Using an Integrated System Model to Simulate a Diesel Engine with Exhaust Aftertreatment

Stephen B. England
Christopher J. Rutland
David E. Foster
Engine Research Center
University of Wisconsin-Madison

Yongsheng He
General Motors Research & Development

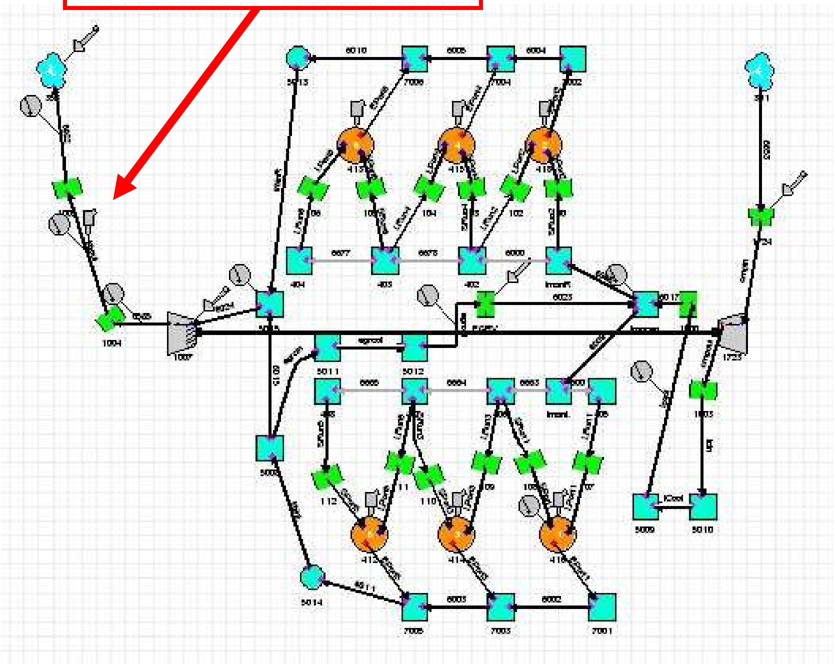
Introduction

- Purpose of this work
  - Develop a computer model to simulate the interaction of a diesel engine with aftertreatment devices
  - Use the model developed to:
    - Investigate the effect of DPF loading and regeneration on engine operation
    - Model and compare different DPF regeneration techniques
    - Simulate an FTP driving cycle
- Approach
  - Integrate existing component models into an overall system level model
  - Develop a controller model for DPF regeneration
  - Use Matlab Simulink to link (integrate) the component models

Component Models

- The available component models include
  - Engine model (includes exhaust system model)
    - WAVE
    - GT Power
  - Emissions models
    - Soot
    - NOx
  - Aftertreatment models
    - DPF
    - DPNT
    - DOC
    - LNT

Diesel Engine Model



Exhaust Fuel Injector (Ahead of DOC)

Example of a WAVE engine model with sensors and actuators

- WAVE, one dimensional engine simulation software, is used
  - Includes:
    - Heat transfer model
    - Flow model
    - Turbocharger model (VGT)
      - Turbine and compressor maps
      - External PI rack controller implemented in Simulink
    - Combustion model
      - Heat release rates extracted from experimental cylinder pressure traces
      - Could also use Wiebe model in WAVE
  - Calibration done using results of experimental study conducted by General Motors researchers
  - Communicates with Simulink using sensors and actuators

Exhaust System Model

- Model created inside of WAVE
- 1-D treatment of exhaust gas flow and 2-D treatment of wall heat transfer
- Can model a variety of piping configurations including close couple or under floor aftertreatment devices
- Heat transfer modes include:
  - Internal convection
  - Radial and axial conduction through pipe walls
  - Conduction, convection, and radiation through pipe air gaps
  - External convection (forced or natural) and radiation
- Models fully transient behavior
- Affect of flow pulsations on heat transfer is captured

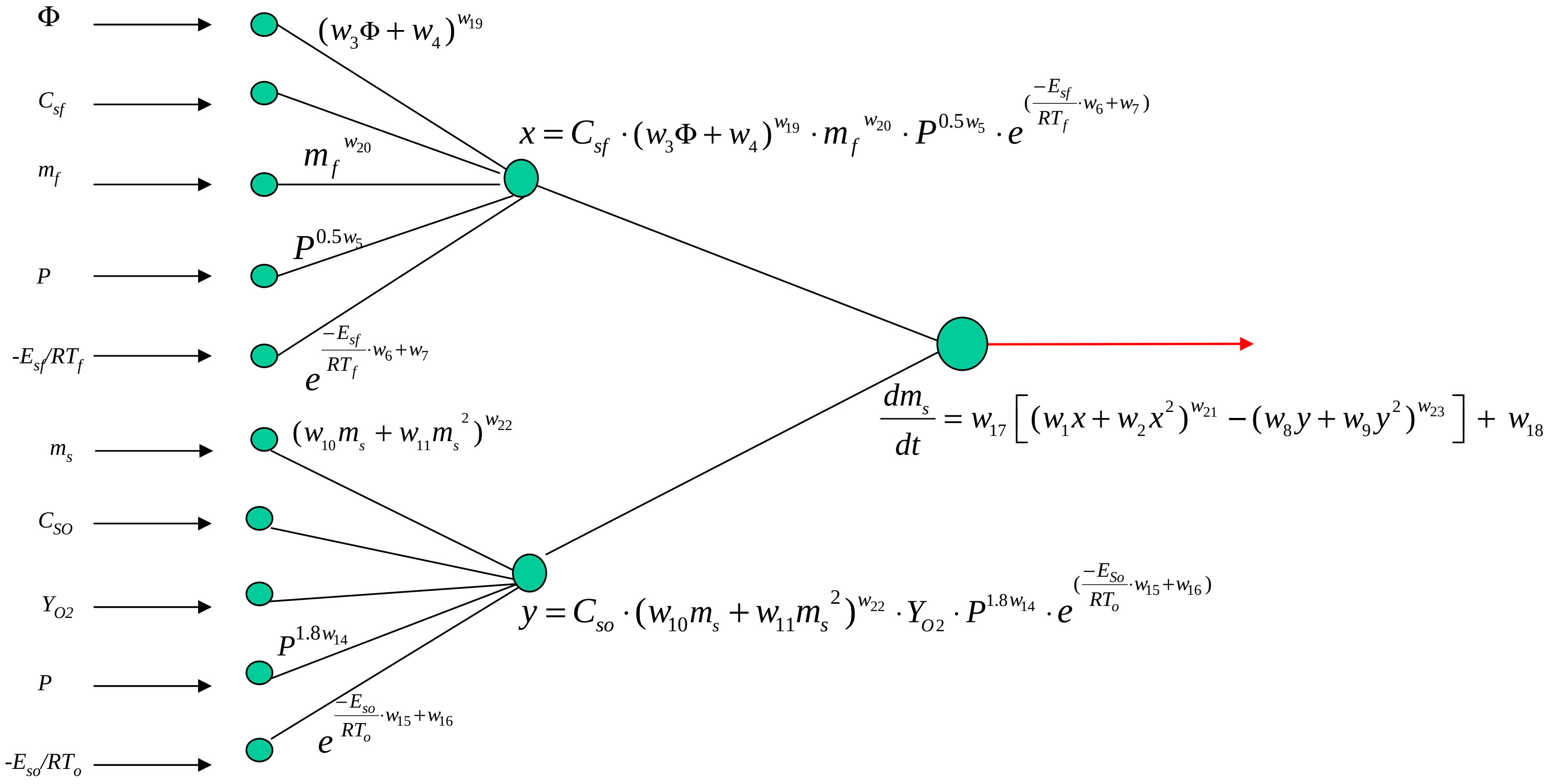
Soot Emissions Model

- Physically based neural network model
- Physical model (Bayer and Foster, SAE 2003-01-1070)
  - Consists of zero-dimensional phenomenological models for
    - Injection spray
      - Predicts spray angle, liquid penetration, liftoff length, local equivalence ratio, temperatures, etc.
    - Particulate formation and oxidation
      - Net particulate mass formation rate is integrated over the engine cycle

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

- 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

Soot Emissions Model



Φ

C_{sf}

m_f

P

$-E_{sf}/RT_f$

m_s

C_{SO}

Y_{O2}

P

$-E_{so}/RT_o$

$(w_3\Phi + w_4)^{w_{19}}$

$m_f^{\;w_{20}}$

$P^{0.5 w_5}$

$e^{\frac{-E_{sf}}{RT_f}\cdot w_6 + w_7}$

$(w_{10} m_s + w_{11} m_s^{\;2})^{w_{22}}$

$P^{1.8 w_{14}}$

$e^{\frac{-E_{so}}{RT_o}\cdot w_{15} + w_{16}}$

$$x = C_{sf} \cdot (w_3\Phi + w_4)^{w_{19}} \cdot m_f^{\;w_{20}} \cdot P^{0.5 w_5} \cdot e^{(\frac{-E_{sf}}{RT_f}\cdot w_6 + w_7)}$$

$$\frac{dm_s}{dt} = w_{17}\left[(w_1 x + w_2 x^2)^{w_{21}} - (w_8 y + w_9 y^2)^{w_{23}}\right] + w_{18}$$

$$y = C_{so} \cdot (w_{10} m_s + w_{11} m_s^{\;2})^{w_{22}} \cdot Y_{O2} \cdot P^{1.8 w_{14}} \cdot e^{(\frac{-E_{So}}{RT_o}\cdot w_{15} + w_{16})}$$

Neural network weights added to the physical particulate formation and oxidation model (Brahma et. at., SAE 2005-01-1122)

Soot Emissions Model

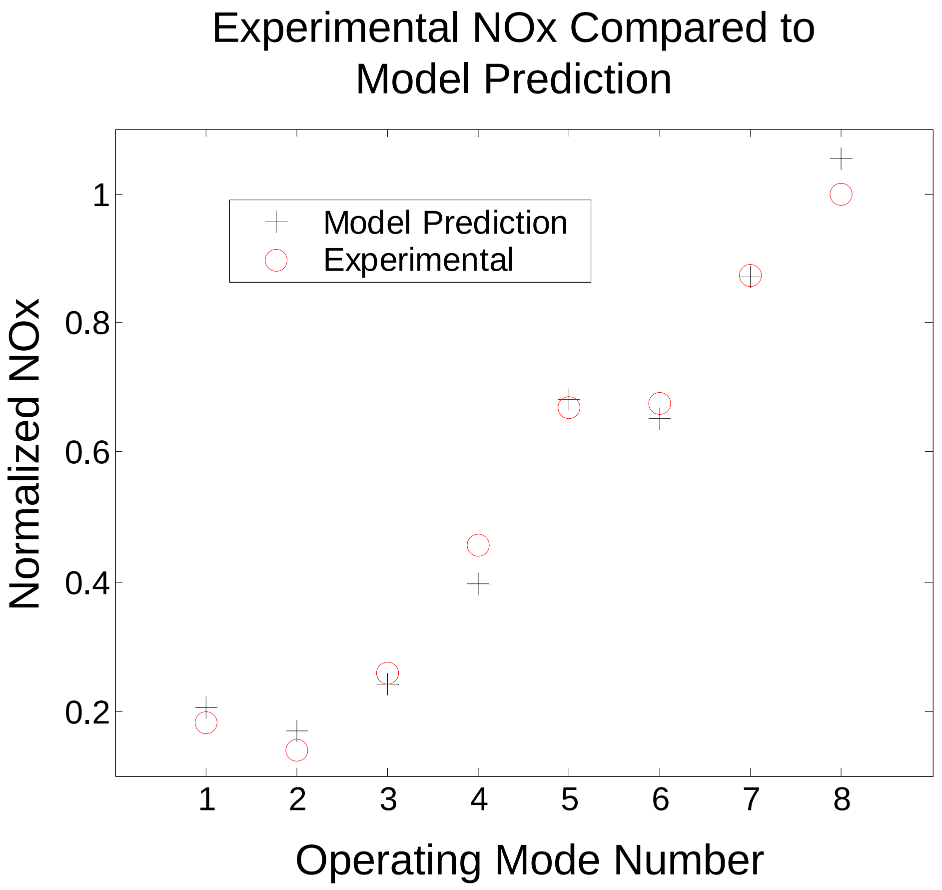
- Neural network weights trained using experimental engine data for 8 modes of operation
  - Approximately 57% reduction in mse of predicted soot
- Originally implemented using Simulink
  - Converted into an M-file because of time step and run time issues
    - M-file version runs in 1/3 the time of the Simulink version
    - Small time steps are no longer propagated to other component models

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
- No integration, NOx prediction is based on maximum rate of formation

$$\mathrm{NO_X}\,(ppm) = K_1\,\mathbf{Max}\left(\frac{e^{\left(\frac{-K_2}{T_{Diffusion\_Flame}}\right)}[\mathrm{O_2}]^{1/2}[\mathrm{N_2}]}{T_{Diffusion\_Flame}^{1/2}}\right)$$

- trained using experimental engine data for 8 modes of operation

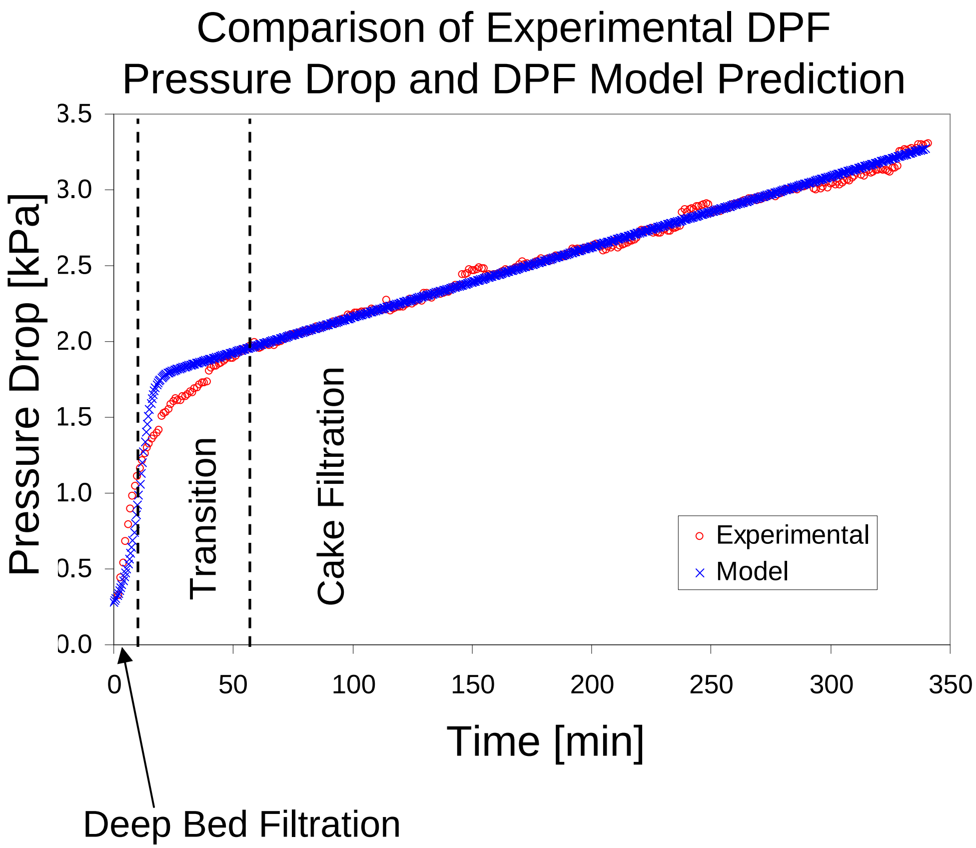


DOC Model

- Internal GM code (Bissett)
  - Includes kinetics for oxidation of CO, HC, and NO to NO2
- For the current work a Simulink version is used
  - Convenient implementation into the integrated model
- Implementation of a Fortran version is underway
  - Fortran version should run faster
  - Small time steps of the integrated model will no longer be imposed on the DOC model

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
- Input and output modified for dynamic use in Simulink
- 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



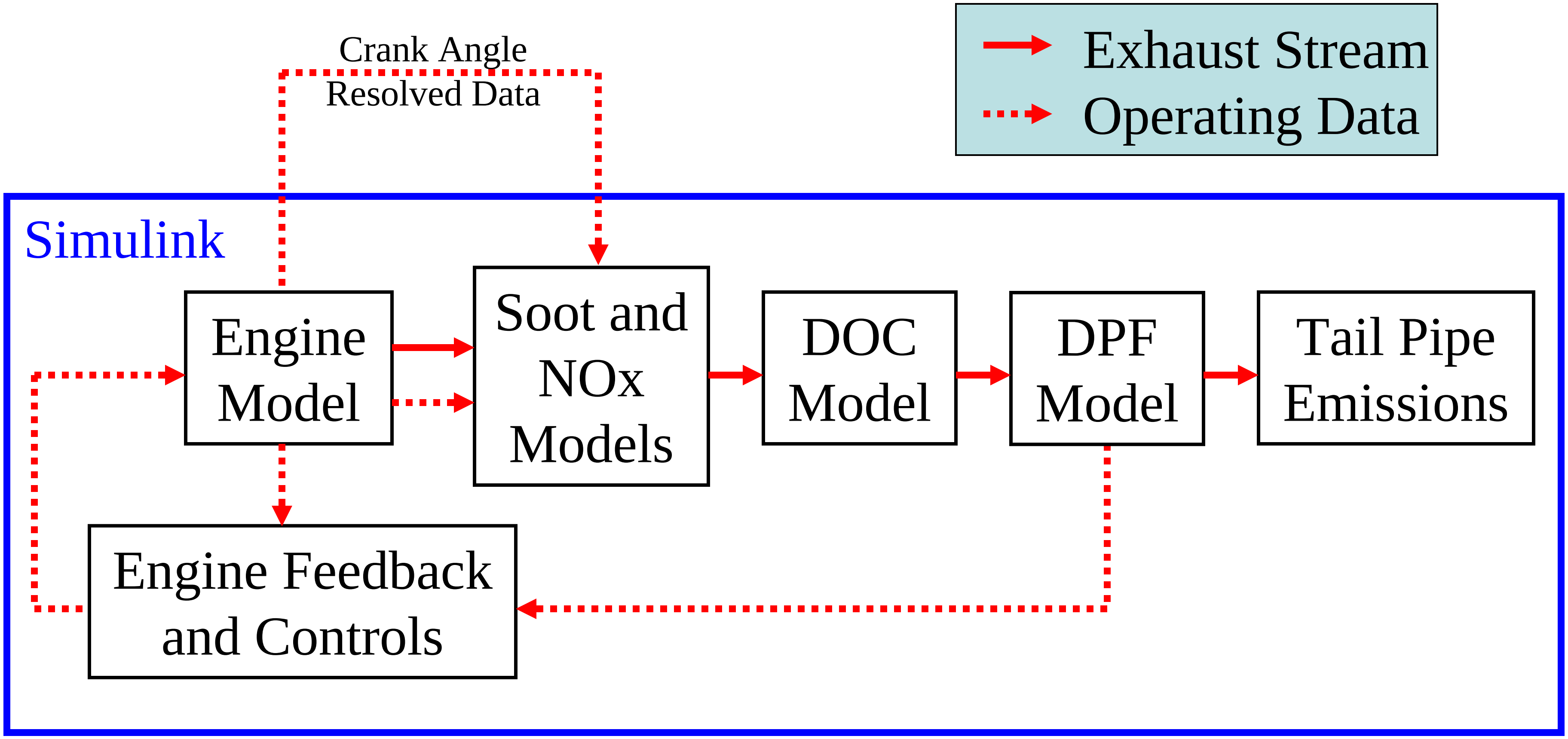
Controllers

- Controllers implemented in Simulink for:
  - Engine load (rate of fueling)
  - Engine speed
  - EGR
  - Turbo boost
  - DPF regeneration
    - Exhaust fuel injection ahead of a DOC
      - Injection rate is predicted based on needed ΔT, then adjusted using PI gains
    - Exhaust fuel injection ahead of a DOC aided by intake throttling

Integration Issues

- Time steps
  - The component models require different time steps but the Simulink integrated model runs at one time step
  - This can be overcome by implementing component models with small time steps in self contained code (M-file, Fortran, or C)
  - The emissions and WAVE engine model can pass crank angle resolved data via M-files and thus avoid running Simulink at excessively small time steps
- Triggering
  - The soot model can be slow but it doesn’t need to be run every engine cycle, triggering every ~19 cycles gives good results
  - Similarly, the DPF can be triggered every ~0.5 seconds and still give good results
- Mixed codes (M-files, Fortran, C, Simulink)
  - If using Simulink to link the models this is not a problem

The Integrated Engine, Emissions, and Aftertreatment Model



Schematic of the Integrated Model

Testing the Model

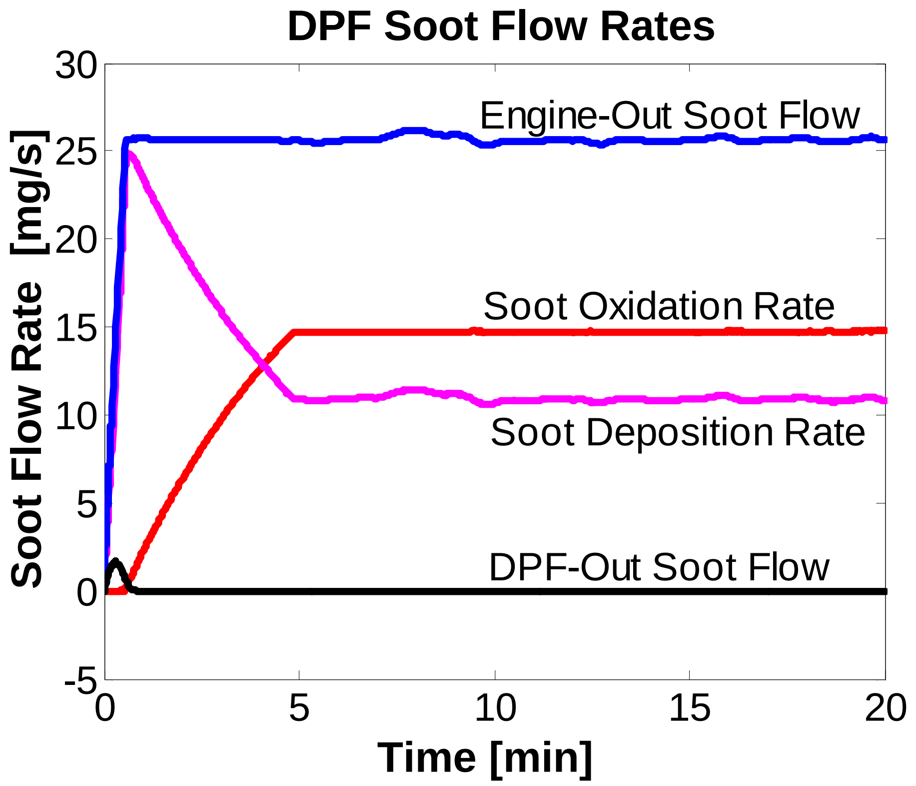
- The component models have been previously tested and validated
- The integrated model needs to be tested to verify proper component interaction
  - DPF loading and regeneration simulations have been done to test the model and component interaction
  - The simulation results are consistent with experimental results presented by Singh et. al. (SAE 2006-01-0879)

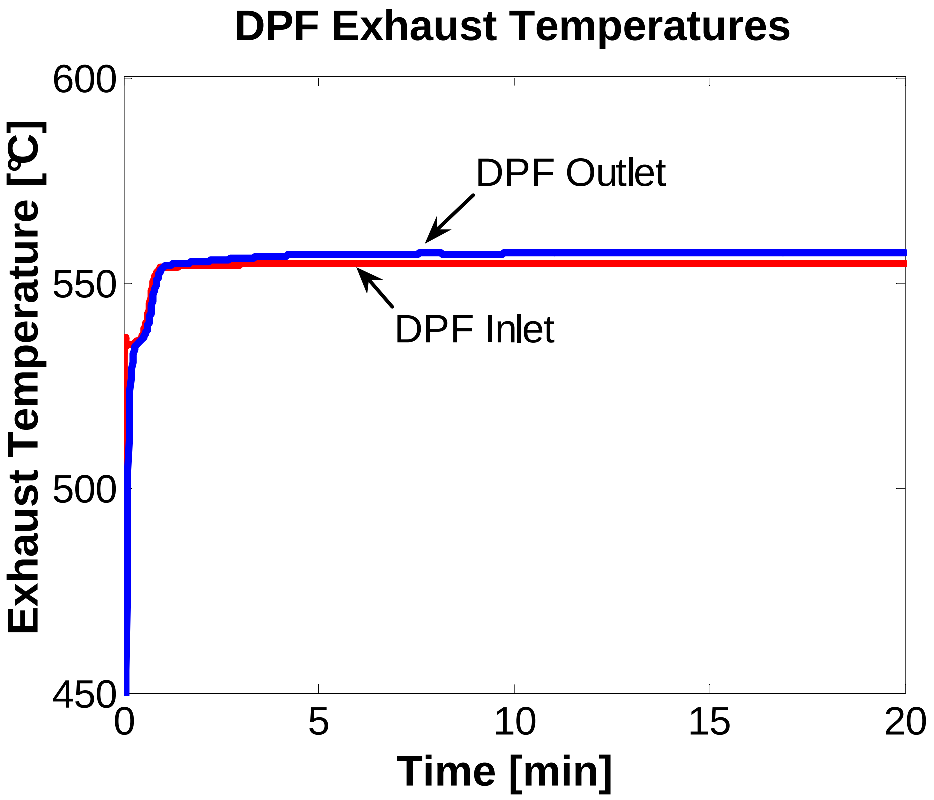
DPF Loading and Regeneration Simulations

- To test the operation of the integrated model a light duty, turbo charged, CIDI engine was simulated
  - The engine is equipped with common rail direct fuel injection and a turbo charger with a variable geometry turbine
  - A DPF loading simulation was done for mode 8 with an initially clean DPF and no DOC
  - Regeneration tests were done for modes 3 and 5 with the DPF initially loaded to 3 [g/l] of soot and regeneration set to start at 3.2 [g/l]
    - Fuel injected ahead of the DOC for mode 5
    - Fuel injected ahead of the DOC with intake throttling assistance for mode 3

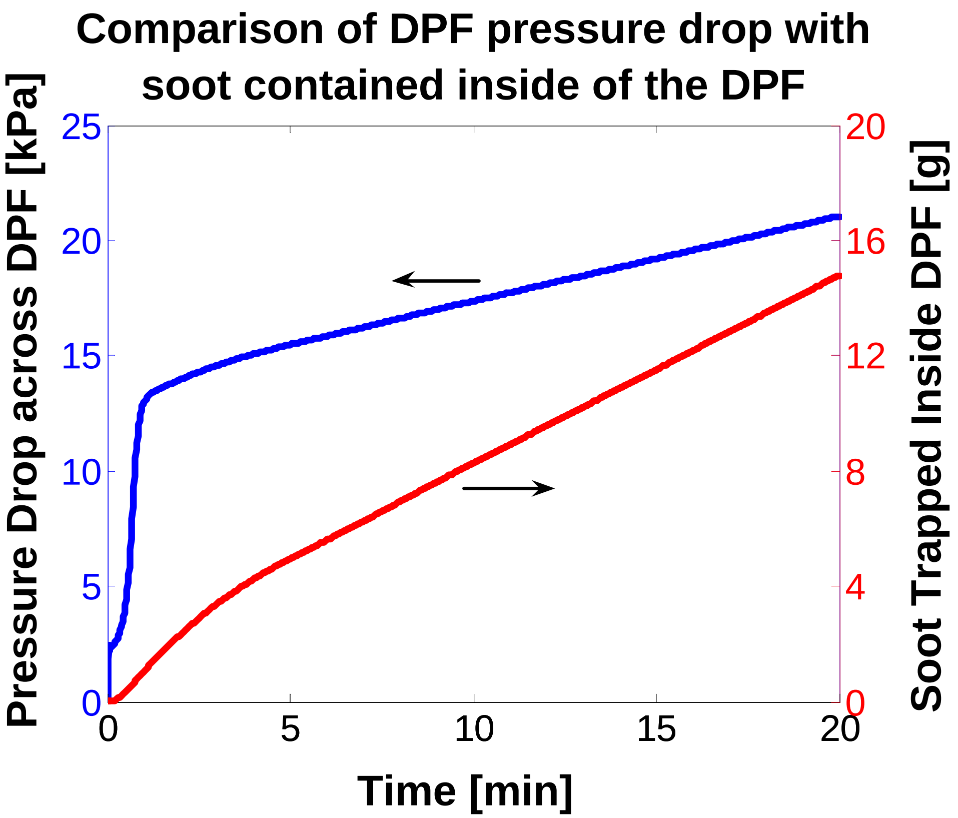
Mode Number	Load	Engine Speed [RPM]	Approximate Fueling Rate [kg/hr]	EGR [mass %]
3	Low	1120	4.035	37.96
5	Moderate-Low	1520	9.365	19.15
8	Moderate	2320	18.85	10.5

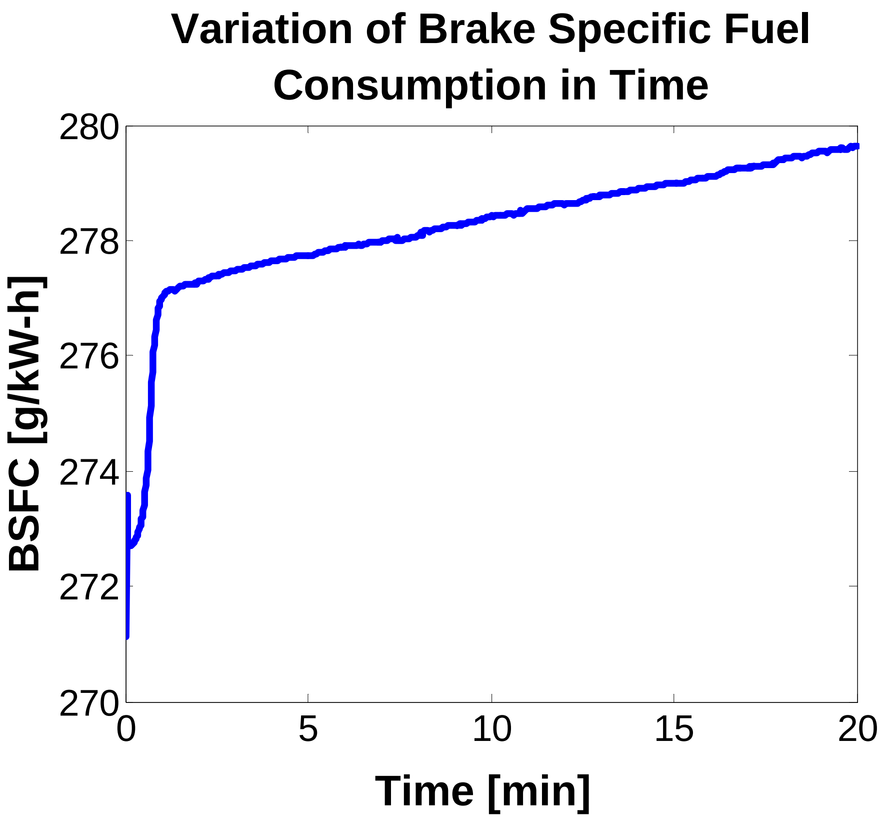
DPF Loading Simulation (Mode 8)


DPF Soot Flow Rates

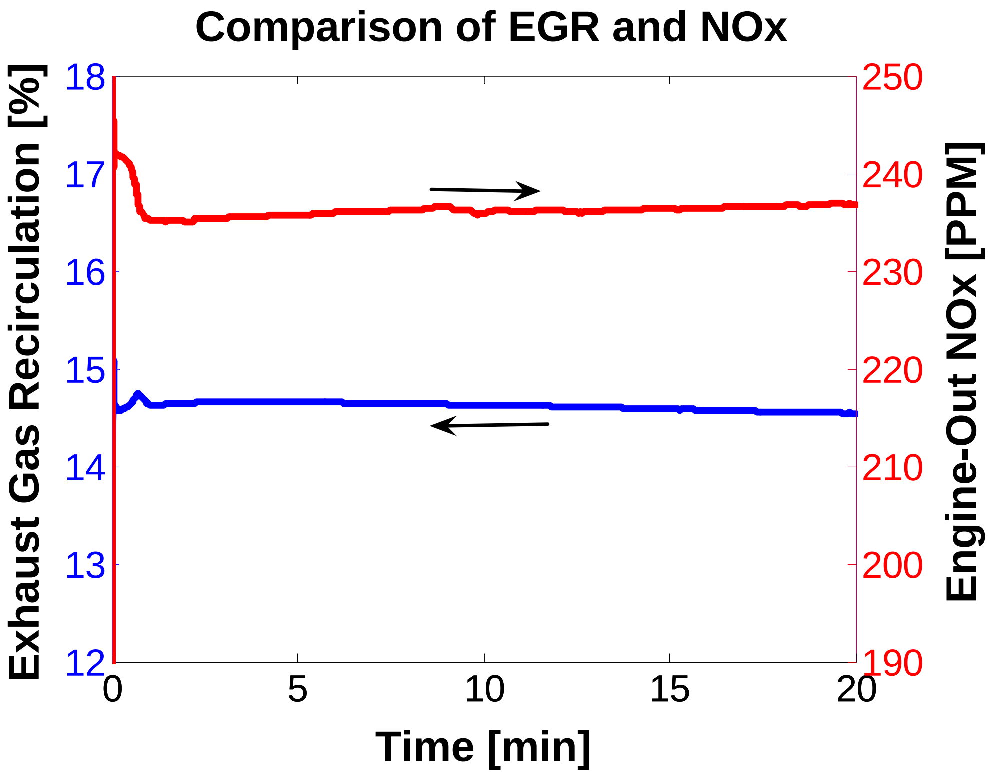

DPF Exhaust Temperatures

DPF Loading Simulation (Mode 8)





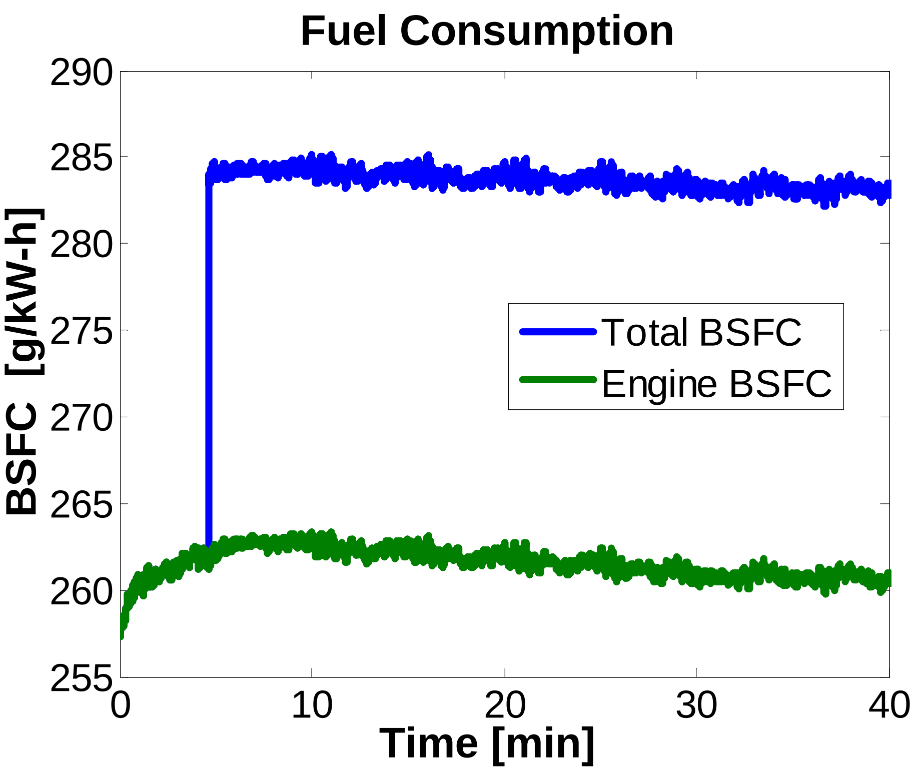
DPF Loading Simulation (Mode 8)

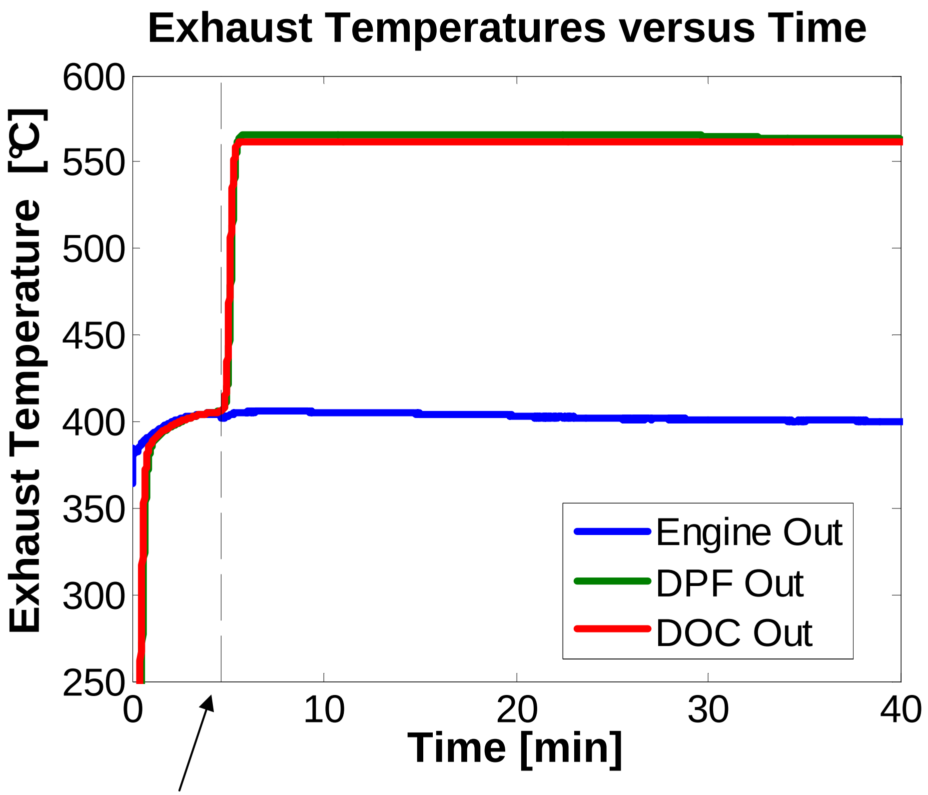


Summary

- Interaction of component models is successful
- BSFC penalty with DPF pressure drop is consistent with experimental results
- Some passive regeneration of the DPF demonstrated

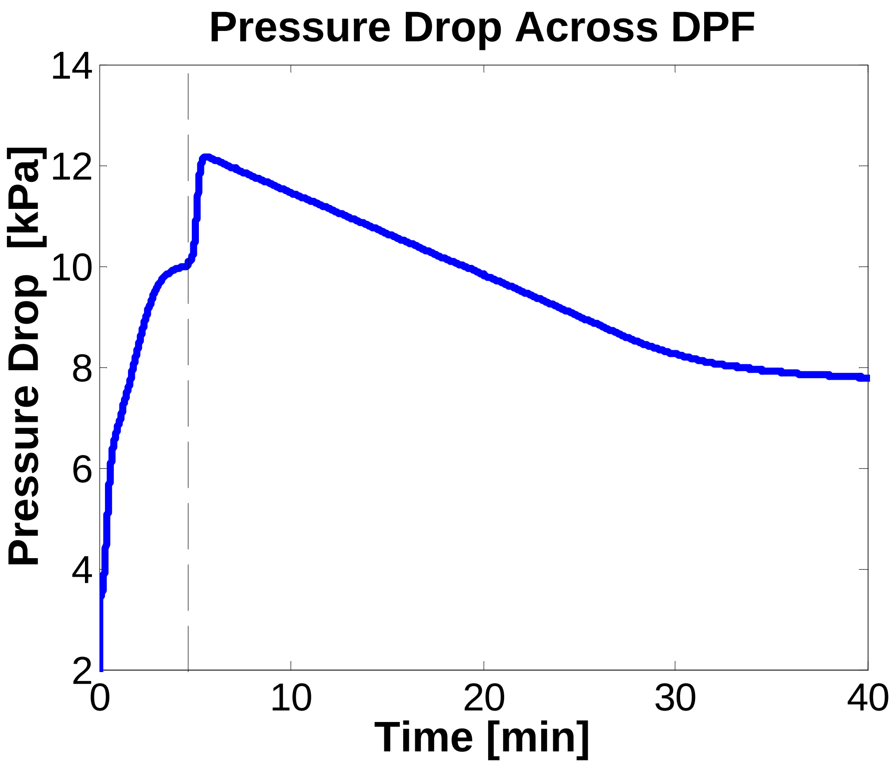
DPF Regeneration (Mode 5)

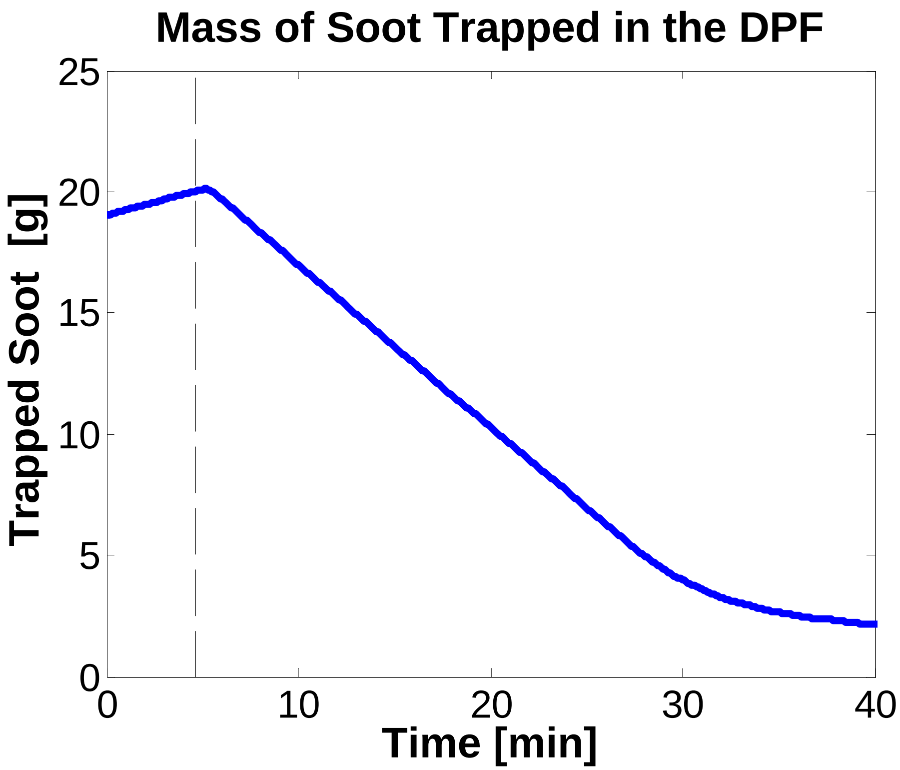

Fuel Consumption


Exhaust Temperatures versus Time

Start of
Regeneration

DPF Regeneration (Mode 5)



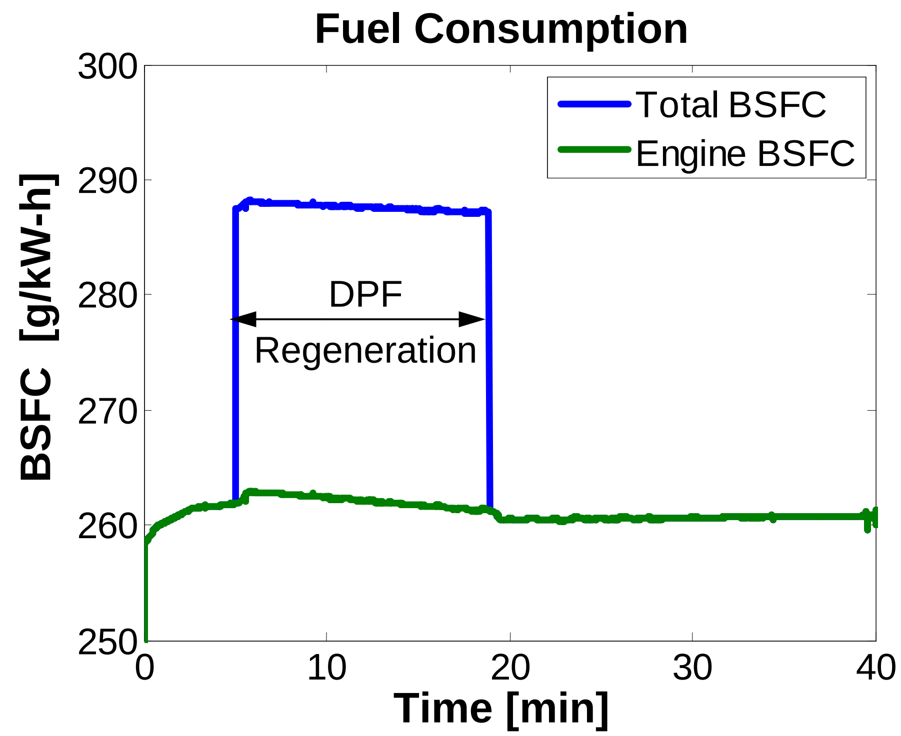


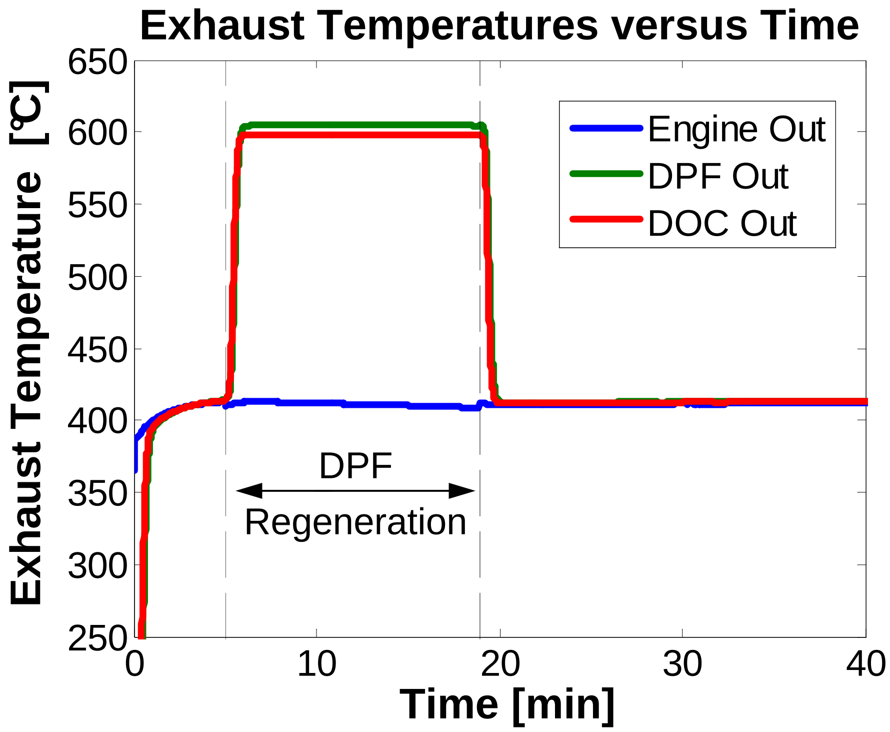
DPF Regeneration (Mode 5)

Summary

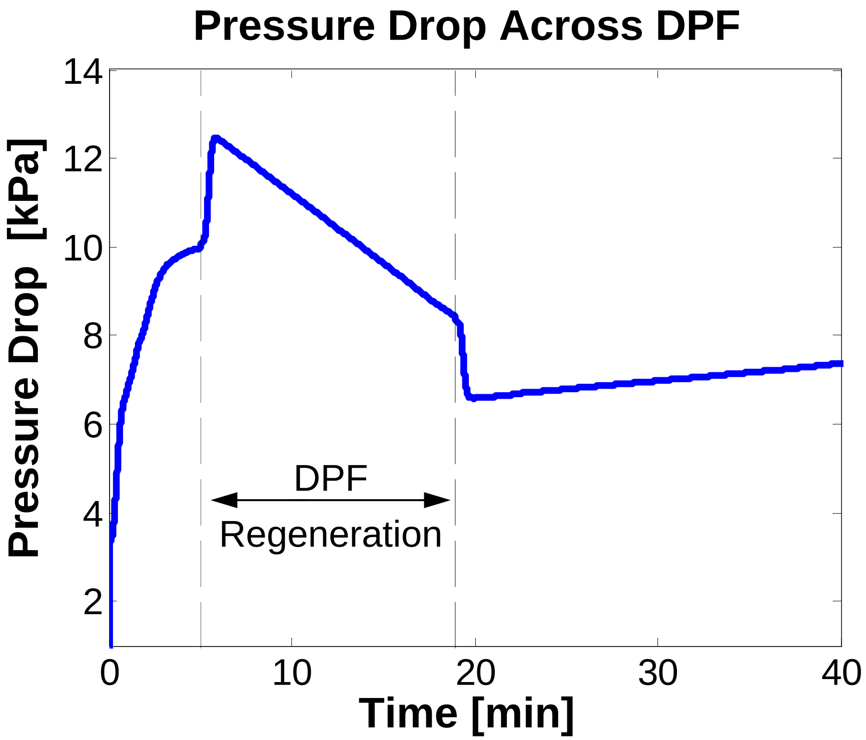
- Regeneration was set to start at a soot loading of 20 [g] (3.2 [g/l])
- The fuel injected ahead of the DOC was controlled to achieve a desired DPF inlet temperature of 560 [℃]
- Results show that the regeneration should have been stopped at about 4 [g] (0.6 [g/l]) to avoid excessive fuel consumption
- The simulation results are consistent with experimental results presented by Singh et. al. (SAE 2006-01-0879)

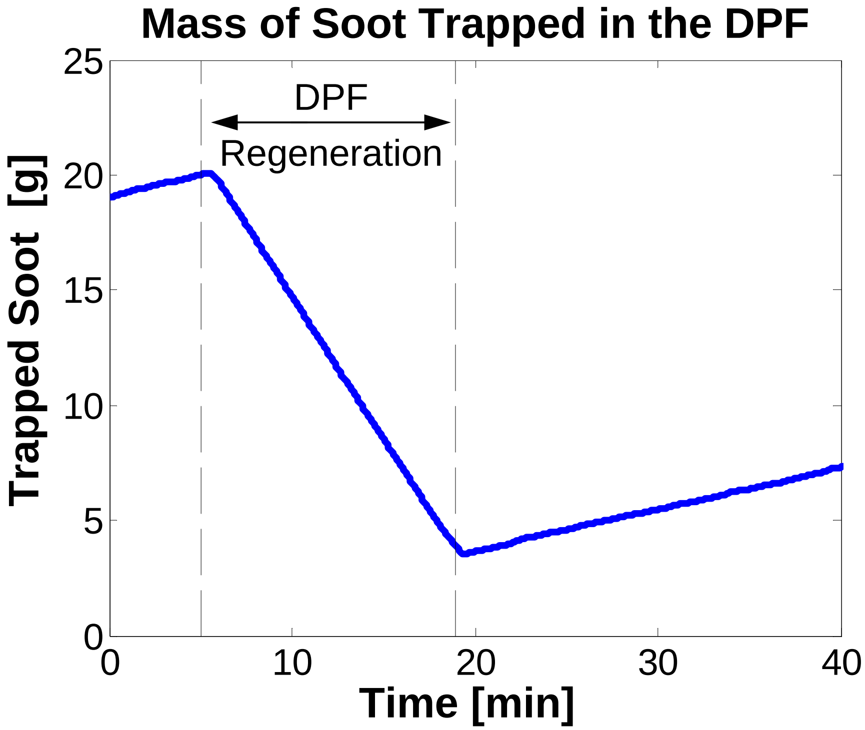
Faster Regeneration (Mode 5)


Fuel Consumption

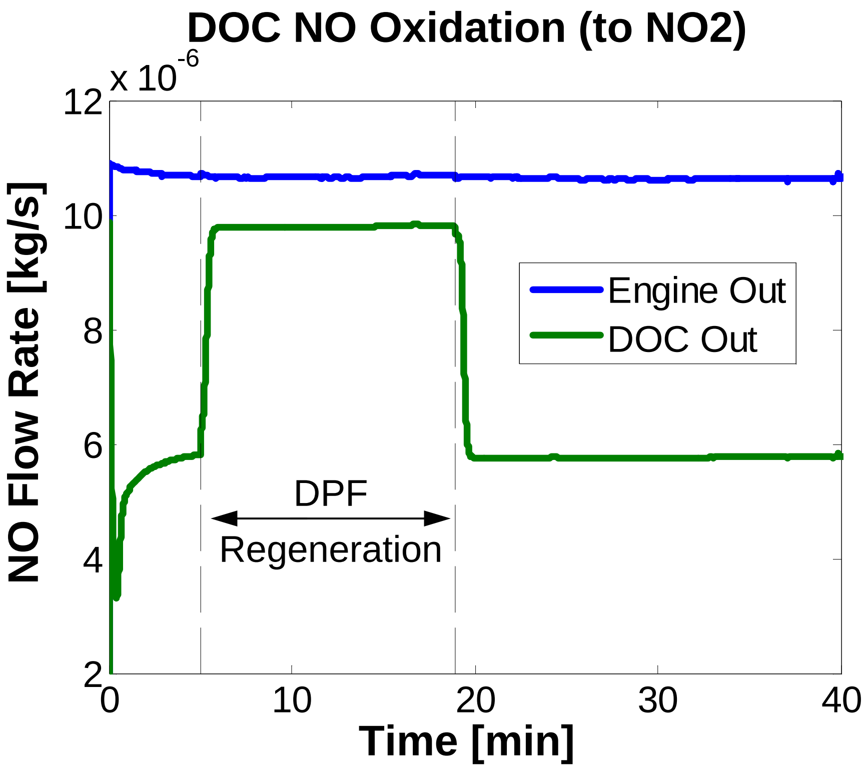

Exhaust Temperatures versus Time

Faster Regeneration (Mode 5)





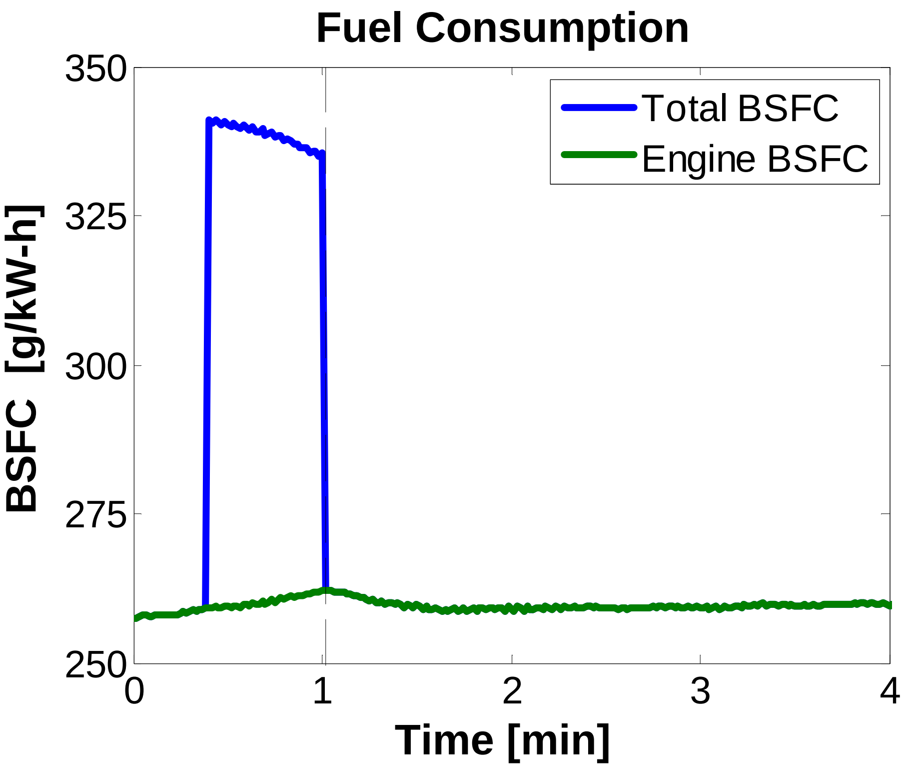
Faster Regeneration (Mode 5)

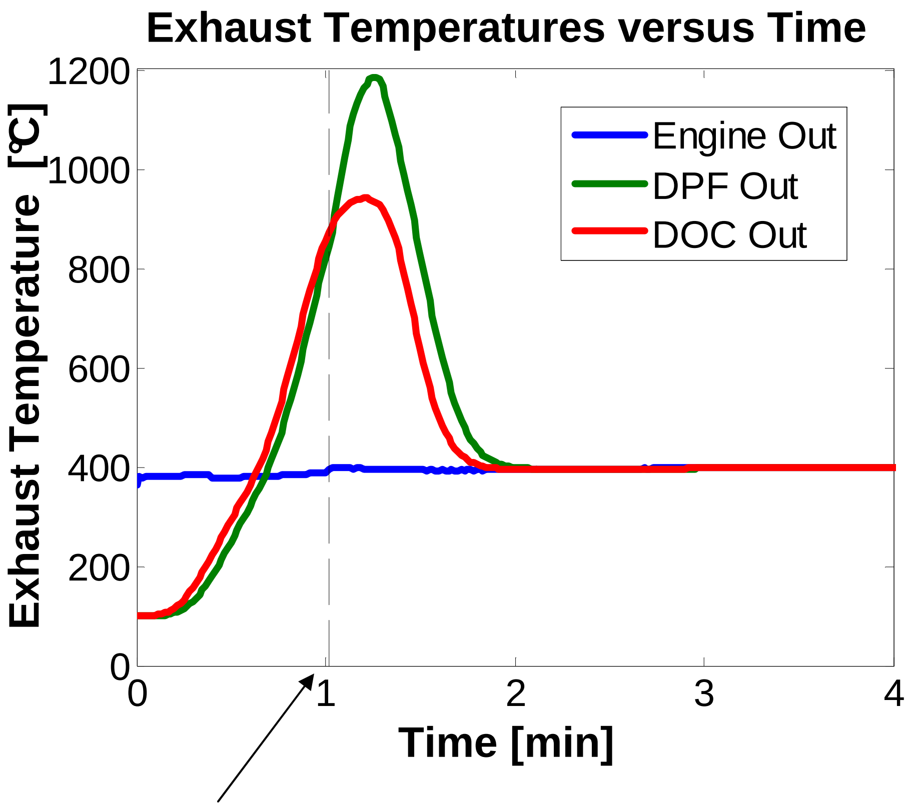


Summary

- Regeneration was set to start at a soot loading of 20 [g] (3.2 [g/l]) and stop when the loading dropped to 4 [g] (0.6 [g/l])
- The fuel injected ahead of the DOC was controlled to achieve a desired DPF inlet temperature of 600 [℃]
- Additional BSFC penalty—over the 560 [℃] regeneration—is made up for by the faster regeneration time
- Again, simulation results are consistent with the experimental results given by Singh et. al. (SAE 2006-01-0879)

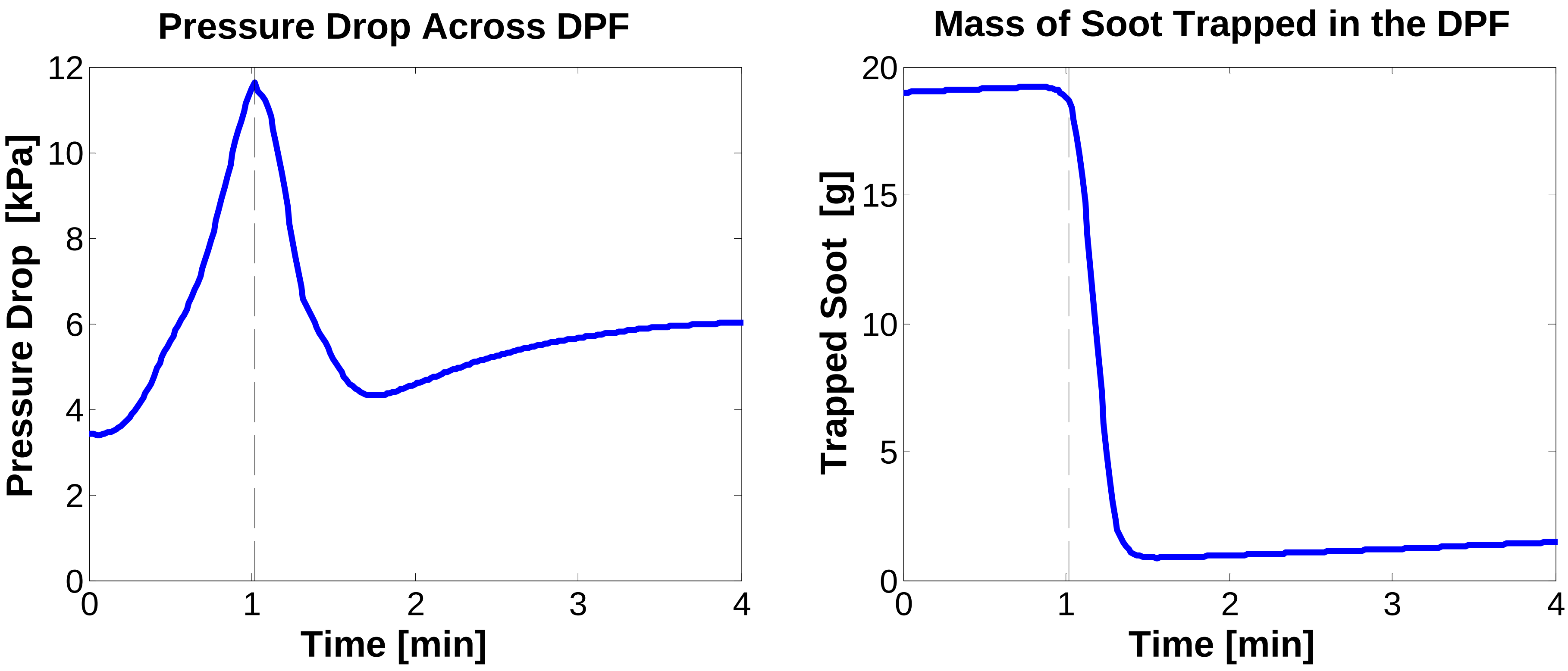
Runaway Regeneration (Mode 5)





DOC fuel
injection stop

Runaway Regeneration (Mode 5)



Runaway Regeneration (Mode 5)

Summary

- Too much fuel was injected ahead of the DOC
- Fuel was then abruptly shut of allowing more oxygen to flow into the very hot DPF causing the runaway regeneration
- Temperatures seen in the DOC and DPF where far too hot for real devices to handle
- Don’t try this with a real DOC and DPF!

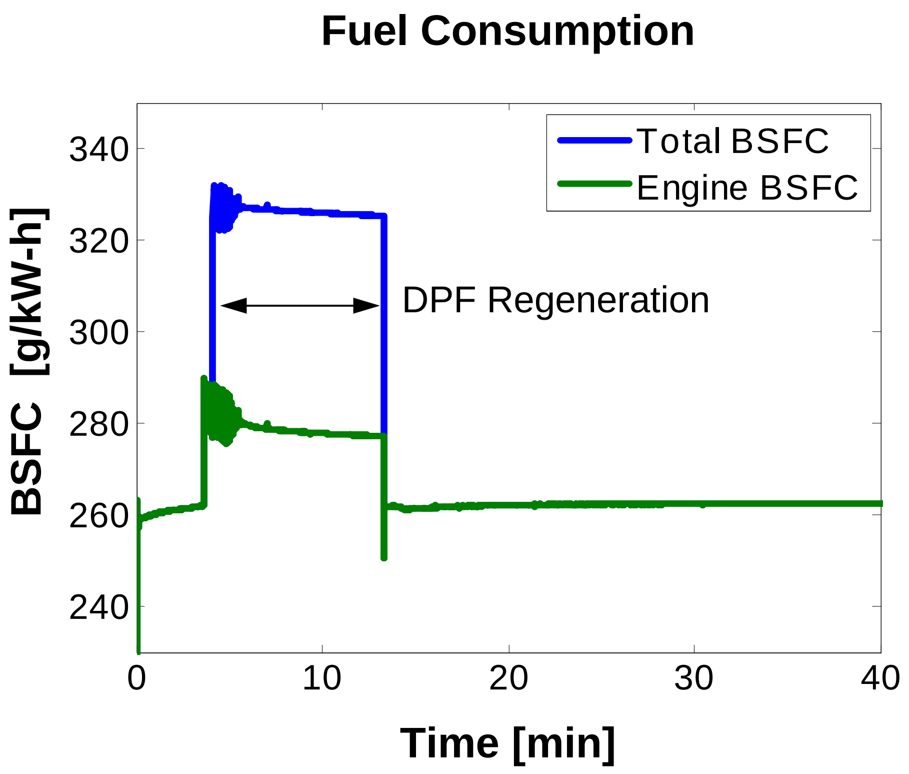
Preventing Runaway Regenerations

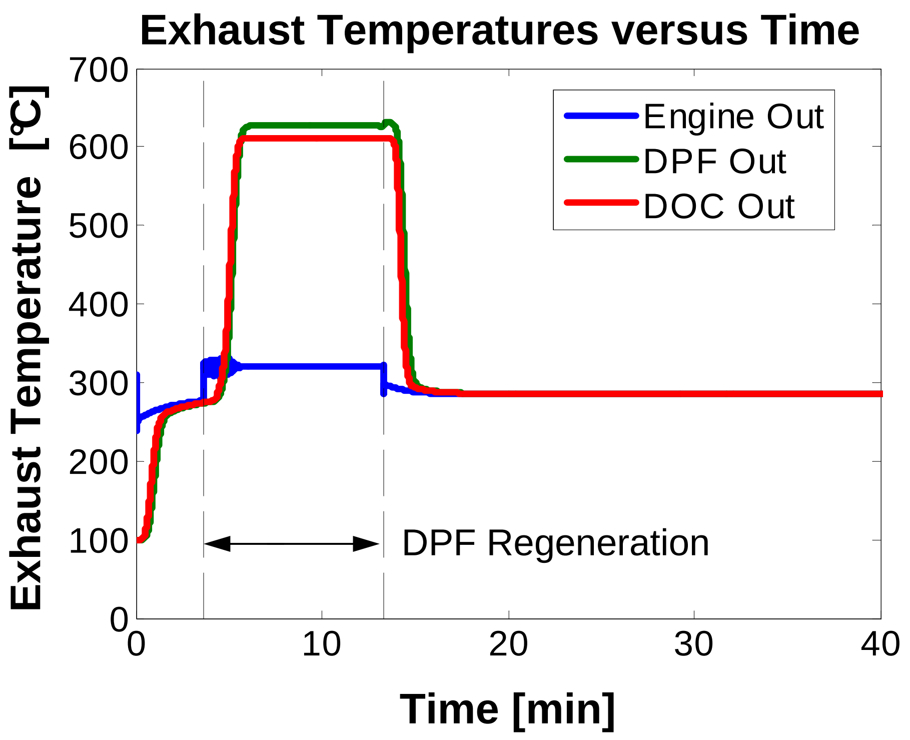
- Use ΔT predictive capability in the regeneration controller via simple energy balance

$$\dot{m}_{fuel} = \frac{\dot{m}_{exh} Cp_{exh} \Delta T_{exh}}{LHV_{fuel}}$$

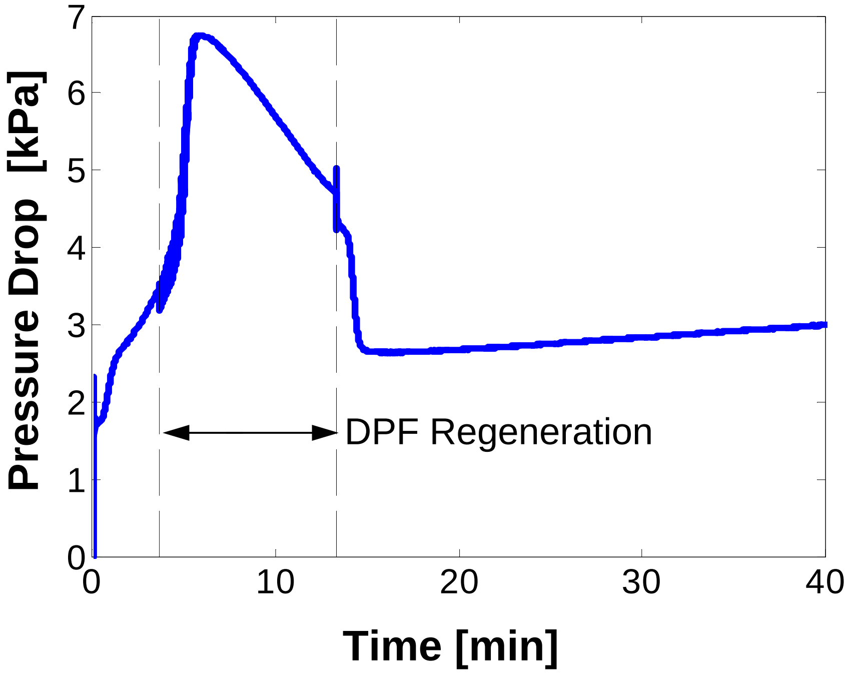
- Very simple energy balance will get the DPF inlet temperature within 10 [℃] from the target
- Use PI feedback control to adjust the predicted exhaust fuel injection rate to obtain the target DPF inlet temperature
- Feedback PI controller alone (without predictive capability) is not recommended

Throttle Assisted Regeneration (Mode 3)

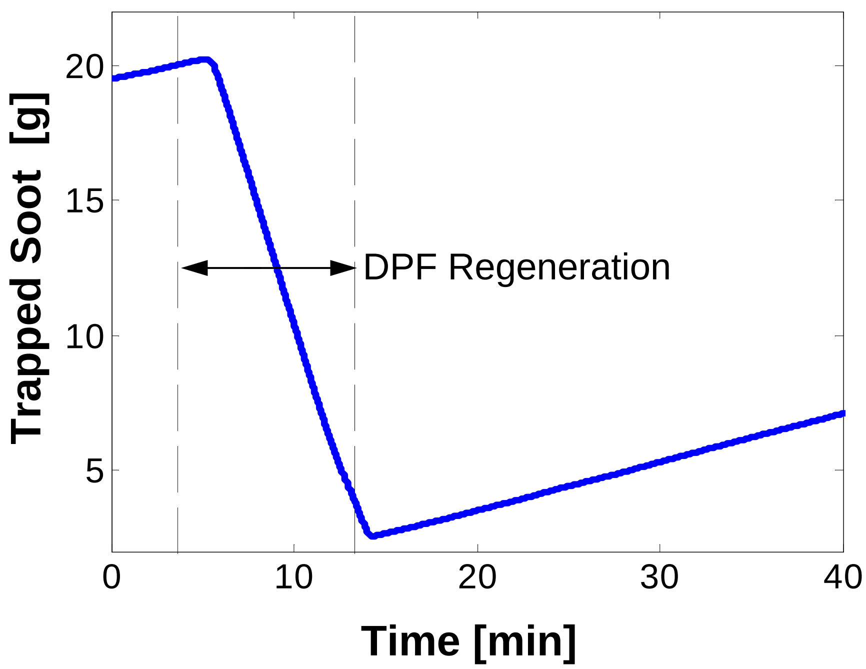

Fuel Consumption


Exhaust Temperatures versus Time

Throttle Assisted Regeneration (Mode 3)

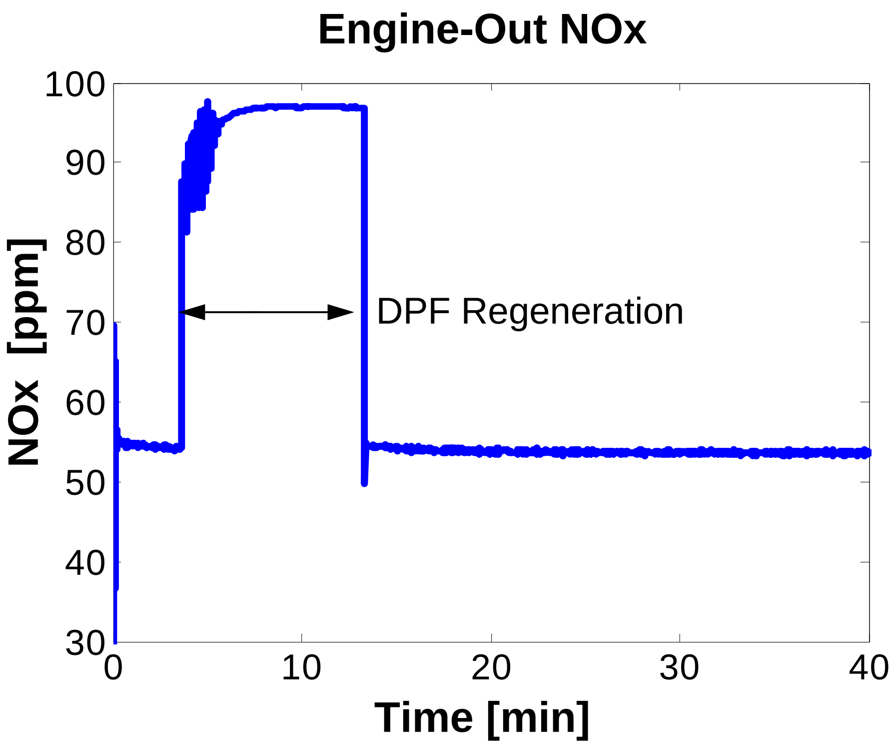


Pressure Drop Across DPF



Mass of Soot Trapped in the DPF

Throttle Assisted Regeneration (Mode 3)



Summary

- DOC exhaust inlet temperatures too low for exhaust fuel injection (DOC inlet temperature should be above ~300 [℃])
- Intake throttling increased the exhaust temperature from 276 [℃] to a target value of 320 [℃] (ΔT = 44 [℃])
  - Greater ΔT’s are possible
- Intake throttling was successfully used to enable DPF regeneration by increasing the exhaust temperature enough to allow for exhaust fuel injection

Conclusion

- An integrated model containing engine, emissions, aftertreatment, and controller sub-models has been created
- The integrated model has been tested under steady engine speed and load to confirm interaction of component models
- DPF regeneration has been simulated using fuel injected ahead of a DOC with and without intake throttle assistance
- The simulation results have given confidence for further development of the integrated model and to use the integrated model as a development tool

Future Work

- More active regeneration simulations
  - Investigate regeneration control strategies and impact on fuel consumption
  - Investigate strategies to stop a runaway regeneration
- Mode switching simulations (engine speed and load transients)
- FTP cycle simulation
- Add additional models for HC and CO emissions

Acknowledgments

Funding Sponsor: General Motors Research & Development

General Motors – University of Wisconsin Collaborative Research Laboratory

Thank You