PTWinSim and vTAG Plugins for rFpro

Interfacing Windows Co-Simulation Environments to the rFpro Graphical Environment

The rFpro Plugins provide a way of hosting vTAG or PTWinSim applications within the rFpro environment. The plugins expose the complete rFpro API to the applications. With @Source and the rFpro block-set, from Podium Technology, the complete API can be accessed using Simulink to develop the applications. The plugins will, however, execute any applications, e.g. 3rd party applications from powertrain or ECU suppliers where the source models are not available.

An example set of applications might be:

- ECU chassis control.
- ECU engine control.
- ECU interface (taking engineering values from the models/rFpro and scaling them to the ECU needs).
- rFpro inputs – reading the rFpro values for use by other applications, such as the track surface, scoring information and driver controls.
- rFpro outputs – taking the vehicle model state and passing it to rFpro for graphical rendering.
- Vehicle model.
- Hardware inputs e.g. for driver controls via CAN or DAQ interface.
- Hardware outputs, e.g. to a steering feedback or dashboard via CAN etc.
- Interface to motion platform.

Applications can be generated in a multitude ways:

- From Simulink models using the Podium Technology @Source Simulink Coder target (vTAG or PTWinSim). No model changes are required when switching between PTWinSim and vTAG.
- From Simulink models using McLaren Applied Technologies GDE (vTAG only)
- Claytex's Simulator Library for Dymola models (vTAG or PTWinSim)
- Microsoft Visual C++, based on a supplied template.

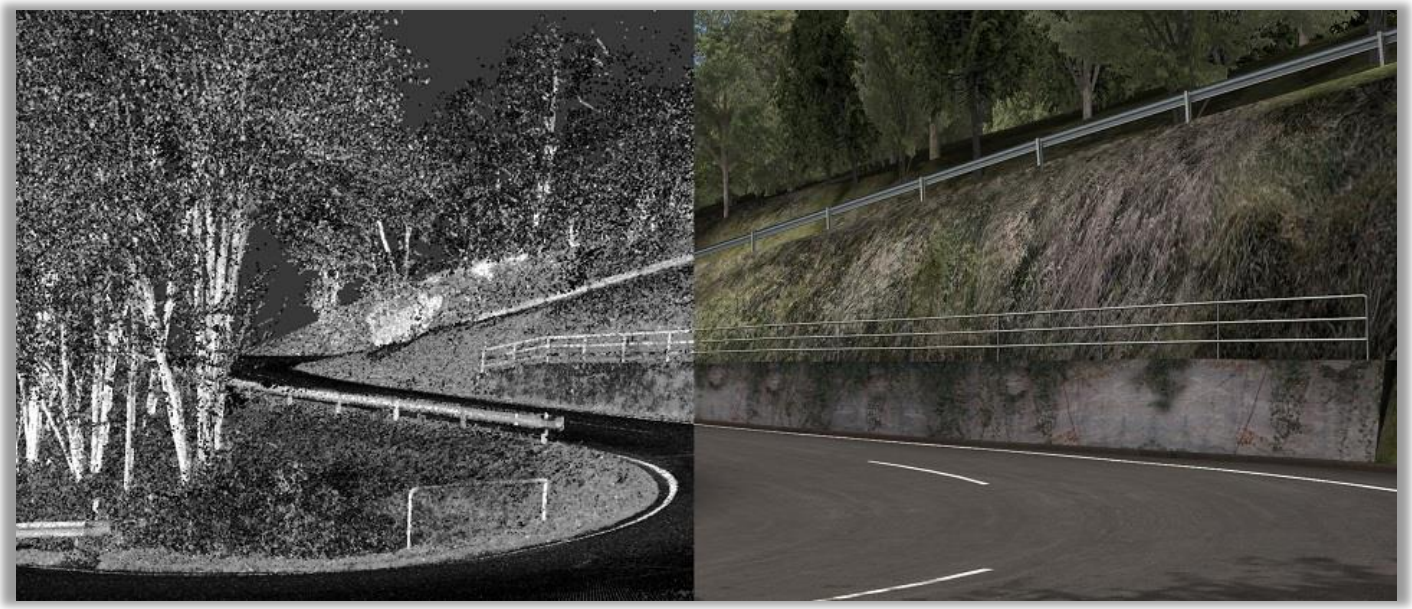
Most vehicle models can be generated as a Simulink S-Function which can be 'wrapped' in a Simulink model and compiled for vTAG or PTWinSim (e.g. CarSim, VI-Grade CarRT, SIMPACK etc.).

The advantages of using the PTWinSim or vTAG plugins, over developing bespoke rFpro plugins, are:

- Several models can be executed in this co-simulation environment. The partitioning of the functionality into separate applications greatly improves maintainability, reusability and enables responsibilities to be spread across departments or companies.
- The applications execute in the same process as rFpro, ensuring perfect synchronisation between the graphics environment and the physical modelling.
- PTWinSim and vTAG provide the basic services that are useful while executing systems such as interfaces for run-time monitoring, tuning and data logging, via MAT System Monitor/ATLAS, Magneti Marelli SYSMA, Cosworth Diablo/Toolbox or any of the standard automotive tools that utilise XCP over Ethernet (e.g. Vector CANape, ETAS Inca, ATI Vision, dSPACE ControlDesk).
- Parameterisation of the models via text files.
- Data exchanges between applications – linking the outputs of applications to the inputs of others, as well as allowing parameters to be shared between applications.

These rFpro plugins can operate in different plugin modes, depending on how they are loaded by rFpro's host plugin SimControl:

- Physics Plugin: The applications have access to the full API enabling vehicle models to be hosted.
- Hardware Plugin: This mode would be used when a 3rd Party plugin was being used for physics, a restricted API is available that allows access to the physics plugin output and other rFpro data, for use in interfacing with other systems such as hardware input and outputs or a motion platform.
- Sensor IG Plugin: A particular API is provided to allow the simulation of ADAS sensors, such as radar, lidar, ultrasonic and cameras.



The most convenient method for developing applications for PTWinSim and/or vTAG is to use Simulink with @Source and the Podium Technology rFpro Blockset. In combination, you receive:

- Access to the complete rFpro/SimControl API to bring in data to models such as terrain, driver controls, scoring, weather, as well as write back data to define the vehicle position and state.
- Access to all the call-backs triggered by rFpro/SimControl with the @Source System Call block.
- Library of default call-backs that save the rFpro data to measurement channels for display and logging.
- Sample models to demonstrate how to make best use of the API. The three models on this page demonstrate how data is extracted from rFpro for input into the vehicle model, the vehicle model and finally writing the vehicle state back to rFpro.