### University of Nebraska - Lincoln DigitalCommons@University of Nebraska - Lincoln

Biological Systems Engineering: Papers and Publications

**Biological Systems Engineering** 

2016

# Comparing various hardware/software solutions and conversion methods for Controller Area Network (CAN) bus data collection

Samuel E. Marx University of Nebraska-Lincoln, samuel.e.marx@gmail.com

Joe D. Luck University of Nebraska-Lincoln, jluck2@unl.edu

Santosh Pitla University of Nebraska-Lincoln, spitla2@unl.edu

Roger M. Hoy University of Nebraska-Lincoln, rhoy2@unl.edu

Follow this and additional works at: https://digitalcommons.unl.edu/biosysengfacpub

Part of the <u>Acoustics, Dynamics, and Controls Commons, Bioresource and Agricultural</u> <u>Engineering Commons, Computer-Aided Engineering and Design Commons, Controls and Control</u> <u>Theory Commons, Electrical and Electronics Commons, Electro-Mechanical Systems Commons,</u> <u>Environmental Engineering Commons, Other Civil and Environmental Engineering Commons,</u> <u>Systems and Communications Commons, and the VLSI and Circuits, Embedded and Hardware</u> <u>Systems Commons</u>

Marx, Samuel E.; Luck, Joe D.; Pitla, Santosh; and Hoy, Roger M., "Comparing various hardware/software solutions and conversion methods for Controller Area Network (CAN) bus data collection" (2016). *Biological Systems Engineering: Papers and Publications*. 474. https://digitalcommons.unl.edu/biosysengfacpub/474

This Article is brought to you for free and open access by the Biological Systems Engineering at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Biological Systems Engineering: Papers and Publications by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Published in *Computers and Electronics in Agriculture* 128 (2016), pp 141–148. doi:10.1016/j.compag.2016.09.001 Copyright © 2016 Elsevier B.V. Used by permission. Submitted 20 November 2015; revised 23 August 2016; accepted 2 September 2016; published online 7 September 2016.



# Comparing various hardware/software solutions and conversion methods for Controller Area Network (CAN) bus data collection

Samuel E. Marx, Joe D. Luck, Santosh K. Pitla, Roger M. Hoy

University of Nebraska–Lincoln, Biological Systems Engineering, 223 L.W. Chase Hall, Lincoln, NE 68583-0726

Corresponding author — J. D. Luck. Email addresses — samuel.e.marx@gmail.com (S. E. Marx), jluck2@unl.edu (J. D. Luck), spitla2@unl.edu (S. K. Pitla), rhoy2@unl.edu (R. M. Hoy)

#### Abstract

Various hardware and software solutions exist for collecting Controller Area Network (CAN) bus data. Digital data accuracy could vary based upon different data logging methods (e.g., hardware/software timing, processor timing, etc.). CAN bus data were collected from agricultural tractors using multiple data acquisition solutions to quantify differences among collection methods and demonstrate potential data accumulation rates. Two types of data were observed for this study. The first, CAN bus frame data, represents data collected for each line of hex data sent from an ECU. One issue with frame data is the resulting large file sizes, therefore a second logging format collected was an averaged frame signal, or waveform dataset. Because of its smaller file size, waveform data could be more desirable for long periods of collection. Percent difference was calculated from two sets of frame data logs using different hardware/software combinations, and a frame data log was also compared to a waveform data log. The resulting difference was less than 0.0025 RPM for engine speed comparisons, zero for fuel rate and fuel temperature comparisons, and the mean percent difference was less than 0.08% between the methods of data collection. The error production could have resulted from noise in hardware and processor times, but was not found to increase as time progressed. This showed that even though errors existed between logging methods, the magnitude of errors would not negatively impact any practical agricultural field research applications. Thus, data logged by the different devices was similar and files requiring less memory would be desired. Selecting a waveform CAN bus data logging option would likely maintain digital data accuracy while reducing file storage and processing needs.

Keywords: Communication, SAE J1939, Field machinery, Digital data logging

#### 1. Introduction

Controller Area Network (CAN) bus use and data logging have become increasingly common in many industries. In the agricultural sector, the CAN bus has become a common source of equipment operational data. A great deal of detailed information is transmitted through the CAN bus regarding field machinery functions (Stone et al., 2008). Many typical row crop tractors today have 12–20 electronic control units (ECU) that are sharing sensed information as well as control signals regarding machine operation. Because there is so much information being broadcast on these machines, many have found it a useful resource to gain greater perspective on machine operating parameters (Darr, 2012). This can include aftermarket third party outfitters, parent company research and development, and scientific research conducted through universities. In recent years, several examples of field logistic and machinery performance studies have shown potential for harnessing these datasets from the CAN bus.

Udompetaikul et al. (2011) utilized CAN bus data to monitor fuel consumption of an agricultural tractor when varying tire pressure. Pitla et al. (2014) presented methods for using CAN bus messages to improve machine field efficiency estimates by monitoring different control states on the bus. In another study (Pitla et al., 2016), CAN bus messages were used to estimate load requirements and fuel consumptions for different implements used during field operations. Marx et al. (2015) verified the accuracy of SAE J1939 fuel flow rate CAN bus messages using physically measured values at the Nebraska Tractor Test Laboratory.

When interested parties begin to look into different data acquisition solutions for CAN bus data collection and analysis, the options are almost overwhelming. SAE J1939 CAN bus messages are broadcast in hexadecimal format (frame data) and can be collected using numerous devices. Because there are so many ways to log and convert the same CAN bus information, different logging and analysis methods could affect the outcome of a study focused on logging J1939 data. File size and ease of conversion can both be observed to determine what the best option is for choosing the hardware or software package.

This leads to the question of what differences or errors may exist as a result of the available CAN collection hardware and software packages. Along with that, does the data collected by different packages portray the same information as it was originally broadcast by the CAN network. Data collected simultaneously from the same machine using three different logging methods would provide information on whether or not there are actually differences between CAN bus logging solutions.

#### 2. Objectives

The goal of this study was to provide information regarding data file accumulation sizes and accuracy in reading digital data using multiple CAN bus logging methods for those conducting agricultural field research. This study used three combinations of hardware and software packages to collect the same information. The first objective was to compare accumulated file size and available options for post processing. The second objective was to compare averaged dataset values (i.e., pre- and post-recorded) collected using the three hardware and software CAN bus logging systems.

#### 3. Methods and materials

The first portion of the study describes how to log CAN messages and convert each line into a useable form, such as an engineering unit with a time stamp. This was accomplished using two approaches, including a simple conversion within Microsoft Excel which has the major limitation of file size, and NI DIAdem, which is useful for large files.

Because there are numerous options available for collecting CAN data, this study sought to identify differences between manufacturers of the CAN collection hardware/software, and determine any differences in the type of log files created from these different packages. Vector has the ability to log different file types, including the ASCII hexadecimal message format. NI LabVIEW TDMS files were additional sources used for this study. Data were recorded using three different methods, Vector frame data (logged as an ASCII file), NI frame data and NI averaged hexadecimal data collected from a waveform chart, (both logged as NI TDMS files).

#### 3.1. Test setup

A 270 engine horsepower row crop tractor (John Deere 8270R) was used as the test subject for this study. The test was conducted over a period of approximately 9.5 h on a power take-off (PTO) dynamometer (Figure 1) at the Nebraska Tractor Test Laboratory (NTTL) facility. The parameters of this study were defined by the dynamometer portion (OECD Code 2 section 4.1.1 (OECD, 2012)) of NTTL official test number 2099, which consisted of varying engine speeds and loads throughout the 9.5 h test. During the testing time, data were collected using a Vector CAN logging hardware/software package (CANcaseXL/CANalyzer 8.0, Vector Informatik, Novi, MI) and NI hardware/software packages (NI CompactDAQ 9482/NI LabVIEW, National Instruments, Austin, TX). Machine interface was achieved through the CAN bus to obtain the three separate representations of data (Vector Frame, NI Frame and NI Waveform).



**Figure 1.** 700 horsepower dry gap eddy current dynamometer used by the Nebraska Tractor Test Laboratory.

#### 3.2. Controller area network interface

For this study, the interface with the tractor CAN bus was achieved through the Deutsch HD10-9-1939 J1939 diagnostic connector (Figure 2). The J1939 diagnostic connector is a universal solution for Heavy Trucks and Off-Road equipment including agricultural equipment.

The Deutsch HD10-9-1939 vehicle diagnostic connection pinout allows for not only vehicle CAN bus interface, but also implement bus interface (Figure 3). The ability to interface into the implement bus allows for collection of various signals including ISO 11783 messages.

#### 3.3. Frame data

Frame data logging consists of reading all messages from the network in the respective frequencies as sent by each ECU (National Instruments, 2014). Frame data were collected from various ECUs during the 9.5 h test period logged at frequencies based upon that ECU's transmission frequency. For example, the SAE J1939 message *Electronic Engine Controller 1* (EEC1; PGN F004) transmits signals including Engine Speed (SPN 190) and Actual Percent Engine Torque (SPN 513) at a rate of 100 Hz. The SAE J1939 message *Fuel Economy (Liquid)* (LFE1; PGN FEF2) transmits signals including the engine fuel rate (SPN 183) and Engine Throttle Position (SPN 52) at a rate of 10 Hz. The SAE J1939 message *Engine Temperature 1* (ET1;



**Figure 2.** Deutsch HD10-9-1939 J1939 diagnostic connector: green = CAN low, yellow = CAN high, red = voltage source, black = vehicle ground.

Pin no.	Allocation			
А	ECU_GND			
В	Unswitched Power <sup>a</sup>			
С	Tractor Bus CAN_H			
D	Tractor Bus CAN_L			
E	Not Specified <sup>b</sup>			
F	Not Specified <sup>c</sup>			
G	Not Specified <sup>c</sup>			
Н	Implement Bus CAN_H			
J Implement Bus CAN_L				
<ul> <li><sup>a</sup> A direct connection to positive battery power through a 10A fuse.</li> <li><sup>b</sup>Used for the shield of an SAE J1939 network in an SAE diagnostic connector.</li> <li><sup>c</sup>Used for SAE J1708 network in an SAE diagnostic connector.</li> </ul>				



Figure 3. SAE J1939 vehicle diagnostic connector terminal pinout (as found in the SAE J1939 standards document).



Figure 4. NI CompactDAQ 9862 single port high speed CAN interface (left) and Vector CANcaseXL dual port high speed CAN interface (right).

PGN FEEE) transmits signals including Engine Coolant Temperature (SPN 110) and Engine Fuel Temperature 1 (SPN 174) at a rate of 1 Hz. Because of these different transmission frequencies (as defined by SAE J1939), frame log files can vary a good deal in size. If data from multiple PGNs were desired, a frame data log file with more PGNs having a 100 Hz transmission rate will be larger than a frame data log file with more PGN's having a 1 Hz transmission rate, an issue that will be addressed later in this study.

#### 3.4. Waveform data

Waveform data logging results from a resampling of the frame data into a waveform with a fixed sample rate (National Instruments, 2014). Because frame data log files can become large in size, depending on the number of PGNs desired to be recorded and ECU transmission frequencies, this additional method was used to collect data using this averaged source. A NI LabVIEW application program interface (API) was used to log frame data as a waveform data set. This API logged the averaged waveform data and recorded it at a rate of 2 Hz. This method was used from the desire to gain the same CAN bus parameters but in a smaller log file size.

#### 3.5. SAE J1939 database

A database was created in the Vector management software using a J1939 template and the SAE J1939-71 document (SAE, 2009) so that

known messages could be logged. Within the Vector database created, each message source type was changed from "Null Address" to  $0 \times 0$  in order to be recognized by the NI software/ hardware applications. By using this database, messages and signals could be filtered from each logging source (e.g., NI and Vector data acquisition solutions) for individual collection (rather than collecting every message broadcast on the CAN bus). These individual message frames were collected, stored, and interpreted later using the same database.

#### 3.6. Data collection methods

Data were collected with two different hardware options, a NI CompactDAQ 9862 and a Vector CANcaseXL (Figure 4). Both hardware solutions were connected to the same computer via USB to help reduce potential of timestamp errors. Three different software methods were used; Vector CANalyzer and two separate APIs written in NI LabVIEW software. One set of LabVIEW code was used to collect raw hex (frame) data and another averaged that data into the waveform data. CAN bus data were logged on the same machine at the same time using all three collection methods. By using the *filter* function in Vector CANalyzer, ten signals were logged at transmission frequencies dictated by each ECU (Table 1).

All signal PGN and SPN information can be found in SAE J1939-71 standard (SAE, 2009). The same signals were logged using the NI LabVIEW Frame data API at identical frequencies. Five of those signals including Engine Speed (100 Hz averaged), Actual Percent

PGN	EEC1 (F004)	Fuel economy (FEF2)	EFL_P1 (FEEF)	FD1 (FEBD)	ET1 (FEEE)	AMB (FEF5)
SPN	Engine speed (190) Actual percent engine torque (512	Fuel rate (183) !)	Engine oil pressure (100)	Fan speed (1639) (174)	Engine coolant temp (110) Engine fuel temperature temperature (172) Engine oil temperature (175)	Ambient air temperature (171) Engine air intake
Frequency (Hz)	100	10	2	1	1	1

Table 1. PGN and SPN information for the 10 messages logged.

Table 2. PGN F004 engine speed values calculated by using Equations (1) and (2) with resolution (0.125 RPM bit<sup>-1</sup>) found in SAE J1939 document.

Time stamp	PGN	Bytes	-	-	-	D4	D5	-	-	-	D4 Dec	D5 Dec	Total Dec	Engine speed (RPM)
0.01096	CF00400x	8	FE	FF	94	2C	29	FF	FF	FF	44	41	10,540	1317.5
0.021415	CF00400x	8	FE	FF	93	26	29	FF	FF	FF	38	41	10,534	1316.75
0.031039	CF00400x	8	FE	FF	93	22	29	FF	FF	FF	34	41	10,530	1316.25
0.041613	CF00400x	8	FE	FF	94	20	29	FF	FF	FF	32	41	10,528	1316
0.051034	CF00400x	8	FE	FF	94	26	29	FF	FF	FF	38	41	10,534	1316.75

Engine Torque (100 Hz averaged), Fuel Rate (10 Hz averaged), Fan Speed (1 Hz) and Fuel Temp (1 Hz) were logged with the NI Lab-VIEW Waveform API.

After the data were collected, Microsoft Excel and NI DIAdem *Bus Log Converter* were used to convert the collected frame data into engineering units. DIAdem was used to synchronize the data from the three sources. Collected data were then imported into MS Excel in 30 s or 60 s time increments (depending on frequency of collected data), from incremental times throughout the 9.5 h overall test run of the machine used for this study. Data for these comparisons included both steady state and transient load test conditions.

#### 3.7. Microsoft Excel Hex to engineering unit conversion

Vector J1939 data files in the ASCII format were converted using Microsoft Excel Hex to Decimal functions according to the SAE 1939 standard. A filter was applied to the PGN column (e.g. if only the engine speed was desired to be converted, a filter could be used to only select Electronic Engine Controller 1 PGN: F004 in the PGN column (Table 2)). After filtering out the desired PGN, data values were seen following the PGN and message data length (e.g. 8 bytes was the length of the F004 message). By using the SAE J1939 Vehicle Application Layer document (SAE, 2009) the Engine Speed SPN 190 was found to start at the fourth byte and have a length of two bytes, and offset of zero and a resolution of 0.125 RPM bit 1. By using the Hex2Dec function in MS Excel, the data bytes for SPN 190 were converted to a decimal value. After converting to decimal format, a total decimal value was calculated using Equation (1) as the original values were in hexadecimal format. After calculating the total decimal, the resolution for the specified engine speed SPN (0.125 RPM bit\_1) was used to convert the total decimal to the engineering unit using Equation (2). Example results are shown in Table 2.

$$256^0 * D_4 + 256^1 * D_5 = Total Decimal$$
 (1)

This procedure is applicable to any message with a database such as the SAE J1939 Vehicle Application Layer document (SAE, 2009). After performing this calculation, an available time stamp and a message value exist in a useable engineering unit. Note the timestamp for this message, which represents a 100 Hz frequency data set (Table 2). The same would occur for other messages depending on ECU logging rate (e.g. Fuel Rate was logged at 10 Hz).

## 3.8. National Instruments DIAdem Hex to engineering units conversion

NI DIAdem was a tool used for viewing, sorting and analyzing large data sets. For this study, DIAdem *Bus Log Converter* function was used because of its ability to easily convert CAN hex data into engineering units. This was accomplished by choosing the correct file type within the *Bus Log Converter* (e.g., NI-XNET, Vector ASCII, Vector BLF) then selecting a database to use for conversion. For this study a database similar to the standard Vector J1939 database was used, but with fewer messages. After using the database in the *Bus Log Converter* a log file was created and then imported into DIAdem for viewing and analysis. This resulted in an individual time stamp for each frame along with each line of hex data produced from that ECU and converted into engineering units. DIAdem created an individual time stamp for each ECU because they were logged at different rates as explained in the previous Frame Data section.

#### 3.9. Frame data synchronization

Frame data from NI and Vector were able to be correlated directly. After converting the NI and Vector frame data into engineering units, the two data sets had to be synchronized because they were started at slightly different times due to operation of the two separate user interfaces. Figs. 5 and 6 show the fuel rate from both sets of frame data before and after synchronization, respectively. The data were synchronized by adjusting the time stamp at the beginning of one set of data for the test period within NI DIAdem.

The same procedure was used to compare other signals as well. For this study three data sets were used for comparison, Engine Speed (100 Hz), Fuel rate (10 Hz), and Fuel Temperature (1 Hz). This gave a subjective representation of a variety of CAN Frame data sets to verify if there was a significant difference between these frame data sets.

#### 3.10. Frame data re-sample/average

To synchronize an average 1 Hz waveform dataset with the frame data, frame data were resampled from 100 Hz and 10 Hz, respectively, to 1 Hz, depending on the ECU (Figure 7) and then aligned with the waveform data in a similar method to the frame to frame data comparison.



Figure 5. Fuel rate (L h<sup>-1</sup>) frame data converted to engineering units from both Vector and NI before time synchronization.



Figure 6. Fuel rate (L h<sup>-1</sup>) frame data from Vector and NI after time synchronization.



Figure 7. J1939 engine speed frame data (RPM) averaged from 100 Hz to 1 Hz.

#### 3.11. Analysis

To determine error between data types, a dynamometer test was conducted over a period of 9.5 h. Frame data and waveform data were synchronized as previously detailed. The 9.5 h test length allowed for enough time to show that if excessive differences were detected, the possibility of an underlying frequency or pattern might also be found. For the 100 Hz dataset (Frame Engine Speed), 35 sets of 30 s data were exported. Out of the 9.5 h test, the first of these 30 s data sets was exported at the beginning of the test (where the two frame data sets were synchronized) and another set thereafter every 15 min, providing the 35 sets of 30 s data. For the 10 Hz data (Fuel Rate), five datasets were exported starting at the frame data synchronization and then every two hours afterwards from the 9.5 h test data set. A 1 Hz data (Engine Fuel Temp) also had five dataset sets exported at an increment of 2 h from the 9.5 h test data set similar to the Fuel Rate export.

Waveform data were originally collected via the LabVIEW API at a 2 Hz rate. Because frame data were resampled to a rate of 1 Hz for an additional study, the 2 Hz waveform signals were also resampled to 1 Hz for comparison with the 1 Hz Frame data. To compare waveform to frame data, 19 sets of 60 s engine speed data were exported from the 9.5 h test data set at increments of 30 min. Like the frame data comparison, this gave a detailed depiction of the actual difference between the frame data logged and waveform data logged.

For each of the exported datasets, percent differences were calculated using Microsoft Excel (Eqs. (3) and (4)). These calculated percent differences provided a subjective indication of the differences between the logging sources. Because both values were measured, and there was no expected (or correct) value, thus the vector data was chosen as the expected value for Equation (3).

$$\text{\%Difference} = \frac{|\text{Vector (or Waveform) data - NI Frame data|} * 100 (3)}{\binom{\text{Vector (or Waveform) data + NI Frame data}}{2}}$$

Difference = Vector (or Waveform) data – NI Frame data (4)

#### 4. Results

#### 4.1. Accumulated file sizes

For this study, data were logged from the John Deere 8270R over a 34,328 s (approx. 9.5 h) period of time. As stated in the methods, the two frame data sets logged identical signals, however the waveform

Tab	le 3.	Log	file	sizes	as	record	led	from	their	sources.	

Source	Log file type	Size (kB)	Time (s)
NI waveform	TDMS	26,702	34349.50
NI frame	TDMS	1,208,869	34327.81
Vector frame	ASC II	443,501	34322.19

data set only logged five of those signals. Table 4 summarizes the file sizes as logged during the 9.5 h test from each logging method. The file types are also shown in Table 4, however it is noteworthy to mention that even though TDMS files could be opened with MS Excel, the NI Frame file could not be opened in Excel because of its size. Based on the difference in file sizes (Table 3), there were advantages to using the Vector Frame collection method. This method generated smaller data files of actual hex data (compared to the NI Frame) whereas using the NI Waveform collection method created much smaller overall file sizes.

However, because more frame signals were logged than waveform signals, a breakdown of the files into signals and samples per signal was performed to show file size by samples. The log files were broken down into their respective signals along with the frequency in which each signal was logged. Only the NI Frame and NI Waveform set were shown because the NI Frame and Vector Frame log file samples were similar. Table 4 shows these two log files broken down into the elements of signals along with signal frequency to portray the composition of each file. From Table 4, it was clear that the Torque and Engine Speed signal are the majority of the log file size for the NI Frame data set. By using an API that has the ability to average the frame data, the file size for those signals in particular was greatly reduced.

#### 4.2. Hex data to engineering units

Two methods of J1939 hexadecimal frame data conversion to engineering units were attempted during this study. Although Microsoft Excel had the built in feature of *HEX2DEC*, it required more time to perform conversions. To perform conversions the use of a database with SPN location, length, offset and resolution was required. Since only one signal could be converted at a time, Excel was somewhat cumbersome for converting hexadecimal frame data to engineering units. Another major limitation was the file size that could be loaded into MS Excel. Excel only accepts 1,048,576 rows of data (Microsoft, 2014).

The NI DIAdem *Bus Log Converter* performed this operation more quickly, and only required the database used for logging in order to convert. The additional benefits of DIAdem were the abilities to further manipulate and analyze the data.

#### 4.3. Difference between logging methods

Four different data sets were analyzed to find the percent difference between the three methods of J1939 data logging as outlined in the methods section. An average of the percent difference was calculated for the each of the comparisons to show an overall result of the logged differences found throughout the 9.5 h test (Table 5).

Table 4. Total samples as logged from NI frame and waveform logging sources with a breakdown of each signal that was logged along with the number of samples for each signal.

Signals	NI frame frequency (Hz)	Number of samples	Waveform frequency (Hz)	Number of samples	
Torque	100	3,432,781	2	68,699	
Engine speed	100	3,432,781	2	68,699	
Fuel rate	10	343,278	2	68,699	
Oil pressure	2	68,655			
Fan speed	1	34,328	2	68,699	
Coolant temp	1	34,328			
Fuel temp	1	34,328	2	68,699	
Oil temp	1	34,328			
Ambient air temp	1	34,328			
Engine air intake temp	1	34,328			
Total samples		7,483,463		343,495	

Table 5. Averaged differences and averaged percent differences as found for each of the comparisons.

	100 Hz frame data	10 Hz frame data	1 Hz frame data	Waveform vs frame data	
Difference	–0.00003 (rpm)	0 (L h <sup>-1</sup> )	0(°C)	-0.00041 (rpm)	
Mean difference	0.03959 (%)	0 (%)	0 (%)	0.00643 (%)	



Figure 8. NI frame vs Vector frame mean difference of engine speed over the 9.5 h test.



Figure 9. NI frame vs vector frame mean percent difference of engine speed over the 9.5 h test.



Figure 10. Frame vs waveform mean difference of engine speed over the 9.5 h test.



Figure 11. Frame vs waveform mean percent difference of engine speed over the 9.5 h test.

NI Frame vs Vector Frame data sets were compared. Because these two files logged the same messages in the same format (hexadecimal), three of the different signal frequencies were compared, Engine Speed (100 Hz), Fuel Rate (10 Hz), and Fuel Temperature (1 Hz). Of those three signal frequencies, only the Engine Speed data (Figs. 8 and 9) produced a measurable difference and percent difference over the test time. After synchronization of both the fuel rate and fuel temperature frame datasets, percent differences were zero at every point of collection over the 9.5 h test.

The second comparison sought to determine the error between a frame data set and the NI Waveform dataset. For this analysis, the NI Frame Engine Speed data were compared to the NI Waveform Engine Speed data over the 9.5 h test. Figs. 10 and 11 show the resulting difference and mean percent difference, respectively for the 19 sets of 60 s data exported and analyzed.

#### 5. Conclusions

With regard to conversion of J1939 hex messages to engineering units, while numerous options exist, each method should be considered depending on the end use of the data. Although MS Excel was a bit cumbersome and took longer than NI DIAdem to perform conversions, MS Excel was significantly less expensive and available for use on a variety of operating platforms (e.g., Windows, Mac, or MS Office for Android applications). This was the one advantage that highly outweighed the quick performance of NI DIAdem.

In comparing the NI and Vector frame data, the only cause for the difference indicated between the two data sets was attributed to either hardware jitter (or delay variations (Nolte et al., 2002)), processor timing, or other sources unseen by the user. Differences between datasets were eliminated by analyzing data immediately after synchronization. Rather than synchronizing data once and then comparing throughout a long data set, if synchronization was done before a point where two small sets of data were desired for comparison, the resulting difference was zero. This was only discovered through trials of various synchronization points and although cumbersome, this would eliminate any difference. But again, with the percent difference as low as it was throughout the 9.5 h data set (<0.07%) it is unlikely that the error would exceed any criteria for scientific field data analysis.

Because field research data may be gathered for long periods on equipment running during a working day (as opposed to a test stand), and the equipment could run for weeks on end, corresponding log file sizes become an important factor in logging methodology. If a compact logging device that allows for only small file sizes were available, the ability to log for multiple days or weeks could greatly outweigh the higher resolutions of actual frame data. Even though half of the parameters were logged with waveform collection methods, resulting file sizes were only 6% of the smallest of the two frame datasets. Although there was some difference shown between the various types of J1939 data collection, for most practical purposes in the agriculture industry, this percent difference is so minimal it would not adversely impact the outcomes of studies using any of these logging sources. This may include scientific study, or manufacturers desiring further study on CAN bus applications.

#### References

- Darr, M., 2012. CAN bus technology enables advanced machinery management. Resour.: Eng. Technol. Sustain. World 19, 510–511.
- Marx, S.E., Luck, J.D., Hoy, R.M., Pitla, S.K., Darr, M.J., Blankenship, E., 2015. Validation of machine CAN bus J1939 fuel rate accuracy using Nebraska Tractor Test Laboratory fuel rate data. Comput. Electron. Agric. 118, 179–185.
- Microsoft, 2014. Excel Specifications and Limits Available at: https:// support. office.com/en-nz/article/Excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3

- National Instruments, 2014. NI X-NET Hardware and Software Manual. Nolte, T., Hansson, H., Norstrom, C., 2002. Minimizing CAN responsetime jitter by message manipulation. In: Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE. IEEE, pp. 197–206.
- OECD, 2012. OECD Standard Code for the Official Testing of Agricultural and Forestry Tractor Performance.
- Pitla, S.K., Lin, N., Shearer, S.A., Luck, J.D., 2014. Use of Controller Area Network (CAN) data to determine field efficiencies of agricultural machinery. Appl. Eng. Agric. 30 (6), 829–839.
- Pitla, S.K., Luck, J.D., Lin, N., Werner, J.P., Shearer, S.A., 2016. In-field fuel use and load states of agricultural machinery. Comput. Electron. Agric. 121, 290–300.
- SAE, 2009. SAE J1939-71. Vehicle Application Layer.
- Stone, M., Benneweis, R., Van Bergeijk, J., 2008. Evolution of electronics for mobile agricultural equipment. Trans. ASABE 51 (2), 385–390.
- Udompetaikul, V., Upadhyaya, S., Vannucci, B., 2011. The effect of tire inflation pressure on fuel consumption of an agricultural tractor operating on paved roads. Trans. ASABE 54 (1), 25–30.