

HW: Controller Area Network (CAN) data analysis.

NAME:

April 11, 2018

1 Objectives

The objective of this homework is to familiarize students with CAN data and analyze them using Excel, MatLab and Google Earth.

2 Introduction

Modern vehicles including cars, trucks/buses as well as off-road equipment have a computer network that serves to manage the drive train, interaction with the operator, as well as to communicate with implements. We can tap into this computer network and record a host of messages that contain data that we are interested in during operations.

The data we are going to analyze were collected during a tillage operation with a John Deere tractor in spring 2016 using a Vector CAN data interface. The Vector unit was programmed to record data such as GPS location, Engine torque, RPM, as well as Wheel Speed, Ground Speed and Fuel consumption. Vector provides a tool that allow us to translate the proprietary Vector (xlsx) format into an Excel readable format. The identifier and the data are in hexadecimal format, so we need to convert them to decimal format to extract the GPS coordinates in Latitude/ Longitude form. The Excel file that contains our data has the filename 60519009.xls. We are going to open this file in Excel, convert the strings into cells using Excel's text to column function, then we are going to extract only GPS data, we will convert them to decimal format, and copy and paste all our coordinates into a text file (either using Notepad or MatLab's editor). We will then use a small MatLab program that translates these coordinates into a kml file. This file will be read by Google Earth which will show our GPS locations graphically.

3 Procedures

The standard CAN message that shows the GPS location of a vehicle, is as follows: Note that the identifier is given in hexadecimal format. You can always convert the hexadecimal number into a decimal number, but the hexadecimal notation is far more compact and

elegant. As you can see from the definition, the GPS location has the identifier CFEF31C, and it contains 8 bytes (that is 64 bits). The first 4 bytes represent the Longitude and the second 4 bytes the Latitude. After we convert the Lat and Lon data to decimal, we need to multiply that decimal number by a scaling factor of 10^{-7} and add an offset of -210. Since this field is in central Illinois, the latitude always starts with 40 and the longitude with -88 (going west from Greenwich in the UK yields a negative number).

- CFEF31C – Vehicle Position

- Latitude

Bit Start: 0

Bits: 32

Scaling Factor: .0000001

Offset: -210

- Longitude

Bit Start: 32

Bits: 32

Scaling Factor: .0000001

Offset: -210

1. First open the CAN data file 60519009.xls in Excel format (answer yes on the question whether you want to open it).

Select column A and now perform a text to column operation using tab, space and the colon (special character as delimiters), see Figure 1. Be patient, for a large datafile like this it will take a while to do this.

If you did it right now column L should contain the CAN message identifiers. Now you want to sort everything on column L so we can select only messages that contain GPS data, which have the identifier CFEF31C. MAKE SURE to select ALL columns before you sort on column L, otherwise the GPS data and the identifier are no longer matched.

2. Remove all records (rows) that do not have the identifier CFEF31C. If all went well you should have 14,261 records.
3. Save your file as "CANDataAnalysis_YourName.xlsx" in excel format (not text).
4. Now we are going to extract the data from the GPS records. The standard says that the Latitude is contained in bytes 1-4 (that is 32 bits) of the datastring, and that the Longitude is contained in bytes 5-8. If you look in the datafile you will see not only numbers but also letters. This is because the data is given in the hexadecimal format so we need to first convert it into decimal format. We do this in a couple of steps: Firstly, we will convert all hexadecimal data into decimal data using Excel's hex2dec

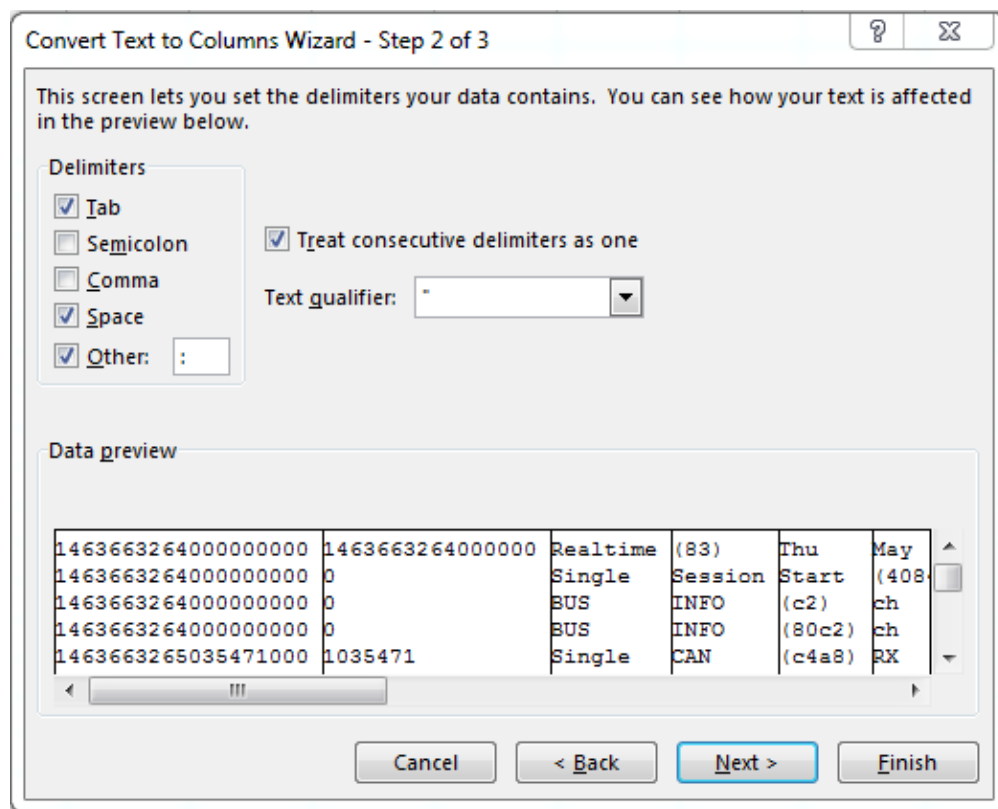


Figure 1: Text To Column Delimiters choice in Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1	1.46E+18	10184 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	18			72	151	144	18	149	14	19	24	40.07275760	-88.21076300
2	1.46E+18	10128 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	18			72	151	144	18	149	14	19	24	40.07275760	-88.21076300
3	1.46E+18	10181 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	18			72	151	144	18	149	14	19	24	40.07275760	-88.21076300
4	1.46E+18	10105 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	18			72	151	144	18	149	14	19	24	40.07275760	-88.21076300
5	1.46E+18	10071 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	18			72	151	144	18	149	14	19	24	40.07275760	-88.21076300
6	1.46E+18	10224 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	19			72	151	144	18	149	14	19	25	40.07275770	-88.21076300
7	1.46E+18	10016 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	19			72	151	144	18	149	14	19	25	40.07275770	-88.21076300
8	1.46E+18	9997 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	13	95	0e	13	19			72	151	144	19	149	14	19	25	40.07275770	-88.21076290
9	1.46E+18	10020 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	19			72	151	144	18	149	14	19	25	40.07275770	-88.21076300
10	1.46E+18	10014 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	19			72	151	144	18	149	14	19	25	40.07275770	-88.21076300
11	1.46E+18	10185 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	13	95	0e	13	19			72	151	144	19	149	14	19	25	40.07275770	-88.21076290
12	1.46E+18	9943 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	13	95	0e	13	19			72	151	144	19	149	14	19	25	40.07275770	-88.21076290
13	1.46E+18	9878 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	13	95	0e	13	19			72	151	144	19	149	14	19	25	40.07275770	-88.21076290
14	1.46E+18	9881 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	13	95	0e	13	19			72	151	144	19	149	14	19	25	40.07275770	-88.21076290
15	1.46E+18	9881 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	12	95	0e	13	19			72	151	144	18	149	14	19	25	40.07275770	-88.21076300
16	1.46E+18	9439 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	11	95	0e	13	19			72	151	144	17	149	14	19	25	40.07275770	-88.21076310
17	1.46E+18	9626 On/Off	CAN	(4a8)	RX	ch	1	Dlc	8	idx	cfef31c	data	48	97	90	11	95	0e	13	19			72	151	144	17	149	14	19	25	40.07275770	-88.21076310

Figure 2: This is what your spreadsheet should look like when you are done with it.

function. Start with the first record and in column W type =hex2dec(N1) copy this over to all 7 cells to the right of it.

Now we need to calculate the Latitude and the Longitude from the decimal data records. The basis of our decimal data is 256. Therefore the Latitude will be (assuming the data is in columns W-AD, see Figure 2):

Cell AE

$$= (AD1 + AC1 * 256 + AB1 * 256^2 + AA1 * 256^3) * 0.0000001 - 210$$

The factor 10^{-7} is the scaling factor and the -210 is the offset, see the definition of the Vehicle position in the standard.

Do the same thing for Cell AF:

$$= (Z1 + Y1 * 256 + X1 * 256^2 + W1 * 256^3) * 0.0000001 - 210$$

Copy cells W1 through AF1 over all the way to the last record in the file. Now you should have two columns one for the Latitude column AE () and one for the Longitude (column AF) containing 14,261 GPS coordinates. ALL GPS coordinates should start with 40 for the latitude and -88 for the longitude records.

5. Copy ALL GPS coordinates (that is two columns and 14,261 rows), start Matlab, and open its editor (type edit). Now paste these two columns into a new file and save that file as GPSdataDemo.txt, **make sure you save it as a text file, not a m-file.**
6. Type in (or copy from the tex file) the code shown below. Save this file as CANData-Analysis.m (with extension m, since it is an executable MatLab script, make sure it is saved in the same folder where your GPSdataDemo.txt file resides). The program will load your GPSdataDemo.txt file and write the GPS coordinates in kml format which Google Earth can read.

```

1      close all
2      clear all
3      clc
4

```

```
5     KML_filename = 'GPSdataDemo.kml';
6     load GPSdataDemo.txt;
7
8     FieldPlots    = struct('lat', GPSdataDemo(:,1), 'lon', ...
9                           GPSdataDemo(:,2));
10    kmlwritepoint([KML_filename], FieldPlots.lat, FieldPlots.lon);
```

7. Find your kml file and double click on it. This will open Google Earth, and if all went well, show the exact coordinates where the machine traveled. If MatLab throws an error in the kmlwritepoint line, that means that the Mapping toolbox is not installed, in that case finish this homework in the computer lab (220).

4 Submission

Add a screenshot of the output that you saw in Google Earth to this file. Given in the tex file is the code to add a figure commented out, save the screenshot as AgEngineeringFarmGPS_GoogleEarth.png.