Spring 5-2015

Controller Area Network (CAN) Bus J1939 Data Acquisition Methods and Parameter Accuracy Assessment Using Nebraska Tractor Test Laboratory Data

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

CONTROLLER AREA NETWORK (CAN) BUS J1939 DATA ACQUISITION METHODS AND PARAMETER ACCURACY ASSESSMENT USING NEBRASKA TRACTOR TEST LABORATORY DATA

by

Samuel E. Marx

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Agricultural and Biological Systems Engineering

Under the Supervision of Professor Joe D. Luck

Lincoln, Nebraska

May, 2015

CONTROLLER AREA NETWORK (CAN) BUS J1939 DATA ACQUISITION METHODS AND PARAMETER ACCURACY ASSESSMENT USING NEBRASKA TRACTOR TEST LABORATORY DATA

Samuel E. Marx, M.S.

University of Nebraska, 2015

Adviser: Joe D. Luck

Electronics have become a standard in agricultural equipment and the trend of "smarter" equipment is on the rise. To have "smarter" equipment, a working knowledge of the accuracy of the data being transmitted throughout that equipment is needed. The controller area network (CAN) bus is the current interface to machine operation data transmitted between electronic control units (ECUs).

Because CAN bus has been around for nearly thirty years, numerous devices have been created for interfacing with the bus. Choosing a device can be a challenge, especially without knowing if there are differences between the logging methods for true data representation. By logging simultaneously with three different methods, data was analyzed to determine if excessive error existed between logged datasets.

Additionally because many messages are calculated (e.g., not measured directly), determining the accuracy of those messages can be important for management decisions or research studies. One calculated CAN bus message that holds a great deal of value is the engine fuel rate, and because it is calculated rather than measured, excessive error may exist in the CAN bus value. A comparison between the calculated CAN bus fuel rate message and a physically measured fuel rate provided information on the message

accuracy. The Nebraska Tractor Test Laboratory (NTTL) has a certified fuel rate measuring system capable of ±0.5% accuracy (OECD, 2012; Wold, et al., 2015). Results showed that error between logging methods was quite low, however file size was an issue with some of the logging methods. Waveform file logging required only 6% memory space compared to the frame logging methods. Fuel rate as recorded from the CAN bus resulted in a ±5% error from physically measured fuel rates. Error for higher fuel rates within the torque curve were closer to ±1%. These results indicated that the fuel rate given by the CAN bus can indeed be used for management or research purposes.

Dedication

I would like to dedicate this book to all of the people who have supported and encouraged me throughout my life. I can guarantee that I wouldn’t be where I am now if was not for them.

Acknowledgements

First and foremost, I must thank God for providing me with daily motivation to complete this project. Also a great deal of thanks to my wife for her constant support.

I would also like to thank the Nebraska Tractor Test Laboratory, in particular the engineers of NTTL, Rodney Rohrer, Justin Geyer, Doug Triplett, and Brent Sampson. John Deere for allowing CAN bus collection during NTTL tests. Dustin Dam for aid in National Instruments setup and programming. Dr. Matt Darr at Iowa State University for aid in learning more about CAN bus at conferences as well as an arranged session at ISU for CAN training. Dr. Erin Blankenship for assistance with the statistical analysis and SAS programming. Finally thanks to Julie Thompson for always being around to help with anything needed.

Table of Contents

Figures

Tables

Equations

Chapter One: Introduction

Electronics in Agriculture

The use of electronic equipment in agricultural field machinery can be traced back to the 1960's with the introduction of a seed population planter monitor by the DICKEY-John Corporation (Stone, Benneweis, & Van Bergeijk, 2008). This was only the beginning of what would become an evolutionary step for agricultural field equipment. As time went on, additional manufacturers added electronic capability to agricultural field equipment. From planter monitors to rate control systems, grain mass flow and moisture sensors, all the way to more user friendly, manufacturer crossable advanced interfaces of total machine operating parameters. Because additional electronic applications were being integrated into agricultural field equipment, an available communication system was needed. With a multi-master serial communication protocol (controller area network (CAN) bus 2.0) already available, the Society of Automotive Engineering (SAE) group began work on a higher layer protocol to use the CAN bus 2.0 layer. This protocol (SAE J1939) was able to use a predefined message set to operate and diagnose machine operating parameters (Voss, 2008). Because this protocol was focused towards heavy on- and off-road trucks, which share many commonalities with agricultural field tractors, the American Society of Agricultural Engineers (ASAE now ASABE) worked jointly with the SAE to create a protocol standard, ASAE IET 353/1 and SAE ORMTC/SC32, which integrated the International Standard Organization (ISO) working group ISO/TC23/SC19/WG1, standard with the SAE J1939 protocol into the agricultural and forestry industry (Stone & Zachos, 1993). An extension of the SAE J1939 operating parameters were later added for advanced implement communication, which are built upon the ISO 11783 platform (Stone et al., 2008).

CAN Bus Use in Agriculture

Because of the joint efforts of SAE and ASAE, J1939 applications were introduced into the agricultural sector. SAE J1939 allows for control and diagnostics of numerous predefined machine operating parameters including engine control, transmission control, brake control, etc., as well as manufacturer proprietary messages (Stone & Zachos, 1993). Although SAE J1939 created messages that allowed control and interface of many vehicle operating parameters, additional tractor and implement control parameters were still needed in the agricultural sector thus creating the German Institute for Standards (DIN) 9684 protocol (Schueller, 1988). DIN 9684 was later integrated into the ISO 11783 standard (Stone, McKee, Formwalt, & Benneweis, 1999). ISO 11783 uses the same protocol layer as SAE J1939, but is focused toward the agriculture and forestry sector allowing for specifically related information to be available from both the tractor bus as well as an implement bus (Stone et al., 1999). Because of these advancements, and manufacturer adoption of both SAE J1939 and ISO 11783 networks, CAN bus interface has become a viable source for monitoring machine operating parameters.

CAN Bus Data Logging in Agriculture

Before the availability of easily accessible machine operation information, different methods were used to calculate these operating parameters (Colvin, McConnell, & Catus, 1989; Grisso, Perumpral, Vaughan, Roberson, & Pitman, 2014). These methods were often tedious and required a great amount of time and effort. However because so many messages are now readily available via the machine diagnostic connector, this has become a common source for manufacturers as well as research institutes to gain knowledge of machine operation and efficiency (Darr, 2012; Fountas et al., 2015; Pitla, Lin, Shearer, & Luck, 2014; Udompetaikul, Upadhyaya, & Vannucci, 2011). With the advent of CAN based protocols such as SAE J1939 and ISO 11783, there has also been an increase in the amount of data being transferred from the machine for availability to machine operators for management decisions. This has grown from having limited parameters to having a great deal of information that terms such as “big data” and “data mining” have entered the agricultural arena (Ruß & Brenning, 2010; Russo, 2013; van Rijmenam, 2013). Because there is so much data being transferred, some questions can be drawn from the principles behind message use for owner/operator decision management and research goals. One of those questions being, of the vast array of messages being broadcast (some of which are a calculated value rather than a sensor-based measurement) are they accurate enough to use in these managerial decision and research outcomes? Rising petroleum costs have caused fuel use to become increasingly important in management decisions for growers (Trostle, 2010), and because SAE J1939 Engine Fuel Rate is one of the calculated messages being broadcast by the CAN bus, determining the message accuracy is needed. By using different tools for CAN bus data logging, different methods of converting the raw data from the CAN bus to engineering units and then comparing SAE J1939 Engine Fuel Rate to a physically measured value of fuel rate consumption, confidence can be gained regarding its accuracy.

CAN Bus

In order to gain access to the answers desired, a fundamental knowledge of both CAN bus and SAE J1939 is needed.

CAN bus is a serial communications protocol network with the ability to transfer data in speeds up to 1Mbit/s (Bosch, 1991). CAN 2.0B allows for the broadcast of prioritized messages between nodes or Electronic Control Units (ECUs) in a multi-master system (Bosch, 1991). This multi-master system allows for any ECU to broadcast a message as long as the bus is free. CAN bus uses a physical layer comprised of a shielded twisted pair, two wire system; CAN high (CAN_H) and CAN low (CAN_L) (Bell, 2002). CAN is a 5 V DC system where both CAN_H and CAN_L sit idle at 2.5 volts, and when a message is broadcast, CAN_H raises to 5 volts and CAN_L drops to 0 volts (Bell, 2002), producing a 5 volt differential to create a square wave of a certain size and timing location to indicate a message and the pertinent information within that message. An oscilloscope reading from a presentation during the 2013 Agricultural Equipment Technology Conference (Darr, 2013) shows a higher layer CAN message (Figure 1).

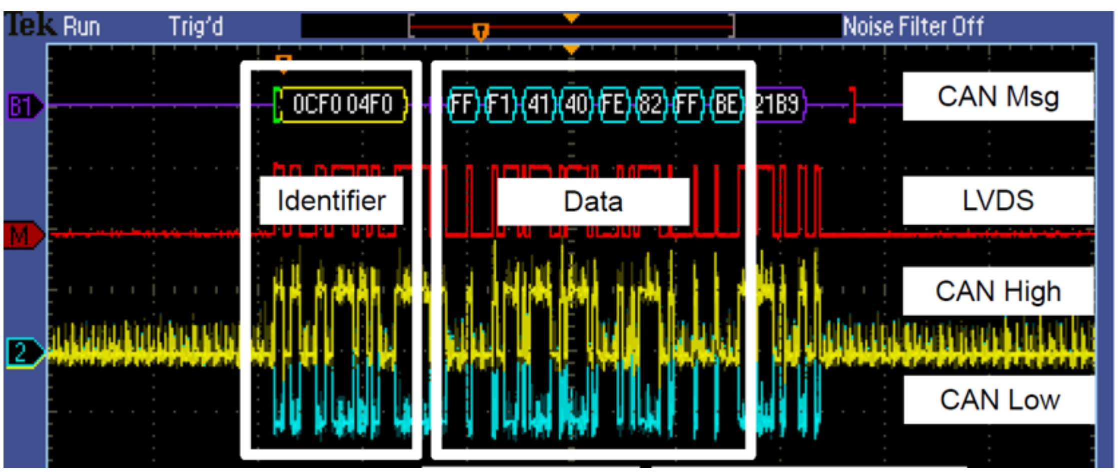


Figure 1: Oscilloscope Image courtesy of Iowa State University shows the 0x00F004 PGN Message with the data following after the identifier

SAE J1939

The Standards of Automotive Engineering began work on a higher layer CAN protocol draft in the early 1990’s. This higher layer protocol is based on the seven layers of the Open Systems Interconnection (OSI) model (Figure 2) (Kvaser, 2014). SAE J1939 utilizes the CAN 2.0B framework to broadcast a 29 bit message identifier (Bell, 2002). SAE J1939 uses a pre-defined message format to allow for multiple manufactures to have similar systems (Voss, 2008). The SAE J1939 message is formed from Parameter Group Number (PGN). SAE J1939 was proposed for use in agricultural equipment in 1993 before the first draft of the document came out (Stone & Zachos, 1993).

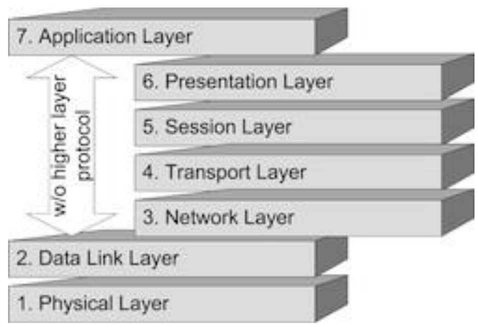


Figure 2: The OSI 7-Layer Reference Model showing higher layer protocol which is implemented in the SAE J1939 standard

SAE J1939 messages are broadcast in hexadecimal format with certain bit timing and byte sizing to indicate the priority of the message, the message identifier, as well as the data within that message. An example of one line of hexadecimal data from PGN F004 (Electronic Engine Controller 1) (Source: SAE J1939 Document) as recorded from a Vector CAN Logging hardware/software package (CANcaseXL, Vector, Novi, MI/ CANalyzer, Vector, Novi, MI) in American Standard Code for Information Interchange (ASCII) shows as:

```
0.012522 1  CF00400x     Rx    d 8 F0 FF 93 8C 1A FF FF FF
```

Figure 1 illustrated the same message (PGN F004) as it is seen by an oscilloscope being broadcast across the bus. The message identifier (F004) is at the beginning of the message to indicate to the other ECU’s on the bus where the message is coming from and the data contained within that message (e.g., F004 contains Actual Percent Engine Torque and Engine Speed messages). A Suspect Parameter Number (SPN) is assigned to specific parameters within each parameter group (Voss, 2008) (e.g., the Engine Speed is defined by SPN 190 within PGN F004).

Engine Fuel Rate

Because energy consumption, specifically Engine Fuel Rate (PGN FEF2; SPN 183), plays a vital role in management decisions, an understanding of the fuel system as implemented by modern mobile agriculture field equipment is needed. Many tractors manufactured today utilize compression ignition diesel engines that use a common rail fuel delivery system. The common rail systems uses a high pressure pump to pressurize the common rail to pressures up to 1800 bar (26,107 psi) which is then available for any of the injection nozzles to deliver to its cylinder (Mudafale, Lutade, & Gosavi, 2013). The fuel rate message is determined by manufacturer specified “fuel mapping” or different timing and pulsation of the solenoid valve injector which is broadcast via the vehicle Electronic Control Unit (ECU) (Goering, Stone, Smith, & Turnquist, 2006). The complexity of the fuel calculation or vehicle fuel map is limited to the ECU operational parameters (Goering et al., 2006) which is the reason that discrepancies may be found between calculated fuel rate and physically measured fuel rate. In the agricultural sector, little work has been published to verify the accuracy of the Engine Fuel Rate message.

Objectives

CAN data within the agricultural industry is very detailed and is becoming increasingly important for use in grower management decisions and research outcomes. The objectives of this study were to 1) recognize some of the different methods available for CAN data logging and provide a comparison among those methods to observe efficiency of logging, file sizes, and conversion methods and 2) determine if there is a difference between the SAE J1939 engine fuel rate and a physically measured fuel rate.

The SAE J1939 Engine Fuel Rate was compared to a sensor-based fuel rate measurement from the Nebraska Tractor Test Laboratory (NTTL), a facility with the ability to measure to fuel consumption with an accuracy of ±0.5%, and deemed accurate for assessing tractor performance by the Organization for Economic Co-operation and Development (OECD). The results of this study have a high impact on the agricultural sector by providing estimates of accuracy in the CAN bus fuel rate which could be used for field operational efficiency and management decisions.

Chapter Two: Comparing Various Hardware/Software Solutions and Conversion Methods for Controller Area Network (CAN) Bus data collection

Abstract

There are various hardware/software solutions available for collecting controller area network (CAN) bus data. The data collected could be skewed based upon different external factors (e.g., hardware/software timing, processor timing, etc.). Because of this, a study was performed to determine if there was a difference in the data collected from these various data acquisition solutions, and to quantify those differences.

Two types of data were observed for this study. The first data type was CAN bus frame data, where a data point is collected for each line of hex data sent from the ECU. One problem with frame data is the resulting large file sizes, therefore a second data type collected was an averaged signal or waveform data. Because of its smaller file size, waveform data is more desirable for long periods of collection. Percent difference was calculated from two sets of frame data, and a set of frame data compared to waveform data.

The resulting difference was less than .0025 RPM for engine speed comparisons, zero for fuel rate and fuel temperature comparisons, and the mean percent difference was less than .08% between the methods of data collection. The error production could have resulted from jitter (or noise) in hardware and processor times, but was not found to grow directly with time. This shows that even though there is error, it is a small enough of an error that for any practical application, data logged by different devices is basically the same.

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 operations 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 to 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; Pitla et al., 2014; Udompetaikul et al., 2011). This can include aftermarket third party outfitters, parent company research and development, and scientific research conducted through universities. However, when any of these groups 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 including, Vector, Kvaser, and National Instruments (NI).

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 exist in some of the available CAN collection hardware and software packages, and along with that, does the data collected by different packages portray the same information? Data collected simultaneously from the same machine using three different logging methods will provide information on whether or not there are actually differences between CAN bus logging solutions. Frame data were collected using both Vector and NI packages as well as an averaged frame data represented as waveform data. All three files were collected simultaneously by different user interfaces from the same source. After synchronization, a simple comparison was performed to determine if there was any difference among the datasets collected. Different sets of data were observed to determine if the difference increased with time, or was related to hardware/software limitations or discrepancies. Two of the file types were CAN frame data, which although had a higher resolution, resulted in extremely large files (over 1 gigabyte for 9.5 hours). Because of the large files, a third method was used to log the same data. Data from the third method, waveform, was compared to a resampled frame data set to determine if the averaged waveform data could be deemed precise enough to use for further studies.

Objectives

This study used three different hardware/software packages to collect similar information. The first objective was to compare accumulated file size and available options for post processing. The second objective was to determine if a difference existed between the data from these different methods of CAN bus logging.

Methods and Materials

The first portion of the study describes how to take CAN messages and convert each line into a useable form, such as an engineering unit with a time stamp. This was accomplished using different methods, 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 shown previously. NI LabVIEW TDMS files were additional sources used for this study. Data was recorded using three different methods, Vector frame data (logged as an ASCII file), NI averaged hexadecimal data collected from a waveform chart, and NI frame data (both logged as NI TDMS files).

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 hours on a power take-off (PTO) dynamometer (Figure 3) 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 hours. 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 cDAQ 9482/NI LabVIEW, National Instruments, Austin, TX). Machine interface was achieved through the controller area network (CAN) bus to obtain the three separate representations of data (Vector Frame, NI Frame and NI Waveform).

Figure 3: 700 Horsepower Dry Gap Eddy Current Dynamometer used by the Nebraska Tractor Test Laboratory

Controller Area Network Interface

For this study, the interface with the tractor's CAN bus was achieved through the Deutsch HD10-9-1939 J1939 diagnostic connector (Figure 4). The J1939 diagnostic connector is a universal solution for Heavy Trucks and Off-Road equipment including agricultural equipment.

Figure 4: Deutsch HD10-9-1939 J1939 Diagnostic Connector: Green= CAN Low, Yellow= CAN High, Red= Voltage source, Black= Vehicle Ground

The Deutsch HD10-9-1939 vehicle diagnostic connection pinout allows for not only vehicle CAN bus interface, but also implement bus interface (Figure 5). The ability to interface into the implement bus allows for collection of various signals including ISO 11783 messages.

Pin no.	Allocation
A	ECU_GND
B	Unswitched Power [a]
C	Tractor Bus CAN_H
D	Tractor Bus CAN_L
E	Not Specified [b]
F	Not Specified [c]
G	Not Specified [c]
H	Implement Bus CAN_H
J	Implement Bus CAN_L

[a] A direct connection to positive battery power through a 10A fuse.
[b] Used for the shield of an SAE J1939 network in an SAE diagnostic connector.
[c] Used for SAE J1708 network in an SAE diagnostic connector

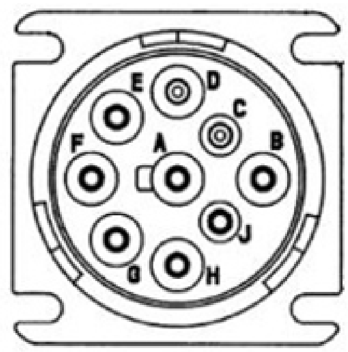


Figure 5: SAE J1939 Vehicle Diagnostic Connector Terminal Pinout (as found in the SAE J1939 Standards document)

Frame Data

Frame data reads in all messages from the network in the respective frequencies as sent by each ECU (National Instruments, 2014).

Frame data were collected from various ECU's during the collection period. These ECUs had different logging frequencies. For example, the Electronic Engine Controller 1 (EEC1; PGN F004) logs signals including Engine Speed (SPN 190) and Actual Percent Engine Torque (SPN 513) at a rate of 100 Hz. The Fuel Economy (Liquid) (LFE1; PGN FEF2) logs signals including the engine fuel rate (SPN 183) and Engine Throttle Position (SPN 52) at a rate of 10 Hz. The Engine Temperature 1 (ET1; PGN FEEE) logs signals including Engine Coolant Temperature (SPN 110) and Engine Fuel Temperature 1 (SPN 174) at a rate of 1 Hz. Because of these different logging frequencies, frame log files can vary in size by a great deal. If data from multiple PGNs were desired, a frame data log file with more PGNs having a 100 Hz log rate will be larger than a frame data log file with more PGN's having a 1 Hz log rate, an issue that will be compared later in this study.

Waveform Data

Waveform data resamples 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, an additional method was used to collect data from an averaged source. NI LabVIEW was used to create an application program interface (API) that logged frame data as a waveform, and then averaged that waveform data and recorded it at a rate of 2Hz. This method was used because of the desire to gain the same data but in a smaller log file size.

SAE J1939 Database

For this study, a vector database was created using a J1939 template and the SAE J1939-71 document(SAE, 2009). By using this database, messages and signals could be filtered 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. A description of how messages were added to the Vector J1939 formatted database can be found in Appendix A.

Data Collection Methods

Data were collected with two different hardware options, a NI CompactDAQ 9862 (Figure 6a) and a Vector CANcaseXL (Figure 6b). 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. The waveform collection method was performed to observe an additional option for collecting the same data, but with a much smaller file size. Data were collected on the same machine at the same time using all three collection methods.

(a) (b)

Figure 6: (a) NI CompactDAQ 9862 Single Port High Speed CAN Interface and (b) Vector CANcaseXL Dual Port High Speed CAN Interface

By using the *filter* function in Vector CANalyzer, 10 signals were logged (Table 1). Two signals (Engine Speed [PGN F004,SPN 190] and Actual Percent Engine Torque [PGN F004, SPN 512]) were logged at 100Hz, one signal (Fuel Rate[PGN FEF2, SPN 183]) was logged at 10Hz, one signal (Engine Oil Pressure [PGN FEEF, SPN 100]) was logged at 2Hz, and six signals (Fan Speed [PGN FEBD, SPN 1639], Engine Coolant Temperature [PGN FEEE, SPN 110], Engine Fuel Temperature 1 [PGN FEEE, SPN

174], Engine Oil Temperature 1 [PGN FEEE, SPN 175], Ambient Air Temperature [PGN FEF5, SPN 171], and Engine Air Inlet Temperature [PGN FEF5, SPN 172]) were logged at 1Hz.

Table 1: PGN and SPN information of the 10 files logged

PGN	F004	FEF2	FEEF	FEBD	FEEE	FEF5
SPN	190	183	100	1639	110	171
	512				174	
					175	
Frequency (Hz)	100	10	2	1	1	1

All signal PGN and SPN information found in SAE J1939-71 standard (SAE, 2009). The same signals were logged using the NI LabVIEW Frame data API at identical frequencies. The NI LabVIEW Waveform API logged only 5 of those signals including Engine Speed (100Hz averaged), Actual Percent Engine Torque (100Hz averaged), Fuel Rate (10Hz averaged), Fan Speed (1Hz) and Fuel Temp (1Hz).

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 Microsoft (MS) Excel in 30 or 60 second time increments (depending on frequency of collected data), from incremental times throughout the 9.5 hour overall test run of the machine used for this study. Data for these comparisons were both steady state as well as transient.

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. The ASCII collected dataset was opened with MS Excel using the *tab delimited* function. After the file was opened, 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)).

Table 2: Filtered Vector ASCII message showing only the PGN F004

Time Stamp	PGN	Bytes	-	-	-	D_4	D_5	-	-	-
0.01096	CF00400x	8	FE	FF	94	2C	29	FF	FF	FF
0.021415	CF00400x	8	FE	FF	93	26	29	FF	FF	FF
0.031039	CF00400x	8	FE	FF	93	22	29	FF	FF	FF
0.041613	CF00400x	8	FE	FF	94	20	29	FF	FF	FF
0.051034	CF00400x	8	FE	FF	94	26	29	FF	FF	FF

After filtering out the desired PGN, data values were seen following the PGN and message data length (e.g. 8 Bytes is the length of the F004 message). By using the SAE J1939 Vehicle Application Layer document (SAE, 2009) the Engine Speed SPN 190 is found to start at the fourth byte and have a length of two bytes, and offset of zero and a resolution of .125 rpm/bit. 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 binary format. After calculating the total decimal, the resolution for the specified engine speed SPN (.125 rpm/bit) was used to convert the total decimal to the engineering unit (Equation 2). Example results are shown in Table 3.

Equation 1: Total Decimal Calculation for Hex to Engineering Unit Conversion

$$256^{0}*D_4+256^{1}*D_5=Total\ Decimal \qquad (1)$$

Equation 2: Using SAE J1939 SPN Resolution for Final Conversion to Engineering Units

$$Total\ Decimal*Resolution=Engine\ Speed \qquad (2)$$

Table 3: PGN F004 Engine Speed values calculated by using equations 1 and 2 with resolution (.125 rpm/bit) 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	10540	1317.5
0.021415	CF00400x	8	FE	FF	93	26	29	FF	FF	FF	38	41	10534	1316.75
0.031039	CF00400x	8	FE	FF	93	22	29	FF	FF	FF	34	41	10530	1316.25
0.041613	CF00400x	8	FE	FF	94	20	29	FF	FF	FF	32	41	10528	1316
0.051034	CF00400x	8	FE	FF	94	26	29	FF	FF	FF	38	41	10534	1316.75

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

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. Use of this tool was accomplished by choosing the correct file type within the *Bus Log Converter* (e.g., NI-XNET, Vector ASCII, Vector BLF, etc.) 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. Within the Vector database created, each message's source type was changed from Null Address to 0x0 to work within NI software/hardware applications. Without changing the source address to 0x0 in the database, NI would not recognize the database message and logging was not possible. After using the database in the *Bus Log Converter* a log file was created and then imported into Diadem for viewing and analysis

(Appendix B(i)). This resulted in an individual time stamp for each ECU, 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 logged at different rates as explained in the previous Frame Data section.

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 from the two separate user interfaces. Figures 7 and 8 show the fuel rate from both sets of frame data before and after synchronization, respectively. The data were synchronized by adjusting the time stamp of one set of data within NI DIAdem.

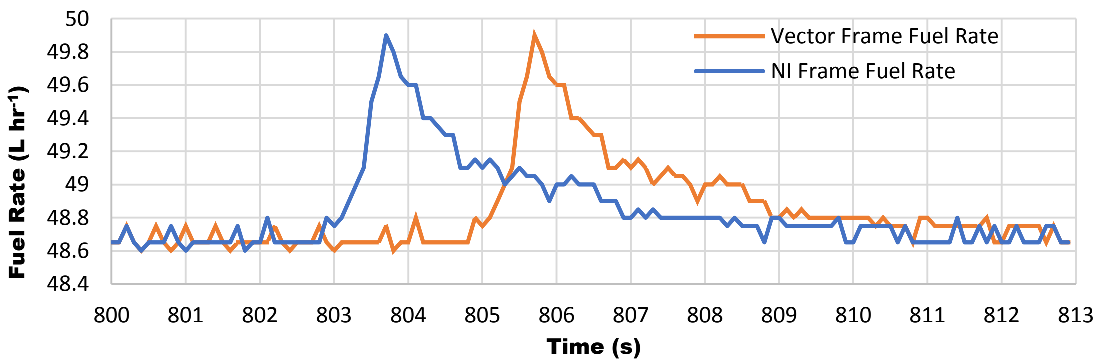


Figure 7: Fuel Rate (L hr^{-1}) frame data converted to engineering units from both Vector and NI before time synchronization

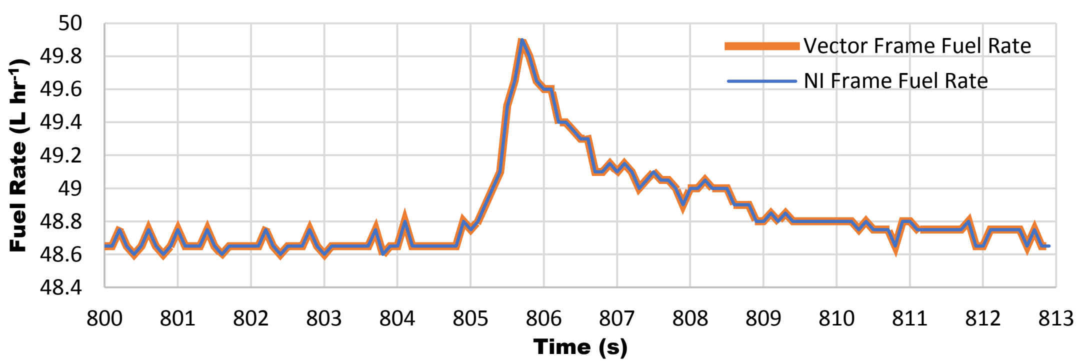


Figure 8: Fuel Rate (L hr^{-1}) frame data from Vector and NI after time synchronization

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

Frame Data Re-Sample/Average

In order to synchronize an average 1Hz waveform data set with the frame data, frame data were resampled from 100Hz/10Hz to 1Hz, depending on the ECU (Figure 9) and then aligned with the waveform data in a similar method to the frame to frame data comparison (Appendix B(iii)). The resample procedure in NI DIAdem averaged the values to either side of the desired time stamp to create a new sample, or an averaged sample.

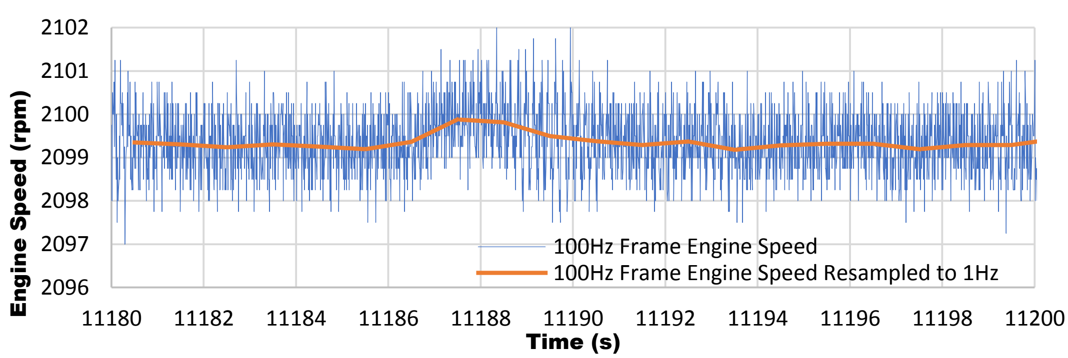


Figure 9: J1939Engine Speed Frame data (RPM) averaged from 100Hz to 1Hz

Analysis

To determine error between the three data types, a dynamometer test was conducted over a period of 9.5 hours. Frame data and waveform data were synchronized as previously detailed. The 9.5 hour 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 100Hz data set (Frame Engine Speed), 35 sets of 30 s data were exported. Out of the 9.5 hour test, the first of these 30 second data sets was exported at the beginning of the test (where the two frame data sets were synchronized) and another set thereafter every 15 minutes, providing the 35 sets of 30 second data. For the 10Hz data (Fuel Rate), five data sets were exported starting at the frame data synchronization and then every two hours afterwards from the 9.5 hour test data set. A 1Hz data (Engine Fuel Temp) also had five datasets exported at an increment of 2 hours from the 9.5 hour test data set similar to the Fuel Rate export. The lower sample rate for the 10Hz and 1Hz data sets was due to the fact that the difference in values were not as significant as the higher frequency data as the results will also indicate.

Waveform data were originally collected via the LabVIEW API at a 2Hz rate due to the program's limitations. Because the API averaged frame data in real time, attempting to average multiple signals in less than 2Hz resulted in program failure. Frame data were resampled to a rate of 1Hz for an additional study, so the 2Hz waveform signals were also resampled to 1Hz for easy comparison with the 1Hz Frame data. To compare waveform to frame data, 19 sets of 60 s engine speed data were exported from the 9.5 hour test data set at increments of 30 minutes. Like the frame data comparison, this gave an accurate depiction of the actual difference between the frame data logged and waveform data logged.

For each of the exported data sets, percent differences were calculated using MS Excel (Equations 3 and 4). These calculated percent differences gave an accurate indication of the true differences between the logging sources.

Equation 3: Percent difference calculation for frame data and waveform data comparisons

$$\%\, Difference = \frac{|(Vector\ (or\ Waveform)data - NI\ Frame\ data)|}{\left(\frac{Vector\ (or\ Waveform)data + NI\ Frame\ data}{2}\right)} * 100 \tag{3}$$

Equation 4: Difference calculation for frame and waveform data comparison

$$Difference = Vector\ (or\ Waveform)data - NI\ Frame\ data \tag{4}$$

Results

File Sizes

For this study, data were logged from the John Deere 8270R over a 34,328 s (approx. 9.5 hour) period of time. As stated in the methods, the two frame data sets logged identical signals, however the waveform data set only logged five of those signals. Table 4 summarizes the file sizes as logged during the 9.5 hour 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 4), 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.

Table 4: Log file sizes as recorded 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

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 in magnitude. Table 5 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 5, 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.

Table 5: 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

Hex Data to Engineering Units

Two methods of J1939 hexadecimal frame data conversion to engineering units were attempted during this study. Although MS 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.

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 differences found throughout the 9.5 hour test (Table 6).

Table 6: Averaged differences and averaged percent differences as found for each of the comparisons

	100Hz Frame Data	10Hz Frame Data	1Hz Frame Data	Waveform vs Frame Data
Difference	-0.00003	0	0	-0.00041
Mean % Difference	0.03959	0	0	0.00643

NI Frame vs Vector Frame data sets were compared first. Because these two files logged the same messages in the same format (hexadecimal), three of the different signal frequencies were compared, Engine Speed (100Hz), Fuel Rate (10Hz), and Fuel Temperature (1Hz). Of those three signal frequencies, only the Engine Speed data (Figures 10 and 11) 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 hour test.

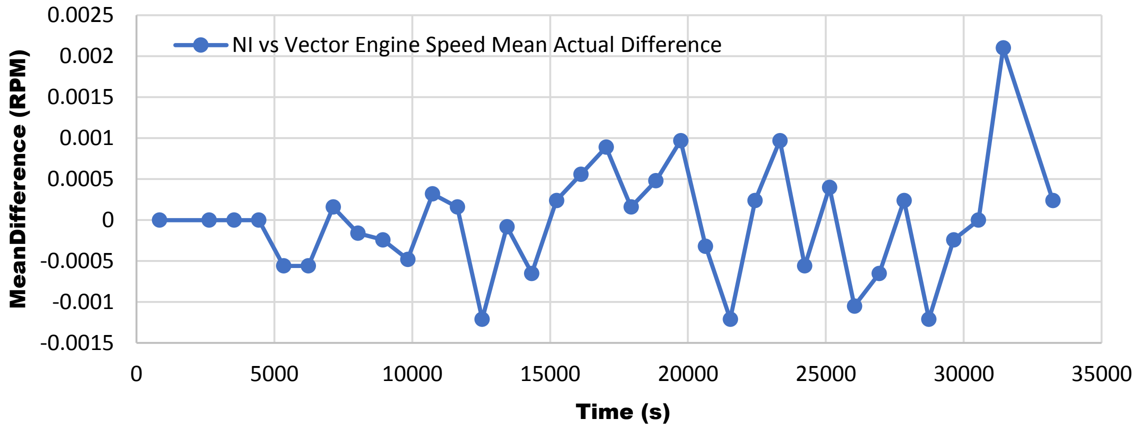


Figure 10: NI Frame vs Vector Frame mean difference of engine speed over the 9.5 hour test

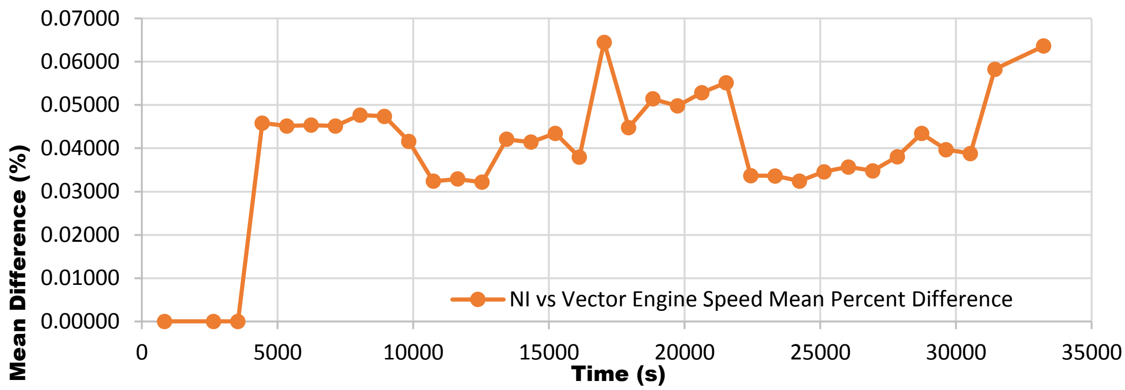


Figure 11: NI Frame vs Vector Frame mean percent difference of engine speed over the 9.5 hour test

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 hour test. Figures 12 and 13 show the resulting difference and mean percent difference, respectively for the 19 sets of 60 s data exported and analyzed.

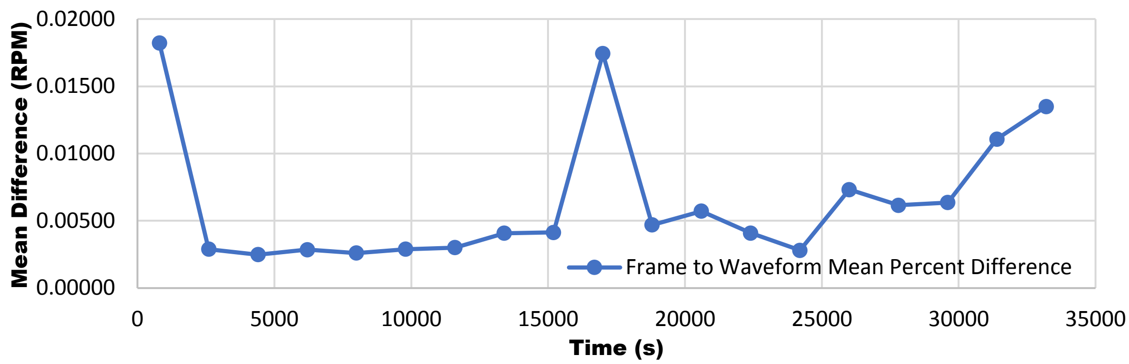


Figure 12: Frame vs Waveform mean difference of engine speed over the 9.5 hour test

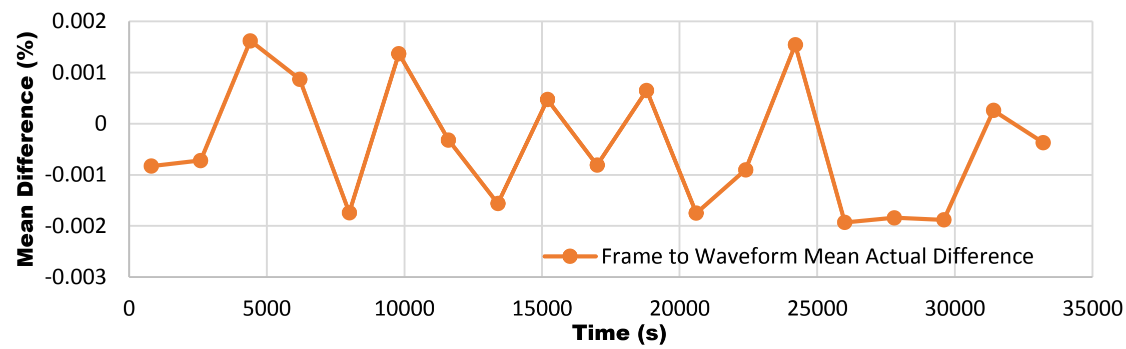


Figure 13: Frame vs Waveform mean percent difference of engine speed over the 9.5 hour test

Synchronization

Synchronization was performed at the earliest available point where there was a clear data transition (e.g., a sharp peak or valley in the two data sets). As seen by the results of the comparisons, for the higher frequency logging it was clear that immediately following synchronization, the resulting difference was zero but then increased as time went on. It was also found that if the data was synchronized immediately before a desired time period, the data would then line up and again have a resulting difference of zero. This could prove useful if a large dataset was available, but only a small portion within that dataset was desired for analysis.

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, Hansson, & Norstrom, 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 hour data set (<.07%) it is unlikely that the error would exceed any criteria for scientific data analysis.

Because research data may be gathered for long periods on equipment running in the field (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 the files where logged with the waveform collection, the waveform data file size was 6% of the smaller of the two frame data sets. 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 would include scientific study, or manufacturers desiring further study on CAN bus applications.

Chapter Three: Validation of machine CAN Bus J1939 fuel rate accuracy using Nebraska Tractor Test Laboratory fuel rate data

Abstract

A pilot study was performed to determine if there were differences between data collected using the machine controller area network (CAN) bus Society of Automotive Engineers (SAE) J1939 standard fuel rate and data collected from a physical measurement system utilized by the Nebraska Tractor Test Laboratory (NTTL). The pilot study concluded that there was a difference between the data (up to a 6.22% error), which indicated a need to perform further studies on this comparison.

The SAE J1939 standard fuel rate message (PGN: FEF2 SPN: 183) utilized by the machine CAN bus has a theoretical value, however little work has been done to verify the accuracy of this value. Because fuel flow rate values reported are rarely measured directly on field equipment using a flow meter, the value is likely estimated based on other operating parameters, (e.g., engine speed, number of cylinders, injector timing and pulsation, etc.). The goal of this study was to compare fuel rate values collected from the CAN bus to the physically measured fuel rate value from tractor performance tests conducted at the Nebraska Tractor Test Laboratory (NTTL). The fuel rate values were collected simultaneously and then synchronized to confirm accuracy of results. The values for comparison where comprised of certain performance test points as described in the Organization for Economic Co-operation and Development (OECD) Code 2. The specific test points consisted of the tractor's engine torque curve, within section 4.1.1,

along with multiple points of varying engine speed and engine power, section 4.1.3.1, (OECD, 2012). The NTTL has a certified fuel rate measuring system with an accuracy of ±0.5% (OECD, 2012; Wold et al., 2015).

Fuel rate, as recorded from the CAN bus, resulted in a ±5% error of actual physically measured fuel rates. Error for higher fuel rates within the torque curve were closer to ±1%. This produced confidence in the ability to use machine data for in field efficiency and/or spatial fuel usage for additional analysis, whether used for research or grower cost analysis with an accurate knowledge of actual fuel consumed during operation.

Introduction

The use of electronics in agricultural field operations began in the 1960's and has progressed to agricultural field machines that utilize electronic control units (ECUs) for full control of engines and almost every other parameter of the machine (Stone et al., 2008). Today, the most common source of data and data transmission on agricultural field machinery is the controller area network (CAN) Bus. The CAN bus protocol was officially introduced by the Robert Bosch GmbH in 1986 in conjunction with the car manufacturer Mercedes Benz (Voss, 2005). Since then, CAN applications have been used throughout numerous industries including light duty passenger automotive, heavy duty on- and off-road automotive, marine, factory and agricultural (Voss, 2005). A higher layer protocol used to manage communication over the bus network and derived from the seven layer Open Systems Interconnection (OSI) model (Kvaser, 2014) based on CAN 2.0B (Voss, 2008) was implemented by the Standards of Automotive Engineering (SAE) as early as 1994. This higher layer was the SAE J1939 protocol which uses predefined parameters in the form of a public database to give manufacturers a guide for ECU programming (Voss, 2008). Application of the CAN protocol standard SAE J1939 was proposed for use in agricultural field machinery in the early 1990's (Stone & Zachos, 1993).

The use of the SAE J1939 public database allows the scientific community to quickly access important machine operating parameters; including machine fuel consumption rate (Darr, 2012; Pitla et al., 2014; Udompetaikul et al., 2011). Because the SAE J1939 data is publicly available, and researchers are using the data to answer questions regarding the efficiency of machines, understanding the accuracy of these messages is important. The SAE J1939 engine fuel rate message is one that is especially important. Because the SAE J1939 fuel rate value is estimated based on different ECU inputs (e.g., number of cylinders, rpm, size and timing of each PWM injection valve, etc.) rather than physically measured, a calculated comparison between J1939 fuel rate messages to a physically measured fuel rate was performed. Physically measured fuel rate data came from the Nebraska Tractor Test Laboratory (NTTL), which is an Organization for Economic Co-operation and Development (OECD) approved test station and has the ability to measure fuel consumption to an accuracy of ±0.5% (Wold et al., 2015). The NTTL utilizes electronic data acquisition to provide physical measurements of fuel consumption on agricultural tractors (Grisso et al., 2012; Ingle, 2011; Kim, Bashford, & Sampson, 2005). By knowing the accuracy of the calculated CAN fuel rate message, information from that message can be used for more confident research and management decisions.

Objectives

The goal of this study was to evaluate the accuracy of fuel flow CAN bus messages for agricultural machinery as this has become one of many readily available data sources for researchers and industry professionals. There are two main objectives to this study, 1) to provide an analysis of the accuracy of the CAN based J1939 fuel rate messages under steady state conditions, and 2) determine if any variation in errors exists across different fuel flow rate ranges for the tractors assessed.

Methods and Materials

Test Setup

Six mid to high horsepower (245-370 engine horsepower) row crop tractors were used for this study. The tractors used were NTTL official tests 2098 through 2103 (8245R, 8270R, 8295R, 8320R, 8370R IVT, and 8345RT). The data collection from both CAN bus and NTTL was done during the Power Take-Off (PTO) portion of the official NTTL tests. The PTO portion of the NTTL tests use a 700 maximum horsepower dry gap Eddy Current Dynamometer (Figure 14) for load variation. During the PTO testing, multiple engine speed and load variations were selected to identify tractor operating outputs. For this study, the engine torque curve (22 RPM ranges from 2100RPM to 1050RPM at full load) and five extra points for fuel consumption characteristics (100% engine speed-80% load, 90% engine speed-80% load, 90% engine speed-40% load, 60% engine speed-60% load, and 60% engine speed-40% load) were used. This created 27 sets of 60 s data for analysis.

Figure 14: 700 HP dry gap Eddy Current Dynamometer used by the NTTL for official testing

CAN Bus Interface

To interface with the vehicle CAN bus, different physical connections are available based upon the type of vehicle being connected to. For this study, a Deutsch HD10-9-1939 J1939 Diagnostic Connector (Figure 15) was used as the physical connection to the tractor CAN bus. CAN_H (Pin C) is indicated by the yellow wire, CAN_L (Pin D) is indicated by the green wire, vehicle voltage source is indicated by the red wire (Pin B), and vehicle ground is indicated by the black wire (Pin A). The J1939 Diagnostic Connector is the standard connection to the CAN bus for agricultural equipment.

Pin no.	Allocation
A	ECU_GND
B	Unswitched Power [a]
C	Tractor Bus CAN_H
D	Tractor Bus CAN_L
E	Not Specified [b]
F	Not Specified [c]
G	Not Specified [c]
H	Implement Bus CAN_H
J	Implement Bus CAN_L
[a] A direct connection to positive battery power through a 10A fuse. [b] Used for the shield of an SAE J1939 network in an SAE diagnostic connector. [c] Used for SAE J1708 network in an SAE diagnostic connector	

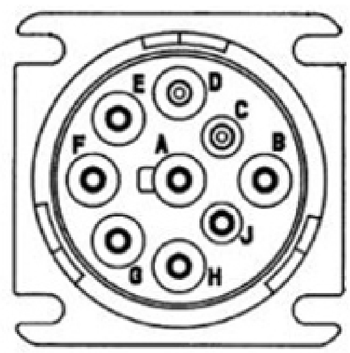


Figure 15: Deutsch HD10-9-1939 J1939 Diagnostic Connector with pinout schematic: Vehicle CAN and Implement CAN (Green= CAN Low, Yellow= CAN High) Red= Voltage source, Black= Vehicle Ground

Signals Logged

Engine Fuel Rate (PGN FEF2, SPN 183), a J1939 message calculated based upon "Command-Fuel-Quantity and verified by the fuel-rail-pressure and fly-wheel feedback" (Walter, 2015) was the primary focus of this study. The calculated fuel rate was compared to the physically measured mass flow rate as recorded by NTTL in National Instruments (NI) technical data management streaming (TDMS) format.

A second message, Engine Speed (PGN F004, SPN 190), was also logged to give an accurate assessment of signal synchronization. Because the engine speed as recorded from both J1939 and NTTL are physically measured by the rotational speed of the same shaft, it was determined that this was an ideal source for synchronizing the two data sets. Based on the shaft rotation speed transitions, as engine speed decreases or increases, both sensors are measuring the same shaft in real-time, therefore transient engine speeds should match. This synchronization was done by taking both data sets, and correcting the time in NI DIAdem on one of the signals to synchronize it with the other. By correcting the J1939 message time, all messages followed the same time stamp of the NTTL messages.

Data Acquisition

CAN bus J1939 data was recorded with a NI Compact Data Acquisition (cDAQ) module (9482 single port high speed module, National Instruments, Austin, TX). A database based on the Vector database (Vector j1939.dbc, Vector Informatik, Novi, MI) was created to allow the NI API to interpret and log the CAN bus messages. The NI API acquired data from the cDAQ 9482 module and utilized the *CAN Input Stream to TDMS Logfile* to log the signals specified from the database. The SAE J1939 signals logged through this program included Engine Speed, Actual Percent Engine Torque, Fuel Rate, Fan Speed, Fuel Inlet Temperature, Coolant Temperature, Oil Temperature, Ambient Air Temperature, Engine Intake Air Temperature and Oil Pressure. Although ten signals were recorded, only two of these signals were used for comparison with NTTL signals, Engine Speed and Fuel Rate. The TDMS file format was then able to be used with NI DIAdem software.

The NTTL used various analog and digital sensors (Figure 16) along with various NI cDAQ modules where signals were processed with a LabVIEW program created by NTTL staff. The NI LabVIEW program logged signals at a 1000 Hz frequency, then averaged each signal over a one second time period to give a 1 Hz value which was then recorded as raw data. The engine speed was recorded by a Banner D12 Fiber Optic Digital Pulse Sensor connected to a NI cDAQ 9435 module.

Figure 16: NTTL Analog and Digital signal collection point for Dynamometer testing

The fuel rate was collected using a Micro Motion mass flow sensor (Micro Motion model CMFS015M324J2BMEZZZ, Emerson Process Management, Boulder, CO), which measured mass flow using the Coriolis method, along with a PUCK800 transmitter connected to a NI cDAQ 9203 module. According to the manufacturers specifications the Micro Motion sensor has a maximum flow rate of 5.4 kg min^{-1} (324 kg hr^{-1}/714.298 lb. hr^{-1}). Figure 17 shows the error based on flow percentage of the Micro Motion flow meter according to the manufacturers specifications provided with the flow meter (Appendix E). Fuel rates for this study ranged between 15.42 kg hr^{-1} and 62.14 kg hr^{-1} (4.8-19.2% of Micro Motion maximum flow rate).

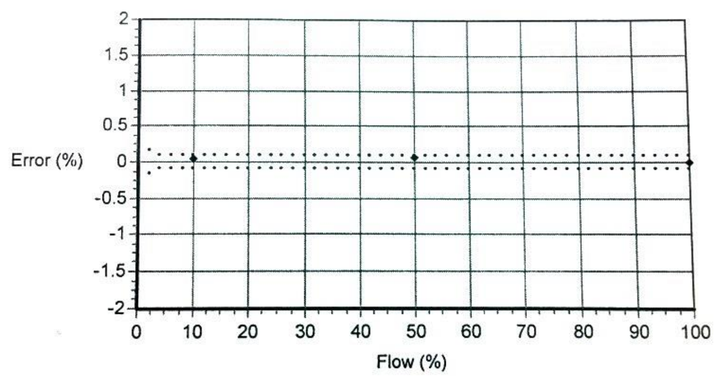


Figure 17: Micro Motion mass flow sensor error based on flow percentage (with 100% being 324 kg hr^{-1})

To confirm that the NTTL fuel measurement setup was accurate with regard to the Micro Motion factory calibrations, results from a previous study provided an assessment of error produced by the NTTL fuel measurement system (Figure 18). The results of this study showed that the NTTL fuel measurement system provided an accuracy of ±0.5% for all flow rates except four points at the lowest tested flow rate (Wold et al., 2015). Figure 18 indicates that most samples had an error within ±0.25% over the majority of the fuel rate ranges.

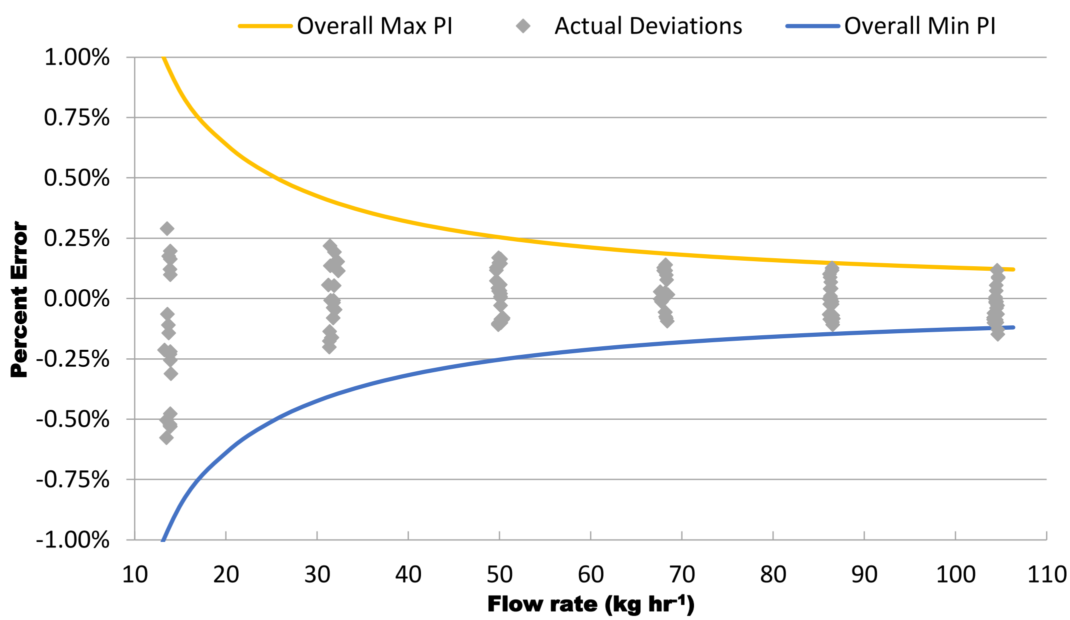


Figure 18: Results of a NTTL fuel calibration with prediction intervals (Wold et al., 2015)

The official OECD tests at the NTTL used red dyed farm grade #2 diesel fuel. Each batch of diesel fuel delivered has its specific gravity measured and then corrected to 15^0C (59^0F). The temperature correction allowed for a density measurement to correspond with OECD code 2.7.1 (OECD, 2012) which was used for mass flow rate to volumetric flow rate conversion. The corrected density used for the batch of fuel during this study was 7.036 lb gal^{-1}. This density was then used in equation 5 to convert fuel recorded by the Micro Motion sensor from lb hr^{-1} to L hr^{-1}.

Equation 5: Calculation used for NTTL mass flow rate to volumetric flow rate conversion

$$Fuel\ Rate\left(\frac{L}{hr}\right) = Fuel\ Rate\left(\frac{lb}{hr}\right) * \frac{gal}{7.036\ lb} * \frac{3.785\ L}{gal} \qquad (5)$$

Figures 19 and 20 outline the NTTL fuel measurement system. In the NTTL fuel system, fuel passed from a holding tank to a filter, then through the Micro Motion mass flow sensor. After travelling through the mass flow sensor, the fuel flowed into a Murphy LM305 lubrication level maintainer (a liquid float). From the outlet of the LM305 fuel float, the fuel entered a fuel heater and then entered into the fuel inlet of the machine. The fuel from the return line of the machine (the fuel not used by the tractor's injection system) was passed through a cooler and then flowed back into the LM305 fuel float. Once the machine reached steady state operation, the fuel rate measured represented the fuel from the holding tank required to keep the LM305 fuel float at a level position.

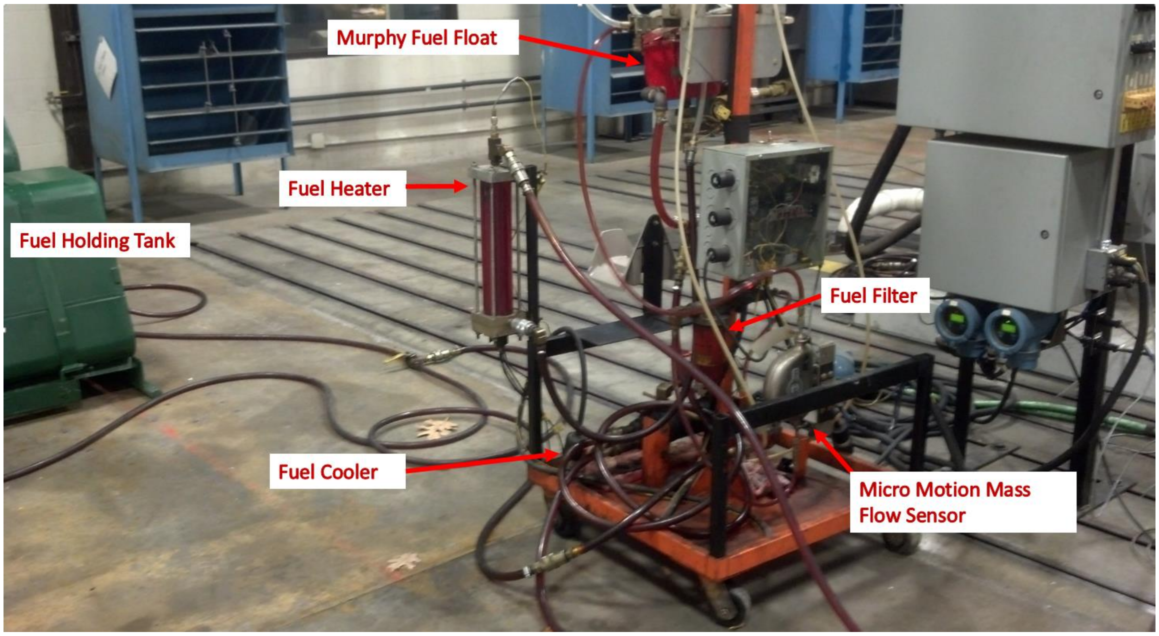


Figure 19: Fuel measurement system used by NTTL

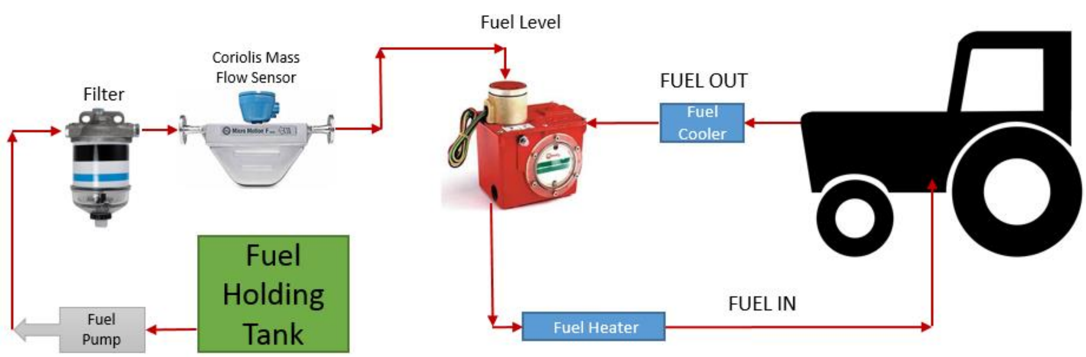


Figure 20: NTTL fuel measurement diagram to show how Micro Motion sensor and Murphey fuel float work together to accurately measure tractor fuel consumption

Signal Comparison

Because J1939 data were logged by NI LabVIEW in a TDMS format, NI DIAdem was used to import the frame hex data and convert it into useable engineering units in preparation for comparison to NTTL data. NI DIAdem has a *Bus Log Converter* function which was used to perform the frame hex to engineering unit conversion. To use the *Bus Log Converter*, the TDMS file logged by the NI X-Net API was selected, along with the

Vector J1939 database created for use with the LabVIEW program as described previously. The *Bus Log Converter* function used the parameters (including data length, resolution, data range, and offset) within the Vector database to convert the frame data to engineering units with a time stamp. This provided the signals logged from the CAN bus in engineering units, but gave them in their original frame transmission frequencies (Engine Speed at 100Hz, Fuel Rate at 10Hz). To compare J1939 signals to NTTL signals, both had to be in similar format and frequency. Because the NTTL data were recorded at a frequency of 1 Hz, the J1939 frame data needed to be converted to 1 Hz to be correctly compared. NI DIAdem was used to resample these signals to a 1 Hz dataset. Within DIAdem, the *Reducing Classification* function was used to resample both the J1939 Engine Speed and Fuel Rate. The *Reducing Classification* function, based on the width of sample (in this case one second) averaged the original data within that width to create a new resampled data set at the desired width. For example, the engine speed was originally 100 Hz and contained 1,000,000 data points, and if a 1 Hz resampled dataset were desired, the original number of data points was divided by the frequency [1,000,000/100=10,000] to get a new dataset at the desired frequency (Figure 21).

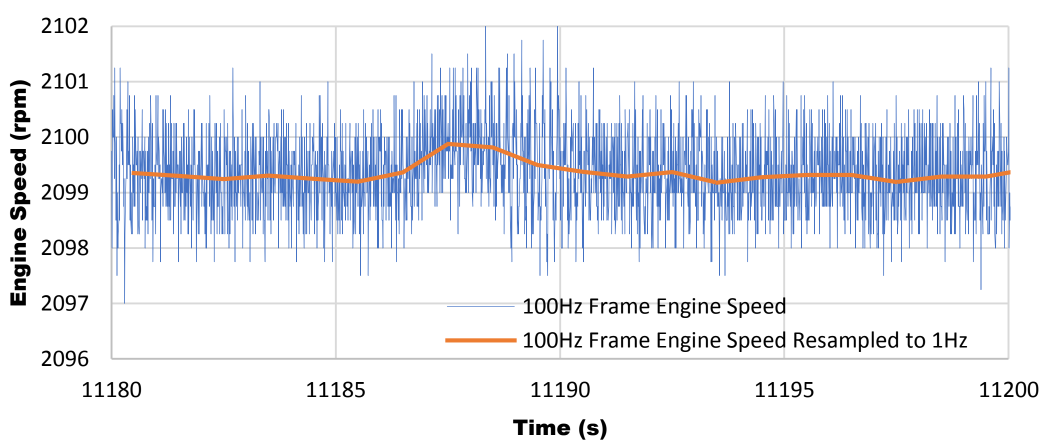


Figure 21: DIAdem *Reducing Classification* function result showing 100Hz J1939 engine speed frame data resampled to 1Hz for comparison to NTTL 1Hz engine speed data

Timing synchronization was accomplished by using the Engine Speed from both datasets because the Engine Speed was logged by both hardware/software suites from the same rotating shaft in real time. The reason that a secondary signal was used (rather than just synchronizing based on fuel rate) was due to the NTTL fuel measurement system method of operation. Because a float was used as the point of reference for fuel demand, there was a lag in transient fuel rates. Results from a previous pilot study (Marx & Luck, 2013) showed that comparison between CAN bus and NTTL fuel rates during transient periods resulted in very high error (up to 6.22%). A regression from a direct comparison of CAN bus fuel rate vs NTTL fuel rate should have an ideal slope of one-to-one. It is clear from Figure 22 that during the pilot study this was not the case.

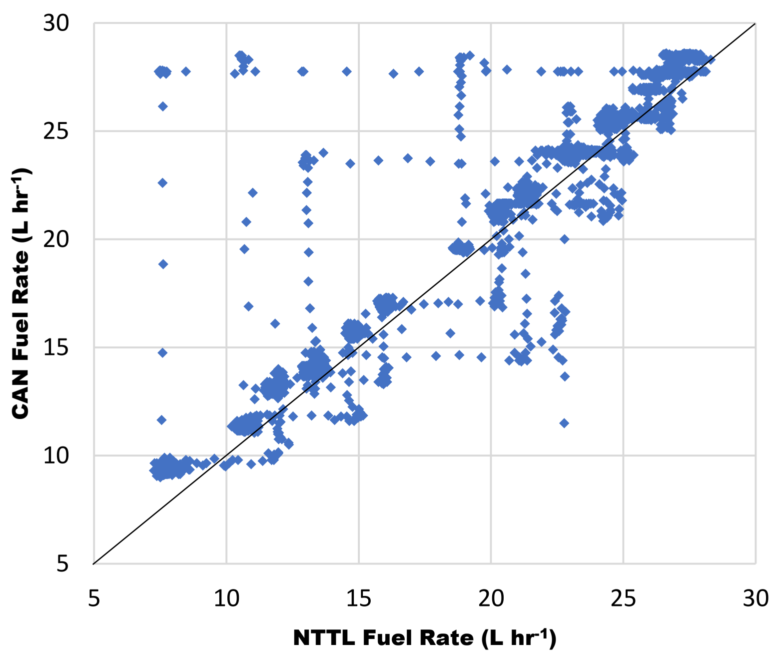


Figure 22: Pilot study results showing that transient fuel rates had outliers resulting in very high overall error

Because of the high error produced from transient fuel rates, synchronizing the two data sets and then truncating and comparing only steady state data was performed to eliminate the discrepancies during the transient fuel rates. To synchronize the datasets, the time stamp for the J1939 messages was shifted to align with the time stamp for the NTTL data. Figure 23a shows the difference in time from the original data sets (because two separate computers were used for data logging, there was no direct correlation between data sets resulting in a 1-5 second difference). Figure 23b shows the two data sets after synchronization.

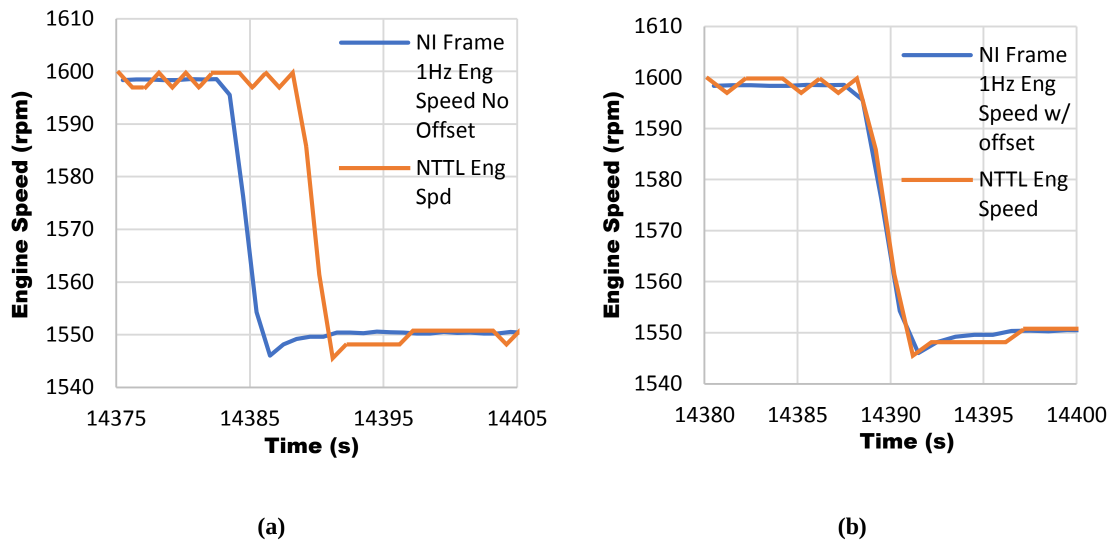


Figure 23: J1939 engine speed and NTTL engine speed shown (a) before time correction and (b) after lining up a transition period to synchronize data sets

By synchronizing the engine speed time stamps, the fuel rate time stamp followed along with the J1939 Engine Speed time stamp to give a direct time correlation to NTTL fuel rate (Figure 24).

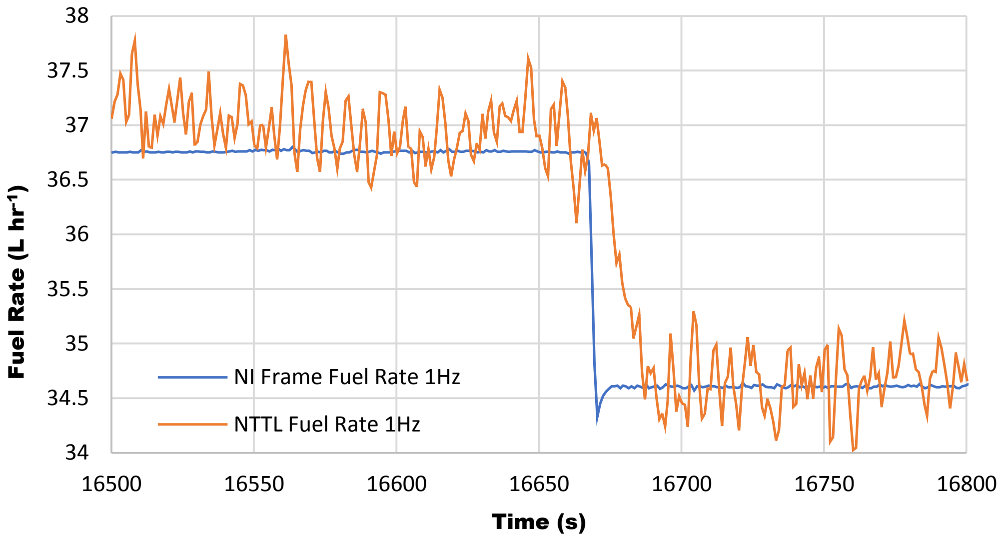


Figure 24: J1939 fuel rate and NTTL fuel rate during a transient period of the test (Note the lag in NTTL data due to fuel measuring system again indicating steady state data only be compared)

After the engine speed and fuel rate data were synchronized, the 27 sets of 60 s truncated steady state data were exported to Microsoft (MS) Excel. Figure 25 illustrates the 27 sets of 60 s fuel rate data as exported into MS Excel for the John Deere 8245R (NTTL Test number 2098), which shows the fuel rates for the torque curve (2100 RPM to 1050 RPM) along with the five additional OECD points.

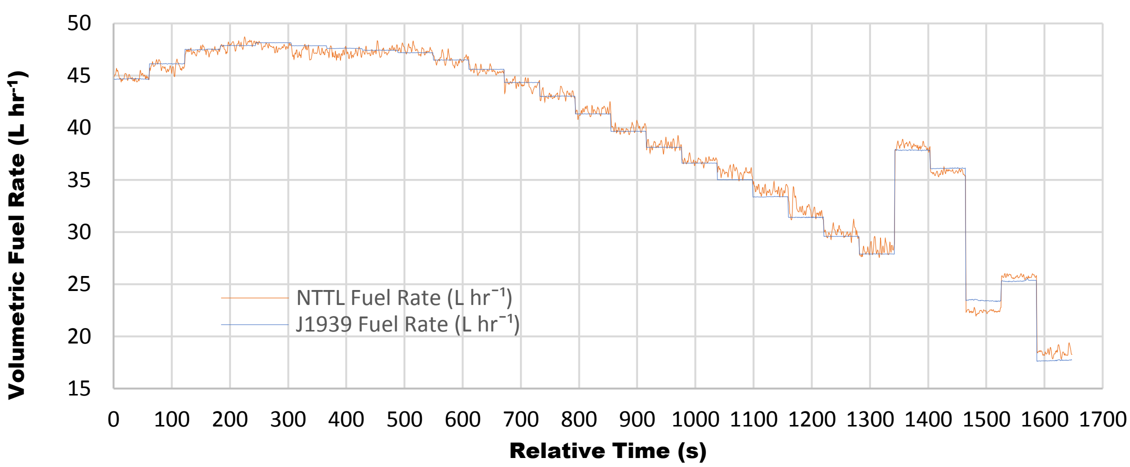


Figure 25: J1939 and NTTL volumetric fuel rate comparison in L hr^{-1} over the torque curve and additional OECD points of one tractor (8245R)

Once the fuel rate and engine speed datasets for all six tractors were exported into MS Excel, an error value was calculated for both the engine speed and fuel rate. To calculate error between NTTL values and J1939 values, a percent error calculation (equation 6) was used. This gave a true error between the two signals, whether positive or negative, at every second for each of the 60 s steady state datasets for the 27 total datasets for each tractor. The average of the percent error values was then calculated for each 60 s dataset and were entered into a master MS Excel program separating values for statistical analysis. This generated two values to run statistical analysis on, engine speed error and fuel rate error.

Equation 6: Percent error calculation for both engine speed and fuel rate with NTTL being set as the theoretical value and the J1939 value as the experimental value

$$\% \, Error = \frac{J1939 - NTTL}{NTTL} * 100 \qquad (6)$$

Statistical Analysis

The calculated error for each machine was based on the data from the torque curve and five additional OECD points, which gave a representation of field operating conditions including varying fuel rate from high to low as well as variations of percent load and percent engine speed. Because each machine has a slightly different fuel consumption rate, these 27 rates were categorized into percentage ranges. To classify the fuel range categories, the highest fuel consumption rate and lowest fuel consumption rate were found for each tractor from the data sets acquired, then based on those high and low flow rates, flow rates from the 27 data sets were entered into percentage ranges from high fuel flow rate to low flow fuel rate (e.g. Tractor 1 had a high flow rate of 48.14 L hr^{-1} and a low flow rate of 17.67 L hr^{-1} giving ranges from 17.67 L hr^{-1} to 48.14 L hr^{-1} in increments of 6.10 L hr^{-1}). The percentage ranges based upon high and low flow of each tractor are 0 to 20% (treatment 5), 20 to 40% (treatment 4), 40 to 60% (treatment 3), 60 to 80% (treatment 2) and 80 to 100% (treatment 1). Table 7 provides the fuel rates (in L hr^{-1}) for each of the treatments for each of the tractors.

Table 7: Machines and respective fuel rates and treatments for statistical analysis. Low fuel rate and high fuel rate is shown for each machine as well as the fuel rates used for percentage calculation for treatment

Tractor	Low Flow Rate (L hr^{-1})	High Flow Rate (L hr^{-1})	Treatment 5 0-20% (L hr^{-1})	Treatment 4 21-40% (L hr^{-1})	Treatment 3 41-60% (L hr^{-1})	Treatment 2 61-80% (L hr^{-1})	Treatment 1 81-100% (L hr^{-1})
1	17.67	48.14	17.67-23.76	23.76-29.86	29.86-35.95	35.95-42.05	42.05-48.14
2	18.88	52.32	18.88-25.57	25.57-32.26	32.26-38.94	38.94-45.63	45.63-52.32
3	20.64	57.09	20.64-27.93	27.93-35.22	35.22-42.51	42.51-49.80	49.80-57.09
4	21.85	62.11	21.85-29.90	29.90-37.95	37.95-46.01	46.01-54.06	54.06-62.11
5	24.33	66.95	24.33-32.85	32.85-41.38	41.38-49.90	49.90-58.42	58.42-66.95
6	26.06	71.99	26.06-35.24	35.24-44.43	44.43-53.62	53.62-62.80	62.80-71.99

A randomized complete block design was used, with the tractor serving as the block. There were a total of six blocks, and the block was treated as a random effect. An analysis of variance (ANOVA), implemented in Statistical Analysis Software (SAS) v9.4 *PROC GLIMMIX*, was used to ascertain whether the responses of engine speed mean percent error and fuel rate mean percent error differed among the treatments (fuel rate percentage ranges). Although an overall average fuel rate error was calculated for each tractor, the five treatment ranges were chosen to provide error analysis of different fuel consumption rates rather than just an average fuel rate error.

Data used for the response variables were compiled based upon the fuel rate treatments from table 7. SAS v9.4 code used for this study can be found in appendix C.

Results

To ensure that the data between the two sources were synchronized, the engine speed measured by both logging systems (CAN bus and NTTL) were compared. Because the speed was measured off of the same shaft simultaneously, the error should be very low if they were properly synchronized. Based on the SAS analysis of the engine speed error, there was no significant differences between the two different measurement methods (P>.05) and with that, the average error (Estimate) throughout the data collection points was small (Table 8).

Table 8: Results of Engine Speed Error from the GLIMMIX procedure showing that there is no significant difference between any of the treatments.

Treatment (#)	Treatment (%)	Engine Speed Average Error (%)	Standard Error	Pr>|t| (Alpha=0.05)
1[A]	80-100	0.003	0.003	0.2227
2[A]	60-80	-0.002	0.004	0.6280
3[A]	40-60	0.000	0.004	0.9672
4[A]	20-40	-0.004	0.006	0.4952
5[A]	0-20	-0.004	0.006	0.5503

The resulting average error of the engine speed between both data acquisition systems was small (<±0.005%) and provided proof in the methodology used to synchronize the two datasets. This provided proof that comparison of the two fuel rates would represent a valid comparison during steady state operation.

To confirm that transient outliers were eliminated, NTTL fuel rate was plotted versus J1939 fuel rate of one of the test tractors to highlight the absence transient outliers (Figure 26). Because transient outliers produced large errors (because of NTTL measurement methods), only steady state portions of data were used to develop a comparison of actual fuel error.

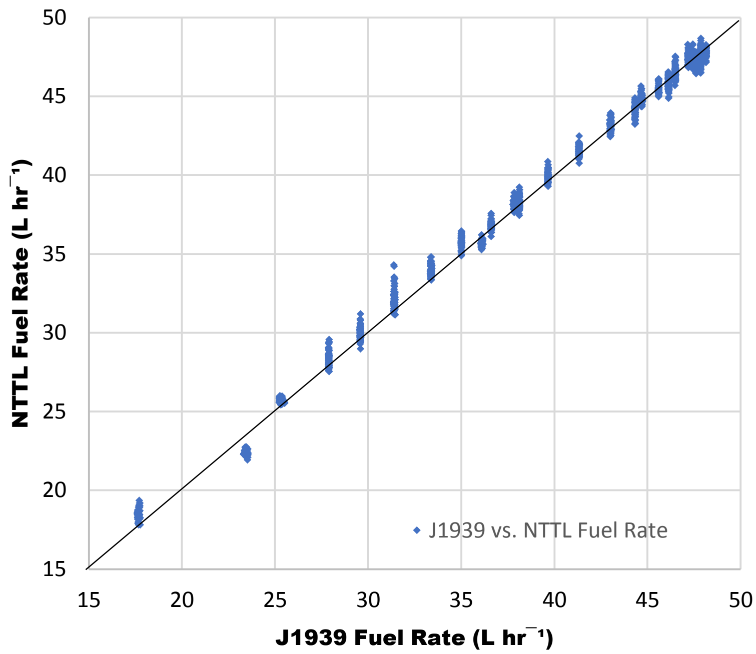


Figure 26: NTTL fuel rate plotted against J1939 fuel rate to show absence of transient outliers

Figure 27 displays the actual fuel error calculated from one of the tractors used (John Deere 8245R), which is comprised of the 22 sets of lug run data and the five additional OECD points (60 s of data within each) entered into one graph. Time transitions were eliminated to produce a chart depicting actual error from those 27 sets of 60 s data. Relative time from 0 to 1320 s shows the torque curve datasets and time from 1329 to 1620 s shows the additional OECD data points.

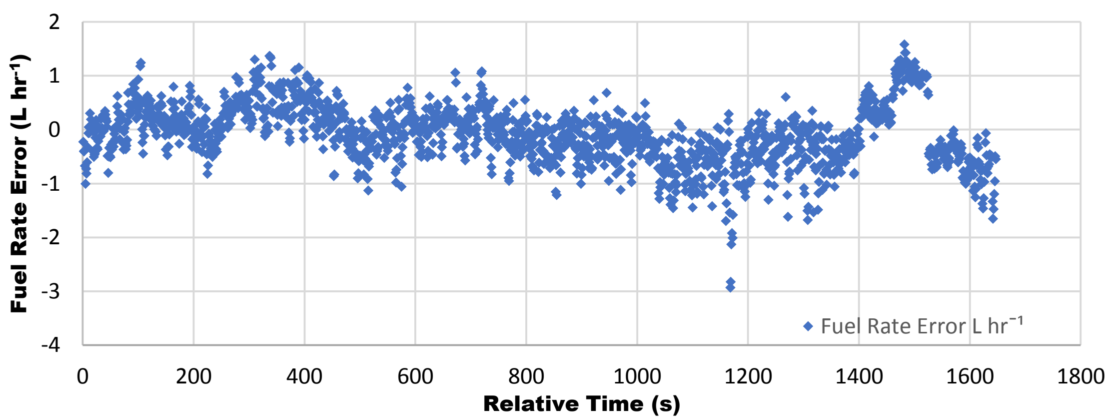


Figure 27: Actual fuel rate error (L hr^{-1}) from one of the test tractors

Because the mean percent fuel error was desired, the results of these datasets (similar to Figure 27) were used to calculate percent error. This produced six separate replications as demonstrated from the John Deere 8245R in Figure 28.

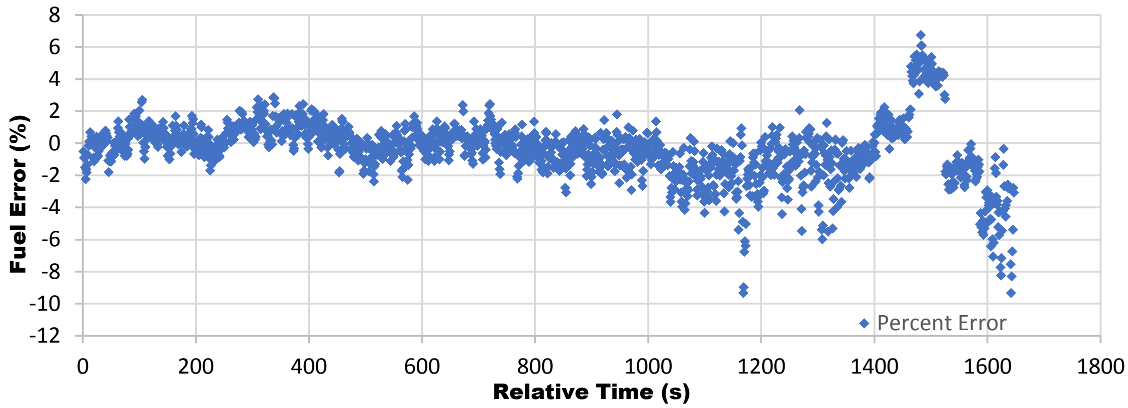


Figure 28: Percent fuel error from one of the test tractors as calculated based on equation 2

After calculating the mean percent error for engine speed error and fuel rate error, average values for each of the tractors was calculated (Table 9) to outline the error differences between the tractors. The values shown only represent an average error for each of these responses over the 27 datasets. Average fuel errors indicated that while tractors tested were similar in machine class, fuel consumption rates did not directly correlate with errors between the two measurement systems, with some tractors' J1939 fuel rate being higher than the NTTL and some being lower than the NTTL.

Table 9: Averaged error for fuel rate error, standard deviation of fuel rate error and engine speed error for each tractor used within study

Tractor	Average Fuel Rate (L hr^{-1})	Average Fuel Error (%)	Average Fuel Error Standard Deviation (%)	Average Engine Speed Error (%)
1	39.152	-0.301	0.333	0.003
2	42.739	0.160	0.370	0.001
3	47.020	0.648	0.414	0.001
4	50.677	0.200	0.488	-0.008
5	54.721	-0.276	0.351	0.000
6	58.433	-0.868	0.311	0.006

Because this represented an average across the entire test ranges, a more detailed look at the fuel error was desired. The SAS program was used to output a report on least squares means (LSM) to determine if there was significance between the different treatments (percent ranges from maximum to minimum fuel rates). Table 10 shows the LSM value outputs as treatments from 0 to 100% in 20% ranges. The Estimate shows the average percent error calculated from each treatment. For example, the 80 to 100% fuel rate treatment had an average error of 0.218%, indicating that among the six tractors used, for the highest fuel rates, the average error was less than 0.25%. Standard error (which is the error associated with each Estimate) was also calculated to show the strength of the calculated error. The probability associated with each of the treatments was provided to determine if there was significant differences between the treatments. As Table 10 shows for fuel rate mean percent error, treatments 80 to 100%, 60 to 80% and 0 to 20% were not significantly different from each other and treatments 40 to 60% and 0 to 20% were not significantly different from each other indicating that these treatments were similar to each other in magnitude of fuel error. The treatment of 20 to 40% was however significantly lower compared to all other fuel rate treatment ranges.

Table 10: Results of the Fuel Error from the GLIMMIX procedure, showing that treatment 1,2 and 5 are not significantly different, 5 and 3 are not significantly different, but 3 is significantly different than 1,2 and 5, and treatment 4 is significantly different than all treatments

Treatment (#)	Treatment (%)	Fuel Rate Average Error (%)	Standard Error	Pr>|t| (Alpha=0.05)
1^{A}	80-100	0.218	0.221	0.3254
2^{A}	60-80	0.116	0.266	0.6636
3^{B}	40-60	-0.441	0.273	0.1085
4^{C}	20-40	-1.312	0.342	0.0002
5^{AB}	0-20	-0.311	0.350	0.376

To illustrate fuel error from tractor by treatment, the fuel errors from each point were plotted (Figures 29 and 30). Figure 29 shows fuel error points as recorded based on the tractor which produced them. Figure 30 shows the fuel error based on flow rate, which provides a comparison to the NTTL fuel measurement system error shown previously in Figure 18. The error shown in Figure 30 indicates higher error at lower flow rates; however, the errors were much greater than those evaluated for the NTTL fuel collection system (Figure 18).

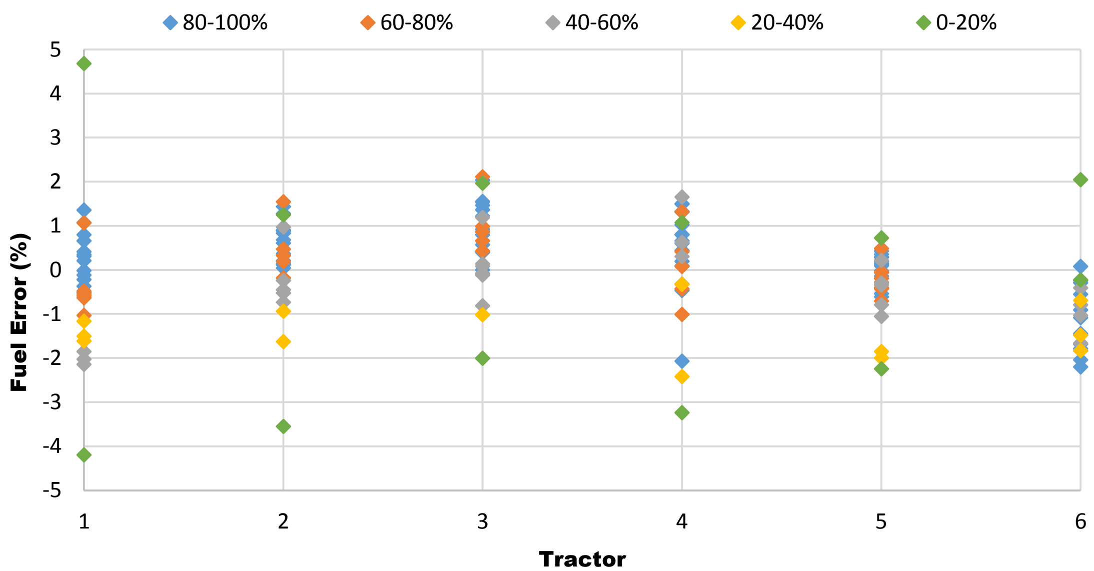


Figure 29: Fuel error from each treatment for the six tractors used in this study

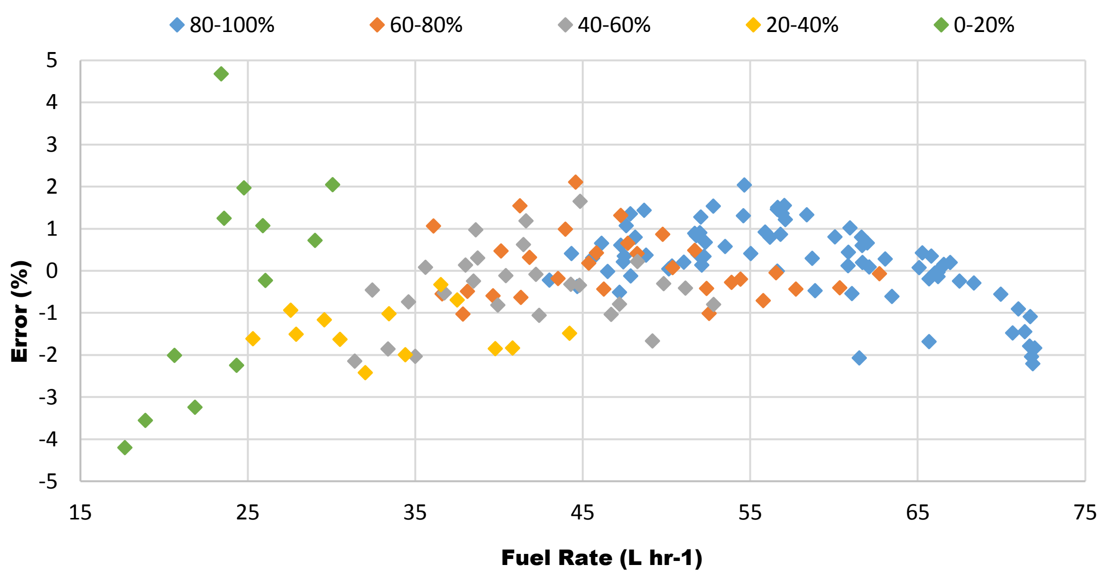


Figure 30: Fuel error by volumetric flow rate

To further represent error derived from treatments, a graph was created to depict error by treatment rather than tractor (Figure 31). This clearly demonstrates that treatment 5 (0 to 20%) had a much wider distribution of error than any of the other treatments.

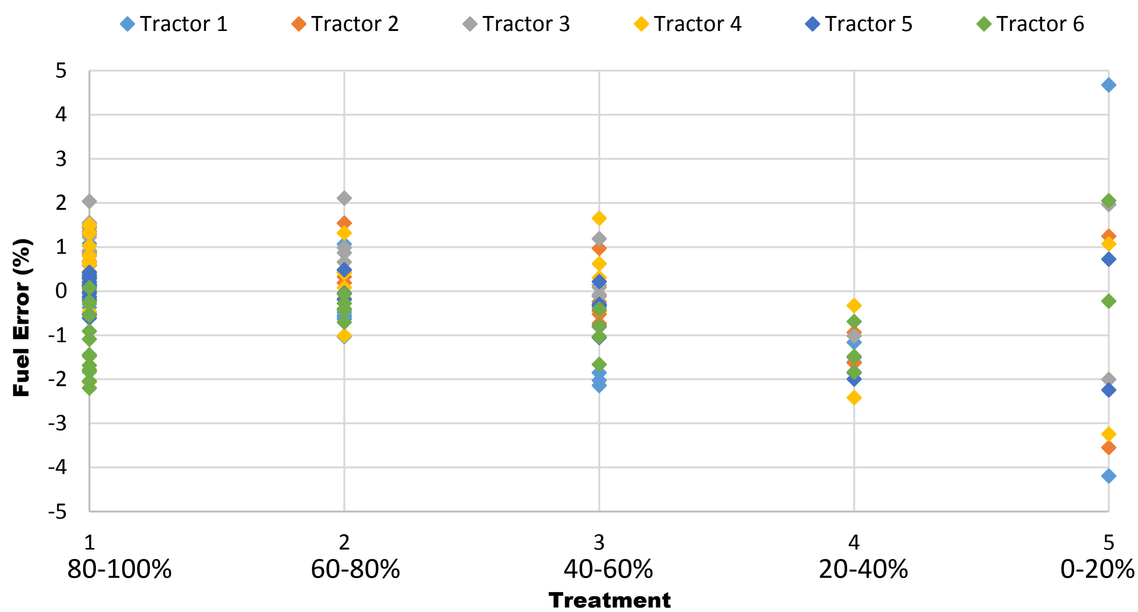


Figure 31: Fuel error from each tractor shown by treatment (Trt 1=80-100%. Trt 2=60-80%. Trt 3=40-60%, Trt 4=20-40%, and Trt 5= 0-20%)

The fuel error by treatment results shown in Figure 31 reinforced that treatment number five (0 to 20% fuel rate) had a good deal of variation. To quantify the magnitude of this variation, standard deviation of the fuel rate error for each treatment was calculated. Table 11 shows the average fuel rate error and standard deviation per treatment to depict the variation within the treatments. From table 11, it is clear that treatment five (0 to 20% fuel rate) had the highest variation with a standard deviation of over 2.75. The datasets used for treatment five were primarily from the five additional OECD points (varying throttle, varying loads).

Table 11: Results of standard deviation calculation of fuel which shows that treatment five had a high variation in fuel error for every tractor.

Treatment (#)	Treatment (%)	Fuel Rate Average Error (%)	Fuel Rate Error Standard Deviation (%)
1	80-100	0.218	0.926
2	60-80	0.116	0.769
3	40-60	-0.390	0.922
4	20-40	-1.420	0.575
5	0-20	-0.311	2.728

Conclusions

Results of the engine speed error analysis proved that J1939 and NTTL data were properly synchronized allowing for analysis of fuel rate comparison. The fuel error analysis (Table 10), indicated that there was error from each of the tractors used, as well as greater error from the lower three fuel rate treatments (0 to 60% of full flow). While the analysis methods were consistent for the study tractors, some showed a positive error between fuel rates where other tractors showed negative error within the same datasets. This caused a reduction in the average error when the results of all six test tractors were combined. For any tractor in this horsepower range, chosen at random, the results represented an accurate error prediction. If an absolute error value were calculated, the results would likely be quite different. Absolute error would indicate positive error between fuel measurement systems, which would indicate that CAN bus fuel rates were always estimated at higher levels that in reality, which was not the case based on this study.

The data collected during testing were considered to give an accurate depiction of actual field conditions due to the high loads as well as varying loads and throttle positions. The results of the standard deviation calculations for each fuel treatment showed that treatment five (0 to 20% fuel rate) had a great deal of variation (2.73%). Treatment five was primarily OECD additional points which consisted of only varying throttle positions and varying load conditions. Tractors equipped with IVT transmissions or when the principle of “throttle back and shift up a gear” are used, the parameters of this study may be inaccurate as a replication of real field conditions. These both result in more partial throttle partial load operations which could lead to a higher standard deviation in error as seen in treatment 5 (0 to 20%). Although this study showed treatment five to have a low mean percent error, because the variation was so high, if only partial throttle partial load operations were used, actual fuel consumption might not be as accurate as depicted by the CAN bus. Only one tractor equipped with an IVT was used for this study, but for the test points used, varying transmission output had little to no effect in comparison with non-IVT tractors for the steady state PTO tests.

Even though there was error produced in the comparison between the calculated J1939 fuel rate and physically measured NTTL fuel rate, whether looking at the statistical analysis or averaged analysis, the error was quite minimal (always less than ±5%). The only time that a greater error was seen was in the pilot study during transient fuel rates. If this test were to be repeated in the field with the same equipment (i.e., NTTL fuel measurement system) a greater error would likely be noticed because of the constant change in engine speed and/or load. However, because the error shown by this study was relatively low during steady state conditions, depending upon the use of the data, J1939 fuel rate data could certainly be used for logistical and research purposes rather than attempting to physically measure fuel rate consumption with flow meters.

Chapter 4: Summary and Conclusions

Controller Area Network including higher layer protocols such as SAE J1939, ISO 11783 and NMEA 2000 has become a common source for a wide variety of information regarding machine operation in mobile agricultural field equipment. Because this trend will only continue to grow, knowing that the information being collected is accurate becomes more and more important. As farm managers and researchers begin to rely on this information rather than calculating based on former procedures or having some type of physically measured value, the need for accuracy knowledge of these messages will also only increase.

As field cropping machines continue to grow in complexity, the use of CAN bus data will also likely become more heavily relied upon. Automation in field cropping systems and equipment is a growing trend in agriculture (Darr, et al., 2004; Powell, 2005). Considering recent advancements in agricultural telematics, data accuracy broadcast to and from this equipment is rather important.

This study was able to produce significant results regarding not only the accuracy of multiple data logging methods, but additionally the accuracy of one of the most desired messages for calculating field efficiency, fuel rate. Because the tractors used for this study varied in horsepower, and gave a general population for mid to high horsepower machines, the results showing a low error from statistical analysis illustrates a confidence in the fuel rate being produced via the SAE J1939 fuel rate message.

From data collected during this study, it was clear that logging from different sources did not yield a great enough difference to choose one over the other. It also provided evidence that re-sampling higher frequency data sets produced a useable and more manageable dataset. The greatest result from the assessment of different data collection methods was actual file size. Because raw hexadecimal files were so large, logging for long periods of time could become a hassle if not impossible. This points towards logging data as an averaged set in a fashion as presented in this study. This would greatly reduce overall file size while still providing accurate, desired information.

This study and the pilot study conducted showed that because of the fuel measurement system used by NTTL, transient data couldn't be used as a result of the lag in the NTTL system. By looking at the J1939 fuel rate data, at transient periods the fuel rate changed almost instantaneously. This leads us to the questions, if this is, in fact the actual fuel consumption, or could it be based primarily on requested fuel rate. If a more accurate depiction of actual fuel consumption could be determined, the lag in the NTTL system might not be as far off as this study shows. Setting aside the question of transient conditions, viewing steady state actual accuracy of the engine fuel rate as logged by the CAN bus, it was evident that there was some amount of error (always less than ±5%) but generally closer to ±1% during high load, steady state operations. This offers an assurance that the fuel rate as portrayed by the CAN bus could be used for management decisions or research objectives.

Future Work

At steady state conditions, little error (±5%) was found between the calculated CAN bus fuel rate and the measured NTTL fuel rate. Steady state however is rarely used in real life operating conditions. When a tractor is working in the field, the terrain is rarely perfectly flat creating, and the load is rarely constant. This shows that although this study validated the accuracy of the SAE J1939 fuel rate message in agricultural tractors, additional studies could be performed to determine results of transient loads. When looking at a graph of the J1939 fuel rate during transient states, the transient is almost a perfect step function. It is highly unlikely that the actual fuel being consumed by the tractor during transient periods creates a perfect step function. By creating and using a system that more accurately measures transient fuel rates, further studies could be performed to indicate true fuel usage during actual farming operations. This would also tie into a better understanding of the partial load/partial throttle applications where in this study a higher percent error was found.

By finding a more accurate transient load error approximation, a might be possible to find a coefficient for calculating a more precise fuel use rate in actual farming operations.

References

Bell, J. (2002). *Network protocols used in the automotive industry.* Unpublished manuscript.

Bosch, R. (1991). *CAN specification version 2.0.* (No. 300240). Stuttgart: Robert Bosch GmbH.

Colvin, S., McConnell, K., & Catus, B. (1989). 'TERMS": A computer model for field simulation. *Transactions of the ASAE, 32*(2), 391-392-396.

Darr, M. (2012, CAN bus technology enables advanced machinery management. *Resource: Engineering & Technology for a Sustainable World, 19.5*, 10-11.

Darr, M. (2013). In AETC (Ed.), *CPD #2- CAN bus*. Kansas City, KS:

Darr, M., Stombaugh, T., Shearer, S., & Fulton, J. (2004). *CAN-based distributed control for autonomous vehicles.* Unpublished manuscript.

Fountas, S., Sorensen, C., Tsiropoulos, Z., Cavalaris, C., Liakos, V., & Gemtos, T. (2015). Farm machinery management information system. *Computers and Electronics in Agriculture, 110*, 131-138.

Goering, C., Stone, M., Smith, D., & Turnquist, P. (2006). *Off-road vehicle engineering principles* (Reprint ed.). St. Joseph, MI: American Society of Agricultural Engineers.

Grisso, R., Perumpral, J., Vaughan, D., Roberson, G., & Pitman, R. (2014, Predicting tractor diesel fuel consumption. *Virginia Tech Cooperative Extension, 442-073*

Grisso, R., Vaughan, D., Perumpral, J., Roberson, G., Pitman, R., & Hoy, R. (2012). Using tractor test data for selecting farm tractors. *POQOPRIVREDNA TEHNIKA AGRICULTURAL ENGINEERING,* , 1.

Agricultural tractor test standards in america, StandardU.S.C. (2011).

Kim, K., Bashford, L., & Sampson, B. (2005). Improvement of tractor performance. *Biological Systems Engineering: Papers and Publications, 1*(1), 46.

Kvaser. (2014). Higher layer protocols. Retrieved from http://www.kvaser.com/about-can/higher-layer-protocols/

Marx, S., & Luck, J. (2013). Assessing accuracy of machine CAN bus data using SAE J1939 and nebraska tractor test laboratory data. *American Society of Agricultural and Biological Engineers Annual International Meeting 2014*, Montreal Canada. *, 1* 1893-1894-1902.

Microsoft. (2014). Excel specifications and limits. Retrieved from https://support.office.com/en-nz/article/Excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3

Mudafale, K., Lutade, S., & Gosavi, G. (2013). A paper on common rail direct injection engine technology. *International Journal of Research in Aeronautical and Mechanical Engineering, 1*(5), 32.

National Instruments. (2014). *NI X-NET hardware and software manual* (1st ed.). Autsin, TX: National Instruments.

Nolte, T., Hansson, H., & Norstrom, C. (2002). Minimizing CAN response-time jitter by message manipulation. *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE,* 197-206.

OECD standard code for the official testing of agricultural and forestry tractor performance, Code 2U.S.C. (2012). doi:July 2012

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. *Applied Engineering in Agriculture, 30*(6), 829-830-839.

Powell, N. (2005). A sensor integration method for autonomous equipment.

Ruß, G., & Brenning, A. (2010). Data mining in precision agriculture: Management of spatial information. *Computational intelligence for knowledge-based systems design* (pp. 350-359) Springer.

Russo, J. (2013). Big data & precision agriculture. Retrieved from http://www.oxfordjournals.org/our_journals/exbotj/for_authors/index.html

SAE J1939-71 vehicle application layer, StandardU.S.C. (2009).

Schueller, J. (1988). A proposed scheme for integrated tractor-implement communications and control. *American Society of Agricultural Engineers (Microfiche Collection)(USA),*

Stone, M., Benneweis, R., & Van Bergeijk, J. (2008). Evolution of electronics for mobile agricultural equipment. *Transactions of the ASABE, 51*(2), 385-386-390.

Stone, M., McKee, K., Formwalt, C., & Benneweis, R. (1999). *ISO 11783: An electronic communications protocol for agricultural equipment* ASAE.

Stone, M., & Zachos, M. (1993). Application of J1939 networks in agricultural equipment. *American Society of Agricultural Engineers. Meeting (USA). no. 931530.*

Trostle, R. (2010). *Global agricultural supply and demand: Factors contributing to the recent increase in food commodity prices* DIANE Publishing.

Udompetaikul, V., Upadhyaya, S., & Vannucci, B. (2011). The effect of tire inflation pressure on fuel consumption of an agricultural tractor operating on paved roads. *Transactions of the ASABE, 54*(1), 25-30.

van Rijmenam, M. (2013). From machines to crops to animals: Big data turns traditional farming upside down. Retrieved from http://smartdatacollective.com/bigdatastartups/141501/machines-crops-animals-big-data-turns-traditional-farming-upside-down

Voss, W. (2005). *A comprehensible guide to controller area network* Copperhill Media.

Voss, W. (2008). *A comprehensible guide to J1939* Copperhill Technologies Corporation.

Walter, R. (2015). Fuel economy using J1939 CAN for heavy duty vehicles. Retrieved from http://www.hemdata.com/products/dawn/enews/fuel-economy-j1939

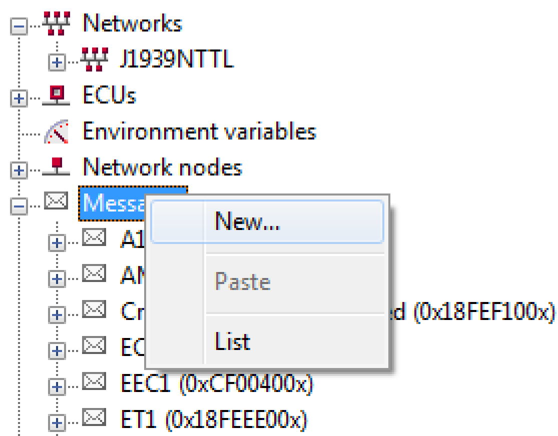
Wold, M., Kocher, M., Hoy, R., Lammers, A., & Blankenship, E. (2015). *Development of a calibration procedure for diesel fuel flow measurement with a coriolis effect meter.* Unpublished manuscript.

Appendices

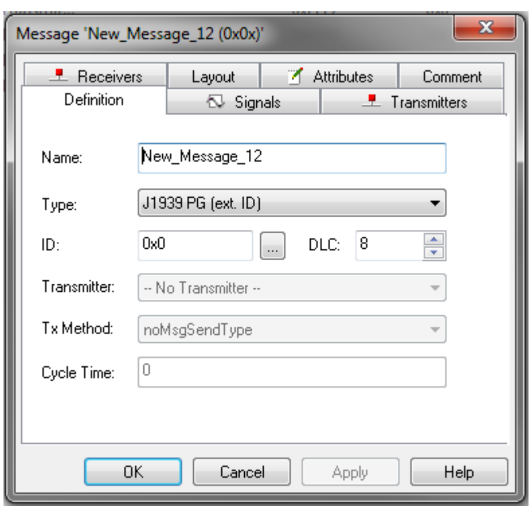
Appendix A: Vector Database Tutorial

Adding Messages to a Vector 8.1 Database

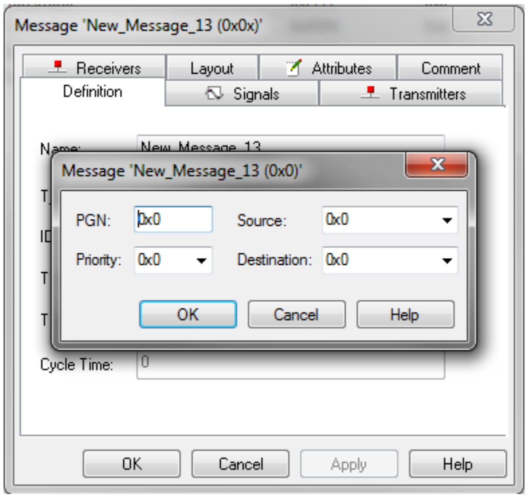
1. On messages, right click and select “New.”



2. This will prompt an area to create the parameters of the new message.



3. Enter the message name. For example, if the message is for the Electronic Engine Controller 1, the acronym could be EEC1.
4. Click on the button for editing the message ID.



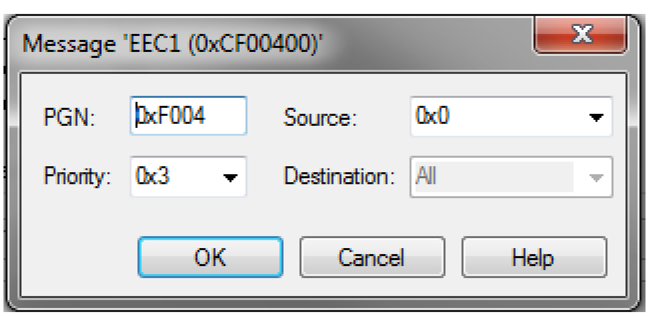
5. Locate the Parameter Group Network information on the SAE J1939 document. Within this document, use the PGN to enter into the database.

PGN 61444 Electronic Engine Controller 1

Engine related parameters

Transmission Repetition Rate:	engine speed dependent	
Data Length:	8	
Extended Data Page:	0	
Data Page:	0	
PDU Format:	240	
PDU Specific:	4	PGN Supporting Information:
Default Priority:	3	
Parameter Group Number:	61444 (0x00F004)	

6. Enter in the PGN as found on the SAE J1939 document.

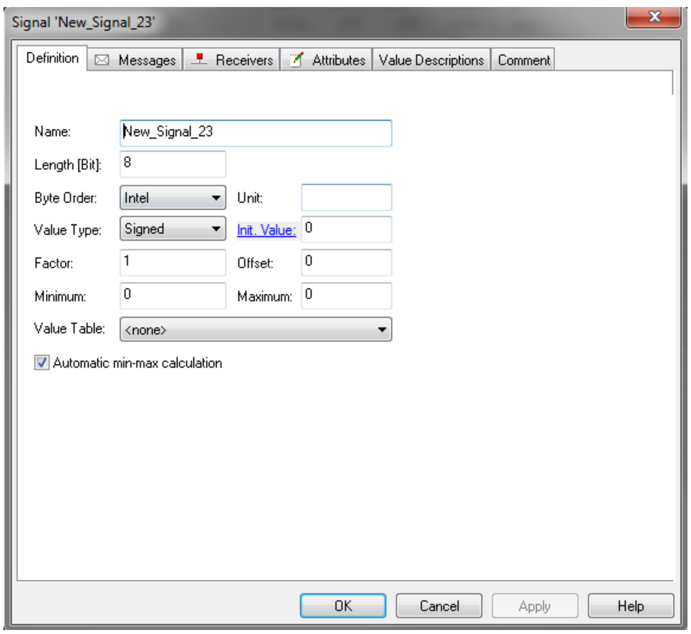


7. After creating the new message, signals can be added. To add a signal, locate the signal desired to be added in the SAE J1939-71 document. For this example,

Engine Speed is the desired signal. Under EEC1, Engine speed is defined as SPN 190. Note that by counting the length of the messages preceding SPN 190, the start bit can be located. For SPN 190, the start bit is 24.

Start Position	Length	Parameter Name	SPN
1.1	4 bits	Engine Torque Mode	899
1.5	4 bits	Actual Engine - Percent Torque High Resolution	4154
2	1 byte	Driver's Demand Engine - Percent Torque	512
3	1 byte	Actual Engine - Percent Torque	513
4-5	2 bytes	Engine Speed	190
6	1 byte	Source Address of Controlling Device for Engine Control	1483
7.1	4 bits	Engine Starter Mode	1675
8	1 byte	Engine Demand – Percent Torque	2432

8. Now under the signals section of the Vector database editor, right click and select new. This will create a new signal ready to be edited.



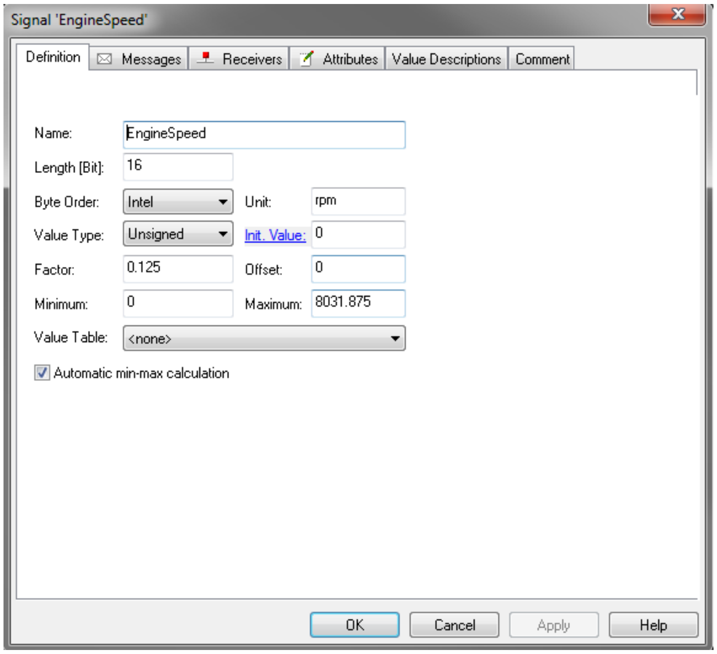
9. In the new signal, specify the name (from the SPN name), and the additional information again as stated in the SAE J1939 document.

SPN 190 ***Engine Speed***

Actual engine speed which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders.

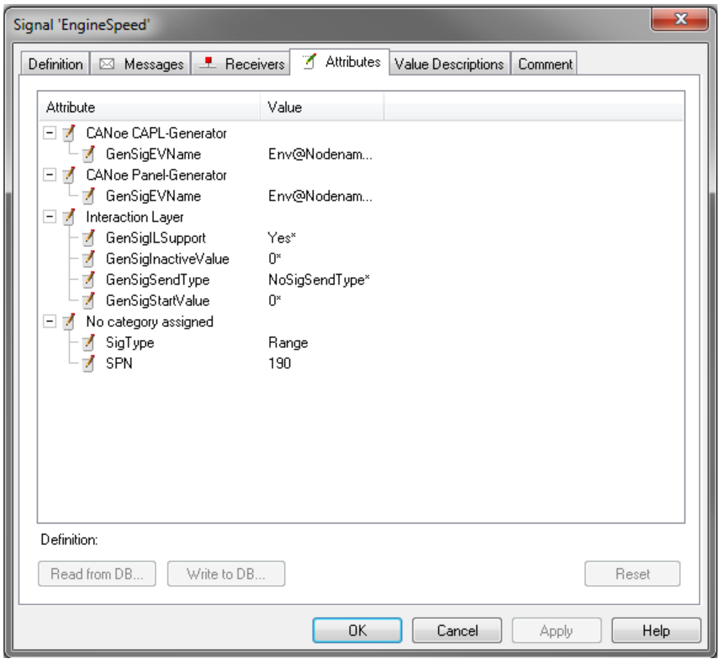
Data Length:	2 bytes	
Resolution:	0.125 rpm/bit, 0 offset	
Data Range:	0 to 8,031.875 rpm	Operational Range: same as data range
Type:	Measured	
Supporting Information:		
PGN reference:	61444	

10. After filling out the information, the new signal will be mostly done.

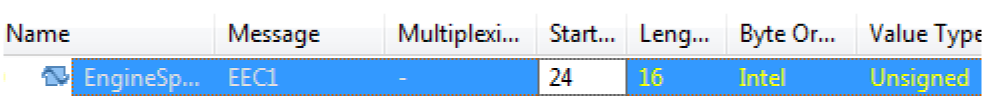


11. To add the signal to the message that was previously created, click on the "Messages" tab at the top of the signal and choose "Add". This will allow you to select the message to which the newly created signal belongs.

12. After adding the signal to the message, a few additional details need to be finished. Under the attributes tab, enter in the SPN number and the value type.



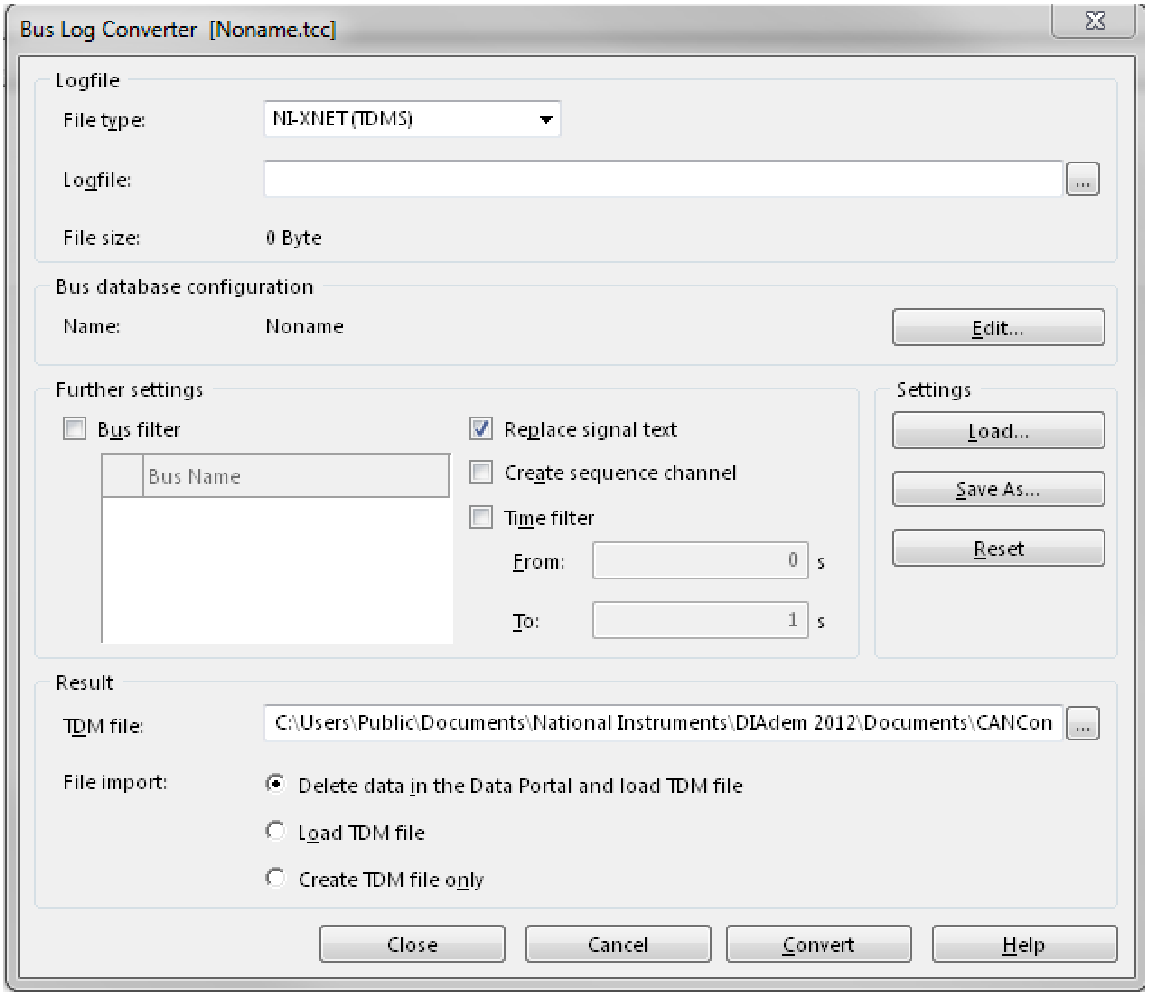
13. The last step to completing the signal is to specify the start bit. This can be done by simply double clicking on the start bit number located in the main pane of the vector DB editor and typing in the start bit.



Appendix B: NI DIAdem Tutorial

Appendix (i): Using NI Diadem (2012 SP1) to Convert CAN to Engineering Units

1. Open NI Diadem
2. In the “Navigator” tab, go to File->Bus Log Converter…
    a. When the Bus Log Converter opens, choose the file type in the drop down box. This allows you to choose file types like NI-XNET, NI-CAN, Vector file formats, and Kvaser file types.
    b. Next choose the Log file that is desired to be converted. (Note: only logfiles of the chosen type will be shown when searching for the desired logfile)
    c. Under the Bus database configuration section, click on the Edit button to allocate a database for the conversion. In the Edit window, click on the Add Bus to Configuration button (it is a button that simply says CAN with a small red animation). Once you choose the database required to convert the CAN messages, you can close the Edit window.
    d. There is the ability to additional filtering before converting, but for all simplicity and to receive all messages recorded, there’s no need for additional filtering here.
    e. Before converting messages, you can choose a file name for the converted logfile, and then choose how to load the converted file into Diadem.
    f. Hit the Convert button and the Bus Log Conversion will begin.



Appendix (ii): Using Diadem to analyze CAN Data

1. Once the data is converted and imported into Diadem (this happens either after you convert the data, or import data through the Navigation tab), you can begin to view the data through the View tab.
    a. Within the view tab, on the right hand side is the Data Portal, this is where data sets can be selected to be put into the display section of the pane. Diadem gives multiple options for customization for viewing data, graphs, etc. but for this example a simple view with a graph, data portal and data view pane will be used.

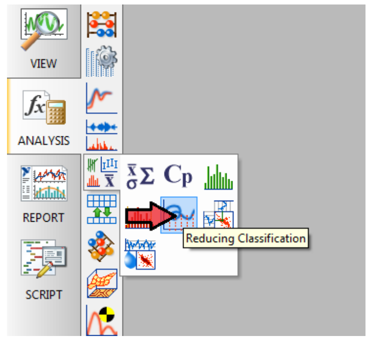
b. In the Data viewing pane, for the data imported from the previous Bus Log Conversion, the data is shown by separate ECU's. Each ECU is given a time stamp, and then the data from that ECU is shown next to that time stamp. This is where it is easy to identify the frequency in which each ECU logs data. For example, the Electronic Engine Control (EEC1) logs Engine Speed and Actual Percent Engine Torque at 100Hz. Whereas Fuel Rate is logged at 10Hz, and Engine Fuel and Coolant Temps are logged at 1Hz.

c. The viewing pane also gives information about each channel. For example, the time for recording was approximately 34,000 seconds (approx.. 9.5 hours), and because Engine Speed was logged at 100Hz, the viewing pane shows that there are over 3,400,000 data points for Engine Speed. Information like this is beneficial for future averaging needs.

All Cha...										
Name	Time	Actual_Engine_P...	EngineSpeed	Time	FuelRate	Time	Fan_Speed	Time	EngCoolantTemp	EngFuelTemp1
Number	1	2	3	12	13	4	5	6	7	8
Length	3432781	3432781	3432781	343278	343278	34327	34327	34328	34328	34328
Unit	s	%	rpm	s	L/hr	s	rpm	s	deg C	deg C
Channel Contents										
1	0.0041376	18	2194.5	0.0152322	16.35	0.9965352	1205	0.2766123	72	26
2	0.0146559	18	2193.5	0.1163792	16.1	1.9965753	1204	1.2766039	72	26
3	0.0240401	18	2194.75	0.2163794	16.3	2.9969282	1206	2.2769604	72	26
4	0.0344895	18	2194.75	0.3163837	16.3	3.9969099	1205.5	3.2770017	72	26
5	0.0437063	18	2193.5	0.4152362	16.15	4.9967611	1205.25	4.2766095	72	26
6	0.0546807	18	2195.25	0.5158485	16.25	5.9969514	1204.5	5.2770172	73	26
7	0.0635631	18	2194	0.616655	16.25	6.9966397	1204.5	6.2767494	73	26
8	0.0746136	18	2194.75	0.7165125	16.1	7.9969572	1204.5	7.2767618	73	26
9	0.0839528	18	2196	0.8166665	16.3	8.9964092	1203	8.2766831	73	26
10	0.0946582	18	2194	0.9158523	16.15	9.9966867	1204.25	9.2759069	73	26
11	0.1040598	18	2195.75	1.0151926	16.25	10.9966963	1202.75	10.2758203	73	26
12	0.1147084	18	2195.25	1.1152904	16.35	11.9964188	1201.25	11.2758082	73	26
13	0.1240252	18	2194	1.2163791	16.25	12.9968444	1202.5	12.2767779	74	26
14	0.1345109	18	2195.25	1.3157635	16.1	13.9967345	1201	13.2761643	74	26
15	0.1436457	18	2194.75	1.4151485	16.3	14.9966697	1201	14.2766143	74	26
16	0.1548825	18	2194	1.5157076	16.25	15.9967499	1200.25	15.2766359	74	26

Appendix (iii): Averaging Data/ Resampling to a lower frequency

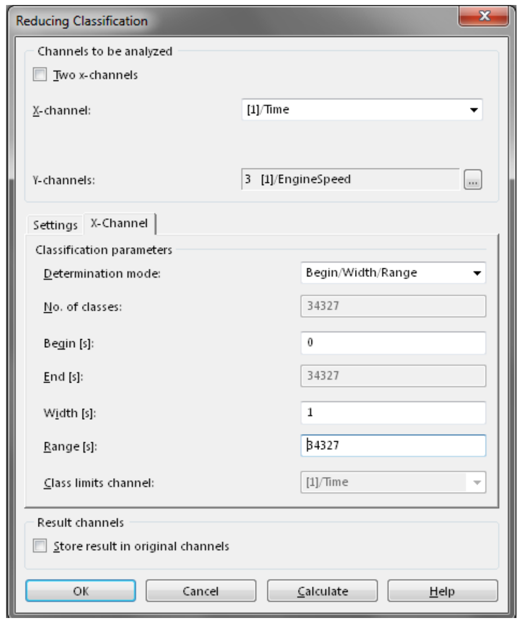
1. To resample high frequency data, Diadem uses a tool called Reducing Classification to average data associated with a time signal. The Reducing



classification is located under the Analysis tab, Statistics button, the Reducing Classification.

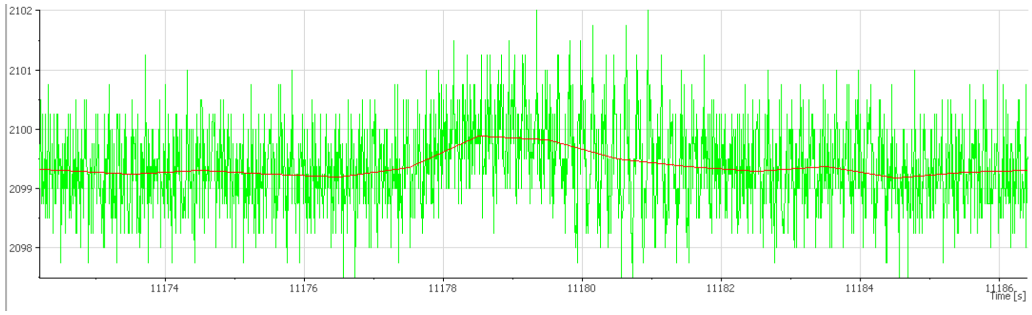
2. Once the Reducing Classification opens, you can choose the time signal as the x-channel, and the desired signal to be reduced as the y-channel. For example, if the engine speed is set as Channel 1 in the Diadem View Pane, you would choose [1] /Time as the x-channel, and [1] /Engine Speed as the y-channel. There are then two tabs to be aware of within the Reducing Classification, the Settings tab and the X-Channel tab. Choose Mean as the reduction mode, then go to the X-Channel tab.

3. Under the X-Channel tab, under the determination mode drop down, choose Begin/Width, Range. This allow for the ability to enter where you want the resampled data to begin, the width of each resampled point, and the range of data in which you desire to resample. Again using the Engine Speed as the desired channel to be resampled, the data length can be found in the Viewing Pane, for



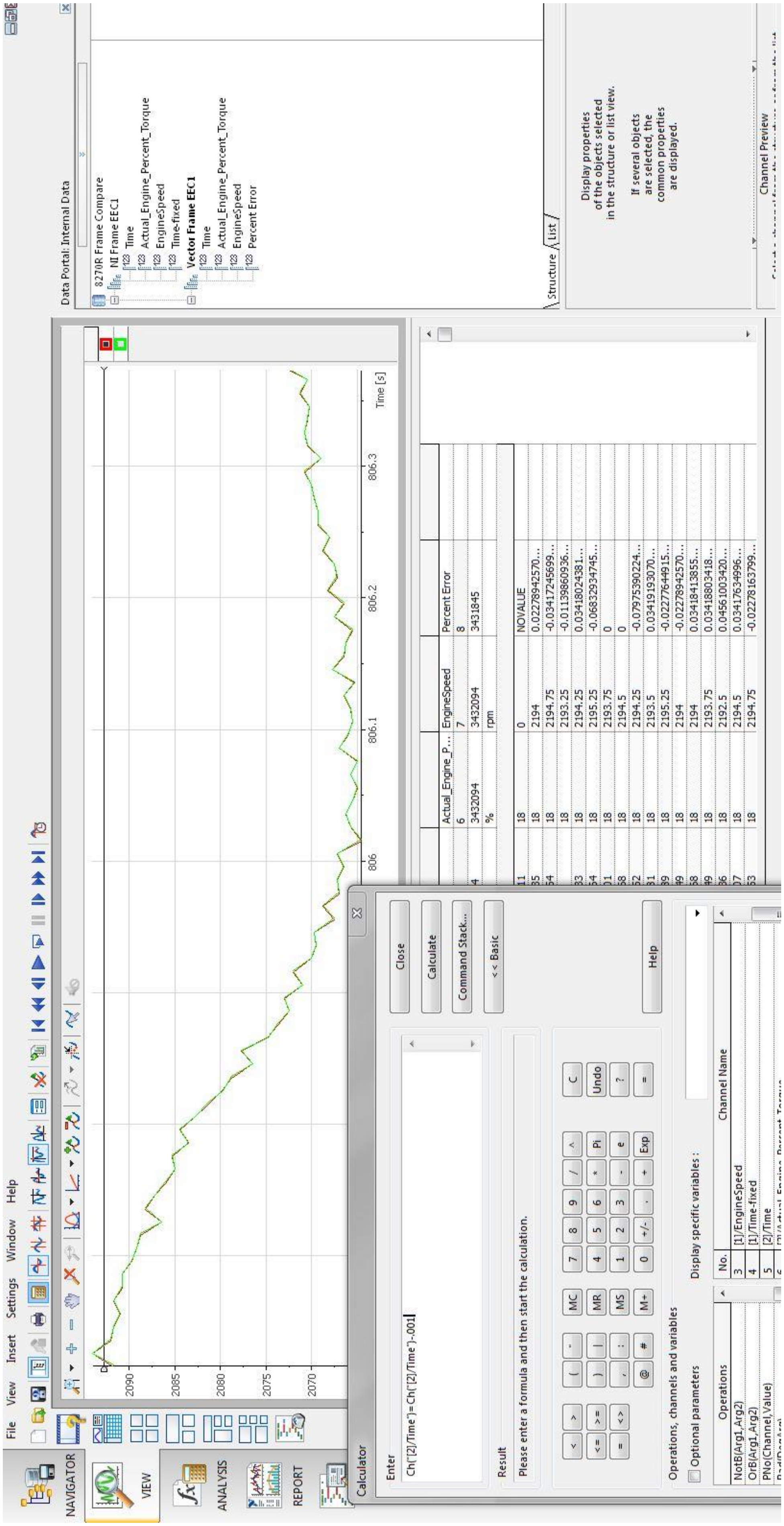
Engine Speed, it has 3432781 data points. Knowing that the recorded frequency was 100Hz, the length can be divided by the frequency to indicate what the desired range is. 3432781/100=34327.81 seconds. Once you calculate, two new channels will be created named ReducingClassificationX and ReducingClassificationY. Ideally, in the Data Portal you can change the names to New Time and Engine Speed 1Hz, or something similar.

4. By choosing the original signal and the reduced signal, it is clear to see that the information is a carbon copy of the original data, simply reduced.



Appendix (iv): Synchronizing Time

1. In the View tab, have two time channels with their respective data (e.g. Engine Speed) set up to be viewed in the graph. Do this by right clicking on the graph, then choose “Display” and enter in the two time channels and the two channels you desire to synchronize.
2. In the View tab, open the calculator function.
3. Double click on one of the times for one of the channels (e.g. if you have two sets of data with similar time stamps, simply chose one of the time columns).
4. After double clicking one of the time channels, then chose “equals” and then double click the same time channel as chosen previously.
5. Add or subtract any number depending on how far you desire to move that channel’s values.
6. Click the “Calculate” button. This will keep the calculator function open, but move the time channel in real time (as you press calculate, one of the times will shift on the graph).
7. Do this as many times as needed, and by changing the magnitude and direction of the value.



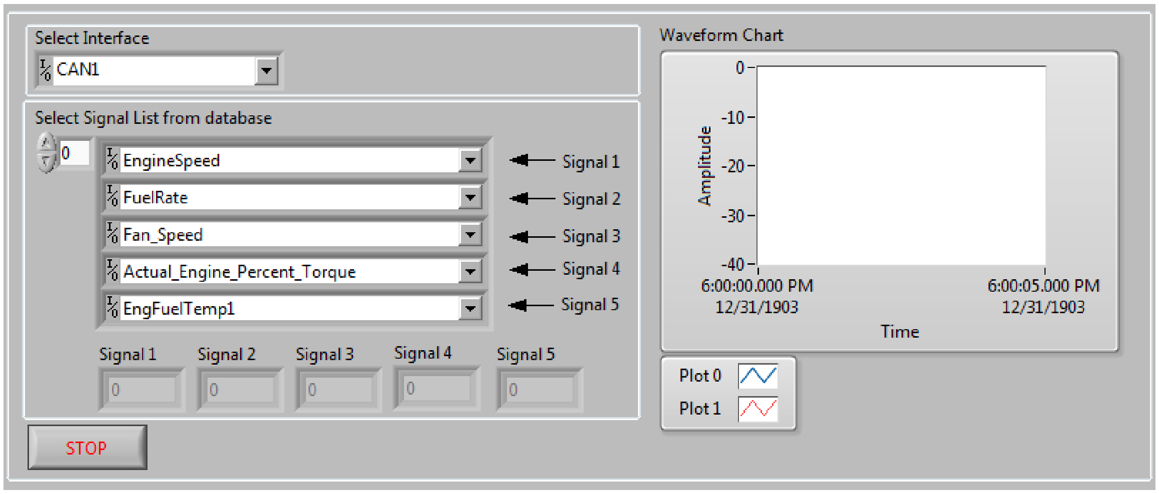
File View Insert Settings Window Help
NAVIGATOR
VIEW
ANALYSIS
REPORT
Data Portal: Internal Data
8270R Frame Compare
NI Frame EEC1
Time
Actual_Engine_Percent_Torque
EngineSpeed
Time-fixed
Vector Frame EEC1
Percent Error
Time [s]
Calculator
Enter
Ch('[2]/Time')=Ch('[2]/Time')-.001
Result
Please enter a formula and then start the calculation.
Close
Calculate
Command Stack...
<< Basic
Help
Operations, channels and variables
Optional parameters
Display specific variables :
Structure
List
Display properties of the objects selected in the structure or list view.
If several objects are selected, the common properties are displayed.
Channel Preview

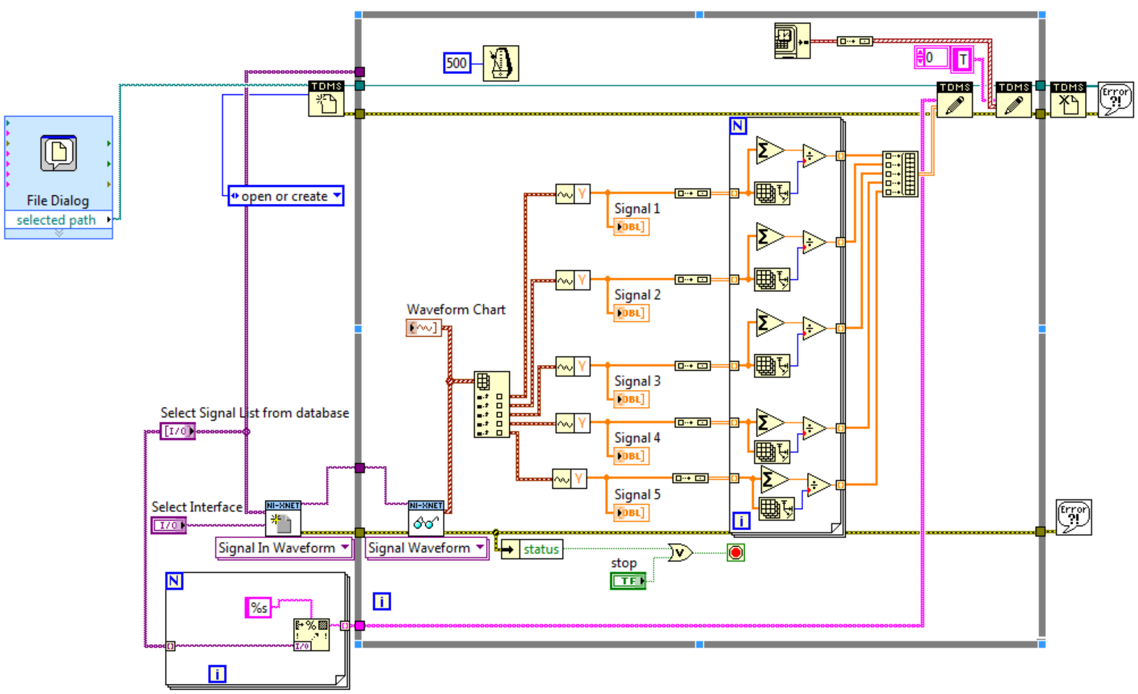
Appendix C: SAS 9.4 Programming

```
data J1939_Fuel_Error;
input tractor trt fuel_error;
datalines;

Data Lines Not Shown

symbol1 color=black value='dot';
 symbol2 color=blue value='dot' ;
 symbol3 color=green value='dot' ;
 symbol4 color=orange value='dot';
 symbol5 color=purple value='dot';
 run;
proc glimmix;
class tractor trt;
model fuel_error=trt;
random tractor;
covtest /cl;
lsmeans trt/diff lines;
run;
```

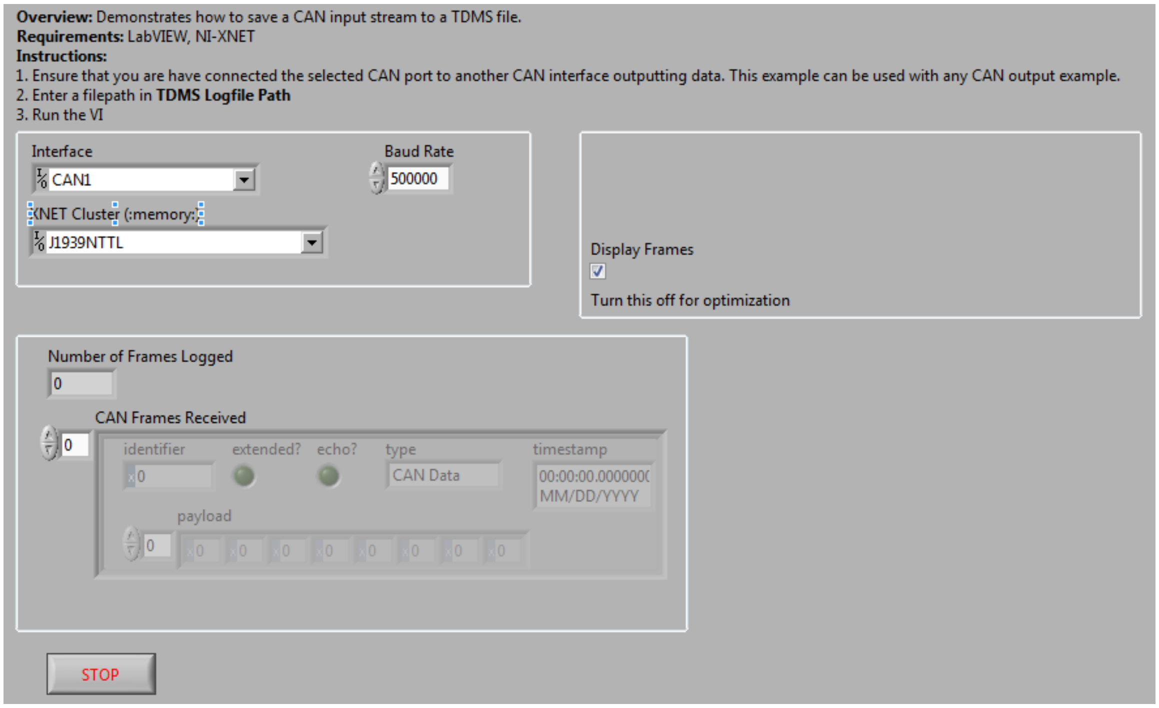
Appendix D: LabVIEW Programming

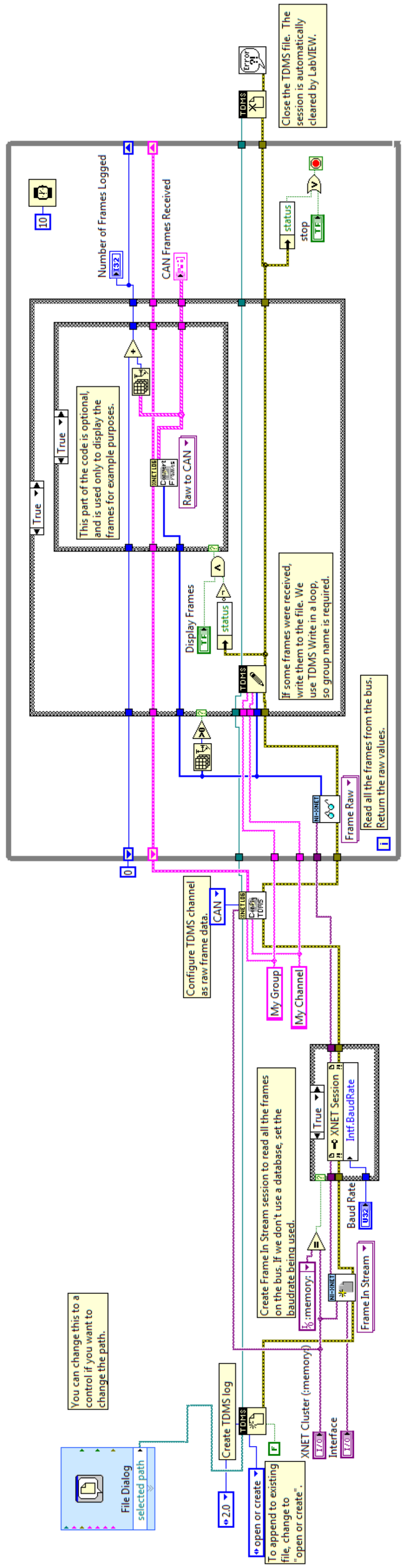
Appendix (i): 2012 NI LabVIEW API for Waveform Data Collection





Appendix (ii): 2012 NI LabVIEW API for Frame Data Collection




Close the TDMS file. The session is automatically cleared by LabVIEW.
Number of Frames Logged
CAN Frames Received
status
stop
This part of the code is optional, and is used only to display the frames for example purposes.
True
Raw to CAN
Display Frames
status
If some frames were received, write them to the file. We use TDMS Write in a loop, so group name is required.
Frame Raw
Read all the frames from the bus. Return the raw values.
Configure TDMS channel as raw frame data.
CAN
My Group
My Channel
XNET Session
Intf.BaudRate
Baud Rate
Create Frame In Stream session to read all the frames on the bus. If we don't use a database, set the baudrate being used.
:memory:
Frame In Stream
XNET Cluster (:memory:)
Interface
You can change this to a control if you want to change the path.
File Dialog
selected path
Create TDMS log
2.0
open or create
To append to existing file, change to "open or create".

Appendix E: Micro Motion Mass Flow Sensor Data

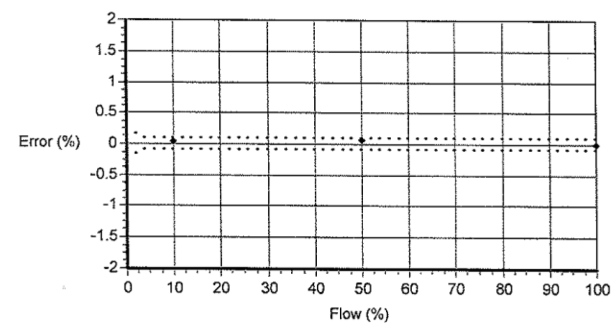
Micro Motion, Inc. — Mass Flowmeter Calibration Certificate — Dyno Fuel — **12046127**

Product Code	Serial ID	Order ID	Line	Item	Customer Tag
CMFS015M324J2BMEZZZ	12046127	10077324	1	1	
PUCK800 (transmitter)	08014396				

Process

Process ID:1.25601285
Process Time:2010.02.26 16:35:07
Process Stand:SSF1A@SSCB
Stand Uncertainty:+/-0.030%
Fluid:H20
100% Rate:5.4 KG/MIN 714.29773 lb/hr
Pickoff:1
100% P/T:63.08 PSIG/23.4 C

Detail



Flow (%)	Flow Rate (kg/min)	Meter Total (kg)	Reference Total (kg)	Error (%)	Specification (±%)
100.0	5.4	6.20496	6.205369	-0.007	0.100
10.0	0.54	1.074069	1.073629	0.041	0.100
50.0	2.7	1.86345	1.862411	0.056	0.100
100.0	5.4	6.0822	6.082263	-0.001	0.100

Results

Status:PASS
D1:0
D2:1
K1:6716.993
K2:7364.347
DT:4.49
FD:1600
DTG:0
DFQ1:0
DFQ2:0
FlowCal:3.37114.88
FFQ:0
FTG:0
DensCal:06717073644.49
FCF:3.3711
FT:4.88

MEIS, ROBERT
Technician

Traceable to International Standards. Meter total based on pulse output. Details at www.micromotion.com. 2010.02.26 16:35:12 1/1

Micro Motion, Inc. — Transmitter Configuration Report — **2294829634**

Product Code	Serial ID	Order ID	Line	Item	Customer Tag
CMFS015M324J2BMEZZZ	12046127	10077324	1	1	
2500D3ABBMEZZZ	2294829634	10077324	2	1	
PUCK800	08014396				

Process

Process ID: 1.25601828
Process Time: 2010.02.26 18:58:13
Process Stand: SSCB-CONFIG1@SSCB

Sensor

D1: 0
D2: 1
DFQ1: 0
DFQ2: 0
DT: 4.49
DTG: 0
Density Meter Factor: 1
Density Press Comp Factor: 0
FCF: 3.3711
FD: 1600
FFQ: 0
FT: 4.88
FTG: 0
Flow Cal Pressure PSI: 0
Flow Press Comp Factor: 0
K1: 6716.993
K2: 7364.347
Mass Flow Meter Factor: 1
Temperature Cal Factor: 1.00000T.00000
Volume Flow Meter Factor: 1

Units

Density Unit: G/CUCM
Mass Flow Unit: KG/MIN
Pressure Unit: PSI
Special Mass Base Unit: GRAM
Special Mass Conv Factor: 1
Special Mass Flow Text: NONE
Special Mass Time Unit: SEC
Special Mass Total Text: NONE

Units

Special Volume Base Unit: LITER
Special Volume Conv Factor: 1
Special Volume Flow Text: NONE
Special Volume Time Unit: SEC
Special Volume Total Text: NONE
Temperature Unit: DEGC
Volume Flow Unit: L/MIN

MVD Channel Assignments

Channel A Assignment: ANALOG 1
Channel B Assignment: ANALOG 2
Channel B Power: INTERNAL
Channel C Assignment: FREQUENCY 1
Channel C Power: INTERNAL

Assignments

Discrete Input 1 Assignment: NONE
Discrete Output 1 Assignment: FAULT
Discrete Output 2 Assignment: FAULT
Event 1 Variable: DENSITY
Event 2 Variable: DENSITY
Frequency Scaling Method: FREQUENCY/FLOW
Frequency Variable: MASS FLOW RATE
mA1 Variable: MASS FLOW RATE
mA2 Variable: DENSITY

Ranges

Event 1 Setpoint: 0
Event 1 Type: LOW ALARM
Event 2 Setpoint: 0
Event 2 Type: LOW ALARM
Frequency Active State: ACTIVE HIGH

Micro Motion, Inc. **Transmitter Configuration Report** **2294829634**

Ranges

Frequency Hertz:1000
Frequency Output Mode:SINGLE
Frequency Pulses/Unit:11111.11
Frequency Rate:5.4
Frequency Units/Pulse:9.000001E-5
mA1 LRV:0
mA1 URV:5.4
mA2 LRV:0
mA2 URV:5

Faults

Frequency Fault Behavior:DOWNSCALE
Frequency Fault Value:15000
mA1 Fault Behavior:DOWNSCALE
mA1 Fault Value:2
mA2 Fault Behavior:DOWNSCALE
mA2 Fault Value:2

Other

Calibration Process ID:1.25601285
Core Software Rev:36
Density Cutoff:0.2
Density Damping:0.8
Density High Limit:5
Density Low Limit:0
Direction:FORWARD
Fault Dwell Time:0
Feature Key:0
Flow Damping:0.8
HART Device ID:8290634
Mass Flow Cutoff:0.001
Pressure Comp Line Pressure:0
Pressure Compensation State:OFF
RS485 Baud:9600
RS485 Parity:ODD
RS485 Protocol:MODBUS RTU
Slug Duration:0
Tag:
Temperature Damping:2.4
Transmitter Software Rev:60

Other

Volume Flow Cutoff:0