Simulating Integrated Engine-Emissions-DOC-DPF Performance

Christopher Rutland
Engine Research Center
University of Wisconsin-Madison

Acknowledgements

David Foster
Dave Kapparos, Indranil Brahma, Kushal Narayanawamy, Stephen England, Andrea Strzelec
ERC University of Wisconsin-Madison

Yongsheng He
General Motors Research & Development

Gurpreet Singh and Kevin Stork
Department of Energy: FreedomCAR and Vehicle Technologies Program
System Models: Current Literature

- He, GM R&D (SAE 2007-01-1138)
  - Integrated in Simulink
- Pischlinger, et al., FEV (SAE 2007-01-1128)
  - In-house integration environment
- Guthenke, et al., DaimlerChrysler (SAE 2007-01-1117)
  - GUI based in-house integration environment
  - Simulink
  - DPF model to test ECU model

- Detailed physics based AT models
- Primary use: evaluate AT configurations
- No engine or emissions models
  - Driven by laboratory data
ERC Work: Integrated System Level Model

- Engine and Emissions
- Aftertreatment Devices
- Engine Flow Models
- Combustion Models
- Emission Models
- DOC Model
- DPF Model
- LNT, etc. Models

Exhaust flow
Numerical parameters
Control connection
System Modeling Approach

- Develop an integrated system level computer model
  - Capture interactions:
    - Device – Device
    - Device - Engine
  - Transient
  - Conventional and LTC (HCCI, PCCI, etc.) diesel combustion

- Develop new component models as needed
  - Examples: LTC heat release, emissions, heat transfer

- Incorporate existing models/modules
  - Examples: DPF, DOC, LNT, …

- Integrate components as modules in an overall system level environment
  - Environment: Matlab-Simulink
    - Also allows controllers
  - Develop modular approach for efficiency and ease-of-use

- Validation
  - Component level: experimental data
  - System level: experimental data (when available) and “sense checks”
Objectives and Results

- Explore steady state and transient scenarios
  - Examine and help explain device interactions
  - Suggest ‘guidelines’ of operation
    - Based on model results and analysis

- Recent results
  - LTC combustion under load transients
  - Effect of DPF loading and regeneration on engine operation
  - Compare different DPF regeneration techniques
  - Model prevention and control of runaway DPF regenerations
Diesel Engine Model

- WAVE or GT-Power
- Industry standard
- Includes:
  - Heat transfer models
  - Flow models
  - Turbocharger model (VGT)
  - Simple Combustion model
    - Heat release rates from experimental pressure
    - Could also use built in heat release models
    - Calibration required
- Communicate with Simulink using sensors and actuators

Example of a WAVE engine model with sensors and actuators
Traditional Diesel Engine Model - Disadvantages

- Simple combustion (heat release) model
  - Prescribed function, calibrated with engine data
  - Simple spray models

- Simple emissions models

- Sufficient for conventional diesel operation but not for new LTC type combustion technologies

- Need improved combustion / heat-release models and emissions models for current application
User Defined Cylinder Models

GT-Power Engine Model

In-Cylinder Models
- Fuel Injection
- Vaporization
- Combustion
- Heat Transfer
- Energy, Species Conservation
- Emissions

IVC → EVO

EGR
Actuator
Injection-Timing
Controller
Gas-Exchange
VVT
GT-Power
Sensor
Multi-Zone Combustion Model

- Combustion chamber initialized with multiple zones
- The zones include sub-models for vaporization, chemical kinetics (CHEMKIN), heat transfer, energy/species conservation
SOI = -128 ATDC; Injection Duration = 32.5CAD

Start of combustion predicted accurately

Capture variations in combustion phasing due to thermal and charge stratifications: Useful for parametric LTC studies

Limitations: Requires specified fuel distribution (e.g. no spray)
No mixing between zones
CFD In-Cylinder Model

- Accurate modeling of fuel injection, spray dynamics, mixing, vaporization, chemistry and emissions calculations
- Fine – coarse mapped grid

Detailed in-cylinder CFD model (KIVA-Chemkin)
CFD In-Cylinder Model

- Accurate modeling of fuel injection, spray dynamics, mixing, vaporization, chemistry and emissions calculations
- Fine – coarse mapped grid
Model Validation

Simulation results agree very well for lower loads
- Captures cool flame ignition and main combustion for early injection operation

SOI = -50 ATDC

20% load, EGR = 71%

30% load, EGR = 72%
Run Times for Different Approaches

<table>
<thead>
<tr>
<th>Model Description</th>
<th>Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Zone user combustion model</td>
<td>4 minutes</td>
</tr>
<tr>
<td>5 zone external cylinder model</td>
<td>17 minutes</td>
</tr>
<tr>
<td>GT-Power with mapped grid approach in 2-D</td>
<td>~70 minutes</td>
</tr>
<tr>
<td>GT-Power with refined grid in 2-D (1052 cells at BDC)</td>
<td>~9 hours</td>
</tr>
<tr>
<td>GT-Power with mapped grid using 3-D sector grids</td>
<td>~90 minutes</td>
</tr>
<tr>
<td>GT-Power with refined 3-D sector grids (5460 cells at BDC)</td>
<td>~1 day</td>
</tr>
</tbody>
</table>

- The modeling approaches can be used in multiple cycle, transient simulations
Step transient induced in CA50
Controller forces a delay in intake valve closure
Dual Control – Control CA50 During Load Increase

- Step change in load
- IVC actuated to maintain combustion phasing

- IMEP Controlled
- Target IMEP
- Sensed IMEP
- Fuel Injected

- CA50 set to -6.6 deg
- IVC actuated

- IMEP (bar)
- Time (s)
- Fuel Injected (mg/cycle)

- CA50 (CAD)
- IVC (CAD)
Diesel-PCCI Mode Change: Cooled EGR and IVC

- Model transition: conventional – PCCI – conventional
- Conventional SOI: –12 deg ATDC
- PCCI mode SOI: IVC + 5CAD
- IVC response for PCCI under different EGR conditions
  - IVC actuation during the mode transition decreases with increase in EGR

![Graph showing IVC response for different EGR conditions](image)

Intake valve closes early with increase in cooled EGR

CA50 set to -5.7 ATDC
IMEP set to 5.3 bar
Emissions Models

- Need accurate, fast emissions models to drive the aftertreatment device models

- Focus: Simpler, faster engine combustion models (for now)
  Long run times for DPF filling

- General approach:
  - Use good physically based phenomenological model as starting point
  - Improve model using neural networks to replace model coefficients

- Models required
  - Soot
  - NOx \{ completed \}
  - CO2
  - HC \{ under development \}
Soot Emissions Model

- Physically based neural network model
- Physical model: Bayer and Foster, SAE 2003-01-1070
- Phenomenological models for
  - Injection spray
    - Predicts spray angle, liquid penetration, liftoff length, local equivalence ratio, temperatures, etc.
  - Particulate formation and oxidation

\[
\frac{dm_s}{dt} = C_{sf} \phi m_f P^{0.5} e^{\frac{-E_{sf}}{RT}} - C_{so} m_s P^{1.8} e^{\frac{-E_{so}}{RT}}
\]

- Inputs are obtained from the engine model
  - Profiles of in-cylinder pressure, in-cylinder mean temperature, mass flow rate of fuel through injector, heat release rate
  - Engine speed, percent EGR, global equivalence ratio
Neural network weights added to the physical model

Neural network weights trained using experimental engine data for 8 modes of operation
  - Approximately 57% reduction in error of predicted soot

Converted into an M-file
  - M-file version runs in 1/3 the time of the original Simulink version
  - Small time steps are no longer propagated to other component models
Soot Model and Time Steps

- Each component / module in integrated system requires a certain time step, $\Delta t$

- Overall system runs at $\sim 2.5$ ms $\Delta t$ (primarily set by engine model)

- Soot requires 0.5 CA data ($\sim 0.05$ ms $\Delta t$)
  - Model requires crank-angle resolved spray and heat release profiles

- Simulink manages time stepping
  - Probes models to determine required times
  - Adjusts overall simulation to smallest $\Delta t$ in system

- Soot model $\Delta t$ requirement can severely limit overall system speed
Solutions for Managing Disparate Time Scales

- Hide inner workings of model from Simulink
  - Use self-contained Matlab or Fortran/C routines
  - Use sub-cycling time integration inside routines
  - Use level 2 S-functions
    - These can specify their own hit time

- Use external file for data transfer
  - Pass small $\Delta t$ data outside of Simulink

- Implement periodic ‘triggering’
  - Skip calculation in model except every ‘n’ engine cycles
  - Use most recent solution until next calculation
NOx Emissions Model

- Physically based neural network model (Brahma and appendix of England et. al., SAE 2006-01-0263)
- Implemented as an M-file
- Inputs are obtained from the engine and soot models
- NOx prediction is based on maximum rate of formation

\[ \text{NO}_x = K_{\text{max}} \exp \left( -\frac{K_2}{T_{\text{Diffusion Flame}}} \right) \left[ \frac{[O_2]}{[N_2]} \right]^{1/2} \]

- Trained using experimental engine data for 8 modes of operation
DOC Model

- Proprietary GM model (Bissett)
  - Includes kinetics for oxidation of CO, HC, and NO to NO2

- Implemented as a Fortran version
  - Faster than Simulink version

- Fortran allows decoupling of integration time
  - DOC model can operate at large time steps (10 ms or longer)
DPF Model

- Modified version of the MTU 1-D model (based on work of Konstandopoulos and Johnson)
- Fortran code linked to Simulink using an S-function (level 2)
- Input and output modified for dynamic use in Simulink and time integration improved
- Calibrated against experimental data
- Darcy law pressure drop for the wall and soot layer
- Device sub models describe:
  - Particle deposition inside wall and on wall surface
  - Flow and temperature fields
  - Catalytic and thermal oxidation of soot

Comparison of Experimental DPF Pressure Drop and DPF Model Prediction

Deep Bed Filtration

Cake Filtration

Transition
Combined DPF-LNT

- **DPF**
  - Exhaust gases in
  - Plug
  - Exhaust gases out (higher CO, CO₂ levels)
  - Soot ‘cake’ deposit

- **LNT**
  - NO + O₂ → NO₂
  - Lean storage

- **SNT**
  - NOx storage
  - Flow chronology
  - Soot deposit
  - Exhaust gases out

Exhaust gases in

Exhaust gases out
Controllers and Validation

- Controllers implemented in Simulink for:
  - Engine load (fueling)
  - Engine speed
  - EGR
  - Turbo boost
  - DPF regeneration

- The component models tested and validated individually

- The integrated model is tested to verify proper component interaction
  - DPF loading and regeneration simulations to test the model and component interaction
  - The simulation results are consistent with experimental results presented by Singh et. al. (SAE 2006-01-0879)
DPF Loading and Regeneration

- Simulate a light duty, turbo charged, CIDI engine
  - Common rail direct fuel injection
  - Variable geometry turbine
- Regeneration: modes 3 and 5
  - DPF initially loaded to 3 [g/l] of soot and regeneration starts at 3.2 [g/l]
  - Mode 5: Fuel injected ahead of the DOC
  - Mode 3: Fuel injected ahead of the DOC with intake throttling assistance
- Regeneration tests were done with mode switching (mode 5 to 2) during regeneration to investigate prevention/control of regeneration runaway

<table>
<thead>
<tr>
<th>Mode Number</th>
<th>Load</th>
<th>Engine Speed [RPM]</th>
<th>Approximate Fueling Rate [kg/hr]</th>
<th>EGR [mass %]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Very Low</td>
<td>970</td>
<td>1.42</td>
<td>62.14</td>
</tr>
<tr>
<td>3</td>
<td>Low</td>
<td>1120</td>
<td>4.035</td>
<td>37.96</td>
</tr>
<tr>
<td>5</td>
<td>Moderate-Low</td>
<td>1520</td>
<td>9.365</td>
<td>19.15</td>
</tr>
<tr>
<td>8</td>
<td>Moderate</td>
<td>2320</td>
<td>18.85</td>
<td>10.50</td>
</tr>
</tbody>
</table>
DPF Regeneration (Mode 5)

- Fuel injected in exhaust before DOC
- Increases reactions in DOC and exhaust T
- Controlled to achieve a desired DPF inlet temperature of 600 °C
Simulation results are consistent with the experimental results given by Singh et. al. (SAE 2006-01-0879)

Note long regeneration time
Throttle Assisted Regeneration (Mode 3)

- Engine exhaust temperatures too low for fuel injection before DOC (> 300 °C required)
- Throttle engine intake air flow
  - Increases equivalence ratio and engine exhaust T (ΔT = 44 °C)
- Higher exhaust T allows fuel injection in exhaust before DOC
Throttle Assisted Regeneration (Mode 3)

Pressure Drop Across DPF

Mass of Soot Trapped in the DPF

Note long regeneration time

DPF Regeneration
Throttle Assisted Regeneration (Mode 3)

- Engine out NOx increased significantly during DPF regeneration
- Intake throttling required reducing EGR to maintain sufficient O2 for given load
- Reduced EGR resulted in higher engine out NOx
Runaway Regeneration (Mode 5)

- Regeneration by fuel injection before DOC
- Try to reduce regeneration time by injecting more fuel
- Results in runaway regeneration

### Fuel Consumption

- Total BSFC
- Engine BSFC

### Exhaust Temperatures

- Engine Out
- DPF Out
- DOC Out

Temperature continues to rise after fuel shutoff
Runaway Regeneration (Mode 5)

Pressure Drop Across DPF

Mass of Soot Trapped in the DPF

Note “runaway” regeneration time
Preventing Runaway Regenerations

- Use $\Delta T$ predictive capability in the regeneration controller via simple energy balance:

$$m_{fuel} = \frac{m_{exh} C_{p_{exh}} \Delta T_{exh}}{LHV_{fuel}}$$

- Very simple energy balance will get the DPF inlet temperature within 10 [°C] of the target value

- Use PI feedback control to adjust the predicted exhaust fuel injection rate to obtain the target DPF inlet temperature

- Feedback PI controller alone (without predictive capability) is not recommended
Regeneration During Mode 5-2 Transient

- Runaway regeneration occurs when DPF is very hot and:
  - Exhaust flow rate is suddenly lowered in mode transitions
  - Excess O2 is available in exhaust
  - Consistent with results from Koltsakis, et al. (SAE 2007-01-1127)
- Prevent runaway regeneration by intake throttling to reduce available O2 in exhaust

Regeneration: Runaway after Mode Switch

Prevent Runaway with Intake Throttling
Regeneration During Mode 5-2 Transient

- Alternative method to prevent runaway:
  - Air injection into exhaust
- Cool air removes heat from soot oxidation reactions

Regeneration: Runaway after Mode Switch

Prevent Runaway with Air Injection into DPF
Controlling Runaway Regenerations

- The integrated model has been used to simulate how a runaway regeneration can be detected and controlled
  - Reactive control

- Can the integrated model be used to detect unfavorable conditions before a runaway regeneration starts to prevent runaway proactively?
  - To investigate this possibility phase diagrams have been generated
DPF Phase Diagrams

- Phase Diagrams:
  - Max DPF wall T
  - Function of:
    - Equivalence ratio (e.g. available O2)
    - Exhaust flow rate

- Example:
  - Mode 8 to mode 2 transient during regeneration

- Conditions:
  - Initial DPF wall temperature = 600 °C
  - Soot loading = 4 g/l
  - Inlet temperature = 162 °C (after switch to mode 2)
Using the Phase Diagrams

- Runaway regeneration occurs when DPF is very hot and:
  - Exhaust flow rate is low
  - Excess O2 is available

- By using the phase diagram unsafe operating conditions can be identified and avoided by:
  - Intake throttling
  - Air injection into DPF inlet
Summary

- Integrated system model for diesel aftertreatment studies
- Advanced engine combustion model for LTC studies
- Accurate emissions models to drive the aftertreatment devices
- Accurate, modular aftertreatment devices models
- Implemented in Simulink for controls and numerical management
- Integrated model used to explore DPF regeneration
  - Phase maps to explore runaway regeneration recovery
Ongoing Activities

- Development of additional submodels
  - CO and HC emissions models
  - Incorporation of an SCR device model

- Continue improvements to integrated system
  - Improve computer run times
  - Improve modularity and ease-of-use

- Continue to use the system model to study device interactions and appropriate operation for transient and regeneration scenarios
  - Develop ‘guidelines for operation’
Thank You