Multi-domain Control Software Design: from the requirements toward the production

Ing. Giovanni Azzone

Argo Tractors S.p.A.

Outlines

- The purpose of this seminar is to describe the complete process that characterizes the design of a complex multi-domain control software, starting from the requirements and going toward the software validation for production intent.

- The case of study that will be considered is a Continuous Variable Transmission tractor.

- The model-based design approach will be deeply described by focusing on its single steps and referring to real industrial application cases, showing simulation and experimental results and underlining the importance of the system modelling phase.

- The used methodologies and tools will be presented in order to show how the complete design process is led in an industrial contest.

Argo Tractors S.p.A

Argo Tractors S.p.A

Capacità produttiva approssimativa
22.000
trattori all'anno

Componenti principali prodotti internamente

1.650
Dipendenti

5
Stabilimenti Produttivi

3
Brands

5,5%
del fatturato annuo investito in
R&D

Argo Tractors è tra
i più importanti produttori globali di trattori

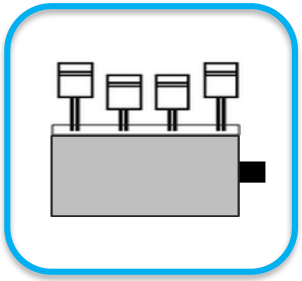
Landini McCORMICK VALPADANA Life ARGOTRACTORS

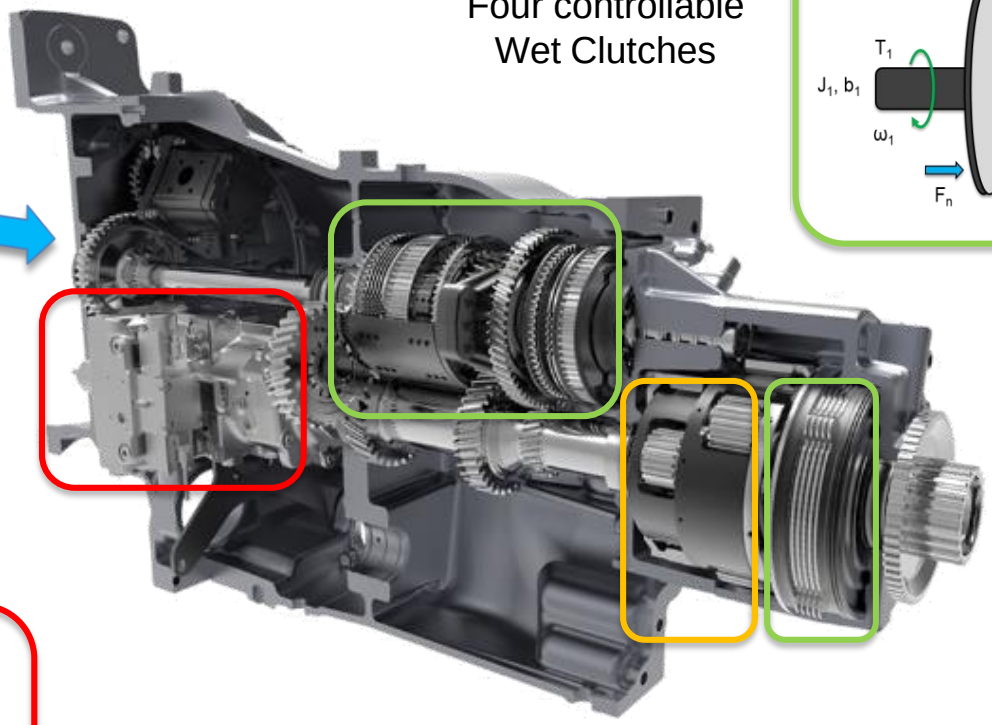
Case of study - McCormick X6 VT Drive

Continuous Variable Transmission (CVT)
Power: 121hp, 133hp, 140hp

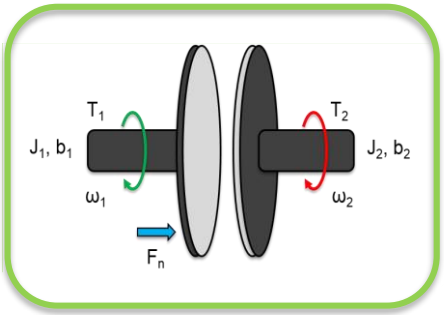
McCormick X6 VT Drive - Driveline

Combustion Engine

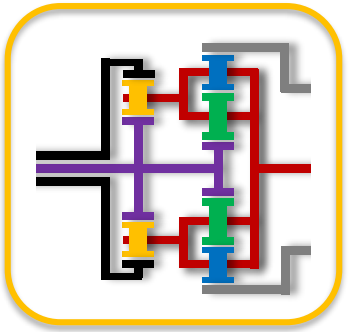


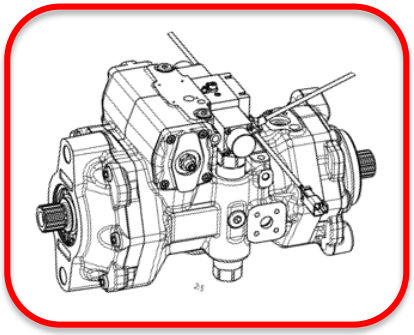


Four controllable Wet Clutches



Double stage Epicyclic Power Mixer





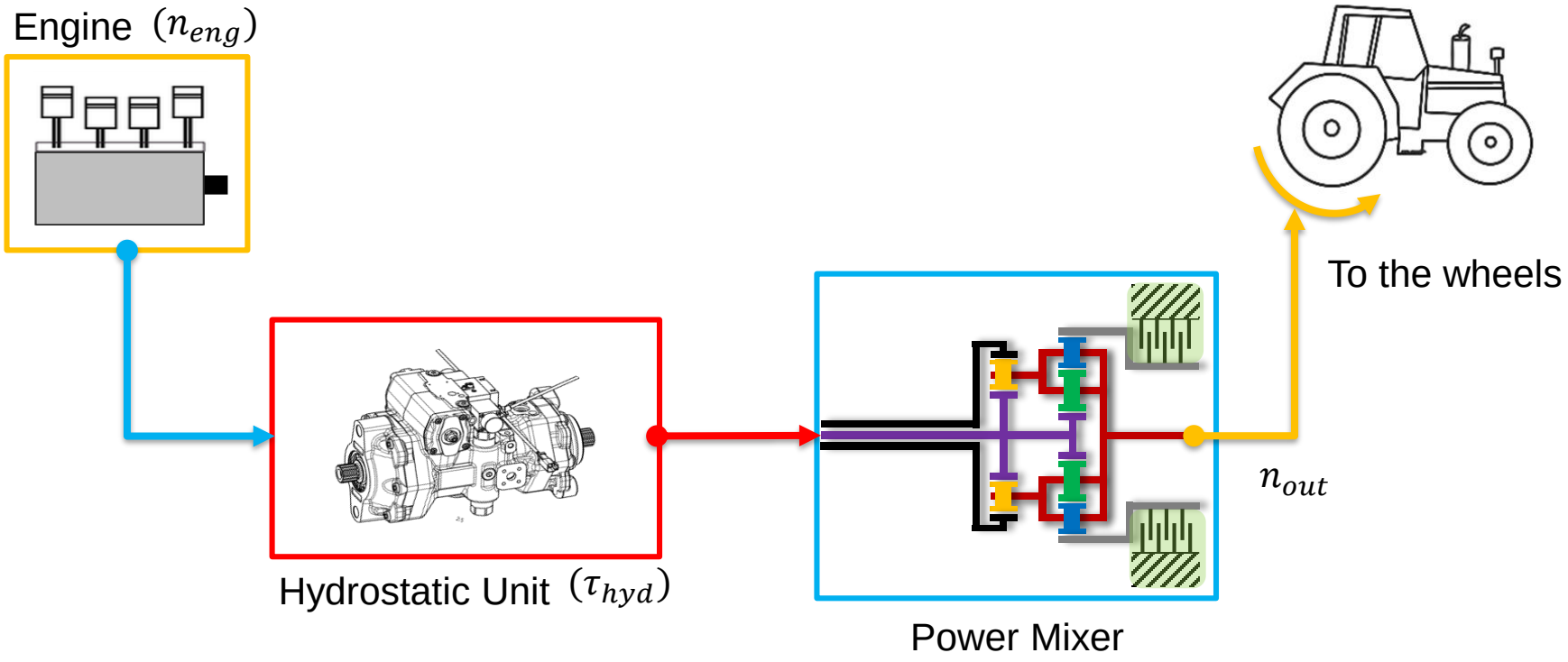
Variable displacement Pump
+ Fixed displacement Motor

McCormick X6 VT Drive - Main kinematics

Ring clutch engaged:

$$n_{out} = n_{eng}\ \tau_{hyd}\ k_1$$

Power directly supplied by the hydrostatic unit

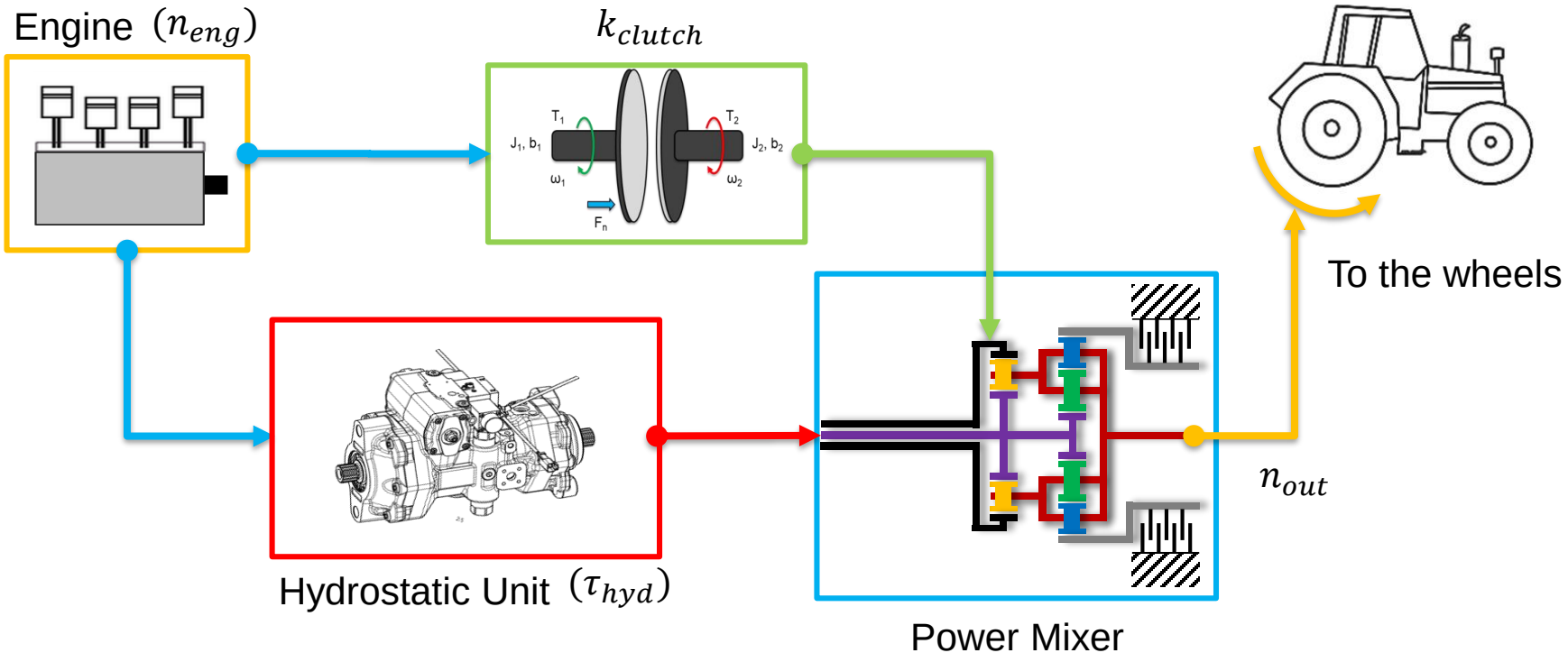


McCormick X6 VT Drive - Main kinematics

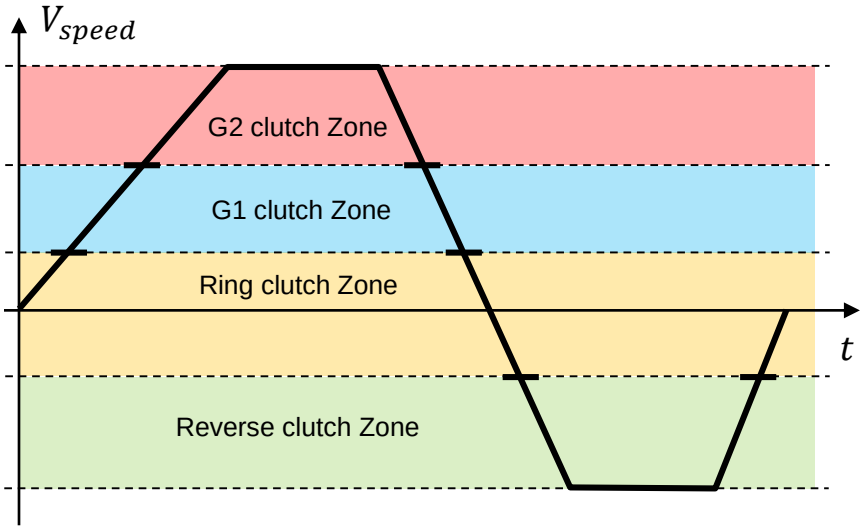
Gear 1, Gear 2 or Gear Reverse clutch engaged:

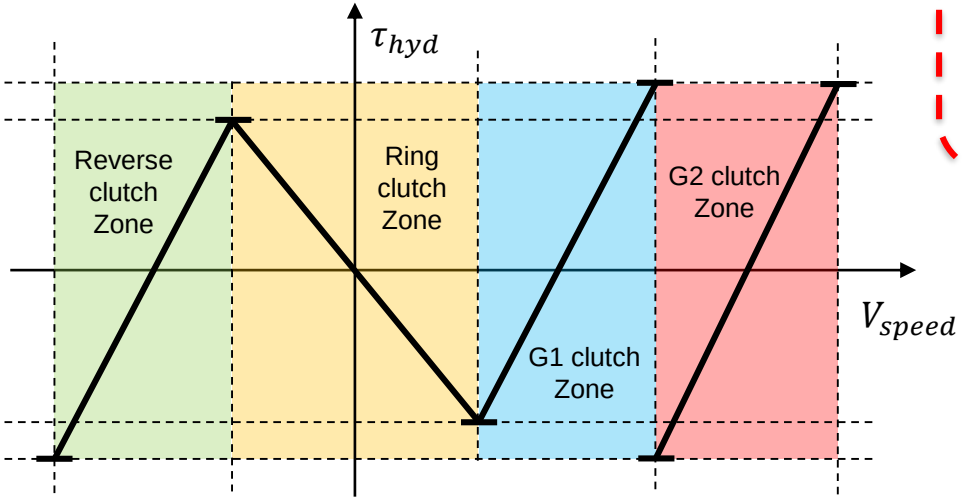
$$n_{out} = n_{eng}\ \tau_{hyd}\ k_{2a} + n_{eng}\ k_{2b}\ k_{clutch}$$

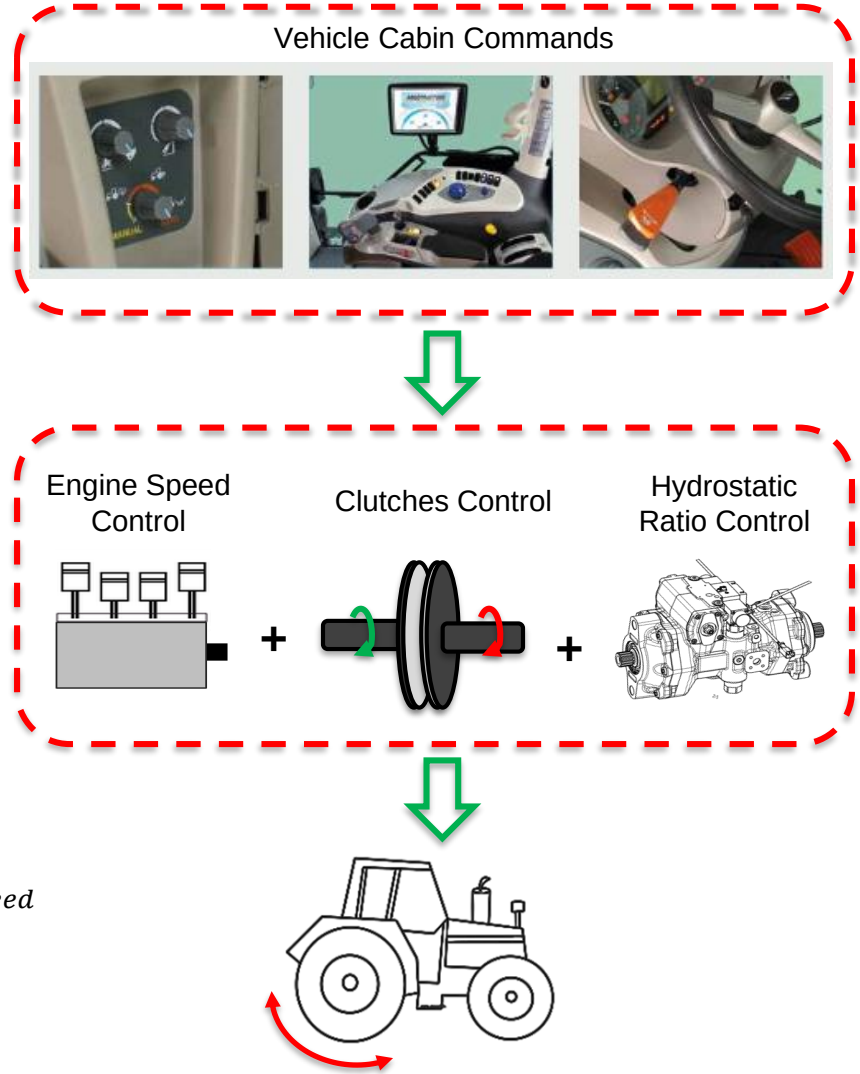
Power split between the hydrostatic unit and the engine connected clutch



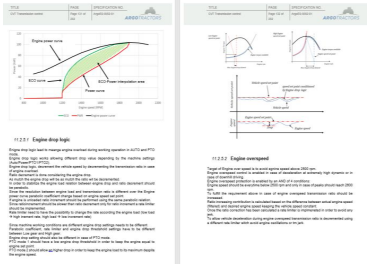
McCormick X6 VT Drive - Auto Commands







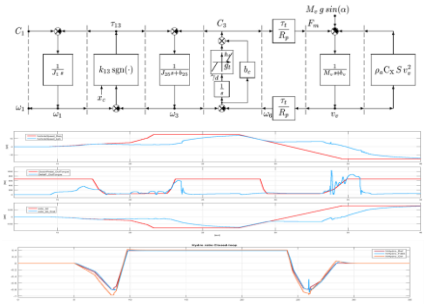
Model Based Design - Development V-Cycle



Requirements Definition

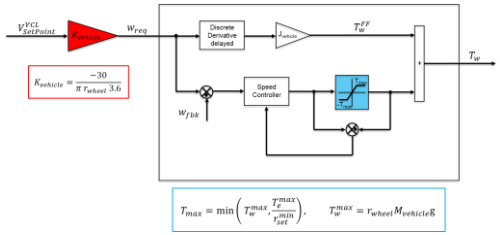
Field Test & Fine Tuning

Modeling & Simulation



SW Validation SIL, Bench, HIL

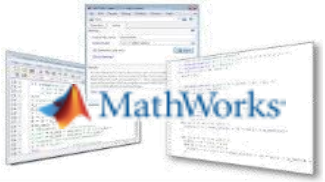
Control Design & Prototyping



Software Integration

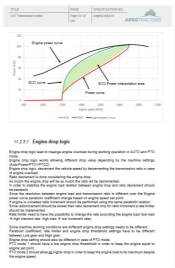
BOSCH

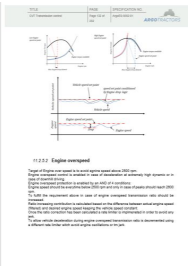
Code Generation



Landini · McCORMICK · VALPADANA · Life · ARGOTRACTORS

Model Based Design - Development V-Cycle Steps



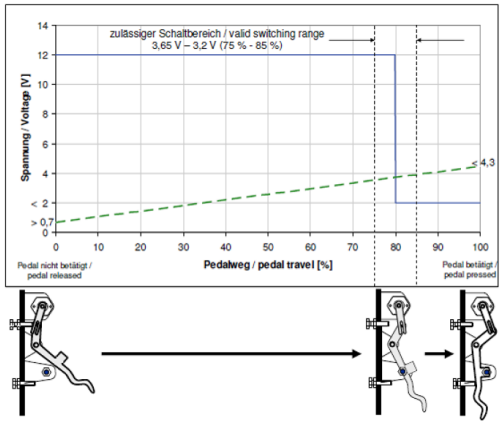


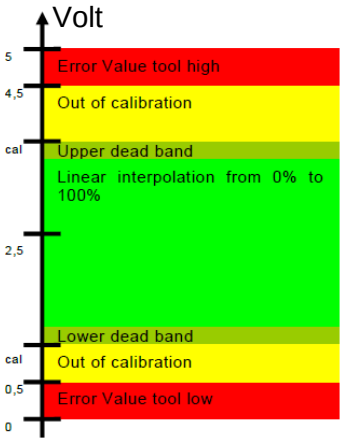
Requirements Definition

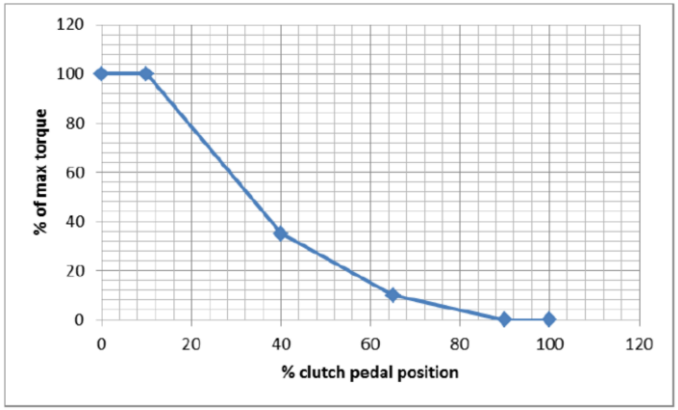
Requirements Definition

The requirements document contains the complete functional description of the system and all the technical details of the involved dynamic subsystems.

- Electrical/electronic characterizations and datasheets (sensors, actuators, etc.).

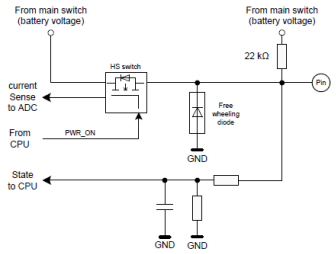






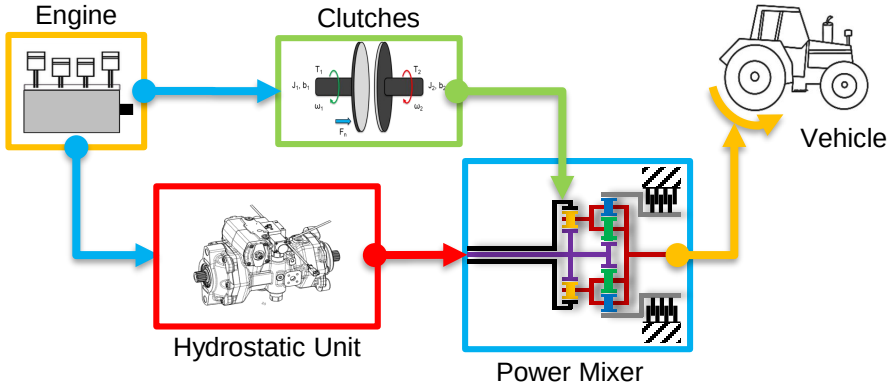
- I/O mapping and HW pinout.

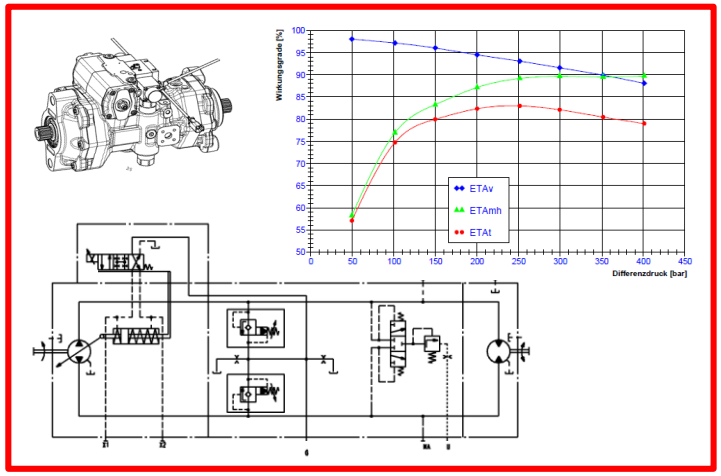
pin	function	channel	description
189	PWM HSD 2,2A	OUT 24	PWM power supply for safety switch
194	PWM HSD 2,2A	OUT 25	
193	PWM HSD 2,2A	OUT 26	
243		OUT 27	HSD park lock engage valve
241		OUT 28	HSD park lock disengage valve

Pin	Description	Main function
107, 106, 190, 189, 194, 193, 242, 256, 244, 257 	**Switching output stage with current sensing** OUT_21 to OUT_26 and OUT_29 to OUT_32	**Switching output stage**[1)] High side switch max. current 2.2 A Spark suppression diode for switching of inductive loads integrated Power supply centrally switchable via VP_2 Current measurement via ‚current sensing'

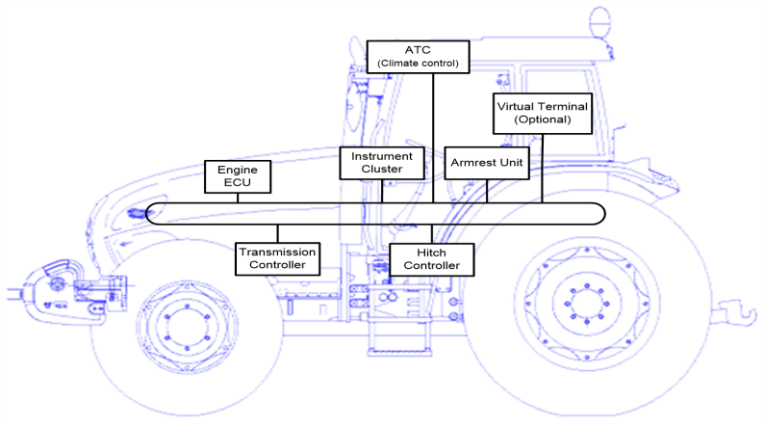
Requirements Definition

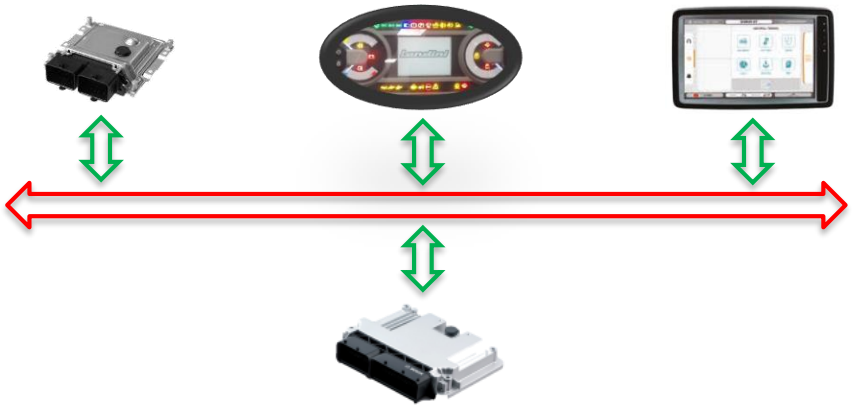
- Details and architecture of the multi-domain system.





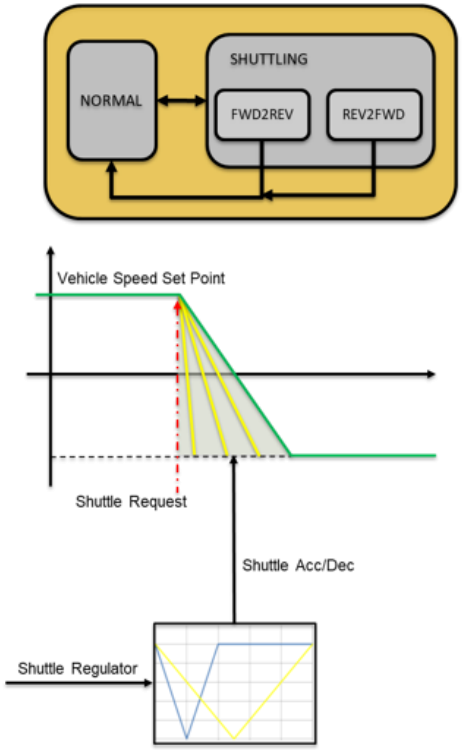
- Communication layout

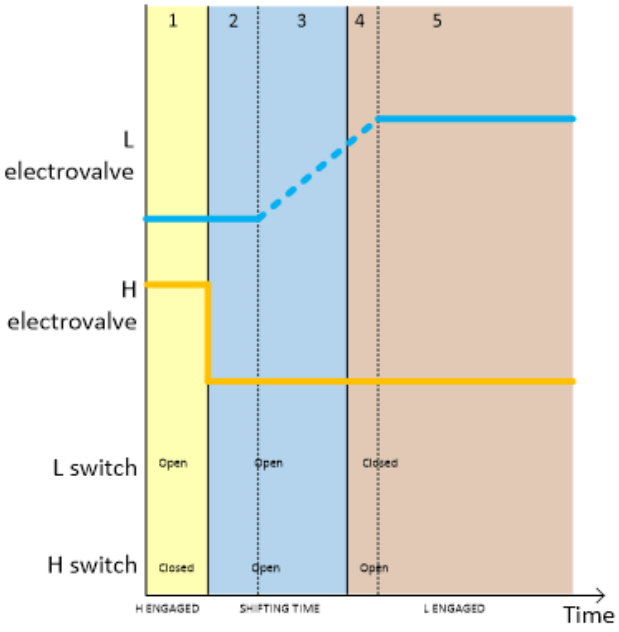




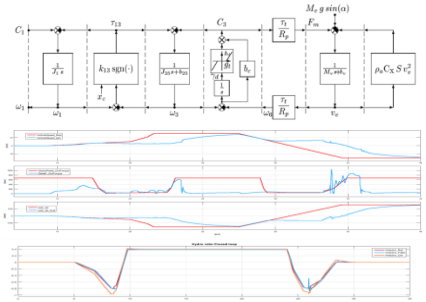
Requirements Definition

- Desired operations, routines and leading dynamics.





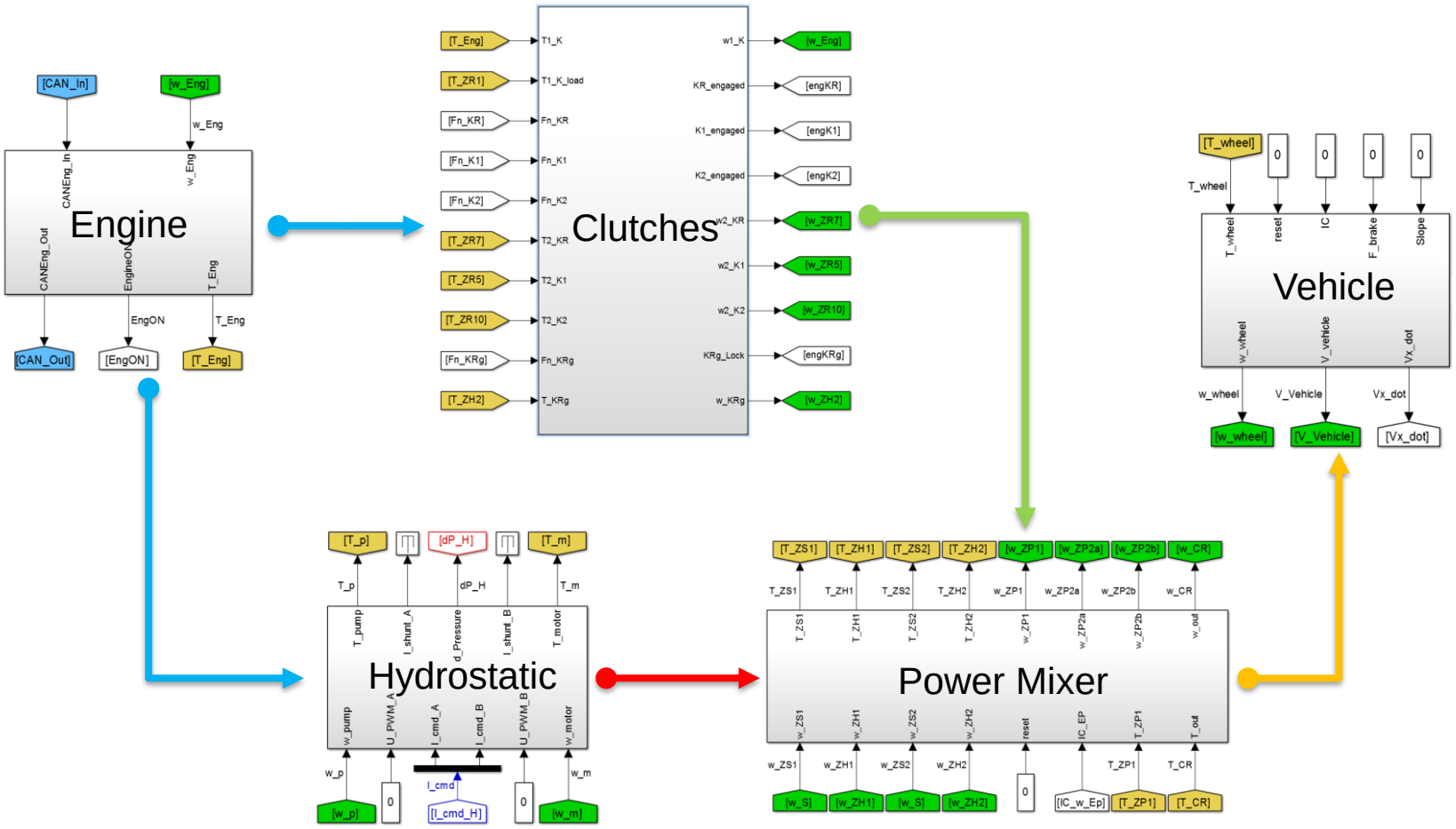
Model Based Design - Development V-Cycle Steps



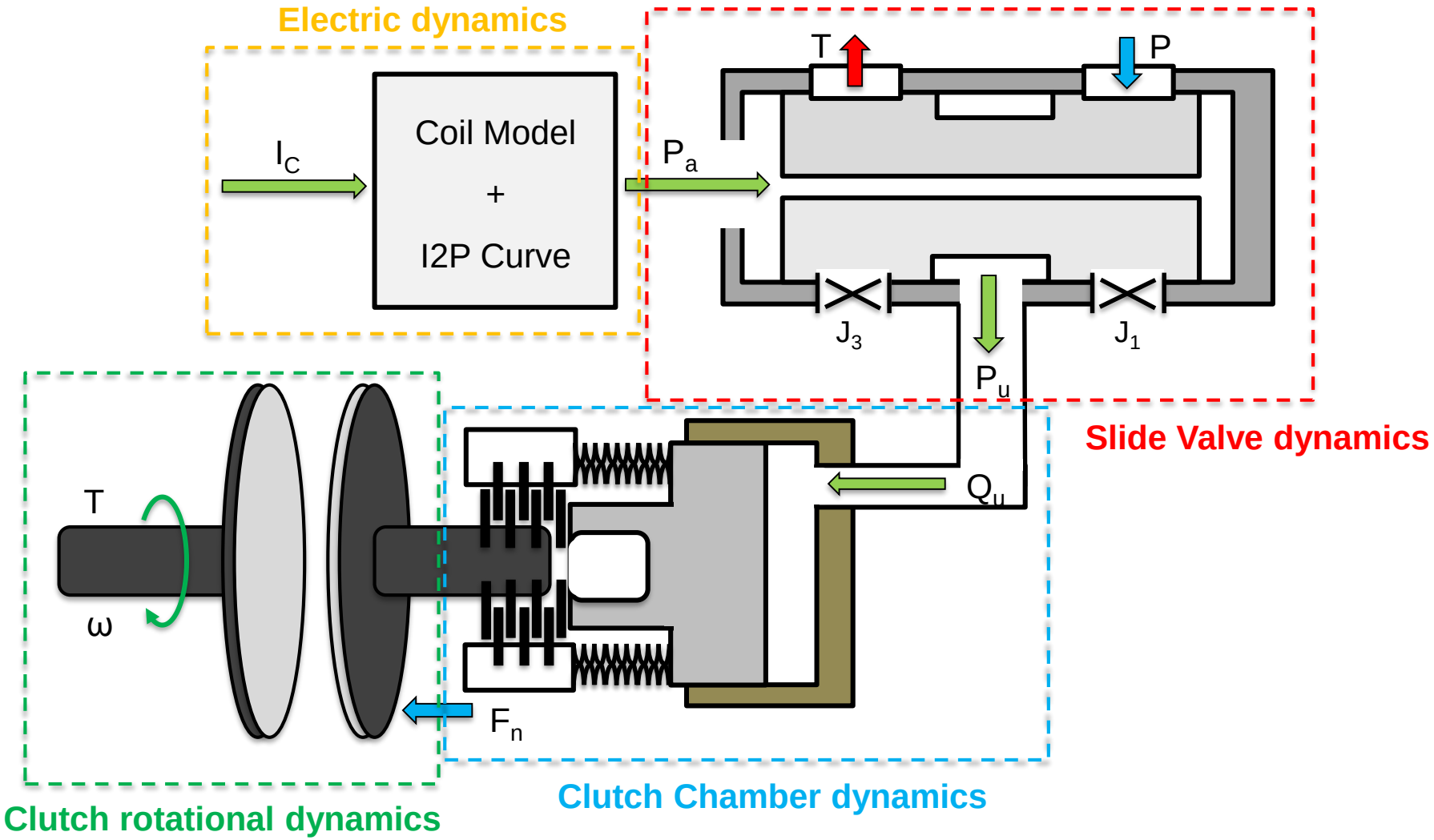
Modeling & Simulation

Modelling & Simulation - Complete POG Model

From the requirements and the system description, the POG model of the entire tractor is implemented in order to describe its main dynamics.

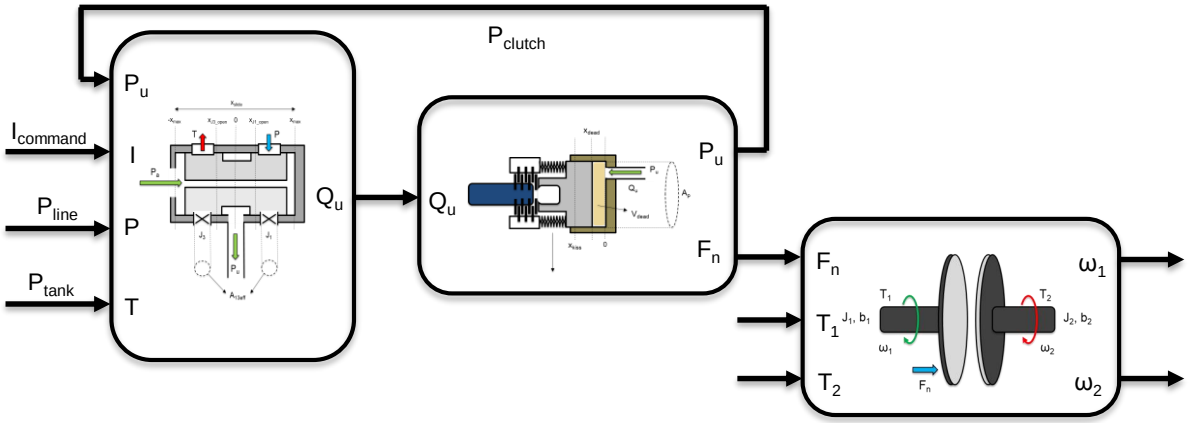


Modelled subsystem - Wet Clutch Model

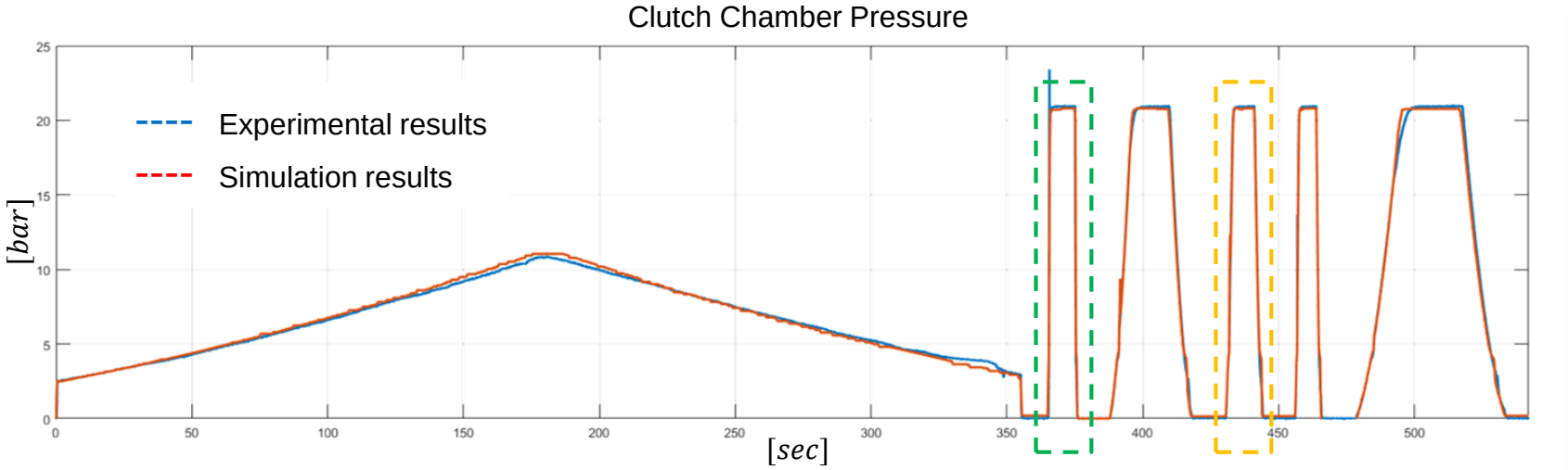


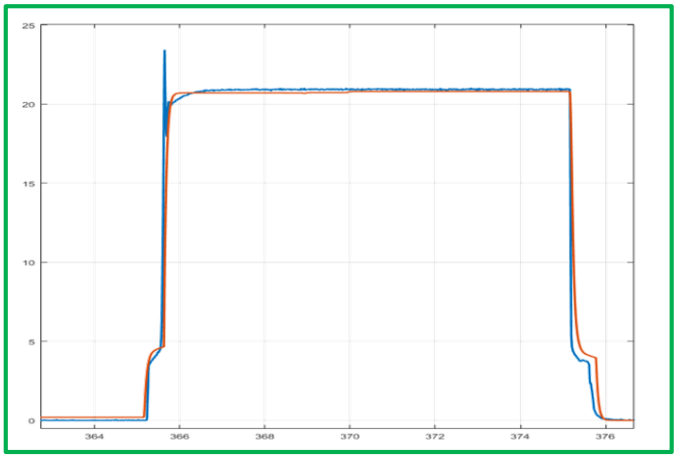
Wet Clutch Model - Validation set-up

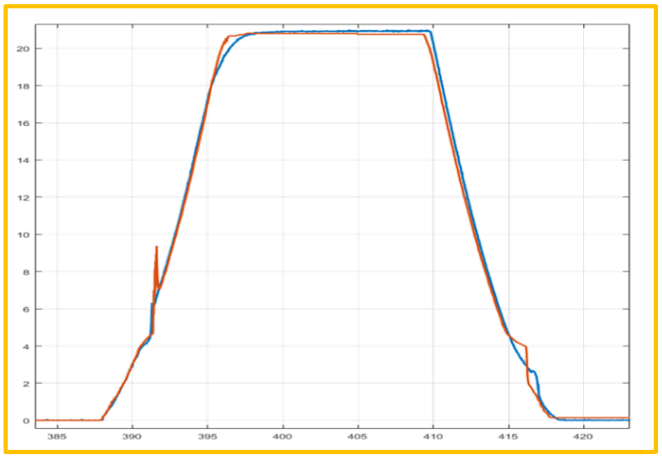
The Data comparison between the mathematical model and the experimental set-up results permits the validation of the implemented subsystem dynamics.



Wet Clutch Model - Validation results

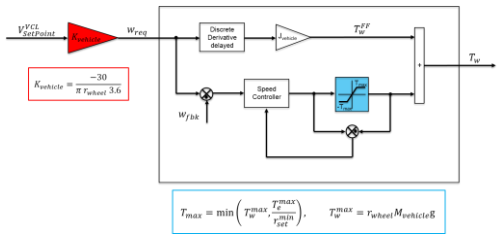






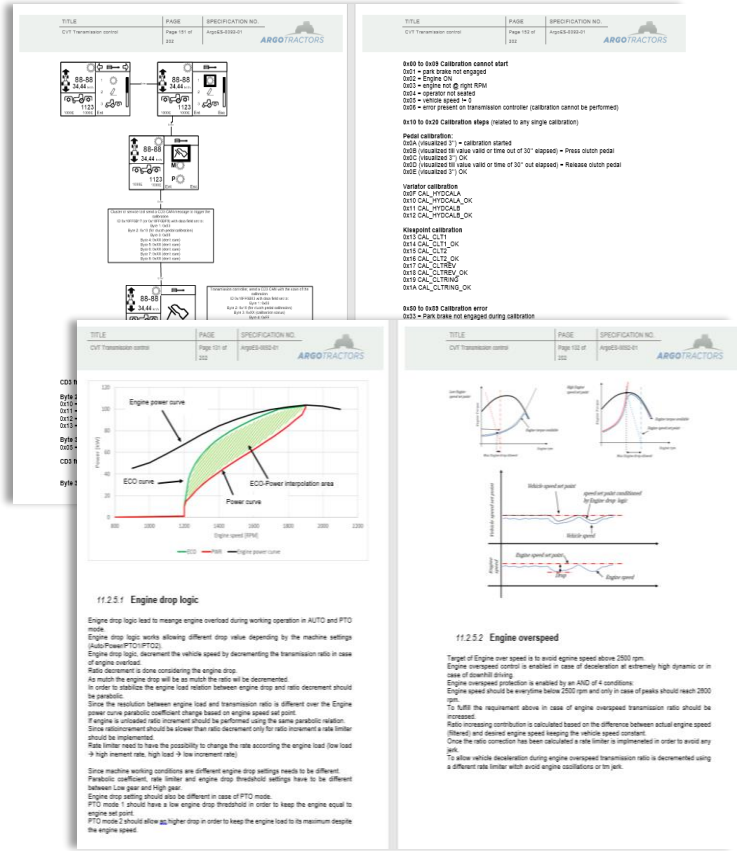
Model Based Design - Development V-Cycle Steps

Control Design & Prototyping



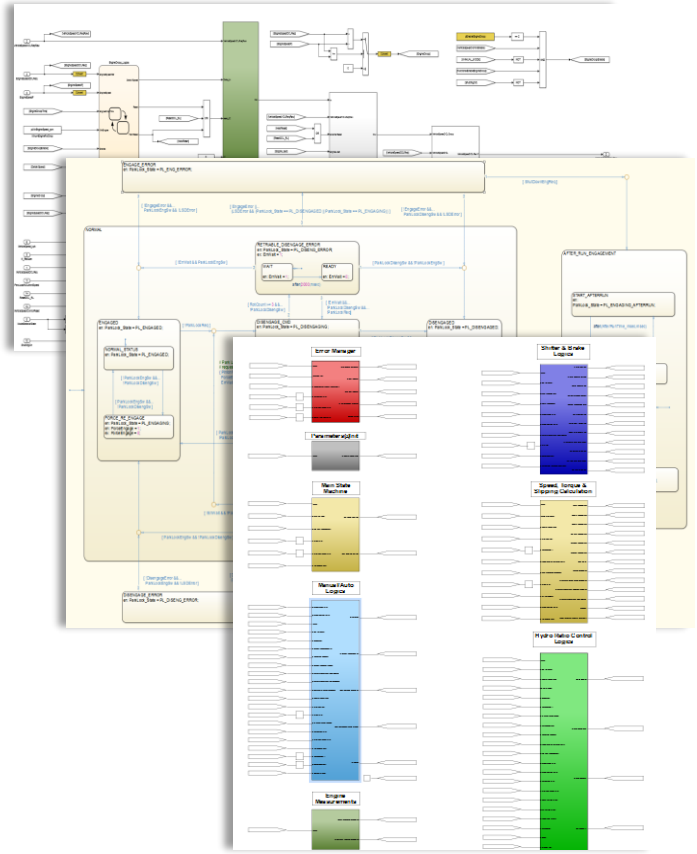
Control Design - Logics and Algorithms

According to the written requirements, the main logics and the control algorithms are implemented in order to match the requests optimizing the vehicle performances (simulation analysis and data processing).



Design

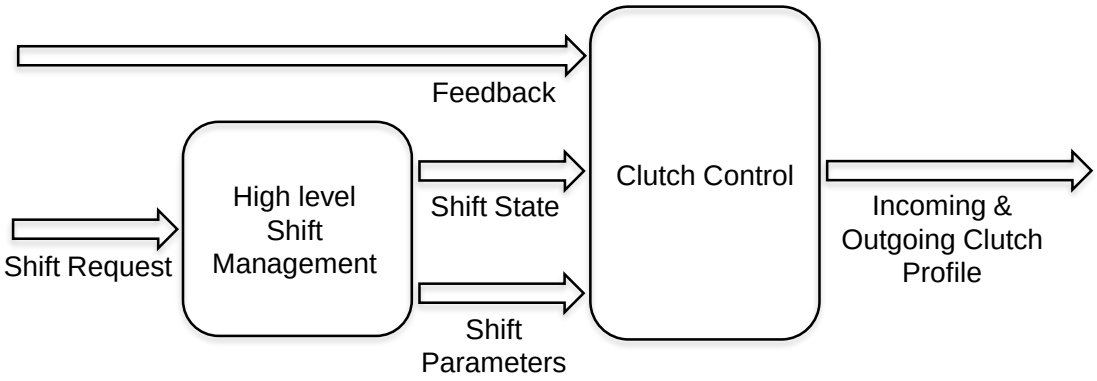
Potential reviews

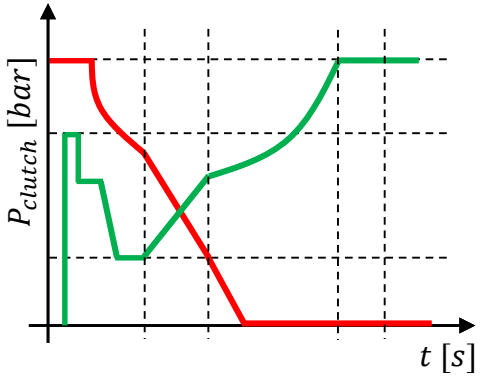


Control Design - Logics and Algorithms

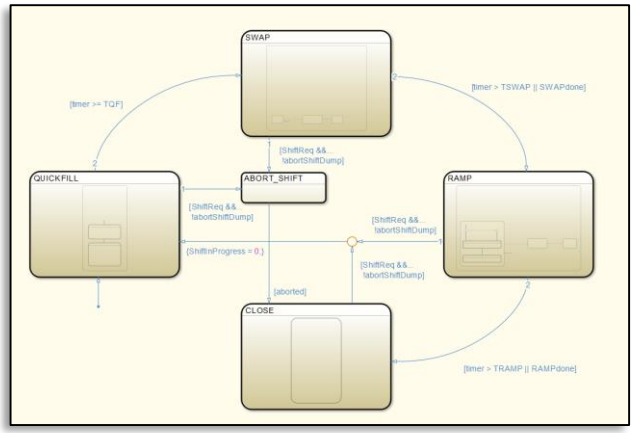
The logics and the control algorithms design is firstly focused on the single subsystem.

Automatic Gear Shift Control

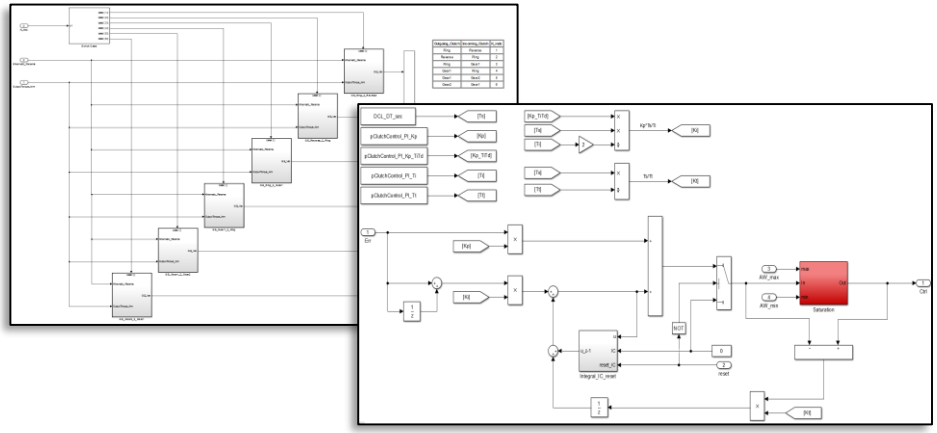




High Level Gear Shift Logics (Stateflow)



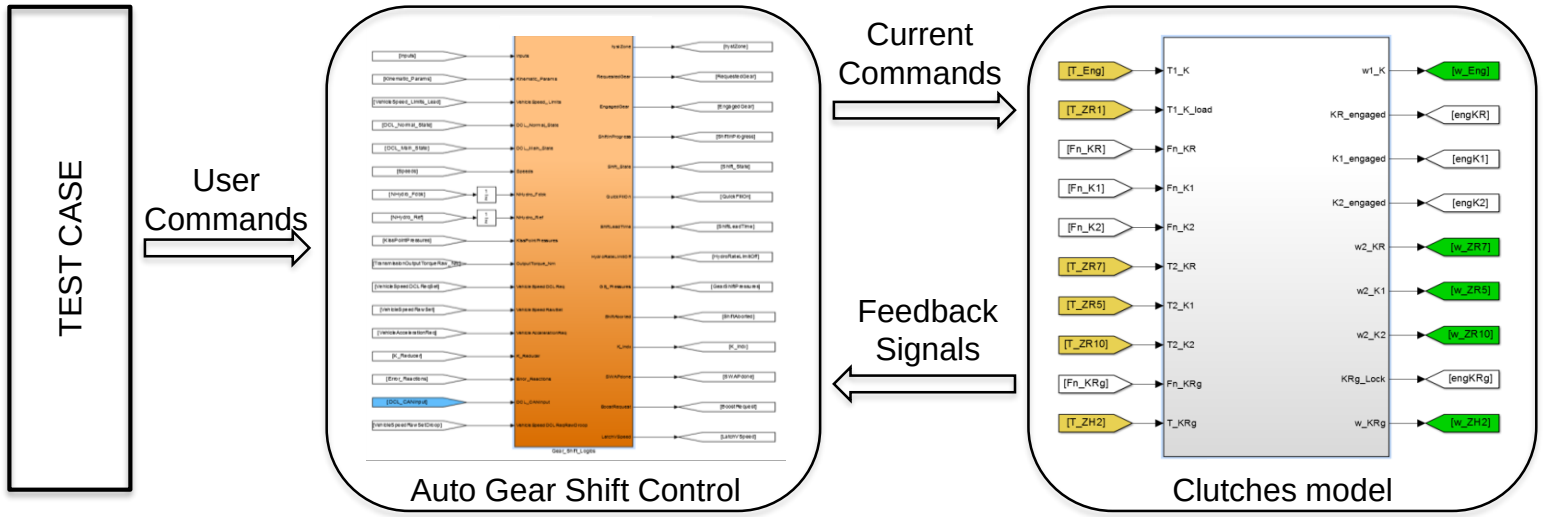
Closed loop Clutch Control algorithms (Simulink)



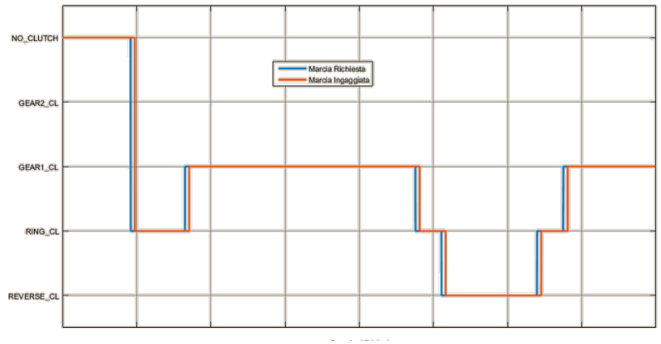
Control Prototyping - Logics and Algorithms

The implemented model allows a rapid prototyping of the designed subsystem.

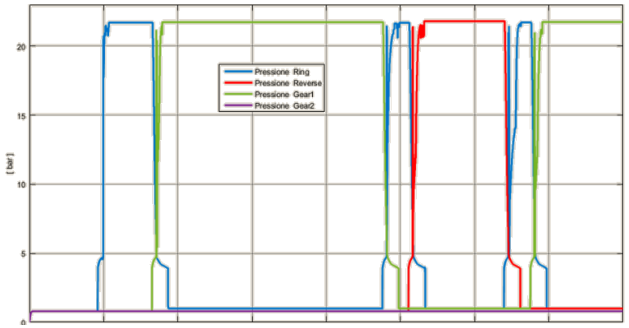
Automatic Gear Shift Control



Shift State

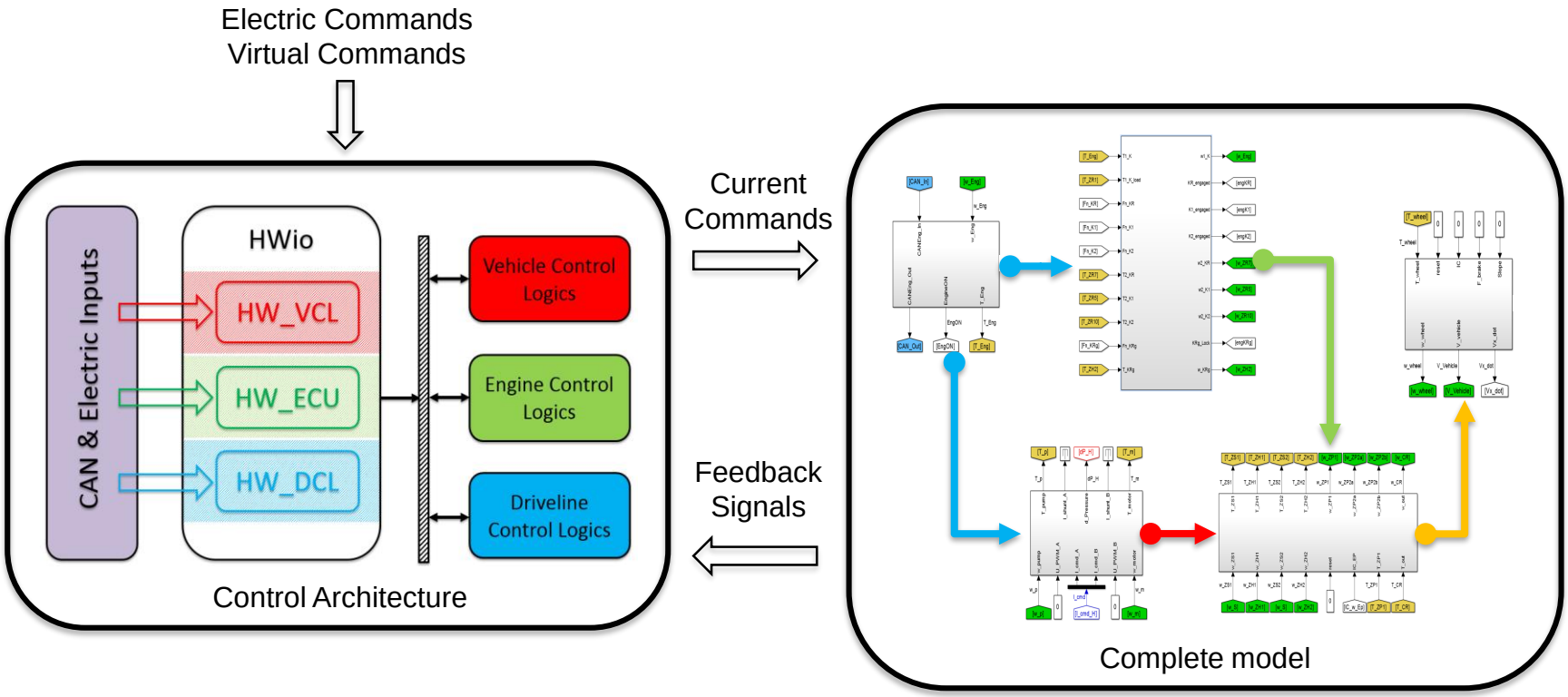


Clutch Pressure Profiles



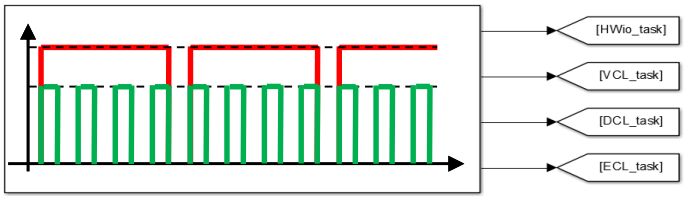
Control Design - Architecture

The mathematical model describes the main dynamics of the system as function of the current commands, that are computed by complex control algorithms as function of the operator commands (pedals, buttons, levers, handles, etc.) and plant feedback signals (speed, pressure, current, etc.).

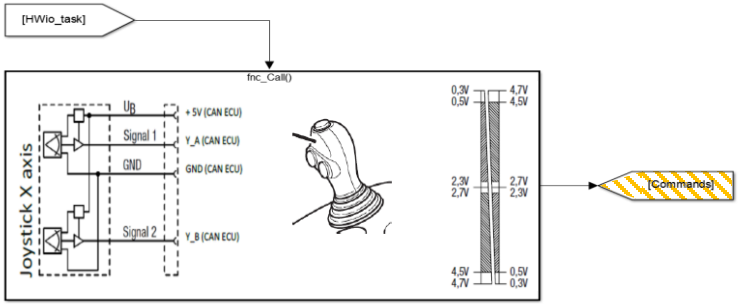


Control Design - Architecture

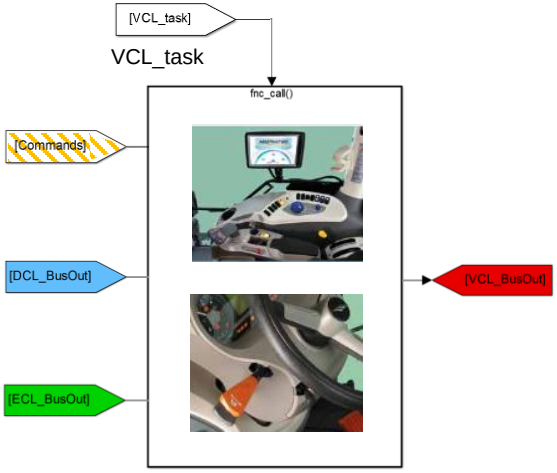
The complete control system is the collection of all the implemented subsystems and must be arranged and connected in order to guarantee the correct causality and scheduling (signals, multi sample time, parameter scaling).



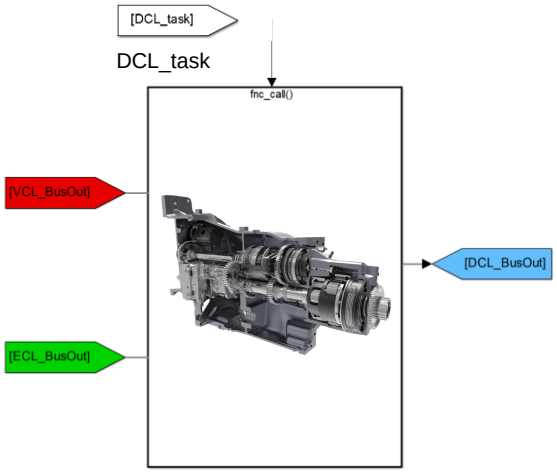
Scheduler



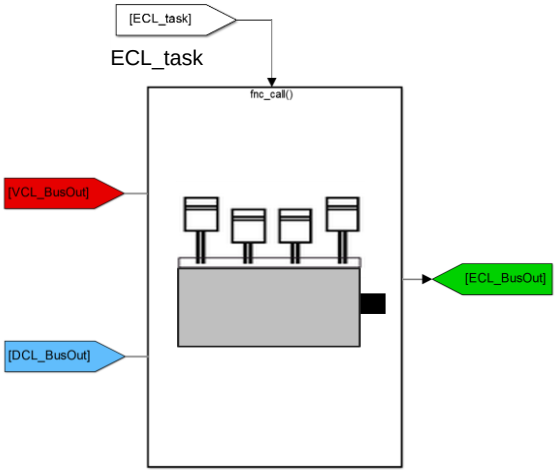
Hardware I/O Logics



Vehicle Control Logics



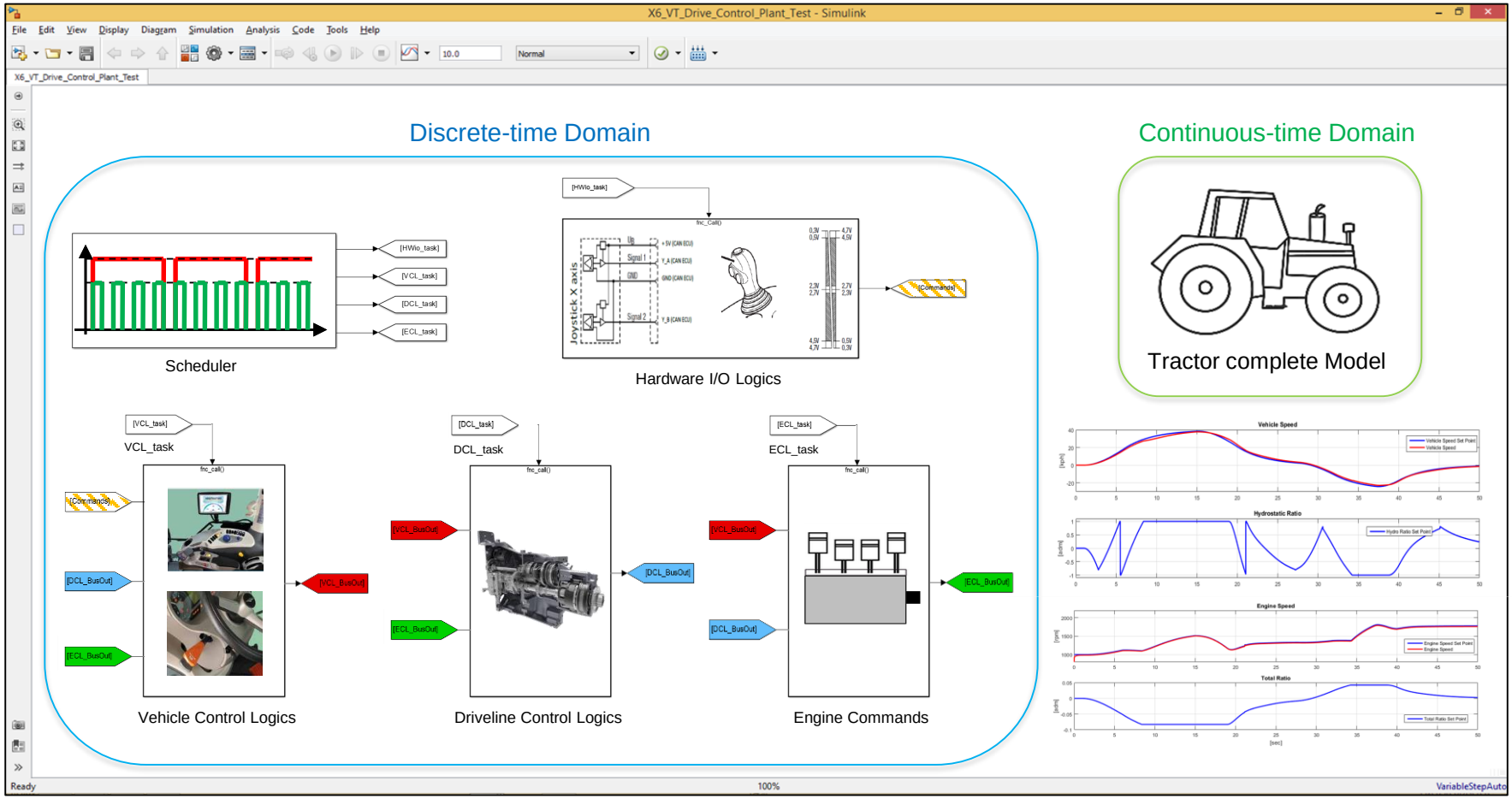
Driveline Control Logics



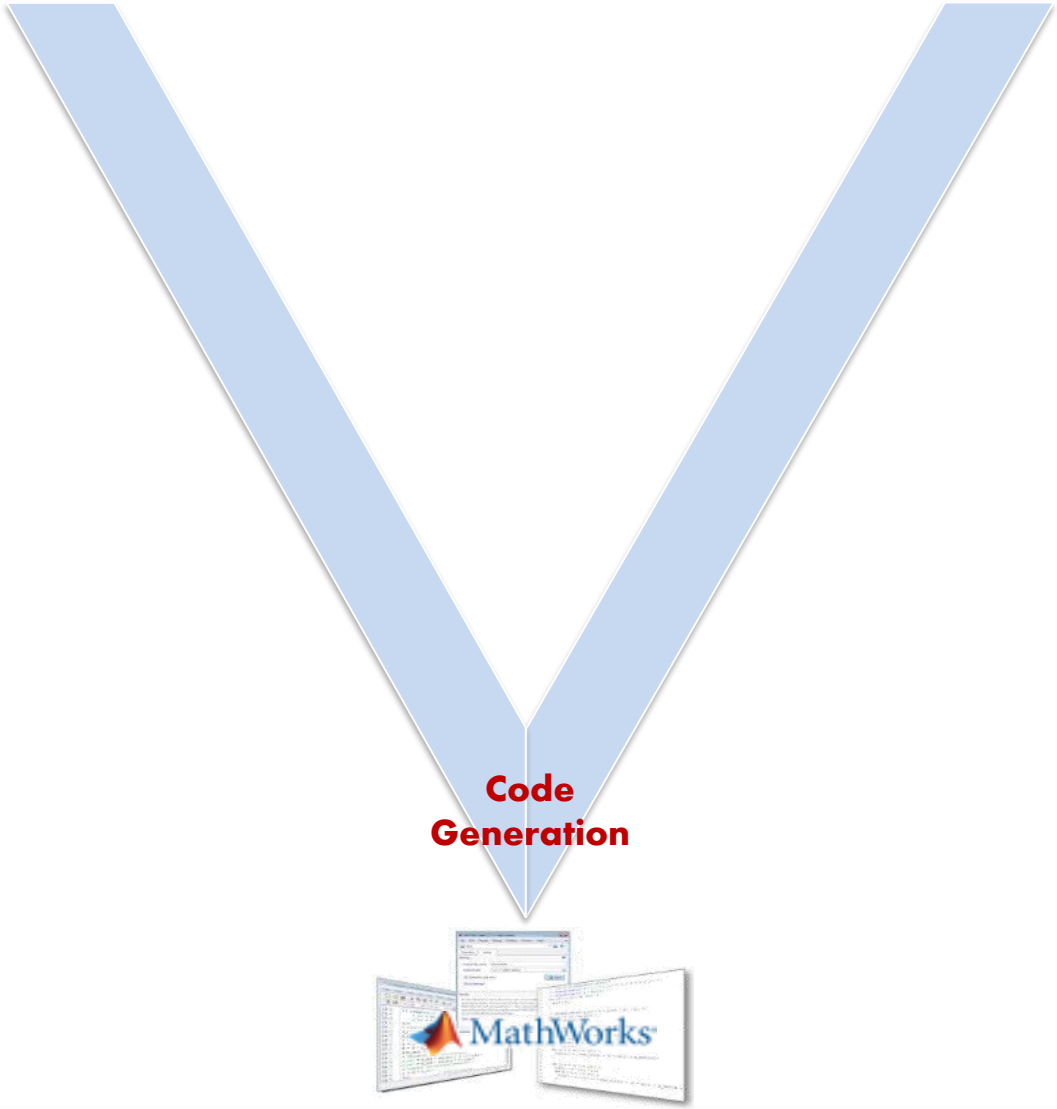
Engine Commands

Control Design - Prototyping simulation

In order to validate the implemented control algorithms and logics, mixed time domain simulations can be performed with the mathematical model of the system.



Model Based Design - Development V-Cycle Steps

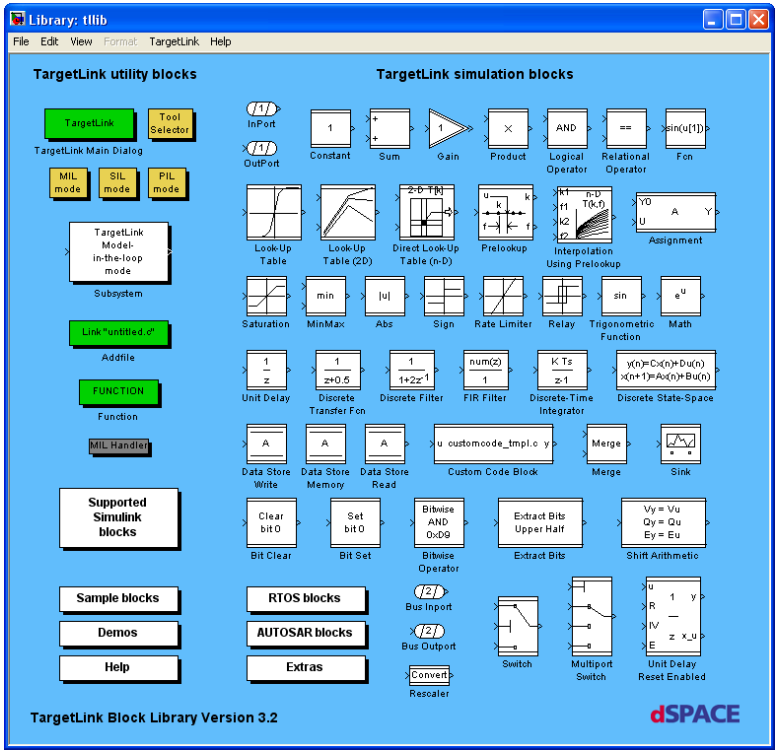


Code Generation - Model Preparation

In order to transform the control algorithms into C source code, the implemented models must be adequately enhanced or prepared.

TragetLink

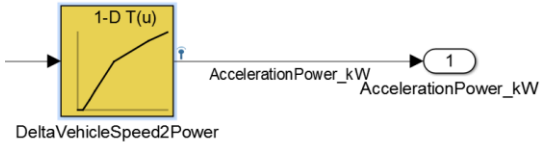
dSPACE

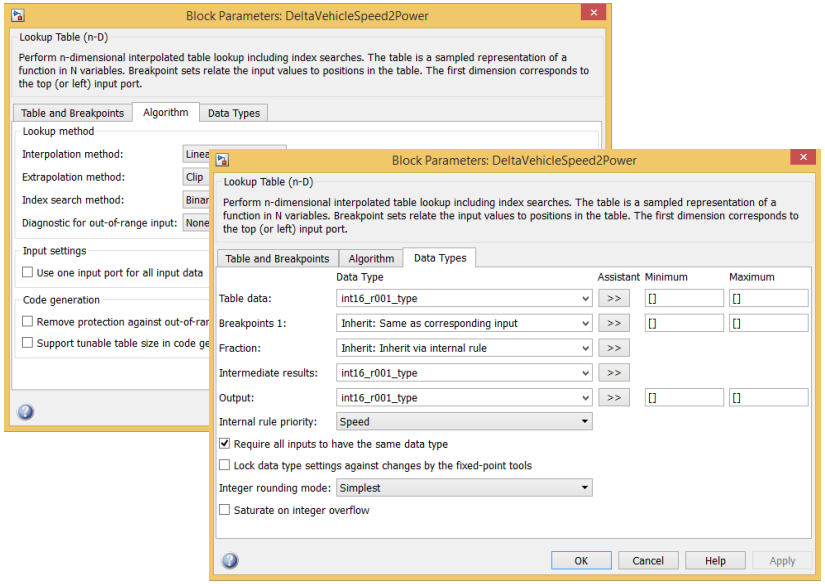


Enhancement: the used blocks are substituted with new ones coming from the TargetLink libraries.

Simulink Coder

MathWorks®





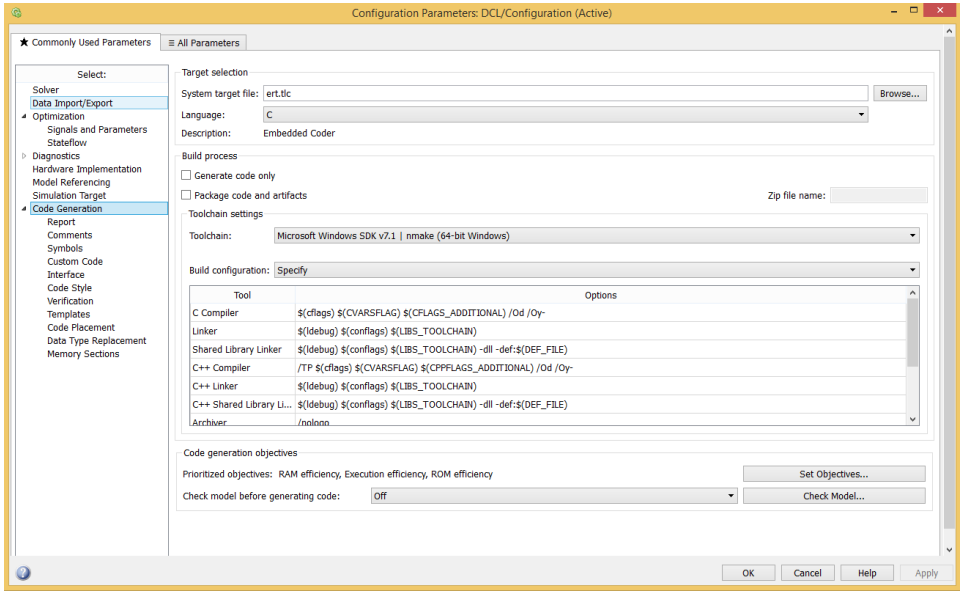
Preparation: the properties needed for the code generation are already in the block parameters.

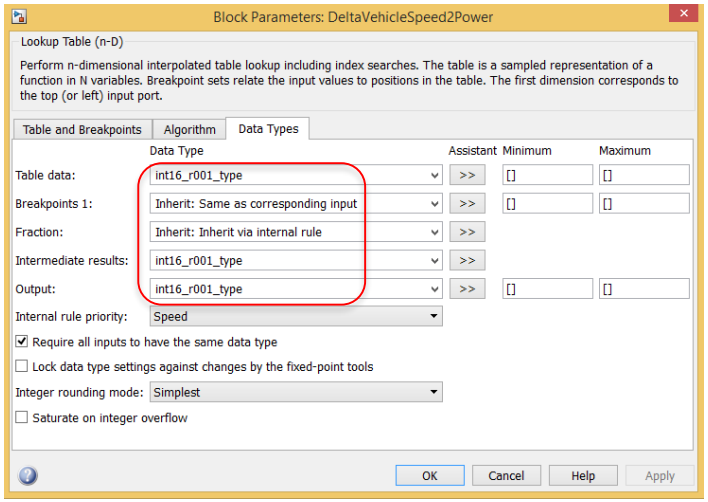
Code Generation - Model Preparation

The model preparation consists of several settings to let Simulink generate and execute C code from the implemented models (Simulink, Stateflow, Matlab functions). These settings strongly depend on the project that will host the code and the hardware that will run it.

MathWorks®

Code Generation Options:
Custom Code, Symbols,
Code Style, Code
Optimization, etc…

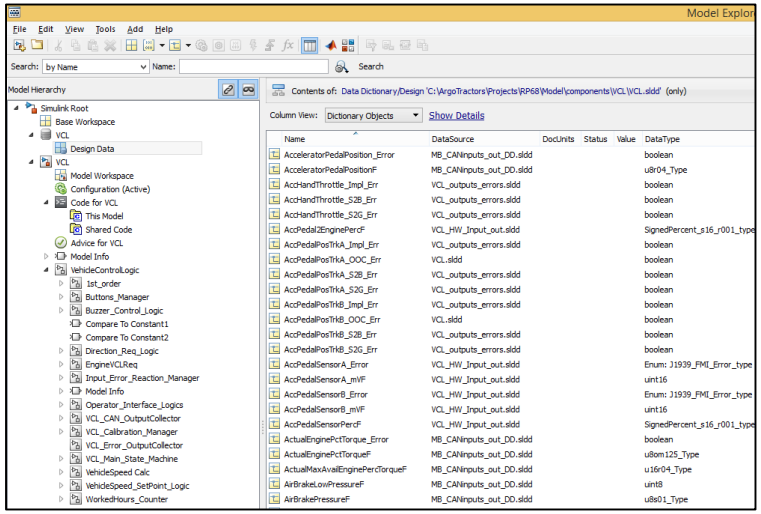


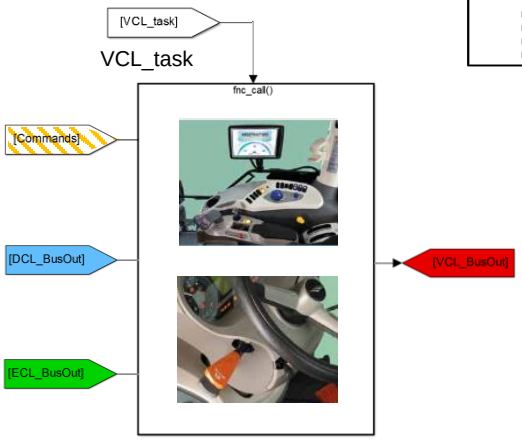


The data type selection and scaling depends on the ECU (integer or floating point).

Code Generation - Data Dictionary

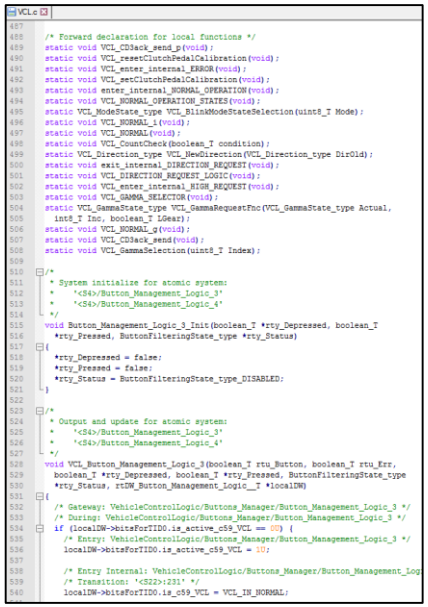
The Data Dictionary is the document that entirely describes the model to be transformed: input/output variables, parameters, data type, storage classes and all those model objects that let the Coder generate the source code that behaves exactly like the implemented Simulink model.





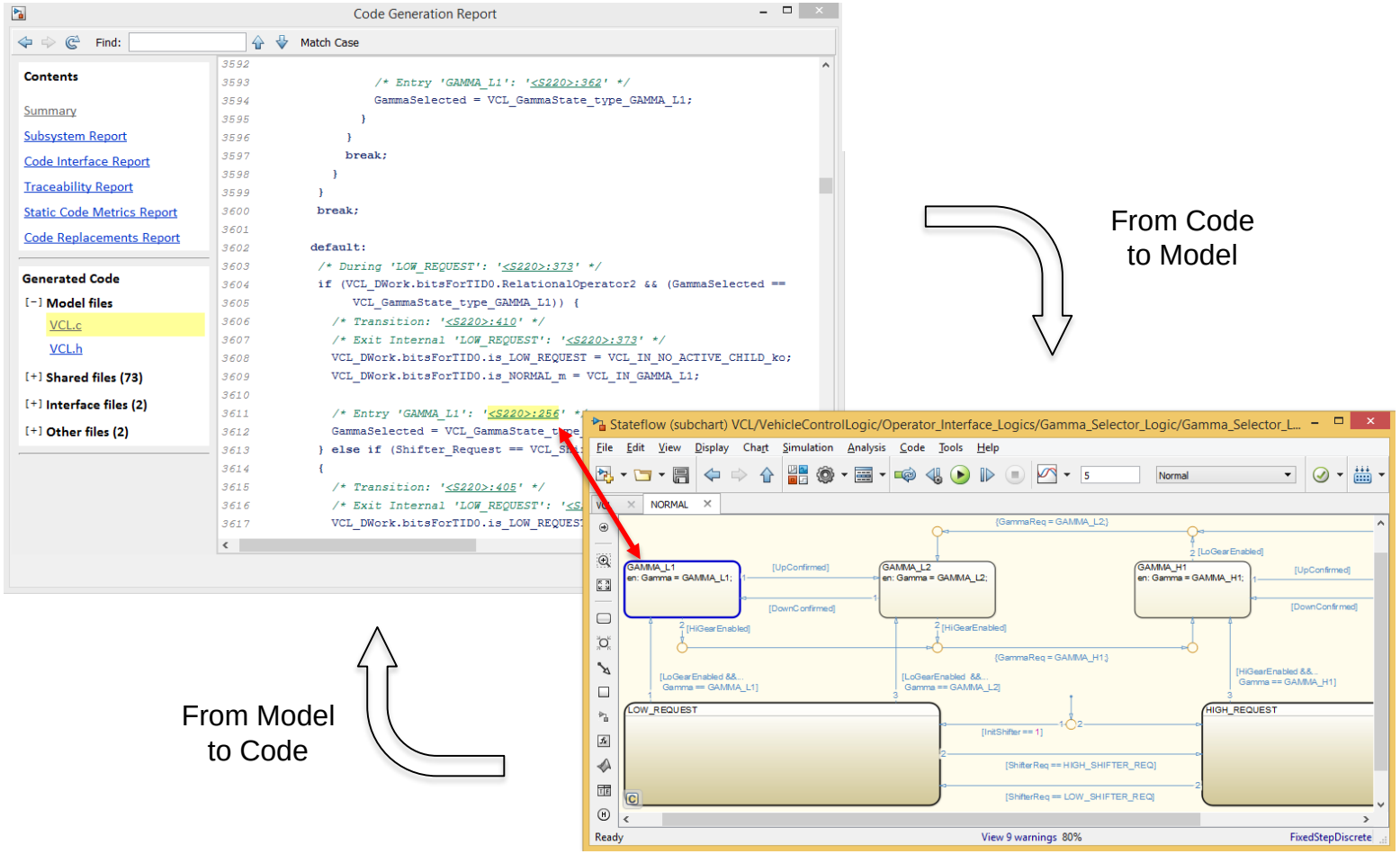
Vehicle Control Logics

Bidirectional links

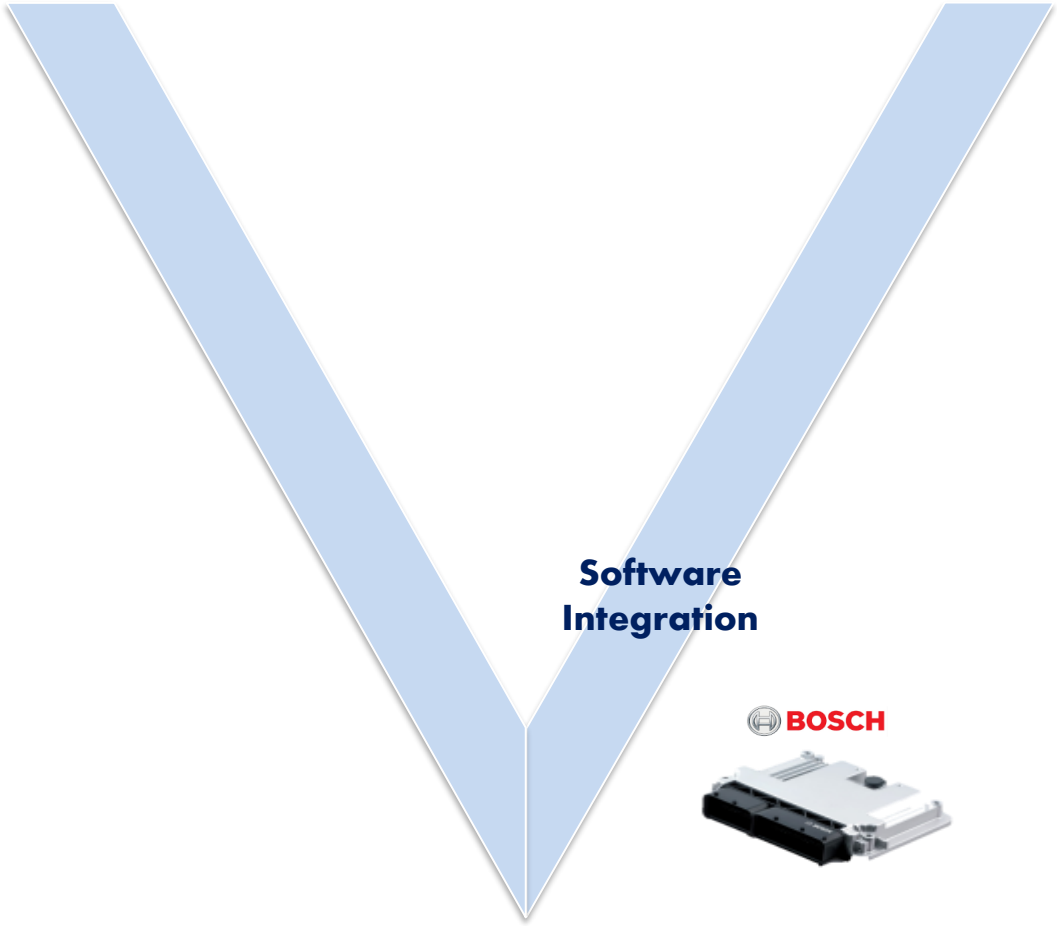


Code Generation Report

Simulink Code Generation Report highlights bidirectional traceability between model and code.

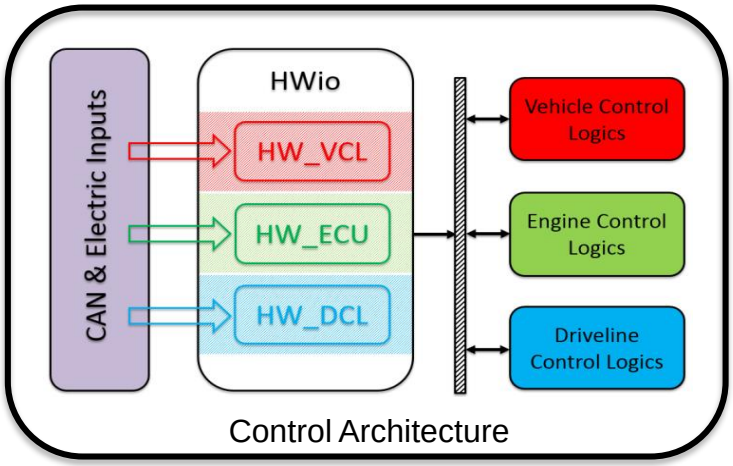


Model Based Design - Development V-Cycle Steps

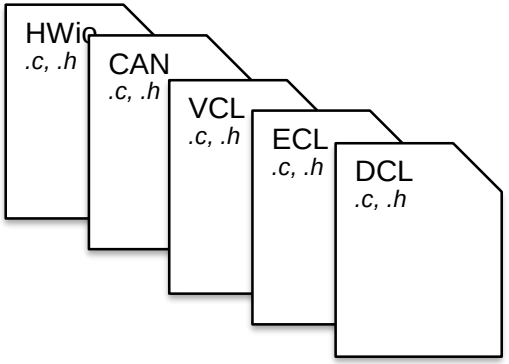


Software Integration

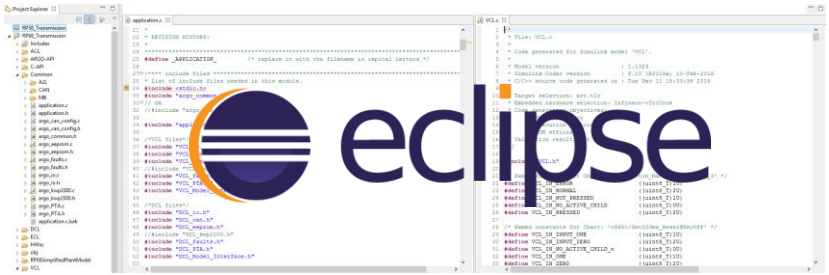
The source code files corresponding to the generated models are then integrated into a global project that is built for custom ECUs.



Code Generation



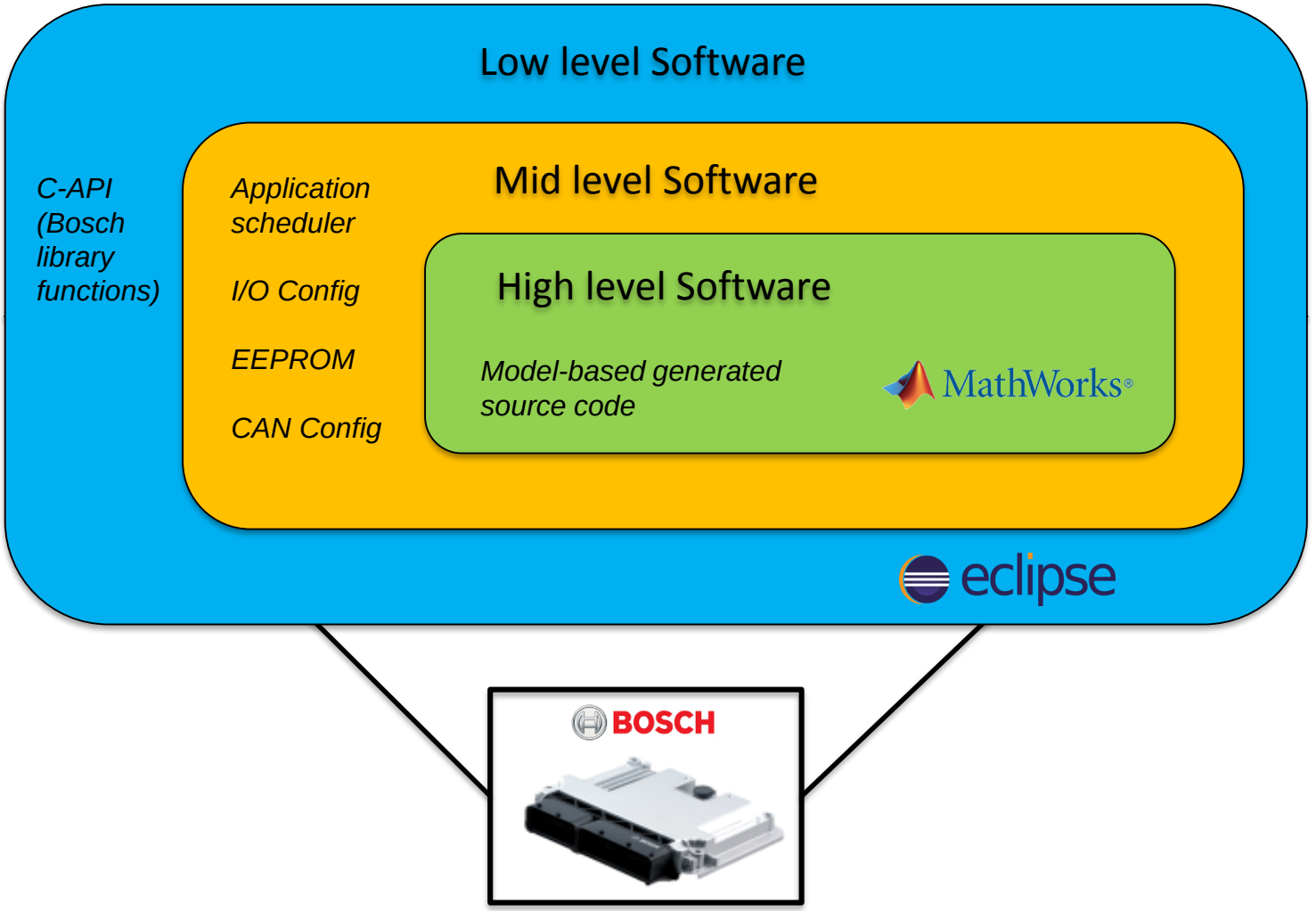
Software Integration



Software Build

Software Architecture

The architecture of the complete ECU software is composed by three main levels.



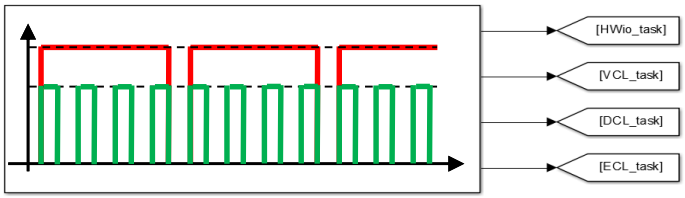
Model Based Design - Development V-Cycle Steps

SW Validation
SIL, Bench, HIL

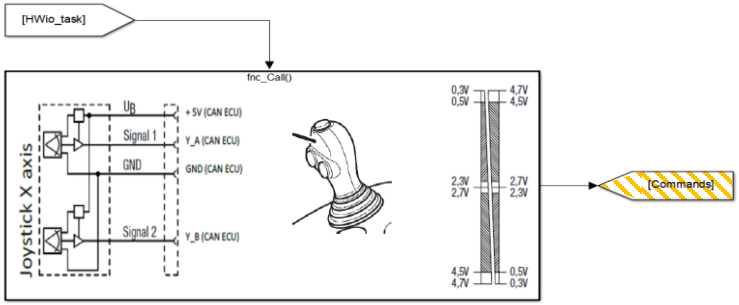
Software Validation - SIL

The model-based generated source code (High level software) must be validated in order to verify if its behaviour matches with the system control algorithms one.

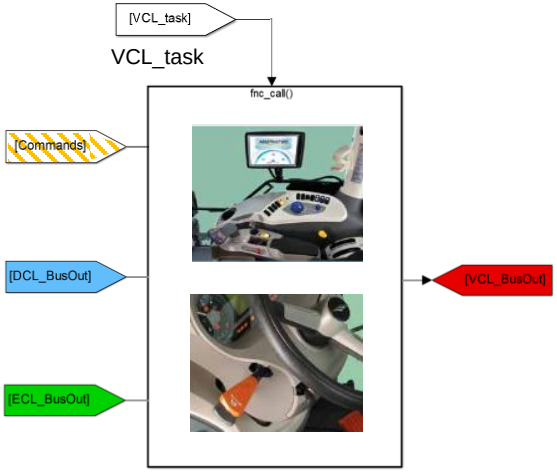
From control prototyping simulation (Model In the Loop, *MIL*)…



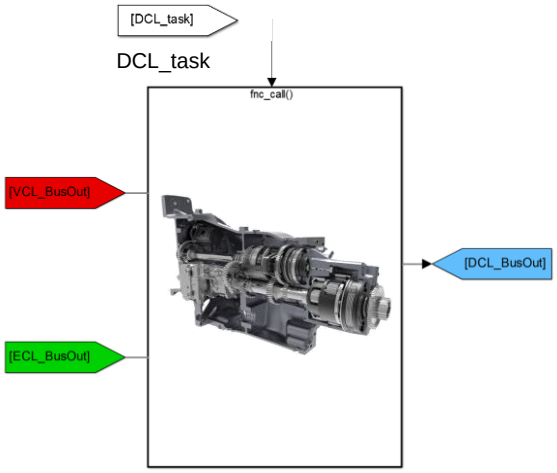
Scheduler



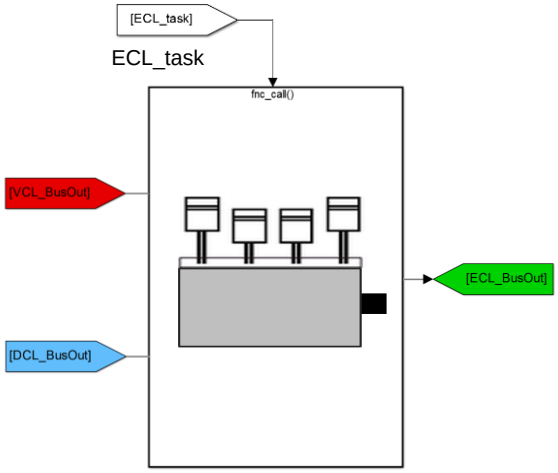
Hardware I/O Logics



Vehicle Control Logics



Driveline Control Logics

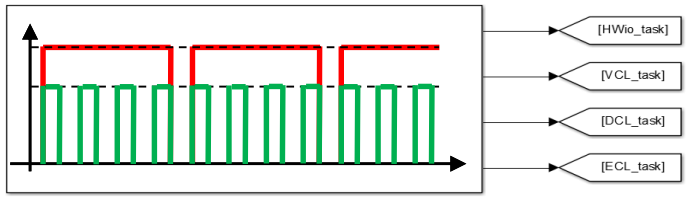


Engine Commands

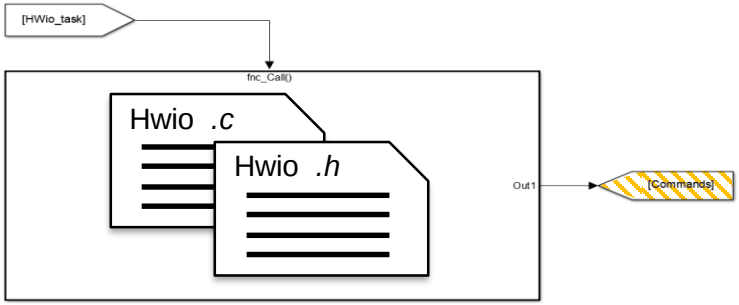
Software Validation - SIL

The generated model-based source code (High level software) must be validated in order to verify if its behaviour matches with the system control algorithms one.

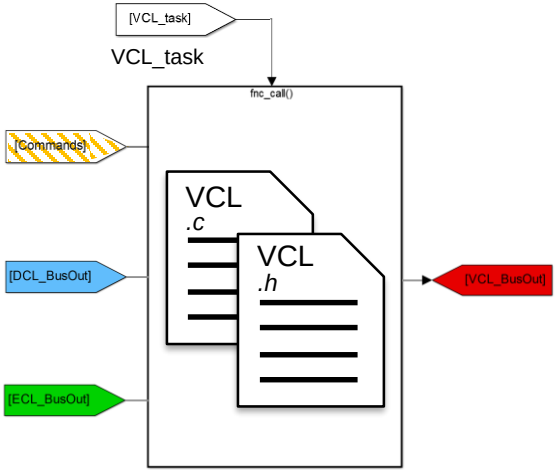
…to Software In the Loop (*SIL*) simulation



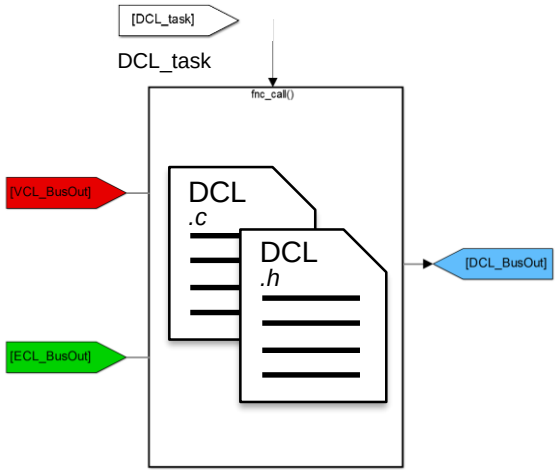
Scheduler



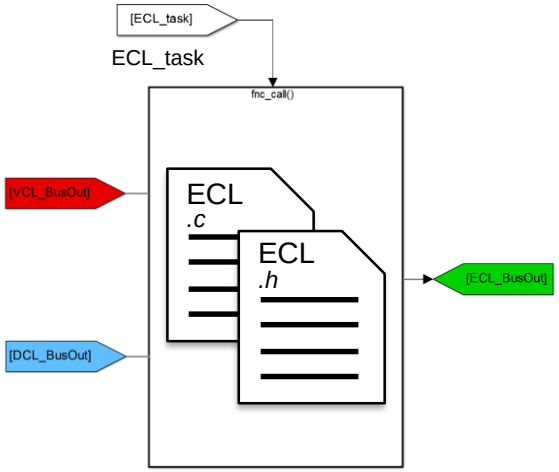
Hardware I/O Logics



Vehicle Control Logics



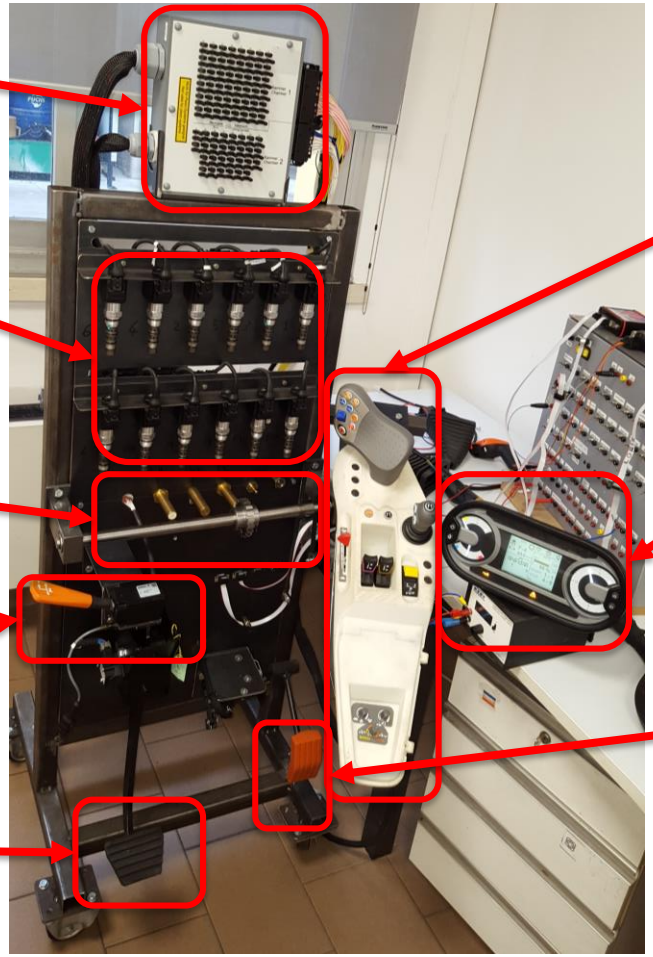
Driveline Control Logics



Engine Commands

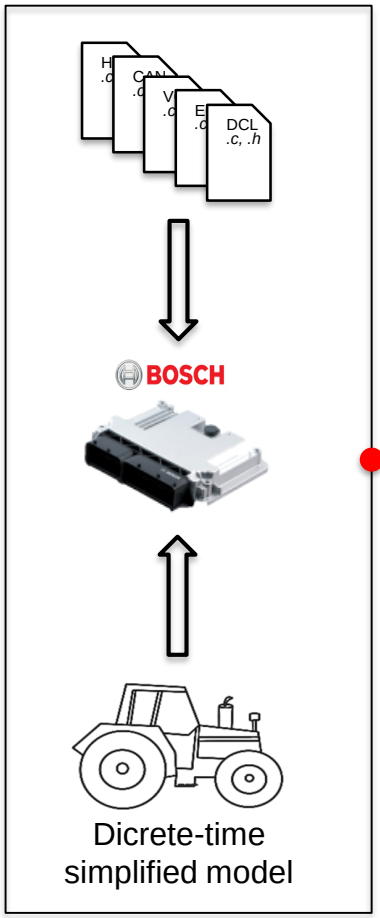
Software Validation - Test Bench

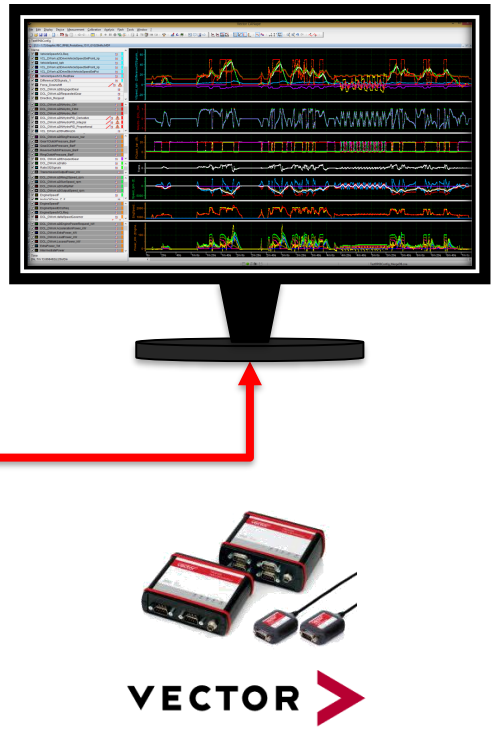
The test bench validation permits the test of the complete ECU software with actual input/output commands.



Software Validation - Test Bench

Adding a simplified dicrete-time version of the model to the ECU project and connecting the test bench to a laptop, a real time simulator of the system is implemented.



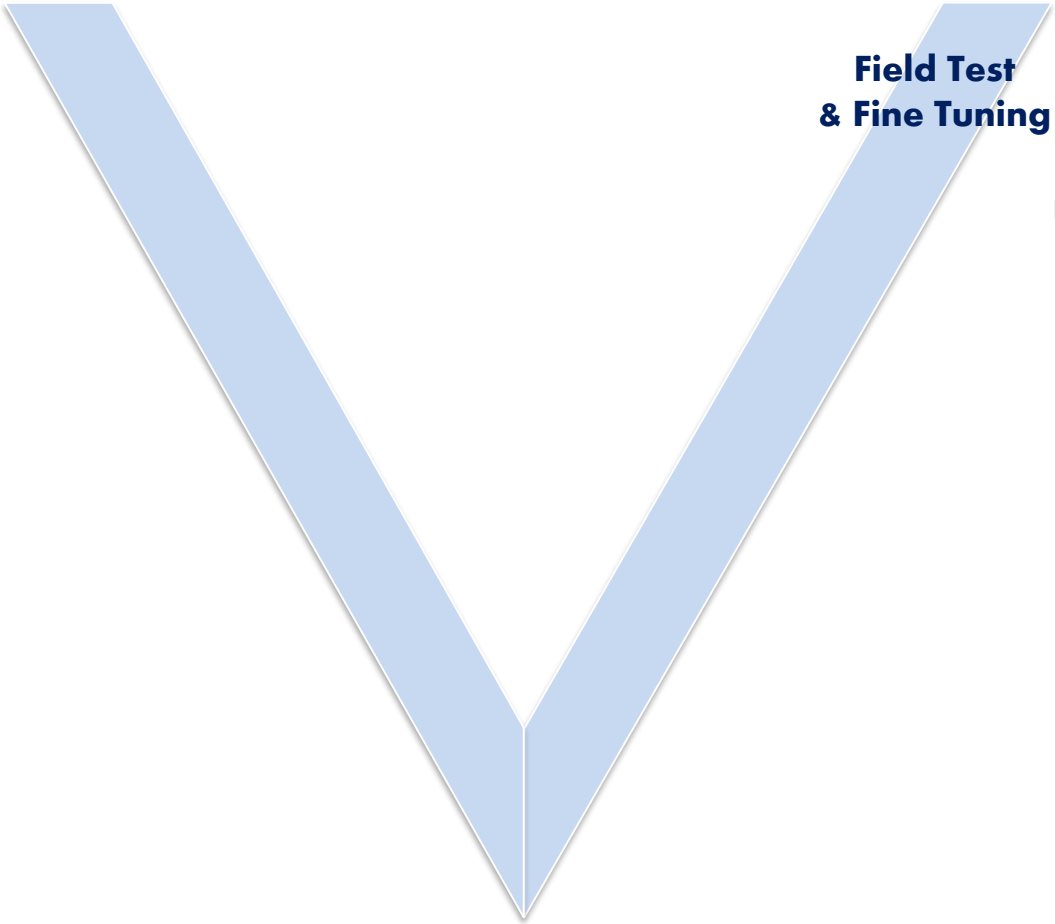


Further Test and Validation

- Hardware In the Loop (HIL):
  - Modelling and Simulation
  - Analysis and Reporting
  - Real-time SW Testing
  - Requirements Traceability
  - Test Automation

- AVL Test bench:
  - HW Endurance Test
  - HW Stress test
  - SW Automation Test

Model Based Design - Development V-Cycle Steps



Field Test
& Fine Tuning

Drive & Field Test

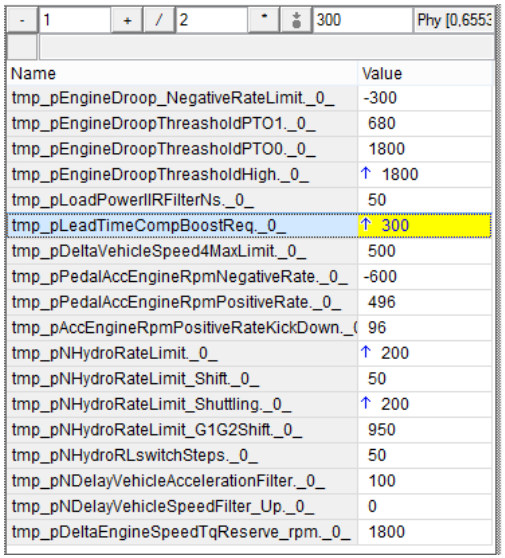
The on board test has a key role to verify all those dynamics that are not possible to model in detail.

Drive Test (Test Track)

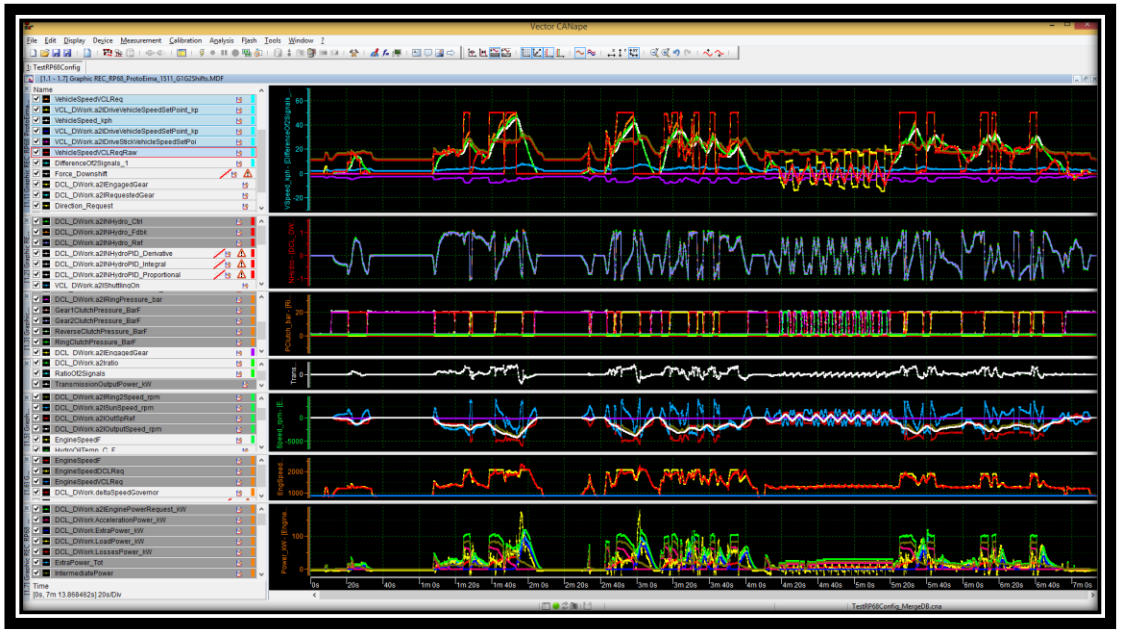
Field Test

Fine Tuning

Real-time parameter changing to evaluate the different system dynamic responses and obtain the best performances.



Name	Value
tmp_pEngineDroop_NegativeRateLimit._0_	-300
tmp_pEngineDroopThreasholdPTO1._0_	680
tmp_pEngineDroopThreasholdPTO0._0_	1800
tmp_pEngineDroopThreasholdHigh._0_	↑ 1800
tmp_pLoadPowerIIRFilterNs._0_	50
tmp_pLeadTimeCompBoostReq._0_	↑ 300
tmp_pDeltaVehicleSpeed4MaxLimit._0_	500
tmp_pPedalAccEngineRpmNegativeRate._0_	-600
tmp_pPedalAccEngineRpmPositiveRate._0_	496
tmp_pAccEngineRpmPositiveRateKickDown._0_	96
tmp_pNHydroRateLimit._0_	↑ 200
tmp_pNHydroRateLimit_Shift._0_	50
tmp_pNHydroRateLimit_Shuttling._0_	↑ 200
tmp_pNHydroRateLimit_G1G2Shift._0_	950
tmp_pNHydroRLswitchSteps._0_	50
tmp_pNDelayVehicleAccelerationFilter._0_	100
tmp_pNDelayVehicleSpeedFilter_Up._0_	0
tmp_pDeltaEngineSpeedTqReserve_rpm._0_	1800



Link Video

Thanks for your attention!!