TECH/OPS

SEVCON

Gen4 Application Note

Title	App note - Graphing Vehicle Behaviour using the DVT
Filename	App Note - Graphing Vehicle Behaviour using the DVT.docx
Date of Creation	27/11/2008 10:04:00 by Paul Shipley
Last Updated	08/01/2009 16:33:00 by Paul Shipley
Revision	3

Contents

1. DVT Graphing Basics 2
2. Initial Controller Configuration 3
3. Using the Vehicle IO Interface 4
4. Identifying Problems 9
5. Advanced Techniques 10

Sevcon Ltd
TVTE
Gateshead
Tyne & Wear
England NE11 0QA

Tel +44 (0)191 4979000
Fax +44 (0)191 4824223
www.sevcon.com

1. DVT Graphing Basics

The ability to produce high resolution graphs using the DVT is indispensable when trying to track down odd lumps and bumps that can occur during drive.

Quite often, especially during the early stages of setting up a vehicle, comments such as "I can feel a bump at this point," or "is the speed constant here," will be made. The only way to scientifically diagnose and fix these problems is to produce a graph of the behaviour and study it to find the root cause.

The DVT comes with a powerful logging tool known as the Vehicle IO interface. Even on its own it can be used to monitor a large number of internal values in real time. But when this data is logged to file, an Excel macro can produce a detailed graph of almost any situation in seconds.

1.1. Prerequisites

In order to take full advantage of the DVT graphing functionality you will need the following:

- IXXAT CAN-USB interface and drivers installed
- DVT (see application note "Installing the DVT")
- Vehicle IO Interface log viewed – Excel macro available from Team Coherence under //Software/tools/veh_if_viewer

2. Initial Controller Configuration

DVT graphing works by configuring TPDOs on the target espAC system. The TDPOs are configured in such that certain pieces of interesting drive data are transmitted automatically by the controller at regular intervals, which can be as fast as around every 10 or 20ms. This is far faster than having the DVT request each piece of data individually. This also reduces the load on the CANbus.

The DVT comes with a script that will automatically configure the TPDOs on the target system to regularly transmit the most commonly monitored values. An in-depth knowledge of how PDOs work is not necessary, but will be an advantage in the following steps.

First, you should establish communications between the DVT and the controller. Start the DVT, and ensure you can see CAN messages from the controller scrolling up the CAN pane of the DVT window. You may need to change the baud rate for this to work – a list of available baud rates is available in the CAN menu.

With communications established, use the following commands[1] to log in to the controller, and request the pre-operational state. It is necessary to go to pre-operational in order to change and PDO configuration.

```
login 1
fpo 1 PRE
```

The line contactor should drop out if it was closed previously. If it does not, cycle the key switch as a sub-system may have been inhibiting pre-operational. The line contactor should not close.

Next, configure a TPDO for any object that you wish to monitor. TPDOs can be mapped in using the DVT's map_in_tpdo command[2]. The syntax is as follows:

```
map_in_tpdo <node-id> <index> <sub-index>
```

As an example, you can choose from the following list of items that are commonly graphed for debug purposes:

```
map_in_tpdo 1 0x4600 5     ;# Id reference
map_in_tpdo 1 0x4600 6     ;# Iq reference
map_in_tpdo 1 0x4600 7     ;# Id actual
map_in_tpdo 1 0x4600 8     ;# Iq actual
map_in_tpdo 1 0x4600 9     ;# Ud
map_in_tpdo 1 0x4600 10    ;# Uq
map_in_tpdo 1 0x4600 11    ;# Mod index

map_in_tpdo 1 0x5100 1     ;# Battery voltage
map_in_tpdo 1 0x5100 3     ;# Capacitor voltage

map_in_tpdo 1 0x2020 3     ;# Target speed (speed mode) / speed limit (torque mode)
map_in_tpdo 1 0x2020 5     ;# Torque limit (speed mode) / target torque (torque mode)

map_in_tpdo 1 0x606C 0     ;# Actual speed
map_in_tpdo 1 0x6077 0     ;# Actual torque
```

To go back to the operational state, enter the following command:

```
fpo 1 OP
```

Cycle the key switch. The line contactor should close and the controller is now configured with the default TPDOs used for graphing.

[1] DVT commands in this document will assume the target node is configured with node ID 1. The commands should be modified as appropriate if necessary.

[2] More information on the map_in_tpdo command can be found at http://intranet.sevcon.net/Functions/engineering/Engineering%20Wiki/map_in_tpdo.aspx.

3. Using the Vehicle IO Interface

The vehicle IO interface is the main window in which values may be monitored and logged. The interface can be accessed by clicking on the vehicle IO interface button on the toolbar of the DVT.

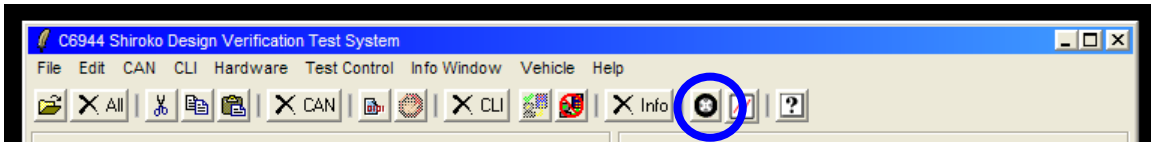

Figure 1 - Image of DVT toolbar showing location of vehicle IO interface button.

3.1. Monitoring Data

The Vehicle IO Interface window will continually update whenever information is being received from the controller. This means that any time the controller is in the operational state and it is sending TPDOs, the interface window will be updated.

The latest version of the vehicle interface will auomatically configure itself for the most commonly graphed items. It reads the TPDO setup from selected nodes and from this it is able to convert the PDO data into viewable information.

By default, the script searches for the presence of nodes 1, 2 and 3. However, this can be changed by modifying the foreach statement in the file veh_if_cfg.tcl.

The file object_data.tcl also contains a large array of object dictionary items that are commonly mapped into TPDOs for graphing. Most items should be covered already, but if you need to add extra ones to graph data that is not visible, then extra lines can be added by following the same format. An object dictionary index and subindex, name and scaling factor should be provided. The final value is either a 1 or 0 and will enable or disable graphing of this item accordingly.

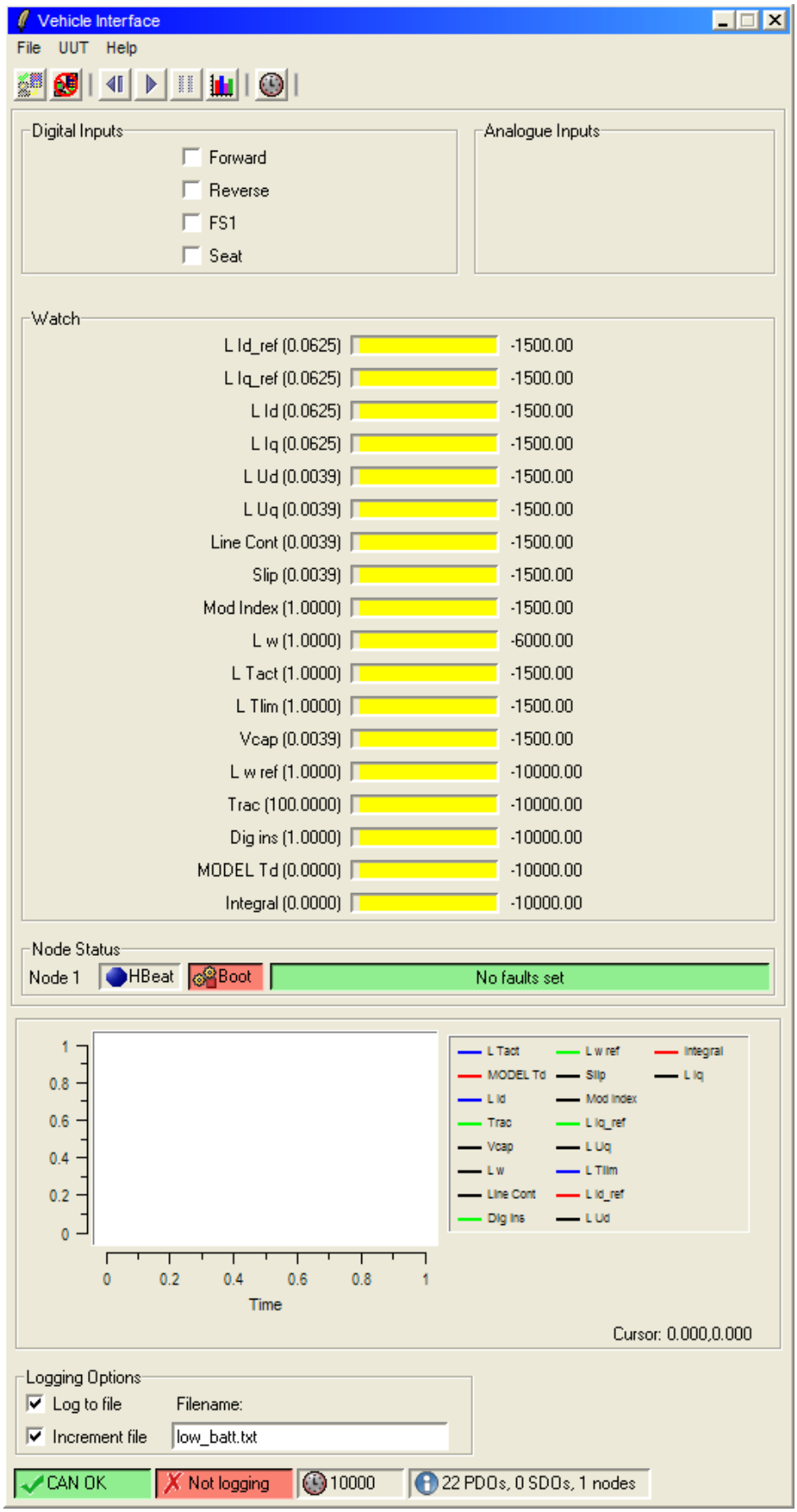

Figure 2 - Screenshot of the Vehicle IO Interface window

If you need to graph any other parameters then these should be mapped in to TPDOs on the controller using the map_in_tpdo command. Each time the graphing window is opened, the DVT will check what TPDOs have been set up on the controller and will graph them accordingly. Simply creating a TPDO on the controller will be enough for them to be shown on the graph. See the PDO fundamentals app note for more information on setting up TPDOs.

3.2. Logging Data

In order to log data to a file, you must first enable the logging option and assign a file name. This information should be entered into the Vehicle IO Interface window.

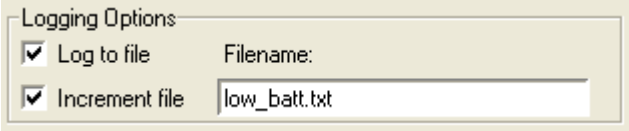

Figure 3 - Logging options in the Vehicle IO Interface window

Ensure the log to file box is checked. Checking the increment file box will append the date and time onto the filename you enter (but before the .txt extension) so each log has a unique filename. The filename can be a text string of your choice.

Figure 4 - Logging control buttons in the Vehicle IO Interface window

Logging is controlled by 4 toolbar icons at the top of the Vehicle IO Interface window. At start up, they will appear as they do above, with the 3rd button disabled. Operation is as follows:

- Click the first button to reset the logging functions.
- Click the second button to begin logging. Once this is clicked, the monitored values are now being stored to file. Perform and manoeuvres you wish to analyse in detail. The third button will also become active at this point.
- Click the third button to end logging. The log file will remain stored in the DVT/veh_if_log directory.
- Clicking the forth button will produce a small preview graph in the Vehicle IO Interface window. You can use the mouse to zoom into particular areas and move around. Clicking the graph with the right mouse button will zoom out to the start.

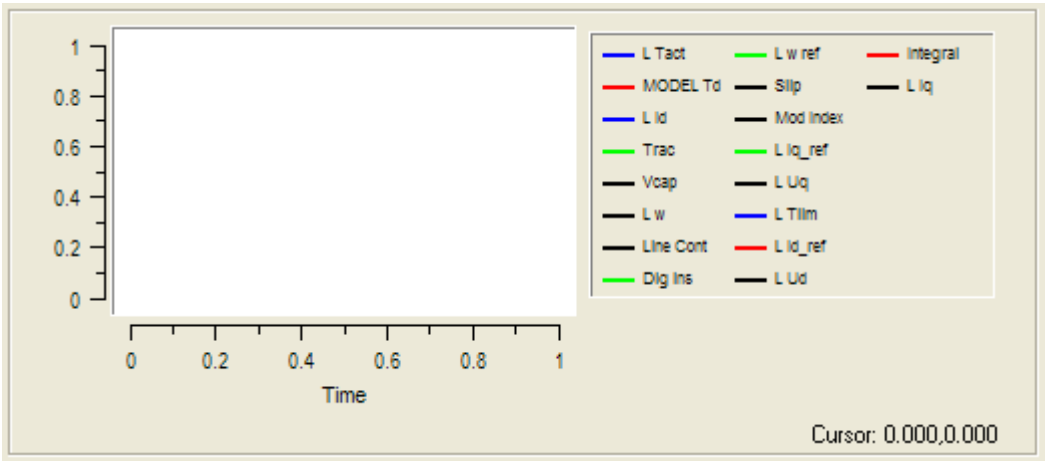

Figure 5 - Graphing pane of the Vehicle IO Interface

3.3. Producing Graphs in Excel

Although the graphing pane of the Vehicle IO Interface window can provide a quick preview of the log file in graph form, it is not practical to use it for any serious data analysis. Much better results can be achieved by importing the log file into Excel.

A spreadsheet containing a macro designed to read log files produced by the DVT and output graphs is available from Team Coherence. It is stored in the //Software/tools/veh_if_viewer directory.

To use the viewer, get the file to your local hard disk. Open the spreadsheet. Please note, this spreadsheet relies on macros being executable, and these should be enabled if requested. You should see a screen similar to that shown below:

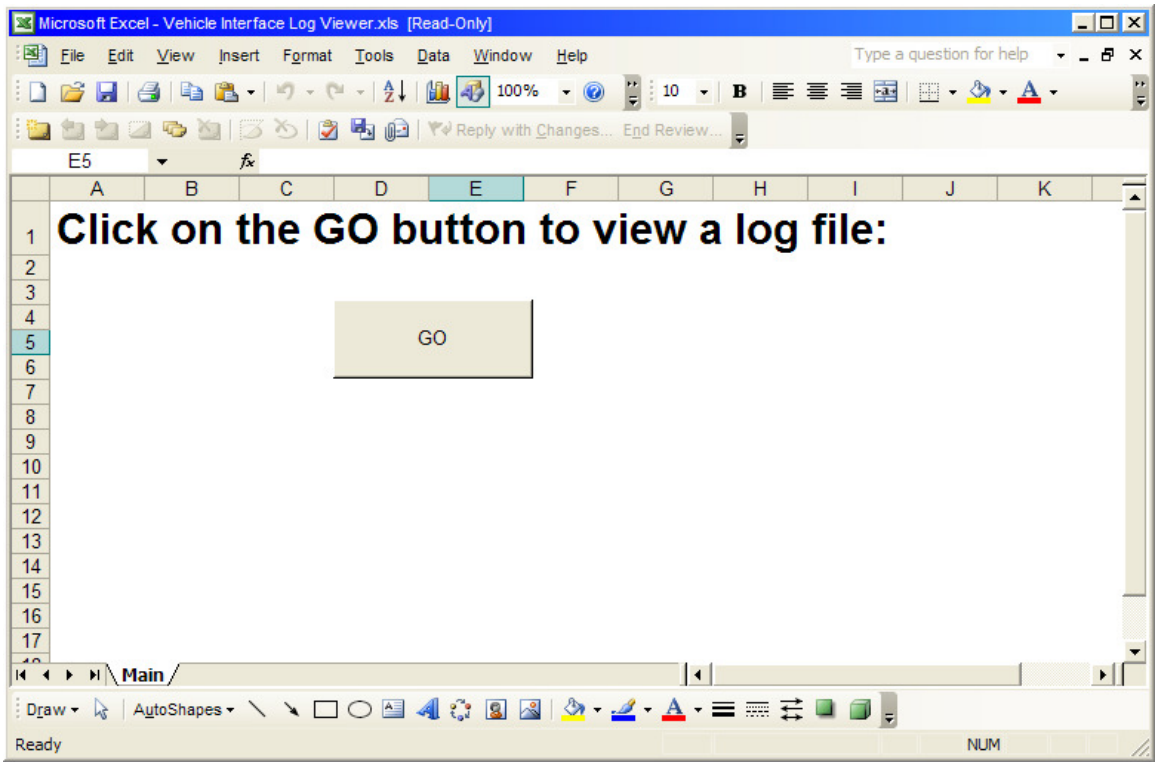

Figure 6 - Screenshot of the Vehicle IO Interface log viewer

To view a log file, click on the button labelled GO. You will be prompted for the filename of the log file you wish to view. By default, the log files will be stored in the following directory:

```
C:\C6944\DVT\veh_if_log\
```

On selecting the file, the macro will produce a graph of the logged data:

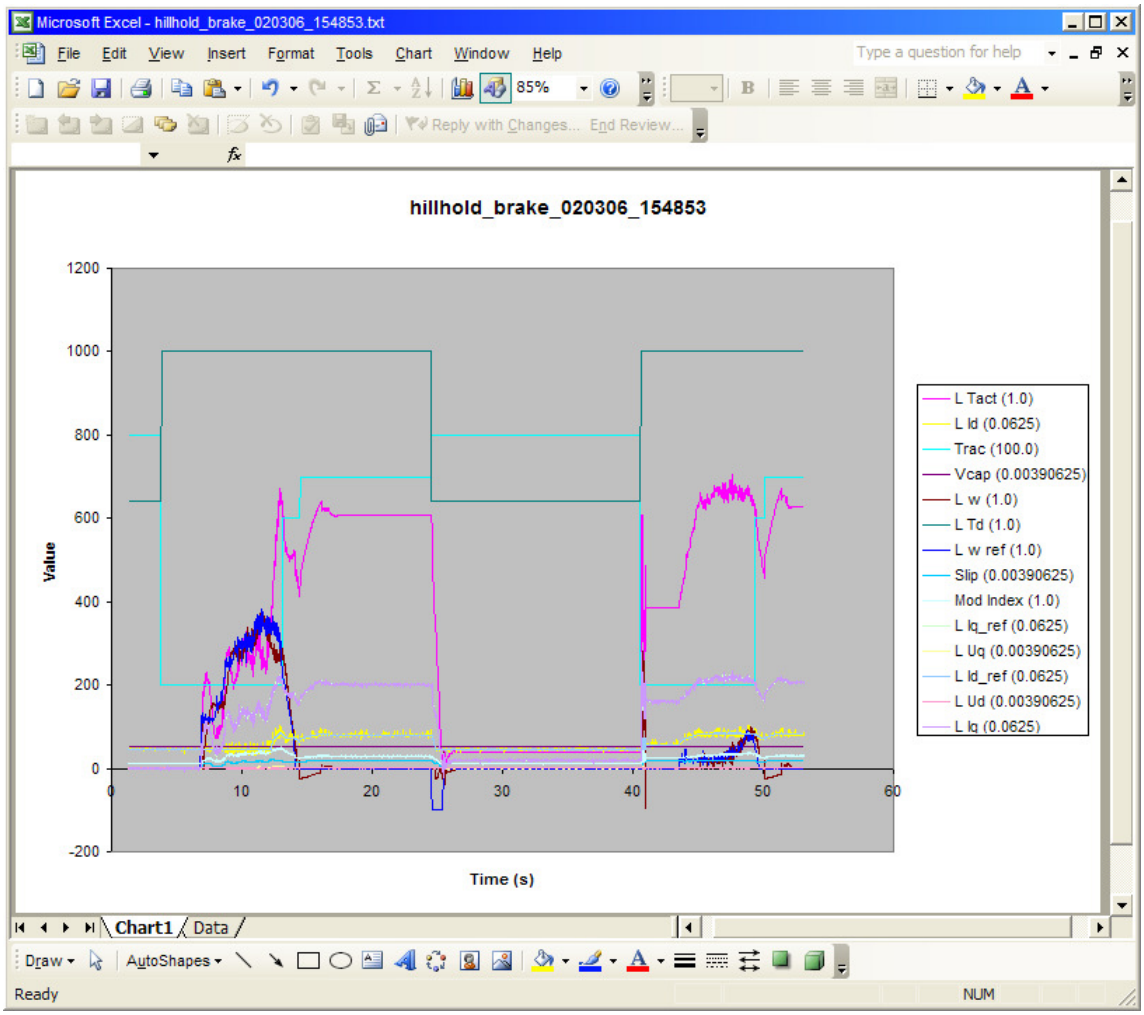

Figure 7 - Screenshot of log file converted to graph format

You can view the data in large scale, and zoom into specific regions of interest by changing the scaling values of the axis.

Because of the scaling, you may find that you need to zoom in to some small details. For instance, in the graph above, the largest values are speed demands and traction states, which peak at 1000. However, if you wanted to look at currents, which peak at around 100, you would need to zoom into that region quite closely.

4. Identifying Problems

Graphs are usually produced as a means of tracking down the cause of some erroneous or unexpected vehicle behaviour. Things to check include:

- Is the actual vehicle speed following the target speed correctly?
- Does the actual vehicle speed seem accurate (no unexpected spikes)?
- Are the d and q-axis currents being controlled correctly?

5. Advanced Techniques

5.1. Changing the Sample Rate

Logged data is captured as data is sent out by the controller. Therefore, in order to change the sample rate you must change the frequency at which the controller transmits its TPDOs.

The default configuration script sets up the controller to send the TPDOs on every SYNC message. SYNC messages are transmitted once every 20ms, giving a 20ms sample rate.

The simplest way to change the sample rate is to modify the SYNC message frequency, which can be done by writing to object 0x1006. However, it is worth bearing in mind the more data you try to transmit causes extra load on the host processor, which may cause other problems. The frequency of the CANopen task is 5ms, so data cannot be transmitted faster than this. For most applications, 20ms provides enough resolution.