Objective Drivability Calibration

MathWorks Automotive Conference

April 11th, 2019

Co-Authors:
Jason Rodgers &
Jan Janse van Rensburg*

MathWorks
Problem Statement

- What is the problem?
  - ECU can have dramatic effect on drivability
  - Manual calibration is time sink
  - Ratings are defined by experienced but subjective drivers

- How to solve the problem?
  - Use objective based approach to tune ECU calibration parameters
    I. Requirements driven
    II. Repeatable and automated
    III. Objective based
Key Takeaways

- Powertrain Blockset is capable of simulating some low frequency drivability behavior

- Model re-use from early planning phase can be used to jumpstart calibration efforts

- Objective-based calibration can:
  - Improve calibration time
  - Account for performance trade-offs
  - Trace back to requirements
  - Objective and not subjective
Agenda
Motivation & Background

- Development time
- Vehicle Variants
- Powertrain Complexity

[Diagram showing trends from 1980 to 2020]
Motivation

- **Current OEM Requirements**
  - Reduce overall operating costs
  - Improvement of vehicle quality for higher customer satisfaction
  - Develop Brand-Specific performance

- **Current OEM Constraints**
  - Decreased development time
  - Increased Powertrain complexity
  - Increasing number of vehicle variants
Motivation

- Current OEM Requirements
  - Reduce overall costs
  - Improvement of vehicle performance
  - Develop Brand-Strategy

- Current OEM Constraints
  - Decreased development time
  - Increased Powertrain complexity
  - Increasing number of vehicle variants

How to juggle requirements and constraints?

Increase efficiency during the early development process!
Motivation

Efficiency Improvements

• Model-Based Development (Process Virtualization)

• Model Reuse

• Objective-Based Calibration Process
Motivation

Efficiency Improvements

• Model-Based Development (Process Virtualization)
  – Front-Loading Development Process
  – Virtual Calibration
  – Check new controller designs
  – Early detection of design deficiencies
  – Reduced number of prototypes
  – Etc.
Motivation

Efficiency Improvements

• Model Reuse
  – FE/Acceleration models for tip-in
  – Early calibration

• Objective-Based Calibration Process
  – Requirements driven
  – Traceable
  – Repeatable
  – Automated
  – Optimal
Background

What is drivability?
- Response characteristic of the vehicle to driver inputs under different driving conditions

- Want the driver to be as comfortable as possible
  - Hesitation
  - Sluggish
  - Hard start
  - Noise/Oscillations

- Drivability is affected by many sources
  - Gear shifts
  - Engine Idle
  - Braking
  - Acceleration
  - Etc.

Background

What are we focusing on?

Background

What are we focusing on?

- Shuffle related to tip in
  - NVH longitudinal effect caused by sudden changes in the drive torque
  - Some room to optimize hardware but controller is more cost effective
  - 2-8 Hz depending on the gear
- Not considering shift shock, clunk, or higher order modes
- <5hz – human feel threshold
- Acceleration is measured at CG
HEV Plant Modeling
Powertrain Blockset – P4 HEV Model

Various Component Modeling Types
- First Principles
- Data-driven
- Balance between accuracy and speed

P4 HEV Architecture
Powertrain Blockset – P4 HEV Model

- **P4 HEV Powertrain model**
  - Started from reference application and modified for testing and added tip-in controller
  - Model fidelity is typical for FE and acceleration studies

- **Engine**
  - 1.5L L4 95kW(126hp) @5500RPM
  - Map-based Model

- **2 P4 30kW Motors**
  - Map Based Model

- **1.3 kWh Battery**
  - Map-Based Model
P4 Component Modeling

- Driveline oscillations are captured by rotational inertia and compliance blocks that exist in reference model

- Linear damping and stiffness
  - Openness of model allows for replacing K/B with nonlinear terms

- 2 Torque Paths
  - Engine
  - Motor
Driving Scenario

What scenario are we using?

1. Accelerate to Constant Speed
2. Hold Speed and shift to desired gear. Allow transients to subside.
3. Let off pedal
4. Apply pedal step input
Tip-In Acceleration Response

- Initial response has large amounts of shuffle oscillations
  - Model is able to capture the first mode (shuffle) for both torque paths
  - Response attenuation is required to improve drivability
Tip-In Acceleration Response

- How to improve?
  - Spark Control (on engine side only)
  - Fixed Rate Limit on torque request or pedal input
  - Scheduled Rate Limit
  - Optimal Control – e.g. Model Predictive Control

- Scheduled Rate Limit can be tuned for each case by engineer-> long manual process (weeks)
- Reduced oscillations but response is slow
- How to balance responsiveness and oscillations
Tip-In Acceleration Response

- How to improve?
  - Spark Control (on engine side only)
  - Fixed Rate Limit on torque request or pedal input
  - Scheduled Rate Limit
  - Optimal Control

Define an objective!

- Scheduled Rate Limit can be tuned for each case by engineer-> long manual process (weeks)
- Reduced oscillations but response is slow
- How to balance responsiveness and oscillations
Defining an Objective Function

What are my goals?

What are my choices?

What restricts my choices?
Optimization Introduction

- **Objective** function – What you are trying to achieve?
  - Minimize measured signal

- **Design variables** – What parameters need to be adjusted?
  - Physical model parameters
  - Controller gains

- **Constraints** – What are the bounds or constraints of the design variables?
  - Min/Max values
  - Parameter dependencies

Minimizing (or maximizing) objective function(s) subject to a set of constraints

**Objective Function**

\[
\min_x f(x)
\]

Design variables (discrete or integer)

**Linear constraints**

\[
Ax \leq b \\
A_{eq}x = b_{eq} \\
l \leq x \leq u
\]

**Nonlinear constraints**

\[
c(x) \leq 0 \\
c_{eq}(x) = 0
\]
Formulating an Optimization Problem for Objective Drivability

What are my goals?
- Minimize oscillations
- Minimize response time

Variables
What are my choices?
- Rate limit
  - Gear
  - $\Delta$Torque Request
  - Vehicle speed

Constraints
What restricts my choices?
- Response Time
- Jerk
- Etc.

Objective
Objective Function

\[
\min J = 0.5(t_{\text{resp}}^*) + \text{ jer}k_{\text{max}}^* + 0.5(VD)^* + \text{ constraints}
\]
Cost Function Metrics

- **Response Time**
  - $t_{resp} = \text{time to reach 50\% steady state acceleration}$
  - Normalized by the slowest desired response time (1s)
  - Defined this way to account for edge cases where motor or engine cannot provide enough torque

Example: Low engine speed with high torque request
Cost Function Metrics

- Vibration Dose Value (VDV)
  \[ VDV = \left( \int_0^T a^4(t) \, dt \right)^{1/4} \]
  - VDV is sensitive to the peaks in the acceleration.
  - Normed to the maximum response with no rate limit

- Maximum Jerk
  \[ jerk_{\text{max}} = \max \left( \frac{da}{dt} \right) \]
  - Normed to the maximum jerk obtained with no rate limit
Cost Function Constraints

- Response Time $\leq 1$sec
Cost Function Constraints

- Response Time $\leq 1$ sec
- Maximum Jerk $\leq 2\frac{m}{s^3}$
Cost Function Constraints

- Response Time $\leq 1$sec
- Maximum Jerk $\leq 2\frac{m}{s^3}$
- $acc_{\text{final}} \geq 0.95acc_{\text{final}}^*$
  - $acc_{\text{final}}^*$ is the steady state acceleration with no rate limit
  - useful for edge cases
Objective Function

- Pareto curve exists between oscillations and response time – the faster the response, the more oscillations

\[
\min_{RL^*} J = 0.5(t_{resp}^* + \text{jerk}_{max}^*) + 0.5(\text{VDV}^*) + \text{constraints}
\]

With, \(\text{constraints} = \begin{cases} 10^6 \text{ if violated} \\ 0, \text{ otherwise} \end{cases}\)

- Objective function can be:
  - non-smooth
  - can have multiple minima
Optimal Calibration
Calibration Process

- Intel Xeon E5 processor – 3.6GHz, 6 cores
- 64GB RAM
- 1806 speed, torque change points
  - 7 total maps (6 for engine, 1 for motor)
  - 24 Δtorque breakpoints
  - 5 speed breakpoints

- Traditionally, this process could take days or weeks for manual calibration
- 10 hours to automatically calibrate using pattern search global optimization algorithm

<table>
<thead>
<tr>
<th>Search Algorithm</th>
<th>Time</th>
<th>Solution Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>fmincon</td>
<td>1.5 minutes</td>
<td>✗</td>
</tr>
<tr>
<td>Particle Swarm</td>
<td>5 minutes</td>
<td>✓ +</td>
</tr>
<tr>
<td>Pattern Search</td>
<td>1.5 minutes</td>
<td>✓</td>
</tr>
</tbody>
</table>
Tip-In Controller

- Rate limit is calculated as a function of $|\Delta\text{Torque request}|$, vehicle speed, and Gear (engine side only)

- Rate limit is applied when judged a tip in response
  - $|\Delta\text{Torque request}| > 10\text{Nm}$
  - Vehicle Speed $> 2\text{ MPH}$

- Rate limit held until modified torque is near final desired torque value.
Tip-In Controller

![Graphs showing engine and motor torque request with time and pedal percentage](image-url)
Calibration Tables

- Areas of high sensitivity in the objective function can be used to redefine map breakpoints
- Example results for 5th gear
Validation
Tip-In Results

- First engine and motor modes have decreased greatly (~50dB)
- Fast Tip-In response – 0.5s
Next Steps

- What are possible next steps?
  - Investigate other control types
    - Model Predictive Control with consideration for FE for example.
Next Steps

- What are possible next steps?
  - Investigate other control types
    - Model Predictive Control with consideration for FE for example.
  - Process can be reused as model fidelity increases
    - GT Engine model
    - Simscape driveline
  - Utilize process for other calibrations
Summary

- A process for using objectives to automate and improve shuffle response was shown.
- Virtual calibration allowed process to be done in hours instead of weeks.
- Along with FE and Acceleration characteristics, can also start to consider some drivability metrics during early phase planning.
Key Takeaways

- Powertrain Blockset is capable of simulating some low frequency drivability behavior

- Model re-use from early planning phase can be used to jumpstart calibration efforts

- Objective-based calibration can:
  - Improve calibration time
  - Account for performance trade-offs
  - Trace back to requirements
Questions?