Systematic Process for Physics Based Modeling and Model Predictive Control of Engine and Aftertreatment Systems

D. Kihas[a,1] and M. Uchanski[b,2]

a Honeywell Automotive Software
500 Brooksbank Ave, North Vancouver, British Columbia V7J 3S4, Canada
b Honeywell Automotive Software
ZA La Pièce 16, 1180 Rolle, Switzerland

Abstract

To make engine and aftertreatment hardware choices for new emissions programs, it is often useful to prototype several candidate engine, aftertreatment and control systems, and run certification cycles. Building such a prototype requires a *hardware selection control system* that gracefully adapts to new hardware and rapidly achieves a level of performance similar to that of a production calibration. This paper assesses an implementation of Model Predictive Controls (MPC) against requirements for a *hardware selection control system.* We use metrics captured from 15 projects at 12 automotive OEMs over a period of several years. Metrics include hardware variety, test bed usage, and certification cycle performance relative to production calibrations for projects where production calibrations were available. The data show that a controls design process using MPC and encapsulated in Honeywell OnRAMP software achieves near-production calibrations in a mean time of 6 days of test bed time. We conclude with remarks on the applicability of such a modeling and control framework in an industrial context and an outlook to the future.

Key Words: Hardware Selection, Control System, Diesel airpath control, Aftertreatment Control, Model Predictive Control, MPC, Optimal control, Multivariable control, Mean value engine model, Calibration, Performance metrics, OnRAMP software, MPC performance in automotive applications, Controller comparisons, Emissions control

---

[1] E-mail: dejan.kihas@honeywell.com, www.honeywellonramp.com
[2] E-mail: michael.uchanski@honeywell.com, www.honeywellonramp.com

1 Introduction

Hardware selection is a system-level task often encountered in early phases of powertrain emissions programs. This paper assesses the fit between Model Predictive Control (MPC) and the hardware selection task. We use the following approach:

- Develop controls requirements for hardware selection (Section 2)
- Describe a MPC-based controls targeted to requirements (Section 3)
- Use metrics taken from a large number of projects to assess how well the MPC controls approach meets requirements (Section 4)

The discussion focuses primarily on engine airpath and aftertreatment controls, even though MPC can be considered for any hardware selection task. Section 5 touches on problems beyond aftertreatment and engine controls.

Two types of literature provide context and motivation for this work: industrial surveys and controls comparison studies.

Examples of industrial surveys include [18] and, more recently, [12]. Both articles provide good overviews of the state of the art of automotive control design and cite the need to improve it through "model-based approaches that help shorten dyno times" and "systematic design with physical-based and experimentally gained mathematical models."

Controls comparison studies typically develop a new controls approach, apply the approach to a problem such as airpath or aftertreatment controls, and then draw performance conclusions. Reference [9] provides a well-structured, recent survey of this type of work.

This article builds on both types of literature. It proposes an MPC-based approach that is "systematic" and thrifty in dyno time as called for in the industrial surveys. It then expands on controls comparison studies by providing a performance comparison of the MPC approach versus production baselines using data from 15 projects on a variety of production engines over a period of several years.

The data indicate that the MPC approach is systematic and highly flexible, that it delivers performance similar to production baseline controllers in 3-11 days, and that it often improves actuator activity. This combination of flexibility, speed, and performance makes the MPC-based approach particularly well-suited for hardware selection problems. In addition, it may provide performance benefits in applications where actuator activity has significant value.

2 Control Requirements for Hardware Selection

Hardware selection teams are typically small interdisciplinary teams tasked with identifying a recipe of combustion strategy, air and EGR handling, aftertreatment, sensors and actuators that meet program requirements for reliability, drivability, fuel economy, emissions compliance, diagnostic compliance, and cost. Hardware selection is a combination of paper studies, simulation, steady state open-loop experimentation, and drive cycle tests for drivability and transient emissions. Requirements for a control system to use in this work include:

- **Fast turn-around and low test bed time**, which allow for more hardware iterations in a given calendar time.

- **Near optimal performance**, so that hardware performance is not masked by controls performance.

- **Flexible with respect to hardware changes**, for example to allow comparison of VGT and wastegate turbocharger options or of high pressure EGR vs. dual loop EGR. Notably, the system should transparently handle the complex multivariable interactions that often arise when a new actuator is added.

- **Interchangeable between simulation and dynamometer**, so that the control system can be coupled with simulation models for virtual screening. This is especially important for companies that buy their control systems as a package with the ECUs from Tier 1 suppliers and so may have difficulty building and maintaining a simulation model for the production controls.

- **Constraint management**, in order to avoid damage to costly prototype hardware and reduce the time and care which must be taken to develop setpoint tables.

- **Production readiness**, so that the prototype control system can gracefully merge into the production workflow if desired.

3 Control Design Process

OnRAMP control design software [1] tools a systematic MPC based control design process. It makes the control design and tuning process more efficient and less dependent on the skill level of the developer than most traditional controls approaches. The design process was developed with the goal of integration into industrial practice [3]. Careful consideration was given to existing development processes for industrial engine control and how the resulting control function interfaces in the hierarchical software structure [14], [21].

A typical sequence of MPC control design steps supported by the mentioned software package is listed below [2], [3], [20]:

1. Definition of the model structure based on the engine configuration and physical laws
2. Design and execution of test cell experiment in order to acquire the data for model calibration (3 test cell days)
3. Automated model calibration based on the obtained experimental data and available component data (e.g. turbocharger maps)
4. Controller configuration definition that includes specification of inputs and outputs
5. Definition of the control optimization problem, the cost function in terms of setpoints, constraints and their relative importance factors
6. Synthesis of the controller dataset that is deployed with the controller on rapid prototyping system or engine control unit
7. Controller fine tuning calibration in the test cell (Typically 3 test cell days)

It typically takes six days of test cell time to reach a near-production calibration on a new hardware configuration. Other work, including project organization, ECU setup, and desktop calibration (steps 1,3,4,5,6) typically takes another 10 days on the calendar for the first calibration. For subsequent calibrations, non-recurring work is removed and recurring work becomes routine, so time per iteration can substantially improve.

3.1 Model Calibration

The model calibration uses the measurement data to fit the model parameters so that the behavior of the true engine is closely matched.

The model calibration step is carried out in two major steps – steady-state matching and transient identification. Steady-state matching includes: 1) steady-state component-level identification that provides reasonable starting points for the global steady-state identification, and 2) the global steady state identification that reconciles the components in order to achieve very high accuracy of the overall model. Finally, transient identification step identifies remaining dynamic parameters. The tool uses an optimization algorithm based on nonlinear least squares for all stages of model identification [3].

3.2 Controller Structure

The controller structure calibrated by OnRAMP design process includes MPC feedback controller with Kalman Filter used as observer, feedforward static look-up tables, constraints manipulation element [2], [15], [16], [20], [21], [22]. The complete structure is scheduled on several selected variables that describe the changes of behavior (including nonlinearities) of the targeted system over the operating range [14], [23].

The main role of feedforward control is to show a correct steady-state actuator position to the controller during heavy transients across a wide set of operating points where the model uncertainty is very significant. The main role of feedback control is to provide steady-state offset-free tracking, transient behavior improvements, constraints handling and disturbance rejection. While feedforward control can react faster during heavy transients and provide initial actuator positions, feedback will ensure adjustments of actuator positions to the desired values. The combination of these two can be governed by feedback range concept to ensure that actuators' signals are constrained to lie within specified constant range from feedforward signal and in that way provide safe control actions in broad set of highly nonlinear engine applications with large transients [22].

3.3 Model Predictive Control Algorithm

Model-based Predictive Control (MPC) is a widely accepted method for designing optimal multivariable control. The main advantage of MPC over the other control techniques, such as H2, LQG, $H\infty$, mu-synthesis (e.g. [24]), is its systematic approach to regulate process with various time-varying constraints on the system inputs, internal states, and outputs. MPC algorithm computes an optimal future trajectory of selected system inputs so that the plant behavior is as close as possible to the requirements. The control goals are expressed by a cost function that contains weighted combination of terms for actuator movements, tracking error and soft constraints. The cost function is minimized by the MPC algorithm over a selected prediction horizon.

The basic MPC cost function with soft constraints can be expressed as:

$$\min_{U_{N_u};\,\varepsilon_1} J(U; x(k); v(k)) = \sum_{j=0}^{N_y} \|e(k+j|k)\|_2^Q + \|\delta u(k+j)\|_2^P + \|\rho_1 \cdot \varepsilon_1\|_2^2$$
$$subject\ to:$$
$$y(k) = G_\sigma(x(k), u(k), v(k)) \qquad \text{(Eq. 01)}$$
$$u_{min} \le u(k+j) \le u_{max}$$
$$y_{min} - \varepsilon_1 \le y(k+j) \le y_{max} + \varepsilon_1$$

Where the following notation is used: Setpoint tracking error is denoted as: $e_i(k+j|t) = (r_i(k+j|t) - y_i(k+j|t))$; Actuator moves as: $\delta u_i(k+j) = (u_i(k+j) - u_i(k+j-1))$; weight on tracking error (CV penalty): *Q;* weight on actuator moves (MV penalty): *R;* Violation of limits or constraints: ε_1; Soft limit penalty weight: ρ_1; $\sigma(k)$ is the operating point index at the time kT_s as a function of $(x(k), u(k), v(k))$; prediction horizon: N_y; control horizon: N_u. Further, $y(k) = G_\sigma(x(k), u(k), v(k))$ represents the set of linearized discrete time approximations of the system model over the operating range:

$$\dot{x}(t) = f(x(t), u(t), v(t))$$
$$y(t) = g(x(t), u(t), v(t))$$

More on MPC can be found in the following literature [2], [4], [5], [6], [7], [8], [10].

3.4 Automatic Tuning Algorithm

The relationship between weights in the cost function of the MPC problem and the desired robustness and performance conditions can be established through frequency domain analysis of a type of transformed MPC control problem. In this way, it is possible to derive robust stability condition using small gain argument applied to additive uncertainty model.

The algorithm manipulates multipliers of weights in the cost function such that the derived robust stability condition is satisfied. The user defined requirements on robustness and bandwidth affect the shape of the derived robust stability condition and in that way the resulting weights of MPC cost function are influenced by the specified requirements for robustness and bandwidth [19], [25].

4 Metrics and Control System Evaluation

To evaluate the MPC-based controls methodology of Section 3 against the requirements of Section 2, we mined data from 15 selected customer projects covering a period of 7 years. The input for each was a new engine or aftertreatment system with steady state setpoints. The output was a transient drive cycle in closed loop control on a dynamometer, or on a vehicle, or both. In this respect, each project resembles an iteration of hardware selection. In several cases, they were executed for that purpose. In the remaining cases, they were executed as evaluations on a production engine in order to assess how quickly the MPC controller of Section 3 could match a production baseline calibration.

Customer names and other information that might identify the engine and aftertreatment systems involved have been suppressed.

Table 1. below provides an indication of controller flexibility. Each row is one engine or aftertreatment project. Each column is an actuator, setpoint, or constraint that is used for aftertreatment and engine control. Colored squares indicate the combination of actuators, setpoints and constraints used on each project.

All control problems in the table were addressed with the same core MPC controller code, which solves the optimization problem of (Eq. 01). Only the parameters of the linear plant models, of the optimization weightings, and of related items were changed. Note that even such different applications as engine airpath control and DOC temperature control used the same control code and required changes only to these calibration parameters. See [16] for a complete account of the DOC work.

	Actuators									Setpoints											Constraints											
Hardware Configuration	EGR cooler bypass	High Pressure EGR	Low Pressure EGR	Variable Geometry Turbine (VGT)	Intake Throttle	Exhaust Throttle	Back Pressure Valve	Waste Gate	DOC inlet HC mass flow	Intake Manifold Air Pressure	Mass Air Flow	Total Inducted Flow	Temperature at exhaust	EGR Mass Flow	EGR fraction	NOx concentration at exhaust	Lambda	Pressure at Turbine Inlet	Intake Manifold Oxygen Mass Fraction	Temperature at the doc outlet	Turbocharger speed	Lambda	Temperature at Intake Manifold	Temperature at Exhaust	Temperature at Compressor Outlet	Pumping Losses	Pressure at Exhaust Manifold	Mass Air Flow	Manifold Air Pressure	Temperature at Turbine Inlet	Temperature at Intercooler Outlet	Temperature at DOC outlet
15		X	X	X			X			X				X																		
14		X		X								X			X						X											
13		X		X			X									X		X			X	X		X			X					
12		X						X		X				X								X										
11		X						X		X							X															
10		X			X			X					X			X						X							X			
9		X		X	X						X				X						X			X		X						
8		X		X						X	X										X	X		X								
7		X		X	X					X									X		X		X	X	X		X	X				
6	X	X		X	X					X	X												X	X	X							
5				X						X											X									X	X	
4		X						X		X	X																	X	X			
3				X						X											X											
2			X			X				X				X		X																
1									X											X												X

Table 1. - Aftertreatment and engine control problems run with the MPC controller described in Section 3.

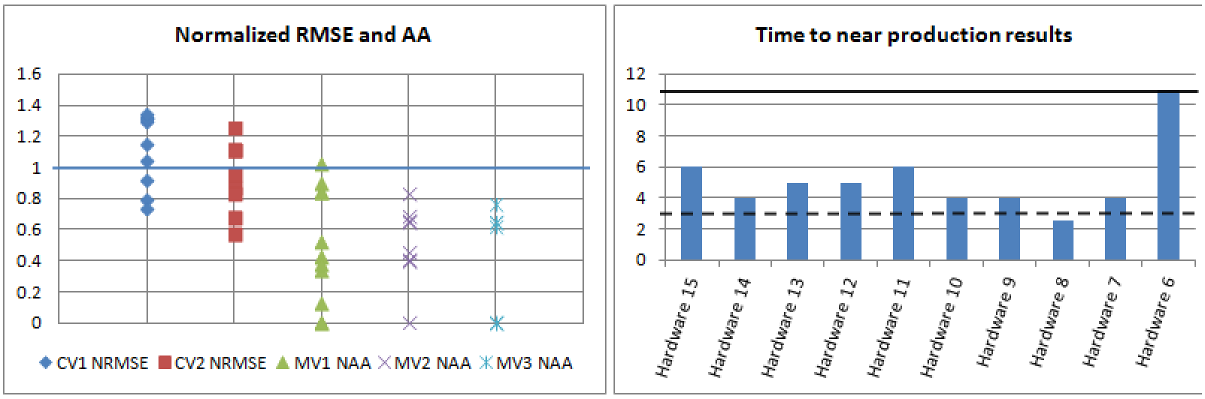

Figure 1 – Left: MPC Setpoint tracking RMSE and AA metrics normalized against production. Right: Test cell time required to achieve results. Dashed and solid lines are markers for 3 and 11 days respectively.

Figure 1 above presents metrics used to assess controller performance and test bed time. For 7 of the 15 hardware configurations in Table 1 production calibrations were available as baselines. A typical production calibration undergoes months of fine tuning, so we will assume that the production calibration delivers performance close to the best that could be extracted from that hardware set. For the 7 hardware configurations where a production baseline was available, we plotted production-normalized performance metrics (described below) for 2-3 MPC calibrations for that configuration. Four calibrations that produced obvious outliers and could be traced to test data issues were excluded.

To measure controller performance, we calculate the root mean squared setpoint tracking error:

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(r(j) - y(j)\right)^{2}}$$

for both the production controller and the MPC controller. We then normalize the RMSE of the MPC controller by that of the production baseline. The left side of Figure 1 shows two normalized root mean squared error (NRMSE) quotients

$$NRMSE_{CV1} = \frac{RMSE_{MPC\_CV1}}{RMSE_{PROD\_CV1}} \quad \text{and} \quad NRMSE_{CV2} = \frac{RMSE_{MPC\_CV2}}{RMSE_{PROD\_CV2}}$$

The subscripts CV1 and CV2 refer to controlled variables for each problem (for example, MAF and MAP). For problems with several setpoints to track (most of our sample), NRMSE quotients for the two setpoints that most impact system performance were used. For problems with only one setpoint to track, only the NRMSE quotient for setpoint 1 is shown. Values less than one indicate better setpoint tracking than the production baseline controller (please note that perfect setpoint tracking would generate an index of 0).

Since low NRMSE setpoint tracking metrics can be achieved by high-gain, and very active controllers, we also compute a metric for sum of squares actuator activity:

$$AA = \frac{1}{N}\sum_{j=1}^{N-1}\big(u(j+1)-u(j)\big)^2$$

for both the production baseline (AA_{PROD}) and the MPC controller (AA_{MPC}) and normalize by the production baseline value, as we did for RMSE. The left side of Figure 1 also shows three normalized actuator activity (NAA) quotients:

$$NAA_{MV1} = \frac{AA_{MPC\_MV1}}{AA_{PROD\_MV1}} \quad NAA_{MV2} = \frac{AA_{MPC\_MV2}}{AA_{PROD\_MV2}} \quad NAA_{MV3} = \frac{AA_{MPC\_MV3}}{AA_{PROD\_MV3}}$$

The subscripts MV_1, MV_2, and MV_3 refer to the actuators (manipulated variables) that participate in the tracking problem, for example, EGR valve, VGT actuator, and intake throttle angle. Projects with only one actuator show only NAA_{MV1}, and problems with more than three actuators show activity on the three actuators that most influence setpoint tracking. Values less than one indicate better actuator activity.

The NAA metrics in the left graph in Figure 1 show that every project measured exhibited better (lower) actuator activity than the production baseline. The mean actuator activity quotient was 0.5-0.6 depending on the actuator. The lower actuator activity may be attributed to the look-ahead nature of the MPC controller implied in the optimization problem of (Eq. 01) and its ability to coordinate the multivariable interactions of actuators on the engine.

The NRMSE setpoint tracking metrics show that the MPC controller sometimes delivered better setpoint tracking than the production controller (NRMSE quotients less than one) and sometimes the production controller delivered better setpoint tracking (NRMSE quotients greater than one). The mean NRMSE values for setpoint 1 and setpoint 2 were 1.13 and 0.92 respectively. A few of the projects with NRMSE setpoint tracking quotients greater than one also exhibited small actuator activity metrics, implying that actuator activity could have been traded-off to improve setpoint tracking. Nonetheless, there were also cases where both NRMSE and NAA were less than one and indicating that OnRAMP controller performed closer to Pareto optimal curve than the production one.

The right side of Figure 1 shows the dynamometer time required to tune the calibrations shown on the left side of Figure 1 and in Table 1.

In some cases this data was not available, so the graph contains fewer projects than Table 1. The mean dyno time is 6 days, and the range is 3 days to 11 days. It is noticeable that the project (market as "Hardware 6") took distinctly longer time than the others did. This is because the controller was tuned on a wider variety of maneuvers, taking 4-5 days longer than usual.

The variety of Table 1, combined with the performance and dynamometer time graphs of Figure 1 imply that the MPC controller delivers calibrations at roughly equivalent performance to production calibrations in a mean of 6 days, over a wide variety of engine and aftertreatment configurations. We therefore conclude that MPC adequately covers the first few requirements of Section 2 for a hardware selection controller.

Other controls requirements from Section 2 are *Interchangeability between simulation and dynamometer*, *Constraint management* and *Production readiness*. Interchangeability between simulation and dynamometer was demonstrated in one of the projects of Table 1, which was completed in simulation using a medium fidelity engine model in a well-known commercial-off-the-shelf engine modeling package. Constraint management is built into the MPC optimization problem (Eq. 01) and is one of the differentiating features of the MPC approach. Finally, Production readiness is built into the particular implementation of the MPC controller used for this study, because it has been coded according to MISRA guidelines and has been optimized to fit in roughly 70kB (depends heavily on problem size and number of constraints). One of the projects in Table 1. was executed directly on a mass-market automotive ECU.

5 Summary and Future work

This study examined the appropriateness of Model Predictive Control (MPC)-based controllers for the hardware selection phase of emissions projects. The data we examined here, as well as our conversations with several automotive OEMs, suggest that there is a good fit.

We discussed six requirements for a good hardware selection controller. We then examined data from 15 emissions projects over 7 years to assess MPC on use of development resources, on flexibility, and on performance.

The same core MPC code worked for 14 different airpath problems and one aftertreatment problem with major changes only to the controller's calibration parameters. Performance and dynamometer usage metrics imply that the MPC controller delivered calibrations at roughly equivalent performance to production calibrations in a mean of 6 days. This combination of flexibility and ability to quickly attain good performance levels positions MPC-based controllers as a good choice when there is a need to quickly get new hardware running for performance evaluations.

Feedback from our partners at automotive OEMs suggest several areas for future development:

- *Control system cost reduction* – In some cases, different projects in Table 1. fulfilled very similar product requirements using significantly different sensor and actuator combinations. This raises the question: which combination meets requirements at the lowest cost? The answer to this question is being explored in the context of evaluating closed-loop performance of the system when facing realistic engine or aftertreatment variability reflecting production dispersion or ageing.

- *Increased generality* – The control design process and MPC code described here has the potential for generalization beyond just DOC and engine airpath controls problems. The crux of this problem is the development of plant models for new phenomena that can easily and systematically be optimized over the plant's operating space. Several new directions are currently underway with OEM partners and will be reported in future publications.

- *Improved production readiness* – As MPC builds a following in the advanced technology and hardware selection community, we see parallels with C code generation tools that gained popularity in the automotive controls community in the early 2000s. At first these tools were used with rapid prototyping ECUs to quickly test ideas. However, as engineers grew used to the fast turn-around and convenience of code generation, they pushed software tool vendors to build in features required for use on production vehicles. To address the analogous needs, our team is working to extend and improve calibration tool support and other features requested by customers with an eye beyond just hardware selection.

Acknowledgements

The authors would like to thank colleagues from Honeywell Automotive Software team for useful discussions, constructive feedback, and providing the summaries and relevant data related to past projects.

References / Literature

[1] Honeywell OnRAMP website: http://www.honeywellonramp.com

[2] Stewart, G.E., Borrelli, F., Pekar, J., Germann, D., Pachner, D., and Kihas, D., "Toward a Systematic Design for Turbocharged Engine Control," in L. del Re, F. Allgower, L. Glielmo, C. Guardiola, "Automotive Model Predictive Control," Lecture Notes in Control and Information Science, Springer-Verlag, Berlin-Heidelberg, 2010.

[3] Pachner, D., Germann, D., and Stewart, G.E., "Identification Techniques for Control Oriented Models of Internal Combustion Engines," in Alberer, D., Hjalmarsson, H., del Re, L., "Identification for Automotive Systems," Lecture Notes in Control and Information Science, Springer-Verlag, Berlin-Heidelberg, 2012.

[4] Maciejowski, J.M., "Predictive Control with Constraints," Pearson Education Ltd, 2002.

[5] Bemporad, A., Morari M., Dua, V., Pistikopoulos, E. N., The explicit linear quadratic regulator for constrained systems. Automatica, 38, 3-20, 2002.

[6] Muske, K. R., Badgwell, T. A., "Disturbance modeling for offset-free linear model predictive control", *Journal of Process Control 12*, 617-632, 2002.

[7] Maeder, U., Morari, M., "Offset-free reference tracking with model predictive control", *Automatica 46*, 1469-1476, 2010.

[8] Borrelli, F., Baotic, M., Pekar, J., Stewart, G., "On the computation of linear model predictive control laws", *Automatica 46*, 1035-1041, 2010.

[9] del Re, L., Allgower, F., Glielmo, L., Guardiola, C., "Automotive Model Predictive Control", *Lecture Notes in Control and Information Science,* Springer-Verlag, Berlin- Heidelberg, 2010.

[10] Qin, S. J., Badgwell, T. A., "A survey of industrial model predictive control technology", *Control Engineering Practice 11*, 733-764, 2003.

[11] Skogestad, S., and Postlethwaite, I., "Multivariable Feedback Control: Analysis and Design," Wiley & Sons, 1996.

[12] Criens, C., Air-Path Control of Clean Diesel Engines for disturbance rejection on Nox, PM and fuel efficiency, PhD Thesis, Technische Universiteit Eindhoven, 2014

[13] Guzzella, L., Onder, C.H., “Introduction to Modeling and Control of Internal Combustion Engine Systems,” Springer-Verlag, 2009.

[14] Stewart, G.E., and Borrelli, F., “System for gain scheduling control”, USPTO number 7603185, October 2009.

[15] Khaled, N., Pekar, J., Fuxman, A., Cunningham, M., and Santin, O., “Multivariable Control of Dual Loop EGR Diesel Engine with a Variable Geometry Turbo.” SAE World Congress, 2014, paper 2014011357.

[16] Kim, Y.W., K., Van Nieuwstadt, M., Stewart, G., Pekar, J., “Model Predictive Control of DOC Temperature during DPF Regeneration.” SAE World Congress, 2014, paper 2014011165.

[17] Isermann, R., “Model-Based Design Workflow for Engine Control Systems.” MTZ Worldwide, Vol. 75, March 2014.

[18] Guzzella, L. and Amstutz, A., “Control of Diesel Engines”, IEEE Control Systems, Vol. 18 (5), October 1998, pp 53-71.

[19] Fan, J., and Stewart., G.E., Patent “Automatic tuning method for multivariable model predictive controllers”, USPTO number 7,577,483, May 2006.

[20] Alfieri, V., Pachner, D., “Enabling Powertrain Variants through Efficient Controls Development”, SAE World Congress, 2014.

[21] Stewart, G.E., Borrelli, F., “A Model Predictive Control Framework for Industrial Turbodiesel Engine Control”, 47th IEEE Conference on Decision and Control, Cancun, Mexico, December 9-11, 2008

[22] Pekar, J., Stewart, G.E., Kihas, D., Borrelli, F., Patent “Method and system for combining feedback and feedforward in model predictive control”, USPTO 8,145,329, March 2012

[23] Rugh, W., and Shamma, J., Research on gain scheduling. Automatica, 1401–1425, 2000

[24] Doyle, J., Francis, B., and Tannenbaum, A., "Feedback Control Theory", 1990

[25] Fan, J., Model Predictive Control for Multiple Cross-Directional Processes: Analysis, Tuning, and Implementation, PhD Thesis, University of British Columbia, 2003