Date:	2017-08-07
Author:	Li Wan, CarMaker Service Team Germany
Release No.:	CarMaker 6.x

How to define the traffic object position by using a real measurements in the form of GPS coordinates

Today I want to share an example about how to define the position of a traffic-object with GPS coordinates. It was built for a request from a customer who wanted to use a real GPS measurement for developing an ADAS system. But the corresponding UAQs, the global position (GPS) of the traffic object, cannot be overwritten by DVA command. Therefore it is necessary to convert the GPS measurement in the global position of the traffic object in the frame Fr0, which can be manipulated via DVA in the mode “Free Motion”. CarMaker offers the possibility to import a measurement data with the functionality “Input from File” (IFF).

Step 1:

Open the infofile “SimParameters” under the project folder “<project>/Data/Config” and enter the following command in it:

```
1:  GCS.Traffic.Active=1
```

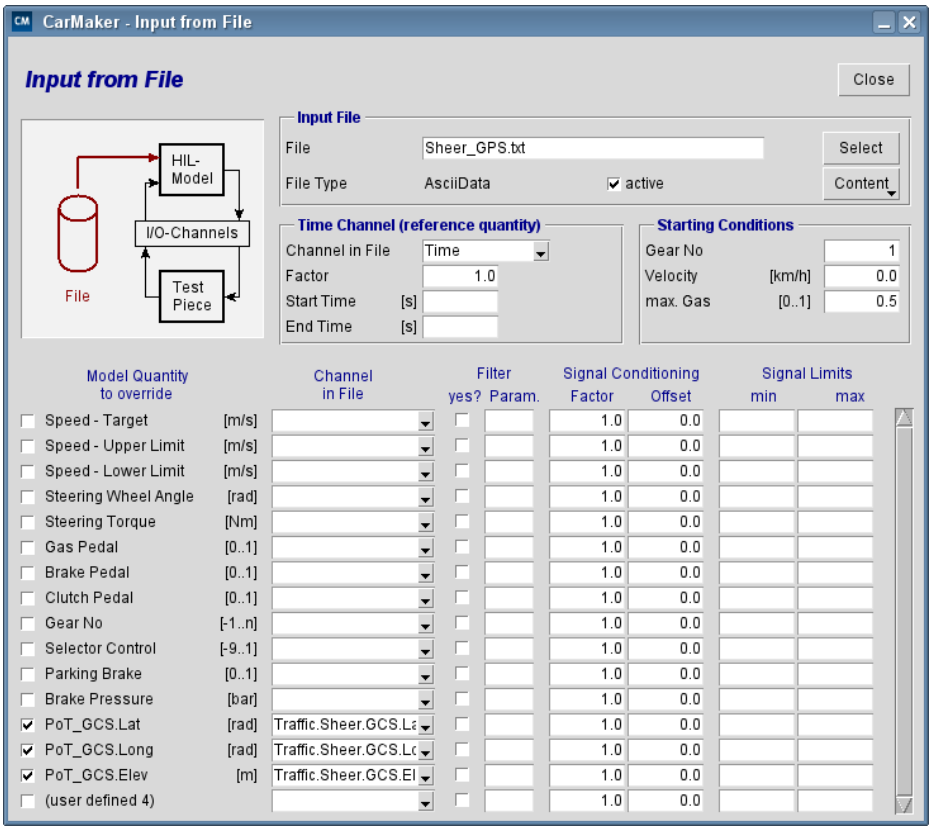
This optional parameter is used to activate the calculation of GPS coordinates for traffic objects.

Step 2:

Add a .CarMaker.tcl file including the following Tcl-code at the top level of the project folder or extend an already existing .CarMaker.tcl file with these codes:

```
1:  SimInput::AddQuInfo {
2:      User1    "PoT GCS.Lat"    "PoT GCS.Lat"      "rad"
3:      User2    "PoT GCS.Long"   "PoT GCS.Long"     "rad"
4:      User3    "PoT GCS.Elev"   "PoT GCS.Elev"     "m"
5:  }
6:  SimInput::New
7:  SetSavedData TestRun
```

It is used to extend the CarMaker GUI with additional channel names and just for the visualization inside the GUI as shown in the figure below.



Step 3:

Open the User.c file in the src folder of your project directory “<project>/src”.

Add the following inclusion and declaration to create three C-Code variables at the beginning of the User.c file:

```
1:  #define GPS
2:  #if defined (GPS)
3:      #include <GCS.h>
4:      #include <Traffic.h>
5:      tGCSPos PoT GCS;
6:  #endif
```

The struct tGCSPos is defined in the header file GCS.h under the installation folder „.../ipg/hil/<version>/include/GCS.h“.

In the function User_DeclQuants (void), add the following lines to create three new UAQs:

```
7:  # if defined (GPS)
8:      DDefDouble4 (NULL, "MyGPS.Lat", "rad", &PoT GCS.Lat, DVA None);
9:      DDefDouble4 (NULL, "MyGPS.Long", "rad", &PoT GCS.Long, DVA None);
10:     DDefDouble4 (NULL, "MyGPS.Elev", "m", &PoT GCS.Elev, DVA None);
11: #endif
```

In the function User_In (const unsigned CycleNo), add the following lines:

```
12: #if defined (GPS)
13:     if (ExtInp File Enabled) {
14:         if (ExtInp IF.User1.Enabled)
15:             PoT GCS.Lat = ExtInp IF.User1.Value;
16:         if (ExtInp IF.User2.Enabled);
17:             PoT GCS.Long = ExtInp IF.User2.Value;
18:         if (ExtInp IF.User3.Enabled);
19:             PoT GCS.Elev = ExtInp IF.User3.Value;
20:     }
21: #endif
```

The C-Code signals for the input channels are ExtInp_IF.User1.Value, ExtInp_IF.User2.Value,.. and so on. In this example they are mapped to the user defined quantities "PoT_GCS.XXX".

In the function User_Traffic_Calc (double dt), add the following lines:

```
22: #if defined (GPS)
23:     GCS ConvGCStoFr0 (&PoT GCS, Traffic GetByTrfId(0)->t 0);
24: #endif
```

The GPS coordinates "PoT_GCS.XXX" will be translated to the cartesian coordinates of the first traffic-object in Fr0 with the function „GCS_ConvGCStoFr0“ which is defined in the header file GCS.h.

Step 4:

Generate a new CarMaker application and choose it after opening the Configuration/Status tab from the CarMaker main GUI.

Step 5:

Import the GPS measurement through the dialog „Input from File“(GUI:Parameters) and assign them to the corresponding channels. A corresponding simple TestRun is available in the download package.

In this example the first traffic object “Sheer” with the mode “Free Motion” will move according to the GPS measurement data “Sheer_GPS.txt”: