Development of a Model Based Transient EGR Controller

Presentation to the GT-Suite Users Conference

Richard Stobart
Rob Bowyer

Arthur D Little

GT-Power User Group
Background to Project

The motivation behind this project is an interest on the part of Caterpillar to introduce a light duty diesel engine in the North American market.

- Light trucks are responsible for close to 50% of light duty sales in North America.
- A diesel solution could increase tank miles/gallon fuel economy by 50%.
- A 3% market penetration would offer a 0.5 MBPD oil import reduction.

The project was sponsored by the US Department of Energy. Caterpillar was main contractor and were supported by Perkins and Arthur D Little.
The Engine

The target engine is a 3 litre, V6 high speed diesel engine designed by Perkins

Main features include:

- Two stage injection
- Twin VNT turbochargers
- Fully modulated EGR valve
- 127 kW at 4000 rpm, 400Nm at 2000 rpm.
- 4 valves/cylinder
- Variable swirl capability
GT-Power engine model

Actuators:

- Flap angle
- Poppet valve area
- VGT rack position
- Fuel injection
- Engine load

Sensors:

- $m_{air}$, $m_{egr}$
- $T_{in}$, $T_{ex}$
- $P_{in}$, $P_{ex}$, and $P_{boost}$
- $N$, $N_{turbo}$
Engine model validation results

Steady-state validation:

- Mass air-flow was predicted very closely over operating range
- BMEP prediction also to within 10%
- Thus GT-Power accurate enough for simulation-based design
EGR layout

The control objective throughout the engine operating regime is a precise oxygen to fuel ratio

- The control system co-ordinates the VNT, the EGR valve and the flap
- Early work was based on a scheduled gain PI loop using an air flow estimator
Why?

The FTP-75 cycle examines EGR control through frequent and severe transient demands

- A simple control (gain scheduled PI) results in too frequent excursions below the demanded air/fuel ratio
- A fast electric EGR valve makes a significant difference to transient behaviour but not enough
- A new approach to control is needed which explicitly accommodates transients
Why Model Based?

The limitations of PI (error feedback) control are avoided:

– finite gain due to stability or power issues
– always “chasing” error - particularly when pure time delays exist in the system to be controlled
– does not handle constraints well

A model based approach offers advantages:

– can handle complex objectives
– extends well to multivariate systems
– fault detection and info for diagnosis comes “free”
The AFR control strategy

Block diagram

Speed_demand → Speed controller → Fuel per injection event

AFR_demand → * → Mair_demand → Mair controller → Throttle area (2-100%)

dP_demand → dP controller → VGT rack (0-100%) → dP > 0 → Poppet area (98-0%)
The AFR control strategy

Notes

• Mair is the mass of air flow per injection event

• Minimum Flap area is assumed to be 2%

• The Flap/Poppet combination act in complement as a single AFR actuator by selecting air from either the intake or the exhaust

• VGT used to maintain a fixed, positive pressure difference between exhaust and intake (dP) to ensure EGR when the Poppet valve is opened

• Poppet is closed if dP < 0 to prevent air escaping via the EGR pipe (energy wasted)
The AFR control strategy using CPower: Top level SIMULINK block diagram
Baseline controller: Description

**PI controller for Mair**

- Simple strategy to contrast and compare with the more advanced methods
- Integral de-saturation employed to prevent wind-up
- Non-linear PI gain scheduled on fuelling (low fuel $\Rightarrow$ low emissions, hence low PI gain to extend actuator life)
- The PI controller proved difficult to tune due to the non-linear behaviour of the system
- The demanded value of $dP=0.75\text{bar}$ was chosen arbitrarily. Further work is needed to obtain an optimal value for $dP$
Adaptive predictive controller for Mair: Description

Adaptation used to overcome non-linearity

- GPC cost function is based on Mair error squared with weighting on control moves determined by $\lambda$
- Predictions generated from linear CARIMA models
- RLS estimator used to provide an approximate linearised model about the current engine operating point
- Model order selected to give optimal trade-off between model fit (residuals) and model complexity (order)
- Future control actions computed by minimising the cost function subject to input magnitude and rate constraints
The Generalised Predictive Control optimisation problem

The GPC cost function

\[
J_{GPC} = \sum_{i=N_1}^{N_2} \left( w_{t+i} - \hat{y}_{t+i} \right)^2 + \lambda \sum_{i=0}^{Nu-1} \Delta u_{t+i}^2
\]

Key

- Setpoint \(w=Mair\text{\_demand}\)
- Output \(y=Mair\) (per injection)
- Control \(u=Flap/Poppet\)
The adaptation mechanism

- Future tracking errors drive GPC predictive controller to generate future control actions (using a linear plant model)
- Past estimation error drives MMRLS estimator to update the linear model parameters
Results: AFR control comparison
AFR control comparison

- Both the PI and predictive controllers improve the open loop AFR when the actuators have authority (i.e., not up against the actuator limits).

- Adaptive predictive GPC provides tighter control than PI due to the on-line adaptation mechanism.

- The noisy behaviour of AFR between 25 and 30 seconds is due to rapid opening and closing of the Poppet valve because dP is so low which is a result of the low turbo speed.
Adaptive GPC results using a smaller (faster) turbo

Adaptive GPC speed control ($N_I=10$, $N_T=25$, $N_U=1$, $\lambda=0.1$)

Adaptive GPC $M_a$ control ($N_I=5$, $N_T=12$, $N_U=1$, $\lambda=0.5$)

PI VGT control

SISO adaptive GPC AFR control

Arthur D. Little Cambridge Consultants

GT-Power User Group
Discussion

• Adaptive GPC speed controller tracks the FTP-75 speed profile closely (lines are on top of one another).

• Lack of authority at low turbo speeds: sudden fueling increases cannot be matched by a corresponding increase in air flow, even with the flap fully open and the poppet fully closed => problem: what size turbo?

• Demonstrates the use of the CPower simulation environment for determining the complicated trade-off between the size of the turbo, the demanded dP (both of which have an impact on fuel economy), and the required dynamic AFR response during transient engine loads and speeds.
Neural Network based EGR Control System

Controller Block Diagram

\[ f(x) = x^2 \]

Optimiser

Neural Network Model

Engine (CPower)

\[ r(t) \]

\[ y(t) \]
Neural Network (RBF) based predictive controller

• Start with a simple one-step ahead prediction horizon (minimum variance).

• The loop indicated by the dotted line represents the iterative optimisation calculation.

• The control signal which minimises the error function, J, is determined by searching the multi-dimensional input space within specified magnitude constraints.

• The optimisation routine may try up to 200 searches before a local minimum is found => need model predictions fast (as afforded by a Neural Network).

• At the end of each set of iterations a control signal is determined which is passed to the real engine actuators.
Neural Network results
Neural Network predictive controller for EGR

The NN control development faces certain numerical and process challenges

• The optimal settings determined by the NN controller are only as good as the NN model

• Obtaining a good NN model is not trivial
  – Working in ~20 dimensions!
  – Locating centres for the RBF structure is not intuitive
  – Data production (from GT-Power) and analysis is slow
  – Never been done before for an automotive application

• CCL have shown promising results but more effort is required before implementation
Conclusions

The development has shown the benefits of a process as well as *model based* control technology

- GT-Power was successfully used to support the control systems development
- Both RBF and linear adaptive controls worked well in the *C-Power* (GT-Power and Simulink) environment
- Some key control issues were exposed during the operation of the model control (turbo lag in particular)
- For non-linear controls speed of execution is vital - and the new mean value model is a welcome addition to the development suite.