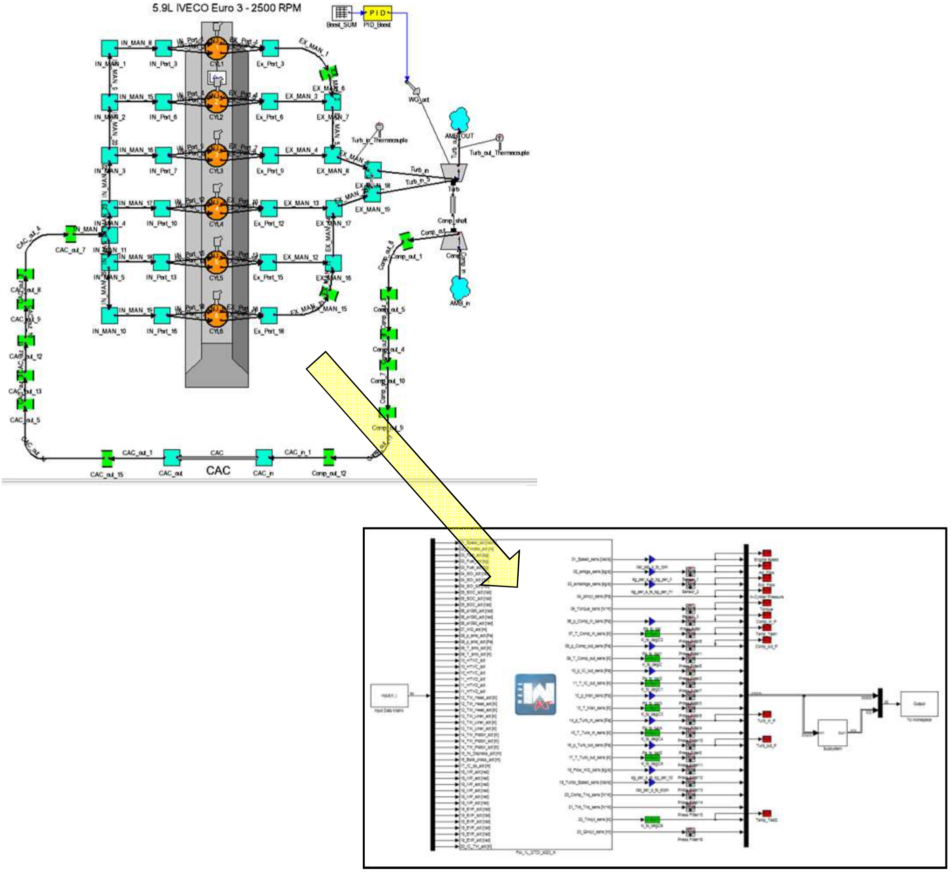
HiL Tools – WAVE-RT

This workshop introduces WAVE-RT conversion process on real WAVE engine model, showing the recommended modelling techniques and example of HiL application



Date 17th Jan 2013

Report by Jiri Navratil and Martin Horacek

Summary

- **Aim:**
  - The aim of this seminar is to introduce a WAVE-RT conversion process on real WAVE engine model and its HiL implementation example and scheme
  - The second aim it to show life demos of WAVE model run vs. WAVE-RT run

- **Method:**
  - Ricardo Software WAVE validated full load model is converged to real time WAVE-RT model
  - C++ (MS Visual Studio 2010) and Simulink are used to drive the WAVE-RT model

- **Conclusions:**
  - WAVE-RT present state of art software code running real time or faster even on common PCs intended to be used for SiL and HiL applications

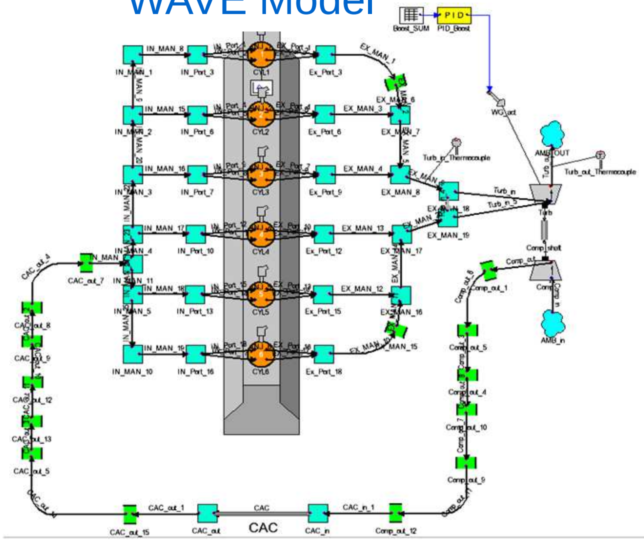
Agenda

- Introduction to WAVE-RT
  - **WAVE-RT Background and Basic Information**
  - Tutorial + Help
  - WAVE-RT vs. WAVE Unsupported Objects and Features
- WAVE Model Preparation
- WAVE-RT Model
- HiL Implementation
- Conclusions

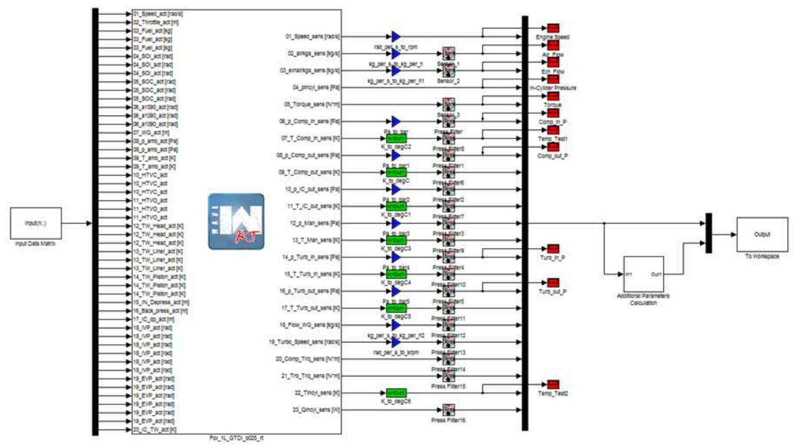
WAVE-RT is simplified 1-D gas dynamic program intended to run real-time

- WAVE-RT is a real-time capable engine simulation solver, based on in-cycle modelling
- It has been developed to provide the real time link between thermodynamic solver RT and control system development and testing environments, including MATLAB/Simulink for Software-in-the-Loop (SiL) testing, Hardware-in-the-Loop (HiL) testing platforms, and even embedding real-time models in future ECUs
- WAVE-RT model is derived (created) from a donor WAVE model using geometry placed on WaveBuild canvas

WAVE Model



RT Model with Simulink Driver



INTERNAL UNAPPROVED DRAFT

WAVE could be described as a luxury car in 1-D gas dynamic simulation and WAVE-RT as a sport car

WAVE

- Luxury - user friendly
- Lots of comfort and possibilities
- Good for long drives – optimisations
- Slower and safe (stable)
- Nice view outside – sensors, actuators, plot options

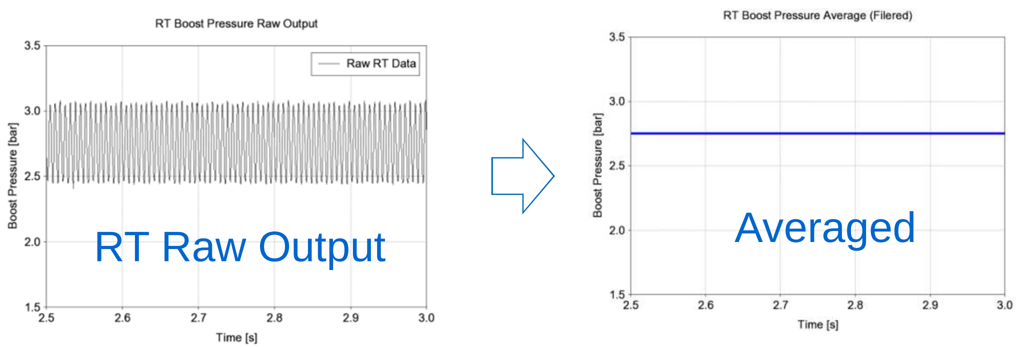
WAVE-RT

- Very fast and powerful
- Needs an experienced driver

INTERNAL UNAPPROVED DRAFT

WAVE-RT is not WAVE; both are different 1-D gas thermodynamic software codes with different solvers!

- Solver and modelling approach is simplified compared to WAVE to allow the model run as fast as possible
- User ‘has to pay’ for the speed by loosing some of the accuracy and user friendliness
- Designed to run within engine ECUs so the interface is similar what we can see at real engines
- All output parameters are crank angle dependent
  - No cycle average values
  - User has to average (apply filters)

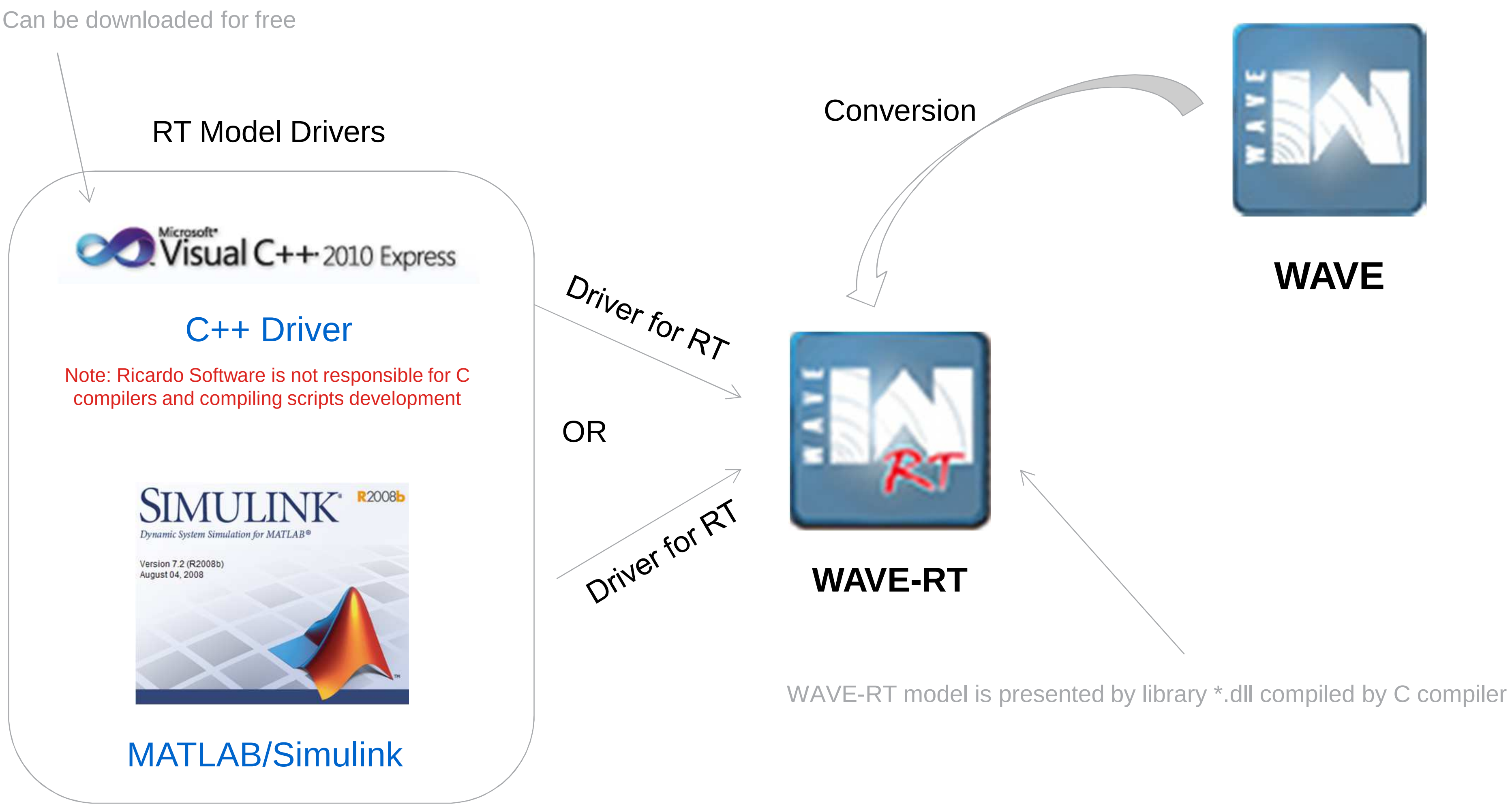


- Still captures pressure waves and other 1-D gas phenomena correctly!

User has to expect slightly different results compared to baseline WAVE model since the solver is different

INTERNAL UNAPPROVED DRAFT

User normally needs 4 programs to be installed: WAVE, WAVE-RT, C++ compiler (e.g. Microsoft Visual Studio 2010) and MATLAB/Simulink



- Any other C program language compilers can be used

Agenda

- Introduction to WAVE-RT
  - WAVE-RT Background and Basic Information
  - **Tutorial + Help**
  - WAVE-RT vs. WAVE Unsupported Objects and Features
- WAVE Model Preparation
- WAVE-RT Model
- HiL Implementation
- Conclusions

Every new user should go and read WAVE-RT Help and go through the basic tutorial at the beginning

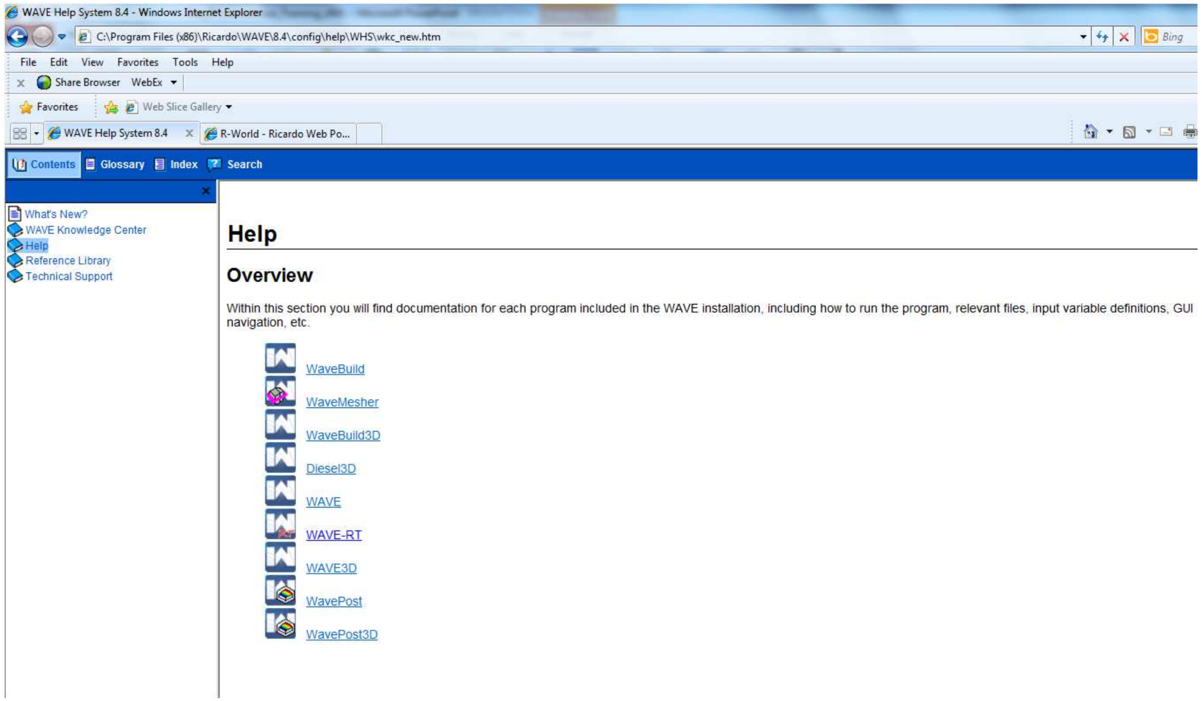
1) Start > All Programs > Ricardo > WAVE > WAVE Help System

2) Help > WAVE-RT

3) Tutorial can be found in: WAVE Knowledge Center > Tutorials > Advanced > WAVE-RT

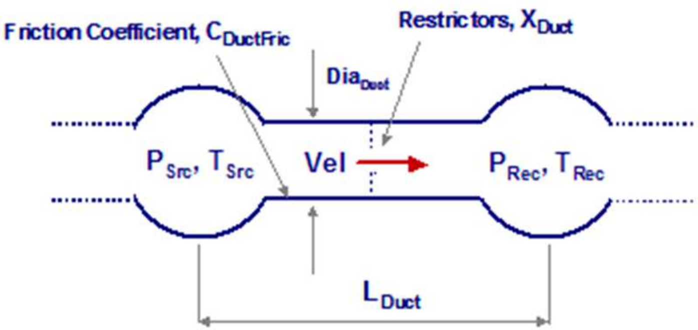
4) Go through the Preliminary Work chapter carefully

- The tutorial and WAVE preliminary work is prepared for 32-bit system mostly
- Several adjustments are needed for 64-bit systems



The WAVE-RT Theory is explained in detail within Help: Help > WAVE-RT > Using WAVE-RT > WAVE-RT Theory

- The flow network solver at the core of WAVE-RT is based upon the Quasi-Propagatory Model (QPM) as described by Cipollone and Sciarretta in numerous publications
- The QPM model tracks only **three transport species** (air, fuel, and exhaust gas) as opposed to WAVE's five (fresh air, vapor fuel, burned air, burned fuel, and liquid fuel)
- The engine system is described by a series of **ducts** and **capacities**. The velocity is calculated at the mid-point of each duct, followed by the mass flow rate and energy flow rate. Flow propagates along the ducts in both directions allowing wave phenomena to be captured in the model
- The volume contained within the duct is split in two, with each half attributed to the adjacent capacity



! WAVE-RT Model has no discretization !
The mathematic discretization is represented by the model geometry

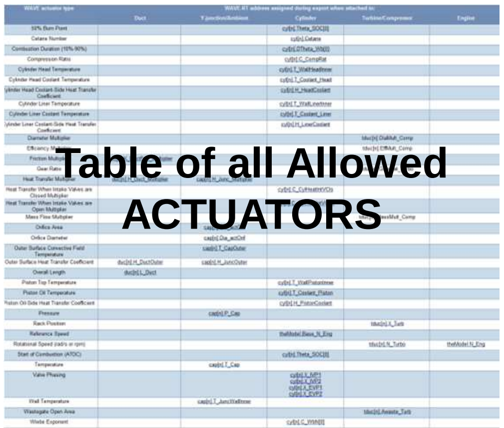
Agenda

- Introduction to WAVE-RT
  - WAVE-RT Background and Basic Information
  - Tutorial + Help
  - **WAVE-RT vs. WAVE Unsupported Objects and Features**
- WAVE Model Preparation
- WAVE-RT Model
- HiL Implementation
- Conclusions

INTERNAL UNAPPROVED DRAFT

WAVE-RT does not have so many objects, sensors and actuators as WAVE

- It is an intention for WAVE-RT not to have so many sensors and actuators – it is trying to replicate the real engine where lots of parameters have to be calculated, not sensed directly
- Regarding missing objects compared to WAVE – the aim for RT is to be as fast and as simple as possible so the calculation should not be slowed down by complicated objects
- WAVE-RT has no control strategy objects (Interpolation Maps, PID controllers, etc.) since it is driven by a driver where all the control strategy is presented
- Please have a look at Help for WAVE-RT supported:
  - Objects List: Help > WAVE-RT > WAVE-RT Files > The model_rt.c File
  - Sensor Table: Help > WAVE-RT > Using WAVE-RT > Creating a WAVE-RT Model > Exporting the WAVE-RT Model from WaveBuild
  - Actuator Table: Help > WAVE-RT > Using WAVE-RT > Creating a WAVE-RT Model > Exporting the WAVE-RT Model from WaveBuild



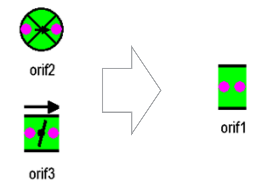
WAVE sensor type	WAVE-RT address assigned during export when attached to:				
	Duct	Y-junction	Cylinder	Turbine/Compressor	Engine
Equivalence Ratio			cyl[n].Phi		
Exhaust Gas Mass Fraction	duc[n].Z_DuctExh	cap[n].Z_CapExh	cyl[n].Z_CylExh		
FMEP					theModel.FMEP_CF
Head Coolant-Side Wall Temperature			cyl[n].T_WallHeadOuter		
Head Gas-Side Wall Temperature			cyl[n].T_WallHeadInner		
Heat Flow		cap[n].Q_Conv	cyl[n].Q_Conv		
Heat Transfer Coefficient	duc[n].H_Duct	cap[n].H_Junc0	cyl[n].H_CylWall		
Inner Wall Temperature	[illegible]	[illegible]			
Liner Coolant-Side Wall Temperature			[illegible]		
Liner Gas-Side Wall Temperature			cyl[n].T_WallLinerInner		
Mass Flow	duc[n].W_Duct			tduc[n].W_Duct_Turbo	
Outer Wall Temperature	duc[n].T_DuctWallOuter	[illegible]			
Piston Gas-Side Wall Temperature			cyl[n].T_WallPistonInner		
Piston Oil-Side Wall Temperature			cyl[n].T_WallPistonOuter		
Pressure	cap[n].P_Cap	cap[n].P_Cap	cyl[n].P_Cyl		
Rack Position				tduc[n].X_Turb	
Rotational Position (rad or deg)			cyl[n].Theta_loc		theModel.Theta
Rotational Speed (rad/s or rpm)			theModel.N_Eng	tduc[n].N_Turbo	theModel.N_Eng
Temperature	duc[n].T_Duct	cap[n].T_Cap	cyl[n].T_Cyl		
Torque				tduc[n].Tq_Turbo	theModel.Tq_Eng
Velocity	duc[n].Vel_Flow				

INTERNAL UNAPPROVED DRAFT

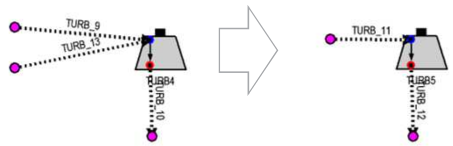
There are number of WAVE objects which have to be re-modelled (simplified) before the model can be properly converted to WAVE-RT

- Throttles and Check valves has to be turned

  into simple orifices



- Twin entry turbines

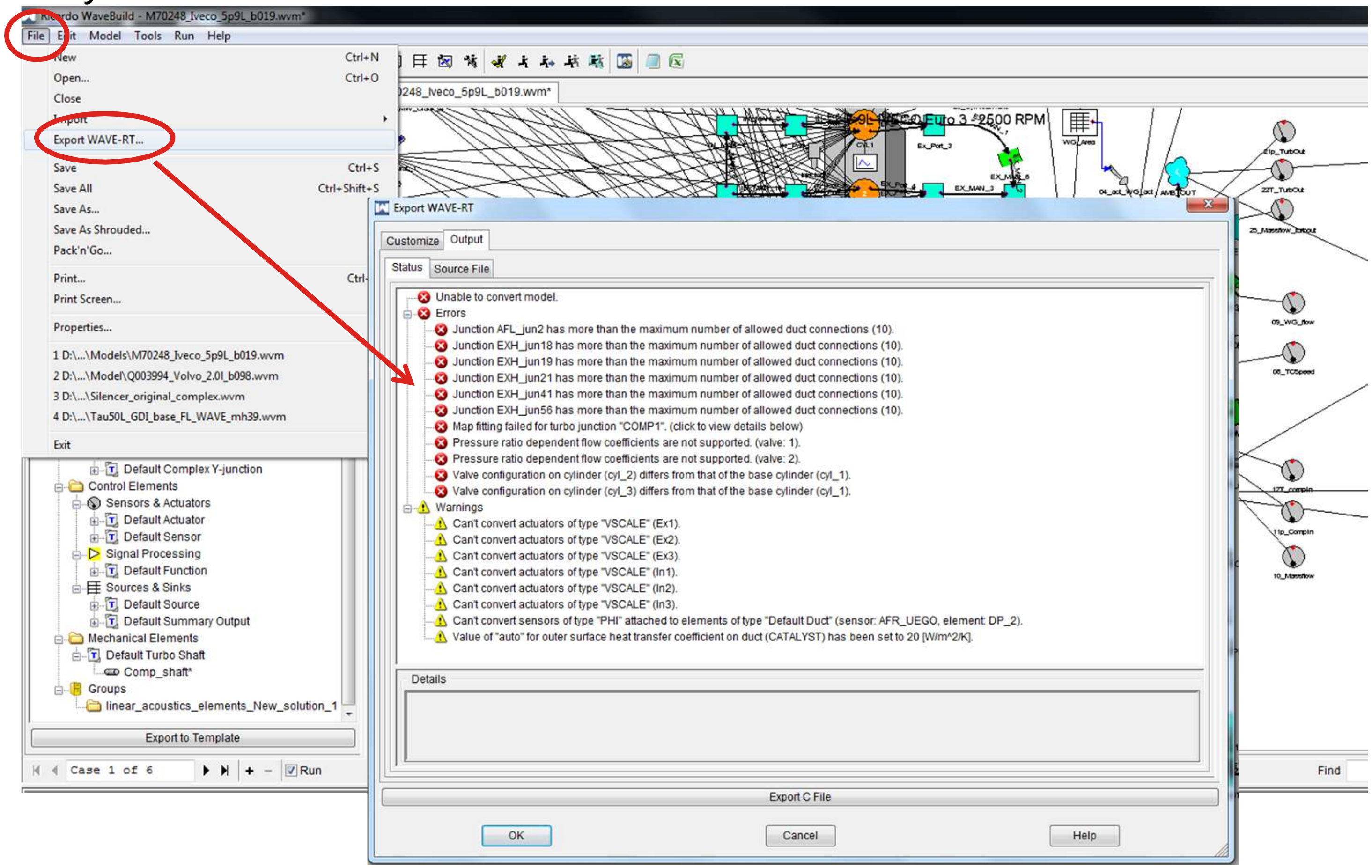


- Discretization
  - Model geometry changes might be necessary
- Thermocouples are not supported
- Burn Profile combustion
  - DI Wiebe, SI Wiebe or Multi-Component Wiebe supported

INTERNAL UNAPPROVED DRAFT

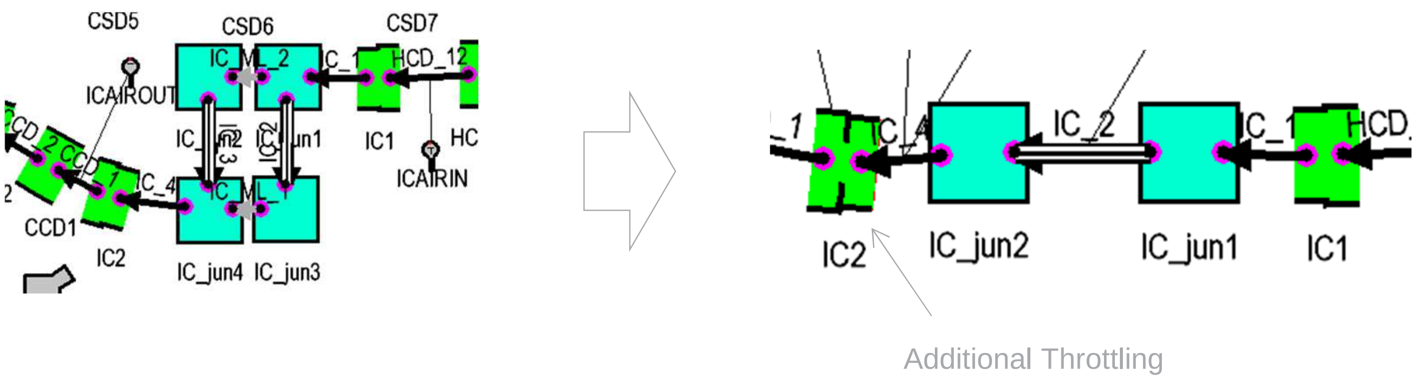
There are several WAVE objects which have to be re-modelled (simplified) before the model can be properly converted to WAVE-RT

- Turbocharger has to have an inertia
- There are more object whose are not supported
  - User can easily check:



User has to be aware that some of components are modelled or calculated with different approach in WAVE-RT

- Multiple duct representation (Intercooler, Catalysts, DPF) is slightly different in RT compared to WAVE
  - There is no multiple duct in WAVE-RT – the inner duct area is increase in WAVE-RT to be equal to multiple duct are in WAVE (done automatically during model export)
  - This is the reason that the pressure drop in WAVE-RT is usually underestimated compared to WAVE
  - The simple solution is to simplify the IC put an additional throttling downstream the intercooler



- The throttling orifice with e.g. diameter of 30 [mm] in WAVE will be giving slightly different pressure drop compared to the same orifice in RT – the reason is the modelling approach (capacities, ducts, …) – see WAVE-RT Theory
  - It is recommended to prepare an actuator to the orifice diameter which can be adjusted to get the same pressure drop

User has to be aware that some of components are modelled or calculated with different approach in WAVE-RT

- WAVE-RT uses linear function for Gamma calculation dependent only on temperature inside of the cylinder (WAVE uses more enhanced method for estimating the gamma)

$$Gamma = KGAM*T\_Cyl+GAM\_PHI.$$

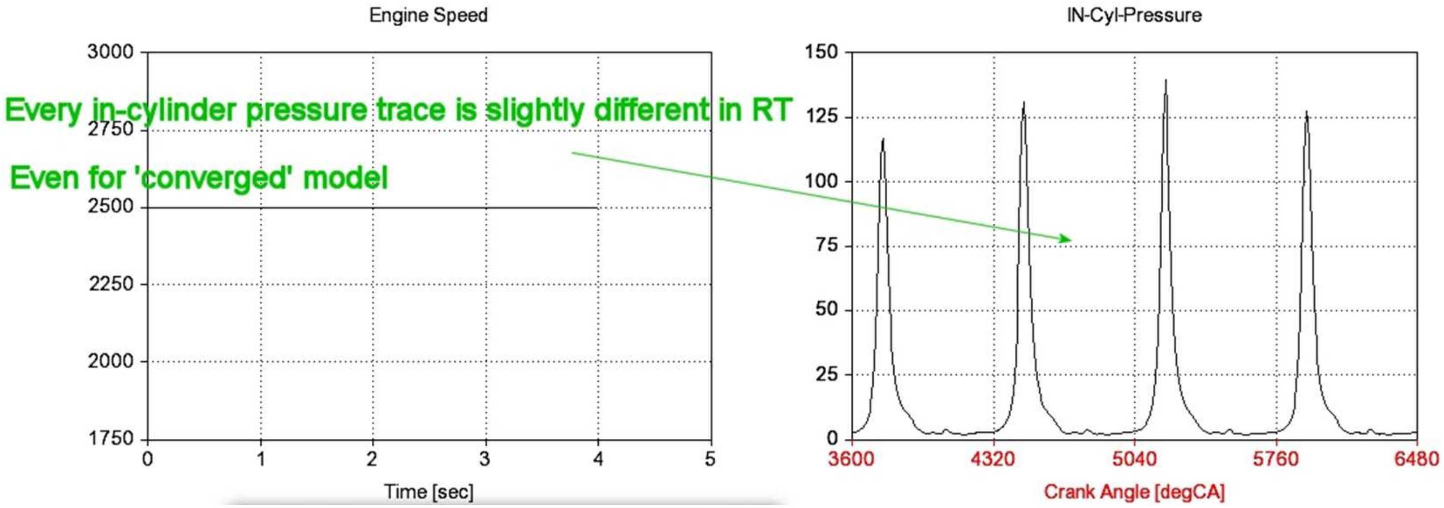
  - Where KGAM and GAM_PHI are constants (can be manually adjusted in *model*_rt.c)
  - T_Cyl is temperature within combustion chamber

- It is usually necessary to change HTVC (heat transfer multiplier when intake valves are closed) and THVO (heat transfer multiplier when intake valves are opened) in WAVE-RT to match the WAVE model results more precisely

- **All the WAVE-RT inputs and outputs are in SI units!**
  - This is done for the compatibility with what is commonly used in real ECU strategies and to avoid confusions with units

WAVE-RT uses different mathematical solver compared to WAVE

- User has to keep in mind that WAVE-RT solver is completely different to WAVE
  - Often behaves as a real engine where e.g. every engine cycle in-cylinder pressure is slightly different even the solution is converged



  - User should do an average of several in-cylinder pressures for fair comparison with WAVE similarly as it is done on test beds
- **RT is driven by time** (time in seconds on x-axis always) so if user has to calculate the crank angle if necessary
  - This is also complies to real engine ECU and their strategy where everything is based on time in seconds

INTERNAL UNAPPROVED DRAFT

WAVE-RT Hints and Tips

- The WAVE-RT model is always created from the 1st WAVE case
  - Users should keep in his mind that only 1st case parameters setting will be exported
- WAVE-RT model is sensitive to time step
  - The proper time step should be calculated – usually ~1deg CA
  - Following formula can be used: $t = \frac{\alpha}{6n}$, where $\boldsymbol{\alpha}$ is crank angle [deg CA] and $\boldsymbol{n}$ is engine speed [rpm]
  - 1degCA @ 4000rpm can be calculated as t = 1/6/4000 = 4.16e-5sec
- The engine geometry should be as simplified as possible
  - Less components leads to higher computational speed
  - Dirty side ducts, air cleaner, clean side ducts should be removed if possible and the pressure depression adjusted using an orifice
  - Entire exhaust aftertreatement should be removed (no mufflers, DPF, CAT, DOC) and the back pressure should be increased by back pressure orifice
- All WAVE-RT angles are in radians!
  - Use conversion formula: 1rad = 1degCA * PI / 180

It is important to re-run the model in the current version and make sure that the results are unchanged

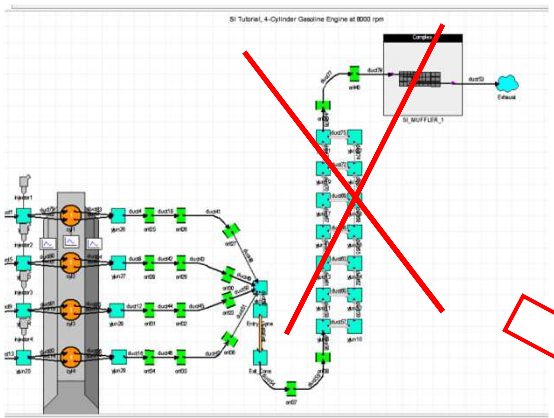
Once user starts to work in one WAVE / WAVE-RT version the version should not be changed anymore!
Lots of WAVE-RT new upcoming features and enhancements can change the results!

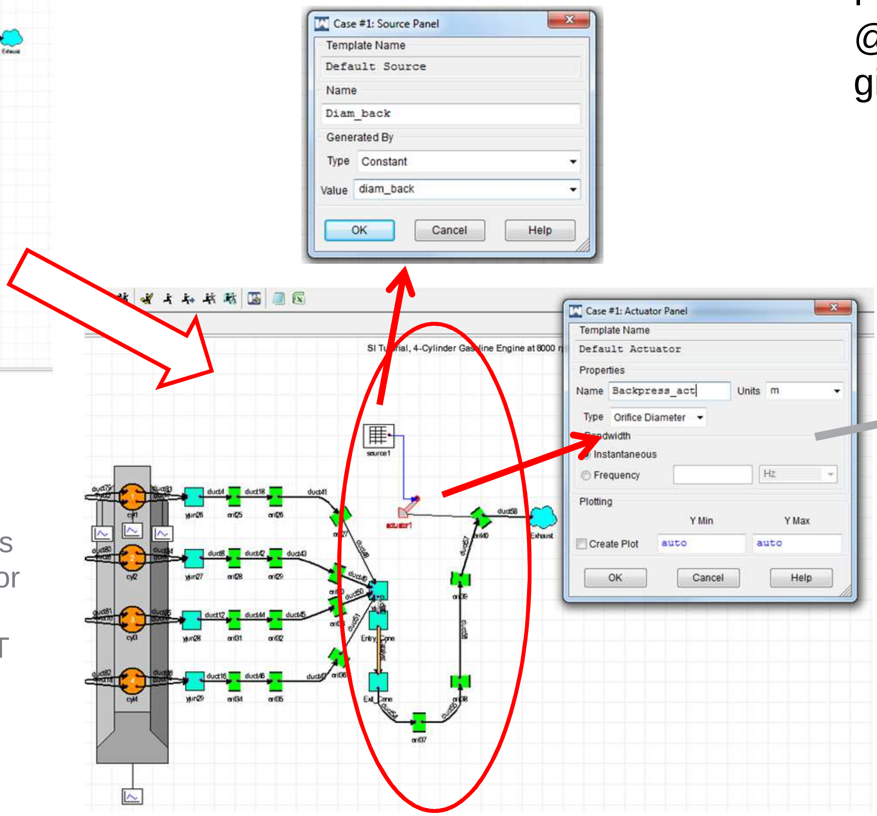
INTERNAL UNAPPROVED DRAFT

Geometry Simplification

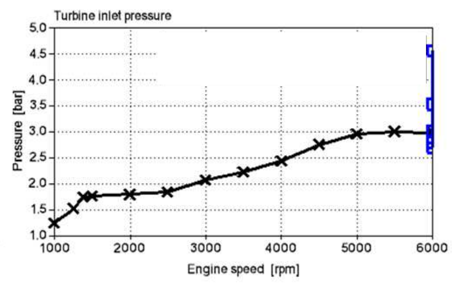
Delete the complex manifold and replace with simple ducts and oriffices

Create an actuator and use default source to actuate the diameter to get the original backpressure

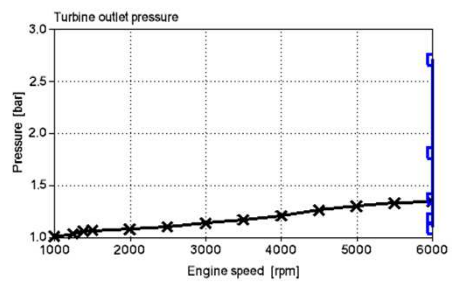




Perform a orifice diameter swing @ 6000rpm to find the diameter giving the same back pressure



The Diameter is actuated using this approach to have prepared actuator value in correct units within Constant Panel Setup ready for RT driver in correct units



Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
  - **Results Plot Template**
  - Not Supported objects remodelling
  - Sensors and Actuators Setting
- WAVE-RT Model
- HiL Implementation
- Conclusions

Baseline model needs to be inspected and understood before user starts any adjustments leading to RT conversion

- Ricardo uses predefined plot templates for various engines
- It is recommended to use a build log strategy naming models with _b001, _b002, …
- Build log file should be created to include all the changes made and model dependences

Creating Buildlog.xlsxy template

– Have a look at build log structure

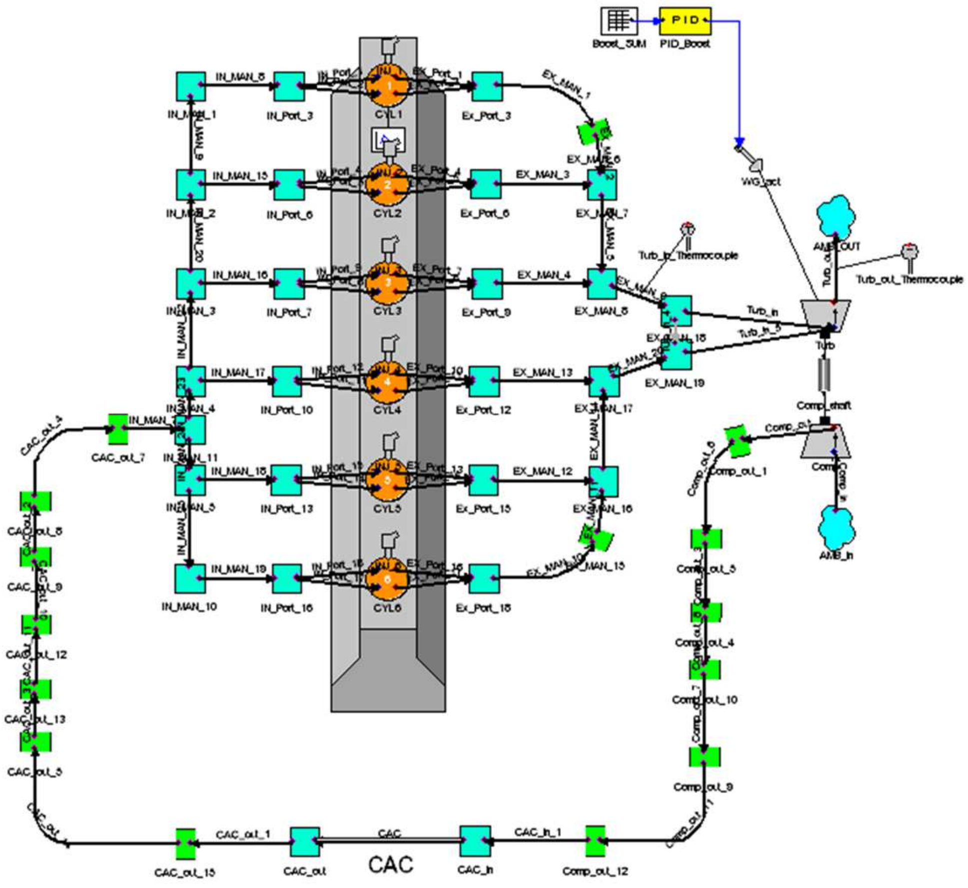
Inspect the model and introduce yourself with the model

– Have a look and check the model – recommended modelling techniques, combustion, heat transfer, FMEP, initial and boundary conditions, Parameter view, …

INTERNAL UNAPPROVED DRAFT

Picture below shows the baseline model

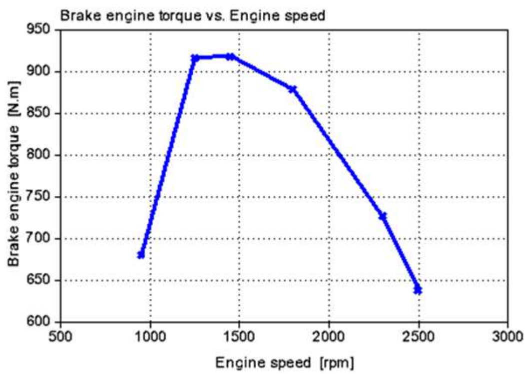
5.9L IVECO Euro 3 - 2500 RPM

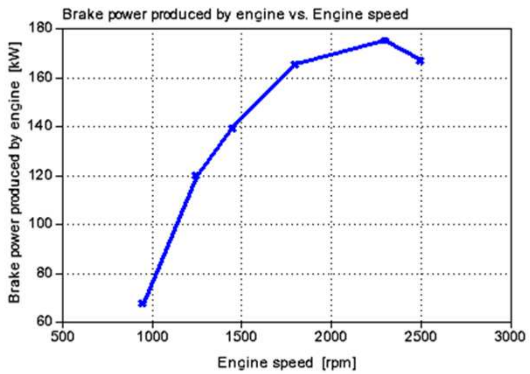


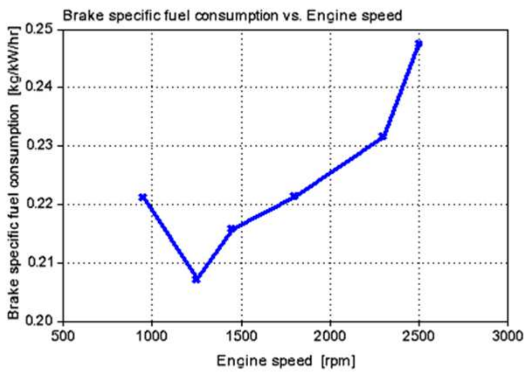
It is important to create a result plot template in WavePost to make sure that the changes applied to WAVE model won't change the resulting parameters

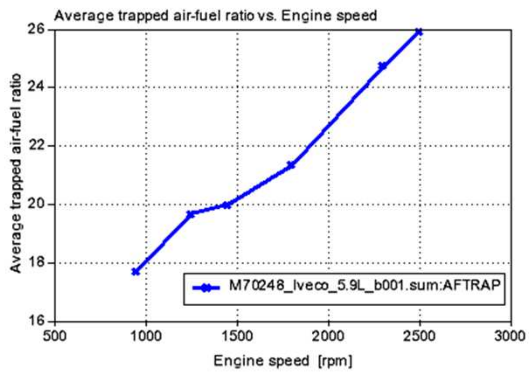
Run WavePost and Create following plot template:

- Use Plot Manager
- Use Multiple Properties
- All major parameters
- Reread = CTRL+R



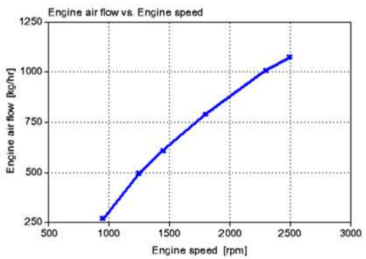


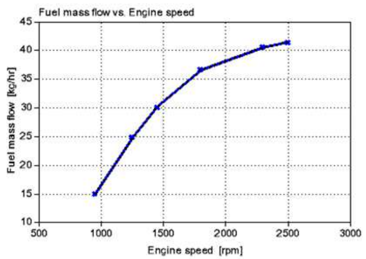


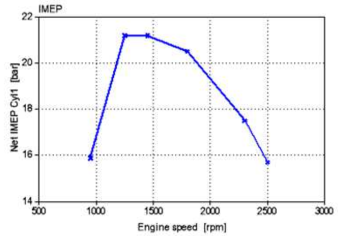


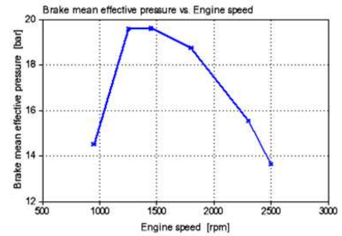
INTERNAL UNAPPROVED DRAFT

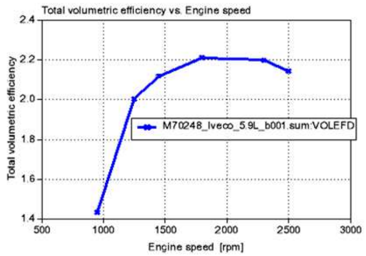
Plot Template: Page 2-5

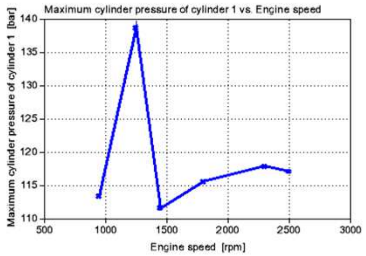


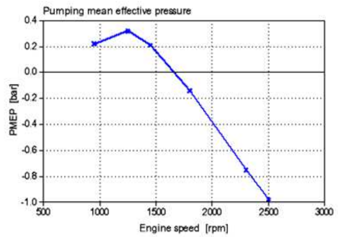


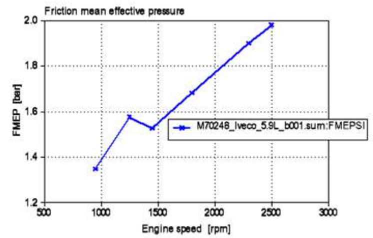


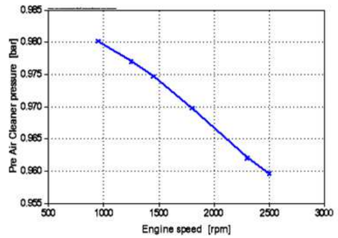


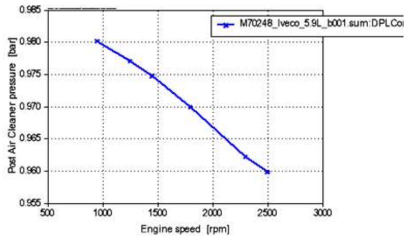


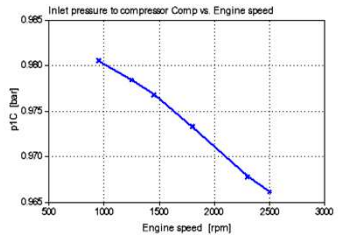


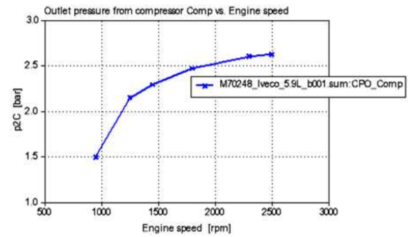


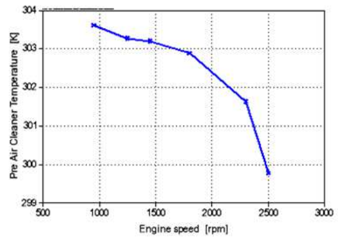


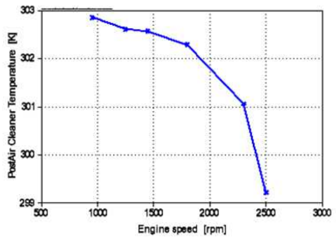


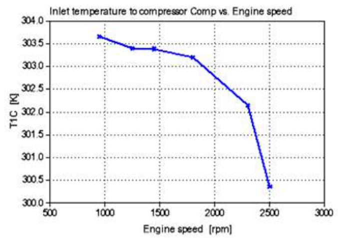


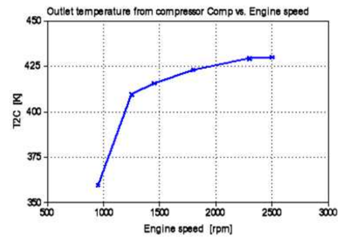




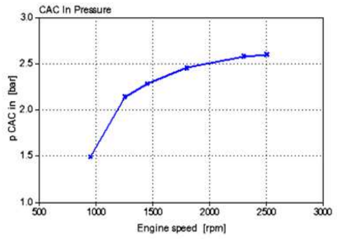


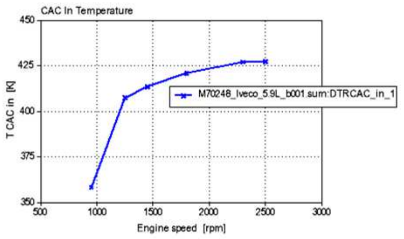


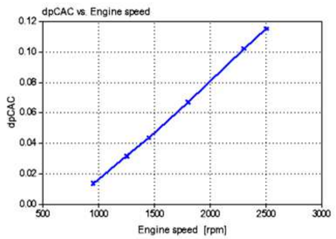


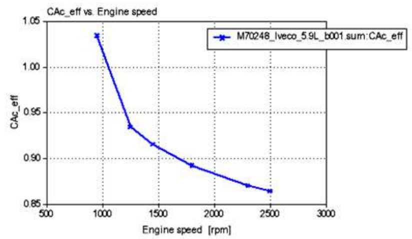


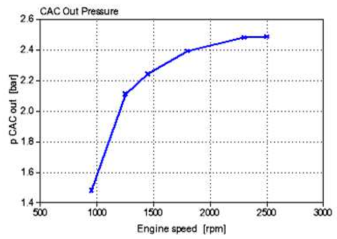
Plot Template: Page: 6-9

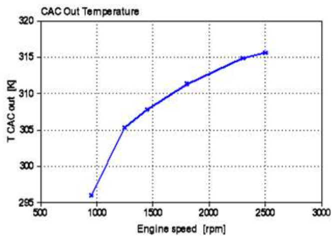


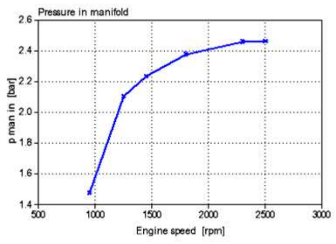


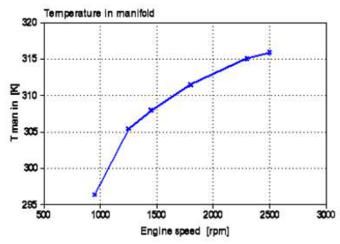


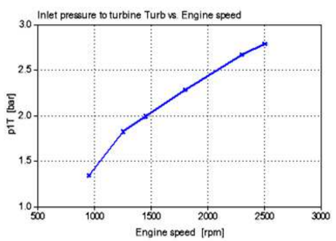


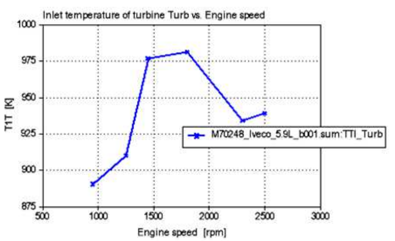


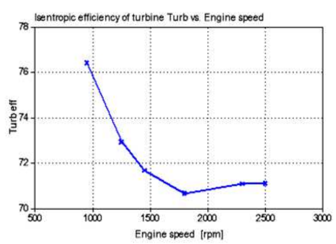


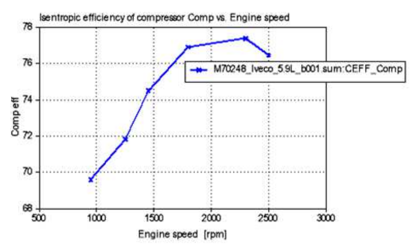


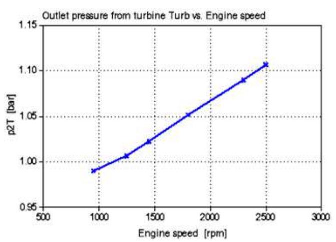


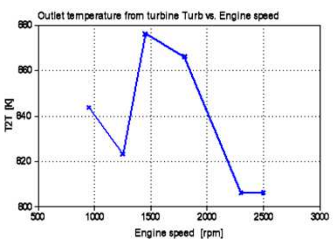


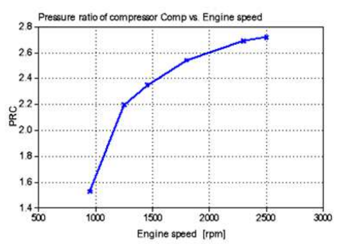


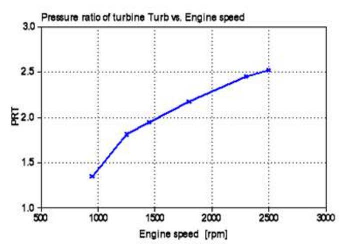






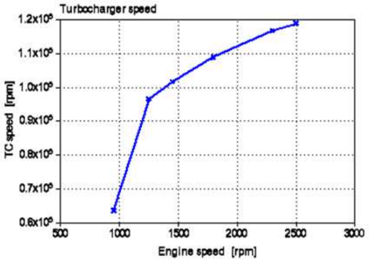


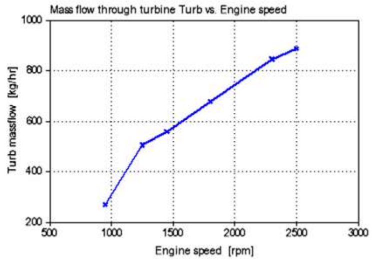


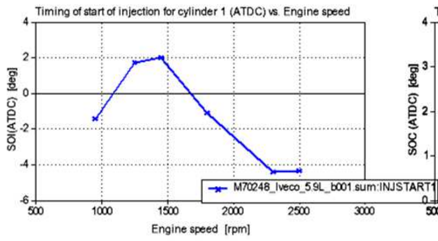


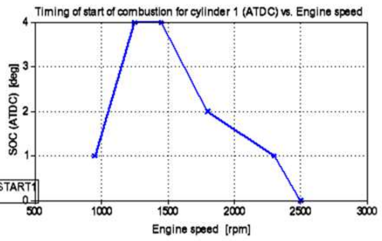
INTERNAL UNAPPROVED DRAFT

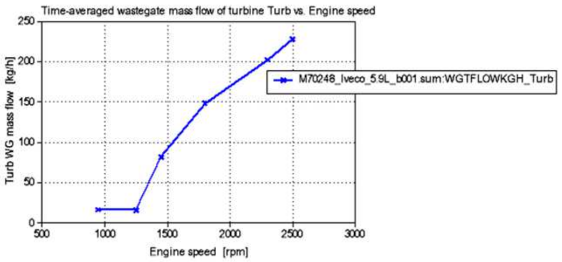
Plot Template: Page: 10-13

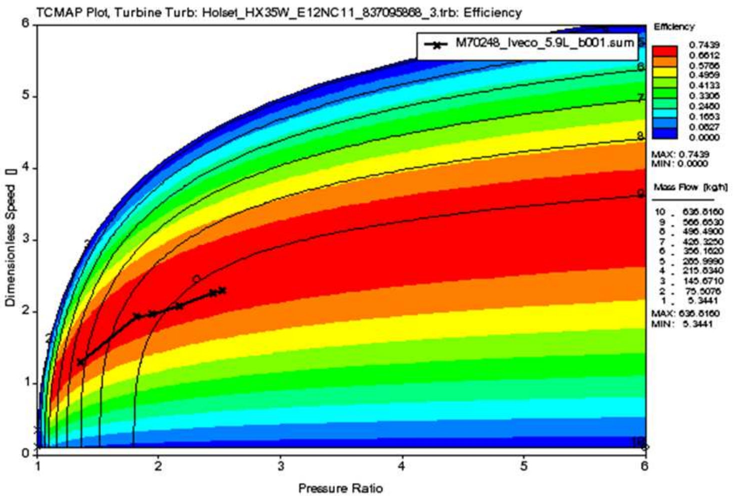


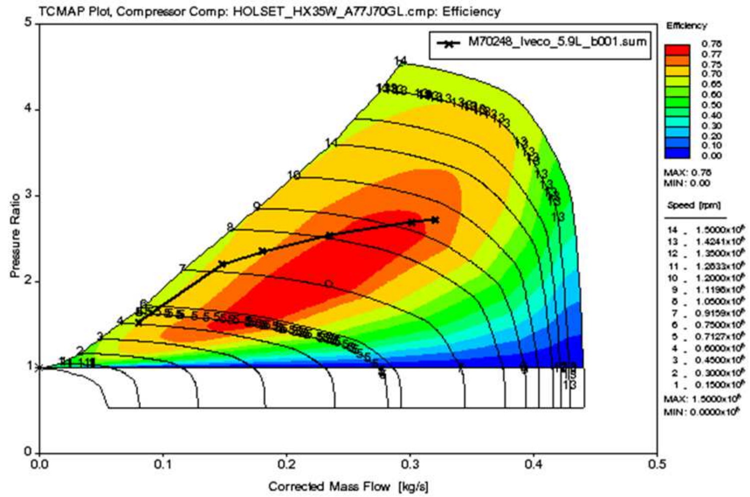












It is important to plot all these plots, they give full model view – understanding!

- This template will be used during entire model conversion process
- The different builds will be plotted over each other and compared to the baseline WAVE model

Practically every OEM has its own list of plots

- This template can be modified to fulfill particular needs

Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
  - Results Plot Template
  - **Not Supported objects remodelling**
  - Sensors and Actuators Setting
- WAVE-RT Model
- HiL Implementation
- Conclusions

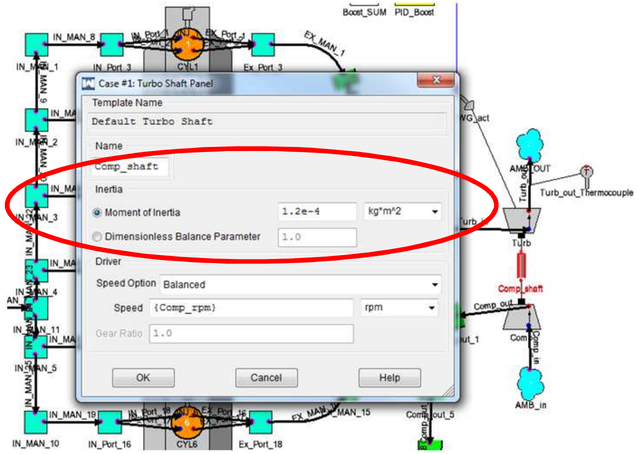
Notes to RT not supported objects

- It is recommended to use predictive Chenn-Flynn model which respects mean piston speed and Pmax for estimating of FMEP

- Need to change complex controlling strategy (boosting, PI actuators,...).

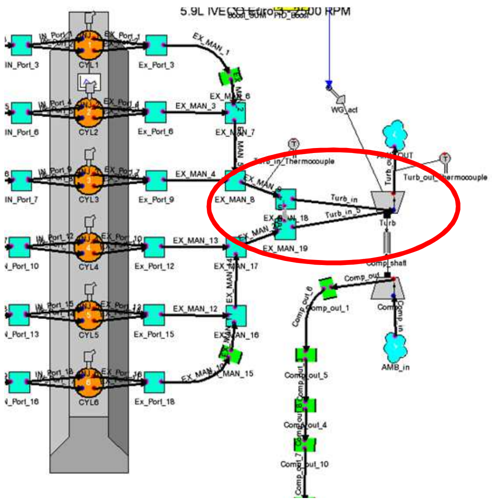
INTERNAL UNAPPROVED DRAFT

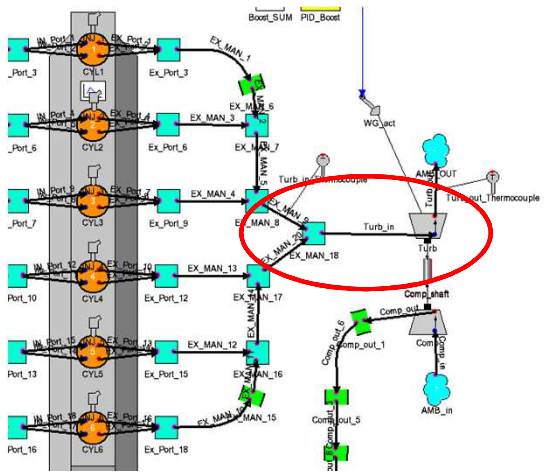
TC moment of inertia, twin entry – single entry turbine



– Checking the **TC moment of inertia** in Turbo shaft object (value corresponding to size of TC)

– Change the turbine inlet manifold from **twin entry** to **single entry** turbine respecting the volume and length of modified manifold



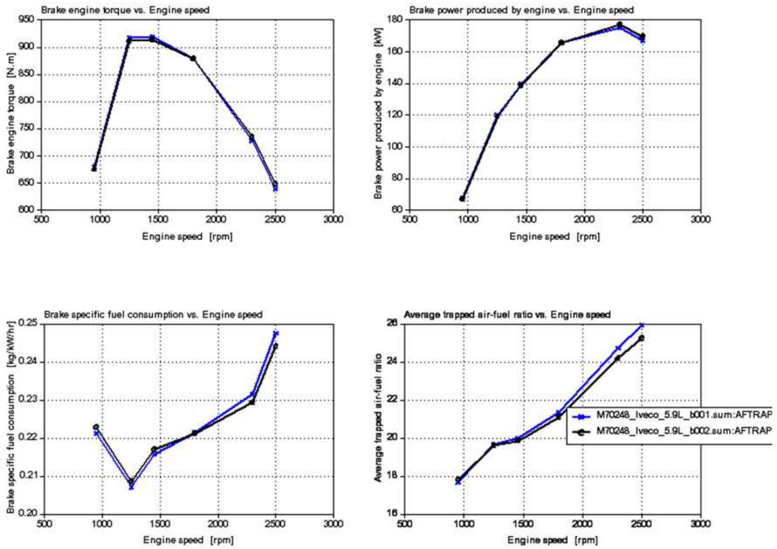


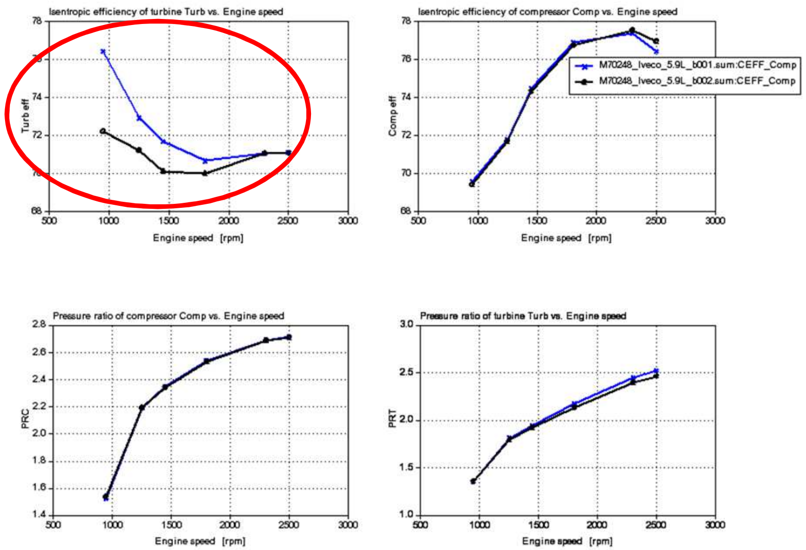
INTERNAL UNAPPROVED DRAFT

TC moment of inertia, twin entry – single entry turbine

- The results are slightly different compared to baseline due to change in turbine entry but still satisfied

- We can expect drop in turbine effciency with single entry turbine





INTERNAL UNAPPROVED DRAFT

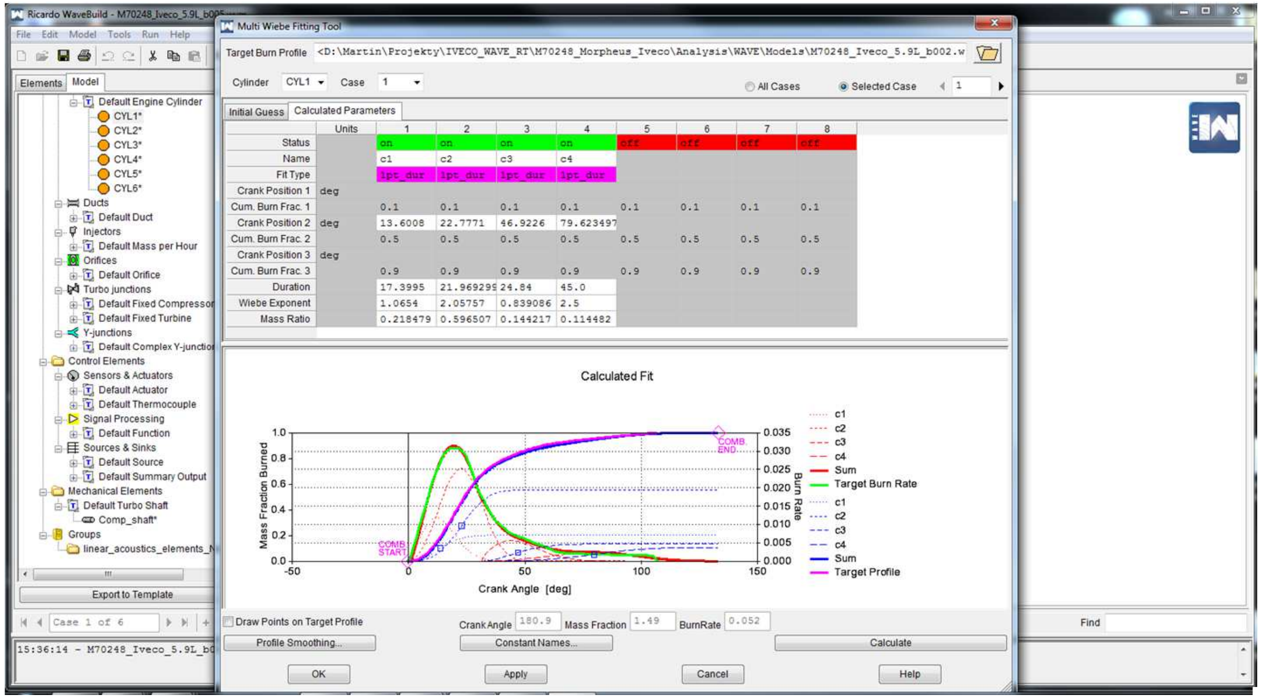
Multi Wiebe curves setting

– Combustion curves represented by Wiebe curves prepared in Multi Wiebe submodel.

– The new utility MultiWiebe Fit was tested (new WAVE release)

– Wiebe curves for each engine
speed were set
(three or four wiebe complex)

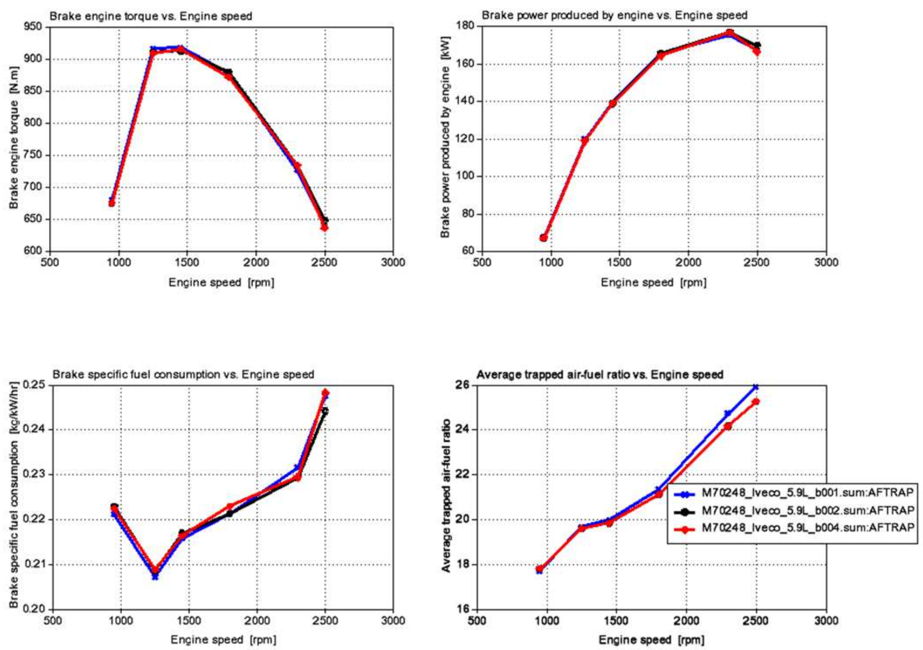
Each curve represented by A50,
combustion duration,
Wiebe exponent and mass fraction



INTERNAL UNAPPROVED DRAFT

Multi Wiebe curves setting

- Results must be comparable.
- Be carefull with mass fraction if modifying once generated wiebe curves by multi wiebe fit utility
  The sum of all fractions has to be equal „**1**“

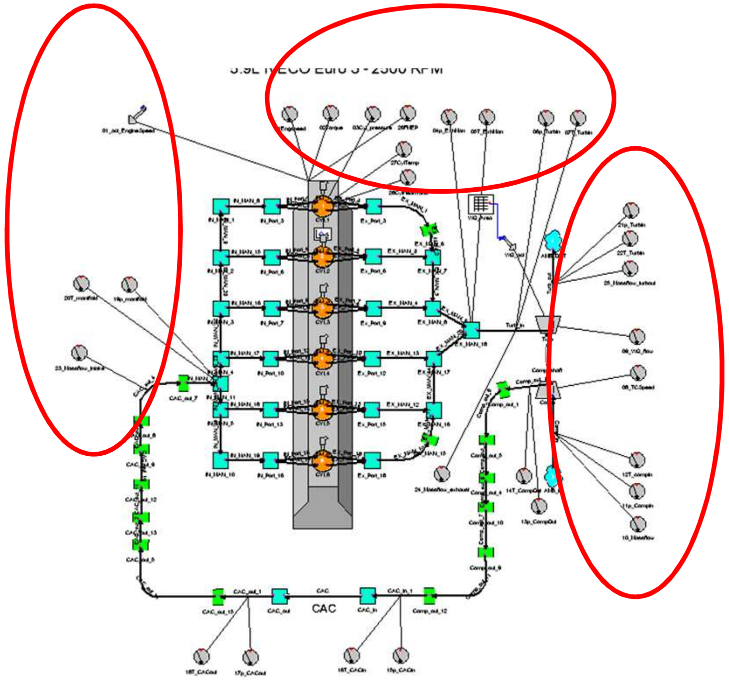


Sensors and actuators

Sensors and actuators adding

- Inlet and outlet pressure and temperature sensors added
- Engine speed actuator added

- It is recommended to build new Wave Post template based on the sensor values for comparison with WAVE RT results



INTERNAL UNAPPROVED DRAFT

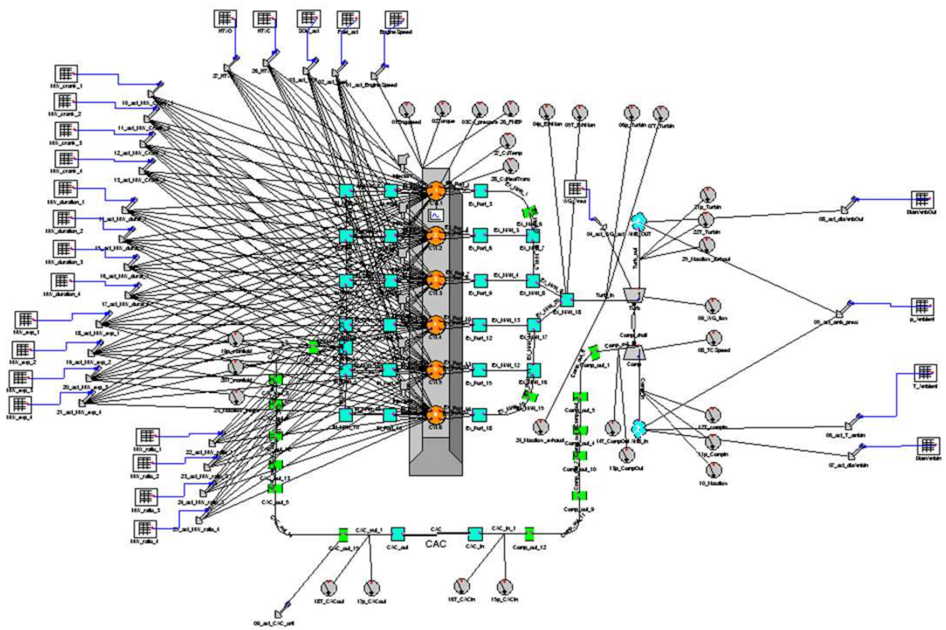
Fuel injectors changed to mg/inj, sensors and actuators added

Remove fuelling injectors (they are not WAVE-RT supported) and replace them by kg per injection injectors

- Add fuelling actuator in kg
  - Use Actuators_Calculation.xlsx including the actuator calculation transferring units
- Add SOI, CAC and
- Wiebe actuators controlling injection and combustion, HTVC, HTVO, engine speed and ambient control actuators

!

- In case of indirect injection Periodic flow rate actuators are suitable for WAVE RT conversion, where injection rate is in kg/hr.
- Therefore it is needed to manualy recalculate the value for WAVE RT from kg/hr to kg/injection

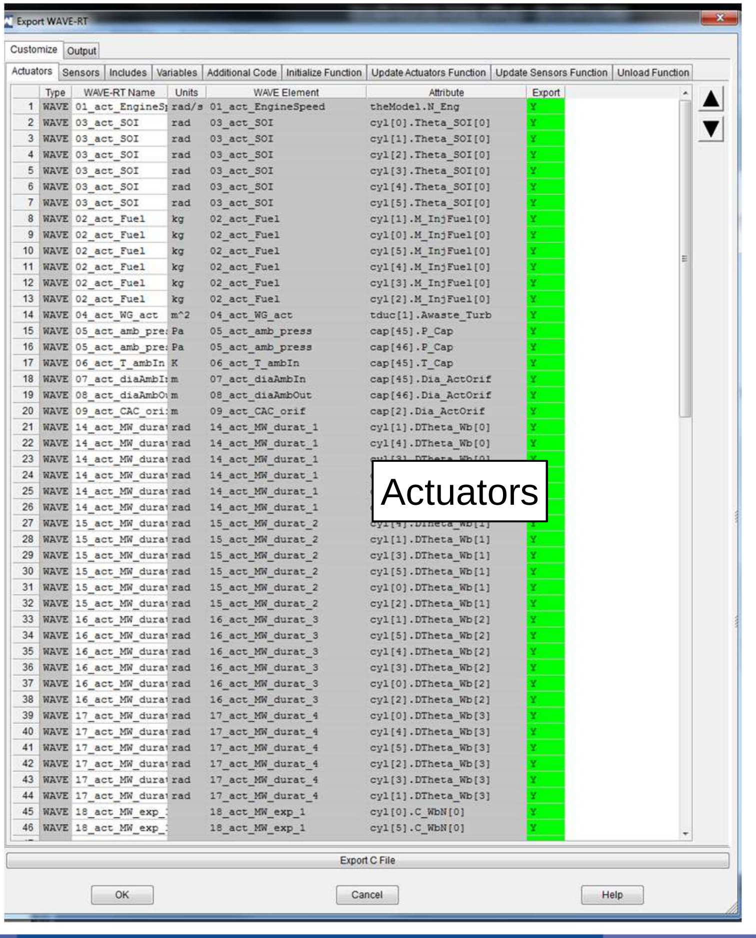


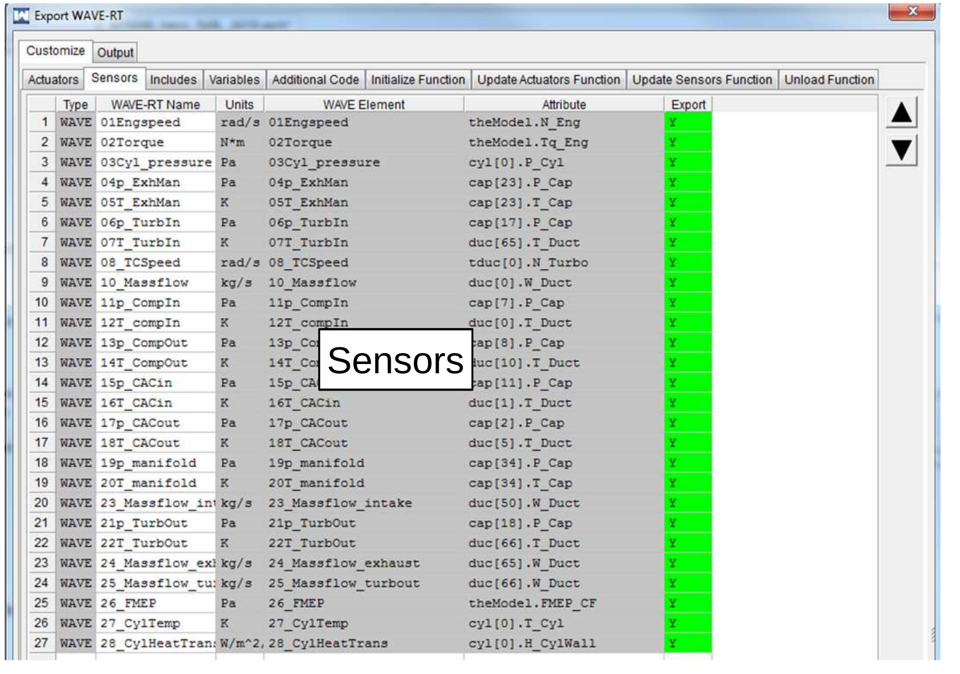
Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
  - Results Plot Template
  - Not Supported objects remodelling
  - **Sensors and Actuators Setting**
- WAVE-RT Model
- HiL Implementation
- Conclusions

Actuators added, number for each actuator and sensor added

– Add Numbers in front of each actuator and sensor
  - This is done to prevent mixing the actuator and sensor list while adding additional sensors or actuator (whose will be automatically added at the end)





Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
  - **The Simplest C++ Driver Model without any Actuators**
  - The Simplest Simulink Driver Model without any Actuators
  - Simulink with all actuators @ 2500rpm
  - Batch Simulink
  - WAVE-RT Model Validation
- HiL Implementation
- Conclusions

INTERNAL UNAPPROVED DRAFT

WAVE-RT conversion 1st case, C++ driver, no actuators

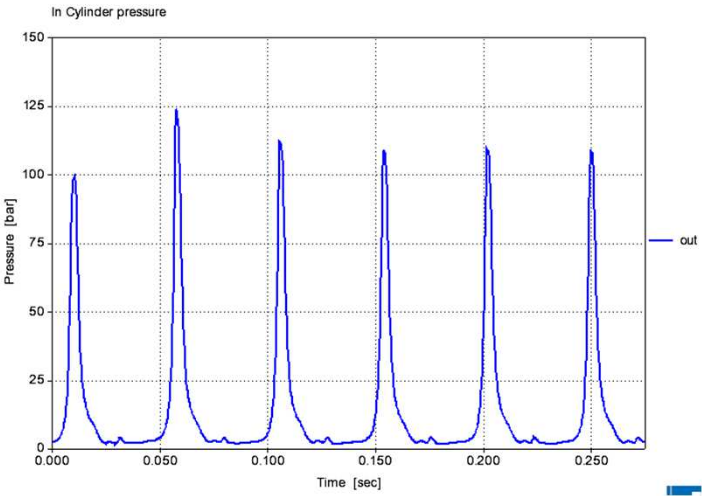
Prepared model template with only engine speed actuator can be used

Run !wrt_compile.bat:

- It includes compile script for 64bit platform
- Change it to 32bit if needed

Run !run_model.bat:

- It includes RT running script
- Use Rplot prepared Plot.rp file for looking at in-cylinder pressure traces
- Inspect the sensitivity on time step
  - Try time step: 10-70e^{-6} sec



Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
  - The Simplest C++ Driver Model without any Actuators
  - **The Simplest Simulink Driver Model without any Actuators**
  - Simulink with all actuators @ 2500rpm
  - Batch Simulink
  - WAVE-RT Model Validation
- HiL Implementation
- Conclusions

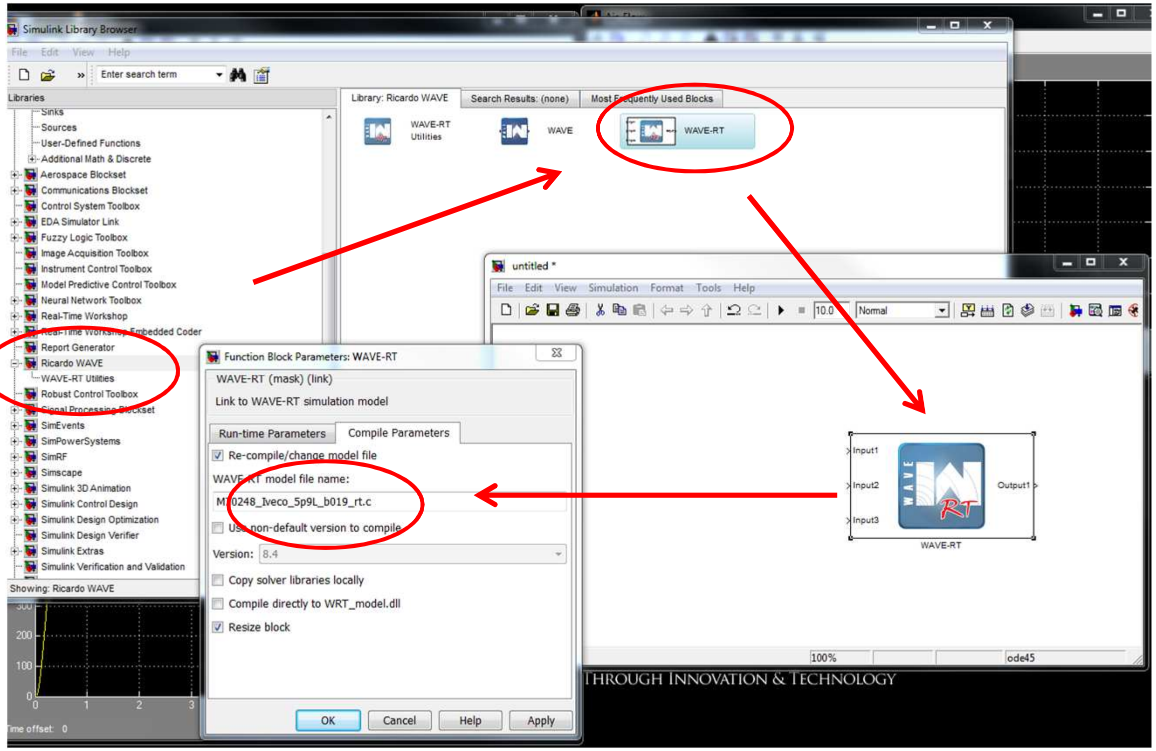
INTERNAL UNAPPROVED DRAFT

WAVE-RT conversion 1st case, Simulink Driver, No Actuators

Run Matlab and set the path to Ricardo WAVE RT

– File/Set Path/Add folder – Program Files/Ricardo/WAVE/8.4/Matlab

- Open Simulink Library Browser and build simple model using Ricardo WAVE libraries and re-compile the model.

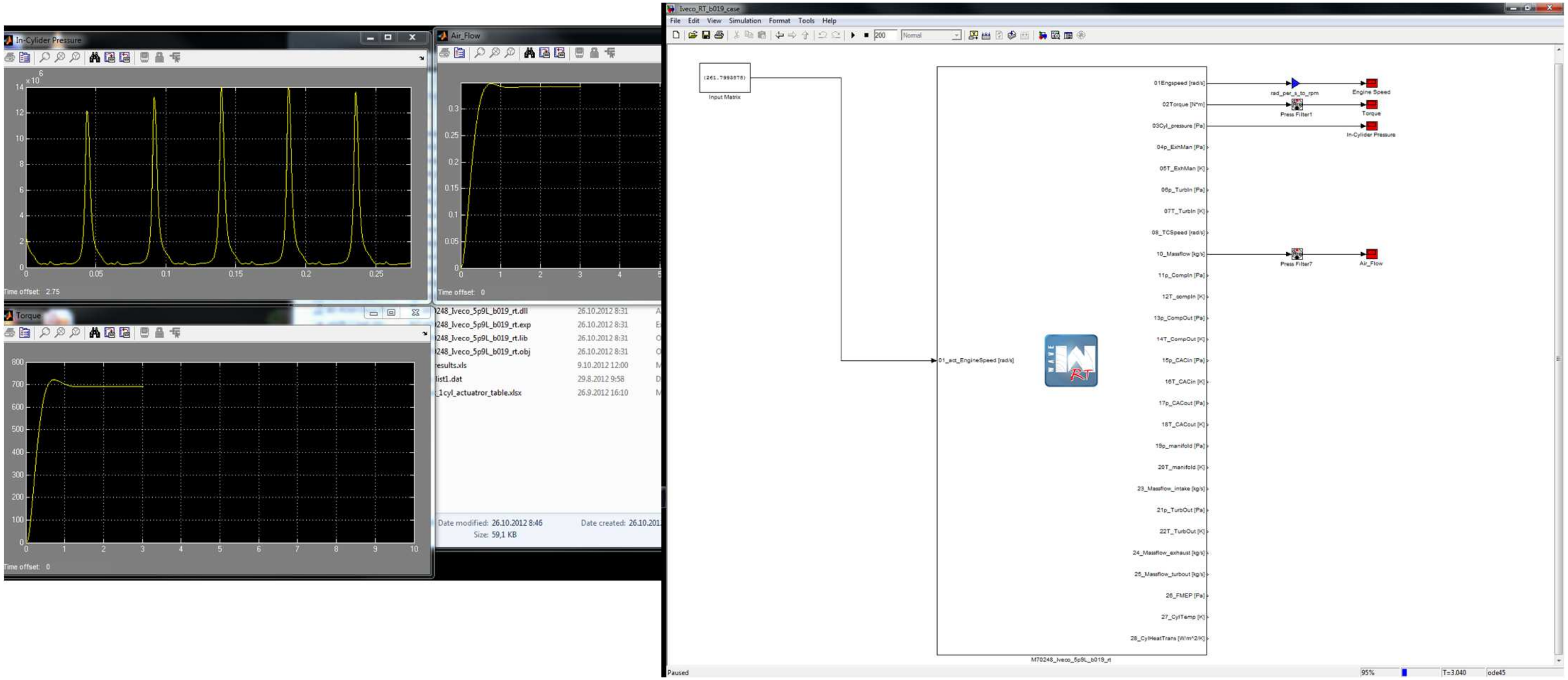


- For detailed info see the help (slide 9)

WAVE-RT conversion 1st case, Simulink Driver, No Actuators

Using of already prepared model with only engine speed actuator activated and copy all the necessary files (*_rt.C, *.mdl)

– Re-compile the model and run it

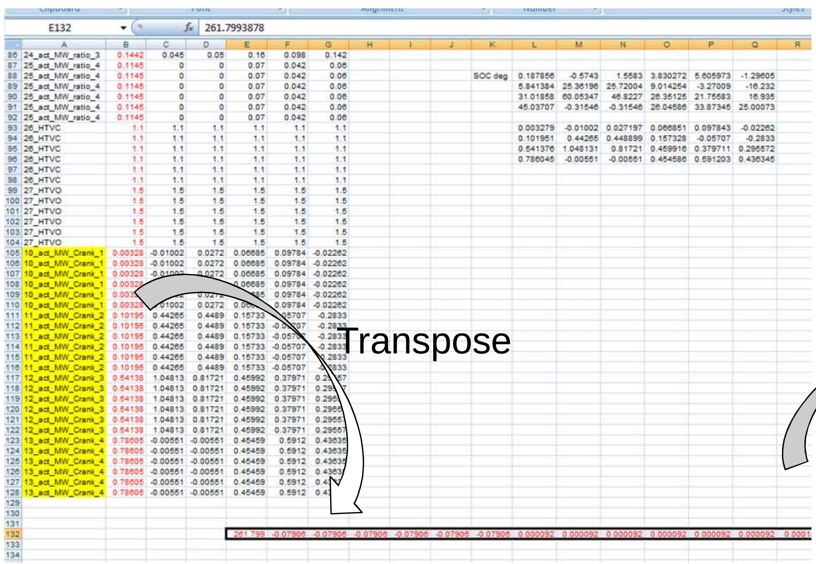


Agenda

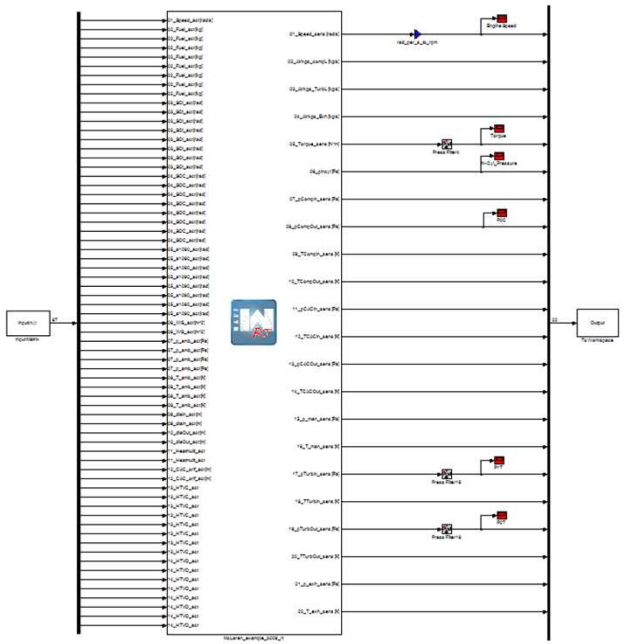
- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
  - The Simplest C++ Driver Model without any Actuators
  - The Simplest Simulink Driver Model without any Actuators
  - **Simulink with all actuators @ 2500rpm**
  - Batch Simulink
  - WAVE-RT Model Validation
- HiL Implementation
- Conclusions

WAVE-RT conversion 1st case, Simulink Driver with all actuator

- Actuators_table.xlsx includes all actuator values
- This is the advantage of using Constant Table values for actuating WAVE model
  - This prevents conversion units errors
- Transpose the 2500 [rpm] and paste it to Simulink model as an input matrix



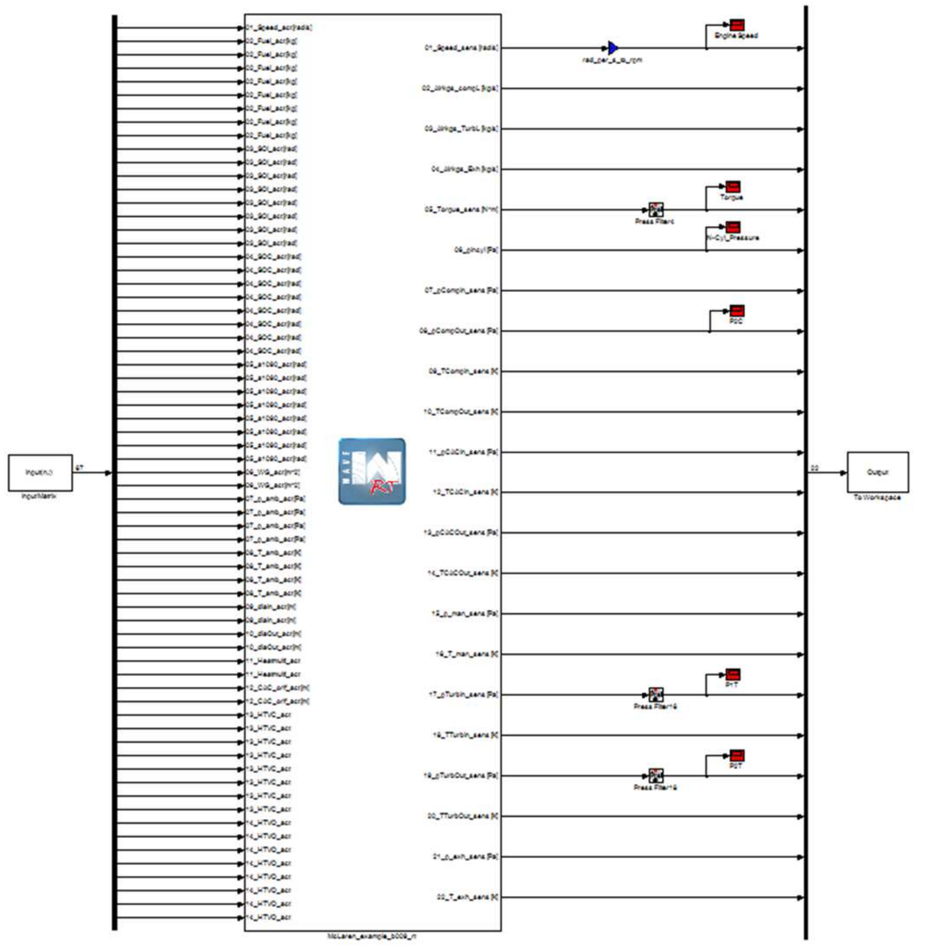
Paste to Simulink



INTERNAL UNAPPROVED DRAFT

WAVE-RT model validation

- Model validation at one engine speed



Agenda

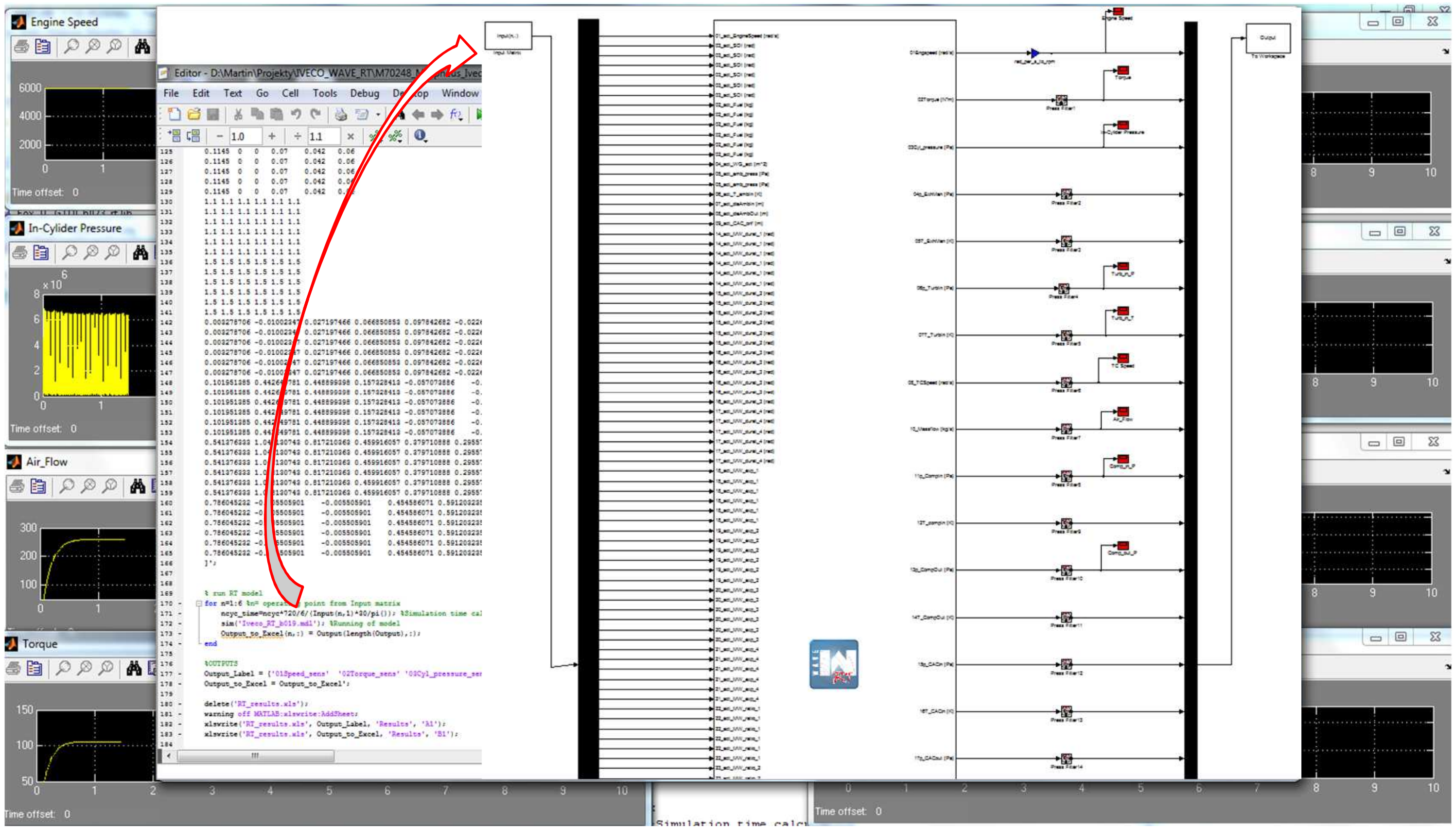
- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
  - The Simplest C++ Driver Model without any Actuators
  - The Simplest Simulink Driver Model without any Actuators
  - Simulink with all actuators @ 2500rpm
  - **Batch Simulink**
  - WAVE-RT Model Validation
- HiL Implementation
- Conclusions

INTERNAL UNAPPROVED DRAFT

WAVE-RT – Simulink Batch Run all cases - Subsystem for parameters calculation added

Template Simulink Batch File provided

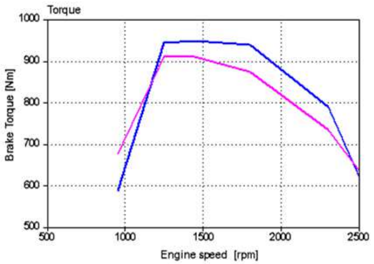
- The loop runs Simulink driven WAVE-RT model with selected number of cycles for chosen cases

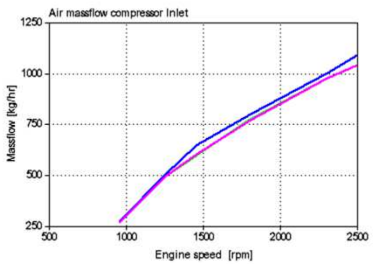


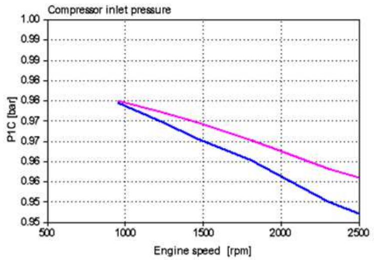
WAVE-RT – Simulink Batch Run all cases - Results

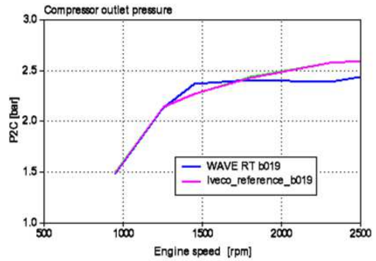
Batch file saves RT_results.xls including all sensor output

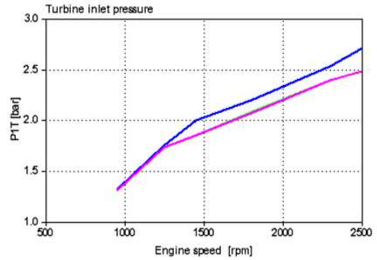
- The WAVE-RT results are added into WavePost Template
  - Adjust all plots using WAVE **sensors** not summary variables is neccesary to be compatible what we are getting from WAVE-RT!
  - Comparison of WAVE and non-validated WAVE RT.

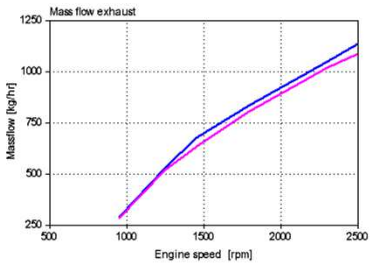


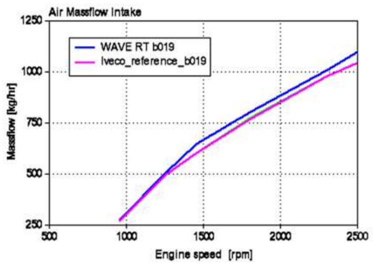


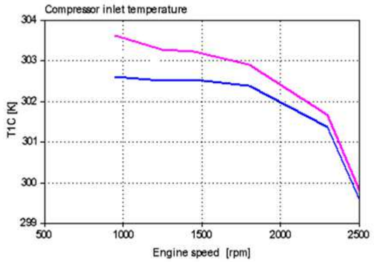


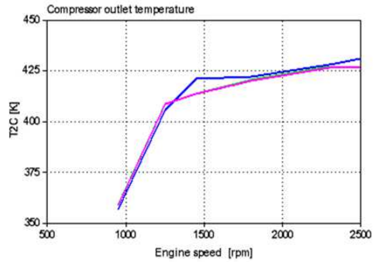


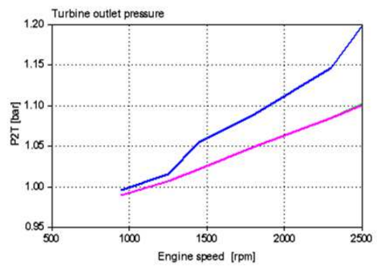












Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
  - The Simplest C++ Driver Model without any Actuators
  - The Simplest Simulink Driver Model without any Actuators
  - Simulink with all actuators @ 2500rpm
  - Batch Simulink
  - **WAVE-RT Model Validation**
- HiL Implementation
- Conclusions

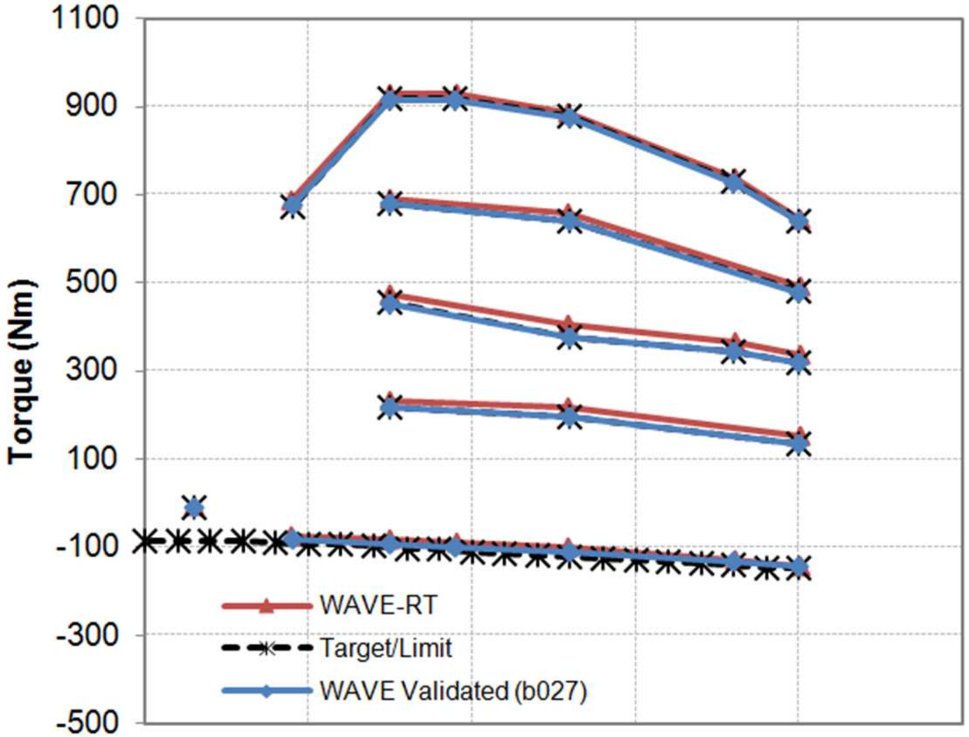
INTERNAL UNAPPROVED DRAFT

WAVE-RT – Full Load and Part Load validation

User can use standard model validation techniques to match the baseline WAVE model

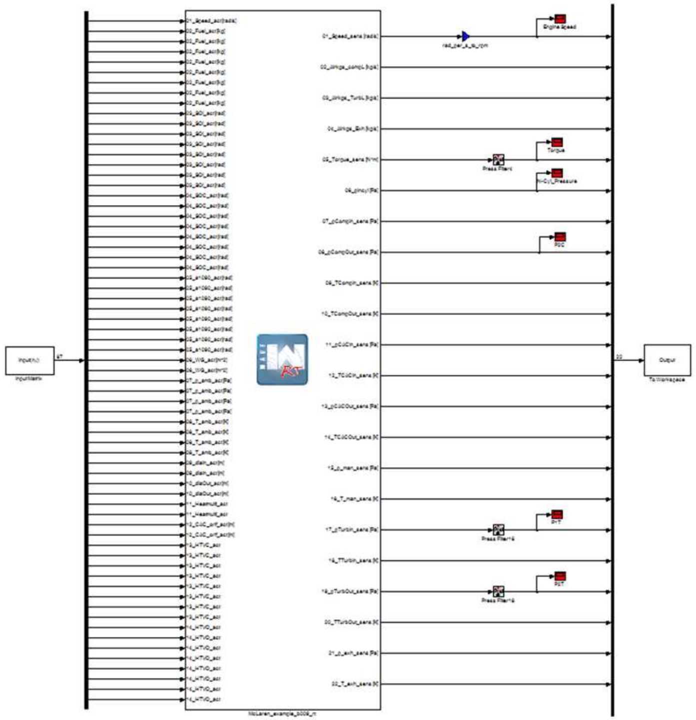
- Validation of boundary condition (p1C, p2T)
- **Diameters** are adjusted to match the pressures

- It is important to validate WAVE-RT model not only at full load but also at part loads, idle and motoring mode



Validated WAVE-RT model is prepared for the HiL implementation

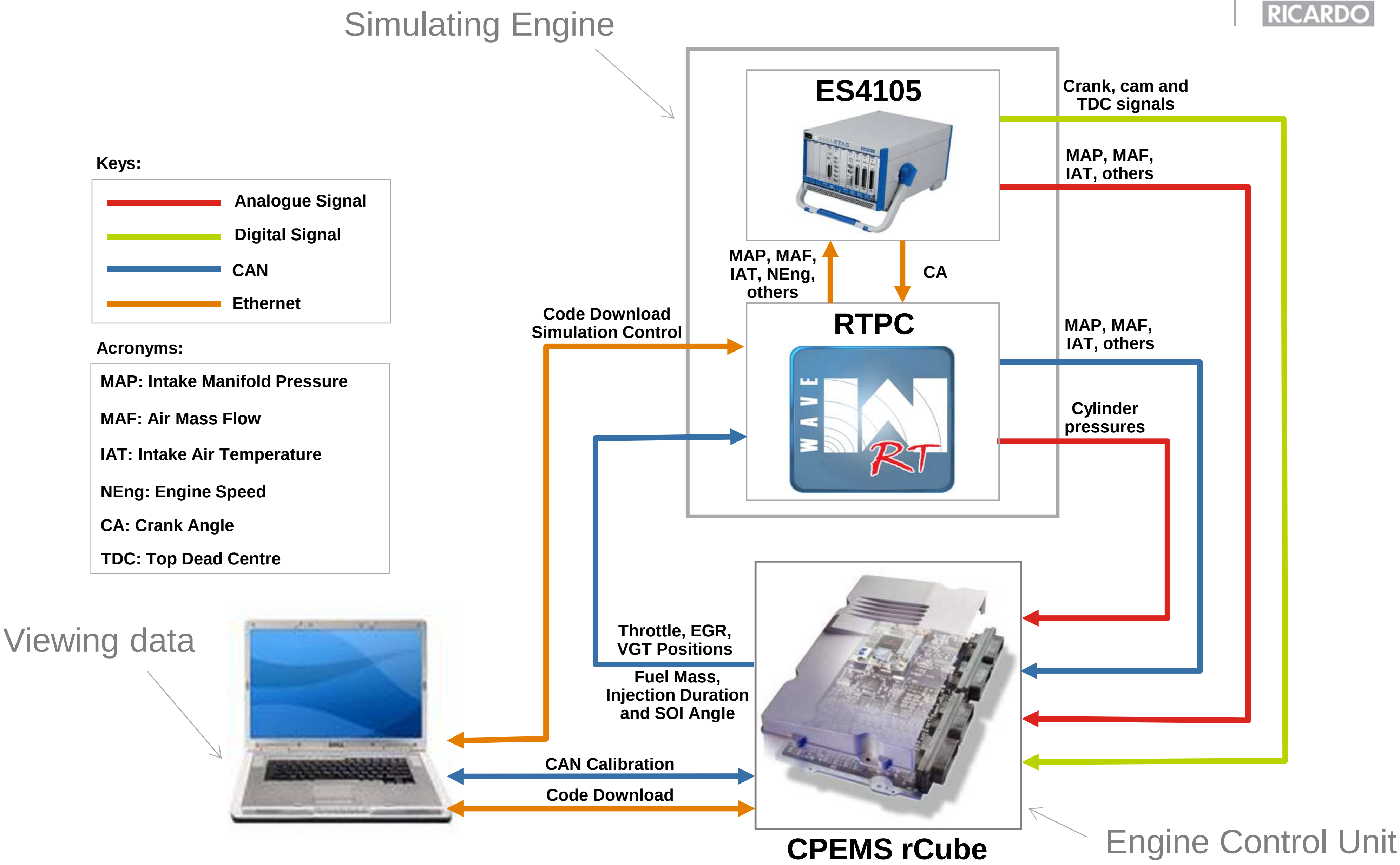
- Transferring WAVE-RT from fluid department to C&E (Control and Electronic) department where the actual HiL implementation is done
- Major of the data from input matrix is replaced by interpolation tables



Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
- **HiL Implementation**
- Conclusions

System Layout



Key Information (1)

RTPC

- ETAS Real-Time PC (RTPC) can be installed on any PC, demo spec
  - Intel Core2 Duo @ 2.66 GHz
- Runs LABCAR-RTPC v4.0.1 HiL software running the WAVE-RT application
- 2 PCI interface
  - Meilhaus Electronic 16bit DAC, ME-6000/4, sending Cylinder pressure signals
  - Ixxat CAN interface, iPC-I XC16, to / from the rCube

WAVE-RT Engine Model

- 2l 4 Cylinder Turbocharged Diesel with EGR and Aftercooler
- WAVE-RT and rCube models developed on Intel Core2 Duo @ 2.20 GHz PC
- Running on 60 us sample time (Half of this for I/O processing)
- At least up to 4500rpm
  - RTPC sample time reduces crank resolution around this engine speed

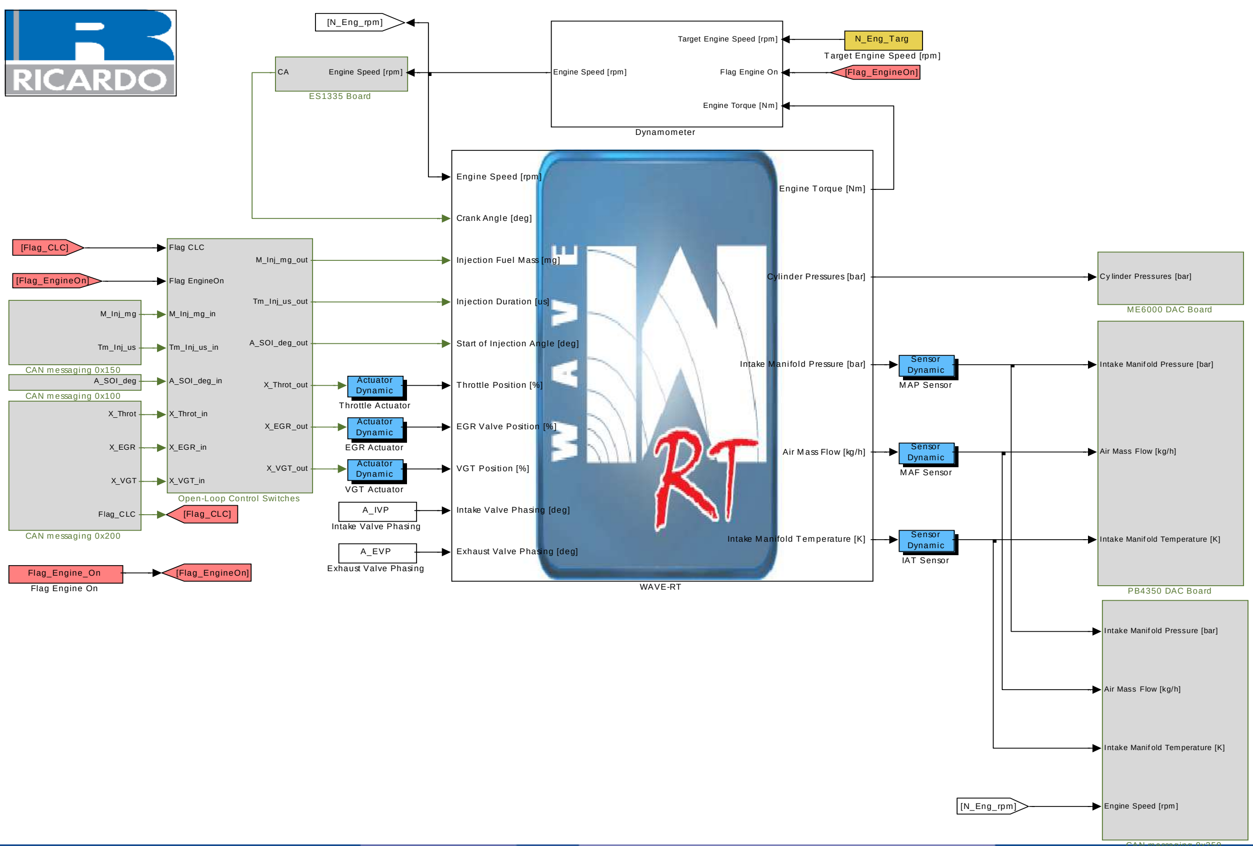
ES4105

- VME (“Versa Module Europe”) board system
- ES1130.3 – Simulation controller Ethernet to RTPC
  - Sends Crank Angles
  - Receives Engine speed, temperatures, pressures etc
- PB4350DAC1.1 (on ES1651.1)
  - Analogue out, 10 ch, 14-bit resolution
  - Generates MAF, MAP to rCube
- ES1335
  - 6 signal generators for speed synchronous signals Crank, CAM, TDC
  - Up to 30,000rpm with 0.011 degCA resolution
- ES1321.1 – PWM board NOT USED
- ES1336.1 – Angle synchronous measurement NOT USED

Key Information (2)

RTPC – HiL Model

- The below diagram shows the model running within the RTPC
  - Grey-blocks are I/O processing and control connection to outside world (~20us required)
  - WAVE-RT runs at around 25us
  - Total step time of 60us is the sum of the two and gives some room for calculation overrun



Key Information (3)

CPEMS rCube – Specification

- Based on Ricardo's own flexible Rapid Prototyping Engine Management System
- Development using either ETAS ASCET or MATLAB/Simulink tool chains
- Calibration conducted over CCP using ETAS INCA software
- rCube Highlights (Base Variant 3)
  - 6- 58V input supply
  - Temp Range (-40/+85^{o}C)
  - 24 Analogue Inputs - S/W configurable 0-5V or 0-20V incl 4 'Fast' Analogue Inputs with combined max 132kHz sampling rate
  - 24 General Purpose Digital I/O (S/W configurable in/out + pull up/pull down/no pull) 12V O/P 0-5V or 0-12V Input
  - 4 Event triggered fast digital inputs (cause s/w interrupt)
  - 4 Variable Reluctance Sensor or TTL level Timed Inputs
  - 2 * 5V sensor reference voltages outputs 200mA each
  - 16 * High (2A) and 16 Low (2A) Side Outputs with Diagnostics and Current feedback 8 * General purpose (0.5A) low side drives with diagnostics
- rCube Highlights
  - 2 * Full H Bridge or Stepper Drives with diagnostics (3A and 1A)
  - 3 * CAN Bus 1Mbit/s
  - 3 * 100 Mbps Ethernet
  - 2 * LinBus/RS232
  - 1 * J1850 Bus
  - 1 * ISO9141 Bus
  - External Buffered SPI Bus
  - Full NEXUS and BDM Ports on 565 processor, Debug and JTAG port (1149.1) on 8260.
  - Watchdog, sensor supply and input power rail Monitoring
  - Internal Temperature Sensor Monitor
  - I/O 42V Tolerant
- Specific to this application
  - A/D Input conditions circuit filters modified process cylinder pressure signals more effectively

Agenda

- Introduction to WAVE-RT
- WAVE Model Preparation
- WAVE-RT Model
- HiL Implementation
- **Conclusions**

Conclusions

- This presentation briefly introduced Ricardo recommended WAVE-RT conversion process on real WAVE engine model and its HiL implementation example and scheme
- Life demos of WAVE model run vs. WAVE-RT run were performed
- WAVE-RT present state of art software code running real time or faster even on common PCs intended to be used for SiL and HiL applications

WAVE-RT

"It is hard but fun to drive a sport car!"

Appendix 1

- The WAVE-RT model can be driven by Simulink together with WAVE
- This approach can be effectively used for detailed WAVE-RT model inspection