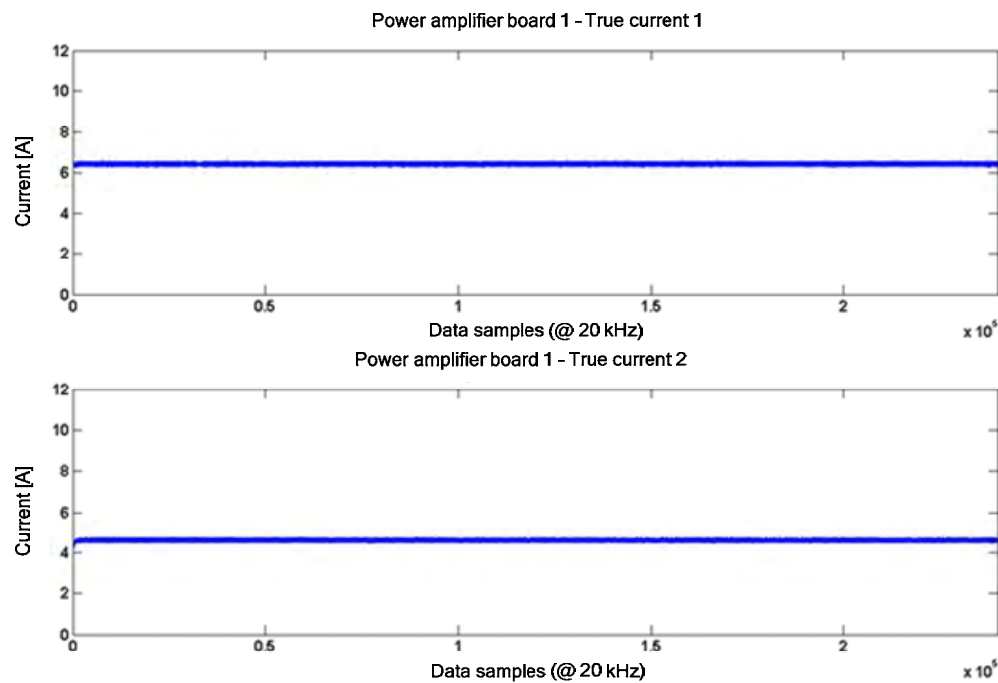


Figure 5-70: Scaled true current values received from power amplifier board 1

It is observed that the true current value one is just above 6 A and constantly stays in that region whilst the ADES is suspending. The same is observed for power amplifier true current value two which stays in the vicinity of ± 4.5 A. This is very accurate according to the predictions. Lastly not one outlier is observed while 250 000 data samples were logged at a rate of 20 kHz (sampling frequency). In Figure 5-71 the user interface is shown which constantly updates the true current values received back from all the power amplifiers. The user interface was programmed in Microsoft Visual Studio. The values are obtained by reading all the DPR values in which the communicated values are written into the DDR, the PCI bus accesses the DDR and transmits the values to the SBC, where the program continuously plots the data.

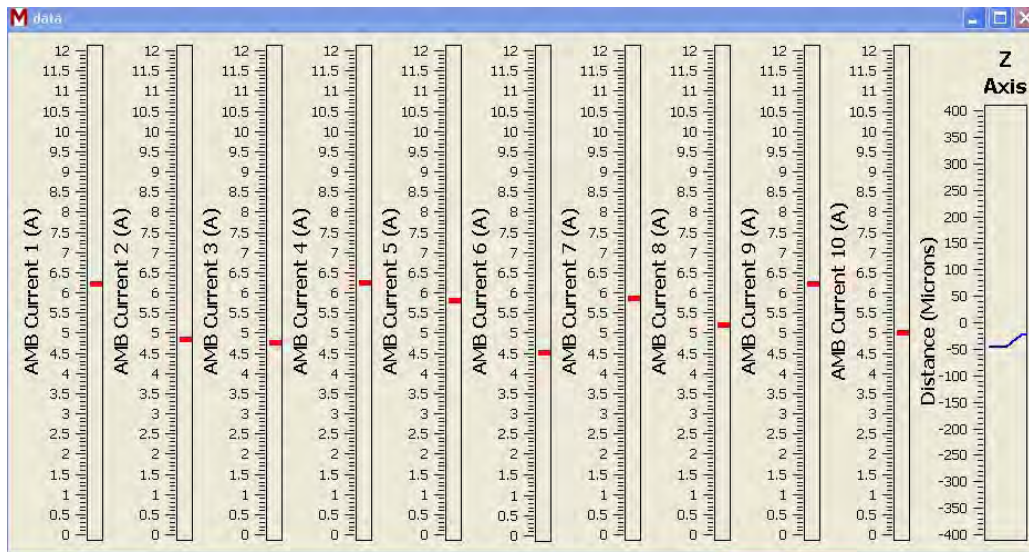


Figure 5-71: User interface of true current values

5.10.1.3 Reference current value estimation

Determining whether the correct reference values are communicated to the power amplifiers will be challenging. Considering that although the values can be accessed by logging the DPR values on the main controller and reading it into the DDR of the SBC, it cannot be determined whether these values are received by the power amplifiers.

The method that will be used to analyse the communicated current reference values is by monitoring the physical layer between the power amplifiers and the main controller by using a digital oscilloscope and saving as much data as possible. This data will then be analysed by writing a program in MATLAB® that will remove the overhead from the frame and convert the data values into decimal values to be analysed. Once again this test will be conducted whilst the ADES is suspending.

In Figure 5-72 the current reference values received by the power amplifier board are shown. These values are determined by the PowerPC and written into the DPR to be handled by the communication controller. The communicated values can be related to the true values by dividing the values by 1000 (to convert the values from floating point to fixed point), and then the reference current is obtained. As can be seen from the figure, the values range between 0 A and 800 mA. This is once again in the specified range of between -6 A and 6 A. Once again no obvious outliers are observed.

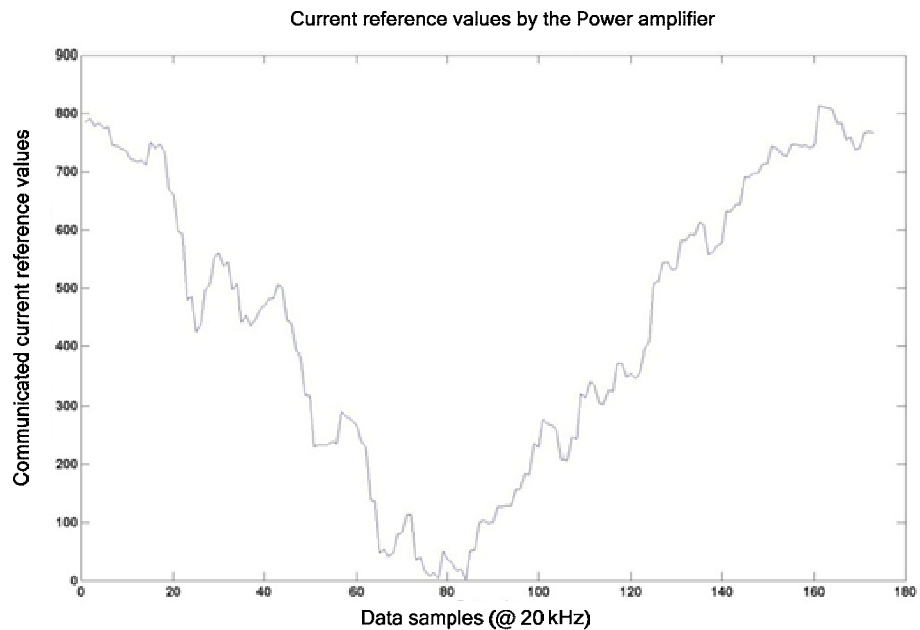


Figure 5-72: Communicated reference value

5.11 Analogue vs. Digital

The previous dSPACE® system setup is shown in Figure 5-73. This AMB system did not employ any digital communication. When comparing the dSPACE® setup with the ADES setup as shown in Figure 5-74 it can be seen that this system is certainly more susceptible to noise than the ADES. Noise has an effect on the dSPACE® system at a variety of places. The places where noise is most likely to influence the dSPACE® system is clearly indicated in Figure 5-73. However although some of the noise is filtered out, it is most likely that the noise will surface again after being filtered.

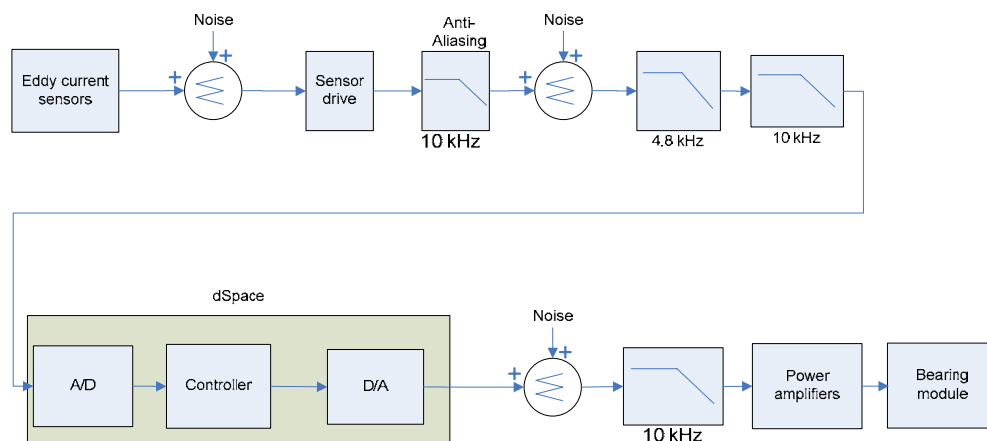


Figure 5-73: dSPACE® setup

In Figure 5-74 it is observed that in the ADES noise can only influence the system at one stage, which is at the beginning of the control process where the position values are sensed. This noise is further minimized by attenuating the position signal values by 80 dB at 20 kHz which is the switching frequency of the power amplifiers which cause tremendous noise in the system. Furthermore this in-house developed sensor driver consists of all the necessary components to enable digital communication. It was decided to compare the previous system's position signals to the ADES position signal to show to what extent this system is more noise immune.

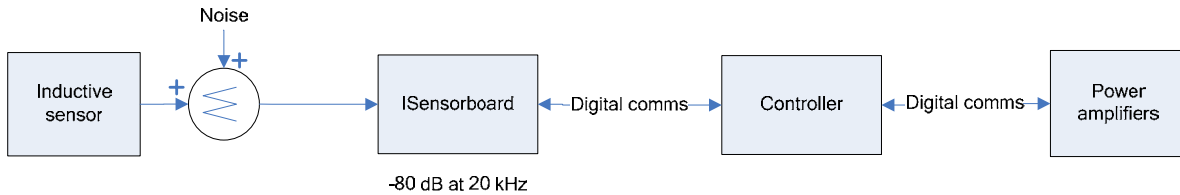


Figure 5-74: ADES setup

In order to compare the noise levels of two systems the position signal was logged while the bearing module was suspended by using the previous dSPACE® system and again while the bearing module was suspended by the ADES. When studying the noise levels of the dSPACE® system a 16.6 μm variation was detected as shown in Figure 5-75. The ADES, however only had a variation of $\pm 2.5 \mu\text{m}$ as shown in Figure 5-76.

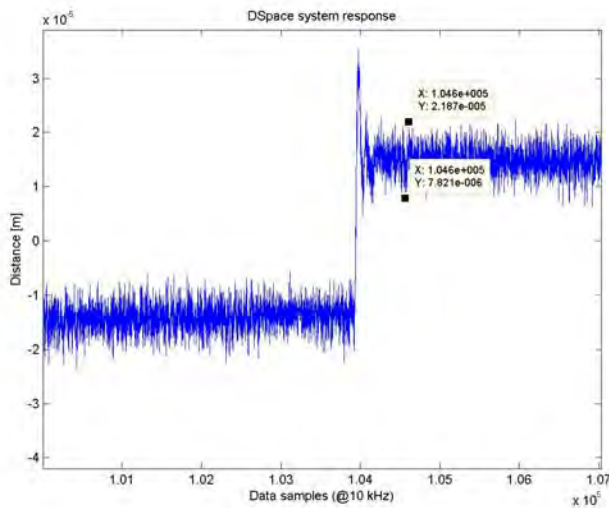


Figure 5-75: Position signal of dSPACE®²⁰

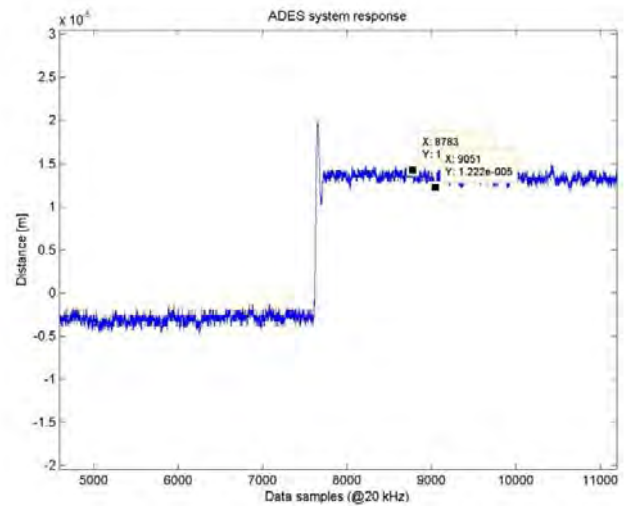


Figure 5-76: Position signal of ADES

²⁰ dSPACE® is limited at a sampling frequency of 10 kHz, however the ADES sampling frequency is every 20 kHz.

The conclusion can be made that the noise immunity of the system is increased tremendously by developing a *totally digital system*, which would not have been possible without implementing digital communication.

5.12 Communication system validation – highest level of assembly

The last part of validation will be conducted at the highest level of assembly in order to identify whether the totally digital system (including digital communication) contributes to an effective AMB system. This will require conducting a sensitivity analysis of the total AMB system including both the ADES (the integrated electronic controller package which is responsible for the effective control of the bearing module) and the bearing module.

Traditional bearing modules are considered stable, where AMBs are considered to be naturally unstable. This is caused by negative stiffness which is caused by static magnetic force. In order to evaluate the stability of the AMBs an ISO standard was formulated. The ISO 14839-3 standard proposes a sensitivity function [72]. This particular sensitivity function will be used to determine the sensitivity of the AMBs controlled by the ADES.

This sensitivity function is given as:

$$G_s(s) = 20 \log \frac{V_R(s)}{V_D(s)} \quad (5.24)$$

where, V_D is the sinusoidal disturbance of which the frequency is varied and V_R is the error signal. In Figure 5-77 it can be seen where V_D is injected into the control loop and where the error signal V_R is measured. Figure 5-78 is a functional breakdown of how the sensitivity analysis is done [72].

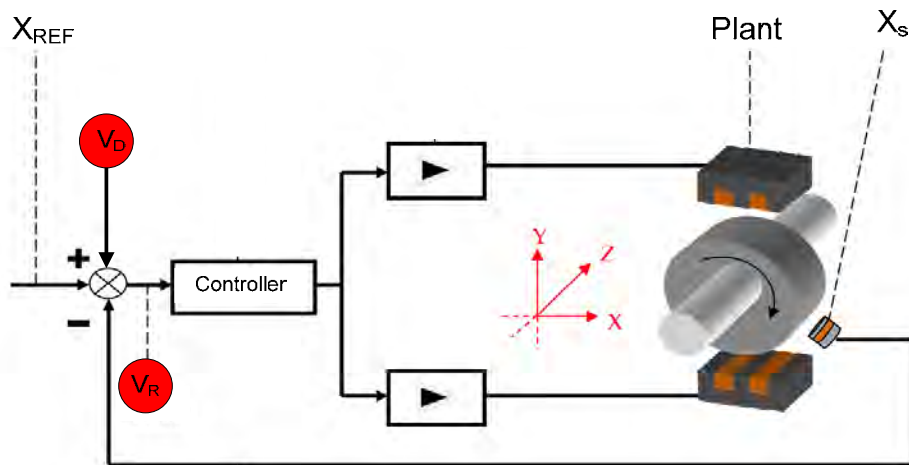


Figure 5-77: Sensitivity analysis setup [73]

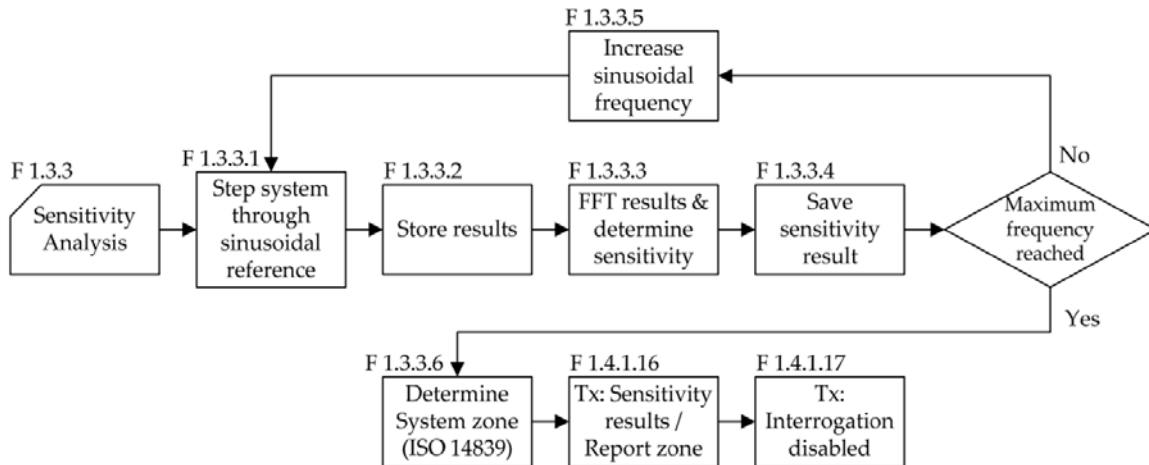


Figure 5-78: Sensitivity analysis functional breakdown [46]

The ISO standard stipulates the following [72][74]:

- The sensitivity has to be characterized up to a frequency, f_{max} , of 3 times the rated speed or 2 kHz. The value that yields the highest f_{max} must be used.
- Each of the AMBs has to be analysed.
- The worst case results among the different control axes will provide the rating.

The ratings can be divided into 4 zones as shown in Table 5-32 [74].

Table 5-32: Sensitivity ratings

Zone	Peak sensitivity
A/B	8 dB
B/C	12 dB
C/D	14 dB

The different Zones indicate the following [72][74]:

Zone A: If the sensitivity rating is characterized as Zone A the system is considered to be a newly commissioned system.

Zone B: If the system is within the Zone B category the system is suitable to operate in unrestricted long-term applications.

Zone C: This indicates that the system cannot operate for long periods of time and is only suitable to operate for a short period until the system can be corrected.

Zone D: If the sensitivity rating is within this category the system can cause damage and it is advised not to operate the machine.

The sensitivity analysis was conducted in “An integrated controller for an Active Magnetic Bearing and Drive Electronic System” [75]. The results obtained are shown in Figure 5-79 and Figure 5-80.

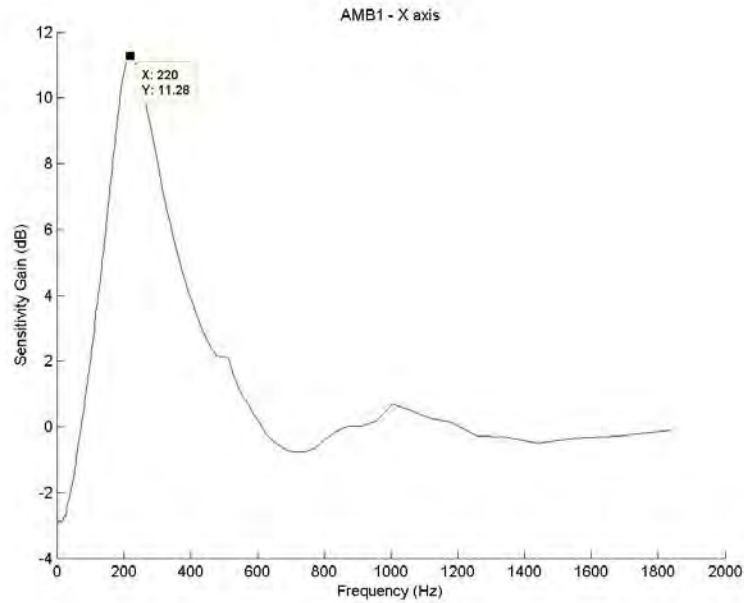


Figure 5-79: Radial AMB's sensitivity of the X axis

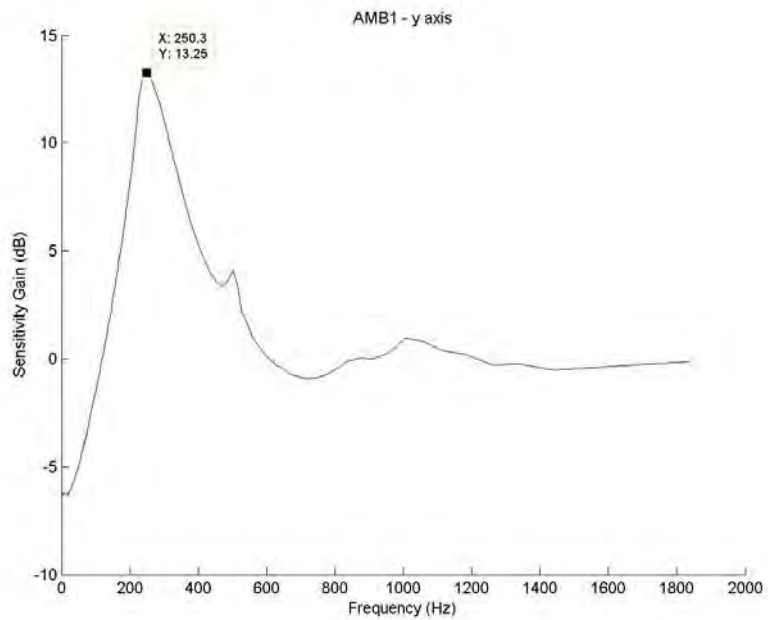


Figure 5-80: Radial AMB's sensitivity of the Y axis

From these results it can be seen that the total system including the ADES and the bearing module falls in Zone C and cannot operate for long periods of time and is only suitable to operate for a short period until the system can be corrected. This is due to the fact that the PID control values were not designed optimally, [75] and does not involve the developed communication sub-system.

5.13 Conclusion

In this chapter thorough test and evaluation plans were formulated to verify and validate the design of the internal communication system. After the verification test plan was conducted objective evidence was provided that showed that the specifications of the internal communication system were satisfied. This was done by conducting rigorous testing on each of the lower assembly units present in the internal communication system. Furthermore the lower assembly units were also tested at design conditions that exceeded the specified limits and the result showed that both the physical and the data link layer was able to handle these circumstances.

The formulated test and evaluation plan for validation was also conducted with a successful outcome which again provided objective evidence that the requirements of the project were fulfilled. This was done by testing the lower and the higher level assembly units in the intended environment while the ADES was operational. Furthermore the ADES was also subjected to error conditions to further validate the response of the system. The overall system performance was also analysed and compared to the previous system. Finally the performance of the system at the highest assembly level was tested to observe whether the digital communication contributes to an effective AMB system. The results of all the tests conducted during the validation phase were exceptionally good.