UC Berkeley
Research Reports

Title
Compression Braking for Longitudinal Control of Commercial Heavy Vehicles

Permalink

Authors
Moklegaard, Lasse
Druzhinina, Maria
Stefanopoulou, Anna G.

Publication Date
2001-04-01

This paper uses Postscript Type 3 fonts. Although reading it on the screen is difficult it will print out just fine.

Compression Braking for Longitudinal Control of Commercial Heavy Vehicles

Lasse Moklegaard, Maria Druzhinina
Anna G. Stefanopoulou

CALIFORNIA PARTNERS FOR ADVANCED TRANSIT AND HIGHWAYS

Compression braking for
Longitudinal Control of Commercial Heavy Vehicles

Lasse Moklegaard
Maria Druzhinina
Anna G. Stefanopoulou

Department of Mechanical and Environmental Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106-5070

MOU 372

PATH Research Report

December 2000

Acknowledgments

Compression Braking for
Longitudinal Control of Commercial Heavy Vehicles

Lasse Moklegaard, Maria Druzhinina and Anna G. Stefanopoulou
December 2000

Abstract

In this report, we develop a detailed crankangle based diesel engine model and a low order engine torque model for a Class-8 commercial heavy vehicle (CHV) as well as longitudinal control algorithms that coordinate the variable compression brake mechanism with service brakes and gear ratios. In particular, we have developed a detailed crankangle based simulation model for a six cylinder, 350 Hp diesel engine equipped with a compression brake. The model is based on energy conservation principles in addition to static engine maps provided by the manufacturers, and it is capable of describing the intrinsic interactions between individual cylinder intake and exhaust processes, and turbocharger dynamics during combustion and braking modes and the transition between those modes. Based on averaging and identification of the instantaneous torque response for changes in brake valve timing and fuel flow, we derive the low order control oriented engine model. This work bridges the gap between the detailed crankangle based models developed in the engine design community, and the low order representation of engine torque response used in the vehicle dynamics community. Moreover, a sensitivity analysis and an evaluation of the model variations across operating regimes allow us to assess the difficulties in the control design. The challenges in the control design are summarized as: mode switching, large parameter variations, control authority allocation, and saturation of redundant actuators.

The second part of this report deals with longitudinal speed control algorithms. Use of compression brake reduces the wear of the service brakes, and it is, thus, a preferred way of controlling the vehicle speed during a steady descent. Subsequently, we integrate the compression brake actuator with the service brakes and gear ratio adjustments to accomplish more aggressive braking maneuvers or control vehicle speed during large changes in the grade. We first design a linear PI controller that emulates the driver's actions to avoid "runaways" during descends on grades. The controller simply uses the engine speed measurement to activate the service brakes only when needed. To control effectively the large signal behavior of the system we then develop nonlinear controllers that accomplish braking maneuvers including in-traffic vehicle following scenario. The design technique is based on the Speed-Gradient approach, whereby control action is selected in the maximum descent direction of a scalar goal function. The nominal goal function is selected to address the speed regulation objective and, then, is appropriately modified by barrier functions to handle the aggressive braking maneuver requirements. Simulation results show that the variable compression brake allows smooth and fast speed regulation and rejection of torque disturbances due to changes in the road grade.

Keywords

Advanced Vehicle Control Systems	Speed Control
Brakes	Vehicle Dynamics
Commercial Vehicle Operations	

Executive Summary

In this report, we summarize our work on longitudinal control of commercial heavy vehicles (CHVs) equipped with variable compression braking mechanism. CHVs are an essential part of our nation's economy, and an efficient link between marine, railroad and air transportation nodes. Increased highway speed and transportation demands, coupled with limitations in traditional service brake actuators (friction pads on the wheels) create a challenging control problem that requires additional retarding actuators and coordination with service brakes. Although, service brakes can theoretically provide a retarding power ten times higher than the accelerating power of the vehicle, they cannot be used continuously because of the generated heat and associated wear of the friction contacts. The presence of delays associated with the pneumatic or the hydraulic actuation subsystem impose additional constraints on the longitudinal control of CHVs. Faced with these difficulties, fleet and engine manufacturers are introducing additional retarding mechanisms with low weight and maintenance requirement so they do not offset the recent improvements in powertrain efficiency. A very promising retarding mechanism that satisfies the above low maintenance and weight-to-power ratio requirements is the engine compression brake. This retarder enhances braking capability by altering the conventional gas exchange process in the cylinders of the engine and effectively converting the turbocharged diesel engine, that powers the CHV, into a compressor that absorbs kinetic energy from the crankshaft. During compression braking mode the engine dissipates the vehicle kinetic energy through the work done by the pistons to compress the air during the compression stroke. The compressed air is consequently released into the exhaust manifold through a secondary opening of the exhaust valve at the end of the compression stroke.

The research results contribute to modeling and control of HDVs. We first develop a detailed crankangle-based simulation model of a diesel engine equipped with a compression released engine brake. Secondly, by employing signal processing to the output of our simulator, we develop a linear reduced order model for use in control design. This work then bridges the gap between the detailed crankangle-based model developed in the engine design community, and the low order representation of engine torque response used in the vehicle dynamics community.

The detailed crankangle-based simulation model models a six cylinder, 350 Hp diesel engine equipped with a compression released engine brake. The model is based on quasi-steady flow through restrictions, "filling and emptying" plenum dynamics, and lumped parameter approximation of the rotational dynamics in addition to static engine maps provided by the manufacturers. The model is capable of describing the intrinsic interactions between individual cylinder intake and exhaust processes, and turbocharger dynamics during combustion and braking modes and the transition between those modes.

A family of linear reduced order (LRO) models are derived by applying input-output identification techniques on an event-based averaged signal of the instantaneous torque response produced by the detailed crankangle-based model at different operating conditions. The inputs to the linear reduced order model are fuel flow, brake valve timing and engine speed, while shaft torque is the output. It is worth mentioning here that most vehicle control problems use a simplified longitudinal model to develop the throttle (fuel) and braking control algorithms by considering only the dominant dynamics of the process [20], and [28]. This is largely achieved due to the clear separation between (i) the fast dynamics of the angular wheel velocity and the fueling system, (ii) the moderate dynamics of the intake manifold, engine speed, and turbocharger rotor, and finally, (iii) the slow dynamics of the vehicle velocity. We find that such an approximation for the engine torque response to

variable compression braking is also appropriate. Specifically, we find that approximately 70 percent of the variable compression braking torque due to a step change in brake valve timing is instantaneous, and the remaining 30 percent of torque output is adequately modeled as a first order lag.

In the second part of this report we concentrate on the longitudinal control problem using variable compression braking to its maximum extent in an effort to minimize the use of service brake and, hence, the service brake wear. It is well known [18, 12] that wear and overheating reduces the DC authority of the service brakes and introduces large parameter variations. Adaptive algorithms have been developed in [20] to address unpredictable changes in brake model parameters. Recent work [27] shows that non-smooth estimation and adaptation techniques can be used to achieve a reasonable brake friction force control. The delays associated with the pneumatic activation of service brakes impose one of the main obstacles in autonomous heavy vehicle following scenarios. These difficulties in autonomous HDVs are mitigated by using aggressive prediction algorithms [44]. The prediction algorithms, however, assume accurate knowledge of the delays and do not perform well during a totally uncertain brake maneuver. To reduce the application and intensity of the service brakes, compression brake can potentially be used as a sole decelerating actuator during low deceleration requests and combined with the service brakes during high deceleration requests.

Subsequently, we consider two types of braking maneuvers that are classified as critical and non-critical. The non-critical braking maneuvers reflect the requirement of maintaining the desired vehicle speed during the long descent down a (possibly varying) grade without critical driving objectives on time of response (such as during collision avoidance). During non-critical maneuvers the compression brake can be used as the only decelerating actuator, possibly combined with gear ratio adjustments to handle large grade changes. On the other hand, critical braking maneuvers require aggressive braking action with both compression and service brakes, where service brake is used to supplement the compression braking capability.

Initially, a linear controller is designed for the linear reduced order (LRO) model. Specifically, we design a high priority PI-controller for the compression brake, combined with a P-controller for the service brakes. The priority of the controllers is scheduled based on the saturation of the compression brake. Nonlinear simulations are used to assess the closed-loop performance. The results show that the variable compression brake allows smooth and fast speed regulation and rejection of torque disturbances due to changes in the road grade. To quantify the benefits of the developed semi-autonomous, coordinated braking system compared to manually using the service brakes alone, we establish two measures: one that is based on settling time and addresses safety, and one that is related to the use of service brakes and addresses maintenance cost. We show that for one specific critical braking maneuver the coordination of the brakes reduces the use of the service brakes by a factor of 45, and that it reduces the settling time by a factor of 2.

The above developments are concerned with controller design based on the linear system model. In reality, the vehicle model is nonlinear. Moreover, a sensitivity analysis and an evaluation of the model variations and uncertainties across operating regimes allow us to show that there is a significant sensitivity to changes in the gear ratios and payload. To deal effectively with the nonlinearities we investigate the application of nonlinear control. The controllers are designed using the Speed-Gradient (SG) methodology [10, 11]. This is a general technique for controlling nonlinear systems through an appropriate selection and minimization of a goal function. The nominal goal function is selected to address the speed regulation objective. Motivated by robotics obstacle avoidance applications [13],

barrier functions are then added to the nominal goal function to handle critical braking requirements. The controller is designed to provide the decrease of the goal function along the trajectories of the system. The local closed-loop stability is verified analytically by checking the achievability condition. It is shown that the controller is guaranteed to have a large region of attraction covering a very reasonable interval of initial values for the vehicle speed. Two approaches to compensate for the uncertainty in the road grade are investigated. One approach is based on the integral action of the Speed-Gradient Proportional-plus-Integral (SG-PI) controller. This controller can be used to compensate for unknown constant (or slowly varying) grade and other uncertainties, including the uncertainty in the aerodynamic coefficient. An alternative approach is based on the derivative action of the Speed-Gradient Proportional-plus-Derivative (SG-PD) controller, that can compensate for fast varying grade essentially by estimating the torque loss due to the unknown grade. These approaches are explored in detail and the controller responses are evaluated using simulation. The above controllers are augmented to address the coordination issues between the compression brake, service brake and the gear ratio adjustment. Specifically, to handle large changes in the grade during non-critical maneuvers, a scheme that coordinates the compression brake with the gear ratio adjustment is described. To handle critical maneuvers another scheme that coordinates the compression brake with the service brake, is also developed.

Contents

List of Figures

1 Introduction

Commercial heavy vehicles ($CHVs$) are an essential part of our nation's economy, and an efficient link between marine, railroad and air transportation nodes. In 1996, the intercity trucking industry accounted for $176.8 billion in revenues and over 2.9 million jobs [24].

Over the last ten years, there has been a significant improvement of the reliability and efficiency of the CHV powertrain. This transformation is primarily achieved by using lightweight material, and by reducing the aerodynamic drag and frictional losses that are the main natural retarding sources for CHVs. From this it is evident that as fuel efficiency increases, the natural braking capability of the CHV decreases. In addition to natural limitations of the braking power, the air brake system on a tractor-semitrailer vehicle introduces a considerable delay during the actuation phase. Also, due to the large mass of a CHV, the air brakes are already operating close to the pressure limit. This can potentially lead to problems during combined longitudinal and lateral maneuvers [26].

In Automated Highway Systems (AHS), the major goals are to increase highway capacity and to enhance driving safety by automatic longitudinal and lateral control of vehicles [37]. The long term objectives of this project are, therefore, (A) to increase retarding power to accommodate higher operating speed, (B) to lower maintenance cost on service brakes by coordinating service and engine brake, and (C) to enhance safety during critical lateral maneuvers, by coordinating vehicle retarders with the steering mechanism. In this work we address the first two objectives. In particular, we develop the models and longitudinal control algorithms using variable compression braking to its maximum extent in an effort to minimize the use of service brake and, hence, the service brake wear.

The report is organized as follows. In Section 2 we review the work that has been done on related research, while we in Section 3 describe the compression braking operating principles as well as two other potential vehicle retarders. A longitudinal vehicle model is presented in Section 4. In Subsections 4.4-4.5 we analyze the stability of the equilibrium descending speed and compare qualitatively the possible equilibria for the experimental brake and the 1966 engine brake reported in [4]. A detailed description of the crankangle-based model and validation results are presented in Section 5. The developed model can be used as an engine simulator; it is not easy to use in control design due its complexity. By employing averaging and system identification techniques to the output of this model, as described in Subsection 5.10, we derive a family of linear reduced order (LRO) engine models in Subsection 5.11. A linear PI controller that coordinates the compression brake with the service brakes and the analysis of the linear reduced order (LRO) system, followed by simulation of five driving scenario tests, are presented in Section 6. Comparison remarks regarding the closed-loop performance of the nonlinear full order (NFO) and LRO models are included in Subsection 6.6. A sensitivity analysis in Subsection 6.7 addresses the necessary development of nonlinear and adaptive control algorithms. In Section 7 we develop nonlinear controllers for longitudinal speed control maneuvers. In particular, in Subsection 7.1 we apply Speed-Gradient methodology (reviewed in Appendix 9.3 for the development of a SG-PI controller, followed by the development of a SG-PD controller. The scheme that coordinates the compression brake with the gear ratio

1

adjustments is described at the end of the same Section. In Subsection 7.2 we address critical maneuvers. In particular, aggressive braking and "vehicle-following" are considered. To perform this kind of aggressive longitudinal maneuvers coordination of service brakes is used to supplement the compression braking capability. The closed-loop performance for all driving scenarios is demonstrated through simulations and concluding remarks are made in Section 8.

2 Background

A critical safety issue for CHVs is the vehicle retarding capability. The turbocharged, compression ignition internal combustion (diesel) engines is the preferred powerplant for CHVs. Unlike gasoline engines, diesel engines operate unthrottled, hence, the pistons do not have to work against intake manifold vacuum during the intake stroke. This, combined with very lean mixtures of fuel and air, contributes to the increased fuel economy and to the decreased natural retarding power. The need for higher brake power is addressed by truck manufacturers and fleet managers by the development of various retarding mechanisms in addition to the service brakes (drum or disc brakes on the vehicle wheel). The main categories of these retarders are engine brakes and shaft brakes. In the first category, compression brakes enhance braking efficiency by modifying the conventional gas exchange process. In the second category, shaft retarders are devices attached to the transmission, driveline, or axle that use high turbulence or electro-magnetic forces to dissipate the rotational energy of the moving parts.

None of these decelerating actuators support fully automatic operation and depend significantly on well trained drivers. It is important to note here that none of these retarding actuators, typically used in Class 7-8 vehicles, have been integrated into a unified automatic retarding system. Integrated schemes of an electric driveline retarding mechanism with advanced braking systems (ABS) and advanced traction systems (ATS) are only found in hybrid-electric experimental vehicles [5].

All engine manufacturers are striving for variable compression brake effort. Work can be summarized by Jacobs variable brake valve timing [17], Cummins' discrete cylinder brake valve actuator [22], and Volvo's variable exhaust throttle actuator [2].

Related work on longitudinal control of CHVs is shown in [43],[44], [45], and [47], while initial results on the coordination of steering and braking is shown in [3]. In [45], control-oriented simulation models of a heavy-duty vehicle, including diesel engine dynamics complete with turbocharger, intercooler, and automatic transmission with torque converter are developed. Using these models, several longitudinal controllers for automated CHVs that can operate reliably and yield good performance, both with and without intervehicle communication, are developed in [44]. One of the most critical obstacles in automated operation of CHVs is the presence of significant delays in the fuel and brake actuators. These delays are particularly important in longitudinal control of platoons without intervehicle communication. The effect of the actuator delays cumulate as they propagate downstream, and the result is performance degradation. In [43], and [47], backstepping-based nonlinear controllers and a PID-based nonlinear controller are developed. These controllers recover, at the expense of higher controller complexity, the original performance of the system without the actuator delays.

3 Auxiliary Retarding Actuators

Although, we will concentrate on the compression brake mechanism in this report, we briefly discuss two other commonly used retarding actuators.

3.1 Driveline Retarder

Hydraulic driveline retarders are based on the principal of hydraulic coupling and resistance between a rotor attached to the crankshaft or driveline of the vehicle, and the fins of a stator. Caterpillar and Voith developed manual and automated mechanisms that allow modulated or full braking power, respectively. These mechanisms enable smooth and somewhat controlled braking effort at non-zero rotational speed. Allison driveline retarders apply braking power directly to the driveline and are, therefore, used for stop-and-go driving conditions typical for buses in intercity operation.

Electric driveline retarders by Jacobs and Rockwell International, on the other hand, are typically based on dissipating kinetic energy from the driveline by means of setting up a magnetic field. These mechanisms are currently the only ones that allow integration with ABS and ATS.

3.2 Exhaust Brake

Exhaust braking is based on an add-on device that restricts the flow out of the exhaust manifold. The result is an increase in the exhaust pressure, which in turn decreases the output power of the engine. The exhaust brake is easy to install, but does not provide high retarding capabilities. Jacobs and Pacific Diesel Brake manufacture this device.

Figure 1: **Left:** Driveline retarder (Source: [23]). **Right:** Exhaust brake (Source: [1]).

3.3 Compression Brake

A very promising retarding mechanism is the engine compression brake. This retarder enhances braking capability by altering the conventional gas exchange process

in the cylinders of the engine and effectively converting the turbocharged diesel engine, that powers the CHV, into a compressor that absorbs kinetic energy from the crankshaft. During the compression braking the fuel injection and combustion are inhibited. Through the work done by the pistons, using the crankshaft kinetic energy, the air in the cylinder is compressed in the compression stroke. At the end of the compression stroke, close to the time when fuel injection usually takes place, the exhaust valve opens dissipating the energy stored in the compressed air into the exhaust manifold. We call the secondary opening of the exhaust valve when the air is released into the exhaust as Brake Valve Opening (BVO) (or braking event), and we refer to the corresponding timing of the exhaust valve opening as Brake Valve Timing (or BVO timing), v_{eb}. Due to geometric constraints, the exhaust valve lift profile is considerably different for the exhaust and brake events, as shown in Figure 2. For simplicity, we call the braking-event profile of the exhaust valve the "brake
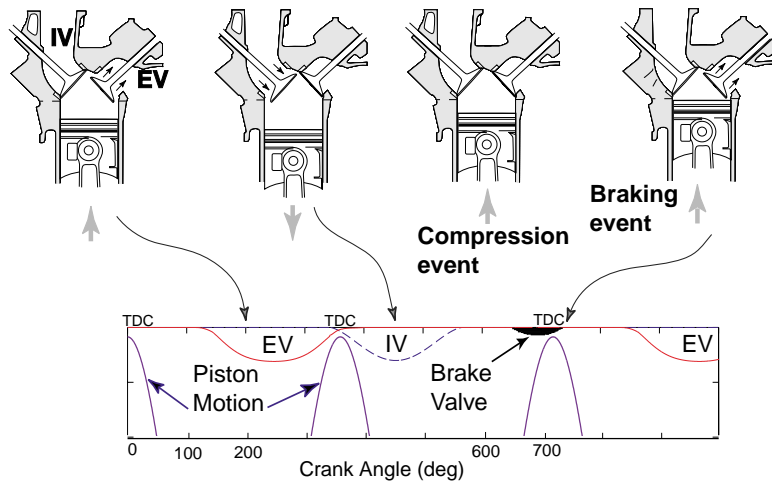
Figure 2: Schematic drawing and traces of piston motion and valve lift profiles.

valve". The activation of the brake valve is typically achieved through a master-slave hydraulic system. The exact profile and timing for the brake valve are designed to maximize the braking power, i.e., to generate the highest peak cylinder pressure. The brake valve profile needs to satisfy constraints on component loading due to high in-cylinder pressure, and geometric clearance between the brake valve trajectory and the piston motion. It is this type of actuator that we focus on in this project.

Figure 3 illustrates the valve timing events plotted in a $\mathbf{PV}$[1] diagram. This diagram shows the engine operation during conventional 4-stroke cycle operation (dotted line) and the 4-stroke cycle operation during compression braking (solid line). It is generated using outputs from the model developed in Section 5, for steady-state engine conditions. Note here that steady-state conditions are defined in a cycle-averaged sense. The engine is considered at steady-state even though its crank angle based behavior is periodic with one firing or braking cycle as the period. In the $\mathbf{PV}$ diagram, the piston work is positive during combustion and negative during braking. Thus, the compression brake is a retarding mechanism that is achieved by inhibiting fuel injection and the combustion event, and, thereby, transforming the engine into

---

[1]$\mathbf{PV}$ diagram is the plot of cylinder pressure vs. cylinder volume. It is used in the engine design and thermodynamic communities extensively, since the line integral represents work: $W = \oint p_{cyl} dV_{cyl}$

a compressor. In that mode, the engine acts like an energy sink [4], because the crankshaft kinetic energy is used to compress the air during the compression stroke. Close to top-dead-center (TDC) the exhaust valve opens (BVO) and the compressed air is released to the exhaust manifold. Note here that in the absence of BVO all the potential energy stored in the compressed gas will return to the wheels by the downwards piston motion. With the brake event the kinetic energy absorbed during the compression stroke is dissipated as heat in the exhaust manifold. The existing engine cooling system typically manages the dissipation of the exhaust manifold heat and no additional cooling subsystems are introduced to the engine. It is important to
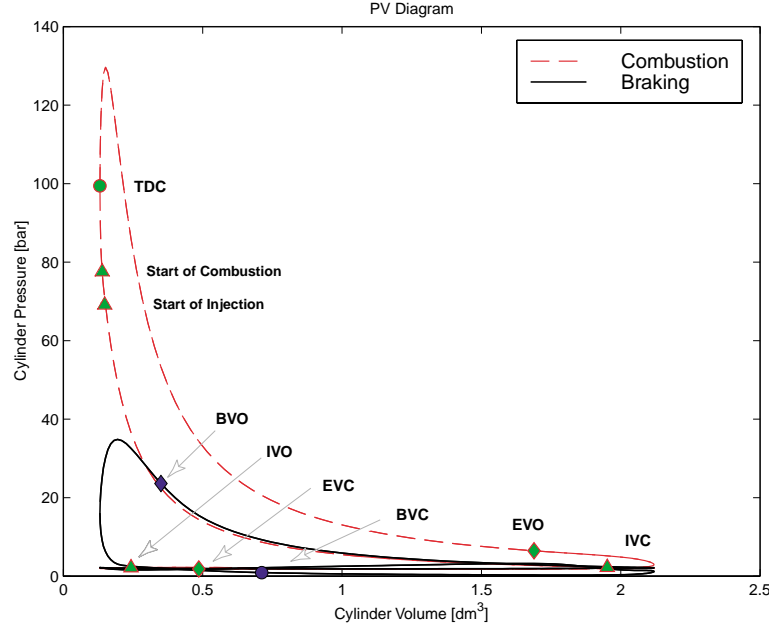
Figure 3: Cylinder pressure versus cylinder volume during combustion (dashed), and braking mode (solid).

note that the engine temperature during braking is not considerably lower than the one during combustion. As a mater of fact, the overheating of the injectors during braking at long descends is of concern to many engineers. The high temperature in the cylinders can damage the injectors that do not enjoy the benefitial cooling effect of fuel injection during the braking periods.

The concept of compression braking was introduced by Cummins in 1966, and typically, depends on an add-on device that opens the exhaust valve at fixed degrees with respect to the piston motion. Considerable effort is dedicated to optimize the fixed valve timing to achieve maximum retarding power for all engine speed, load and environmental conditions. The fixed brake valve timing mechanism is an on-off device and produces a fixed brake torque for a given engine speed. However, for applications in Intelligent Transportation Systems (ITS) full control over the brake valve timing is desirable. A continuously varying brake valve timing allows smooth changes in the compression brake torque response while maintaining constant engine speed. This variability can be used in various CHV applications such as speed regulation, brake-by-wire systems, cruise control, and finally, vehicle-following maneuvers. Moreover, full integration of the compression brake with the service brakes (drum or disc brakes

6

on the vehicle wheel rim) can be achieved. Due to the potential benefits, many engine manufacturers are striving for variable compression braking effort. Work is summarized by Jacobs variable brake valve timing [17], Cummins' discrete cylinder brake valve actuator [22], and Volvo's variable compression braking with exhaust throttle actuator [2].

We define the Brake Valve Timing v_{eb} as the number of crank angle degrees from when the piston is at top-dead-center at the beginning of the intake stroke to the opening of the brake valve at the end of the compression stroke. The following figure, Figure 4 shows compression braking power for two different brake valve timings.
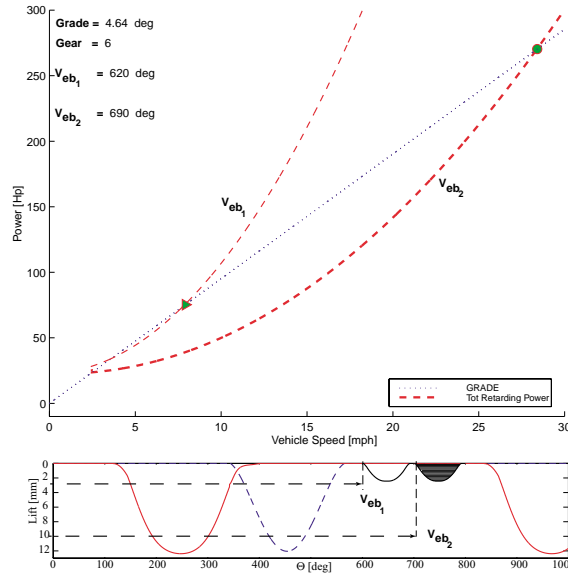
Figure 4: Variable Engine Retarder.

Control of the brake valve timing allow us to continuously vary the retarding power of the compression brake mechanism. The left plot in Figure 5 shows the fixed activation of the brake valve events based on the mechanical link between the crankshaft and the camshaft. Here, the valve timing is completely determined by the engine speed. The right plot in Figure 5, on the other hand, shows a system where the valves are activated by electro-mechanical or electro-hydraulic actuators. The mechanical connections between the valve profile and the crankshaft are replaced by electronics, hence continuous variable valve timing is possible.
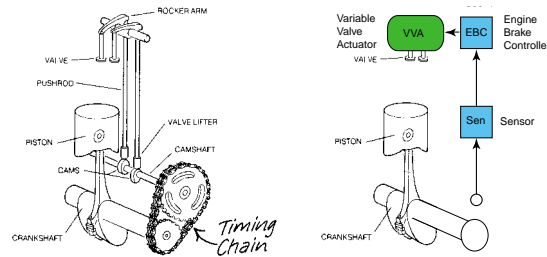
Figure 5: **Left:** Conventional valve lift system for fixed valve timing (Source: [33]). **Right:** Valve lift system that enables variable valve timing.

4 Longitudinal Vehicle Model

Consider the vehicle operation during a driving maneuver on a descending grade with β degrees inclination ($\beta = 0$ corresponds to no inclination, $\beta < 0$ corresponds to a descending grade). It is assumed that during the descent, the engine is not fueled and is operated in the compression braking mode.

4.1 Vehicle Dynamics Model

A lumped parameter model approximation is used to describe the vehicle longitudinal dynamics during compression braking. For fixed gear operation the engine crankshaft rotational speed, ω, is expressed by:

$$J_t \dot{\omega} = TQ + r_g(F_\beta - F_r - F_{qdr} + F_{sb}),\tag{1}$$

where, ω is the engine rotational speed, (rad/sec), related to the vehicle speed value, v, (m/sec), by the following relation

$$v = \omega r_g \tag{2}$$

$r_g = \frac{r_\omega}{g_t g_{fd}}$ is the total gear ratio, where r_ω is the wheel diameter, g_t is the transmission gear ratio, g_{fd} is the final drive gear ratio

$J_t = mr_g^2 + J_e$ is the total vehicle inertia reflected to the engine shaft (depends on the vehicle loading conditions), where J_e is the engine crankshaft inertia

m is the mass of the vehicle (depends on the mass of payload),(kg)

F_{qdr} is the quadratic resistive force (primarily, force due to aerodynamic resistance, but we also include friction resistive terms):

$$F_{qdr} = C_q v^2 = C_q r_g^2 \omega^2$$

$C_q = \frac{C_d A \rho}{2} + C_f$ is the quadratic resistive coefficient, where C_d is the aerodynamic drag coefficient, ρ is ambient air-density, A is the frontal area of the vehicle, C_f is the friction coefficient

$F_\beta(m, \beta)$ is the gravitational force due to grade (β): $F_\beta = -mg \sin \beta$

$F_r(m, \beta, \mu)$ is the force due to rolling resistance of the road(μ): $F_r = \mu g m \cos \beta$

g is the acceleration due to gravity

$TQ = f_1(\omega, v_{eb}, v_f)$ is the engine shaft torque applied to the crankshaft (negative during compression braking)

$F_{sb} = f_2(v_{sb})$ is the force on the vehicle due to application of the conventional service/friction brake (applied to the wheel rim, negative during friction braking).

The dynamical equations describing F_{sb} and TQ are discussed in detail below.

4.2 Service Brakes

The conventional service brake force on the wheel is modeled using a static nonlinear function, a first order differential equation with a time constant τ_{sb}, and a delay t_{sb}:

$$\begin{aligned} F_{sb} &= f_{sb}(v_{sb}), \\ \tfrac{d}{dt}v_{sb}(t + t_{sb}) &= \tfrac{1}{\tau_{sb}}(u_{sb} - v_{sb}(t + t_{sb})). \end{aligned}\tag{3}$$
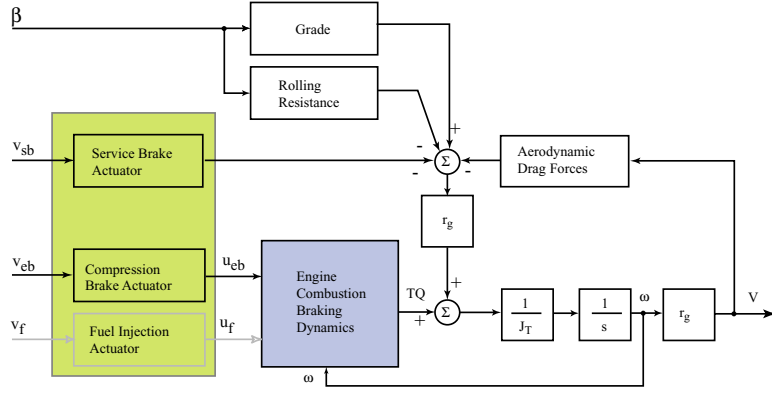
Figure 6: Block diagram of the vehicle model.

The static nonlinear function is a function of the applied pedal force, or simpler, the pedal displacement u_{sb} (for details see [12],[32],[47]). The time constant and time delay τ_{sb} and t_{sb}, respectively, are nonlinear and uncertain functions of temperature and brake conditions.

The service brakes can provide a sufficient retarding power to decelerate the vehicle to a desired speed. They cannot, however, be used continuously to maintain the desired speed because of the potential damage/loss of performance due to overheating and increased wear [12]. The current practice of "snubbing" the service brake (application of high pressure pulses) rather than "dragging" (application of a constant low pressure) exemplifies their low DC authority [9]. The presence of delays associated with the pneumatic or the hydraulic subsystem in the service/friction brake actuators imposes additional difficulties in using service brakes for the longitudinal control of CHVs [47]. To maintain operational speeds comparable to passenger vehicles, without compromising safe braking performance, high retarding power with consistent magnitude and unlimited duration is required. Thus, augmenting the braking performance of CHVs with auxiliary retarding mechanisms is increasingly important in order to integrate CHVs into the advanced transit and highway systems.

4.3   Compression Brake

The additional retarding power from the compression brake is used during severe deceleration commands or during long descends. The compression brake can be used continuously due to the heat dissipation through the engine cooling system. The compression brake is typically used in on-off fashion but there is no prior work done on identifying DC components or on identifying the dynamics for this actuator.

Identifying the speed of response of the compression brake for different engine and drivetrain operating conditions, requires an accurate yet simple model of the cylinder-to-cylinder gas exchange process and combustion. During compression, the working fluid is air, therefore, thermodynamic models are generally accurate. In the following Section 5, we document the mathematical model that describes the engine operation during combustion mode (dashed line in Figure 3) and during braking mode (solid line). We consider the engine response to changes in crankshaft speed and brake valve timing. Analysis of the braking effort during valve timing changes sheds light into

a variable retarding mechanism that can be potentially integrated with the service brakes. The inputs to the model are: (i) fuel flow, (ii) engine speed, and (iii) brake valve opening (fixed duration and profile). The model is capable of representing transitions from combustion to braking by cutting-off the fuel injection and initiating the brake event. The crank angle representation allows us to capture the cylinder-to-cylinder interactions and in-cylinder pressures that dictate the dynamic phenomena during the transition from combustion to braking. Based on the crank angle based model, we derive a mean-value representation of the steady-state and the dynamic characteristics of the engine braking mechanism for control design and integration with the vehicle dynamics and service brakes. Further control development will be tested on the crank angle based model.

4.4   Equilibrium Analysis

In the preceding subsection, we developed a longitudinal vehicle model based on summation of torque on the engine shaft, given by Newton's laws. In the transportation community, however, the general practice is to use power rather than torque in equations and figures describing vehicle dynamics. Using the well-known relation between power and torque $P = TQ\omega$, where ω is engine speed and P is power, the longitudinal vehicle model is given by the following expression:

$$\frac{d\omega}{dt} = \frac{1}{J_T\omega}\left(P_e(\omega, v_{eb}, v_f) + P_{sb}(v_{sb}) - P_{qdr}(\omega) + P_\beta(\beta) - P_r(\beta)\right). \tag{4}$$

At equilibrium ω^* the rate of change in engine speed is zero, and the power generated by the gravitational force during a grade descend is equal to the power dissipated by the engine retarding mechanisms:

$$P_\beta = -P_e - P_{sb} - P_0, \tag{5}$$

where P_β is the gravitational power generated when the vehicle descends on a grade β, and P_e is the power *generated* by the engine if it is in combustion mode, otherwise it is the power *dissipated* by the engine if it is in braking mode. The power P_e dissipated by the compression brake is derived in Section 5.11, based on TQ calculated in (36). P_{sb} is the power dissipated by the service brakes (supposed to be negative), while $P_0 = -(P_{qdr} + P_r)$, is the natural braking power due to aerodynamic and rolling resistance.

Figure 7 shows a diagram of power versus engine speed that is used extensively in the trucking industry to find equilibrium descending engine speed ω^*, given a descending grade and the engine compression braking power. The equilibrium descending engine speed ω^*, is defined at the intersection of the retarding vehicle power $-P_e - P_0$, with the power due to gravity P_β. This figure is informative because it depicts the allowable range of engine speed; it can be used for sizing the engine brake actuator based on a range of expected grades and vehicle loading conditions.

On the other hand, Figure 7 does not provide any information about the system transient behavior. The brake valve timing affects both the maximum braking effort and the speed at which this maximum effort is achieved. The delays and time constants of the brake actuators, coupled with the dynamic response of the turbocharged
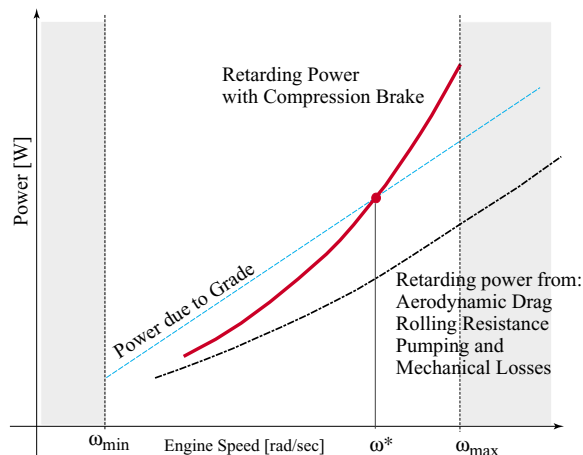
Figure 7: Vehicle speed versus braking power during a steep descent.

engine, can impose severe limitations on the stability of the vehicle velocity. Nowadays, these limitations are primarily treated by the predictive action of trained drivers. Identifying the speed of response of the compression brake for different engine and drivetrain operating conditions is important for the design of safer and faster heavy commercial vehicles.

4.5 Past and Present set of Equilibrium Points

It is of interest to see how the compression brake technology has evolved since 1966, when it first was introduced to the transportation industry by Cummins [4]. Figure 8 shows how the equilibrium points ω_{66}, and ω_{99}, that define the engine descending speed, have changed from 1966 to 1999. The data for the curve denoted "Compression Brake, 1966," is adapted from [4], while the data for the curve denoted "Compression Brake, 1999," is calculated using (36). First of all, the current descending engine speed is higher than the one attained in 1966, $\omega_{99} > \omega_{66}$. It is important to note that ω_{99} is a *stable* equilibrium point, while ω_{66} is *unstable*. The reasons for this change in stability are two fold: for low speed, there has been a big improvement in the reduction of mechanical losses, such as friction, pumping losses and aerodynamic drag. This improvement is due to an increasing demand for better fuel economy and reduction in emissions. For high speed, on the other hand, the increase in braking power is most likely due to improvements in the actuator components. Old technology engine braking was based on low pressure hydraulic actuators for the activation of the engine brake valve. At high engine speed, the hydraulic actuator could not support the fast opening and closing of valve. Nowadays, high pressure hydraulic systems allow consistent opening of the exhaust valve timing for all engine speed. The combination of these improvements result in a higher, and stable equilibrium ω_{99}. Note here that safe operation at higher equilibrium descending speed is preferred due to a potential increase of average vehicle speed and, consequently, a decrease in CHV trip duration.

11
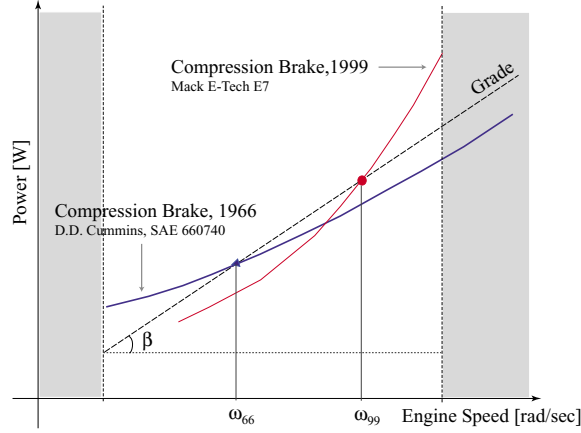
Figure 8: Figure show how the equilibrium point has changed from ω_{66} in 1966 to ω_{99} in 1999.

5 Crankangle-Based Engine Model

To analyze and to quantify the effects of the brake valve opening v_{eb}, we develop a physics-based nonlinear engine model for a six cylinder, 350 Hp diesel engine. Our modeling approach (see [30], [29]) is based on work by Watson et. al. ([41]), ([42]), and work by Kao et. al. ([25]). The model is capable of describing the intrinsic interactions between individual cylinder intake and exhaust processes, and turbocharger dynamics during combustion and braking modes and the transition between those modes.The nonlinear dynamic engine model consists of static elements (described by empirical equations), and dynamic elements (described by physically based equations). The parameters of the nonlinear static relations are determined by regression analysis of data collected from an experimental engine equipped with an engine braking mechanism. The simulation model has been implemented in a Simulink/Matlab environment using C-coded S-functions (see Appendix 9.4 for details).

5.1 Notation

Notation can be summarized as follows (see Appendix 9.1 for details): masses are denoted by m, pressures p, temperatures T, flows W, efficiencies η, powers P, and volumes V. We use the subscript c for compressor, t for turbine, i for the intake manifold, and e for the exhaust manifold. For cylinder j, we use cyl_j, where $j = 1 \ldots 6$. The flow from control volume x into control volume y is denoted by W_{xy}. And we use N_e to denote engine speed in RPM, and ω to denote engine speed in rad/sec.

5.2 State Equations, Plenums

The intake and exhaust manifolds, and the cylinders are modeled as plenums with homogeneous pressure and temperature distributions. The plenum model is described by mass and pressure states that are based on the principle of conservation of mass and energy, and the assumption of the ideal gas law. The schematic of the engine and the definition of the plenums are shown in Figure 9.
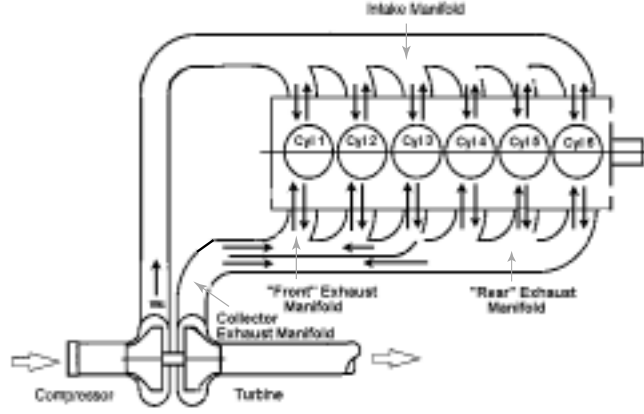
Figure 9: Schematic drawing of the engine and its control volumes.

The equations for the gas filling dynamics for the intake manifold are:

$$\frac{dm_i}{dt} = W_c - \sum_{j=1}^{6} W_{icyl_j} + \sum_{j=1}^{6} W_{cyl_j i}, \tag{6}$$

$$\frac{dp_i}{dt} = \frac{R}{V_i}\left[T_c W_c - T_i \sum_{j=1}^{6} W_{icyl_j} + T_{cyl_j} \sum_{j=1}^{6} W_{cyl_j i} \right], \tag{7}$$

where m_i is the intake manifold (IM) mass air charge, and W_c and T_c are the compressor mass flow and temperature, respectively (see Section 5.5). W_{icyl_j} is the mass air flow from the IM into cylinder j, and $W_{cyl_j i}$ is the backflow from the cylinder into the IM (see Section 5.4). The intake manifold pressure is denoted p_i, while the IM temperature is denoted by T_i. T_{cyl_j} is the temperature in cylinder j, and it is calculated based on the ideal gas law, and the mass and pressure states in the intake manifold.

As shown in Figure 9, the exhaust manifold (EM) is divide into a front, a rear and a collector manifold. The front and the rear manifolds are governed by the same set of state equations. So for simplicity, we only give the state equations for the front exhaust manifold dynamics, given by the rate of change in mass m_{e_f} and pressure p_{e_f}:

$$\frac{dm_{e_f}}{dt} = \sum_{j=1}^{3} W_{cyl_j e_f} - \sum_{j=1}^{3} W_{e_f cyl_j} - W_t, \tag{8}$$

$$\frac{dp_{e_f}}{dt} = \frac{R}{V_{e_f}}\left[\sum_{j=1}^{3} T_{cyl_j} W_{cyl_j e_f} - \sum_{j=1}^{3} T_{e_f} W_{e_f cyl_j} - T_{e_f} W_{e_f e_c} \right], \tag{9}$$

where $W_{cyl_j e_f}$ is the mass air flow out of cylinder j into the front EM, and $W_{e_f cyl_j}$ is the backflow from the front EM into the cylinder. The equations for mass air flow are based on flow through a restriction model that is described in Section 5.3. W_t is the turbine air flow described in Section 5.5, while V_{e_f} and T_{e_f} are the front EM volume and temperature, respectively. Similarly, the state equations for the collector

13

EM are given by:

$$\frac{dm_{e_c}}{dt} = W_{e_f e_c} - W_{e_c e_f} + W_{e_r e_c} - W_{e_c e_r} - W_t, \tag{10}$$

$$\frac{dp_{e_c}}{dt} = \frac{R}{V_{e_c}} \left[T_{e_f} W_{e_f e_c} - T_{e_c} W_{e_c e_f} + T_{e_r} W_{e_r e_c} \right.$$
$$\left. - T_{e_c} W_{e_c e_r} - T_{e_c} W_t \right], \tag{11}$$

where $W_{e_f e_c}$ and $W_{e_c e_f}$ are the flow and backflow from the front EM into the collector EM, whereas $W_{e_r e_c}$ and $W_{e_c e_r}$ are the flow and backflow from the rear EM into the collector EM. V_{e_c} is the collector EM volume, whereas T_{e_r} and T_{e_c} are the temperature in the rear and the collector EM, respectively.

The state equations for the cylinder mass m_{cyl_j}, and pressure p_{cyl_j}, include the time varying cylinder volume V_{cyl_j}, and heat transfer and release descriptions:

$$\frac{dm_{cyl_j}}{dt} = W_{icyl_j} - W_{cyl_j i} - W_{cyl_j e} + W_{ecyl_j}, \tag{12}$$

$$\frac{dp_{cyl_j}}{dt} = \frac{\gamma}{V_{cyl_j}} \left(RT_i W_{icyl_j} - RT_{cyl_j} W_{cyl_j i} - RT_{cyl_j} W_{cyl_j e} \right.$$
$$\left. + RT_e W_{ecyl_j} - \dot{V}_{cyl_j} p_{cyl_j} \right) + \frac{\gamma - 1}{V_{cyl_j}} W_{afb_j} Q_{lhv}, \tag{13}$$

where V_{cyl_j} is volume for cylinder j, and γ is the ratio between the specific heat capacities. We have for simplicity coupled e_f and e_r into one variable e. We calculate the released heat based on the apparent fuel burn rate W_{afb}, and the lower heating value of light duty diesel fuel Q_{lhv}. In the calculation of the apparent fuel burn rate W_{afb}, we have coupled the heat transfer. Thus, the integral of the apparent fuel burn rate over a cycle is less than the averaged mass fuel injected.

$$W_{afb_j} Q_{lhv} = W_{fb_j} Q_{lhv} + Q_{ht}, \tag{14}$$

$$\int_0^{\Delta T} W_{afb_j} dt < W_f \Delta T, \tag{15}$$

where Q_{ht} is the heat transfer, and W_f is the engine cycle (ΔT) averaged fuel flow. The fuel burned rate W_{fb} is calculated using the Wiebe approximation of the heat released during the premixed and the diffusion periods of the combustion. This is described in detail in Section 5.6.

The cylinder volume V_{cyl_j} is a functions of the crank angle θ:

$$V_{cyl_j} = V_{cl} \left[1 + \frac{1}{2} (r_c - 1) \left(R_r + 1 - \cos \theta_j - \sqrt{R_r^2 - \sin^2 \theta_j} \right) \right], \tag{16}$$

$$\theta_j = \left(\frac{N_e}{60} 360 \cdot t + 120 \cdot j \right) \bmod 720^o. \tag{17}$$

Here, V_{cl} is the cylinder clearance volume, V_{cd} is the maximum cylinder displacement volume, and $r_c = \frac{V_{cd} + V_{cl}}{V_{cl}}$ is the compression ratio. The ratio between connecting rod and crank radius is denoted R_r, while the engine speed in RPM is denoted N_e. The engine has six cylinders, and therefore, a separation of 120 degrees between each cylinder, expressed by $120 \cdot j$.

5.3 Flow Through a Restriction

A quasi-steady model of flow through an orifice is used to derive the mass air flow through all exhaust restrictions and cylinder valves. The quasi-steady relation of the air flow through a restriction is based on the assumptions of one-dimensional, steady, compressible flow of an ideal gas [15] and [31]:

$$W = C_d A_v \Psi(p_d, p_u, T_u), \tag{18}$$

where W is the general mass air flow, C_d is the discharge coefficient, A_v is the flow area function for the valve, and $C_d A_v$ is the effective flow area for the valve. The term Ψ is the standard orifice flow function that depends on the downstream pressure and temperature, p_d and T_d, and upstream pressure and temperature, p_u and T_u:

$$\Psi(p_d, p_u, T_u) = \begin{cases} \frac{p_u}{\sqrt{RT_u}} \Psi_o\left(\frac{p_d}{p_u}\right) & \text{if } p_d \leq p_u \\ 0 & \text{if } p_d > p_u, \end{cases} \tag{19}$$

with

$$\Psi_o(x) = \begin{cases} \gamma^{\frac{1}{2}} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} & \text{if } x \leq c_r \\ x^{\frac{1}{\gamma}} \sqrt{\frac{2\gamma}{\gamma-1}\left(1 - x^{\frac{\gamma-1}{\gamma}}\right)} & \text{if } x > c_r, \end{cases} \tag{20}$$

where $c_r = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}}$ is the critical pressure ratio across the orifice.

5.4 Valve Flow

The mass air flow W_{icyl_j} from the intake manifold into cylinder j is calculated using (18)–(20):

$$W_{icyl_j} = f_{iv}\left(l_{iv}(\theta)\right) \cdot \Psi(p_{cyl_j}, p_i, T_i). \tag{21}$$

The effective flow area though the intake valve, $f_{iv} = C_d A_{iv}$, is calculated using crank angle data of the intake valve lift $l_{iv}(\theta)$. Both maps are given by the engine manufacturer.

Similarly, the flow through the exhaust valve is given by:

$$W_{cyl_j e} = f_{ev}\left(l_{ev}(\theta)\right) \cdot \Psi(p_e, p_{cyl_j}, T_{cyl_j}), \tag{22}$$

where the exhaust valve map $f_{ev} = C_d A_{ev}$ is a function of the exhaust valve lift map l_{ev}. Both maps are provided by the engine manufacturer.

The brake valve is technically speaking the same as the exhaust valve, the only difference is the valve lift profile. This means that the flow through the brake valve is given by (22), where $l_{ev}(\theta)$ is replaced by $l_{bv}(\theta)$. Again, this valve lift is provided by the engine manufacturer.
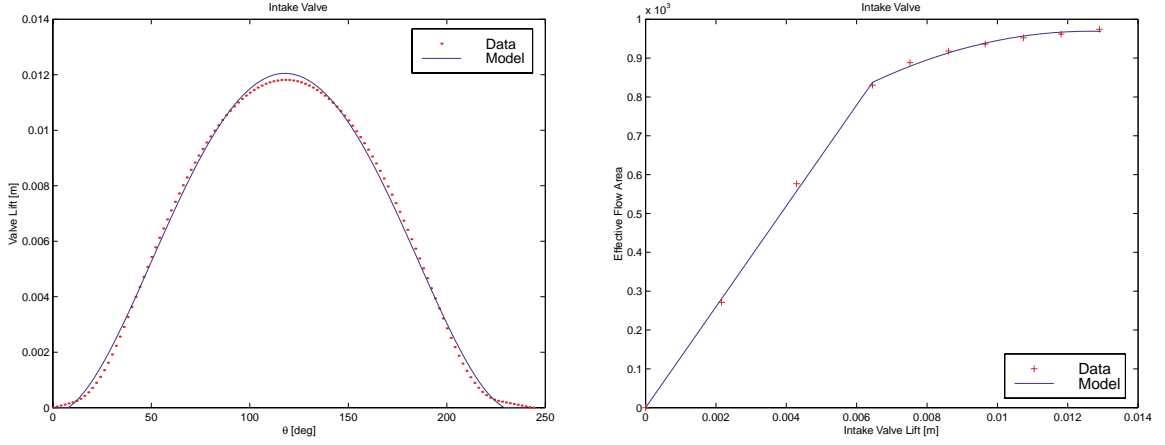
Figure 10: **Left:** Intake valve lift versus crank angle. **Right:** Intake valve effective flow area versus valve lift.
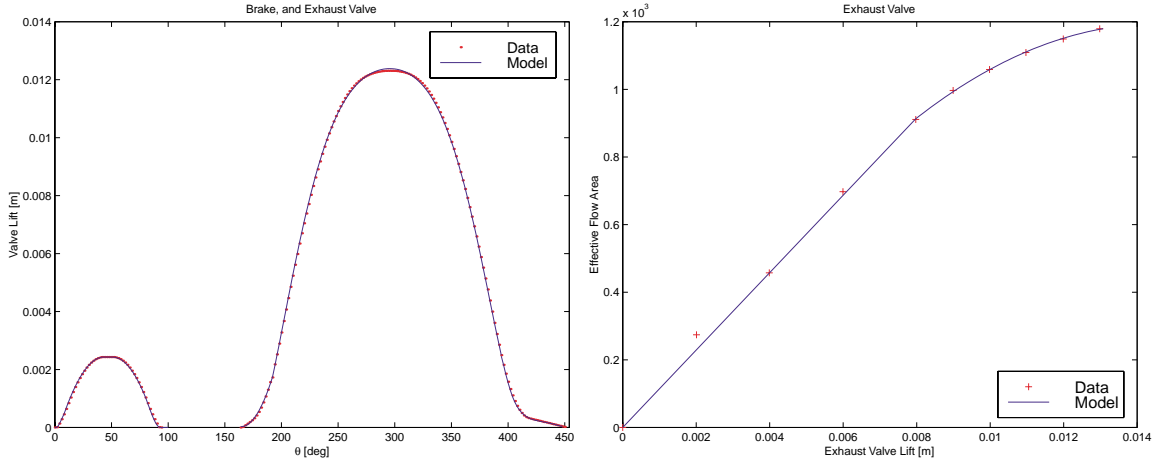
Figure 11: **Left:** Exhaust and brake valve lift versus crank angle. **Right:** The effective flow area for both the exhaust valve and the brake valve versus valve lift.

5.5   Turbocharger Dynamics

The turbocharger state equation is based on the conservation of energy on the turbocharger shaft, and consists of the rate of change of the turbocharger speed N_{tc}:

$$\frac{dN_{tc}}{dt} = \frac{P_t - P_c}{I_{tc}N_{tc}}, \tag{23}$$

where P_t is the turbine power, P_c is the compressor power, and I_{tc} is the mass polar moment of inertia of the turbocharger. P_t and P_c are calculated based on an ideal adiabatic process, and steady-state data provided by the turbocharger manufacturer.

In particular, turbine maps f_{t_w} and f_{t_η} are used to determine the mass air flow W_t, and the efficiency η_t. Both turbine maps are functions of the turbocharger speed

N_{tc} and the pressure ratio r_t across the turbine.

$$P_t = W_t c_p \eta_t T_2 \left(1 - \frac{1}{r_t}^{(\frac{\gamma-1}{\gamma})} \right), \tag{24}$$

$$W_t = f_{t_w} (N_{tc}, r_t), \tag{25}$$

$$\eta_t = f_{t_\eta} (N_{tc}, r_t), \tag{26}$$

$$r_t = \frac{p_2}{p_0},$$

where p_0 is the ambient pressure, and c_p is the specific heat capacity for constant pressure.
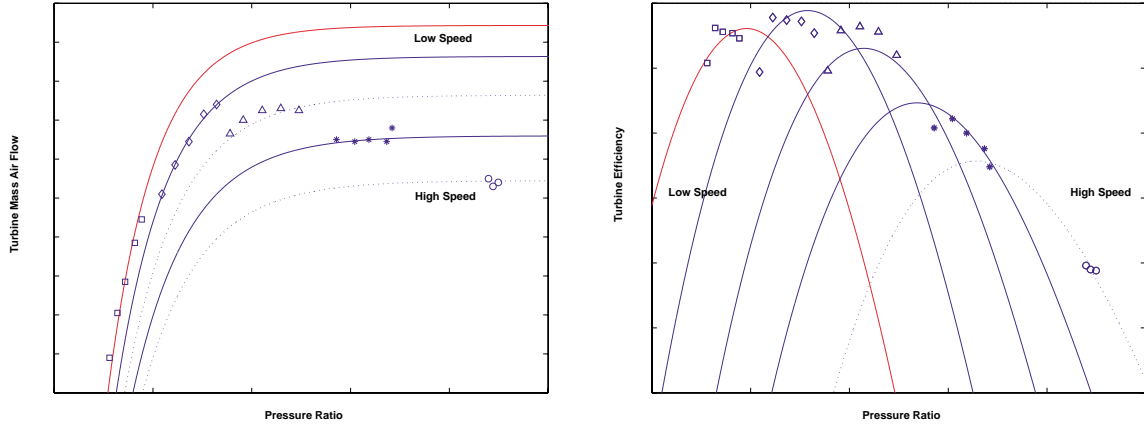
Figure 12: **Left:** Map for turbine mass flow W_c. **Right:** Map for turbine efficiency η_t.

Similarly, using data from the compressor maps we obtain the following compressor characteristics:

$$P_c = W_c T_0 \frac{c_p}{\eta_c} \left(r_c^{(\frac{\gamma-1}{\gamma})} - 1 \right), \tag{27}$$

$$W_c = f_{c_w} (N_{tc}, r_c), \tag{28}$$

$$\eta_c = f_{c_\eta} (N_{tc}, r_c), \tag{29}$$

$$r_c = \frac{p_1}{p_0},$$

where f_{c_w} is the compressor map for mass flow, and f_{c_η} is the map for the compressor efficiency η_c. Both of these maps are functions of the turbocharger speed N_{tc}, and the pressure ratio r_c across the compressor.

5.6   Apparent Fuel Burn Rate

The apparent fuel burn rate is identified based on cylinder pressure data and a modification of (13) for zero flow into and out of the cylinder:

$$W_{afb} = \frac{1}{Q_{lhv}(\gamma - 1)} \left(\gamma p_{cyl} \frac{dV_{cyl}}{dt} + V_{cyl} \frac{dp_{cyl}}{dt} \right). \tag{30}$$
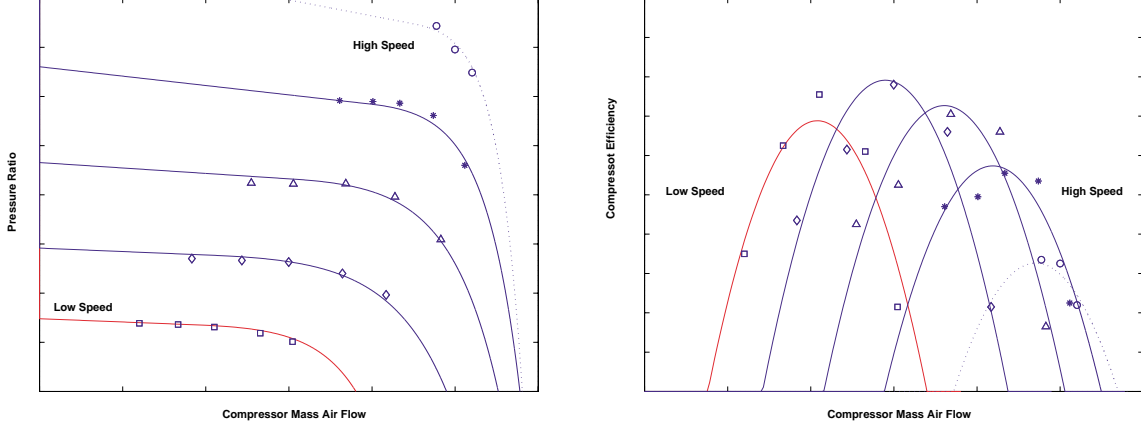
Figure 13: **Left:** Map for compressor mass air flow W_c. **Right:** Map for compressor efficiency η_t.

We assume here that heat capacities are constant and couple the heat transfer to the calculation of an effective or apparent fuel burn rate. We stress here that the apparent fuel burn rate we calculate in (30) is less than the actual one, since we for simplicity assume zero heat loss.

The apparent fuel burn rate curves for ten speed and load points are used to develop a crank angle functional approximation. Two Wiebe basis functions are combined to capture the premixed and the diffusion burning:

$$W_{afb} = max\left(W_{afb}^d, W_{afb}^p\right), \tag{31}$$

$$W_{afb}^d = C_d k_{d2}\left(k_{d1}+1\right)\theta_d^{k_{d1}}\exp\left(-k_{d2}\theta_d^{(k_{d1}+1)}\right), \tag{32}$$

$$W_{afb}^p = C_p k_{p1} k_{p2}\left(1-\theta_p^{k_{p1}}\right)^{(k_{p2}-1)}\theta_p^{(k_{p1}+1)}. \tag{33}$$

The variables $\theta_d = \frac{\theta-\theta_{soc}}{\Delta\theta_d}$ and $\theta_p = \frac{\theta-\theta_{soc}}{\Delta\theta_p}$, are defined over the diffused combustion duration $\Delta\theta_d$, and the premixed combustion duration $\Delta\theta_p$, respectively. To simplify the data fitting, we assume that both premixed and diffused-based combustion start at the same time; i.e. at start of combustion θ_{soc}. Using the max function in the combination of the two curves in (31) resolves the two distinct starts for premixed and diffused-based combustion, as shown in Figure 14.

The start of combustion is computed based on the start of injection θ_{soi}, and the ignition delay $\Delta\theta_o$:

$$\theta_{soc} = \theta_{soi} + \Delta\theta_o. \tag{34}$$

The six coefficients C's and k's, the two combustion mode durations $\Delta\theta$'s, and the ignition delay $\Delta\theta_o$, are identified and regressed on fuel injected, and engine speed, using the ten available data points. A good compromise between data over-fitting and small prediction errors is achieved with polynomials of second order in fuel flow and first order in engine speed for all variables.

We also regress the start of injection θ_{soi} based on fuel input and engine speed. Although, this does not give us the flexibility to use the injection timing as another model input, it allows us to maintain the simple combustion analysis in our model. The predictive ability of the model in fuel injection timing changes can be examined in future work.

18
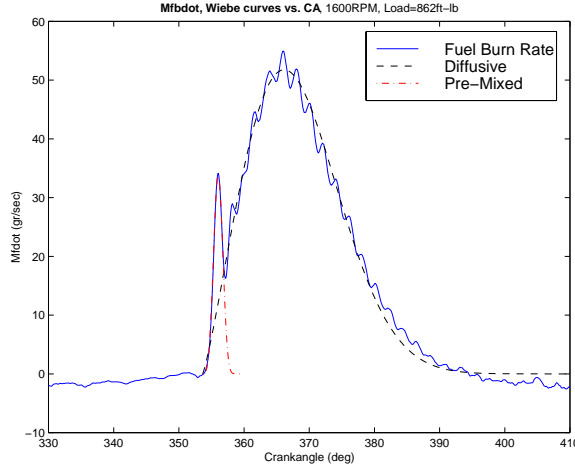
Figure 14: Combustion analysis and apparent fuel burn rate.

5.7 Shaft Torque

The torque produced at the crankshaft during combustion, or transmitted from the crankshaft during braking, TQ_{cyl_j}, is calculated based on the individual cylinder pressure and piston motion, using the idealized slider-crank mechanism [36]:

$$TQ_{cyl_j} = p_{cyl_j} \frac{\pi B^2}{4} r \sin(\theta_j) \left[1.0 + \frac{r}{R} \frac{\cos(\theta_j)}{\sqrt{1.0 - \left(\frac{r}{R}\sin(\theta_j)\right)^2}} \right], \tag{35}$$

where B is the cylinder bore, r the crank radius, and R the length of the connecting rod. The shaft torque is calculated by the summation of the individual cylinder torque, and the average torque from the piston to the shaft during combustion is positive, while the average torque from the shaft to the piston during braking is negative (see Section 5.10).

5.8 Engine Simulation Results

The engine model has been implemented in Matlab/Simulink using S-functions. It has 23 states with significantly different rates of convergence (stiff system), and multiple discontinuous functions that have been implemented with *if-else* statements. The stiffness and discontinuity are handled satisfactorily with the *ode23s* integration algorithm (one-step solver based on a modified Rosenbrock formula) within Matlab. Simulation of the engine model during a transition from combustion to braking is shown in Figure 15. The simulation is performed at constant engine speed, $N_e = 1600$ RPM, as the first subplot on the left indicates. The last plot in the right column shows the fuel flow command. One can clearly see the fuel cutoff at the fifth cycle. At that point the software is implemented to run with zero fuel (motoring) for one cycle before it opens the brake valve at $v_{eb} = 685^o$. This brake valve command is shown in the last plot in the left column of Figure 15. The model has been implemented with one cycle delay between the combustion and the braking mode to avoid extrapolation errors. It is shown that if we do not introduce this delay in the transition process the turbine efficiency drops below 0.2 which is outside the model region of validity. The delay
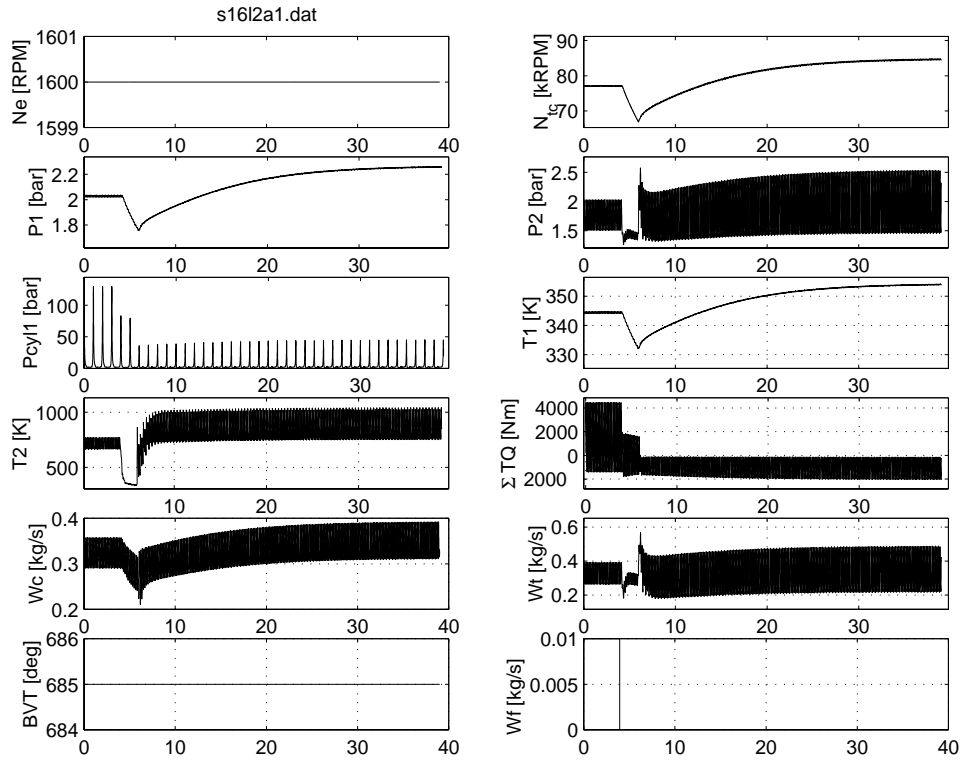
Figure 15: Simulation response during transition from combustion to braking.

can be viewed by observing the trace of cylinder pressure in the third plot of the left column.

The traces of intake and exhaust manifold pressure, as well as the turbocharger speed, demonstrate the importance of the turbocharger dynamics in predicting the transient torque response. The instantaneous torque response is shown in the fourth plot in the right column. One can see that the time necessary to transition from the steady-state combustion mode to steady-state braking mode is approximately equal to **ten cycles**. It is also evident that a first order lag will not be adequate to capture the mean-value behavior. Detailed analysis and signal processing of the simulation data follow in the next section. However, the above rough estimate of settling time indicates that the actuator dynamics will dominate the engine dynamics. Currently, the conventional devices have a fixed delay of approximately 0.6 sec, which is necessary to pump-up the system's hydraulic pressure sufficiently to open the brake valve against the high cylinder pressure. For comparison, we state here that conventional service/friction brakes that are pneumatically actuated, have a typically 0.3 sec delay, and a pneumatic pressure transient of 0.5 sec. The new generation variable compression braking mechanisms are expected to be very fast (10 msec).

The plots clearly show that that interactions between the compression brake and the turbocharger dynamics are important to the retarding performance of the engine. Recent work by Hu et al. [16] emphasize the importance of such interactions. Dynamic coupling between the compression effort and the turbocharger power determines the engine response during the transient operation of switching back to the conventional power generation mode.

5.9   Model Validation

To validate the developed model, we compare model outputs with selected measurements from an experimental engine at an engine-dynamometer facility.

In Figure 16 we compare mean-value engine and brake shaft power, for steady-state engine conditions. The upper part of the plot shows the engine running in combustion mode, for nine different speed and fueling levels. Overall, the predicted values from our model (+) indicate an error less than 14 percent from the measured values (o). However, if we neglect the lowest fueling level, the error is less than 9 percent.

In the lower part of the plot, we compare modeled and measured compression brake power for the fixed brake valve timing used on the experimental engine. Overall, the maximum absolute error between modeled and measured values is less than 16 percent. For the speed range $(1100 < N_e < 1900$ RPM), the maximum absolute error is less than 8 percent.
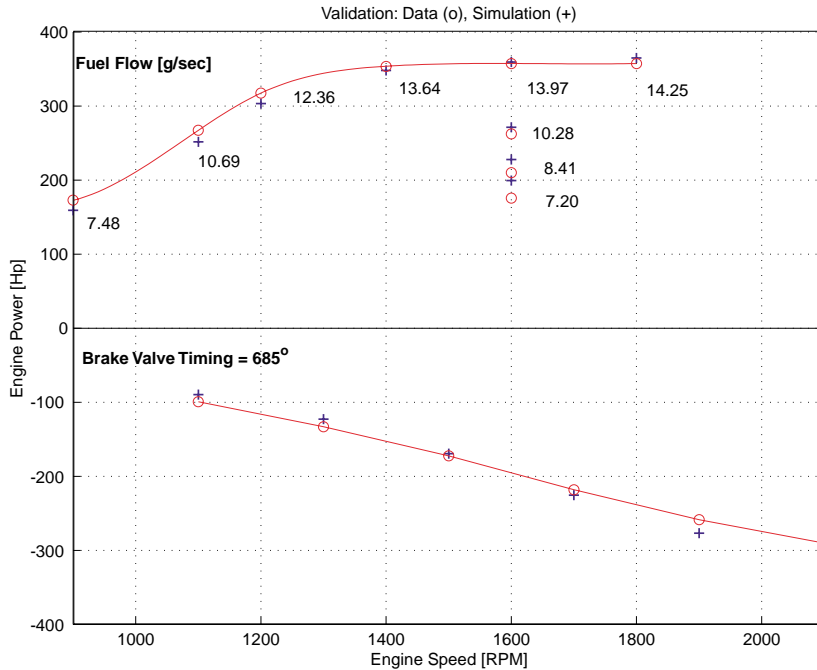
Figure 16: Comparison of predicted and measured mean-value shaft power.

Another way to validate our modeling work is to compare modeled and measured steady-state cylinder pressure in the crank angle domain. The four plots in Figure 17 show modeled and measured cylinder pressure for the engine operating in both combustion and braking mode, for different speed and loads.

The lower left plot in Figure 17 shows *qualitatively* an almost perfect match between modeled and measured cylinder pressure. However, to *quantify* how well the our models match the experimental engine, we calculate mean and standard deviation for the residual vector, $E = P_{mes} - P_{mod}$, for each simulation case. The following table lists the key statistical measures for the combustion mode:
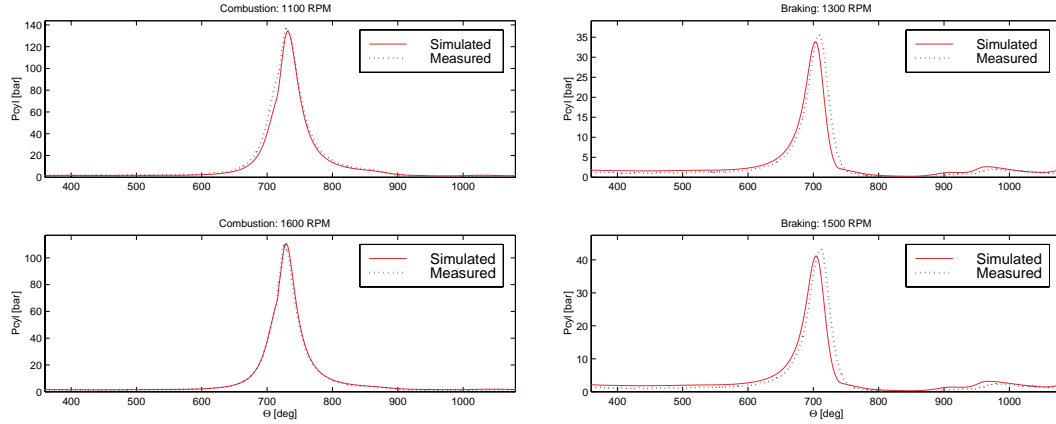
Figure 17: **Left:** Comparison of modeled and measured cylinder pressure for different speed and load during combustion mode. **Right:** Comparison of modeled and measured cylinder pressure for different speed and load during braking mode.

Combustion:	$N_e = 1100$	$N_e = 1600$	Total
Mean	1.16	- 0.44	0.36
Standard deviation	2.63	0.98	2.14

The mean value of the residual vector is zero for a perfect match, while the mean value is - 0.4 bar for the data shown in the lower left plot in Figure 17. Based on the standard deviation, we predict that 95 percent of the data is within $\pm$ 2 bar from the mean. The reason why this particular case matches the experimental data so well is that this case was used to calibrate the model during the modeling phase.

The upper left plot indicates a very good match for an operating point not used for calibration. The mean value of the residual vector is 1.2 bar, and 95 percent of the data for modeled cylinder pressure are predicted to be within $\pm$ 5.3 bar from the mean.

The comparison for the braking mode, shown in the two plots in the right column of Figure 17, is not quite as impressive as the one for the combustion mode. One reason for this is that, although we use the same nominal brake valve timing as the test engineers did when they collected the data, it is impossible to measure when the valve opening really occurred. A small deviation from the nominal value will affect the cylinder pressure. Again, to quantify the match, the following table of the key statistical measures for the braking mode is used:

Braking:	$N_e = 1300$	$N_e = 1500$	Total
Mean	- 0.19	- 0.33	- 0.26
Standard deviation	2.39	3.11	2.77

Overall, the mean value for the residual vector is around - 0.3, and 95 percent of the braking data are predicted to be within $\pm$ 5.5 bar from the mean.

5.10 Averaging and Identification of Mean Value Torque

The engine simulation results in [30] demonstrate that the dominant torque dynamics are in the order of engine cycles (10^{-2} second) and not in the order of crank angles (10^{-4} second). Moreover, the complexity of the 23 states in the crank angle based model precludes the development of control algorithms for in-vehicle applications. Hence, we want to average all the quasi-periodic, crank angle based dynamics. Analytically this can be done using singular perturbation theory, but such a rigorous treatment is not currently available. Developing such analytical techniques is, however, an active research topic.

We demonstrate here the dominant input-output model characteristics with brake valve timing v_{eb}, engine speed N_e, and fuel flow v_f, as inputs, and torque TQ, as output. The crank angle based model simulation for a step change in v_{eb} from 650° to 643° is shown in Figure 18. The lower plot in Figure 18 shows the cycle-averaged torque response obtained by processing the summation of torque in the crank angle domain with a third order Butterworth filter. The cutoff frequency corresponds to one engine cycle.
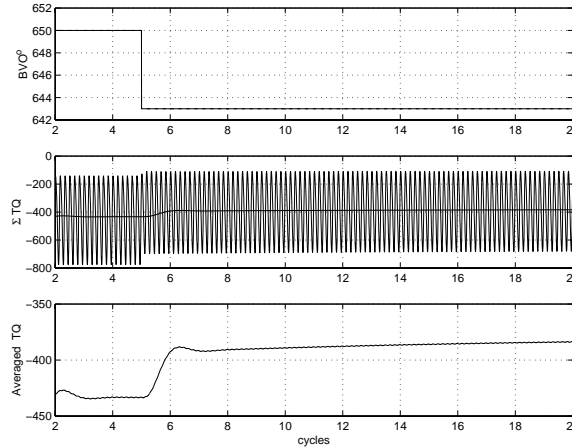
Figure 18: Simulation result show how the summation of torque is affected by a step change in v_{eb} from 650° to 643°. The third plot shows the event-averaged torque response.

To avoid the delay in the filtered output torque associated with the Butterworth filter, we use a moving average technique to extract the dominant cycle-to-cycle dynamical behavior of the crank angel based instantaneous shaft torque response. The shaft torque is sample every 2 crank angle degrees during simulations, and then averaged over one fundamental cylinder event. For a six cylinder engine, one cylinder event corresponds to 120 crank angle degrees.

Using the event-based averaged torque signal, we apply system identification techniques to a series of output perturbations due to small input step changes around various equilibrium points. Note here that the "first" data point in the averaged torque signal is sensitive to the choice of the window of the crank angle resolved data. The "first" data point refers to the averaged torque data point after the input step change. All the other averaged torque data points are robust with respect to the data window.

Our goal is to approximate the nonlinear averaged torque using a set of linear time invariant (LTI) systems in the neighborhood of selected operating points that span

23

the input excitation domain. Specifically, we generate LTI systems that represent both combustion and braking modes by a series of averaged torque simulations in response to fuel flow, brake valve timing and engine speed steps. These series of simulation serve as input-output identification experiments. The output-error model in the System Identification Toolbox in Matlab is used to extract the parameters of the LTI systems that capture the input-output behavior of each identification experiment.

To facilitate the development of a nonlinear controller, we develop a reduced engine model by interpolating the parameters of the series of LTI systems with polynomials of engine speed and the unified engine signal (scaled fuel flow and brake valve timing). Developing the polynomial regression of the LTI system parameters requires constant model order for all the LTI system approximations. A first order dynamical system is used as the target LTI approximation for all identification experiments. This is a challenging task because each engine mode is governed by different dynamics. In particular, the step responses to changes in engine speed while in braking mode, are more accurately approximated with second order systems than with first order ones. Furthermore, efficient system identification methodologies for a fixed structure (order) are not readily available. We, thus obtain some approximations by applying the classical identification rule of the rise time of a first order lag.

A few examples of the averaging and system identification we performed for braking, and combustion modes are shown in Figure 19 and 20, respectively. For braking mode, we show in Figure 19, the torque responses to: (i) a step change in brake valve timing v_{eb}, from 685 to 678 degrees, for a constant engine speed $\omega = 157$ rad/sec, and (ii) a step change in ω, from 157 to 165 rad/sec, for a constant $v_{eb} = 685$ degrees.

In Figure 20, on the other hand, we show two examples for the combustion mode. Specifically, we show the torque responses to: (i) a step change in fuel flow v_f, from 10 to 11 g/sec, for a constant engine speed $\omega = 157$ rad/sec, and (ii) a step change in ω, from 157 to 165 rad/sec, for a constant $v_f = 10$ g/sec.
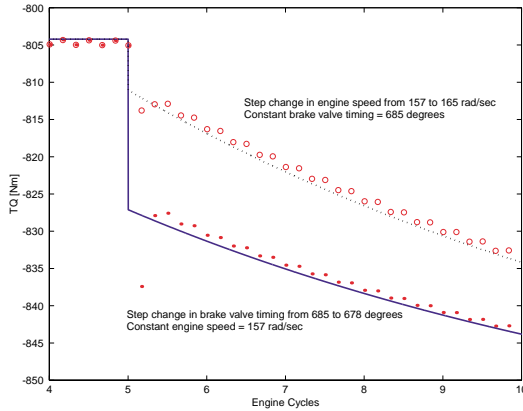
Figure 19: Averaged (dots) and identified torque response to: step in brake valve timing (solid line), and engine speed (dotted line).
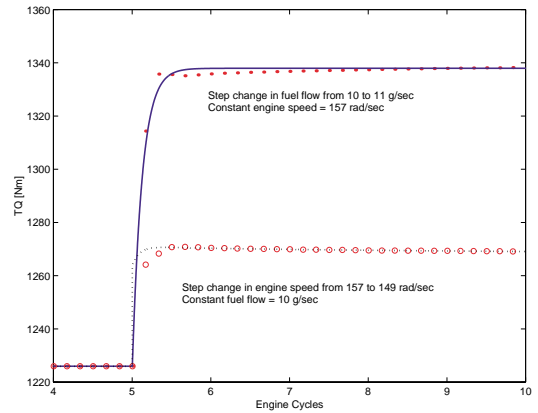
Figure 20: Averaged (dots) and identified torque response to step in: fuel flow (solid line) and engine speed (dotted line).

5.11 Reduced Order Engine Model

Based on the extracted family of first-order, local linear time invariant models, we develop a reduced order, nonlinear dynamic model of the engine as a unified torque actuator. In accordance with engine torque identification process described in Section 5.10, the dynamics describing the combustion mode mirror the dynamics describing the braking mode. Using standard regression techniques, we employ the following polynomial parametrization for the unified torque $TQ(t)$ (see Appendix 9.2 for details):

$$TQ(t) = \mathcal{P}_{TQ}(\tilde{\omega}(t), W(t)). \tag{36}$$

Here, $\tilde{\omega}(t)$ is characterized by the following dynamics:

$$\begin{aligned}
\tilde{\omega}(t) &= \Delta\tilde{\omega}(t) + \omega_{nom}, \\
\tau_{\omega}\frac{d}{dt}(\Delta\tilde{\omega}(t)) &= -\Delta\tilde{\omega}(t) + \Delta\omega(t) + c_{\omega}\frac{d}{dt}(\Delta\omega(t)),
\end{aligned} \tag{37}$$

where ω_{nom} is the nominal engine speed, and $\Delta\omega$ is the deviation of the engine speed ω, from nominal engine speed; i.e. $\Delta\omega(t) = \omega(t) - \omega_{nom}$. $W(t)$ is the unified signal in the range [-100,100] percent; it takes the value $W^F(t)$ when the engine is in combustion mode, and $W^B(t)$ when the engine is in braking mode. The dynamics of $W(t)$ is described by the following equations:

$$\begin{aligned}
W(t) &= \Delta W(t) + W_{nom}, \\
\tau\frac{d}{dt}(\Delta W(t)) &= -\Delta W(t) + \Delta v(t) + c\frac{d}{dt}(\Delta v(t))
\end{aligned} \tag{38}$$

where W_{nom} refers to a nominal operating condition for the engine; i.e. W_{nom}^F corresponds to a nominal fuel flow, while W_{nom}^B corresponds to a nominal brake valve timing.

The input Δv, in (38), denotes fuel flow from the fuel pump actuator when the engine is in combustion mode, and brake valve timing when the engine is in braking mode. The equation that describes the dynamics of Δv, is given by:

$$\tau_a\frac{d}{dt}(\Delta v(t)) = -\Delta v(t) + \Delta u(t), \tag{39}$$

where $\Delta u(t)$ is the deviation of the output of the in-vehicle controller u(t) from the nominal signal W_{nom}; i.e. $\Delta u(t) = sat(u(t)) - W_{nom}$. (A PI-controller for braking only is described in details in Section 6.2). Note that the combustion operating mode is activated when $u(t)$ is positive; i.e. $u \geq 0$, and the braking mode is activated when $u(t)$ is negative, i.e. $u < 0$. The minimum and maximum values of u that define $sat(u(t))$ are -100, and 100, respectively.

The time constants τ_{ω}, τ, and τ_a, for the systems in (37), (38), and (39), respectively, and the zeros c_{ω} and c of the systems in (37) and (38), respectively, are obtained for different operating modes by the following set of polynomials of nominal

engine speed ω_{nom}, and nominal signal W_{nom} (see Appendix 9.2 for details):

$$\tau = \begin{cases} \mathcal{P}_{F1}(\omega_{nom}, W_{nom}^F), & \text{for combustion mode} \\ \mathcal{P}_{B1}(\omega_{nom}, W_{nom}^B), & \text{for braking mode} \end{cases}$$

$$c = \begin{cases} \mathcal{P}_{F2}(\omega_{nom}, W_{nom}^F), & \text{for combustion mode} \\ \mathcal{P}_{B2}(\omega_{nom}, W_{nom}^B), & \text{for braking mode} \end{cases}$$

$$\tau_\omega = \begin{cases} \mathcal{P}_{F3}(\omega_{nom}, W_{nom}^F), & \text{for combustion mode} \\ \mathcal{P}_{B3}(\omega_{nom}, W_{nom}^B), & \text{for braking mode} \end{cases}$$

$$c_\omega = \begin{cases} \mathcal{P}_{F4}(\omega_{nom}, W_{nom}^F), & \text{for combustion mode} \\ \mathcal{P}_{B4}(\omega_{nom}, W_{nom}^B), & \text{for braking mode} \end{cases}$$

$$\tau_a = \begin{cases} \mathcal{P}_{F5}(\omega_{nom}, W_{nom}^F), & \text{for combustion mode} \\ \mathcal{P}_{B5}(\omega_{nom}, W_{nom}^B), & \text{for braking mode} \end{cases}$$

The unified engine torque given in (36) is shown Figure 21. Positive values on the x-axis indicates fuel flow, and negative values indicates brake valve timing. An interesting observation is that the rate of change for the brake torque changes sign for high valve timings in the reduced order model. However, the torque sign reversal is eliminated by imposing a hardware constrain on the maximum value of W^B.

Typical step responses for the dynamics given by (37) and (38) for a set of nominal operating points are shown in Figure 22. The normalization of the time scale is: $t_{norm} = \alpha t$, where $\alpha = 700$ for combustion, and $\alpha = 0.5$ for braking.
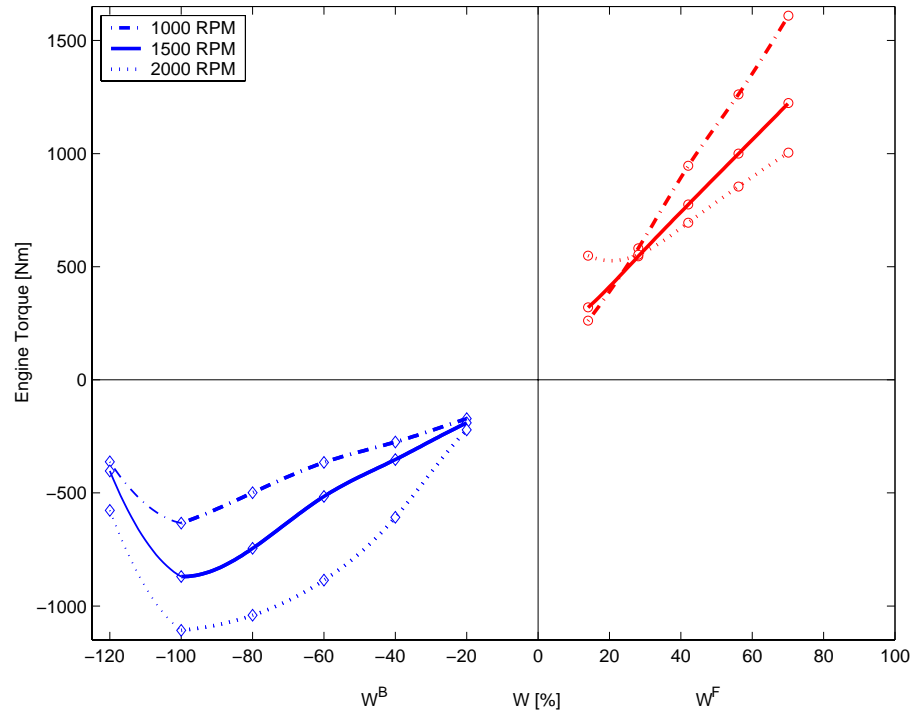
Figure 21: By combining fuel flow, and brake valve timing into one, we generate a map for the unified steady-state engine torque.
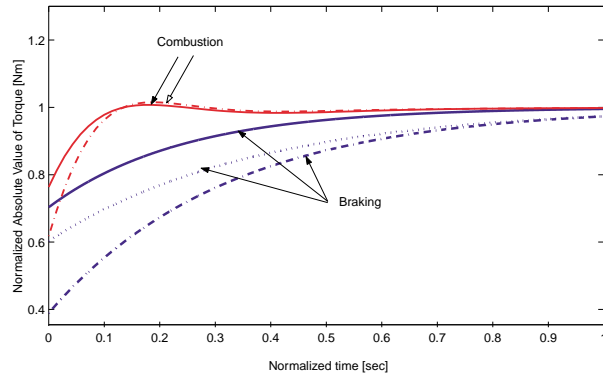
Figure 22: Step responses to a family of nominal operating conditions for braking and combustion modes.

27

6   Linear Controller Design

Using the results from the compression brake modeling section, Section 5, we obtain the full nonlinear longitudinal vehicle dynamics. In [29] we employ a classical PI-controller design to regulate the vehicle speed $v(t)$ to the desired constant vehicle speed v_d, during a long descent down a grade. The developments of the controller design based on the linear system model are discussed in details in this Section. Since the engine rotational speed ω, is related to the vehicle speed by $v = \omega r_g$, this ensures that $\omega \rightarrow \omega_d(t)$ as long as the gear is constant. Additionally, we assume that the braking with compression brakes is preferable, because we want to minimize the use of service brakes to potentially reduce the wear of the friction pads in the brakes.

6.1   Linearization of Longitudinal Vehicle Dynamics

We first linearize the model around a nominal point and use frequency domain and time domain response analysis to quantify the open-loop sensitivity to parameter variation. The longitudinal vehicle model is given in (40):

$$J_T \frac{d\omega}{dt} = TQ(v_{eb}, \omega) + r_g \left(F_\beta(\beta) + F_{sb}(v_{sb}) - F_{qdr}(\beta) - F_r(\beta) \right). \tag{40}$$

Linearization of (40) around a nominal point, indicated by $|_0$, leads to the following expression:

$$
\begin{aligned}
J_T \frac{d}{dt}(\Delta\omega) &= \frac{\partial TQ}{\partial v_{eb}} \Big|_0 \Delta v_{eb} + \frac{\partial TQ}{\partial \omega} \Big|_0 \Delta\omega - r_g M g \cos(\beta_0) \Delta\beta \\
&\quad + r_g \frac{\partial F_{sb}}{\partial v_{sb}} \Big|_0 \Delta v_{sb} - \rho C_q A r_g^3 \omega_0 \Delta\omega - \mu M g \sin(\beta_0) \Delta\beta.
\end{aligned} \tag{41}
$$

Laplace transformation of (41) results in following linear model for the vehicle dynamics:

$$
\begin{aligned}
J_T s \Delta\omega(s) &= G_{eb_v}(s)\Delta v_{eb}(s) + G_{eb_\omega}(s)\Delta\omega(s) + G_\beta(s)\Delta\beta(s) \\
&\quad + G_{sb_v}(s)\Delta v_{sb}(s) + G_a(s)\Delta\omega(s) + G_r(s)\Delta\beta(s).
\end{aligned} \tag{42}
$$

Grouping of terms, and closing the internal feedback loop that exists for the compression brake, the following expression for the linear vehicle dynamics is derived:

$$\omega(s) = G_{eb}(s)\,\Delta v_{eb}(s) + G_{sb}(s)\,\Delta v_{sb}(s) + G_\beta(s)\,\Delta\beta(s), \tag{43}$$

$$G_{eb}(s) = \left(\frac{G_{eb_v}(s)}{J_T s - G_{eb_\omega}(s) + \rho C_d A_v r_g^3 \omega_0} \right), \tag{44}$$

$$G_{sb}(s) = \left(\frac{r_g G_{sb_v}(s)}{J_T s - G_{eb_\omega}(s) + \rho C_d A_v r_g^3 \omega_0} \right), \tag{45}$$

$$G_\beta(s) = -\left(\frac{r_g M g \left(\cos(\beta_0) + \mu \sin(\beta_0) \right)}{J_T s - G_{eb_\omega}(s) + \rho C_d A_v r_g^3 \omega_0} \right), \tag{46}$$

where $G_{eb_v}(s)$ is the Laplace transformation of (37), and $G_{eb_\omega}(s)$ is the Laplace transformation of (38). $G_{sb_v}(s) = r_g F_{sb}(s)$, where $F_{sb}(s)$ is given by the Laplace transformation of the linearization of (3).

Figure 23 shows the frequency response of the transfer functions from each of the three inputs: $v_{eb}(s)$, $v_{sb}(s)$, and $\beta(s)$, to the speed output $\omega(s)$ as defined in (43). Included in the plot in Figure 23 is also the dynamics of the engine retarder when
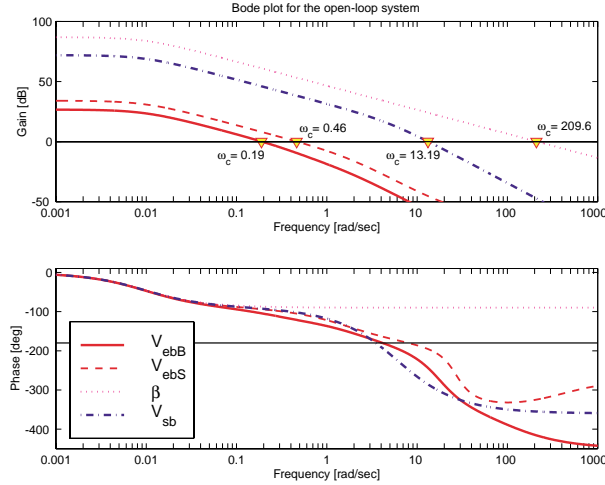
Figure 23: Bode plot for the open-loop system.

we switch from combustion to braking mode. Although "lowpassed" through the vehicle rotational dynamics, it is clear that these two modes are governed by different dynamics (see discussion in Sections 5.10 and 5.11).

6.2 PI Controller

The PI-controller for *braking* only is given by:

$$u(t) = k_b \left((\omega_d - \omega(t)) + \frac{1}{\tau_b} \int_0^t (\omega_d - \omega(\tau)) d\tau \right) + W_{nom}^B. \tag{47}$$

Recall that the output of (47), $u(t)$, is the input to the dynamics in (39).

Coordination of the compression brake with the service brakes is achieved using the following P-controller:

$$u_{sb}(t) = -k_{s1}(u - sat_{min}(u)) + k_{s2}(\omega - sat_{max}(\omega)), \tag{48}$$

where u_{sb}, is the input to the service brake actuator. Based on (48), the service brakes are activated when the control signal for compression brake (47) reaches its minimum value $sat_{min}(u) = -100$, or when the engine speed exceeds a safe operating level, $sat_{max}(\omega) = 250$ rad/sec. The control strategy based on (47)-(48) assigns high priority to the compression brake and uses the service brakes only when absolutely necessary. This reduces the use of conventional service brakes, thus potentially reduce maintenance costs. The block diagram of the controller scheme is shown in Figure 24 .

6.3 Analysis of the Linear Reduced Order Model

In this subsection we perform analysis of the linear reduced order model. Figure 25 shows the frequency response for the open-loop compression brake sub-system, where the input
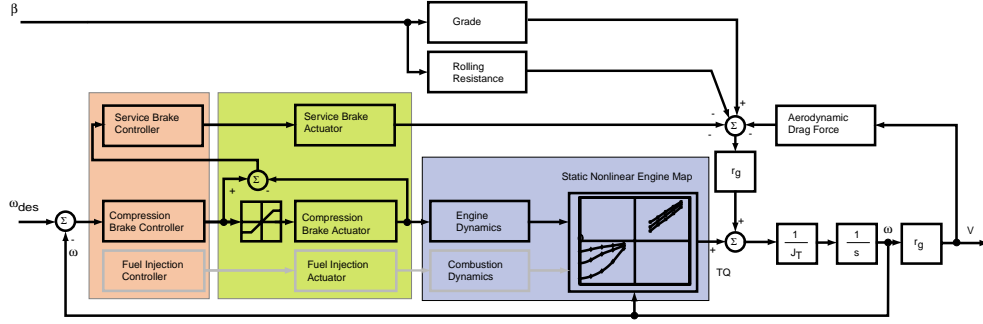
Figure 24: Block diagram for a coordinated braking controller.

is the controller input $e = \omega_{des} - \omega$, and the output is the engine speed ω. The important information here is that for our choice of controller, the phase margin $\phi = 74^o$ at a frequency $\omega_c = 1.5$ rad/sec; i.e. the bandwith of our system is 1.5 rad/sec.
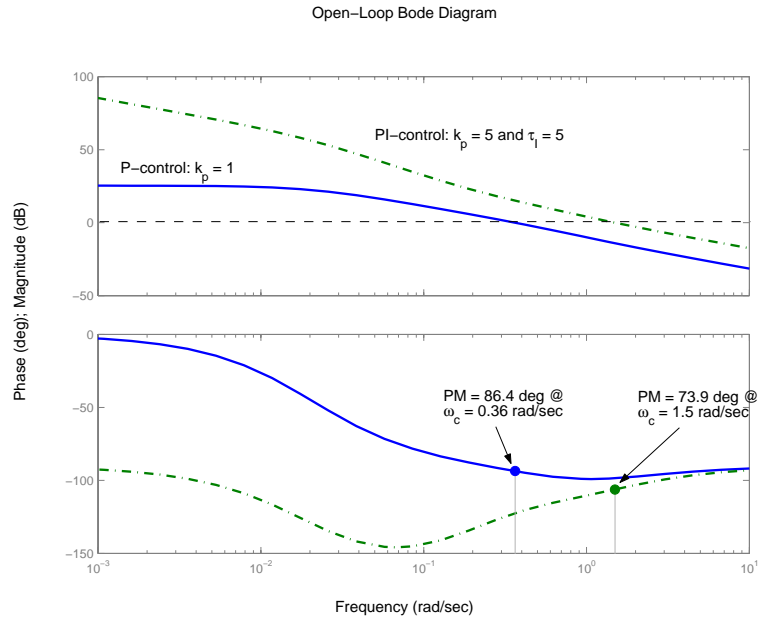
Figure 25: Open-loop frequency response a P-controller with unity gain, and a PI-controller with $k_p = 5$ and $\tau_I = 5$.

We denote L as the standard loop transfer function, and S and T as the sensitivity and the complementary sensitivity function, respectively:

$$
\begin{aligned}
L &= G_{eb}G, && \text{loop transfer function} \\
S &= (I + G_{eb}G_c)^{-1} & = (I + L)^{-1}, & \text{sensitivity function} \\
T &= (I + G_{eb}G_c)^{-1}G_{eb}G_c & = (I + L)^{-1}L, & \text{complementary sensitivity function} \quad (49)
\end{aligned}
$$

where G_c denotes the controller transfer function, and G_{eb} denotes the transfer function defined in (44). To be able to quantify stability and robustness properties of our controlled system, we adapt the commonly accepted upper bounds on S and T, as defined in [38]: $S_{max} = \parallel S \parallel_\infty$ should be less than about 6 dB and $T_{max} = \parallel T \parallel_\infty$ should be less than about 2 dB. In terms of gain and phase margins, $S_{max} < 6$ dB implies the common design

rule, $GM > 6$ dB and $PM > 30^o$. Values larger than 12 dB on either S or T indicate poor performance as well as poor robustness.

Figure 26 shows the frequency response for L, S, and T with the given controller setting. As indicated on the figure, the maximum values for S and T are well below the stated upper bounds providing a conservative design.
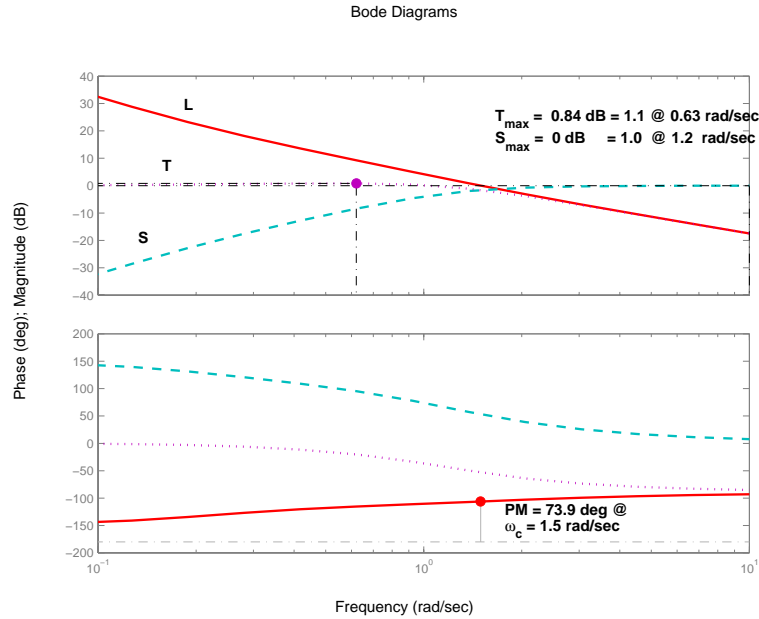
Figure 26: Frequency response for PI-controller with $k_p = 5$ and $\tau_I = 5$.

The closed-loop response to a unit step in the reference input ω_{des} is given in Figure 27. The time constant $\tau_{63} = 0.6$ sec and the settling time $t_s = 4.8$ sec. The maximum overshoot is 11 percent.
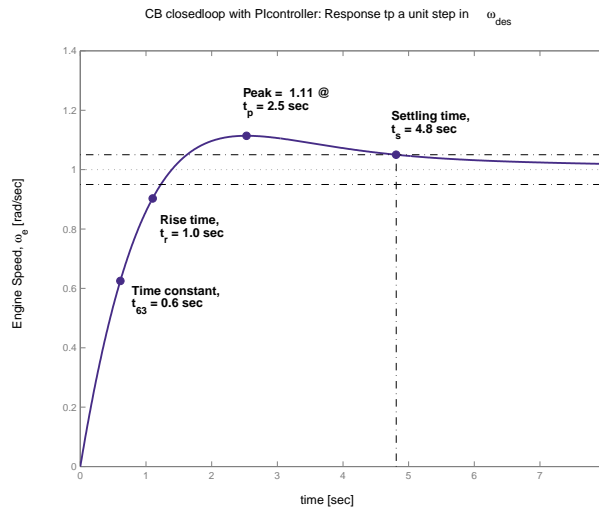
Figure 27: The closed-loop response to a unit step in the reference input ω_{des}

A plot of the open-loop pole and zero locations is shown in Figure 28. The open-loop system with a P-controller has three real poles and two zeros. However, one pole ($pole_2$)
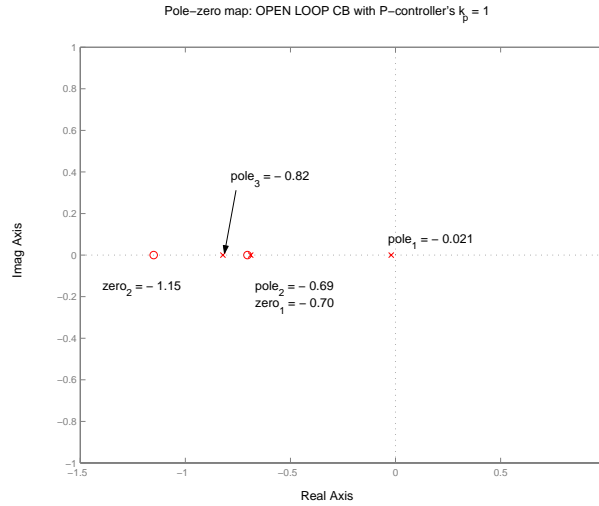
Figure 28: Map of poles and zeros for open-loop system.

and one zero ($zero_1$) are very close, and will in reality cancel each other out. Hence, the open-loop system has in fact only two poles and one zero:

$$H_{red} \quad = \quad \frac{(s + 1.15)}{(s + 0.82)(s + 0.02)} \tag{50}$$

Neglecting the zero, and comparing H_{red} to a second order transfer function on standard form, we find that the time constant $\tau = 7.8$ sec, and the damping $\zeta = 3.3$. The system is overdamped as expected, and it has a very low natural frequency $\omega_n = 1/\tau = 0.13$ rad/sec.

The locations of the closed-loop poles and zeros for our chosen controller setting is shown in Figure 29. The zero of the controller cancels out the smallest pole ($pole_1$) of the
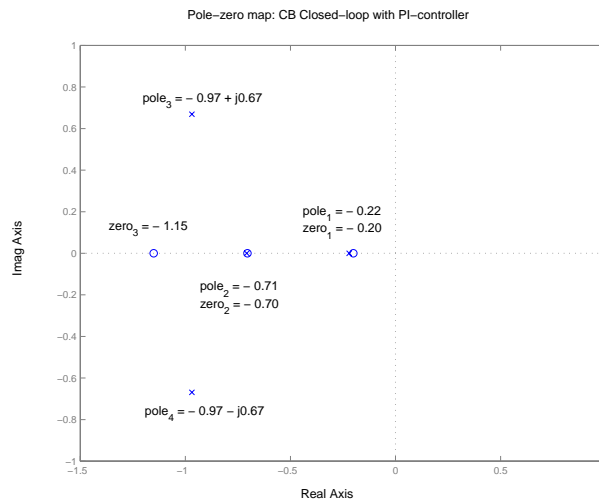
Figure 29: Map of poles and zeros for closed-loop system.

system. In addition, $pole_3$ in Figure 28 has been split into a complex conjugated pair of poles denoted $pole_3$ and $pole_4$ in Figure 29 due to the PI-controller.

6.4   Linear Controller Closed-Loop Performance

Simulation results of the designed longitudinal closed-loop system are included in this section. The simulations show the closed-loop performance of the nonlinear simplified vehicle model coupled with the high priority PI-controller. We identify five critical longitudinal maneuvers for which engine braking is instrumental. These maneuvers or driving scenarios are typical for CHVs and, therefore, important for the assessment of our work. The first driving scenario tests the closed-loop performance for a small step input change in desired vehicle speed. The second scenario demonstrates the disturbance rejection capability of the closed-loop system. The two first scenarios are within the linear operating region of the controller, so the service brakes are not activated. In the last three scenarios, large retarding demands are required. Specifically, in scenario three, a large step change in grade while in braking mode causes saturation of the compression brake. In scenario four, a small step change in grade while in combustion mode causes the activation of the compression brake and switching from combustion to braking. And finally, in scenario five, a large step change in grade while in combustion mode, triggers and saturates the compression brake and activates the service brakes.

In the following Figures 30-34, the right plot contains important engine and vehicle variables plotted on the same time axis, with the desired step changes in vehicle speed or road grade. The left column in each of the same figures, shows the phase plot of the retarding power versus vehicle speed. The phase plot demonstrates the importance of a non-equilibrium analysis of the system behavior.

Driving Scenario 1:

The driving scenario shown in Figure 30, illustrates a CHV in braking mode with a constant descending speed of 16.6 mph at a constant road grade of 4°. At $t =2$ sec, a unit step change in desired speed is required. The phase plot shows the speed-power trajectory between the two equilibrium points. The time plots to the right show the compression brake effort and the braking torque on the engine shaft. The last subplot shows a small overshoot in the vehicle speed before settling to the new equilibrium value after 14 seconds.

Driving Scenario 2:

Figure 31 shows the simulation results of a CHV in braking mode, on a 5° grade. At $t = 2$ sec, a step change in grade from 5° to 7° is introduced, and again speed regulation and disturbance rejection are the objectives. As seen in the last subplot in Figure 31, the speed increases slightly before returning to its initial value. The torque disturbance due to the step change in grade is effectively rejected by the closed-loop control scheme, and a stable and safe descending speed is achieved using the compression brake alone.
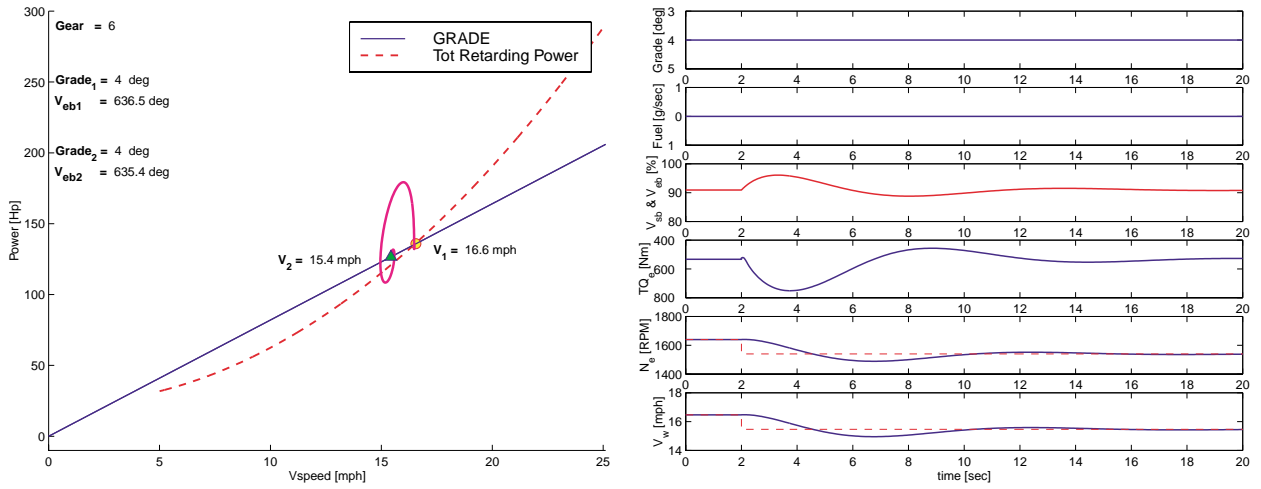
Figure 30: Step change in speed, closed-loop control of v_{eb}.
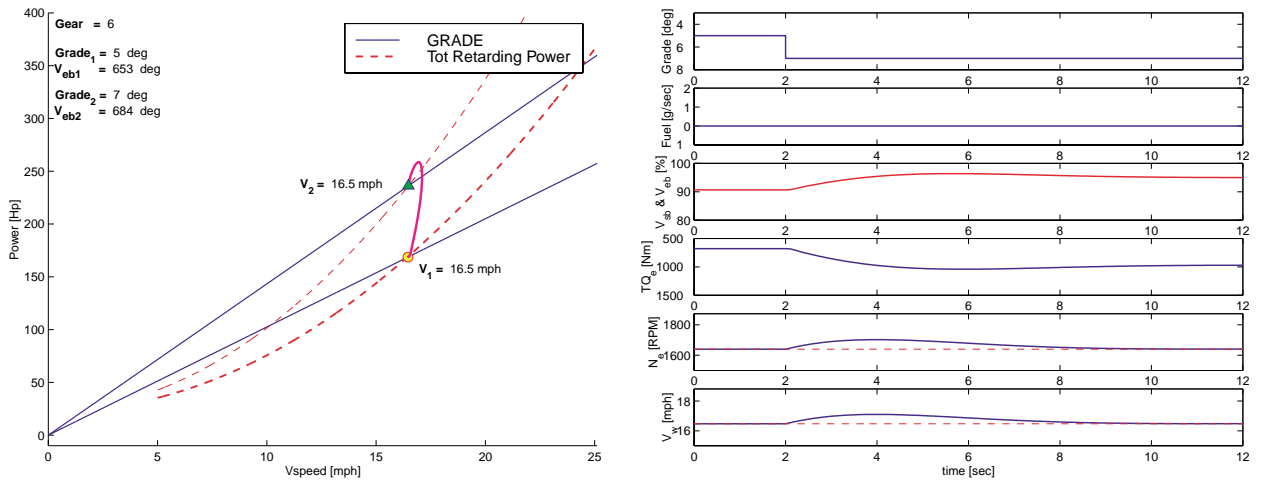
Figure 31: The system rejects torque disturbance introduced by a small step change in grade.

Driving Scenario 3:

In the third simulation shown in Figure 32, the CHV is once again in braking mode, and on a road with a descending grade of 5°. This time, however, the step change in grade at $t = 2$ sec is from 5° to 9° which is too large for the compression brake to handle alone. Upon saturation of the compression brake, the service brakes are activated, and the vehicle speed is regulated to its desired value.

Driving Scenario 4:

Figure 33 shows the resulting trajectories for a transition from combustion to braking. Initially, the CHV is cruising on a flat terrain at a constant speed. At $t = 2$ sec, the CHV enters a descending grade of 3°. Subplot 2 and 3 show that the controller cuts off the fuel and activates the compression brake. Speed regulation is achieved without using the service brakes.

Driving Scenario 5:

In the fifth and most demanding driving scenario shown in Figure 34, the CHV is once again cruising on a flat terrain with fuel flow, $v_f = 1.2$ g/sec. At $t = 2$ sec, the vehicle encounters a large descending grade of 6°, and subplot 2 shows the fuel cut. Subplot 3, on the other hand, shows first the activation, and then the saturation of the compression brake. Immediately following the saturation of the compression brake, the service brakes are activated through the high priority controller. By coordinating theses two actuators once again, the vehicle descending speed is regulated using a PI-controller on the compression brake, and a P-controller on the service brakes.
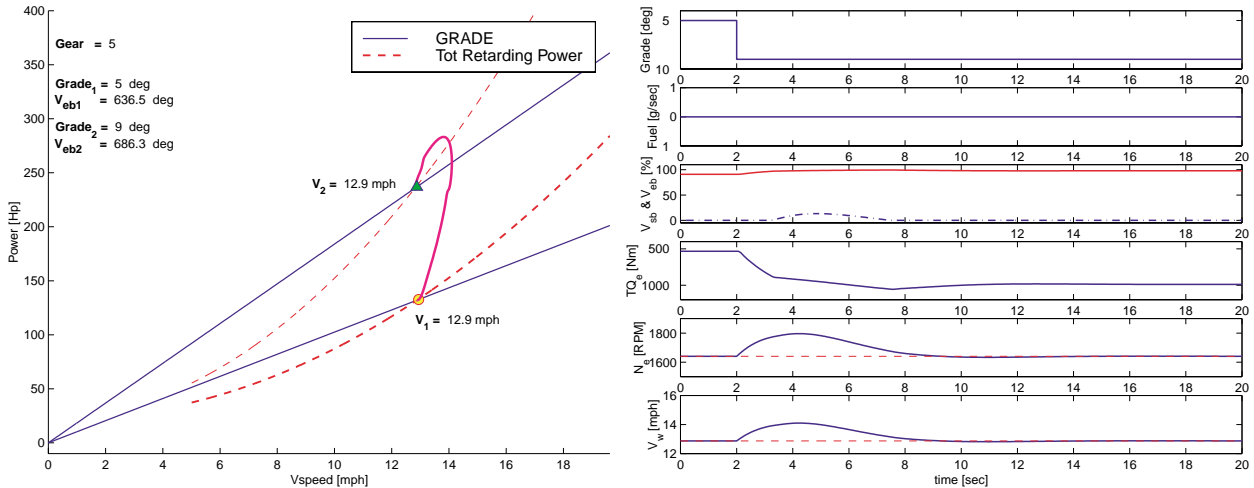
Figure 32: Large step change in grade requires the use of service brakes.
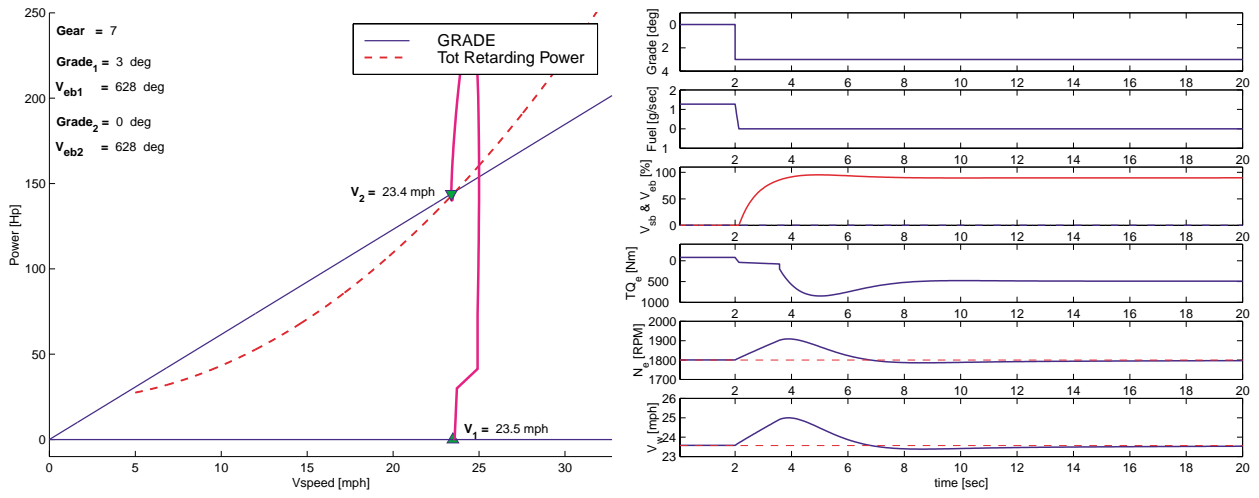
Figure 33: Switching from combustion to braking.
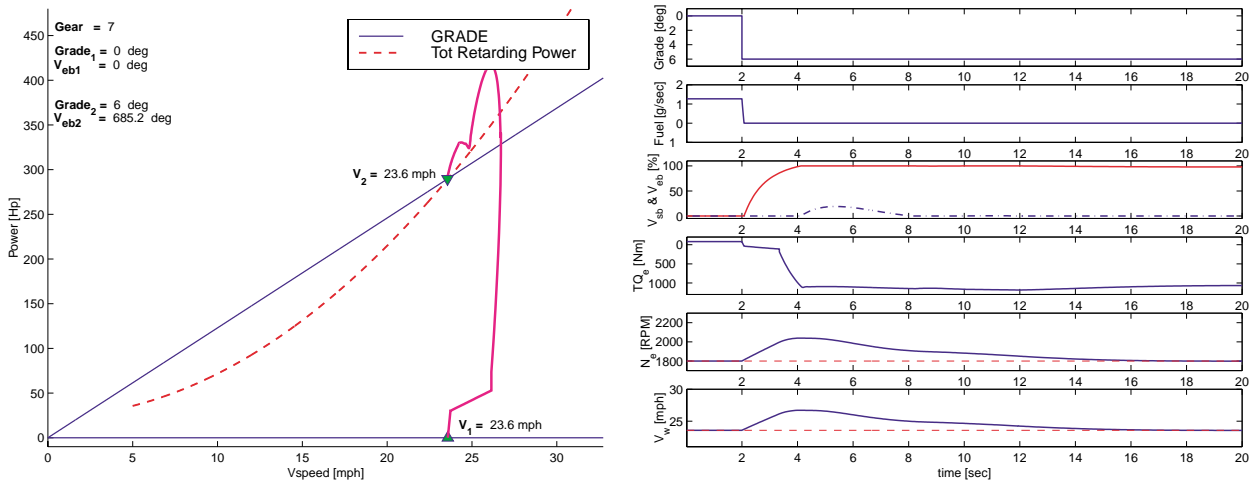
Figure 34: Switching from combustion to braking. This time the grade change is large and forces the system to apply service brakes.

6.5 Coordinated brake benefits

To assess the improvements we achieve using the CBC controller scheme, we simulate the same five driving scenarios described in Section 6.4, with the compression brake disabled. That is, we use PI-control on the service brakes only (SBO).

To quantify the improvements we achieve using the CBC controller scheme, we define the following performance index:

$$I = \int_0^{t_{set}} v_{sb}(\tau)^2 d\tau, \tag{51}$$

where the settling time t_{set} is defined as the time required for the service brake actuator signal v_{sb} to reach and remain inside a band whose width is equal to $\pm$ 5 % of the new steady state value [34].

The following Figures 35–39, show the same five driving scenarios shown in Figures 30–34 for both the CBC and the SBO controller scheme. In addition, the performance index defined in (51), and the corresponding settling time t_{set} are shown.
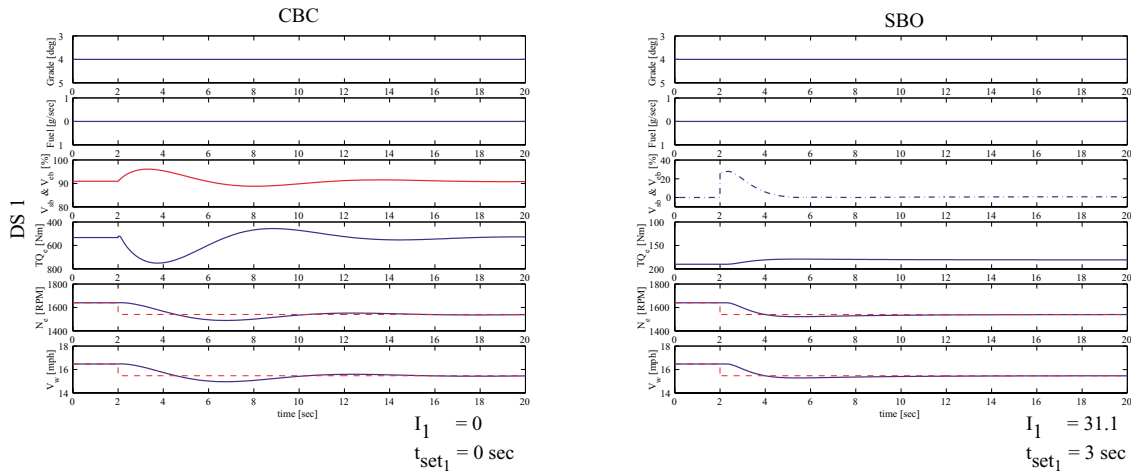
Figure 35: **Driving Scenario 1:** Step change in speed, closed-loop control of v_{eb}.

In the most challenging driving scenarios, DS 3 and DS 5, service brakes are used in both the CBC and the SBO controller schemes. The ratio between SBO and CBC is approximately 17.5 for DS 3, and 45 for DS 5. This means that using a coordinated braking controller scheme in critical longitudinal maneuvers, reduces the use of service brakes by a factor of 45.

If we, in addition to comparing performance indices, compare settling times t_{set} for the same two driving scenarios, we see that t_{set} is reduced from about 6.5 to about 4.2 seconds for DS 3, and from about 10 to 4 seconds for DS 5. DS 5 is the most critical longitudinal maneuver, so a decrease in the settling time by a factor of 2 is substantial when it comes to wear and tear on the friction pads in the service brakes.
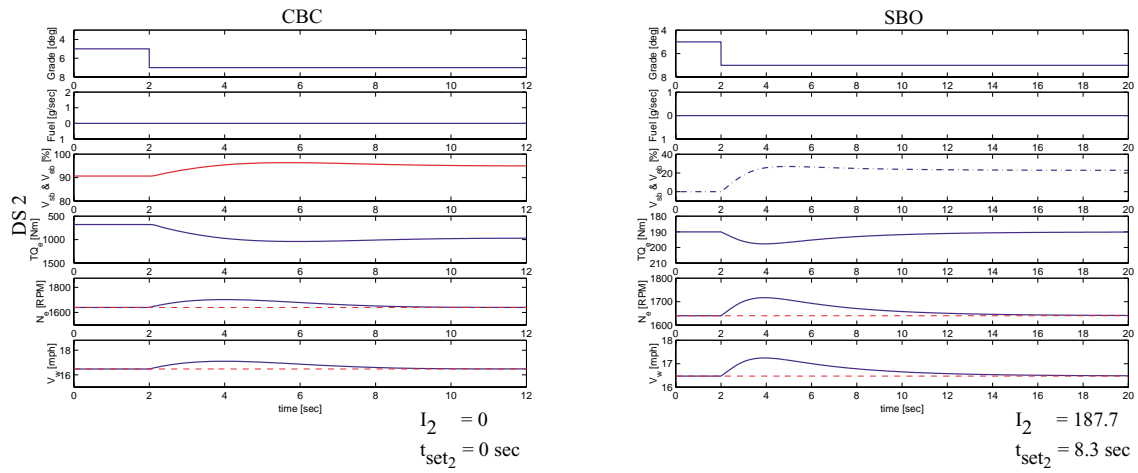
Figure 36: **Driving Scenario 2:** The system rejects torque disturbance introduced by a small step change in grade.
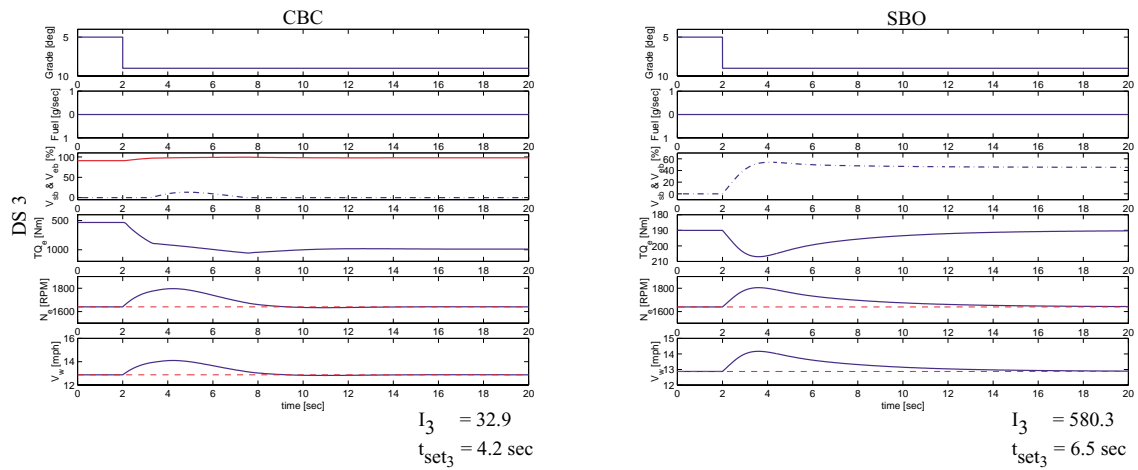
Figure 37: **Driving Scenario 3:** Large step change in grade requires the use of service brakes.
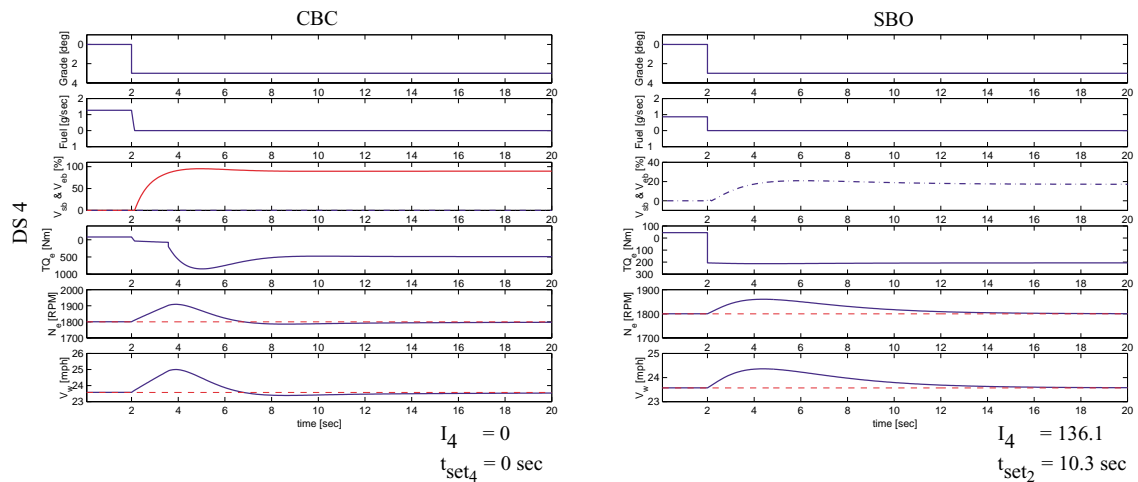
Figure 38: **Driving Scenario 4:** Switching from combustion to braking.
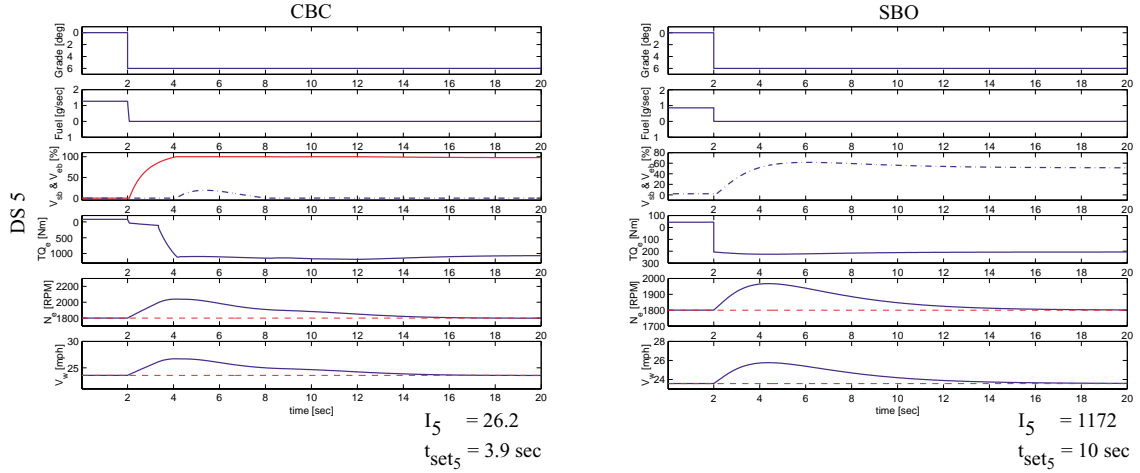
Figure 39: **Driving Scenario 5:** Switching from combustion to braking. This time the grade change is large and forces the system to apply service brakes.

6.6 Simulations with Nonlinear Full Order (NFO) and Linear Reduced Order (LRO) models: Comparison Remarks

In this section, we compare the full nonlinear and the reduced-order model's closed-loop performance. The closed-loop simulations with the full nonlinear model show good agreement with the closed-loop system when the linear reduced order model is used.

The first driving scenario shown in Figure 40, illustrates a HDV in braking mode, descending on a constant grade, $\beta = -2$ degrees (Subplot 1). The total gear ratio r_g (here, corresponding to the fifth gear), is kept constant throughout the simulations. After two seconds, we command a step change in desired engine speed from ω 157 to 149 rad/sec (Subplot 5). In Subplots 1 – 4 we show simulations on the nonlinear, full order (NFO) vehicle model (solid line) and simulations using the linear reduced order (LRO) model (dotted line). We are using the same PI controller with the same gains for both models, and currently, the proportional gain, $k_p = 5$ and the integral time, $\tau_I = 5$. Obviously, the nonlinear relationship between the torque and the v_{eb} is not fully captured by the linear reduced order model. Due to these nonlinearities, there is a small discrepancy between the closed-loop responses for v_{eb} and torque. However, the engine speed ω shown in Subplot 5 is practically identical for both models. This is due to the low-pass filtering effect of the large total vehicle inertia J_T.

The other critical driving scenario, shown in Figure 41, illustrates the HDV during a transition from combustion to braking mode, with a constant engine speed $\omega = 157$ rad/sec. The vehicle is initially cruising in combustion mode. After two seconds, the vehicle encounters a large grade change from $\beta = 2.4$ to -7.6 degrees (Subplot 1). Again we show nonlinear, full order (NFO) vehicle model and the linear reduced order (LRO) one. This time, the discrepancy between the two models is noticeably. The control signal, and subsequently v_{eb}, for the NFO model saturates due to the large grade change (Subplot 2 and 3). The saturation of u leads to the application of the service brakes as shown in the upper most part of Subplot 2. However, the LRO model does not saturate at all.
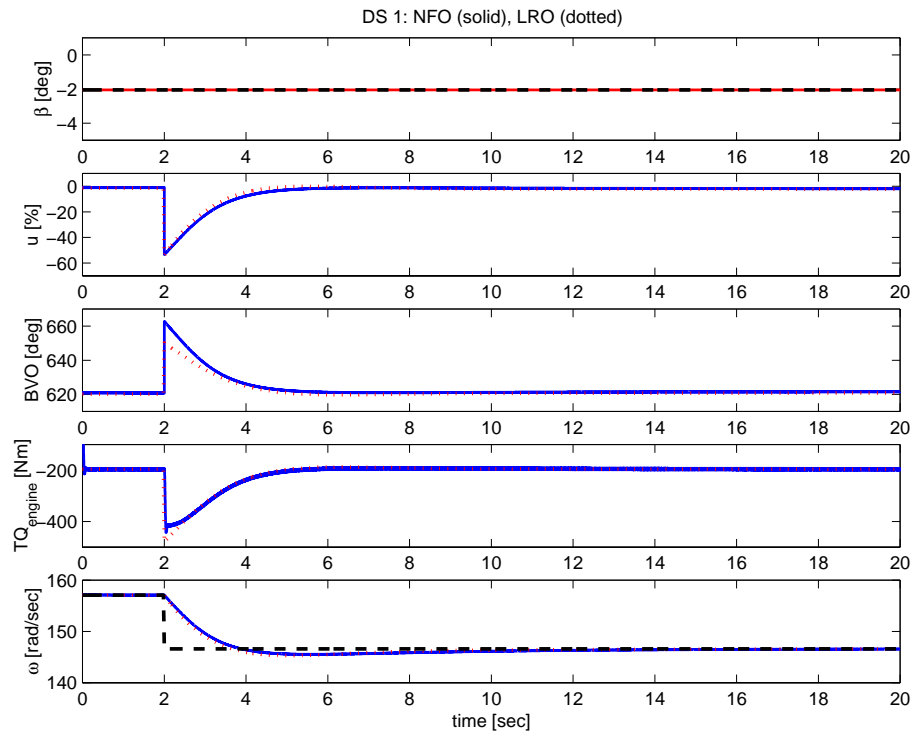
Figure 40: Step change in desired engine speed from 157 to 149 rad/sec when operating in braking mode.
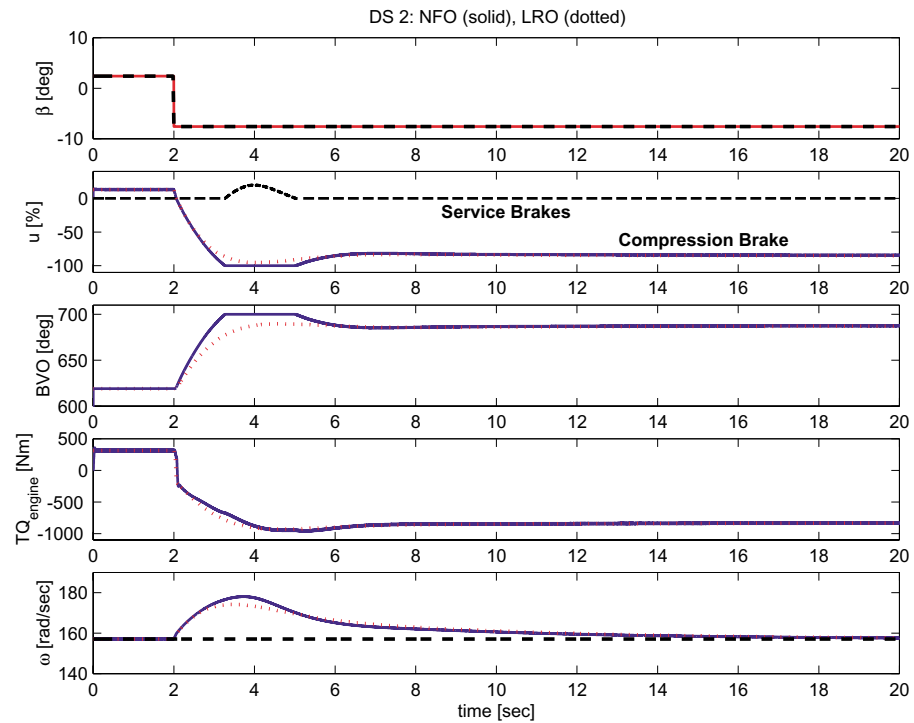
Figure 41: Switching from combustion to braking mode.

6.7 Sensitivity Analysis

In this subsection, we use time and frequency domain plots to investigate how sensitive the linear vehicle dynamics are to parameter variations.

Figure 42 shows the engine speed time responses for three different vehicle masses and gear ratios, due to a unit step change in u from nominal value corresponding to $v_{eb0} = 620^o$. There is a significant sensitivity to changes in gear ratios as seen in the upper plot in
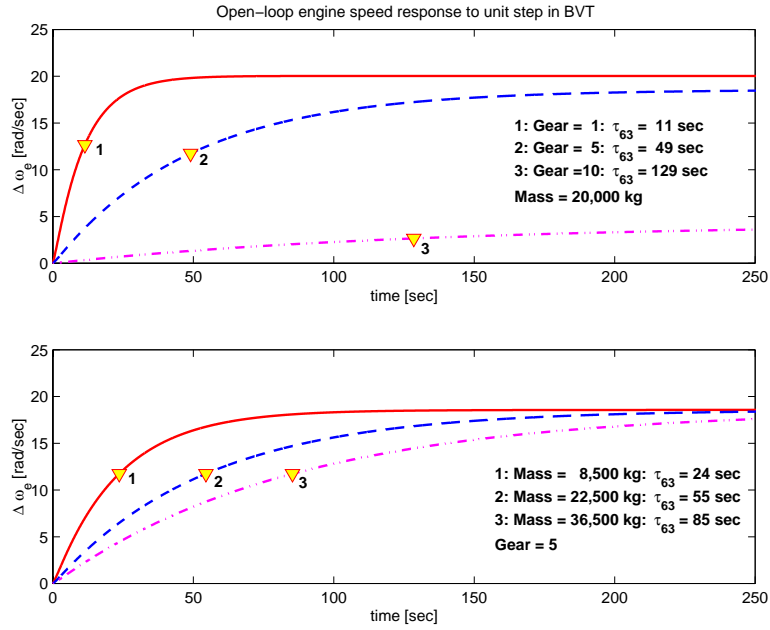
Figure 42: Engine speed time response to step change in brake valve timing.

Figure 42. In fact, the time constant for the system varies from 11 seconds when using the first gear, to 129 seconds when using the tenth gear.

Variations in the vehicle mass also greatly influence the vehicle dynamics. The mass for the system can vary as much as 400 percent from being tractor only, to being a system of tractor and trailer(s) with maximum allowable load. The lower plot in Figure 42, shows that the time constant τ_{63}, increases from 24 seconds for tractor alone to 85 seconds for the combination of a tractor and fully loaded trailer(s).

It is also of interest to investigate how much a unit step change in grade, from nominal grade $\beta_0 = -2^o$, affects the engine speed time response for the same three vehicle masses and gear ratios. This is shown in Figure 43. In the upper plot of this figure, we observe a non-monotonic behavior in the steady-state engine speed. The reason for this is that the aerodynamic force has a quadratic dependency on vehicle speed, $F_{qdr} = C_q v^2$. In addition, the speed component of the compression brake significantly contributes to this effect. The time constants, on the other, do show a monotonic behavior in accordance with the other time constants in Figure 42 and 43. Here, τ_{63}, varies from 11 to 128 seconds for gear one and ten, respectively.

For variations in mass, as shown in the lower plot in Figure 43, the time constant varies from 23 seconds when the system consists of trailer alone, to 85 seconds for a system of tractor and fully loaded trailer(s).

Our sensitivity analysis indicates the need for nonlinear and adaptive control design to deal with model uncertainties due to large parameter variations. Adaptive algorithms
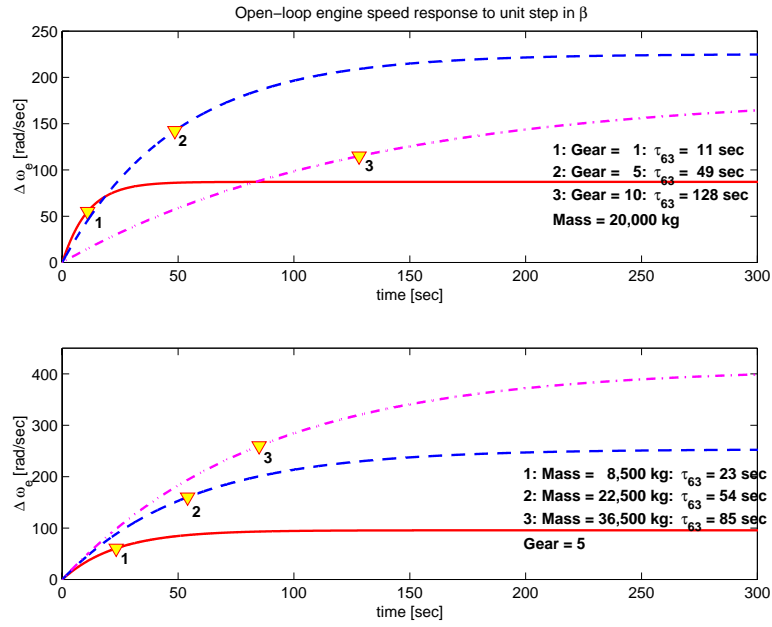
Figure 43: Engine speed response to step change in grade.

have been developed by Ioannou et al. [19] to address unpredictable changes in brake model parameters. Recent work by Maciuca and Hedrick [27], shows that non smooth estimation and adaptation techniques can be used to achieve satisfactory brake torque control under uncertainties and modeling errors. The authors of [6] are using the model reference adaptive control approach to ensure good CHV speed tracking performance during large variations in vehicle mass and road grade. To control effectively the large signal behavior of the system, the application of nonlinear control for the longitudinal speed control problem is investigated in [7],[8]. This is the topic of the following Section 7.

7 Nonlinear Controller Design

In this Section, we develop nonlinear controllers that accomplish both non-critical and critical braking maneuvers, including in-traffic vehicle following objectives. Recall that we consider the vehicle operation during a driving maneuver on a descending grade with β degrees inclination. We further assume that during the descent, the engine is not fueled and is operated in the compression braking mode. Recent developments in the area of valve actuation hydraulics (see e.g., [16]) allow us to assume that the actuator opening the brake valve (see the equation 39) is considerably faster than the engine dynamics. Consequently, the BVO timing, u_{eb}, can be treated as a control input while the engine dynamics have the dominant effect on compression braking dynamics (see Figure 6 in Subsection 4).

A detailed crank-angle based engine model that captures the effects of the variable compression braking and its reduced order approximation has been discussed in previous Sections. To simplify the analysis and control design we are going to treat the compression braking torque, TQ, as a static nonlinear function of brake valve opening timing, u_{eb}, and engine speed, ω (see Appendix 9.2 for details):

$$TQ(\omega, u_{eb}) = \alpha_0 + \alpha_1\omega + \alpha_2 u_{eb} + \alpha_3 u_{eb}\omega. \tag{52}$$

We will, however, include the compression braking dynamics in the simulation model that we use to test our control schemes.

Thus, the system under consideration has the following form:

$$J_t\dot{\omega} = \alpha_0 + \alpha_1\omega + (\alpha_2 + \alpha_3\omega)u_{eb} - C_q r_g^3 \omega^2 + r_g F_{r\beta} + r_g F_{sb}, \tag{53}$$

where $F_{r\beta}(m, \beta)$ is the force due to road grade (β) and the rolling resistance of the road(μ):

$$F_{r\beta} = F_\beta - F_r = -\mu g m \cos\beta - mg\sin\beta$$

and the timing of brake valve opening, u_{eb}, is the control input. The timing of brake valve opening is physically limited to the range $u_{eb}^{min} = 620$ to $u_{eb}^{max} = 680$ degrees. These BVO limits translate into limits on the torque

$$TQ^{min}(\omega) = TQ(u_{eb}^{max}, \omega), \ TQ^{max}(\omega) = TQ(u_{eb}^{min}, \omega).$$

The previous developments were concerned with controller design based on the linear system model. In reality, the vehicle model (53) is nonlinear. The nonlinearities are mainly due to the quadratic dependence of the aerodynamic force on ω and nonlinear dependence of the compression brake torque, TQ, on the engine speed and the BVO timing. To deal effectively with these nonlinearities we now consider the application of nonlinear control.

Recall that the speed control problem is to ensure that the vehicle speed $v(t)$ tracks the desired reference vehicle speed v_d as the truck proceeds the descending grade: $v \to v_d$. Since the engine rotational speed $\omega(t)$ is related to the vehicle speed by $v = \omega r_g$, this ensures that $\omega \to \omega_d$, where $\omega_d = \frac{v_d}{r_g}$ is the desired engine speed.

The desired controller is designed using the Speed-Gradient (SG) methodology [11] reviewed in Appendix 9.3. This is a general technique for controlling nonlinear systems through an appropriate selection and minimization of the goal function. The goal function Q is selected to address the speed regulation objective, i.e.

$$Q = \frac{J_t\gamma_0}{2}(v - v_d)^2 \geq 0, \ \gamma_0 > 0. \tag{54}$$

The controller is designed to provide the convergence to zero of the goal function (54) along the trajectories of the system (53) that implies the achievement of the speed regulation problem $v \to v_d$. Taking into account the relation between engine and vehicle speeds $v = \omega r_g$, the goal function can be rewritten as follows:

$$Q = \frac{J_t \gamma}{2}(\omega - \omega_d)^2 \geq 0, \;\; \gamma = \gamma_0 / r_g^2 > 0. \tag{55}$$

7.1 Speed Control During Non-Critical Maneuvers

In this subsection we develop nonlinear Speed-Gradient controllers that accomplish non-critical braking maneuvers during a long descent down a grade. Recall that during non-critical maneuvers the time necessary to achieve the desired speed is not critical. This is a frequent situation when, e.g., collision avoidance requirements are not a defining factor for the maneuver. To sustain the desired vehicle speed during a steady descent, we use compression brake only. However, we do consider large road grade changes and we account for compression brake saturation. To handle these large grade changes, the compression brake must be coordinated with gear changes. In the sequel, we first develop a SG-PI controller for compression braking that ensures robustness to a constant (or slowly varying) uncertainty in the grade. This development is a subject of Subsections 7.1.1, 7.1.2 and 7.1.3. To handle more general time-varying grades, an estimator for the torque on the vehicle due to the unknown grade is needed. This observer is developed and combined with the SG controller in Subsection 7.1.4. Finally, the development of the coordinated controller for the compression brake and gear ratio is a subject of Subsection 7.1.5.

7.1.1 Control Design

The control design is based on the vehicle model with compression brake only:

$$J_t \dot{\omega} = \alpha_0 + \alpha_1 \omega + (\alpha_2 + \alpha_3 \omega)u_{eb} - C_q r_g^3 \omega^2 + r_g F_{r\beta}, \tag{56}$$

where the timing of brake valve opening, u_{eb}, is the control input.

In accordance with SG method, we first calculate a time derivative of the goal function

$$Q = \frac{J_t \gamma}{2}(\omega - \omega_d)^2 \geq 0,$$

where $\omega_d = v_d / r_g$, along the trajectories of (56):

$$\dot{Q} = \gamma(\omega - \omega_d)\left(\alpha_0 + \alpha_1 \omega + \alpha_2 u_{eb} + \alpha_3 u_{eb}\omega - C_q r_g^3 \omega^2 + r_g F_{r\beta}\right)$$

and the derivative of $\dot{Q}$ with respect to u_{eb} ("speed-gradient"):

$$\nabla_{u_{eb}}\dot{Q} = \gamma(\omega - \omega_d)(\alpha_2 + \alpha_3 \omega).$$

Then the SG-PI control law looks as follows:

$$u_{eb} = u_d - k_p \nabla_{u_{eb}}\dot{Q}(\omega) - k_i \int_0^t \nabla_{u_{eb}}\dot{Q}(\omega(s))ds \tag{57}$$

where $k_p > 0$, $k_i > 0$ are the controller gains and u_d is the feedforward of desired value for the input:

$$u_d = \frac{r_g(C_q v_d^2 - F_{r\beta}) - \alpha_0 - \alpha_1 \omega_d}{\alpha_2 + \alpha_3 \omega_d}. \tag{58}$$

Note that (57) can be interpreted as traditional PI controller but with *nonlinear* gains which depend on engine speed ω. The feedforward term (58) depends on road grade β and aerodynamic coefficient that are usually unknown. However, as shown in Appendix 9.3, the implementation of the SG-PI controller (57) is possible without knowing precisely the value of u_d, due to the integral action.

7.1.2 Verifying Achievability Condition

The verification of the closed-loop stability is done in accordance with the procedure in Remark 4 (see Appendix 9.3 for details). According to the procedure let us consider a set

$$\Upsilon_C = \{\omega : \ Q(\omega) \leq C\} = \{\omega : \ (\omega - \omega_d)^2 \leq \frac{2C}{J_t \gamma}\}$$

for some $C > 0$ and then specify a value of $\tilde{C}$ such that for all $\omega \in \Upsilon_{\tilde{C}}$ the strong achievability condition

$$\dot{Q}(\omega, u_d) \leq -\rho Q(\omega), \tag{59}$$

where $\rho > 0$, holds. We re-arrange the model (56) as follows

$$J_t \dot{\omega} = \alpha_0 + \alpha_1 \omega + (\alpha_2 + \alpha_3 \omega) u_{eb} - C_q r_g^3 (\omega - \omega_d)(\omega + \omega_d) + T, \tag{60}$$

where $T = r_g F_\beta - r_g^3 C_q \omega_d^2$ is assumed to be a known function. Then, the feedforward term (58) has the following form

$$u_d = \frac{-\alpha_0 - \alpha_1 \omega_d - T}{\alpha_2 + \alpha_3 \omega_d}. \tag{61}$$

Let us first calculate $\dot{Q}$ under the assumption that $u = u_d$. We obtain,

$$
\begin{aligned}
\dot{Q}(\omega, u_d) &= \gamma(\omega - \omega_d)(\alpha_0 + \alpha_1 \omega + \frac{\alpha_2 + \alpha_3 \omega}{\alpha_2 + \alpha_3 \omega_d}(-\alpha_0 - \alpha_1 \omega_d - T) - \\
&\quad - C_q r_g^3 (\omega - \omega_d)(\omega + \omega_d) - T) = \\
&= \gamma(\omega - \omega_d)(\alpha_1(\omega - \omega_d) + (\alpha_0 + \alpha_1 \omega_d + T) \cdot (1 - \frac{\alpha_2 + \alpha_3 \omega}{\alpha_2 + \alpha_3 \omega_d}) - \\
&\quad - C_q r_g^3 (\omega - \omega_d)(\omega + \omega_d)) = \\
&= \gamma(\omega - \omega_d)\left(\alpha_1(\omega - \omega_d) + \alpha_3 u_d(\omega - \omega_d) - C_q r_g^3(\omega - \omega_d)(\omega + \omega_d)\right) = \\
&= -\gamma(\omega - \omega_d)^2 \left(-\alpha_1 - \alpha_3 u_d + C_q r_g^3(\omega + \omega_d)\right).
\end{aligned}
$$

Calculating the ratio $\frac{\dot{Q}(\omega, u_d)}{Q(\omega)}$, we have:

$$\frac{\dot{Q}(\omega, u_d)}{Q(\omega)} = -\frac{2}{J_t}(-\alpha_1 - \alpha_3 u_d + C_q r_g^3(\omega + \omega_d)) = -\frac{2C_q r_g^3}{J_t}(\omega + G), \tag{62}$$

45

where

$$G = \omega_d - \frac{\alpha_3 u_d + \alpha_1}{C_q r_g^3}. \tag{63}$$

It can be verified numerically that G is always positive for all physically feasible values of the grade β, mass m and desired engine speed ω_d. Note that (62) reaches its maximum value on the compact set Υ_C at $\omega = \omega_d - \sqrt{\frac{2C}{J_t \gamma}}$, i.e.,

$$k(C) = \max_{\omega \in \Upsilon_C} \left(\frac{\dot{Q}(\omega, u_d)}{Q(\omega)} \right) = -\frac{2 C_q r_g^3}{J_t} \left(\omega_d - \sqrt{\frac{2C}{J_t \gamma}} + G \right).$$

Therefore, for all $\omega \in \Upsilon_{\tilde{C}} = \{\omega : \ Q(\omega) \leq \tilde{C}\}$, where $\tilde{C}$ is any positive number such that

$$\tilde{C} < \frac{J_t \gamma}{2} (\omega_d + G)^2,$$

we can guarantee that the achievability condition (59) with $\rho = k(\tilde{C})$ always holds. In particular, since G is positive, we can select $\tilde{C} = \frac{J_t \gamma}{2} \omega_d^2$. Then, the set of initial states $\omega(0)$ in

$$\Upsilon_{\tilde{C}} = \{\omega : \ (\omega - \omega_d)^2 \leq \omega_d^2\}$$

is guaranteed to be recoverable by the controller (57) with $\theta(0) = 0$ provided that T and u_d are known (see remark 4 Appendix 9.3 for the case when T and u_d are unknown). This implies that the controller (57) with any positive gains $k_p > 0$, $k_i > 0$ is guaranteed to have a large region of attraction covering a very reasonable interval of initial values for the vehicle speed that corresponds to the engine speed interval of $[0, 2\omega_d]$, where $\omega_d = v_d / r_g$ is the desired engine speed.

7.1.3 Controller Performance during Small Changes in the Grade

We tested through simulations the operation of the SG-PI controller during a non-critical maneuver, when only compression braking is used to sustain the desired vehicle speed during a long descent. The vehicle mass is $20,000$ kg, and the value of desired vehicle speed $v_d = 8.78$ m/sec (or 31.6 km/h) corresponds to desired engine speed $\omega_d = 1500$ rpm in the gear number seven. Figures 44, 45 illustrate the SG-PI controller response to unmeasured changes in road grade. The implementation of the controller is done with a value of the feedforward term u_d calculated assuming a grade of $\beta = 2.5$ deg while the actual grade changes from 1.8 to 4.2 degrees. The unknown grade creates an unmeasured disturbance which is additive to the control input. As shown in Theorem 2 (see Appendix 9.3), the SG-PI controller ensures robustness properties to such kind of disturbances since the controller has an integral state which corrects the error in the feedforward u_d. The compression brake is used as the sole decelerating actuator, i.e., without activating service brakes. It can be seen that although the timing of BVO, u_{eb}, saturates during the initial transients the antiwindup compensation that we used in combination with our controller preserves good speed regulation performance.

7.1.4 Time-Varying Disturbance Rejection

In Section 3 we have shown that the SG-PI controller ensures robustness to unknown disturbances which are additive to the control input, in particular, unmeasured changes in

Figure 44: Controller responses to disturbance in road grade from 1.8 to 4.2 degrees: trajectories of grade, engine speed and vehicle speed. The desired engine and vehicle speeds are shown by the dashed line.
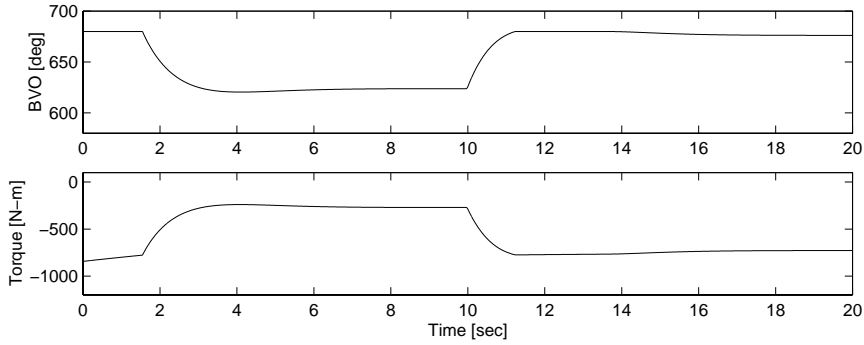
Figure 45: Controller responses to disturbance in road grade from 1.8 to 4.2 degrees: trajectories of BVO timing and compression torque.

road grade, due to the integral action of the controller. However, these disturbances must be constant (or slowly varying) for the integral action to compensate for them. In order to reject more general, unmeasurable fast varying disturbances due to road grade changes, an alternative approach can be used.

We treat the deviation from the nominal force due to grade as an unknown time-varying disturbance, i.e. the system (56) has the following form:

$$J_t\dot{\omega} = TQ - r_g F_{qdr} + r_g(F_{r\beta}^{nom} + \triangle F_{r\beta}), \tag{64}$$

where, $TQ = \alpha_0 + \alpha_1\omega + (\alpha_2 + \alpha_3\omega)u_{eb}$ is the compression braking torque, and $\triangle F_{r\beta} = F_{r\beta} - F_{r\beta}^{nom}$ is treated as an unknown function of time. It is reasonable to assume that the unknown function $\triangle F_{r\beta}$ and its derivative are bounded.

We, first, re-arrange the model (64) as follows

$$J_t\dot{\omega} = TQ - C_q r_g^3(\omega - \omega_d)(\omega + \omega_d) + TQ_{nom} + \chi, \tag{65}$$

where $TQ_{nom} = r_g F_{r\beta}^{nom} - r_g^3 C_q \omega_d^2$ is a known function, while $\chi(t) = r_g \triangle F_\beta(t)$ is an unknown

function of time, bounded together with its derivative, i.e.,

$$|\chi(t)| \leq L, \ |\dot{\chi}(t)| \leq \tilde{L},$$

for some constant $L > 0, \ \tilde{L} > 0$.

Our approach is to estimate the unknown disturbance torque, $\chi(t)$, with an observer that provides an estimate, $\hat{\chi}$, and then combine the observer with SG-P controller, i.e.,

$$u_{eb} = u_d - k_p \nabla_{u_{eb}} \dot{Q}(\omega) - \frac{\hat{\chi}}{\alpha_2 + \alpha_3 \omega}, \tag{66}$$

where $\nabla_{u_{eb}} \dot{Q} = \gamma(\omega - \omega_d)(\alpha_2 + \alpha_3\omega)$, $k_p > 0$ is the controller gain and the feedforward term u_d is selected as before (see Section 4.1) to balance the nominal system at the desired equilibrium:

$$u_d = \frac{-\alpha_0 - \alpha_1\omega_d - TQ_{nom}}{\alpha_2 + \alpha_3\omega_d}. \tag{67}$$

The observer for $\chi(t)$, motivated by [39], is defined as follows:

$$\hat{\chi} = \tau J_t \omega - \epsilon, \tag{68}$$

where $\tau > 0$ is an observer gain and ϵ is the solution of the following differential equation:

$$\dot{\epsilon} = -\tau(-TQ + C_q r_g^3 (\omega - \omega_d)(\omega + \omega_d) - TQ_{nom} - \hat{\chi}). \tag{69}$$

Denoting the estimation error as

$$e = \hat{\chi} - \chi = \tau J_t \omega - \epsilon - \chi,$$

let us consider the following Lyapunov function $V_{obs} = \frac{1}{2}e^2$. Calculating the time derivative of V_{obs} along the solutions of the system (65)-(69), we obtain:

$$\dot{V}_{obs} = e\dot{e} = e(\tau J_t \dot{\omega} - \dot{\epsilon} - \dot{\chi}) = e(\tau\chi + \tau\epsilon - \tau J_t \omega - \dot{\chi}) = e(-\dot{\chi} - \tau e).$$

Then, using the well-known inequality $ab \leq \frac{a^2}{2c} + \frac{b^2 c}{2}$, we have

$$\dot{V}_{obs} \leq -\tau e^2 + \tilde{L}e \leq -\tau e^2 + \frac{\tau e^2}{2} + \frac{\tilde{L}^2}{2\tau} = -\tau V_{obs} + \frac{\tilde{L}^2}{2\tau}$$

This implies that the estimation error $e(t)$ can be made arbitrarily small by amplifying the observer gain τ. Moreover, $e(t) \to 0$ if $\chi = const$, for any $\tau > 0$.

Remark: This observer design is equivalent to the estimation of $\dot{\omega}$ via dirty derivative, which is obtained by filtering both parts of (65) with $\frac{s}{\tau s + 1}$. In this case the estimate of χ can be calculated directly from (65).

We now consider the following Lyapunov function

$$V_1(\omega, e) = Q + \frac{1}{2}e^2 = \frac{J_t \gamma}{2}(\omega - \omega_d)^2 + \frac{1}{2}e^2,$$

and calculate its time derivative along the solutions of the system (65)-(69), taking into account the above expression for $\dot{V}_{obs}$:

$$\dot{V}_1 \quad \leq \gamma(\omega - \omega_d)(\alpha_0 + \alpha_1\omega + (\alpha_2 + \alpha_3\omega)(u_d - k_p\gamma(\omega - \omega_d)(\alpha_2 + \alpha_3\omega) - \frac{\hat{\chi}}{\alpha_2 + \alpha_3\omega}) -$$

$$- C_q r_g^3(\omega - \omega_d)(\omega + \omega_d) + TQ_{nom} + \chi(t)) - \tau\frac{e^2}{2} + \frac{\tilde{L}^2}{2\tau} =$$

$$= \dot{Q}(\omega, u_d)|_{\chi=0} - k_p\gamma^2(\omega - \omega_d)^2(\alpha_2 + \alpha_3\omega)^2 - \gamma(\omega - \omega_d)e - \tau\frac{e^2}{2} + \frac{\tilde{L}^2}{2\tau},$$

where $\dot{Q}(\omega, u_d)|_{\chi=0} = -\gamma(\omega - \omega_d)^2\left(-\alpha_1 - \alpha_3 u_d + C_q r_g^3(\omega + \omega_d)\right).$

As in Subsection 7.1.2, we can guarantee that the achievability condition

$$\dot{Q}(\omega, u_d)|_{\chi=0} \leq -\rho Q(\omega)$$

holds for some $\rho > 0$ and all $\omega \in \Upsilon_{\tilde{C}} = \{\omega : \ Q(\omega) \leq \tilde{C}\}$ with $\tilde{C} < \frac{J_t\gamma}{2}(\omega_d + G)^2$, where G is given by (63) with u_d calculated by (67). If $\omega \in \Upsilon_{\tilde{C}}$, we obtain:

$$\dot{V}_1 \quad \leq \quad -\rho \, Q(\omega) - k_p\gamma^2(\omega - \omega_d)^2(\alpha_2 + \alpha_3\omega)^2 - \gamma(\omega - \omega_d)e - \tau\frac{e^2}{2} + \frac{\tilde{L}^2}{2\tau} \leq$$

$$\leq \quad -\rho \, Q(\omega) - k_p\left(\gamma(\omega - \omega_d)(\alpha_2 + \alpha_3\omega) + \frac{e}{2k_p(\alpha_2 + \alpha_3\omega)}\right)^2 -$$

$$- \quad \frac{1}{2}e^2(\tau - \frac{1}{2k_p^2(\alpha_2 + \alpha_3\omega)^2}) + \frac{\tilde{L}^2}{2\tau}.$$

Suppose that the initial condition at time $t = 0$ is $(\omega(0), e(0))$ such that $V_1(\omega(0), e(0)) = Q(\omega(0)) + \frac{1}{2}e(0)^2 \leq \tilde{C}$. From the analysis of the expression for $\dot{V}_1$ it can be shown that for τ sufficiently large the set $\Omega_{\tilde{C}} = \{(\omega, e) : V_1(\omega, e) \leq \tilde{C}\}$ becomes positively invariant, and, in particular, $Q(\omega(t)) \leq V_1(\omega(t), e(t)) \leq \tilde{C}$ for all $t \geq 0$ so that $\omega(t) \in \Upsilon_{\tilde{C}}$ for all $t \geq 0$. Furthermore, both $e(t)$ and $Q(\omega(t))$ can be made ultimately bounded in a given (arbitrarily small) neighborhood of the origin. Consequently, the initial accuracy of grade estimation is important to guarantee a large domain of attraction for our controller. Note that the bandwidth of the observer, τ, does not depend on the magnitude of $\chi(t)$, only on the bound for the time-rate of change of χ, $\tilde{L}$. Furthermore, stability conditions place no restriction on the controller gain k_p as long as it is positive.

The observer (68)-(69) and TQ_{nom} depend on the aerodynamic coefficient C_q. For the SG-PI controller (66) of Subsections 7.1.1-7.1.3, the robustness to uncertainties in C_q was assured as these uncertainties only affected the feedforward term u_d. If the value of C_q is not accurately known the observer-based design (66)-(69) can use the best estimate of C_q, C_q^{nom} as shown next.

We denote the deviation of C_q from the nominal value by $\triangle C_q = C_q - C_q^{nom}$. Then, the model (64) can be rewritten as follows

$$J_t\dot{\omega} = TQ - C_q r_g^3(\omega - \omega_d)(\omega + \omega_d) + TQ_{nom,1} + \chi_1, \tag{70}$$

where

$$TQ_{nom,1} = r_g F_{r\beta}^{nom} - r_g^3 C_q^{nom}\omega_d^2$$

is a known function, while

$$\chi_1(t) = r_g\triangle F_\beta(t) - \triangle C_q\omega_d^2$$

49

is an unknown function that will be estimated by an observer. We assume that $|\chi_1(t)| \leq L_1$, $|\dot{\chi}_1(t)| \leq \tilde{L}_1$ for some constant $L_1 > 0$, $\tilde{L}_1 > 0$.

The feedforward is selected as before, in the form

$$u_d = \frac{-\alpha_0 - \alpha_1 \omega_d - TQ_{nom,1}}{\alpha_2 + \alpha_3 \omega_d}. \tag{71}$$

To estimate χ_1 we use the same observer as before, replacing the unknown C_q by its estimate C_q^{nom}, i.e.,

$$\begin{aligned}
\hat{\chi}_1 &= \tau_1 J_t \omega - \epsilon_1, \ \tau_1 > 0, \\
\dot{\epsilon}_1 &= -\tau_1(-TQ + C_q^{nom}(\omega - \omega_d)(\omega + \omega_d) - TQ_{nom,1} - \hat{\chi}_1).
\end{aligned} \tag{72}$$

It is, furthermore, possible to show that if the observer gain τ_1 is sufficiently large, then the same ultimate boundedness result can be obtained, i.e., both estimation error $e_1(t) = (\chi_1 - \hat{\chi}_1)$ and $Q(\omega(t))$ can be made ultimately bounded in a given (arbitrarily small) neighborhood of the origin.

The controller (66)-(69) is referred to as Speed Gradient Proportional-plus-Derivative (SG-PD) controller. This controller relies on the fast differential action to estimate and compensate the unmeasured disturbances, as opposed to the slow integral action of the SG-PI controller. Hence, one can expect much faster disturbance rejection with SG-PD controller in response to a grade change.

The operation of the SG-PD controller are tested through simulations during a non-critical maneuver, when the unknown road grade creates unmeasured time-varying disturbances from 2 to 4 degrees (see Figures 46,47). The implementation of the controller is done assuming the nominal grade value of 3 degrees. The responses show that the disturbance due to grade essentially does not affect the vehicle speed.
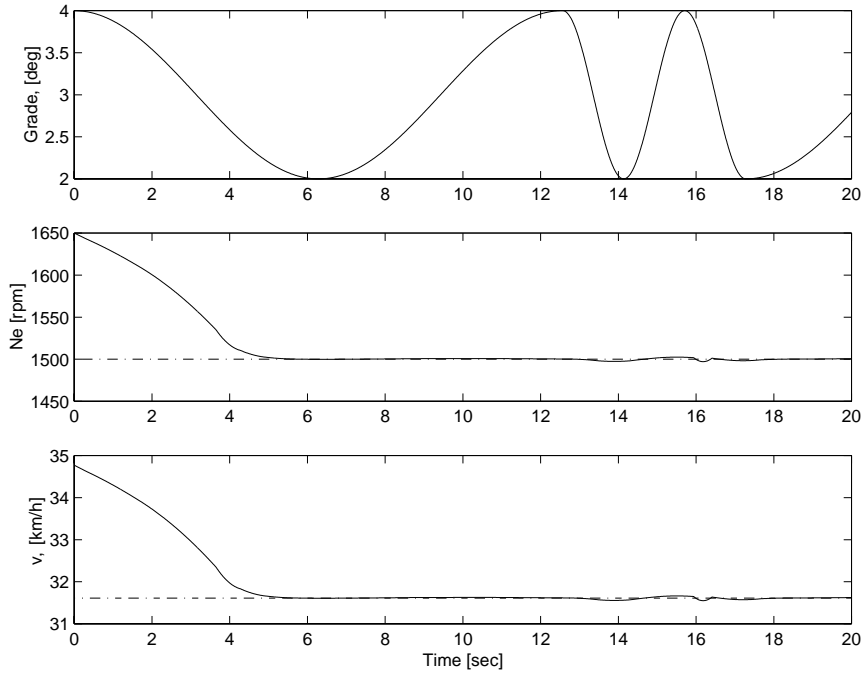
Figure 46: Controller responses to time-varying disturbance in road grade from 2 to 4 degrees: trajectories of grade, engine speed and vehicle speed. The desired engine and vehicle speeds are shown by the dashed line.

7.1.5   Coordination with Gear

Since the braking torque TQ is limited, in steady-state the compression brake can only support a certain range of vehicle speeds, v_d (or ω_d), for a given grade, β. Or, stated differently, given the desired vehicle velocity, v_d, we can only drive down a hill of a grade that falls within a certain range. To calculate this range, consider the steady-state balance of forces (or torques):

$$-TQ/r_g + F_{qdr}(v, r_g) = F_{r\beta}(m, \beta).$$

Given desired velocity v_d, gear ratio r_g and vehicle mass m, the determination of feasible grade range β_{min}, β_{max} is an elementary root-finding problem:

$$-\mu gm \cos \beta_{min} - mg \sin \beta_{min} = -TQ^{max}(v_d)r_g^{-1} + C_q v_d^2,$$

$$-\mu gm \cos \beta_{max} - mg \sin \beta_{max} = -TQ^{min}(v_d)r_g^{-1} + C_q v_d^2.$$

In the driving scenario, shown in Figure 44, the feasible values for the road grade are within the range $\beta_{min} = 1.62$ degrees, $\beta_{max} = 4.37$ degrees. Therefore, for given vehicle mass and gear ratio the resulting compression brake is capable to support the desired speed v_d during the maneuver on a descending grade with inclination from 1.8 to 4.2 degrees. However, if we operated on a grade β that exceeds the maximum value β_{max}, the compression brake would not be able to support the desired velocity v_d under the same values of the mass and gear ratio. In this case we need to switch the gear number to a lower one (downshift) in order to increase the braking capability. The gear switching can be done by following rule: we downshift from the gear number k to the gear number $k-1$ if the

51
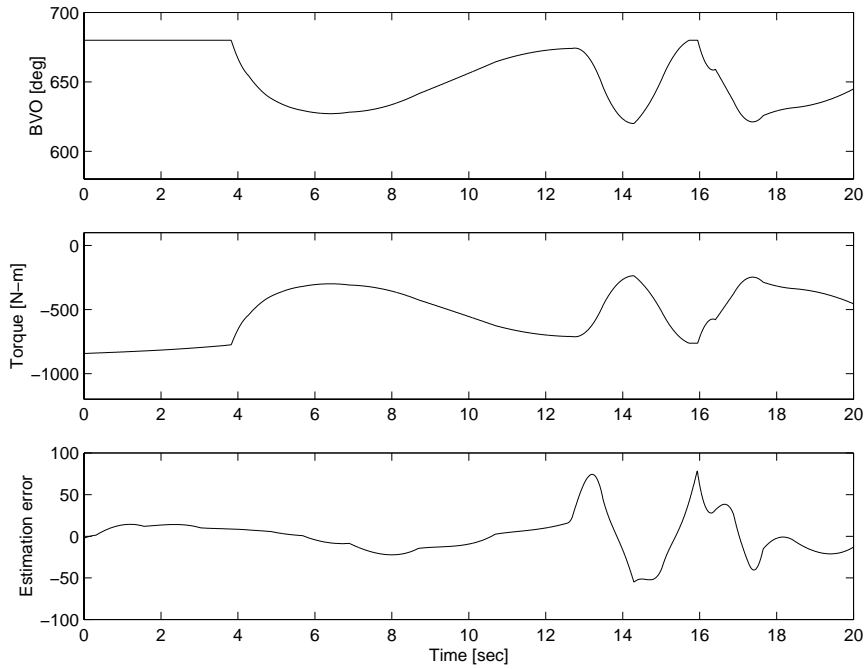
Figure 47: Controller responses to time-varying disturbance in road grade from 2 to 4 degrees: trajectories of BVO timing, compression torque and estimation error $e(t) = \hat{\chi} - \chi$.

timing of BVO u_{eb} saturates at the upper limit, (i.e. $u_{eb} = u_{eb}^{max}$) and the speed fails to decrease, i.e., $\dot{\omega} > 0$. If the gear $(k-1)$ is not sufficient (i.e, still $u_{eb} = u_{eb}^{max}$ and $\dot{\omega} > 0$) we downshift to gear number $(k-2)$, etc. Note that in this scenario it can happen that there exists no gear ratio which would be able to guarantee the desired speed v_d for given grade β within the allowable range of engine speed, $\omega_{min} \leq \omega \leq \omega_{max}$, where $\omega_{min} = 600$ rpm, $\omega_{max} = 2100$ rpm. In this case we need to activate the friction brake to supplement the lack of compression braking capability. A similar procedure is used for the upshifting based on the condition $u_{eb} = u_{eb}^{min}$ and $\dot{\omega} < 0$.

Figures 48, 49 illustrate the driving maneuver on a descending grade which changes from 1.8 to 7 deg. The value of the desired vehicle speed is $v_d = 8.78$ m/sec (or 31.6 km/h). The switch from the gear number seven to the gear number six takes place at $t = 10$ seconds that, therefore, implies the change in the desired engine speed ω_d. In particular, the value of desired vehicle speed $v_d = 8.78$ m/sec (or 31.6 km/h) corresponds to desired engine speed $\omega_d = 1500$ rpm in the gear number seven and $\omega_d = 1955$ rpm in the gear number six.
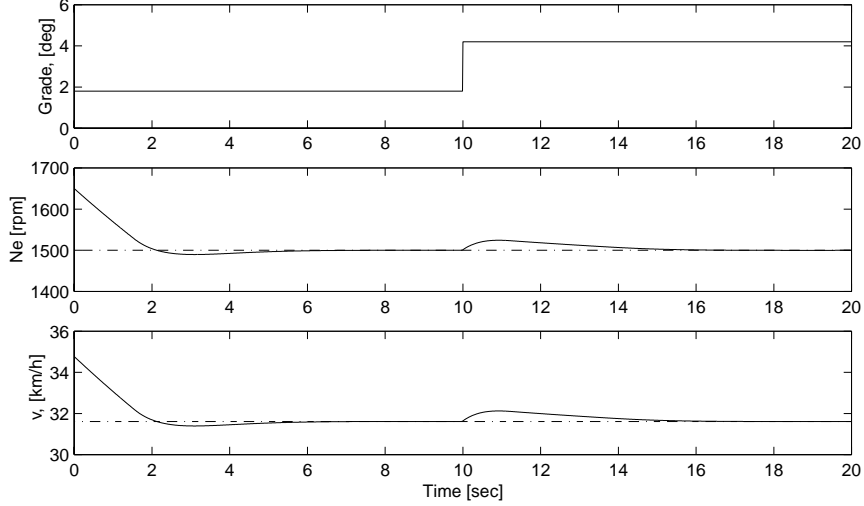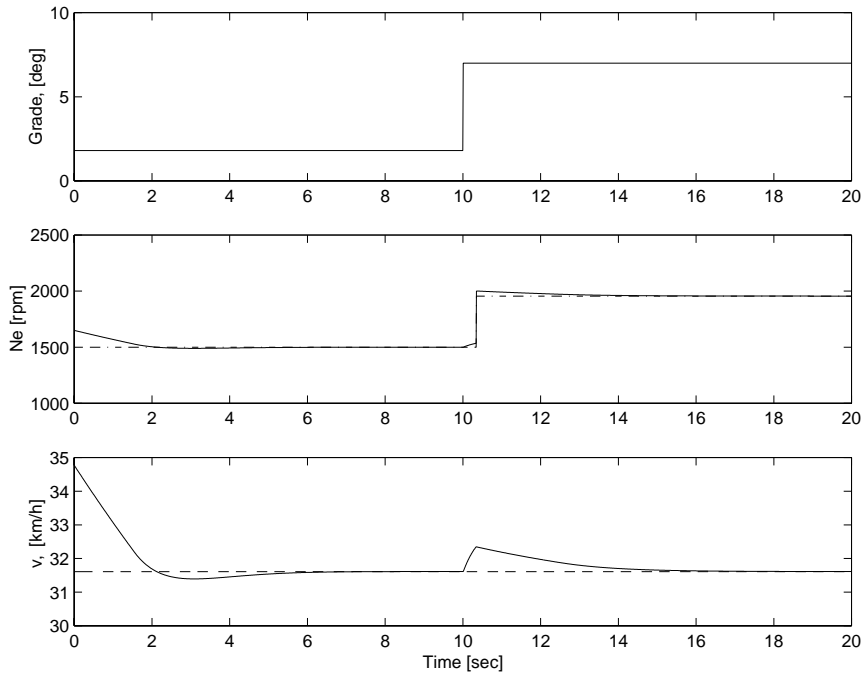
Figure 48: Controller responses to a disturbance due to road grade change from 1.8 to 7 degrees: trajectories of grade, engine speed, vehicle speed. The desired engine and vehicle speed are shown by the dashed lines.

7.2 Speed Control during Critical Maneuvers

In this section we address maneuvers that require aggressive braking as in cases of vehicle-following. We call these maneuvers critical maneuvers because the time necessary to achieve the desired speed is important/vital. This is the type of maneuvers that can be employed for collision avoidance.

The control design for critical maneuvers is based on the Speed-Gradient approach with the goal function appropriately modified by barrier functions to take into account the critical driving requirements.

As can be seen from Subsection 7.1, the compression brake coordinated with gear adjustments can be potentially used as the sole decelerating actuator without the assistance of service brakes during non-critical maneuvers. However, to perform critical maneuvers, the compression brake must be coordinated with service brakes to supplement the compression braking capability.

Recall that the braking with the compression brake is preferable, because we want to preserve the service brakes. Hence, we use the service brakes only when absolutely necessary. Specifically, if u_{eb} saturates, (i.e., $u_{eb} > u_{eb}^{max}$ or $u_{eb} < u_{eb}^{min}$) we calculate the torque deficit

$$\Delta TQ = TQ(\omega, u_{eb}) - TQ(\omega, sat(u_{eb})),$$

and deliver it with the service/friction brake, $F_{sb} = \frac{\Delta TQ}{r_g}$. Having made this convention, it is sufficient to consider the compression brake only with the idea that any extra braking effort required during the critical maneuver will be supplemented by the service brakes, according to the expression that we gave. Although our control design and analysis does not treat friction brake actuator dynamics or uncertainties (see Subsection 4.2 for details), we take them into account in all our simulations. Specifically, we used 0.2 sec as the time constant
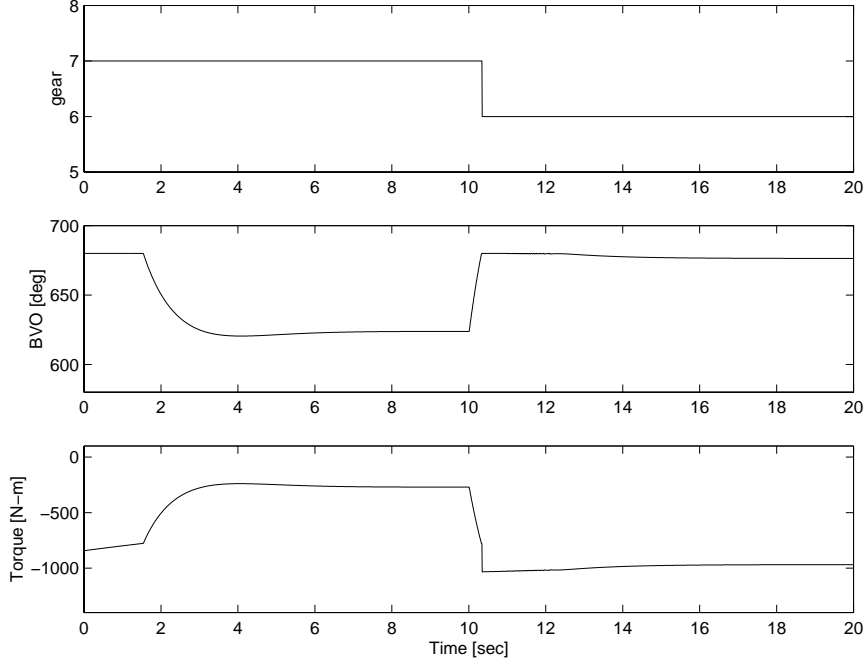
Figure 49: Controller responses to disturbance in road grade from 1.8 to 7 degrees: trajectories of gear ratio, BVO timing and compression torque.

of the service/friction brake actuator [40] and show that the control schemes maintain speed regulation without serious degradation in performance.

7.2.1 Aggressive Braking

In addition to speed regulation it is important, here, to ensure aggressive braking maneuvers when the difference between the current vehicle velocity, v, and the desired one, v_d, is sufficiently large, i.e., when $|v - v_d|$ does become greater then a given number $\varepsilon_1 > 0$. Assuming that the gear ratio r_g remains constant, the aggressive braking is needed when

$$|\omega - \omega_d| \geq \varepsilon,$$

where $\varepsilon = \varepsilon_1/r_g$, $\omega_d = v_d/r_g$. To capture the new requirement, the new objective function Q_1 has to include the nominal objective function $Q = \frac{J_t\gamma}{2}(\omega - \omega_d)^2$ and a smooth barrier function Φ_1 which is zero when $|\omega - \omega_d|$ is smaller than ε and is monotonically and rapidly increasing when $|\omega - \omega_d|$ is larger than ε:

$$Q_1 = \frac{J_t\gamma}{2}(\omega - \omega_d)^2 + \frac{J_t\gamma_1}{3}\Phi_1(\omega - \omega_d) \geq 0, \ \gamma_1 > 0,$$

where (see Figure 50)

$$\Phi_1(\omega - \omega_d) = \begin{cases} 0, & \text{if } \omega - \omega_d \leq \varepsilon \\ (\omega - \omega_d - \varepsilon)^3, & \text{if } \omega - \omega_d > \varepsilon \\ -(\omega - \omega_d + \varepsilon)^3, & \text{if } \omega - \omega_d < -\varepsilon \end{cases}$$

54
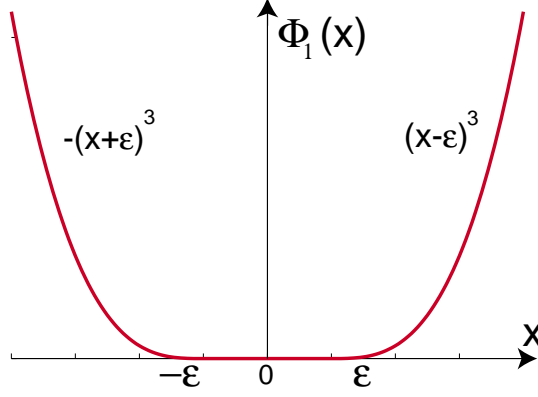
Figure 50: Barrier function for aggressive braking maneuver.

In this case the SG controller has the following form:

$$u_{eb} = u_d - k_p \nabla_{u_{eb}} \dot{Q}_1 - k_i \int_0^t \nabla_{u_{eb}} \dot{Q}_1(\omega(s)) ds,$$

where $k_p > 0$, $k_i > 0$ are the controller gains and

$$\nabla_{u_{eb}} \dot{Q}_1 = \begin{cases} \gamma(\omega - \omega_d)(\alpha_2 + \alpha_3\omega), & \text{if } \omega - \omega_d \leq \varepsilon \\[2mm] \left(\gamma(\omega - \omega_d) + \gamma_1(\omega - \omega_d - \varepsilon)^2 \right)(\alpha_2 + \alpha_3\omega), & \text{if } \omega - \omega_d > \varepsilon \\[2mm] \left(\gamma(\omega - \omega_d) - \gamma_1(\omega - \omega_d + \varepsilon)^2 \right)(\alpha_2 + \alpha_3\omega), & \text{if } \omega - \omega_d < -\varepsilon \end{cases}$$

If the $|\omega - \omega_d|$ falls outside the acceptable range $[-\varepsilon \ \varepsilon]$ then Φ_1 takes a large value and forces the controller to respond rapidly. Thus, this control design ensures that normally the speed control is accomplished with the compression brake only but if we need to brake suddenly the barrier function amplifies the braking action and potentially causes the service brakes to engage. In this critical maneuver both the compression brake and service brakes are coordinated to decelerate rapidly. A similar longitudinal speed control design which allows fast compensation for large errors was achieved in [46] by introducing a signed-quadratic (Q) term into the PI controller.

Figures 51, 52 illustrate the critical driving scenario with aggressive braking. The value of $\varepsilon_1 = 0.29$ m/sec (or 1.05 km/h) corresponds to $\varepsilon = 50$ rpm in the gear number seven. In Figure 51 we compare the engine and vehicle speed during aggressive control action with the engine and vehicle speed during nominal control action (without the barrier function). As can be seen, the response of the controller with the barrier function is much faster than that of the nominal design.
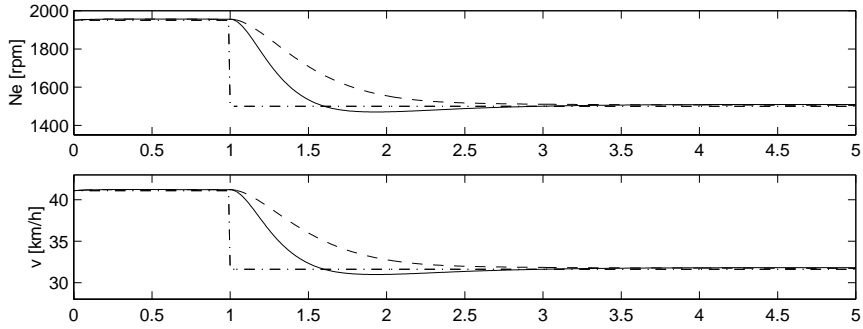
Figure 51: The engine and vehicle speed during aggressive control action (solid lines) and nominal control action (dashed lines). The desired engine and vehicle speed are shown by dash-dotted lines.
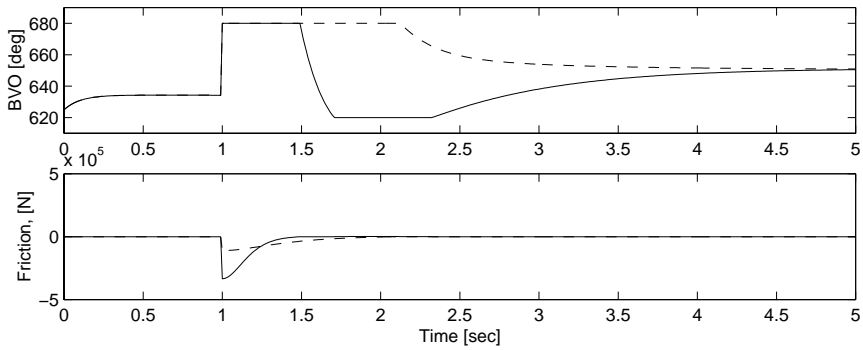
Figure 52: BVO timing and friction force during aggressive control action (solid lines) and nominal control action (dashed lines).

7.2.2 Vehicle Following: Collision Avoidance Case

We next study a problem where our vehicle follows a leading vehicle (also a truck). This is an important driving scenario in Automated Highway Systems (AHS) (see [37, 18, 3, 46]). We want to avoid any collisions between our vehicle and the leading truck. It means that we want to ensure that there is a sufficient distance between our vehicle and the vehicle in front of our vehicle. Let s be the position of our truck as it goes down the hill, so that $\dot{s} = v$, and s_l be the position of the leading vehicle as it goes down the hill.

The objective is then to always ensure that the separation distance (in seconds of travel) does not fall below a given number $\delta_1 \geq 0$

$$|\frac{s - s_l}{v}| \geq \delta_1. \tag{73}$$

As in previous Subsection, here we assume that the gear ratio r_g remains constant. Therefore, the objective (73) can be restated as

$$|\frac{s - s_l}{\omega}| \geq \delta,$$

where $\delta = \delta_1/r_g$ and the new objective function Q_2, which captures the new requirement (73) will include the nominal objective function $Q = \frac{J_t \gamma}{2}(\omega - \omega_d)^2$ and a smooth barrier

56

function Φ_2 that penalizes the small headway between the trucks in seconds, i.e.,

$$Q_2 = \frac{J_t\gamma}{2}(\omega - \omega_d)^2 + J_t\gamma_2\Phi_2(\frac{s - s_l}{\omega}) \geq 0, \ \gamma_2 > 0,$$

where ϕ_2 has to be zero when $|\frac{s-s_l}{\omega}|$ is larger than δ and monotonically and rapidly increasing when $|\frac{s-s_l}{\omega}|$ is smaller than δ.

Because of $s - s_l < 0$ (since our truck follows the leading vehicle), the function Φ_2 can be introduced as follows:

$$\Phi_2(\frac{s - s_l}{\omega}) = \begin{cases} -1 - \delta\frac{\omega}{s-s_l}, & \text{if } \frac{s-s_l}{\omega} < -\delta \\ 0, & \text{otherwise}, \end{cases}$$

where δ is the minimum headway distance allowed between the trucks (see Figure 53).
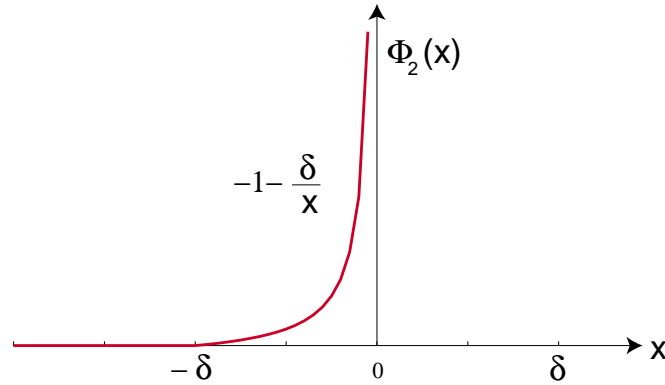
Figure 53: Barrier function for "vehicle-following" maneuver.

Then the SG-PI controller has the following form:

$$u_{eb} = u_d - k_p\nabla_{u_{eb}}\dot{Q}_2 - k_i\int_0^t \nabla_{u_{eb}}\dot{Q}_2(\omega(s))ds,$$

where $k_p > 0$, $k_i > 0$ are the controller gains and

$$\nabla_{u_{eb}}\dot{Q} = \begin{cases} (\gamma(\omega - \omega_d) - \gamma_2\frac{\delta}{s-s_l})(\alpha_2 + \alpha_3\omega), & \text{if } \frac{s-s_l}{\omega} < -\delta \\ \\ \gamma(\omega - \omega_d)(\alpha_2 + \alpha_3\omega), & \text{otherwise}. \end{cases}$$

This control design ensures that normally the speed control is accomplished with the compression brake but if $|\frac{s-s_l}{\omega}|$ becomes smaller than δ, a high gain braking action is produced and both the compression brake and service brake are engaged to prevent the collision.

The idea of the simulation scenario is that the lead vehicle decelerates to $0.5v_d$ at $t = 5$ seconds and then accelerates again to v_d at $t = 10$ seconds. The minimum distance is $\delta = 10$ seconds (corresponding to $\delta_1 = 0.56$) is allowed. The responses are shown in Figures 54, 55. Note the aggressive braking action that the controller uses to prevent the collision with a decelerating leading vehicle.
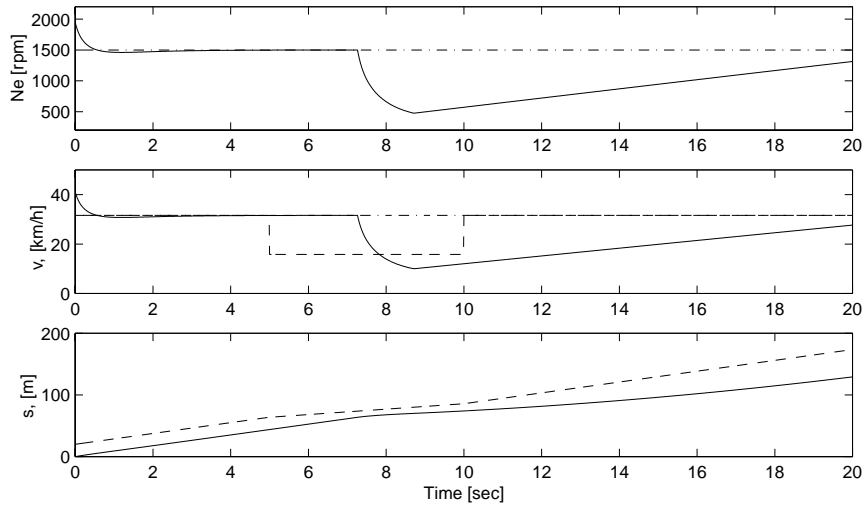
Figure 54: The engine speed, vehicle speed, vehicle position during vehicle-following maneuver (solid lines). The dash-dotted line shows the desired engine and vehicle speeds while the dashed lines show the vehicle and position trajectory of the leading vehicle.
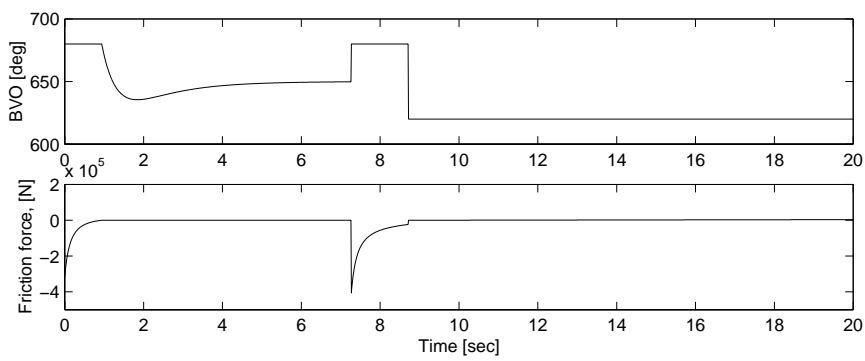
Figure 55: BVO timing and friction force during vehicle-following maneuver.

8   Concluding Remarks

The key issues, research accomplishments, and directions for future work are summarized as follows:

1) Model Development

Derivation of a novel control-oriented engine model able to represent transition between combustion and braking mode, and the dynamical interactions between engine and CHV longitudinal subsystems. The key issues are:

- Integration of different time scales such as crank angle dynamics (10^{-4} sec), manifold filling dynamics (10^{-1} sec), and vehicle dynamics (10 sec) in a nonlinear, multivariable dynamical model.

- Low order mean-value engine model is derived by applying the technique of signal processing and input-output identification of step input responses, using crank angle simulation data. This work bridges the gap between the detailed crankangle based models developed in the engine design community, and the low order representation of engine torque response used in the vehicle dynamics community.

- Validation of the developed models using measurements obtained from an engine-dynamometer facility.

2) Sensitivity and Control Analysis

Analysis of the dynamics shed light into the critical transient phase that quasi-steady equilibrium analysis typically conceals.

- Analysis of the detrimental consequences of the actuator delay and saturation levels for the stability of a CHV. Even if a stable equilibrium for descending speed exists for a given grade, vehicle mass, rolling resistance, and large actuator delays can lead to maximum engine speed ("runaway").

- Characterization of the relative control authority for the redundant retarding actuators, and their relative importance in critical braking maneuvers. Nowadays, the control allocation is primarily resolved by predictive actions of trained drivers.

- CHVs operate under considerable changes in vehicle parameters, such as gear selection, vehicle mass, aerodynamic drag, rolling resistance, etc. Some of these parameters are unknown and unmeasured, and introduce large uncertainties in the longitudinal control system.

3) Control Design

Retarding power and retarding control are critical in accommodating higher operational speed and acceleration performance of modern CHVs. They also are fundamental requirements in achieving increased highway capacity and enhanced driving safety which are the major goals of Automated Highway Systems (AHS). The use of compression braking in coordination with the service brakes increases the overall retarding power of the vehicle and lowers maintenance costs on the service brakes. The compression brake can be used continuously without danger of damage and overheating and it is, thus, a natural actuator to be used for speed control.

- We first design a linear PI controller that emulates the driver's actions to avoid "runaways" during descends on grades. The controller simply uses the engine speed measurement to activate the service brakes only when needed. Controller gains are tuned to maintain nominal performance and robust stability during operations in braking and transition modes.

- To control effectively the large signal behavior of the system we then develop nonlinear Speed-Gradient controllers to accomplish both non-critical and critical longitudinal speed control maneuvers, including vehicle following. Two ways to handle the uncertainty in the road grade have been explored, one through the use of an integral action of the SG-PI controller for constant (but unknown) grade, and another one through the differential action of the SG-PD controller for time-varying grade. The uncertainty in the aerodynamic coefficient has been also addressed. For large grade or desired vehicle speed changes, we proposed a controller that coordinates the compression brake with the gear ratio adjustment for non-critical maneuvers and also with the service brakes during critical maneuvers.

9 Appendices

9.1 Nomenclature

In the following table, the symbol † indicates a variable where x is replaced with: f for front exhaust manifold, r for rear exhaust manifold and c for collector exhaust manifold.

β	road grade (^{o})
η_c	compressor efficiency
η_t	turbine efficiency
γ	specific heat ratio of air
μ	rolling resistance
ω	engine speed (rad/sec)
ψ	standard orifice flow function
ρ	air density (kg/m^3)
θ	crank angle (^{o})
A	frontal area of the truck (m^2)
B	cylinder bore (m)
BVT	see v_{eb}
C_d	drag coefficient
c_p	specific heat capacity for constant pressure (J/kg K)
c_v	specific heat capacity for constant volume (J/kg K)
C_q	quadratic resistive coefficient
EM	exhaust manifold
F_β	gravity force due to grade (N)
F_{sb}	friction brake force (N)
F_{qdr}	aerodynamic force (N)
F_r	rolling resistance force (N)
g_f	differential gear ratio
g_t	transmission gear ratio
IM	intake manifold
I_{tc}	mass polar moment of inertia (Nm sec^2)
J_e	engine inertia (Nm sec^2)
J_t	total vehicle inertia (Nm sec^2)
m_{cyl_j}	mass, cylinder j (kg)
m_{e_x} †	mass, EM (kg)
N_e	engine speed (RPM)
$N_t c$	turbocharger speed (RPM)
p_{cyl_j}	pressure, cylinder j (Pa)
p_{e_x} †	pressure, EM (Pa)
p_0	ambient pressure (Pa)
P_c	compressor power (W)
P_t	turbine power (W)
Q_{ht}	heat transfer (J/sec)
Q_{lhv}	lower heating value for diesel fuel (J/kg)
r	crank radius (m)
R	gas constant (J/kg K)
r_ω	wheel radius (m)

r_c	compression ratio
r_g	total rear ratio
T_{cyl_j}	temperature, cylinder j (K)
T_{e_x} †	temperature, EM (K)
t_{sb}	service brake time constant (sec)
T_0	ambient temperature (K)
T_c	compressor temperature (K)
T_i	temperature, IM (K)
TQ	engine shaft torque (Nm)
TQ_{cyl_j}	torque contribution from cylinder j (Nm)
u_{sb}	service brake pedal displacement
v	vehicle speed (m/sec)
V_{cyl_j}	volume, cylinder j (m^3)
$\dot{V}_{cyl_j}$	rate of change in volume, cylinder j (m^3/sec)
V_{e_x} †	volume EM (m^3)
V_{e_x} †	volume, EM (m^3)
v_{eb}	brake valve timing $(^o)$
v_{sb}	air pressure (Pa)
v_f	fuel flow (kg/sec)
V_i	volume IM (m^3)
W^B	unified engine signal corresponding to brake valve timing
W^F	unified engine signal corresponding to fuel flow
W_{afb_j}	apparent fuel burn rate, cylinder j (kg/sec)
$W_{cyl_je_x}$ †	mass air flow from cylinder j into EM (kg/sec)
W_{cyl_ji}	mass air flow from cylinder j into IM (kg/sec)
$W_{e_ce_f}$	mass air flow from collector to front EM (kg/sec)
$W_{e_ce_r}$	mass air flow from collector to rear EM (kg/sec)
$W_{e_fe_c}$	mass air flow from front to collector EM (kg/sec)
$W_{e_re_c}$	mass air flow from rear to collector EM (kg/sec)
$W_{e_xcyl_j}$ †	mass air flow EM into cylinder j (kg/sec)
W_{fb_j}	fuel burned rate, cylinder j (kg/sec)
W_{icyl_j}	mass air flow from IM into cylinder j (kg/sec)
W_{icyl_j}	mass air flow from IM into cylinder j (kg/sec)
W_c	compressor mass flow (kg/sec)
W_t	turbine mass flow (kg/sec)

9.2 Parameterization of Reduced Engine Model

Based on standard regression techniques, we employ the following parameterizations for engine torque (Nm), and time delays and zeros (all in seconds) for the reduced engine model described in Section 5.11:

Braking Mode:

$$TQ = \mathcal{P}_{TQ}(\tilde{\omega}(t), W(t))$$
$$= -[1, x_1, x_2, x_1 \cdot x_2] \, \vec{a_1}$$
$$\vec{a_1} = \begin{pmatrix} 1.893010866200470 \cdot 10^3 \\ -5.041142241925328 \\ -2.858890575907517 \\ 8.210279510665771 \cdot 10^{-3} \end{pmatrix}$$
$$\tau = \mathcal{P}_{B1}(\omega_{nom}, W_{nom}^B)$$
$$= [1, x_2, x_1, x_1 \cdot x_2, x_2^2] \, \vec{a_2}$$
$$\vec{a_2} = \begin{pmatrix} 1.435144306531251 \cdot 10^2 \\ -3.784002747703378 \cdot 10^{-1} \\ -1.468095754322008 \cdot 10^{-2} \\ 2.044443841780719 \cdot 10^{-5} \\ 2.501359693951508 \cdot 10^{-4} \end{pmatrix}$$
$$c = \mathcal{P}_{B2}(\omega_{nom}, W_{nom}^B)$$
$$= [1, x_2, x_1, x_1 \cdot x_2, x_2^2] \, \vec{a_3}$$
$$\vec{a_3} = \begin{pmatrix} 7.075201422653511 \cdot 10^1 \\ -1.649263714596686 \cdot 10^{-1} \\ -1.310740126225251 \cdot 10^{-2} \\ 1.860622742042733 \cdot 10^{-5} \\ 9.176575813120126 \cdot 10^{-5} \end{pmatrix}$$
$$\tau_\omega = \mathcal{P}_{B3}(\omega_{nom}, W_{nom}^B)$$
$$= [1, x_1, x_2, x_1 \cdot x_2, x_1^2] \, \vec{a_4}$$
$$\vec{a_4} = \begin{pmatrix} 2.497098890622600 \cdot 10^1 \\ -9.107643541881476 \cdot 10^{-3} \\ -3.734043775951978 \cdot 10^{-2} \\ 1.601859896099090 \cdot 10^{-5} \\ -6.870876942538606 \cdot 10^{-7} \end{pmatrix}$$
$$c_\omega = \mathcal{P}_{B4}(\omega_{nom}, W_{nom}^B)$$
$$= [1, x_1, x_2, x_1 \cdot x_2, x_1^2] \, \vec{a_5}$$
$$\vec{a_5} = \begin{pmatrix} 1.267328293029872 \cdot 10^1 \\ -6.585733957889821 \cdot 10^{-3} \\ -1.711384794384464 \cdot 10^{-2} \\ 8.225198567910289 \cdot 10^{-6} \\ 2.486366283519799 \cdot 10^{-7} \end{pmatrix}$$
$$\tau_a = \mathcal{P}_{B5}(\omega_{nom}, W_{nom}^B)$$
$$= 10 \cdot 10^{-3}$$

Combustion Mode:

$$TQ = \mathcal{P}_{TQ}(\tilde{\omega}(t), W(t))$$
$$= [1, x_1, x_3, x_1 \cdot x_3] \, \vec{b_1}$$
$$\vec{b_1} = \begin{pmatrix} -5.844338623392951 \cdot 10^2 \\ 4.595064022923466 \cdot 10^{-1} \\ 2.968538926767046 \cdot 10^5 \\ -1.220340922356248 \cdot 10^2 \end{pmatrix}$$
$$\tau = \mathcal{P}_{F1}(\omega_{nom}, W_{nom}^F)$$
$$= [1, x_3] \, \vec{b_2}$$
$$\vec{b_2} = \begin{pmatrix} -6.499999999999687 \cdot 10^{-3} \\ 1.499999999999980 \end{pmatrix}$$
$$c = \mathcal{P}_{F2}(\omega_{nom}, W_{nom}^F)$$
$$= 0$$
$$\tau_\omega = \mathcal{P}_{F3}(\omega_{nom}, W_{nom}^F)$$
$$= [1, x_3, x_1 x_3] \, \vec{b_4}$$
$$\vec{b_4} = \begin{pmatrix} 2.125000000000203 \cdot 10^{-3} \\ 1.126766417290105 \cdot 10^{-5} \\ -7.834580216126365 \cdot 10^{-4} \end{pmatrix}$$
$$c_\omega = \mathcal{P}_{F4}(\omega_{nom}, W_{nom}^F)$$
$$= 0$$
$$\tau_a = \mathcal{P}_{F4}(\omega_{nom}, W_{nom}^F)$$
$$= 10 \cdot 10^{-3}$$

Scaling:

$$x_1 = \frac{30}{\pi} \cdot \omega_{nom}$$
$$x_2 = -0.8 \cdot W_{nom}^B + 620$$
$$x_3 = 14.25 \cdot 10^{-5} \cdot W_{nom}^F$$

9.3 Speed-Gradient Methodology

In this section we review the necessary results of the Speed-Gradient (SG) Control Methodology [10, 11]. Consider a nonlinear system model of the form

$$\dot{x} = f(x) + g(x)u, \tag{74}$$

where $x \in R^n$ is the state vector, $u \in R^m$ is the control input vector, $f(x)$ and $g(x)$ are continuously differentiable vector-functions.

The control design objective is to stabilize to a desired equilibrium $x = x_d$ (that satisfies $f(x_d) + g(x_d)u_d = 0$) while at the same time shaping the transient response via the minimization of the following scalar goal function

$$Q(x(t)) \to 0, \ when \ t \to \infty, \tag{75}$$

where $Q(x)$ is assumed to be twice continuously differentiable and radially unbounded function that satisfies $Q(x) \geq 0$, $Q(x_d) = 0$. The function Q may, for example, represent a weighted sum of the squares of the deviations of the different components of x from the corresponding components of x_d.

We first present an intuitive argument leading to the derivation of the SG controller. Consider the evolution of $Q(x(t))$ over a sufficiently small time interval $[t, \ t + \Delta t]$. Then, the objective of minimizing Q can be restated as

$$Q(t + \Delta t) \approx Q(t) + \xi(x(t), u(t))\Delta t \to \min,$$

where the function $\xi(x, u)$ is determined as a time derivative of $Q(x)$ along the trajectories of the system (74) (i.e. the speed of change of Q):

$$\xi(x, u) = \dot{Q} = \frac{\partial Q}{\partial x}(f(x) + g(x)u).$$

To prevent large control excursions from the desired steady-state value, u_d, we can augment a control penalty and consider the minimization of the following function for $\Pi > 0$:

$$Q(t) + \xi(x(t), u)\Delta t + \frac{1}{2}(u - u_d)^{\mathrm{T}}(\frac{\Pi}{\Delta t})^{-1}(u - u_d) \to \min_u. \tag{76}$$

Since $\xi(x(t), u)$ is affine in u the minimizer is obtained by setting the gradient with respect to u to zero. This leads to the controller

$$u(t) = u_d - \Pi\Psi(x), \tag{77}$$

where Ψ is the gradient of the "speed" $\dot{Q} = \xi(x(t), u)$ with respect to u:

$$\Psi(x) \triangleq \nabla_u \xi(x(t), u) = (\frac{\partial Q}{\partial x}g(x))^{\mathrm{T}}. \tag{78}$$

This controller is referred to as the Speed-Gradient Proportional (SG-P) controller. One can also augment a penalty on the control increment and consider the minimization of

$$Q(t) + \xi(x(t), u)\Delta t + \frac{1}{2}(u(t) - u(t - \Delta t))^{\mathrm{T}}\Gamma^{-1}(u(t) - u(t - \Delta t)) \rightarrow \min_{u(t)}, \ \Gamma > 0. \tag{79}$$

This results in the Speed-Gradient Integral (SG-I) controller:

$$\dot{u}(t) \approx \frac{u(t) - u(t - \Delta t)}{\Delta t} = -\Gamma\nabla_u\xi(x(t), u) = -\Gamma\Psi(x). \tag{80}$$

The general class of controllers of interest for this paper are Speed-Gradient Proportional plus Integral (SG-PI) controllers of the form:

$$u(t) = u_d - \Pi\Psi(x(t)) - \Gamma\int_0^t \Psi(x(s))ds. \tag{81}$$

where $\Pi = \Pi^T > 0$ and $\Gamma = \Gamma^T > 0$ are symmetric positive definite matrices (usually diagonal). In general, there is no guarantee that the controller (81) results in the stable closed loop system and is robust to disturbances. However, one may provide stability and robustness properties under some sufficient stability conditions as reviewed next.

We start by rewriting the control law (81) in a more convenient equivalent form:

$$\begin{aligned} u &= u_d - \Pi\Psi(x) + \theta, \\ \dot{\theta} &= -\Gamma\Psi(x) \end{aligned} \tag{82}$$

where θ is the integrator state. Let us consider the following Lyapunov function

$$V(x, \theta) = Q(x) + \frac{1}{2}\theta^{\mathrm{T}}\Gamma^{-1}\theta \geq 0 \tag{83}$$

and calculate its time-derivative along the trajectories of the closed loop system (74), (82):

$$\begin{aligned} \dot{V} &= \frac{\partial Q}{\partial x}(f(x) + g(x)u_d - g(x)\Pi\Psi(x) + g(x)\theta) - \theta^{\mathrm{T}}\Gamma^{-1}\Gamma\Psi(x) \\ &= \frac{\partial Q}{\partial x}(f(x) + g(x)u_d) - \Psi^{\mathrm{T}}(x)\Pi\Psi(x). \end{aligned} \tag{84}$$

Now, let us determine the following sets:

$$\Upsilon_C \triangleq \{x : \ Q(x) \leq C\},$$

$$\Omega_C \triangleq \{(x, \theta) : \ V(x, \theta) \leq C\},$$

and suppose that the so called *achievability condition* holds:

$$\frac{\partial Q}{\partial x}(f(x) + g(x)u_d) \leq -\rho(Q(x)) \quad \text{for all } x \in \Upsilon_C, \tag{85}$$

where ρ is a continuously differentiable function that satisfies

$$\rho(0) = 0, \ \rho(z) > 0 \ if \ z \neq 0.$$

Since the achievability condition holds for $x(t) \in \Upsilon_C$, then $\dot{V}(t) \leq 0$ as long as $x(t) \in \Upsilon_C$. Assume that the initial condition at time $t = 0$ is $(x(0), \theta(0))$ such that $(x(0), \theta(0)) \in \Omega_C$, i.e. $x(0), \theta(0)$ satisfy the following inequalities:

$$Q(x(0)) \leq C \cdot \lambda,$$

$$\frac{1}{2}\theta(0)^{\mathrm{T}}\Gamma^{-1}\theta(0) \leq C \cdot (1 - \lambda), \ 0 \leq \lambda \leq 1.$$

Then, for all t, $V(x(t), \theta(t)) \leq V(x(0), \theta(0)) \leq C$ and $Q(x(t)) \leq C$ so that the achievability condition holds on the trajectory $x(t, x(0), \theta(0)), \theta(t, x(0), \theta(0))$. Therefore, $V(x(t), \theta(t)$ and $Q(x(t))$ are bounded and the closed-loop system trajectories $x(t), \theta(t)$ are bounded as well due to radial unboundness of $Q(x)$.

Since the function $V(x(t), \theta(t))$ is nonincreasing and bounded from below by zero, it has a limit V_∞. Integrating the differential inequality (84) over an interval $[0, t]$ we obtain:

$$\lim_{t \to \infty} \int_0^t \rho(Q(x(\tau)))d\tau \leq -\lim_{t \to \infty} \int_0^t \dot{V}(x(\tau), \theta(\tau))d\tau =$$
$$= \lim_{t \to \infty} \{V(x(0), \theta(0)) - V(x(t), \theta(t))\} = V(x(0), \theta(0)) - V_\infty \tag{86}$$

which implies $\int_0^\infty \rho(Q(x(s)))ds < \infty$. Then, taking into account boundedness of $x(t)$, $\theta(t)$, $u(t)$ and boundedness of $f(x)$ and $g(x)$ on any compact set we get that $\dot{x}(t)$ is bounded and, therefore, $x(t)$ is uniformly continuous in t. Further since $\rho(Q)$ is continuous in Q and $x(t)$ is uniformly continuous in t, then $\rho(Q(x(t)))$ is uniformly continuous in t. Then from the Barbalat's lemma we obtain that $Q(x(t)) \to 0$ as $t \to \infty$. Additionally, let us assume that $Q(x_d) = 0$, $Q(x) > 0$ for $x \neq x_d$. Then, it follows that $x \to x_d$.

Hence, the above facts prove the following result.

Theorem 1: *Consider the SG-PI controller (82) applied to the system (74). Assume that the achievability condition (85) holds for all $x \in \Upsilon_C$. Then for all initial conditions $(x(0), \theta(0))$ in Ω_C the closed loop trajectories satisfy*

$$\lim_{t \to \infty} Q(x(t)) = 0.$$

Moreover, if $Q(x)$ satisfies $Q(x_d) = 0$ and $Q(x) > 0$ for $x \neq x_d$, the closed loop system meets the control objective $\lim_{t \to \infty} x(t) = x_d$.

Remark 1: The set $\Omega_C = \{(x, \theta) : Q(x) + \frac{1}{2}\theta^{\mathrm{T}}\Gamma^{-1}\theta \leq C\}$ is a region of attraction of the equilibrium $(x_d, 0)$. Typically, $\theta(0)$ is set to zero, and then all initial states $x(0) \in \Upsilon_C = \{x : Q(x) \leq C\}$ are guaranteed to be recoverable by the controller (82).

Remark 2: The same result can be proved for the SG-P controller (77). Indeed, in this case the Lyapunov function V coincides with the objective function $Q(x)$ and the region of attraction is the set Υ_C.

The vector u_d in the SG-P controller (77) and SG-PI controller (82) can be interpreted as an ideal feedforward term: $f(x_d) + g(x_d)u_d = 0$. Due to plant parameter variations, u_d may be unknown. However, in the case of the SG-PI controller (since

the controller employs an integral action), we expect some robustness properties to disturbances that are additive to the plant input. Let w be an unknown constant additive disturbance affecting the plant input. Using w to represent the error in the feedforward term, the controller then can be viewed as applying an erroneous feedforward $\tilde{u}_d$ in the form $\tilde{u}_d = u_d + w$. Thus, the SG-PI controller can be represented as

$$
\begin{aligned}
u &= u_d - \Pi\Psi(x) + \theta + w, \\
\dot{\theta} &= -\Gamma\Psi(x),
\end{aligned}
\tag{87}
$$

where θ can be interpreted as an estimate of $-w$. The desirable property $\theta(t) \to -w$, and, therefore, $\tilde{u}_d + \theta(t) \to u_d$ means that the integrator state corrects for the error in the feedforward asymptotically.

Consider the following Lyapunov function

$$
\tilde{V}(x,\theta) = Q(x) + \frac{1}{2}(\theta + w)^{\mathrm{T}}\Gamma^{-1}(\theta + w)
\tag{88}
$$

and determine the set:

$$
\tilde{\Omega}_C \triangleq \{(x,\theta) : \ \tilde{V}(x,\theta) \leq C\}.
$$

Theorem 2: *Consider the SG-PI controller (87) applied to the system (74). Assume that the achievability condition (85) holds for all $x \in \Upsilon_C$. Then for all the initial conditions $(x(0), \theta(0))$ in $\tilde{\Omega}_C$ the closed loop trajectories satisfy:* $\lim_{t\to\infty} Q(x(t)) = 0$. *Moreover, if $Q(x)$ satisfies $Q(x_d) = 0$ and $Q(x) > 0$ for $x \neq x_d$, the closed-loop system meets the control objective $\lim_{t\to\infty} x(t) = x_d$. If, furthermore, the $n \times m$ matrix $g(x_d)$ has a full column rank m, then $\lim_{t\to\infty} \theta(t) = -w$, $\lim_{t\to\infty}(\tilde{u}_d + \theta(t)) = u_d$.*

The proof of the theorem is based on calculating the time-derivative of the Lyapunov function (88) along the trajectories of the closed loop system (74),(87):

$$
\begin{aligned}
\dot{V} &= \frac{\partial Q}{\partial x}(f(x) + g(x)u_d - g(x)\Pi\Psi(x) + g(x)(\theta + w)) - (\theta + w)^{\mathrm{T}}\Gamma^{-1}\Gamma\Psi(x) \\
&= \frac{\partial Q}{\partial x}(f(x) + g(x)u_d) - \Psi^{\mathrm{T}}(x)\Pi\Psi(x).
\end{aligned}
$$

Since the achievability condition holds for $x(t) \in \Upsilon_C$, then $\dot{V}(t) \leq 0$ as long as $x(t) \in \Upsilon_C$. Suppose that the initial condition at time $t = 0$ is $x(0)$ such that

$$
Q(x(0)) \leq c \cdot \lambda, \ 0 \leq \lambda \leq 1,
\tag{89}
$$

while $\theta(0)$ (the initial "estimate" of $-w$) yields

$$
\frac{1}{2}(\theta(0) + w)^{\mathrm{T}}\Gamma^{-1}(\theta(0) + w) \leq C \cdot (1 - \lambda).
\tag{90}
$$

Then, for all t, $V(x(t), \theta(t)) \leq V(x(0), \theta(0)) \leq C$ and $Q(x(t)) \leq C$ so that the achievability condition holds on the trajectory $x(t, x(0), \theta(0))$, $\theta(t, x(0), \theta(0))$. Applying the Barbalat's lemma to

$$
\dot{V}(t) \leq -\rho(Q(x)) - \Psi^{\mathrm{T}}(x)\Pi\Psi(x) \leq 0
$$

one can prove that

$$\lim_{t \to \infty} \Psi(x(t)) = \lim_{t \to \infty} (\frac{\partial Q}{\partial x} g(x(t)))^{\mathrm{T}} = 0,$$

and that

$$\lim_{t \to \infty} Q(x(t)) = 0.$$

Moreover, if $Q(x)$ satisfies $Q(x_d) = 0$ and $Q(x) > 0$ for $x \neq x_d$, then

$$\lim_{t \to \infty} x(t) = x_d.$$

The fact that $\Psi(t) \to 0$, $Q(t) \to 0$, in general, does not guarantee that $\theta(t) \to -w$. Additional analysis is needed to establish this convergence. The convergence $\theta \to -w$ can be assured if the $n \times m$ matrix $g(x_d)$ has a full column rank. To show this we apply the Barbalat's lemma to $\dot{x}$, to obtain $\dot{x} \to 0$. From the closed loop system equations

$$\dot{x} = (f(x) + g(x)u_d) - g(x)\Pi\Psi(x) + g(x)(\theta + w),$$

and $\Psi(x) \to 0$, $x \to x_d$ it follows that $g(x_d)(\theta(t) + w) \to 0$ and, hence, $\theta(t) \to -w$.

Remark 3: The requirement of twice continuous differentiability of Q is only needed to guarantee that $\dot{Q}$ is uniformly continuous. The latter property allows us to apply Barbalat's lemma to prove convergence. It might be, however, technically difficult to use a twice continuously differentiable goal function (e.g., see Section 7.2). In the case when Q does not satisfy this requirement, one should check the uniform continuity of $\dot{Q}$ via a direct argument.

Remark 4: The set $\tilde{\Omega}_C = \{(x, \theta) : Q(x) + \frac{1}{2}(\theta + w)^{\mathrm{T}}\Gamma^{-1}(\theta + w) \leq C\}$ describes the set of initial conditions for which the closed loop system trajectories are assured to meet the control objective (75). Although it is advantageous to have an initial estimate of $-w$, $\theta(0)$, as close as possible to $-w$, we typically set $\theta(0)$ to zero, because w is unknown. Then, the set of initial states $x(0)$ that are guaranteed to be recoverable by the controller (82), decreases when w increases.

Remark 5: To check the achievability condition (85) the following procedure is used. Assume that $\Upsilon_C = \{x : Q(x) \leq C\}$ for some $C > 0$ is a compact set with x_d in its interior and $Q(x) > 0$ if $x \neq x_d$, $Q(x_d) = 0$. We need to find a value of $\tilde{C}$ such that for all $x \in \Upsilon_{\tilde{C}} = \{x : Q(x) \leq \tilde{C}\}$ the strong achievability condition

$$\dot{Q}(x, u_d) = \frac{\partial Q}{\partial x}(f(x) + g(x)u_d) \leq -\rho Q(x), \tag{91}$$

where $\rho > 0$, holds. Essentially, ρ is a low bound on a rate of convergence of $Q(x(t))$ to zero on the trajectories of the open-loop system.

Let us define the following function

$$\kappa(C) \triangleq \max_{\Upsilon_C} \left(\frac{\dot{Q}(x, u_d)}{Q(x)} \right). \tag{92}$$

Note that $\kappa(C)$, in general, may take an infinite value since $Q(x_d) = 0$. On the other hand, $\dot{Q}(x_d, u_d) = 0$ and, hence, $\frac{\dot{Q}}{Q}$ may have a removable singularity at 0 and we can, therefore, set $\dot{Q}(x_d, u_d)/Q(x_d) = \lim_{\|x\| \to x_d} \frac{\dot{Q}(x,u_d)}{Q(x)}$. In this case $\kappa(C)$ takes a finite value due to the compactness of Υ_C. The case that $\frac{\dot{Q}}{Q}$ has a removable singularity at x_d is, actually, rather usual in many applications. Moreover, $\kappa(C)$ is nondecreasing in C. The value of $\kappa(C)$ can be calculated using numerical optimization. From the graph of $\kappa(C)$ we may be able to specify $\tilde{C} > 0$ such that $\kappa(C) < 0$ for all $C \leq \tilde{C}$. Then, $\dot{Q}(x, u_d) \leq -\rho Q(x)$ as long as $Q(x) \leq \tilde{C}$, i.e., the strong achievability condition (91) holds.

Remark 6: We emphasize that achievability conditions are only sufficient stability conditions; the actual domain of attraction may be much larger than the sets Ω_C and $\tilde{\Omega}_C$. These stability conditions, however, place no restriction on the controller gains Π and Γ as long as they are positive definite matrices and, therefore, allow a considerable freedom in the selection of the controller.

Remark 7: The Speed-Gradient Methodology is related to other constructive nonlinear design techniques, for example, those based on Control Lyapunov Functions (CLF) methods and $L_g V$-techniques [35]. For affine in control systems, the differences are mainly in the approach: Q is selected by the designer to capture the performance objectives in the SG approach; Q is constructed as a Control Lyapunov Function in the other methodologies. The strength of SG approach is in the strong linkage between control objectives and the selection of the goal function Q. The weakness of SG approach is that if achievability conditions with the particular function Q do not hold the procedures to modify Q are not readily available. Note that the achievability conditions are only sufficient (see Remark 6) and the stability may be verified by other procedures.

9.4 Source code for full nonlinear diesel engine simulator

The simulation model has been implemented in a Simulink/Matlab environment using C-coded S-functions. The model block diagram is shown in Figures 56, 57 followed by the attached C-coded S-function.
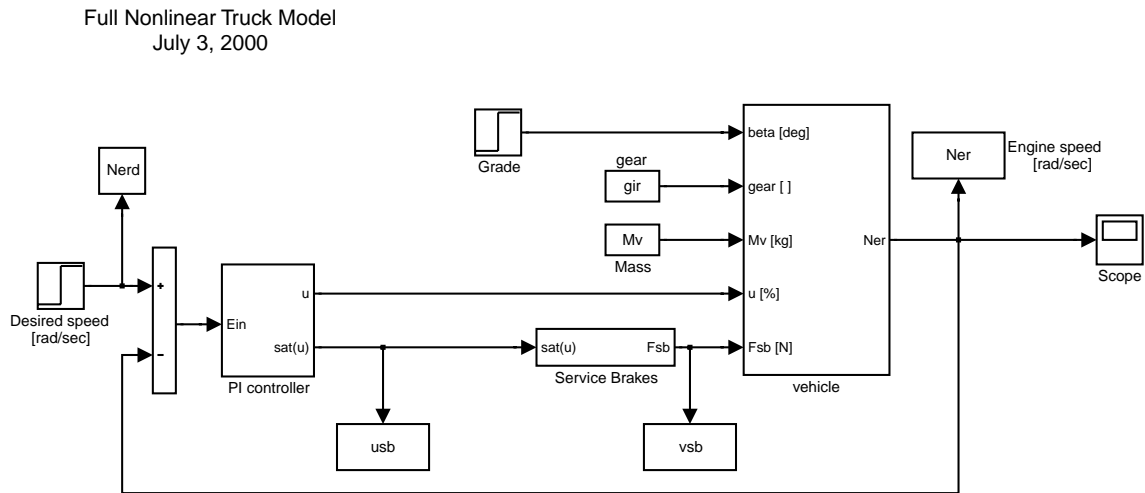
Figure 56: The block-diagram of the Full Order Nonlinear Model.
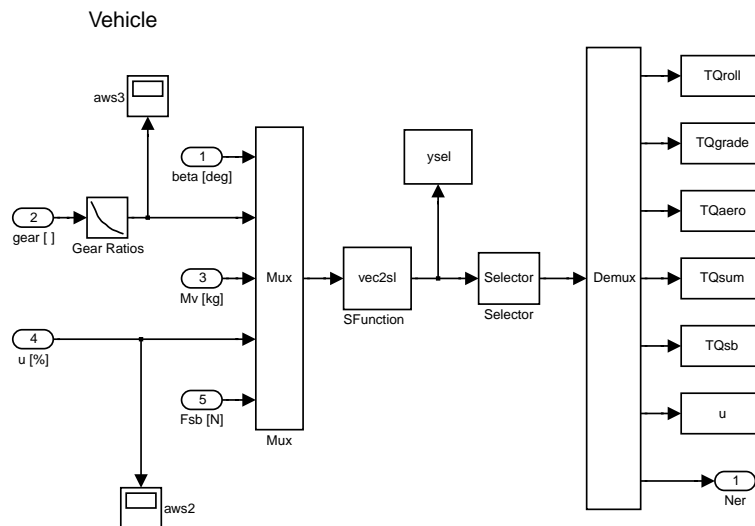
Figure 57: Engine and vehicle dynamics subsystem

70

```
/*===========================================================
 * vec.c --> 06-09-00
 * Source code for full nonlinear diesel engine simulator
 *
 * Developed by:
 * L. Moklegaard, A. Stefanopoulou, and F. Loquasto
 * Department of Mechanical and Environmental Engineering
 * University of California, Santa Barbara
 *
 * For more info, email annastef@umich.edu
 *
 * Original etech.c code augmented with vehicle dynamics
 *
 ===========================================================*/

#include "stdio.h"
#include "stdlib.h"
#include "math.h"

/********************************
 * Vector Sizes
 ********************************/
#define NU          5               /* Number of INPUTS  */
#define NX          24              /* Number of STATES  */
#define NY          23              /* Number of OUTPUTS */

/********************************
 * Useful Macros
 ********************************/
#define pi 3.14159265368979
#define pow2(a) ((a) * (a))
#define pow3(a) ((a) * pow2(a))
#define pow4(a) ((a) * pow3(a))
#define pow5(a) ((a) * pow4(a))
#define pow6(a) ((a) * pow5(a))
/*
#define min(a,b) ( (a)<(b) ? (a) : (b))
#define max(a,b) ( (a)>(b) ? (a) : (b))
*/

/*===============================================================================*
 * Vehicle Parameters
 *===============================================================================*/
#define Av   10.03         /*[m^2], Frontal Area                                 */
#define g    9.81          /*[m/sec^2], Gravity                                  */
#define rw   0.512         /*[m], Wheel radius                                   */
#define gd   4.28          /*[], Axle ratio (differential) gd=4.28 => Vveh=70mph @ 2500rpm */
#define Ie   2.82          /*[Nm*sec^2], Engine Inertia                          */
#define fr   0.0055        /*[], Rolling res = 5.5lb/1000lb                      */
#define roa  1.20          /*[kg/m^3], Mass Density Air. Assump:Tempair=20degC,Pres=101kPa */
#define Cd   0.55          /*[], Drag Coefficient                                */
#define etad 0.95          /*[], Driveline Efficiency, exclud tires              */

/********************************
 * Valve Parameters
 ********************************/
#define IVO          336.0           /* Intake  Valve Opens  */
#define IVC          580.0           /* Intake  Valve Closes */
```

```
#define EVO         116.0              /* Exhaust Valve Opens  */
#define EVC         404.0              /* Exhaust Valve Closes */


/********************************
 * Geometrical Parameters
 ********************************/
#define V1          4.918e-03    /* Intake Manifold Volume ,       [ m^3   ] */
#define V21         1.2052e-3    /* Front Exhaust Manifold Volume, [ m^3   ] */
#define V22         1.2768e-3    /* Rear Exhaust Manifold Volume.  [ m^3   ] */
#define V2          2.448e-03    /* Exhaust Manifold Volume (Tot), [ m^3   ] */
#define V2t         1.e-4        /* Path between exhaust manifolds   3.29e-4 */
#define Vcd         1.988e-03    /* Cylinder Displacement Volume,  [ m^3   ] */
#define Ved         11.93e-03    /* Engine Displacement Volume,    [ m^3   ] */
#define Vcl         131.31e-06   /* Cyl Clearance Vol + Piston Bowl [ m^3  ] */
#define Stroke      0.1651       /* Piston Stroke,                 [ m     ] */


/********************************
 * Cylinder Parameters
 ********************************/
#define alfa1       1.0
#define alfa2       1.0
#define alfa3       1.0
#define alfa4       1.0
#define alfa5       1.0
#define alfa6       1.0


/********************************
 * Atmospheric Conditions
 ********************************/
#define P_amb       1.0e5              /* Ambient Pressure,      [ Pa ] */
#define T_amb       316.0              /* Ambient Temperature,   [ K  ] */


/********************************
 * Thermodynamic Properties
 ********************************/
#define R           287.0       /* Difference of the Specific Heats,         [ J/(kg * K) ]  */
#define cv          716.5       /* Specific Heat Capacity at Const. Volume,  [ J/(kg * K) ]  */
#define cp          1003.5      /* Specific Heat Hapacity at Const. Pressure , [ J/(kg * K) ]  */
#define gm          1.34        /* Ratio of the Specific Heats 1.34               1.30 - 1.35   */
#define Qlhv        43.2e3      /* Lower Heating Value,                      [ J/g      ]   */
#define AFs         14.5        /* Stoich air-to-fuel (mass) ratio for diesel fuel            */


/********************************
 * States
 ********************************/
#define Tindex      x[ 0]              /*  1 */
#define Pcyl1       x[ 1]              /*  2 */
#define Pcyl2       x[ 2]              /*  3 */
#define Pcyl3       x[ 3]              /*  4 */
#define Pcyl4       x[ 4]              /*  5 */
#define Pcyl5       x[ 5]              /*  6 */
#define Pcyl6       x[ 6]              /*  7 */
#define Mcyl1       x[ 7]              /*  8 */
#define Mcyl2       x[ 8]              /*  9 */
#define Mcyl3       x[ 9]              /* 10 */
```

```
#define Mcyl4          x[10]           /* 11 */
#define Mcyl5          x[11]           /* 12 */
#define Mcyl6          x[12]           /* 13 */
#define P1             x[13]           /* 14 */
#define P2             x[14]           /* 15 */
#define M1             x[15]           /* 16 */
#define M2             x[16]           /* 17 */
#define Ntc            x[17]           /* 18 */
#define Theta          x[18]           /* 19 */
#define P21            x[19]           /* 20 */
#define P22            x[20]           /* 21 */
#define M21            x[21]           /* 22 */
#define M22            x[22]           /* 23 */
#define Ner            x[23]           /* 24 */


/*******************************
 * Derivatives of States
 *******************************/
#define Tindexdot    xdot[ 0]
#define Pcyl1dot     xdot[ 1]
#define Pcyl2dot     xdot[ 2]
#define Pcyl3dot     xdot[ 3]
#define Pcyl4dot     xdot[ 4]
#define Pcyl5dot     xdot[ 5]
#define Pcyl6dot     xdot[ 6]
#define Mcyl1dot     xdot[ 7]
#define Mcyl2dot     xdot[ 8]
#define Mcyl3dot     xdot[ 9]
#define Mcyl4dot     xdot[10]
#define Mcyl5dot     xdot[11]
#define Mcyl6dot     xdot[12]
#define P1dot        xdot[13]
#define P2dot        xdot[14]
#define M1dot        xdot[15]
#define M2dot        xdot[16]
#define Ntcdot       xdot[17]
#define Thetadot     xdot[18]
#define P21dot       xdot[19]
#define P22dot       xdot[20]
#define M21dot       xdot[21]
#define M22dot       xdot[22]
#define Nerdot       xdot[23]


/*******************************
 * Inputs
 *******************************/
#define betae        input[0]
#define geare        input[1]
#define Mve          input[2]
#define ue           input[3]
#define TQsbe        input[4]


/*******************************
 *   Initial Condition
 *******************************/
#define X0 {0,1,1,1,1,1,1,0,0,0,0,0,0,1,1,0,0,1,0,0,0,0,0,0}

/*******************************
```

```
 * Declaration of Functions
 *******************************/
double powd(double a, double b);
double IVP(double theta);
double IVAP(double IVL);
double EVP(double theta);
double EVP2(double theta);
double EVAP(double EVL);
double BVP(double theta, int brake, double BVO);
double Events(double theta,double Vcyl, double Vcyldot, double Pcyl, double midot,
double T1, double Tcyl, double W1e, double We1, double We2, double W2e, double T2);
double Cyltemp(double theta, double Pcyl, double Vcyl, double mass);
double mfbdot(double theta, int fcut);
double Cmassflow(double ntc, double t1, double p1, double *tcd);
double Ceff(double ntc, double t1, double p1, double mcdot);
double Tmassflow(double ntc, double t2, double p2, double *tcd);
double Teff(double ntc, double t2, double p2);
double mfbdotnew(double theta, int mode, double n, double mfd);

double k11(double n, double mfd);
double k12(double n, double mfd);
double C1(double n, double mfd);
double k21(double n, double mfd);
double k22(double n, double mfd);
double C2(double n, double mfd);
double injdelay(double n, double mfd);
double PMdur(double n, double mfd);
double Diffdur(double n, double mfd);
double SOI(double n, double mfd);
double injpw(double n, double mfd);

double dtoo(double pd, double pu, double Td, double Tu, int *flowsw);
double fuelflow(double theta, double n, double mfd, int fcut);
double cut(double mfd, int fcut);

double Control(double u, double *controlsw1, int *controlsw2);

void   model(double *x, double *input, double *xdot, double *y);


/*******************************************************************************
 * MODEL
 *******************************/

void model(double *x, double *input, double *xdot, double *y) {
   double theta[6];
   double Vcyl[6], Vcyldot[6], Rr, rc;
   double Tcyl1, Tcyl2, Tcyl3, Tcyl4, Tcyl5, Tcyl6, T1, T2, T21, T22;
   double Mtdot, Mcdot, eta_c, eta_t, Power_t, Power_c;
   double tcd[6], PRc, Mcc, NcO, PRt, Phi, NtO;
   double IVL[6], EVL[6], IEFA[6], EEFA[6], BVL[6], BEFA[6];
   double FLOWIN, FLOWOUT, TEMPFIN, TEMPFOUT, MCYL, F21, F22, TEMPF21, TEMPF22;
   double flowin1, flowin2, flowin3, flowin4, flowin5, flowin6;
   double flowout1, flowout2, flowout3, flowout4, flowout5, flowout6;
   double midot1, midot2, midot3, midot4, midot5, midot6;
   double K, mf21dot, mf22dot;
   double We21, We22, We23, We24, We25, We26, Wf21, W21f, Wf22, W22f, A21f, A22f;
   double We11, We12, We13, We14, We15, We16;
   double W1e1, W1e2, W1e3, W1e4, W1e5, W1e6, A1e1, A1e2, A1e3, A1e4, A1e5, A1e6;
   double W2e1, W2e2, W2e3, W2e4, W2e5, W2e6, Ae211, Ae212, Ae213, Ae224, Ae225, Ae226;
```

74

```c
    double TQcyl1, TQcyl2, TQcyl3, TQcyl4, TQcyl5, TQcyl6;
    double TQsum, TQroll, TQgrade, TQaero, Jt, rg;

    double controlsw1[2], BVO, Mfdot, dummy, gt;
    double u    = ue;
    double beta = betae;
    double TQsb = TQsbe;
    double Mv   = Mve;
    double gear = geare;

    double rad2RPM = 30.0/pi;

    int flowsw[2], controlsw2[2];
    int i, fcut, brake;


/*===============================================================================================*
 * Demux'ing u into Bvo, Mfdot
 *===============================================================================================*/
    dummy = Control(u, controlsw1, controlsw2);
    Mfdot = controlsw1[0];
    BVO   = controlsw1[1];
    fcut  = controlsw2[0];
    brake = controlsw2[1];


/*********************************
 * Cylinder Parameters
 *********************************/
    Rr = 2*Rod/Stroke;
    rc = (Vcd + Vcl )/Vcl;

/*********************************
 * Firing Order: 1-5-3-6-2-4
 *********************************/
    theta[0] = fmod((Theta + 0*120.0), 720.0);          /* Cyl 1 */
    theta[1] = fmod((Theta + 2*120.0), 720.0);          /* Cyl 2 */
    theta[2] = fmod((Theta + 4*120.0), 720.0);          /* Cyl 3 */
    theta[3] = fmod((Theta + 1*120.0), 720.0);          /* Cyl 4 */
    theta[4] = fmod((Theta + 5*120.0), 720.0);          /* Cyl 5 */
    theta[5] = fmod((Theta + 3*120.0), 720.0);          /* Cyl 6 */

    for(i=0; i<6; i++)
    {
/*********************************
 * Cylinder Volumes
 *********************************/
        Vcyl[i]    = Vcl*(1 + 0.5*(rc - 1)*(Rr + 1 - cos(theta[i]*pi/180) -
                     sqrt(pow2(Rr) - pow2(sin(theta[i]*pi/180)))));
        Vcyldot[i] = (Ner*rad2RPM/60)*2*pi * (Vcl/2)*(rc - 1)*sin(theta[i]*pi/180)*(1 +
                       cos(theta[i]*pi/180)/sqrt(pow2(Rr) - pow2(sin(theta[i]*pi/180))));
    }

/*********************************
 * Intake Manifold Temp
 *********************************/
    T1 = (P1 * V1)/(M1 * R);

/*********************************
 * Exhaust Manifold Dynamics
```

```
 ********************************/
   T21 = (P21 * V21)/(M21 * R);
   T22 = (P22 * V22)/(M22 * R);

   T2  = P2 * V2t / (M2 * R);


/********************************
 * Cylinder Temperatures
 ********************************/
   Tcyl1 = Cyltemp(theta[0], Pcyl1, Vcyl[0], Mcyl1);
   Tcyl2 = Cyltemp(theta[1], Pcyl2, Vcyl[1], Mcyl2);
   Tcyl3 = Cyltemp(theta[2], Pcyl3, Vcyl[2], Mcyl3);
   Tcyl4 = Cyltemp(theta[3], Pcyl4, Vcyl[3], Mcyl4);
   Tcyl5 = Cyltemp(theta[4], Pcyl5, Vcyl[4], Mcyl5);
   Tcyl6 = Cyltemp(theta[5], Pcyl6, Vcyl[5], Mcyl6);


/********************************
 * Compressor Dynamics
 ********************************/
   Mcdot   = Cmassflow(Ntc, T1, P1, tcd);
   eta_c   = Ceff(Ntc, T1, P1, Mcdot);
   Power_c = Mcdot * (cp / eta_c) * T_amb * (powd(P1/P_amb, (gm - 1)/gm) - 1);

   PRc     = tcd[0];              /* P1 / P_amb;               */
   Mcc     = tcd[1];              /* Mcdot * sqrt(T1/298) / PRc;   */
   Nc0     = tcd[2];              /* Ntc / sqrt(T1/298);       */


/********************************
 * Turbine Dynamics
 ********************************/
   Mtdot   = Tmassflow(Ntc, T2, P2, tcd)*0.95;
   eta_t   = Teff(Ntc, T2, P2)*1.15;
   Power_t = Mtdot * (cp * eta_t) * T2 * (1 - powd(P_amb/P2, (gm - 1)/gm));

   PRt     = tcd[3];              /* P_amb / P2;               */
   Phi     = tcd[4];              /* Mtdot * 1e3 * sqrt(T2) / P2;  */
   Nt0     = tcd[5];              /* Ntc / sqrt(T2/923);       */

   for(i=0; i<6; i++)
   {
/********************************
 * IV Profile
 ********************************/
       IVL[i]  = IVP(theta[i] - IV0);
       IEFA[i] = IVAP(IVL[i]);


/********************************
 * EV Profile
 ********************************/
       EVL[i]  = EVP(theta[i] - EV0);
       EEFA[i] = EVAP(EVL[i]);


/********************************
 * BV Profile
 ********************************/
       BVL[i]  = BVP(theta[i], brake, BV0);
       BEFA[i] = EVAP(BVL[i]);
   }


/********************************
```

```
 * Mass Air Flow
*********************************/
  A1e1 = dtoo(Pcyl1, P1, Tcyl1, T1, flowsw);        /* Flows through IV */
  W1e1 = alfa1*IEFA[0]*A1e1*flowsw[0];
  We11 = alfa1*IEFA[0]*A1e1*flowsw[1];


  A1e2 = dtoo(Pcyl2, P1, Tcyl2, T1, flowsw);
  W1e2 = alfa2*IEFA[1]*A1e2*flowsw[0];
  We12 = alfa2*IEFA[1]*A1e2*flowsw[1];


  A1e3 = dtoo(Pcyl3, P1, Tcyl3, T1, flowsw);
  W1e3 = alfa3*IEFA[2]*A1e3*flowsw[0];
  We13 = alfa3*IEFA[2]*A1e3*flowsw[1];


  A1e4 = dtoo(Pcyl4, P1, Tcyl4, T1, flowsw);
  W1e4 = alfa4*IEFA[3]*A1e4*flowsw[0];
  We14 = alfa4*IEFA[3]*A1e4*flowsw[1];


  A1e5 = dtoo(Pcyl5, P1, Tcyl5, T1, flowsw);
  W1e5 = alfa5*IEFA[4]*A1e5*flowsw[0];
  We15 = alfa5*IEFA[4]*A1e5*flowsw[1];


  A1e6 = dtoo(Pcyl6, P1, Tcyl6, T1, flowsw);
  W1e6 = alfa6*IEFA[5]*A1e6*flowsw[0];
  We16 = alfa6*IEFA[5]*A1e6*flowsw[1];


  Ae211 = dtoo(P21, Pcyl1, T21, Tcyl1, flowsw);      /* Flows through EV/BV */
  We21  = alfa1*(EEFA[0] + BEFA[0])*Ae211*flowsw[0];
  W2e1  = alfa1*(EEFA[0] + BEFA[0])*Ae211*flowsw[1];


  Ae212 = dtoo(P21, Pcyl2, T21, Tcyl2, flowsw);
  We22  = alfa2*(EEFA[1] + BEFA[1])*Ae212*flowsw[0];
  W2e2  = alfa2*(EEFA[1] + BEFA[1])*Ae212*flowsw[1];


  Ae213 = dtoo(P21, Pcyl3, T21, Tcyl3, flowsw);
  We23  = alfa3*(EEFA[2] + BEFA[2])*Ae213*flowsw[0];
  W2e3  = alfa3*(EEFA[2] + BEFA[2])*Ae213*flowsw[1];


  Ae224 = dtoo(P22, Pcyl4, T22, Tcyl4, flowsw);
  We24  = alfa4*(EEFA[3] + BEFA[3])*Ae224*flowsw[0];
  W2e4  = alfa4*(EEFA[3] + BEFA[3])*Ae224*flowsw[1];


  Ae225 = dtoo(P22, Pcyl5, T22, Tcyl5, flowsw);
  We25  = alfa5*(EEFA[4] + BEFA[4])*Ae225*flowsw[0];
  W2e5  = alfa5*(EEFA[4] + BEFA[4])*Ae225*flowsw[1];


  Ae226 = dtoo(P22, Pcyl6, T22, Tcyl6, flowsw);
  We26  = alfa6*(EEFA[5] + BEFA[5])*Ae226*flowsw[0];
  W2e6  = alfa6*(EEFA[5] + BEFA[5])*Ae226*flowsw[1];



  flowin1  = W1e1 - We11;
  flowout1 = We21 - W2e1;


  flowin2  = W1e2 - We12;
  flowout2 = We22 - W2e2;


  flowin3  = W1e3 - We13;
  flowout3 = We23 - W2e3;
```

```
    flowin4  = W1e4 - We14;
    flowout4 = We24 - W2e4;


    flowin5  = W1e5 - We15;
    flowout5 = We25 - W2e5;


    flowin6  = W1e6 - We16;
    flowout6 = We26 - W2e6;


    FLOWIN   = flowin1  + flowin2  + flowin3  + flowin4  + flowin5  + flowin6;
    FLOWOUT  = flowout1 + flowout2 + flowout3 + flowout4 + flowout5 + flowout6;



    TEMPFIN  = T1*(flowin1 + flowin2 + flowin3 + flowin4 + flowin5 + flowin6);
    TEMPFOUT = Tcyl1*flowout1 + Tcyl2*flowout2 + Tcyl3*flowout3 + Tcyl4*flowout4
              + Tcyl5*flowout5 + Tcyl6*flowout6;


    F21      = flowout1 + flowout2 + flowout3;
    F22      = flowout4 + flowout5 + flowout6;


    TEMPF21  = Tcyl1*flowout1 + Tcyl2*flowout2 + Tcyl3*flowout3;
    TEMPF22  = Tcyl4*flowout4 + Tcyl5*flowout5 + Tcyl6*flowout6;

/*******************************
 * Mfdot = 0 when fcut = 1
 *******************************/
     Mfdot = cut(Mfdot, fcut);


/*******************************
 * Fuel Burn Rate
 *******************************/
    midot1 = mfbdotnew(theta[0], fcut, Ner*rad2RPM, Mfdot);
    midot2 = mfbdotnew(theta[1], fcut, Ner*rad2RPM, Mfdot);
    midot3 = mfbdotnew(theta[2], fcut, Ner*rad2RPM, Mfdot);
    midot4 = mfbdotnew(theta[3], fcut, Ner*rad2RPM, Mfdot);
    midot5 = mfbdotnew(theta[4], fcut, Ner*rad2RPM, Mfdot);
    midot6 = mfbdotnew(theta[5], fcut, Ner*rad2RPM, Mfdot);


/*******************************
 * Cylinder Air Charge
 *******************************/
    Mcyl1dot = flowin1 + fuelflow(theta[0], Ner*rad2RPM, Mfdot, fcut) - flowout1;
    Mcyl2dot = flowin2 + fuelflow(theta[1], Ner*rad2RPM, Mfdot, fcut) - flowout2;
    Mcyl3dot = flowin3 + fuelflow(theta[2], Ner*rad2RPM, Mfdot, fcut) - flowout3;
    Mcyl4dot = flowin4 + fuelflow(theta[3], Ner*rad2RPM, Mfdot, fcut) - flowout4;
    Mcyl5dot = flowin5 + fuelflow(theta[4], Ner*rad2RPM, Mfdot, fcut) - flowout5;
    Mcyl6dot = flowin6 + fuelflow(theta[5], Ner*rad2RPM, Mfdot, fcut) - flowout6;


    MCYL     = Mcyl1 + Mcyl2 + Mcyl3 + Mcyl4 + Mcyl5 + Mcyl6;

/*******************************
 * Cyl Pressure
 *******************************/
    Pcyl1dot = Events(theta[0], Vcyl[0], Vcyldot[0], Pcyl1, midot1, T1, Tcyl1,
                            W1e1, We11, We21, W2e1, T21);
    Pcyl2dot = Events(theta[1], Vcyl[1], Vcyldot[1], Pcyl2, midot2, T1, Tcyl2,
                            W1e2, We12, We22, W2e2, T21);
    Pcyl3dot = Events(theta[2], Vcyl[2], Vcyldot[2], Pcyl3, midot3, T1, Tcyl3,
                            W1e3, We13, We23, W2e3, T21);
    Pcyl4dot = Events(theta[3], Vcyl[3], Vcyldot[3], Pcyl4, midot4, T1, Tcyl4,
```

```
                                    W1e4, We14, We24, W2e4, T22);
      Pcyl5dot = Events(theta[4], Vcyl[4], Vcyldot[4], Pcyl5, midot5, T1, Tcyl5,
                                    W1e5, We15, We25, W2e5, T22);
      Pcyl6dot = Events(theta[5], Vcyl[5], Vcyldot[5], Pcyl6, midot6, T1, Tcyl6,
                                    W1e6, We16, We26, W2e6, T22);


/*******************************
 * Intake Manifold Dynamics
 *******************************/
   M1dot = Mcdot - FLOWIN;
   P1dot = gm * R * T1/V1 * M1dot;


/*******************************
 * Front Exhaust Manifold
 *******************************/
   K       = 10.0;
   A21f    = dtoo(P2, P21, T2, T21, flowsw);                  /* d(pd, pu, Td, Tu) */;
   W21f    = K*pi*pow2(0.0127)/4*A21f*flowsw[0];
   Wf21    = K*pi*pow2(0.0127)/4*A21f*flowsw[1];
   mf21dot = W21f - Wf21;
   P21dot  = gm * R/V21 * (TEMPF21 - mf21dot*T21);
   M21dot  = F21 - mf21dot;


/*******************************
 * Rear Exhaust Manifold
 *******************************/
   A22f    = dtoo(P2, P22, T2, T22, flowsw);                  /* d(pd, pu, Td, Tu) */;
   W22f    = K*pi*pow2(0.0127)/4*A22f*flowsw[0];
   Wf22    = K*pi*pow2(0.0127)/4*A22f*flowsw[1];
   mf22dot = W22f - Wf22;
   P22dot  = gm * R/V22 * (TEMPF22 - mf22dot*T22);
   M22dot  = F22 - mf22dot;


/*******************************
 * Exhaust Manifold
 *******************************/
   P2dot = gm * R/V2t * (mf21dot*T21 + mf22dot*T22 - Mtdot*T2);
   M2dot = mf21dot + mf22dot - Mtdot;


/*******************************
 * Turbocharger
 *******************************/
   Ntcdot = (91.2)*(Power_t - Power_c) / (Itc * Ntc);



/*******************************
 * TQcyl
 *******************************/
   TQcyl1 = Pcyl1*pi*pow2(Bore)/4*Rcrank*sin(theta[0]*pi/180)*(1.0+Rcrank/Rod*cos(theta[0]*pi/180));
   TQcyl2 = Pcyl2*pi*pow2(Bore)/4*Rcrank*sin(theta[1]*pi/180)*(1.0+Rcrank/Rod*cos(theta[1]*pi/180));
   TQcyl3 = Pcyl3*pi*pow2(Bore)/4*Rcrank*sin(theta[2]*pi/180)*(1.0+Rcrank/Rod*cos(theta[2]*pi/180));
   TQcyl4 = Pcyl4*pi*pow2(Bore)/4*Rcrank*sin(theta[3]*pi/180)*(1.0+Rcrank/Rod*cos(theta[3]*pi/180));
   TQcyl5 = Pcyl5*pi*pow2(Bore)/4*Rcrank*sin(theta[4]*pi/180)*(1.0+Rcrank/Rod*cos(theta[4]*pi/180));
   TQcyl6 = Pcyl6*pi*pow2(Bore)/4*Rcrank*sin(theta[5]*pi/180)*(1.0+Rcrank/Rod*cos(theta[5]*pi/180));

   TQsum  = TQcyl1 +TQcyl2 +TQcyl3 +TQcyl4 +TQcyl5 +TQcyl6;


/*===============================================================================================*
 * Vehicle Dynamics
```

```
   *============================================================================================*/
   rg        = rw/(gd*gear);                    /*Tot gear ratio*/
   Jt        = Mv*pow2(rg) + Ie;                /*[Nm*sec^2], Total Vehicle Inertia. Assump: Id=It=Iw=0*/

   TQroll   = -fr*Mv*rg*g*cos(beta*pi/180);
   TQgrade =     -Mv*rg*g*sin(beta*pi/180);
   TQaero   = -0.5*roa*Cd*Av*pow3(rg)*pow2(Ner);

   Nerdot   = 1/Jt*(TQaero +TQroll +TQgrade +TQsum +TQsb*rg);        /*Ner rad/sec*/



/*******************************
 * Crank Angle
 *******************************/
   Thetadot = 6.0*Ner*rad2RPM;

/*******************************
 * Time Vector
 *******************************/
   Tindexdot = 1.0;

/*******************************
 * Outputs
 *******************************/
   y[ 0] = Theta;
   y[ 1] = Tindex;

   y[ 2] = ue;
   y[ 3] = BVO;
   y[ 4] = Mfdot;

   y[ 5] = T1;
   y[ 6] = T2;
   y[ 7] = P1;
   y[ 8] = P2;

   y[ 9] = Ntc;
   y[10] = Mtdot;
   y[11] = Mcdot;

   y[12] = Pcyl1;
   y[13] = Mcyl1;
   y[14] = TQcyl1;

   y[15] = TQroll;
   y[16] = TQgrade;
   y[17] = TQaero;
   y[18] = TQsum;
   y[19] = TQsb;

   y[20] = Ner;

   y[21] = beta;
   y[22] = Ner*rg;
}


/*============================================================================
 * Demux'ing u into Bvo, Mfdot, etc
```

```
 *=============================================*/
 double Control(double u, double *controlsw1, int *controlsw2)
{
    double mfdot, bvo;
    double deadb = 0.0;
    int fcut, brake;

    if (u <= deadb)              /* Braking Mode*/
    {
        mfdot = 0.0;
        bvo   = -0.8*u +620.0;   /*BVT in deg, 0>u>=-100*/
        fcut  = 1;
        brake = 1;
    }


     else if (u > deadb)          //* Combustion Mode
    {
      mfdot = 14.25e-5*u;        //*Fuel flow in kg/sec, when 0<=u<=100
      bvo   = 0.0;
      fcut  = 0;                 //*0=NoCut; 1=Cut
      brake = 0;                 //*0=Comb i.e. bvl=0;  1=Brake
    }


    controlsw1[0] = mfdot;
    controlsw1[1] = bvo;
    controlsw2[0] = fcut;
    controlsw2[1] = brake;

    return(deadb);
}




/*********************************************
 * Power fcn: a^b
 ******************************/
double powd(double a, double b) {
    return exp(b * log(a));
}

/*********************************************
 * New Press Difference
 *******************************************/
double dtoo(double pd, double pu, double Td, double Tu, int*flowsw)
{
    double pdpu, pupd, dp, a, cr;

    cr   = 0.5386;      /* for gm = 1.34 */

    pdpu = pd / pu;
    pupd = 1.0 / pdpu;
    a    = (gm + 1.0)/(2.0 * (gm - 1.0));

    if (pdpu <= cr)
    {
      dp = pu / sqrt(R*Tu) * sqrt(gm) * powd((2.0/(gm+1.0)), a);
      flowsw[0] = 1;
      flowsw[1] = 0;
```

```c
    }

    else if (pdpu <= 1.0)
    {
      dp = pu / sqrt(R*Tu) * powd(pdpu,(1.0/gm))*sqrt(2.0*gm/(gm-1.0)*
               (1.0-powd(pdpu,((gm-1.0)/gm)))));
      flowsw[0] = 1;
      flowsw[1] = 0;
    }

    else if (pdpu <= (1.0/cr))
    {
      dp = pd / sqrt(R*Td) * powd(pupd,(1.0/gm))*sqrt(2.0*gm/(gm-1.0)*
               (1.0 - powd(pupd,((gm-1.0)/gm)))));
      flowsw[0] = 0;
      flowsw[1] = 1;
    }

    else
    {
      dp = pd / sqrt(R*Td) * sqrt(gm)*powd((2.0/(gm+1.0)), a);
      flowsw[0] = 0;
      flowsw[1] = 1;
    }

    return(dp);

}

/********************************************************************************
 * IV Profile
 ******************************/
 double IVP(double theta)
{
    static double c0, c1, c2, c3, c4, max_IVL;
    double IVL;

/*===============================*/
    c0  = -1.913187294204836e-04;
    c1  = -1.729451938600456e-06;
    c2  =  3.564963677743456e-06;
    c3  = -3.020542173514134e-08;
    c4  =  6.413922150362387e-11;

    max_IVL = 1.29032e-02;
/*===============================*/

    if((theta >= 0) && (theta <= 230))
       IVL = c0 + c1*theta + c2*powd(theta,2) + c3*powd(theta,3) + c4*powd(theta,4);
    else
       IVL = 0;

    return(max(IVL,0));
}

/********************************************************************************
 * IV Area Profile
 ******************************/
 double IVAP(double IVL)
{
```

```
   double d0, e0, e1, e2, iefa;


/*==============================*/
   d0  =   1.298274768672963e-01;
   e0  =   4.199524905580524e-04;
   e1  =   8.692227804361272e-02;
   e2  = -3.439079824903442e+00;
/*==============================*/

   if((IVL >= 0) && (IVL <= 6.4516e-3))
       iefa = d0 * IVL;
   else
       iefa = e0 + e1*IVL + e2*powd(IVL,2);

   return(iefa);
}


/**********************************************************************************
 * EV Profile
 ******************************/
 double EVP(double theta)
{
   static double a0, a1, a2, b1, b2, b3, b4, b5, b6;
   static double c0, c1, c2, c3, c4, max_EVL;
   double EVL;

/*==============================*/
   a0 =   2.765424056667856e-05;
   a1 = -1.671945516804518e-06;
   a2 =   2.225235902209503e-06;

   b1 = -1.512742598990643e-04;
   b2 =   1.164582640282355e-05;
   b3 = -1.750410921928115e-07;
   b4 =   1.204555533965746e-09;
   b5 = -4.020037227878294e-12;
   b6 =   5.184851197366564e-15;

   c0 = -8.209566538328625e+03;
   c1 =   1.312368652685806e+02;
   c2 = -7.847657077438077e-01;
   c3 =   2.079606755813267e-03;
   c4 = -2.061665746775699e-06;

   max_EVL =   1.29794e-02;
/*==============================*/

   if((theta >= 0) && (theta <= 28))
      EVL = a0 + a1*theta + a2*pow2(theta);
   else if((theta > 28) && (theta <= 226))
      EVL = b1*theta + b2*pow2(theta) + b3*pow3(theta) + b4*pow4(theta) +
            b5*pow5(theta) + b6*pow6(theta);
   else if((theta > 226) && (theta <= 284))
      EVL = exp(c0 + c1*theta + c2*pow2(theta) + c3*pow3(theta) + c4*pow4(theta));
   else
      EVL = 0;

   return(EVL);
}
```

```c
/********************************************************************************
 * EV Area Profile
 ******************************/
 double EVAP(double EVL)
{
    static double a0, b0, b1, b2;
    double eefa;

/*==============================*/
    a0 = 1.144895396937455e-01;

    b0 = -1.800349211707203e-04;
    b1 =  1.887961146825884e-01;
    b2 = -6.486453877010925e+00;
/*==============================*/

    if(( EVL >= 0 ) && (EVL <= 7.9756e-3))
        eefa = a0 * EVL;
    else
        eefa = b0 + b1*EVL + b2*pow2(EVL);

    return(eefa);
}

/********************************************************************************
 * BV Profile
 ******************************/
 double BVP(double theta, int brake, double BV0)
{
    static double a0, a1, a2, a3, b0, b1, b2;
    static double c0, c1, c2, c3, d0, d1, d2, d3, bvlmax;
    double bvl, ta;

/*==============================*/
    a0 = -1.471653442718320e-006;
    a1 = -2.499086176405385e-005;
    a2 =  1.851175393202057e-005;
    a3 = -1.036658975462242e-006;

    b0 = -5.447747128075851e-004;
    b1 =  1.373507637942792e-004;
    b2 = -1.585112302287139e-006;

    c0 = -1.037779668219718e-003;
    c1 =  1.281702419189956e-004;
    c2 = -1.005967124905654e-006;
    c3 = -3.407340361235717e-009;

    d0 = -1.727734535320311e-001;
    d1 =  6.509636442427389e-003;
    d2 = -8.001173312552896e-005;
    d3 =  3.224896963318234e-007;

    bvlmax = 2.429962374e-003;
/*==============================*/
    switch(brake)
    {
        case 0:
            bvl = 0;
        break;
```

```
        case 1:
        {
            if (theta >= BVO)
                ta = theta - BVO;
            else
                ta = theta + (720 - BVO);

            if((ta >= 0) && (ta < 8))
                bvl = a0 + a1*ta + a2*pow2(ta) + a3*pow3(ta);
            else if (ta < 42.7)
                bvl = b0 + b1*ta + b2*pow2(ta);
            else if (ta < 51.3)
                bvl = bvlmax;
            else if (ta < 78)
                bvl = c0 + c1*ta + c2*pow2(ta) + c3*pow3(ta);
            else if (ta < 92)
                bvl = d0 + d1*ta + d2*pow2(ta) + d3*pow3(ta);
            else
                bvl = 0;
        }
        break;
    }

    return(max(bvl,0));
}


/*********************************************************************************
 * Cyl Temp
 *******************************/
double Cyltemp(double theta, double Pcyl, double Vcyl, double mass)
{
    return((Pcyl * Vcyl) / (mass * R));
}


/*********************************************************************************
 *  Engine Events
 *******************************/
double Events(double theta, double Vcyl, double Vcyldot, double Pcyl, double midot,
double T1, double Tcyl, double W1e, double We1, double We2, double W2e, double T2)
{
    double pcyldot;

    pcyldot = (gm/Vcyl)*(R*T1*W1e + R*T2*W2e - R*Tcyl*(We2+We1) - Pcyl*Vcyldot) +
              ((gm-1.0)/Vcyl)*midot*Qlhv;

    return(pcyldot);
}


/*********************************************************************************
 *  Compressor Mass Flow
 *******************************/
double Cmassflow(double ntc, double t1, double p1, double *tcd)
{
    static double m0, m1, m2, d0, d1, d2, a0, a1, a2, s0, s1, s2;
    double max_M0, M0, max_Pr0, Pr0, alpha, sline;
    double b, dxdy, cut, ycut, prcut;
    double Ncor, sc_Ncor, max_Ncor, PR, sc_PR, max_PR;
    double max_mcc, mcc, sc_mcc, mcdot;
```

```
/*===============================*/
   m0 = -2.820933230281464e-01;
   m1 =  2.745882090631977e+00;
   m2 = -1.468018156292942e+00;

   d0 =  2.715989857759699e-01;
   d1 = -4.020050903354849e-01;
   d2 =  1.130410038751046e+00;

   a0 =  5.546419475011604e+01;
   a1 = -1.403616187275394e+02;
   a2 =  1.050524987190445e+02;

   s0 = 3.974485071351443e-02;
   s1 = 2.628924666713938e-02;
   s2 = 1.397591201309391e-02;

   max_Pr0  =    4.8069;
   max_M0   =    0.5900;
   max_Ncor = 122.3220;
   max_PR   =    4.7180;
   max_mcc  =    0.5205;
/*===============================*/

   Ncor    = (ntc / 1000) / sqrt(t1/298);
   sc_Ncor = Ncor / max_Ncor;

   PR      = p1 / P_amb;
   sc_PR   = PR / max_Pr0;

   M0      = m0 + m1*sc_Ncor + m2*powd(sc_Ncor, 2);
   Pr0     = d0 + d1*sc_Ncor + d2*powd(sc_Ncor, 2);
   alpha   = a0 + a1*sc_Ncor + a2*powd(sc_Ncor, 2);

   sline   = s0 + s1*PR + s2*powd(PR,2);

   cut     = 0.99;

   if (sc_PR <= (cut * Pr0))
   {
      sc_mcc  = M0 + log(1 - sc_PR/Pr0) / alpha;
   }
   else
   {
      dxdy    = -alpha * Pr0 * fabs(1 - cut);
      ycut    = M0 + log(1 - cut) / alpha;
      prcut   = cut*Pr0;
      b       = prcut - ycut*dxdy;
      sc_mcc  = (sc_PR - b) / dxdy;
   }
      mcc     = max(max_M0*sc_mcc, 0);
      mcdot   = mcc * PR / sqrt(t1/298);

      tcd[0]  = PR;
      tcd[1]  = mcc;
      tcd[2]  = Ncor * 1000;

   return(mcdot);
}
```

```c
/*******************************************************************************
 * Compressor Efficiency
 *******************************/
double Ceff(double ntc, double t1, double p1, double mcdot)
{
    static double a0, a1, a2, b0, b1, b2, g0, g1, g2;
    double al, be, ga, eff, max_eff, eta_c;
    double Ncor, sc_Ncor, max_Ncor, PR, sc_PR, max_Pr0, mcc, sc_mcc, max_mcc;

/*==============================*/
    a0 =   3.136296865138709e-01;
    a1 =   2.234549489372691e+00;
    a2 =  -3.638634591377077e+00;

    b0 = 6.437067078911609e-01;
    b1 = 2.814099449804274e+00;
    b2 = 8.817550108048638e-01;

    g0 = -4.106542839177144e+00;
    g1 =  3.373707579047213e+00;
    g2 = -1.659103398244337e+00;

    max_Ncor = 122.3220;
    max_Pr0  =   4.8069;
     max_mcc  =   0.5205;
    max_eff  =   0.7560;
/*==============================*/

    Ncor    = (ntc / 1000) / sqrt(t1/298);
    sc_Ncor = Ncor / max_Ncor;

    PR      = p1 / P_amb;
    sc_PR   = PR / max_Pr0;

    mcc     = mcdot * sqrt(t1/298) / PR;
    sc_mcc  = mcc / max_mcc;

    al      = a0 + a1*sc_Ncor + a2*powd(sc_Ncor,2);
    be      = b0 + b1*sc_Ncor + b2*powd(sc_Ncor,2);
    ga      = g0 + g1*sc_Ncor + g2*powd(sc_Ncor,2);

    eff     = al + be*sc_mcc + ga*powd(sc_mcc,2);
    eta_c   = max(min(eff * max_eff, 0.8), 0.6);

    return(eta_c);
 }


/*******************************************************************************
 *  Turbine Mass Flow
 *******************************/
   double Tmassflow(double ntcc, double t2, double p2, double *tcd) {
   double m0, m1, m2, a0, a1, a2, max_Ncor, max_phi, max_PR;
   double Ncor, PR, M0, Aa, phi, mtdot;

/*==============================*/
   m0 =   5.756086855589766e-002;
   m1 =   9.730463340680169e-005;
   m2 =  -1.245523837473848e-006;

   a0 =  -6.101771247056622e+000;
```

```
      a1 =   4.092078758800726e-002;
      a2 = -1.569220820088179e-004;

      max_Ncor = 121.5950;
      max_phi  = 0.0548;
      max_PR   = 3.2480;
/*==============================*/

      Ncor    = (ntcc / 1000.0) / sqrt(t2/923.0);
      PR      = p2 / P_amb;

      M0      = m0 + m1*Ncor + m2*pow2(Ncor);
      Aa      = a0 + a1*Ncor + a2*pow2(Ncor);

      phi     = M0*(1.0 - exp(Aa * (PR - 1.0)));
      mtdot   = phi * p2 /(1.0e3 * sqrt(t2));

      tcd[3] = PR;
      tcd[4] = phi;
      tcd[5] = Ncor * 1000.0;

      return(mtdot);
}


/*********************************************************************************
 *  Turbine Efficiency
 ********************************/
   double Teff(double ntc, double t2, double p2)
   {
   static double b0, b1, b2, b3, b4, b5, b6;
   double Ncor, sc_Ncor, max_Ncor, PR, sc_PR, max_PR, eff, max_eff, eta_t;

/*==============================*/
   b0 = -1.661774609486501e-001;
   b1 =  4.929785058417423e-001;
   b2 =  4.835483545659804e+000;
   b3 = -3.783460392358137e+000;
   b4 = -1.202764645681852e+001;
   b5 =  8.032677520347596e+000;
   b6 =  3.361762044441541e+000;

   max_eff  = 0.7890;
   max_PR   = 3.2480;
   max_Ncor = 121.5950;
/*==============================*/

   Ncor    = (ntc / 1000.0) / sqrt(t2/923.0);
   sc_Ncor = Ncor  / max_Ncor;

   PR      = min(p2 / P_amb, max_PR);
   sc_PR = PR/ max_PR;

   eff    = b0 + b1*sc_Ncor + b2*sc_PR + b3*powd(sc_Ncor,2) + b4*powd(sc_PR,2) +
            b5*sc_Ncor*sc_PR + b6*powd(sc_PR,3);

   eta_t = max(min(eff * max_eff, 0.8), 0.5);

   return(eta_t);
}
```

```
/****************************************************************************
 * Fuel Burn Rate
 *******************************/
  double mfbdotnew(double theta, int fcut, double n, double mfd)
  {
  double FBR1, FBR2, FBR, thunit, Diffend, PMend, SOC;

  SOC     = injdelay(n,mfd) + SOI(n, mfd) + 360.0;
  Diffend = SOC + Diffdur(n, mfd);
  PMend   = SOC + PMdur(n, mfd);

  /* Diffusive burning - assumes diffusive burning is longer than 720-SOC*/
  if(theta > SOC)

   {
     thunit = (theta - SOC)/(Diffend-SOC);
     FBR1   = C1(n, mfd)*k12(n,mfd)*(k11(n,mfd)+1)*powd(thunit,k11(n,mfd))
              *exp(-k12(n,mfd)*powd(thunit, (k11(n,mfd)+1)));
   }
  else if(theta < (Diffend-720.0))

   {
     thunit = (theta-(SOC-720))/(Diffend-SOC);
     FBR1   = C1(n,mfd)*k12(n,mfd)*(k11(n,mfd)+1)*powd(thunit,k11(n,mfd))
              *exp(-k12(n,mfd)*powd(thunit, (k11(n,mfd)+1)));
   }
  else
     FBR1 = 0;

  /* Pre-mixed burning */
  if(PMend < 720.0)
  {
      if((theta>SOC) && (theta<PMend))
      {
          thunit = (theta-SOC)/(PMend-SOC);
          FBR2   = C2(n,mfd)*k22(n,mfd)*k21(n,mfd)*powd((1-powd(thunit,k21(n,mfd))),
                   (k22(n,mfd)-1))*powd(thunit,(k21(n,mfd)+1));
      }
      else
          FBR2 = 0;
  }
   else    /* PMend >720deg */
   {
       if(theta > SOC)
       {
          thunit = (theta - SOC)/(PMend - SOC);
          FBR2   = C2(n,mfd)*k22(n,mfd)*k21(n,mfd)*powd((1-powd(thunit,k21(n,mfd))),
                   (k22(n, mfd) - 1))*powd(thunit, (k21(n, mfd) + 1));
       }
       else if(theta < (PMend - 720.0))
       {
          thunit = (theta - (SOC - 720.0))/(PMend - SOC);
          FBR2   = C2(n, mfd)*k22(n, mfd)*k21(n, mfd)*powd((1 - powd(thunit, k21(n, mfd))),
                   (k22(n, mfd) - 1))*powd(thunit, (k21(n, mfd) + 1));
       }
       else
           FBR2 = 0;
   }
```

```
    switch(fcut)
    {
        case 0:
        {
                if(FBR1 >= FBR2)
                    FBR = FBR1;
                else
                    FBR = FBR2;
        }
        break;

        case 1:
            FBR = 0;
        break;
    }

    return(FBR); /* g/sec */
}


/**********************************************************************************
 * Fuel Flow Cut
 *****************************************/
    double cut(double mfd, int fcut) {
    switch(fcut)
    {
        case 0:
                mfd = mfd;
         break;

        case 1:
            mfd = 0;
        break;
    }
    return(mfd);
}


/**********************************************************************************
 * Injection pulse flow rate
 *****************************************/
    double fuelflow(double theta, double n, double mfd, int fcut) {
    double ipw, injstart, EOI, wf;
    int injsw;

    ipw       = injpw(n,mfd);
    injstart = 360.0 + SOI(n, mfd);
    EOI       = injstart + ipw;

    /* Note: EOI can and will sometimes be > 720 - care must be taken */
    if (EOI < 720.0)
    {
        if((theta >= injstart) && (theta <= EOI))
            injsw = 1;
        else
            injsw = 0;
    }
    else     /* EOI > 720 */
    {
        if(theta >= injstart)
            injsw = 1;
```

```
            else if(theta <= (EOI-720.0))
                  injsw = 1;
            else
                  injsw = 0;
     }

     switch(fcut)
     {
         case 0:
                  wf = 720.0*mfd/(6*ipw)*injsw;
          break;

         case 1:
                  wf = 0;
         break;
     }

     return(wf);
}


/*********************************************************************************
 * Calculation of Combustion Parameters
 * k11, k12, C1, k21, k22, C2, injection delay, Pre-Mixed duration, Diffusive duration
 *      ("Q"is generic name for above parameters)
 *********************************************************************************/

/***************** Function k11
                      Q is k11  */

double k11(double n, double mfd)

{
     static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
     double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */ /* Linear in Ne*/
     QC0 =    8.98e-1;
     QC1 =    -2.64e-1;
     QC2 =    -2.81;
     QC3 =    5.37;     /*Linear in Ne fit params*/
     QC4 =    1.52;
     QC5 =    -4.01;


/* Quadratic in Ne*/
     /*QC0   =    8.33e-1;     Quadratic in Ne*/
     /*QC1   =    -2.78e+2;
     QC2 =    1.79e+2;
     QC3 =    -1.05e+2;
     QC4 =    -1.65e+2;
     QC5 =    3.51e+2;
     QC6 =    3.47e+2;
     QC7 =    -1.46e+2;
     QC8 =    -1.83e+2;*/

     maxRPM  =    1800.0;   /* max and min values needed for proper scaling */
     minRPM  =    800.0;
     maxWf   =    113.1; /* in lb-hr! */
     minWf   =    48.64; /* in lb-hr! */
     maxQ    =    1.2;
     minQ    =    0.9;
```

```
/****************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots);*/
    Qest=(maxQ-minQ)*Qs + minQ;

    return(Qest);
 }
    /* end function k11*/


/****************** Function k12
                   Q is k12  */

double k12(double n, double mfd)

{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    1.03;
    QC1 =    -3.5e-1;
    QC2 =    -2.9;
    QC3 =    2.6;
    QC4 =    1.69;
    QC5 =    -1.91;
/*  QC0 =    1.0;     Quadratic in Ne*/ /*  QC1 =    -6.73e+1;
    QC2 =    4.15e+1;
    QC3 =    -2.93e+1;
    QC4 =    -3.83e+1;
    QC5 =    8.72e+1;
    QC6 =    8.37e+1;
    QC7 =    -2.94e+1;
    QC8 =    -4.9e+1;*/

    maxRPM  =    1800.0;  /* max and min values needed for proper scaling */
    minRPM  =    800.0;
    maxWf   =    113.1;
    minWf   =    48.64;
    maxQ    =    28.0;
    minQ    =    8.0;
/****************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest=(maxQ-minQ)*Qs + minQ;

    return(Qest);
}
    /* end function k12 */
```

```
/******************** Function C1
                      Q is C1  */


double C1(double n, double mfd)

{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    -2.02e-2;
    QC1 =    2.73e-1;
    QC2 =    8.25e-1;
    QC3 =    1.08e-1;
    QC4 =    2.17e-1;
    QC5 =    -4.2e-1;
/*  QC0 =    -2.25e-15;   Quadratic in Ne*/ /*  QC1 =    5.22e+1;
    QC2 =    -3.34e+1;
    QC3 =    2.17e+1;
    QC4 =    3.18e+1;
    QC5 =    -6.84e+1;
    QC6 =    -6.5e+1;
    QC7 =    2.66e+1;
    QC8 =    3.55e+1;*/

    maxRPM  =    1800.0;  /* max and min values needed for proper scaling */
    minRPM  =    800.0;
    maxWf   =    113.1;
    minWf   =    48.64;
    maxQ    =    22.0;
    minQ    =    7.4;
/***************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest=(maxQ-minQ)*Qs + minQ;

    return(Qest);
}
    /* end function C1 */


/******************** Function k21
                      Q is k21  */


double k21(double n, double mfd)

{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    -1.13e-3;
    QC1 =    1.02;
    QC2 =    7.4e-1;
```

```
      QC3 =    -1.02;
      QC4 =    -8.14e-1;
      QC5 =    9.69e-1;
/*    QC0 =    2.25e-15;    Quadratic in Ne*/ /*  QC1 =    -3.06e+1;
      QC2 =    2.23e+1;
      QC3 =    -3.45e+1;
      QC4 =    -1.5e+1;
      QC5 =    4.78e+1;
      QC6 =    3.94e+1;
      QC7 =    8.56;
      QC8 =    -3.71e+1;*/

      maxRPM  =    1800.0;   /* max and min values needed for proper scaling */
      minRPM  =    800.0;
      maxWf   =    113.1;
      minWf   =    48.64;
      maxQ    =    3.0;
      minQ    =    1.5;
/*****************************/

      mfdotlbhr=mfd*3600.0*2.2046;
      Ns=(n-minRPM)/(maxRPM-minRPM);
      Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
      Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
          + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

      Qest=(maxQ-minQ)*Qs + minQ;

      return(Qest);
}
      /* end function k21 */


/****************** Function k22
                       Q is k22  */

double k22(double n, double mfd)

{
      static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
      double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
      QC0 =    -5.5e-2;
      QC1 =    8.11e-1;
      QC2 =    1.28;
      QC3 =    -2.26;
      QC4 =    -1.82;
      QC5 =    3.01;
/*    QC0 =    -3.6e-14;   Quadratic in Ne*/ /*  QC1 =    -2.77e+2;
      QC2 =    1.78e+2;
      QC3 =    -7.76e+1;
      QC4 =    -1.66e+2;
      QC5 =    3.29e+2;
      QC6 =    3.47e+2;
      QC7 =    -1.82e+2;
      QC8 =    -1.5e+2; */

      maxRPM  =    1800.0;   /* max and min values needed for proper scaling */
      minRPM  =    800.0;
```

```
    maxWf    =    113.1;
    minWf    =    48.64;
    maxQ     =    28.0;
    minQ     =    3.8;
/****************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
       + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest=(maxQ-minQ)*Qs + minQ;


    return(Qest);
}
    /* end function k22 */


/****************** Function C2
                    Q is C2  */

double C2(double n, double mfd)
{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    1.77e-1;
    QC1 =    1.19;
    QC2 =    1.27e-1;
    QC3 =    -2.55;
    QC4 =    -6.39e-1;
    QC5 =    2.38;
/*  QC0 =    1.37e-1;        Quadratic in Ne*/ /*  QC1 =    -3.33e+2;
    QC2 =    2.15e+2;
    QC3 =    -8.13e+1;
    QC4 =    -2.09e+2;
    QC5 =    4.06e+2;
    QC6 =    4.18e+2;
    QC7 =    -2.38e+2;
    QC8 =    -1.76e+2; */

    maxRPM   =    1800.0;  /* max and min values needed for proper scaling */
    minRPM   =    800.0;
    maxWf    =    113.1;
    minWf    =    48.64;
    maxQ     =    1.2e2;
    minQ     =    5.05e1;
/****************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
       + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots);*/

    Qest=(maxQ-minQ)*Qs + minQ;
```

```
    return(Qest);
}
    /* end function C2 */


/****************** Function injdelay
                    Q is injection delay (deg)*/


double injdelay(double n, double mfd)
{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    3.79e-1;
    QC1 =    1.12;
    QC2 =    -3.4;
    QC3 =    1.87;
    QC4 =    3.76;
    QC5 =    -3.3;
/*  QC0 =    3.33e-1;      Quadratic in Ne*/ /*  QC1 =    1.81e+2;
    QC2 =    -1.17e+2;
    QC3 =    5.16e+1;
    QC4 =    1.09e+2;
    QC5 =    -2.14e+2;
    QC6 =    -2.25e+2;
    QC7 =    1.16e+2;
    QC8 =    9.77e+1; */

    maxRPM  =    1800.0;  /* max and min values needed for proper scaling */
    minRPM  =    800.0;
    maxWf   =    113.1;
    minWf   =    48.64;
    maxQ    =    5.0;
    minQ    =    2.6;
/***************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest=(maxQ-minQ)*Qs + minQ;

    return(Qest);
}
    /* end function injdelay */



/****************** Function Diffdur
                    Q is diffusive burning duration (deg)   */


double Diffdur(double n, double mfd)
{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    6.32e-1;
```

```
      QC1  =    -4.93e-1;
      QC2  =    -1.72;
      QC3  =    2.11;
      QC4  =    1.63;
      QC5  =    -1.21;
/*    QC0  =    5.69e-1;         Quadratic in Ne*/ /*  QC1 =    -2.01e+2;
      QC2  =    1.31e+2;
      QC3  =    -9.94e+1;
      QC4  =    -1.15e+2;
      QC5  =    2.65e+2;
      QC6  =    2.5e+2;
      QC7  =    -8.0e+1;
      QC8  =    -1.51e+2; */


      maxRPM  =    1800.0;  /* max and min values needed for proper scaling */
      minRPM  =    800.0;
      maxWf   =    113.1;
      minWf   =    48.64;
      maxQ    =    60.0;
      minQ    =    54.2;
/***************************/

      mfdotlbhr=mfd*3600.0*2.2046;
      Ns=(n-minRPM)/(maxRPM-minRPM);
      Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
      Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
         + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

      Qest=(maxQ-minQ)*Qs + minQ;

      return(Qest);
}
      /* end function Diffdur */


/******************* Function PMdur
                     Q is pre-mixed burning duration (deg)  */

double PMdur(double n, double mfd)
{
      static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
      double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
      QC0  =    2.04e-2;
      QC1  =    1.09;
      QC2  =    3.94e-1;
      QC3  =    -1.41;
      QC4  =    -1.01;
      QC5  =    1.91;
/*    QC0  =    1.02e-1;          Quadratic in Ne*/ /*  QC1 =-2.41e+2;
      QC2  =    1.52e+2;
      QC3  =    -4.68e+1;
      QC4  =    -1.46e+2;
      QC5  =    2.74e+2;
      QC6  =    3.03e+2;
      QC7  =    -1.81e+2;
      QC8  =    -1.13e+2; */
```

```
    maxRPM   =   1800.0;  /* max and min values needed for proper scaling */
    minRPM   =    800.0;
    maxWf    =    113.1;
    minWf    =     48.64;
    maxQ     =      8.0;
    minQ     =      3.1;
/****************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest=(maxQ-minQ)*Qs + minQ;

    return(Qest);
}
    /* end function PMdur */


/***********************************************************************************
 * Function SOI
 * Q is start of injection (relative to combustion at 360deg) (deg)
 ***********************************************************************************/
double SOI(double n, double mfd)
{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    3.22e-1;
    QC1 =    2.73e-2;
    QC2 =    2.75;
    QC3 =   -2.53;
    QC4 =   -2.80;
    QC5 =    2.26;
/*  QC0 =    3.94e-1;         Quadratic in Ne*/ /*  QC1 =    1.11e+2;
    QC2 =   -7.23e+1;
    QC3 =    6.86e+1;
    QC4 =    6.17e+1;
    QC5 =   -1.56e+2;
    QC6 =   -1.38e+2;
    QC7 =    2.8e+1;
    QC8 =    9.68e+1; */

    maxRPM   =   1800.0;  /* max and min values needed for proper scaling */
    minRPM   =    800.0;
    maxWf    =    113.1;
    minWf    =     48.64;
    maxQ     =    353.1;
    minQ     =    346.5;
/****************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
```

```
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest=(maxQ-minQ)*Qs + minQ;

    return(Qest);
}
    /* end function SOI */


/****************** Function injow
                    Q is injector pulse width (deg)   */

double injpw(double n, double mfd)
{
    static double QC0,QC1,QC2,QC3,QC4,QC5,maxRPM,minRPM,maxWf,minWf,maxQ,minQ,QC6,QC7,QC8;
    double Ns, Mfdots, Qs, Qest, mfdotlbhr;

/* Definition of coefficients */
    QC0 =    1.88e-2;
    QC1 =   -2.18e-1;
    QC2 =    1.28;
    QC3 =   -4.08e-2;
    QC4 =   -2.31e-1;
    QC5 =    1.41e-1;
/*  QC0 =   -4.49e-15;        Quadratic in Ne*/ /*  QC1 =    6.28e+1;
    QC2 =   -3.83e+1;
    QC3 =    1.26e+1;
    QC4 =    3.74e+1;
    QC5 =   -7.14e+1;
    QC6 =   -7.88e+1;
    QC7 =    4.63e+1;
    QC8 =    3.03e+1; */

    maxRPM  =    1800.0;   /* max and min values needed for proper scaling */
    minRPM  =    800.0;
    maxWf   =    113.1;
    minWf   =    48.64;
    maxQ    =    29.0;
    minQ    =    16.4;
/***************************/

    mfdotlbhr=mfd*3600.0*2.2046;
    Ns=(n-minRPM)/(maxRPM-minRPM);
    Mfdots=(mfdotlbhr-minWf)/(maxWf-minWf);
    Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots);
/*  Qs=QC0 + QC1*Ns + QC2*Mfdots + QC3*Ns*Mfdots + QC4*pow2(Mfdots) + QC5*Ns*pow2(Mfdots)
        + QC6*pow2(Ns) + QC7*pow2(Ns)*Mfdots + QC8*pow2(Ns)*pow2(Mfdots); */

    Qest = (maxQ-minQ)*Qs + minQ;

    return(Qest);
}
    /* end function injpw */


/*===================== END
=================================================*/



/**********************************************************
```

```
* vec2sl.c --> 01-26-99
*
**************************************************************/

#define S_FUNCTION_NAME vec2sl

#include <simstruc.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <mex.h>

#include "vec.c"

static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumContStates(    S, NX);      /* number of continuous states */
    ssSetNumDiscStates(    S, 0);       /* number of discrete states */
    ssSetNumInputs(        S, NU);      /* number of inputs */
    ssSetNumOutputs(       S, NY);      /* number of outputs */
    ssSetDirectFeedThrough(S, 0);       /* direct feedthrough flag */
    ssSetNumSampleTimes(   S, 1);       /* number of sample times */
    ssSetNumInputArgs(     S, 0);       /* number of input arguments */
    ssSetNumRWork(         S, 0);       /* number of real work vector elements */
    ssSetNumIWork(         S, 0);       /* number of integer work vector elements */
    ssSetNumPWork(         S, 0);       /* number of pointer work vector elements */
}

static void mdlInitializeSampleTimes(SimStruct *S)
{
   ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}

static void mdlInitializeConditions(double *x0, SimStruct *S)
{
  double x0tmp[NX] = X0;
  int i;

  for (i=0; i<NX; i++)
     x0[i] = x0tmp[i];
}

static void mdlOutputs(double *y, double *x, double *u, SimStruct *S, int tid)
{
  double dummydx[NX] = {0.};
  double ytmp[NY];
  int i;

  model(x, u, dummydx, ytmp);

   for (i=0; i<NY; i++)
      y[i] = ytmp[i];
}

static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{
}

static void mdlDerivatives(double *dx, double *x, double *u, SimStruct *S, int tid)
```

```
{
    double dummy_y[NY] = {0.};
    model(x, u, dx, dummy_y);
}

static void mdlTerminate(SimStruct *S)
{
}

#ifdef  MATLAB_MEX_FILE      /* Is this file being compiled as a MEX-file? */
#include "simulink.c"         /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"          /* Code generation registration function */
#endif
```

References

[1] BD. Exhaust Brake. Retrieved on: http://www.bd-vfi.com/Exhaust_Brakes/Turbo_Mount_Exhaust_Brake/, September 1999.

[2] Carlstrom. P. 1999. Volvo High Power Engine Brake. Retrieved on: http:/www.truck.volvo.se.

[3] Chen. C. and Tomizuka. M. 1995. Steering and Independent Braking Control for Tractor-Semitrailer Vehicles in Automated Highway Systems. Proceedings of CDC.

[4] Cummins. D. D. 1966. The Jacobs Engine Brake Application and Performance. *SAE*, (660740).

[5] Daimler-Chrysler. 1999. Hybrid Electric Vehicle. Matlab News Bulletin.

[6] Druzhinina M., Moklegaard L., Stefanopoulou A. 2000. Compression braking control for heavy duty vehicles. *Proc. of American Control Conference.*

[7] Druzhinina M., Moklegaard L., Stefanopoulou A. 2000. Speed gradient approach to longitudinal control of heavy duty vehicles equipped with compression brake. *Proc. of International Symposium on Advanced Vehicle Control.* Ann Arbor, Michigan.

[8] Druzhinina M., Moklegaard L., Stefanopoulou A. 2000. Speed gradient approach to longitudinal control of heavy duty vehicles equipped with variable compression brake. *Submitted to IEEE Transactions on Control System Technology.*

[9] Fitch. J. W. 1994. *Motor Truck Engineering Handbook.* SAE Inc., 4th. edition.

[10] A.L. Fradkov. 1979. Speed-Gradient scheme in adaptive control. *Autom. Remote Control*, Vol.40(9), pp.1333-1342.

[11] A.L. Fradkov, A.Yu. Pogromsky. 1999. *Introduction to Control of Oscillations and Chaos.* World Scientific.

[12] Gerdes C. J., Brown S. A., and Hedrick. K. J. 1995. Brake System Modeling for Vehicle Control. *Advanced Automotive Technologies*, pages 105–112.

[13] Guldner, J., and V.I. Utkin. 1995. Sliding mode control for gradient tracking and robot navigation using artificial potential fields. *IEEE Transactions on Robotics and Automation*, Vol. 11, no. 2, pp. 247-254.

[14] Guzzella L., Amstutz A. 1998. Control of Diesel Engines. *IEEE Control Systems*, October 1998, pp. 53-71.

[15] Heywood. J. B. 1988. *Internal Combustion Engine Fundamentals.* McGraw-Hill.

[16] Hu. H., Israel. M. A., and Vorih. J. M. 1997. Variable Valve Actuation and Diesel Engine Retarding Performance. *SAE*, (970342).

[17] Hu. H., Vorih. J., and Israel. M. 1998. Lost-Motion VVT Diesel Engine Retarder. *Automotive Engineering International*, Jan, 1998.

[18] Ioannou P., C.C. Chien. 1993. Autonomous intelligent cruise control. *IEEE Trans. on Vehicular Technology.* Vol. 42, pp.657-672.

[19] Ioannou. P. and Xu. Z. 1993. Throttle Control for Vehicle Following.

[20] P. Ioannou and Z. Xu. 1994. Throttle and Brake Control Systems for Automatic Vehicle Following," *PATH Research Report UCB-ITS-PRR-94-10*, 1994.

[21] Israel. M. A. and Hu. H. 1993. Impact of Ambient Parameter on Operating Efficiency of Compression Release Engine Retarder Systems on Heavy Duty Diesel Engines. *SAE*, (932972).

[22] Jacobs Vehicle System. 1999. Intebrake Engine Braking Systme for Signature 600. Retrieved on: http:/www.jakebrake.com/products/engine.

[23] Jacobs Vehicle Systems. 1999. "Driveline Brakes". Retrieved on: http://www.jakebrake.com/.

[24] John Deere. Sharing the Road Media Group. Retrieved on: http://www.deere.com/truckerimage/sharing.

[25] Kao. M., Moskwa. J. J. 1995. Turbocharged Diesel Engine Modeling for Nonlinear Engine Control and State Estimation. ASME Journal of Dynamic Systems, Measurement and Control, Vol. 117.

[26] Kanellapooulos. I. 1999. Integrated Longitudinal Control for Safe Automation of Commercial Heavy Vehicles. Proposal for the awarded MOU 393.

[27] Maciuca D. B. and Heidrick. K. J. 1998. Nonsmooth Estimation and Adaptive Control with Application to Automotive Brake Torque. Proceedings of ACC.

[28] McMahon D.H., Hedrick J.K. and S.E. Shladover. 1990. Vehicle Modelling and Control for Automated Highway Sysytems.

[29] Moklegaard L., Druzhinina M., Stefanopoulou A. 2000. Brake valve timing and fuel injection: a unified engine torque actuator for Heavy-Duty Vehicles. *5th International Symposium on Advanced Vehicle Control*, Ann Arbor.

[30] Moklegaard. L., Schmidt. J., and Stefanopoulou. A. G. 2000. Transition from Combustion to Variable Compression Braking. *SAE*, (2000-01-1228).

[31] Novak. J. M. 1977. Simulation of the Breathing Process and Air-Fuel Ratio Distribution Characteristics of Three-Valve, Stratified Charge Engines. *SAE*, (770881).

[32] Raza. H., Xu. Z., and Yang. B. 1994. Brake Modeling for AVCS Application. Technical Report 94-01-01. PATH.

[33] Ribbens. W. B. 1992. *Understanding Automotive Electronics*. SAMS. 4th. edition.

[34] Seborg. D. E., Edgar. T. F., and Mellichamp. D. A. 1989. *Process Dynamics and Control*. John Wiley & Sons, Inc.

[35] R. Sepulchre, M. Jankovic, P. Kokotovic. 1997. *Constructive Nonlinear Control*. Springer.

[36] Shigley. J. E. and Uicker. J. J. 1995. *Theory of Machines and Mechanisms*. McGraw-Hill. 2nd. edition.

[37] Shladover. S. E., Desoer. C. S., Hedrick. J. K., Tomizuka. M., Walrand. J., Zhang. W. B., McMahon. D., Peng. H., Sheikholeslam. S., and McKeown. N. 1991. Automatic Vehicle Control Developments in the PATH Program. *IEEE Trans. on Vehicular Technology*. 40(1):114–130.

[38] Skogestad. S. and Postlethwaite. I. 1996. *Multivariable Feedback Control Analysis and Design.* John Wiley & Sons.

[39] A. Stotsky, B. Egart and S. Eriksson. 1999. Variable structure control of engine idle spped with estimation of unmeasurable disturbances. *Proceedings of the 38th CDC.* Phoenix.

[40] Y. Tan, A. Robotis, I. Kanellakopoulos. 1999. Speed control experiments with an Automated Heavy Vehicle. *Proceedings of the 1999 IEEE.* International Conference on Control Applications. Hawai'i.

[41] Watson. N., Marzouk. M. 1977. A Nonlinear Digital Simulation of Turbocharged Diesel Engines Under Transient Conditions. *SAE*: (770123).

[42] Watson. N. 1984. Dynamic Turbocharged Diesel Engine Simulator for Electronic Control System Development. ASME Journal of Dynamic Systems, Measurement and Control. Vol. 106.

[43] Yanakiev. D., Eyre. J., and Kanellakopoulos. I. 1997. Longitudinal Control of Heavy Vehicles with Air Brake Actuation Delays. Proceedings of ACC.

[44] Yanakiev. D., Eyre. J., and Kanellakopoulos. I. 1998. Longitudinal Control of Automated CHV's with Significant Actuator Delays. Technical Report UCB-ITS_PRR-98-15. PATH.

[45] Yanakiev. D. and Kanellakopoulos. I. 1995. Engine and Transmission Modeling. California PATH Technical Note 95-6.

[46] D. Yanakiev and I. Kanellakopoulos. 1996. Speed tracking and vehicle follower control design for Heavy-Duty Vehicles. *Vehicle System Dynamics*, Vol. 25. pp. 251-276.

[47] Yanakiev. D. and Kanellakopoulos. I. 1997. Longitudinal Control of Automated CHV's with Significant Actuator Delays. Proceedings of CDC.