Minimizing Fuel Use During Power Transients for Naturally Aspirated and Turbo Charged Diesel Engines

Tomas Nilsson[a], Anders Fröberg[b], Jan Åslund[c]

[a] *Department of Electrical Engineering, Linköping University, Linköping, Sweden, (email: tnilsson@isy.liu.se)*
[b] *Volvo CE, Eskilstuna, Sweden, (email: anders.froberg@volvo.com)*
[c] *Department of Electrical Engineering, Linköping University, Linköping, Sweden, (email: jaasl@isy.liu.se)*

Technical Report: LiTH-ISY-R-3077

Abstract

Recent development has renewed the interest in drivetrain concepts which gives a higher degree of freedom by disconnecting the engine and vehicle speeds. This freedom raises the demand for active control, which especially during transients is not trivial, but of which the quality is crucial for the success of the drivetrain concept. In this work the fuel optimal engine operating point trajectories for a naturally aspirated and a turbocharged diesel engine, connected to a load which does not restrict the engine speed, is derived, analysed and utilized for finding a suboptimal operating point trajectory. The analysis and optimization is made with dynamic programming, Pontryagin's maximum principle and a suboptimal strategy based on the static optimal operating points. Methods are derived for using Pontryagin's maximum principle for finding the optimal operating point trajectories, for simple load cases. The time needed for computation is reduced a factor $1000 - 100$, depending on engine layout, compared to dynamic programming. These methods are only applicable to very simple load cases though. Finally, a suboptimal calculation method which reduce the time needed for computation a factor > 1000 compared to dynamic programming, while showing a $< 5\%$ increase in fuel consumption compared to the optimal, is presented.

1. Introduction

1.1. Background and motivation

Faster, smaller and cheaper computers have created the opportunity for more intricate control of mechanical systems, or even the introduction of new mechanical solutions that would have been unfeasible without a high level of control. In the field of vehicle engineering this can be seen in the recent diversification of drivetrain architectures [1]. The motivation for altering the drivetrain is often to reduce the fuel consumption, for environmental or economical reasons. It is easy to realize that the fuel consumption also depends on the driving cycle in which the vehicle operates [2].

The study presented in this report is motivated by wheel loader operation, and the distinct properties of the operation of such machines. For wheel loaders there are no standardized driving cycle, but it is clear that the common operation is highly transient [3] both in power requirement and in vehicle speed. This is exemplified by the scaled engine output in Figure 1, which has been recorded during two consecutive loading cycles.

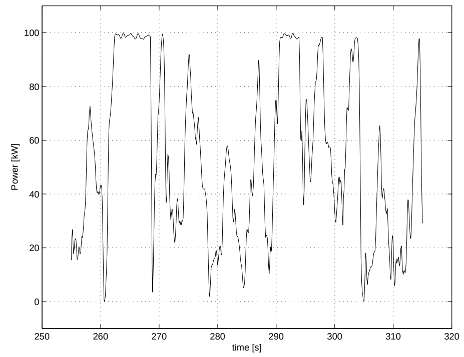


Figure 1: Power consumption of a wheel loader performing two short loading cycles

The drivetrain of the in-production reference vehicle uses a diesel engine, a torque converter and an automatic gearbox. This solution has the advantage that it is mechanically robust since the torque converter provides some disconnection of the wheels from the engine, and that it automatically adapts to changes in torque. The drawback is that there is always some slip in the converter, which reduces the efficiency. The low efficiency is the motivation for investigating other types of transmissions for these machines. Any alternative transmission must be able to handle all the distinct features of the operation. The frequent operation at very low speeds indicates that some type of continuously (or infinitely) variable transmission (CVT), such as a diesel-electric solution might be suitable. The introduction of such a layout increases the degree of freedom in the control, and especially allows for a free choice of engine speed, independent of the vehicle speed. The choice of engine speed during transients however, is not trivial. The extremely transient operation of wheel loaders, along with new possibilities of realizing optimal operation, motivates further examination of optimal and predictive control. This report therefore focuses on the derivation of the fuel optimal engine speed trajectories during power transients.

1.2. Previous work

There have been some work done on advanced wheel loader transmission control, but mainly in the fields of low level actuator control [4], autonomous vehicles [5] [6], and hybrid-electric powertrains with heuristic controls [7] [8]. There is also a vast amount of research on similar drivetrains for on-road passenger vehicles. Most of these use heuristic control laws [9] [10] or some variant of the ECMS [11] approach [12] [13]. Apart from these, there are articles such as [14] and [15] in which optimal trajectories are derived, but not thoroughly explained. In [16] a thorough investigation of the optimal solution is made, but only for a fully stochastic future load.

Since it in general is optimal to operate at a stationary point during static conditions, the online optimization might only require prediction at transients, and then with a short horizon. Some proposals on how to achieve this can be found in [17] [18] [19]. In case the vehicle is made autonomous, as proposed by [5] [6], the controller may also inform the optimizer about upcoming actions.

1.3. Problem outline

Transmissions that enables higher efficiency through higher controllability are for example belt type CVTs or hydrostatic or electric drives. These can all be configured in numerous ways to emphasize desired properties. This makes it impossible to make a general analysis that includes any detail of the transmission. Since transients are a fundamental part of wheel loader usage, this report is made to provide deeper understanding of the mechanisms behind the fuel optimal solutions during transients, without obscuring these by including any possible restrictions imposed by the transmission. This is done by subjecting the engine model to a load in the form of a non-stationary output power, and use different methods for analyzing the fuel optimal solution.

2. System setup

As a first approximation the powertrain of a CVT vehicle can be divided into one power producing and one power consuming part. In a diesel electric transmission the partitioning could be made at the electric connection by using electric power instead of voltage and current, in a hydraulic hybrid it could be made by using hydraulic power instead of pressure and flow, and in a belt type CVT it could be made by using belt power instead of belt force and speed. It is assumed here that the device has no maximum or minimum gear ratio. If such a partitioning can be made, any driving cycle can be translated, including efficiencies on the power consuming side, to an output power trajectory $P_{load}(t)$. The efficiencies in the power producing side of the transmission, see Figure 2, can be included in the engine efficiency.

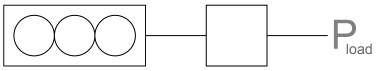


Figure 2: The system consists of an engine, the engine side of an infinitely variable transmission (e.g. an electric generator) and an output power.

This report is based on the papers [20], [21] and [22], which use engines with different maximum output powers. In this report the engine from [22] is used. The report treats both a naturally aspirated and a turbocharged engine. The differences between the setups are mentioned as they appear. The naturally aspirated engine is referred to as the NA-engine, while the turbocharged is referred to as the TC-engine.

2.1. Engine model

The engine speed ω_e dynamics is modeled as an inertia I_e which is affected by the engine torque T_e and a load power P_{load}.

$$\frac{d\omega_e(t)}{dt} \cdot I_e = T_e(t) - \frac{P_{load}(t)}{\omega_e(t)} \tag{1}$$

The engine torque T_e depends on fuel mass per injection m_f and engine speed ω_e according to a quadratic Willan's model, as described in [23]. Introduce the lower heating value q_{lhv}, the number of cylinders n_{cyl}, the number of strokes per injection n_r and the parameters $\eta_{e00}, \eta_{e01}, \eta_{e02}, \eta_{e10}, \eta_{e11}$ and define

$$A = \frac{q_{lhv} n_{cyl}}{2\pi n_r} \tag{2}$$

$$\eta_e = \eta_{e0} - \eta_{e1} m_f \tag{3a}$$

$$\eta_{e0} =\eta_{e00} + \eta_{e01}\omega_e + \eta_{e02}\omega_e^2 \tag{3b}$$

$$\eta_{e1} = \eta_{e10} + \eta_{e11}\omega_e \tag{3c}$$

$$\eta_{eL} = \eta_{eL0} + \eta_{eL2}\omega_e^2 \tag{3d}$$

The Willan's model, expanded with an additional torque loss T_t caused by lack of air intake pressure, can then be described by Equation (4). The torque loss T_t is introduced for the modeling of the turbocharged engine, and for the naturally aspirated engine this loss is zero $T_t = 0$.

$$T_e = A \cdot \eta_e \cdot m_f - \eta_{eL} - T_t \tag{4}$$

The engine is also subject to the state and control restrictions

$$\begin{array}{l} \omega_{e,min} \leq \omega_e \\ 0 \leq m_f \\ T_e \leq T_{e,max}(\omega_e) \end{array} \tag{5}$$

2.2. Turbocharger model

The torque loss T_t is caused by low air intake pressure, a pressure which depends on the rotational speed of the turbocharger. The turbocharger speed is assumed to be a first order dynamic system with the time constant $\tau_t(\omega_e)$ and an asymptotic speed that is a function of ω_e, m_f. The dynamic relations are expressed in the corresponding asymptotic and dynamic air intake pressures. Denote the asymptotic intake pressure by $p_{t,set}$ and the time dependent pressure by p_t. Introduce the model and effeiciency parameters $\xi_{\tau 0}$, $\xi_{\tau 1}$, ξ_{t1}, ξ_{t2}, ξ_{t3}, η_{t10}, η_{t11}, η_{t20} and η_{t21} and define

$$\tau_t = \xi_{\tau 0} + \xi_{\tau 1}\omega_e \tag{6a}$$

$$p_{t,set} = \xi_{t1}\omega_e + \xi_{t2}m_f + \xi_{t3} \tag{6b}$$

$$\eta_{t1} = \eta_{t10} + \eta_{t11}\omega_e \tag{6c}$$

$$\eta_{t2} = \eta_{t20} + \eta_{t21}\omega_e \tag{6d}$$

The pressure dynamics can then be described by

$$\frac{dp_t(t)}{dt} \cdot \tau_t(\omega_e) = p_{t,set}(\omega_e, m_f) - p_t(t) \tag{7}$$

By defining $p_{t,off} = p_{t,set}(\omega_e, m_f) - p_t$ the torque loss can then be described by

$$T_t = \begin{cases} \eta_{t1}(\omega_e) \cdot p_{t,off}^2 + \eta_{t2}(\omega_e) \cdot p_{t,off} & \text{if } p_{t,off} > 0 \\ 0 & \text{if } p_{t,off} \leq 0 \end{cases} \tag{8}$$

2.3. Efficiency definitions

The quasi-static peak efficiency points Σ are defined as the (ω_e, T_e) that maximize (9a) as a function of P_{load} under the restrictions (5) and $\frac{d\omega_e}{dt} = \frac{dp_t}{dt} = 0$ as described by the Equations (9).

$$\eta_{e,static} = \frac{P_{load}}{P_{m_f}} = \frac{T_e\omega_e}{\omega_e A m_f} \tag{9a}$$

$$\omega_{e,\Sigma}(P_{load}) = \underset{\omega_e}{\text{argmax}}\eta_{e,static}(P_{load}) \tag{9b}$$

$$m_{f,\Sigma}(P_{load}) = \underset{m_f}{\text{argmax}}\eta_{e,static}(P_{load}, \omega_{e,\Sigma}) \tag{9c}$$

The Equations (9) also define $T_{e,\Sigma} = T_e(\omega_{e,\Sigma}, m_{f,\Sigma})$. Individual points along the line Σ is referred to as (quasi) static optimal operating points or SOOPs.

3. Problem statement

The problem studied is the minimization of the total amount of fuel used, according to Equation (10)

$$\min \int_0^T A\omega_e m_f \mathrm{d}t \tag{10}$$

while fulfilling the engine dynamics Equation (1), the constraints (5) and, in case the engine is turbocharged, the turbo dynamics (7). This also means that no deviations from the output load trajectory $P_{load}(t)$ is allowed.

3.1. Load cases

In Equation (1) the time dependent load $P_{load}(t)$ is introduced. In this report two different types of loads are used. The first type is from measurements in a short loading cycle, 'DDP *sc*' and a long loading cycle, 'DDP *lc*'. The total output power is calculated from the measured wheel torque and speed, and hydraulic pressure and flow. These load cases are presented in Figure 3.

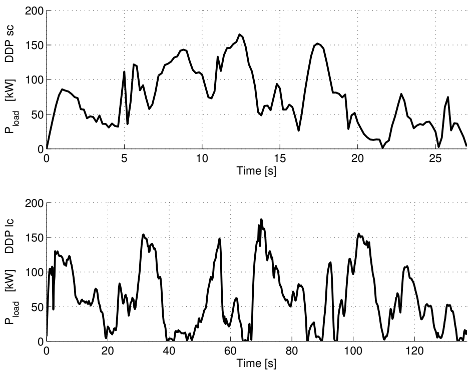


Figure 3: The output power trajectories in the load cases 'DDP *sc*' and 'DDP *lc*'.

The other type is artificial load cases, and consists of the four pulse and step cases presented in Table 1. The 'DDP *sc*' and 'DDP *lc*' load cases are applied to both engine setups while the pulse load cases are used for the NA-engine and the steps load cases are used for the TC-engine. In all four artificial load cases the time before the first and after the last steps are selected so that an increase in any of the times would not affect the transient optimization result. The time scales in the pulse load cases are selected so that in the slow pulse the engine has time to settle at the static optimal operating point (SOOP) of the intermediate output power, while in the quick step it does not. Due to the increased complexity of the TC-engine, this is only subjected to the single step load cases. The power levels in these load cases are selected so that the low step is between two SOOPs on the minimum engine speed limit, while both of the SOOPs of the high step are above this limit ($\sim 85kW$).

Table 1: Stylized load cases for engine-generator set.

Name	Load case: Power(Duration)
Slow pulse	$100kW(5s)$-$180kW(5s)$-$100kW(5s)$
Quick pulse	$100kW(5s)$-$180kW(0.8s)$-$100kW(5s)$
Low step	$50kW(5s)$-$80kW(5s)$
High step	$100kW(5s)$-$180kW(5s)$

4. Optimization Methods

4.1. General problem statement

Introduce the states $x(t)$ of the system, the decision variables, or control signals, $u(t)$ and the time dependent, non-controllable, disturbance signals $w(t)$. Here the only disturbance signal is the applied load. The problem studied in this report can then be stated as

$$\begin{aligned}&\min_{u \in U} \left\{ J_N(x(T)) + \int_0^T G(x,u,w)\mathrm{d}t \right\} \\ &\dot{x} = F(x(t),u(t),w(t)) \\ &x(0) = x_0\end{aligned} \tag{11}$$

along with posible state and control constraints. This problem is, regardless of the timespan, equivalent to an infinite dimension optimization problem. The problem is in general discretized for computerized numerical solving, transforming the problem into a large, but finite, dimensional optimization problem

$$\begin{aligned}&\min_{u \in U} \left\{ J_N(x(T)) + \sum_{k=0}^{N-1} g_k(u_k,x_k,w_k) \right\} \\ &x_{k+1} = f(x_k,u_k,t), \quad k = 0,\ldots,N-1\end{aligned} \tag{12}$$

4.2. Dynamic programming (DP)

Dynamic programming is a recursive method for solving optimization problems which develop in stages, such as a discrete time. According to [24] and [25] the recursion can be stated as

$$J_k(x_k) = \min_{u \in U} \left\{ g(x_k,u_k,w_k) + J_{k+1}(x_{k+1}(x_k,u_k,w_k)) \right\} \tag{13}$$

The implementation of the recursion as an algorithm includes a strategic choice. Denote the discretized states $x \in X$. The 'cost-to-go', J_{k+1}, is then only calculated and stored at the grid points $x_{k+1} \in X$, and is not explicitly known for $x_{k+1} \notin X$. The method selected for handling this highly affects the calculatory effort. Three possible choices are presented here.

If the function $x_{k+1}(x_k,u_k,w_k)$ is invertible, that is if $u_k(x_k,w_k,x_{k+1})$ is well defined, then $g + J_{k+1}$ can be evaluated for each $\{x_k,x_{k+1}\} \in X$ combination. With this choice the calculatory effort increase with the square of the size of X but is independent of the controls. If inverting $x_{k+1}(x_k,u_k,w_k)$ is not possible or desirable (for example if X is large) $x_{k+1}(x_k,u_k,w_k)$ can be calculated for the discretized $u \in U$, not requiring that $x_{k+1} \in X$. Then $\tilde{u}_k(x_k,w_k,x_{k+1} \in X)$ can be found by interpolation among these u_k, followed by the calculation of $g(x_k,\tilde{u}_k,w_k)$. Another option is to make the same calculation of $x_{k+1}(x_k,u_k,w_k)$, but to determine $\tilde{J}_{k+1}(x_{k+1}(x_k,u_kw_k))$ by interpolation among the $J_{k+1}(x_{k+1} \in X)$. In this case the calculatory effort increase linearly with the number of possible state and control combinations. In this thesis the third option is used, producing the following algorithm

```
For $x_N \in X_N$, declare $J_N(x) = J_N$
for $k = N-1,\ldots,1$ do
    For each $x_k \in X_k$, simulate $\frac{dx}{dt}$ for $t_k$ to $t_{k+1}$ for all $u \in U$ to find $x_{k+1}(x_k,u,w_k)$
    For each $x_k \in X_k$
        $J_k(x_k) = \min_{u \in U} \big(g(x_k,u,w_k) + \tilde{J}_{k+1}(x_{k+1}(x_k,u,w_k))\big)$   (14)
    with $\tilde{J}_{k+1}(x_{k+1})$ interpolated from $J_{k+1}(x_{k+1} \in X)$
end for
```

This first part establishes a cost-to-go map $J(x \in X, t)$. In the following part the optimal trajectory $x^*(t), u^*(t)$ is calculated

```
Select an initial state $x_0^* = x_0$
for $m = 1,\ldots,N$ do
    For $x_{m-1}^*$, simulate $\frac{dx}{dt}$ for $t_{m-1}$ to $t_m$ for all $u \in U$ to find $x_m(x_{m-1}^*,u)$
    Select
        $u_{m-1}^* = \underset{u \in U}{\mathrm{argmin}}\big(g(x_{m-1}^*,u,w_{m-1})\mathrm{d}t + ...$
        $+\tilde{J}_m(x_m(x_{m-1}^*,u,w_{m-1}))\big)$   (15)
    with $\tilde{J}_m(x_m)$ interpolated from $J_m(x_m \in X)$
    $x_m^* = x_m(x_{m-1}^*,u_{m-1}^*,w_{m-1})$
end for
```

This second part also indicates how DP can be used to implement an optimal state feedback scheme. In each repetition of the for-loop the optimal control action u_{m-1}^* is calculated, depending on the state x_{m-1}^*. Here the state x_{m-1}^* is found by simulation, but in a feedback application the actual state of the system at $t = m-1$ would be used instead. If there is then an unexpected state disturbance so that $\hat{x}_{m-1} \neq x_{m-1}^*$, in which $\hat{x}$ is the actual state of the system, the algorithm will find the control that minimizes the cost-to-go from this state $\hat{x}_{m-1}$. Apart from this attractive property, the method also guarantees that if a solution is found, this is the global optimum. This does however require that the grids are sufficiently dense, not least to avoid infinite cost spread [26]. A well written introduction to dynamic programming can be found in [27], which also mentions some tricks and pitfalls.

4.3. Pontryagin's maximum principle (PMP)

Pontryagin's maximum (or minimum) principle is a condition necessary for optimality. Before the condition is stated, a function called the Hamiltonian is introduced

$$H = G(x(t), u(t), w(t)) + \lambda^T(t) F(x(t), u(t), w(t)) \tag{16}$$

in which G and F is the cost and dynamics functions from (11) and λ is a set of continuous functions with one component corresponding to each of the components of x. Then the Pontryagin's maximum principle, which was presented in [28] and is described and used in [29], state that for x^*, u^* to be optimal, λ^* must exist and

$$H(x^*, u^*, w, \lambda^*) \leq H(x^*, u, w, \lambda^*) \quad \forall u, t \in [t_0, T] \tag{17}$$

along with boundary conditions for λ^*, which depend on whether the final time T is fixed or subject of optimization, must be fulfilled. By differentiating H this condition can be rewritten as a set of necessary conditions. For the unconstrained problem (11)

$$\frac{\partial H}{\partial u} = 0 \tag{18a}$$

$$\frac{\partial H}{\partial x} = -\dot{\lambda} \tag{18b}$$

$$\frac{\partial H}{\partial \lambda} = \dot{x} \tag{18c}$$

$$x(0) = x_0, \quad \lambda(T) = \frac{\partial J_N}{\partial x}(x^*(T)) \tag{18d}$$

must be fulfilled for x^*, u^* to be optimal. Condition (18c) is trivially fulfilled, as can be seen by differentiating (16). If the problem includes state or control constraints the Hamiltonian must be expanded, but the conditions (18) are sufficient for the analysis in Section 7.

4.4. Application of optimization

The application of dynamic programming to this problem is straightforward. The cost to be minimized is the total amount of fuel used. In general this cost formulation will cause all energy stored in the system to be drained at the end of the cycle. Here this would be seen as the terminal engine speed approaching $\omega_{e,min}$, regardless of the terminal output power. Especially for output power steps and pulses, it is instead desired that the engine settles at the SOOP corresponding to the terminal output power. Since the energy in the system increase with increasing $\omega_e(T), p_t(T)$, introducing a J_N with a sufficient penalty for $\omega_e(T) < \omega_{e,\Sigma}(T)$, $p_t(T) < p_{t,set}(\omega_{e,\Sigma}(T), m_{f,\Sigma}(T))$ is sufficient for bringing the end state toward the static optimal operating point. In this work the terminal cost

$$J_N = \begin{cases} 0 & \text{for } x_N \geq \Omega \\ \infty & \text{else} \end{cases} \tag{19}$$

is used, with Ω being equal to $x_\Sigma(P_{load}(T))$ except when stated otherwise. The states and controls for the two engine setups are collected in Table 2.

Also recapitulate the PMP conditions for these two setups. For the unconstrained TC-engine the Hamiltonian become

$$H = A\omega_e m_f + \frac{\lambda_1}{I_e}(T_e - \frac{P_{load}}{\omega_e}) + \frac{\lambda_2}{\tau_t}(p_{t,set} - p_t) \tag{20}$$

in which λ_1 is the adjoint variable related to the engine speed dynamics (1) and λ_2 is the adjoint variable related to the turbo pressure dynamics (7). This gives the following conditions necessary for optimality

$$\frac{\partial H}{\partial m_f} = A\omega_e + \lambda_1 \frac{\partial}{\partial m_f}\frac{d\omega_e}{dt} + \lambda_2 \frac{\partial}{\partial m_f}\frac{dp_t}{dt} = 0 \tag{21a}$$

$$\frac{\partial H}{\partial \omega_e} = Am_f + \lambda_1 \frac{\partial}{\partial \omega_e}\frac{d\omega_e}{dt} + \lambda_2 \frac{\partial}{\partial \omega_e}\frac{dp_t}{dt} = -\frac{d\lambda_1}{dt} \tag{21b}$$

$$\frac{\partial H}{\partial p_t} = \lambda_1 \frac{\partial}{\partial p_t}\frac{d\omega_e}{dt} + \lambda_2 \frac{\partial}{\partial p_t}\frac{dp_t}{dt} = -\frac{d\lambda_2}{dt} \tag{21c}$$

The optimality conditions for the unconstrained NA-engine can be retrieved by using $\lambda_2 = 0$ and disregarding equation (21c).

Table 2: Standalone engine states and controls.

	NA-engine	TC-engine
States X	ω_e	ω_e, p_t
Controls U	m_f	m_f

5. Engine map and static optimal solution

The quasi-static optimal line Σ is defined in (9). The Σ for the turbo engine is identical to that of the naturally aspirated engine, since $\dot{p}_t = 0 \Rightarrow T_t = 0$. This is a simple problem which can be solved either direct as the problem (9) or by solving the PMP problem with $\frac{d}{dt}[\omega_e, \lambda_1, p_t, \lambda_2] = 0$. The later is valid only when the solution fulfills $\omega_{e,min} \leq \omega_e$ though, since the state and control constraints is not included in the presented PMP formulation. The engine efficiency map is presented in Figure 4 along with $\omega_{e,min}$, $T_{e,max}$, output power ($T_e\omega_e$) lines and the Σ-line.

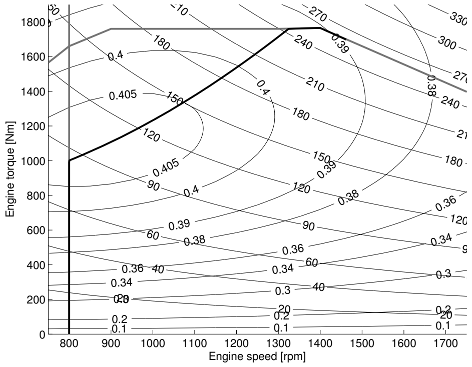


Figure 4: Engine map showing efficiency curves, output power lines with kW markings, state and control restrictions according to (5) and the quasi-static optimal line which for output powers below $\sim 85kW$ coincide with $\omega_{e,min}$ and above $\sim 240kW$ with $T_{e,max}$.

6. DP derived optimal trajectories

The optimal engine map trajectories for the pulse load cases for the NA-engine are presented in Figure 5. In both these cases the operating point moves in a counter clockwise direction; before the output power increase the operating point diverges toward high speed. When the step occur, the operating point motion changes direction toward the new static optimum by reducing the speed and increasing the torque. Before the power reduction the engine speed decreases, and at the step the motion changes direction and the speed increases while the torque falls and the operating point converges to the new static optimum.

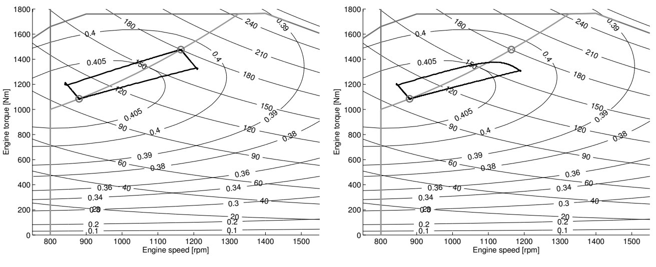


Figure 5: Engine map trajectories for the naturally aspirated engine in the slow (5(a)) and quick (5(b)) pulse load cases.

The optimal engine map trajectories for the steps load cases for the TC-engine are presented in Figure 6. Just as for the NA-engine, the engine speed increases before the step, and when the step occurs the direction of movement of the operating point changes. After the step the engine speed drops while the torque increases, converging toward the new static optimum. Both the trajectories displayed in Figure 6 are less smooth than those for the NA-engine. This is caused by a somewhat sparse discretization, which is motivated by the increase in calculation time caused by the added state.

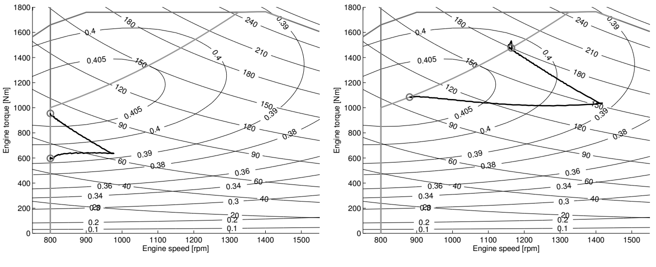
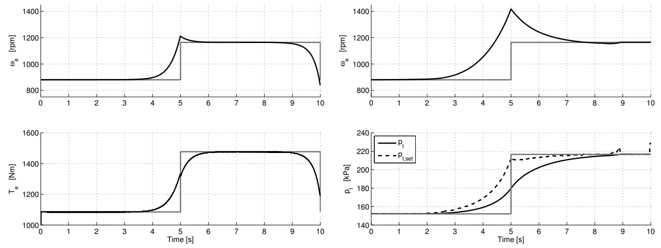


Figure 6: Engine map trajectories for the turbocharged engine in the low (6(a)) and high (6(b)) step load cases.

In Figure 7 the engine operation trajectories of the NA- and TC-engines are compared. Figure 7(a) shows the engine speed and torque during the first $10s$ of the slow pulse load case for the NA-engine and Figure 7(b) shows the engine speed and turbo-pressure during the high step load case for the TC-engine. The load case parts are identical, apart from that the NA-engine does not need to remain at the higher SOOP at $10s$. The NA-engine starts changing its state about one second before the step, while the TC-engine starts about three seconds before the step. Note that while both setups cause a speed overshoot, this is substantially larger for the TC-engine. Figure 7(b) shows that before the step, the increasing engine speed alters the turbo set-pressure so that it is roughly at the new static optimal level when the step occur. The actual pressure starts to increase as soon as the set pressure starts to change, but at the time of the step it still is far from the new static level. After the step, the pressure keeps increasing while the set pressure remains fairly constant and the engine speed falls back toward the new static optimum.



(a) Engine speed and torque in the slow pulse load case for the NA-engine

(b) Engine speed and turbo pressure in the high step load case for the TC-engine

Figure 7: Engine operation during steps for the NA- and TC-engines.

Figure 8 shows the engine map trajectories for the two engine setups in the short loading cycle. These trajectories should be compared to those in Figures 5 and 6. The movement is still counter clockwise, and the patterns of the movement remain, though the direction changes are less pronounced than in the solutions for the steps and pulses load cases since the output power changes are more ramped. The engine speed is generally higher for the TC-engine ($972rpm$ mean) than for the NA-engine ($861rpm$ mean), which is caused by the need for keeping the turbo pressure up. It should be noted that this is despite having access to perfect prediction of future load. Note that the initial operating point for the TC-engine is at a much higher engine speed than for the NA-engine. The initial conditions $x(t_0)$ are selected so that the results could be readily used for evaluation of the suboptimal methods described in Section 8.

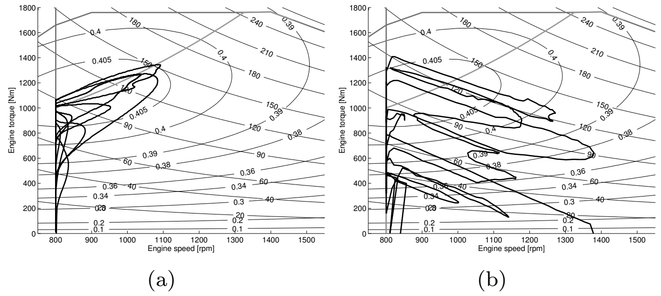


Figure 8: Engine map trajectories for the naturally aspirated (8(a)) and the turbocharged (8(b)) engine in the 'DDP sc' cycle.

7. PMP trajectory derivation

The solution to a DO problem must fulfill the conditions stated by Pontryagin's maximum principle (PMP). Section 7.1 analyze the NA-engine step/pulse results presented in Section 6 using these conditions. In Section 7.2 this analysis is utilized for developing a method for deriving the same optimization results. Section 7.3 expands this method for application on the TC-engine.

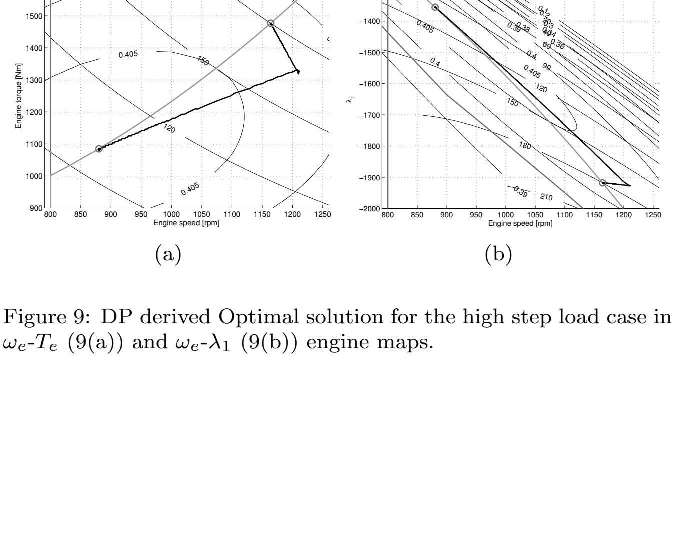
The PMP formulation in Section 4.3 does not include the constraints (5). A solution to the unconstrained problem (10) for a specific load $P_{load}(t)$ is optimal also for the constrained problem if and only if it does not violate the constraints (5). It is obvious that solutions for the unconstrained problem for steps to or from loads with $\omega_{e,\Sigma}(P_{load}) = \omega_{e,min}$ will violate these constraints. Therefore this section only treat load cases with $\omega_{e,\Sigma}(P_{load}) > \omega_{e,min}$.

7.1. Analysis of optimization results

This analysis treats the high step load case, which is identical to the first part of the slow pulse load case, applied to the NA-engine. The DP result for the slow pulse load case is presented in Figure 5(a), and the part used is presented again in Figure 9(a). Equation (21a) can be used for transformation of positions in an ω_e-T_e engine map into an ω_e-λ_1 engine map. For the NA-engine this relation can be rewritten as

$$\lambda_1 = \frac{\omega_e I_e}{2\eta_{e1} m_f - \eta_{e0}} \tag{22}$$

Figure 9(b) shows such a transformation of the map of Figure 9(a), including efficiency curves, output power lines with kW markings, the static optimal line Σ, the constraints (5) and the DP derived optimal operating point trajectory. In Figure 9(a) the trajectory starts at the lower left, moving toward the upper right, and when the step occur the direction of motion changes so that the maximum engine speed occur at the instant of the step. In Figure 9(b) this translates to initial movement toward the lower right and a change of direction of motion at the instant of the step.



(a) (b)

Figure 9: DP derived Optimal solution for the high step load case in ω_e-T_e (9(a)) and ω_e-λ_1 (9(b)) engine maps.

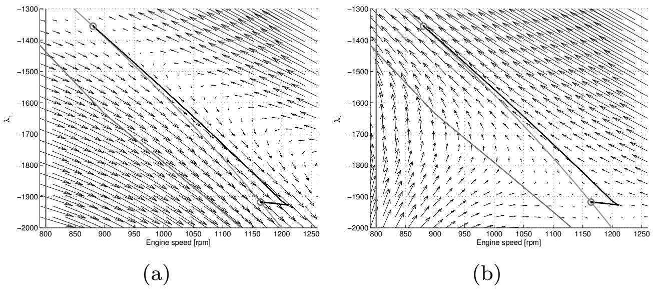
The dynamics of the adjoint variable $\lambda_1(t)$ is described by Equation (21b) (with $\lambda_2 = 0$). This equation can for the NA-engine be rewritten as

$$\dot{\lambda}_1 = -Am_f - \frac{\lambda_1}{I_e}\left(\frac{\partial T_e}{\partial \omega_e} + \frac{P_{load}}{\omega_e^2}\right) \tag{23}$$

in which

$$\frac{\partial T_e}{\partial \omega_e} = (\eta_{e01} + 2\eta_{e02}\omega_e - \eta_{e11} m_f)Am_f - 2\eta_{eL2}\omega_e \tag{24}$$

Since Equation (22) eliminates the only degree of freedom, all dynamics of the optimal solution is governed by Equations (1) (the engine speed) and (23) (the adjoint variable). The properties of a two dimensional autonomous dynamic system can be visualized by phase planes. The time dependent load means this system is not autonomous, though for piecewise constant loads, such as steps or pulses, the system can be regarded as piecewise autonomous. The phase planes for the system (1),(23) at the two output power levels of the high step load case are presented in Figure 10. The figure also shows the constraints (5), the static optimal line Σ and the DP-derived optimal trajectory, as shown in Figure 9(b).



(a) (b)

Figure 10: DP derived Optimal solution for the high step load case along with the $100kW$ (10(a)) and $180kW$ (10(b)) ω_e-λ_1 phase planes.

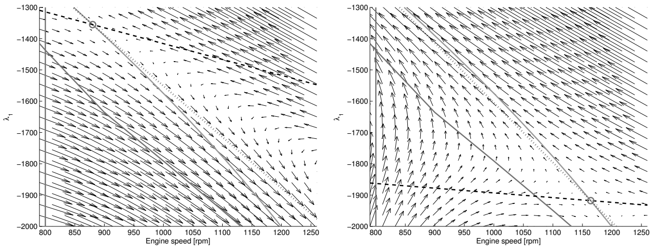
Figure 10 shows the dynamics behind the optimal solution for the high step load case. The first segment, the movement toward the lower right, occur when $P_{load} = 100kW$ and is therefore governed by the $100kW$ phase plane (Figure 10(a)), while the second segment, the approach of the second SOOP, is governed by the $180kW$ phase plane (Figure 10(b)). Section 7.2 starts with these phase planes and presents a method not only for visualizing but also for deriving the optimal solutions for similar load cases.

7.2. Optimal trajectory derivation for the NA-engine

This section shows how the reasoning in the previous section can be reversed and optimal trajectories be derived from the PMP conditions. The phase planes shown in Figure 10 indicate that, for each constant P_{load}, the SOOP is a saddle point of the corresponding autonomous system (25). This is confirmed by the eigenvalues of the Jacobian of this system, evaluated at the corresponding SOOP, since one is positive and the other is negative.

$$\frac{d}{dt}[\omega_e, \lambda_1]^T(P_{load}) \tag{25}$$

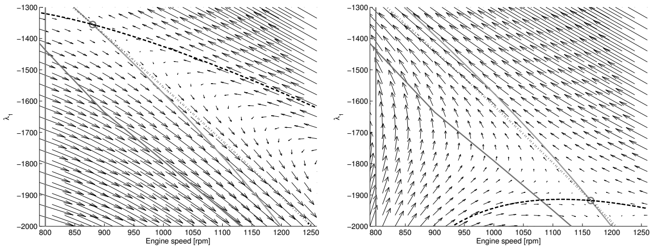
The unstable and stable manifolds of the autonomous system can, in a small region near the SOOP, be approximated by the eigenvectors of the Jacobian. The stable (dashed) and unstable (dotted) eigenvectors and the previously presented phase-planes corresponding to $P_{load} = 100kW$ and $P_{load} = 180kW$ are shown in Figure 11. More accurate approximations of the manifolds, valid outside the vicinity of the SOOP, can be obtained by simulations backward in time for the stable manifolds and forward in time for the unstable manifolds initiated from the SOOP with small, ε, disturbances in the directions of the eigenvectors. The result of such simulations, corresponding to the situations of Figure 11, are displayed in Figure 12.



(a)

(b)

Figure 11: Phase planes along with stable (dashed) and unstable (dotted) eigenvectors of the Jacobian of the dynamic system (25) with $P_{load} = 100kW$ (11(a)) and $180kW$ (11(b)).



(a)

(b)

Figure 12: Simulation derived stable (dashed) and unstable (dotted) manifolds of the system (25) with $P_{load} = 100kW$ (12(a)) and $180kW$ (12(b)).

The optimal operating point trajectory for an output power step (in this example $100kW - 180kW$) which starts and ends at the SOOPs of the initial and terminal output powers, must start by leaving the first SOOP along a path in the unstable manifold of the earlier autonomous system. At the instant of the step the operating point must switch to a path in the stable manifold of the later autonomous system. Since the trajectory must be continuous the operating point must be at an intersection of these manifolds at the instant of the step. In general there is only one such intersection, which is easily found from the simulated paths. When the point of intersection is found the excess parts of the simulated paths are cropped of and the time-scales of the simulations behind Figure 12 are adjusted so that a single, continuous, $\omega_e(t)$,$\lambda_1(t)$ trajectory is obtained. This trajectory is then the optimal solution. Graphically, this solution can be found by simply superposing Figure 12(a) with Figure 12(b) and cropping of excessive parts of the paths. Figure 13 shows the results as derived with this method (continuous) and with dynamic programming (dashed) for the upward and downward steps of the slow pulse load case. This solution can then be translated into an $\omega_e(t)$,$T_e(t)$ trajectory by Equation (22).

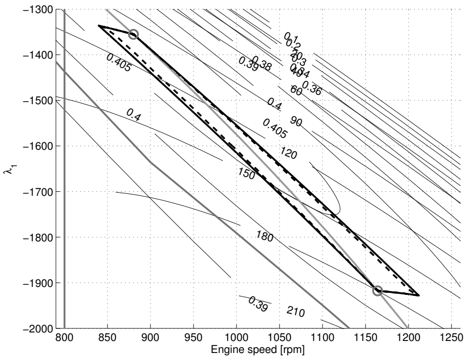
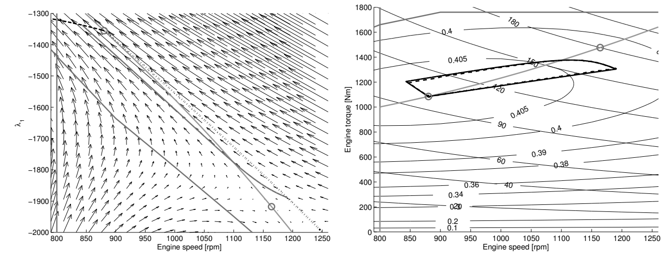


Figure 13: PMP (continuous) and DP (dashed) derived optimal solutions for the slow pulse load case.

This method can be expanded to somewhat more complicated load cases. If the case starts and ends with episodes of constant power, the optimal $\omega_e(t)$,$\lambda_1(t)$ trajectory must start with a leaving of the SOOP of the initial output power along the corresponding unstable manifold, and end with an approach of the SOOP of the terminal output power along the stable manifold. This is illustrated in Figure 14 by the solving of the quick pulse load case. This case consists of $5s$ at $100kW$, $0.8s$ at $180kW$ and finally $5s$ at $100kW$. The optimal trajectory must therefore start with a leaving of the $100kW$ SOOP along a path in the corresponding unstable manifold (dotted) and end by approaching the same SOOP along the stable manifold (dashed). Solving the quick pulse optimization problem therefore translates to finding a path in the $180kW$ phase plane, as shown in the figure, that starts on the dotted line, ends on the dashed line and has a transition time $t_T = 0.8s$. If the starting point of the transition is at t_i from the initial SOOP along the unstable manifold, the problem can be formulated as $\min_{t_i} |t_T - 0.8|$, which is locally convex, making the problem easily solved. The resulting transition trajectory is indicated in Figure 14(a) by the gray line. In Figure 14(b) this solution (continuous) is translated to an ω_e,T_e map and compared to the solution derived with DP (dashed).



(a) (b)

Figure 14: Illustration of the PMP-method for solving the quick pulse load case. Figure 14(a) shows the stable and unstable $100kW$ manifolds along with the $180kW$ phase plane and the $0.8s$, $180kW$ transition path. In Figure 14(b) the PMP (continuous) and DP (dashed) derived optimal solutions are compared.

7.3. Optimal trajectory derivation for the TC-engine

This section expands the method derived in the previous section for use with the TC-engine. The optimal solutions for the TC-engine is governed by the four dynamics Equations (26) and the static control relation (21a). The four dimensions of this problem means that phase planes can no longer be drawn and the problem can therefore not be solved graphically.

$$\frac{d}{dt}[\omega_e, \lambda_1, p_t, \lambda_2](P_{load}) \tag{26}$$

The formulation of the torque loss T_t in Equation (8) may cause discontinuities in the optimality conditions (21) due to the differentiation, which severely complicates simulation. One solution may be to approximate the discontinuities with a tangent function. In a step however it can instead be assumed that the intake pressure will not cross the discontinuity; p_t will fulfill $p_t < p_{t,set}$ in an upward step and $p_{t,set} < p_t$ in a downward step, so that for steps the discontinuity can be disregarded. In this section, just as in the previous, the upward high step load case is studied.

In the same way as for the NA-engine, the Jacobian of the system (26) is evaluated at the SOOPs of, in this example, $P_{load} = 100kW$ and $P_{load} = 180kW$ and the eigenvalues are calculated. These show that the SOOPs are saddle points, since two of the four eigenvalues are positive while the other two are negative. For the NA-engine, the optimization problem is easily solved since the trajectories simulated and presented in Figure 12 covers the entire stable and unstable manifolds within the reasonable engine operating region, and the point of intersection is easily found. For the TC-engine however, each of the manifolds are two dimensional. Calculation of the complete unstable manifold would therefore require infinitely many simulations, initiated from the SOOP with small disturbances in all directions that are combinations of the eigenvectors corresponding to the positive eigenvalues, and vice versa for the stable manifold. Recall however that the objective is not to find the manifolds, but only the trajectories within these manifolds that connect the SOOPs of the initial and terminal P_{load}. Since the manifolds are two dimensional and the state space is four dimensional, there is in general a single point at which these manifolds intersect, and therefore only one combination of eigenvectors that produce trajectories that intersect. Since the location of the intersection is unknown, the problem is reformulated as a problem of finding the combination of eigenvectors that minimizes the minimum distance between the simulated trajectories. Similar problems are treated for example in [30]. Denoting the initial and terminal output powers P_1 and P_2 and using the notation $v_{1,1}, v_{1,2}$ for the unstable eigenvectors corresponding to P_1 and $v_{2,1}, v_{2,2}$ for the stable eigenvectors corresponding to P_2 the problem is formulated as

$$\min_{s_1,t_1,s_2,t_2} \|X_1(P_1,t_1) - X_2(P_2,t_2)\|_2 \tag{27}$$

$$0 < [t_1, -t_2]^T, \quad 0 \leq [s_1, s_2]^T \leq 2\pi \tag{28}$$

in which

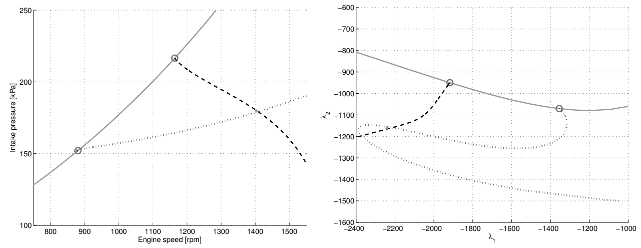
$$X_n = [\omega_e, \lambda_1, p_t, \lambda_2]^T(P_n, t_n), \quad n = 1, 2 \tag{29}$$

are simulated from $t_n = 0$ forward and backward in time with initial conditions that are small, ε, perturbations from the SOOPs according to

$$X_n(t_n = 0) = X_\Sigma(P_n) + \varepsilon\big(\sin(s_n)v_{n,1} + \cos(s_n)v_{n,2}\big), \quad n = 1, 2 \tag{30}$$

and the components of X_n in (27) being scaled with the average of the values of the component at the two SOOPs. Numerically this is solved as one external and one internal minimization problem. The external minimizes $\|X_1 - X_2\|_2$ over the disturbance direction combination s_1, s_2. Inside this, with s_1, s_2 given, $X_1(0<t_1)$,$X_2(t_2<0)$ is simulated and the minimum distance between the trajectories is determined by minimizing $\|X_1 - X_2\|_2$ over t_1, t_2. Each of the two internal simulations start at $t_1 = t_2 = 0$ and proceed until some state leave a predefined reasonable operating range. If a solution to the problem is found, the result of (27) should approach 0. The resulting point $X_1(t_1) \approx X_2(t_2)$ is then the intersection of the manifolds. This is the point at which the output power step occur and the operating point movement switch from one manifold to the other. Finally the times are shifted so that t_1 and t_2 coincide with the instant of the step. The result is a continuous operating point trajectory that start at $X_\Sigma(P_1)$, ends at $X_\Sigma(P_2)$ and has the step correctly placed in time.

The method is illustrated by the high step load case. Figure 15 shows the static optimal line (gray), the SOOPs (markers), the unstable (dotted) and stable (dashed) trajectories and a dark gray line which indicate the position of the minimum distance between the trajectories. Figure 16 shows the ω_e,T_e translated trajectories in an engine map. Figure 17 shows the time-adjusted unstable and stable engine speed and turbo pressure trajectories along with the DP-derived solution (gray). Typical calculation times experienced for finding this solution have been around 30s, which is considerably faster than the more than 2500s needed for finding the solution with dynamic programming. On the other hand, this method works only for load steps and, since the engine speed overshoots are larger for the TC-engine than for the NA-engine, at a narrow output power range.



(a) (b)

Figure 15: Intersecting stable (dashed) and unstable (dotted) trajectories for the high step load case in ω_e,p_t (Figure 15(a)) and λ_1,λ_2 (Figure 15(b)) maps. The minimum distance between the trajectories is marked with gray.

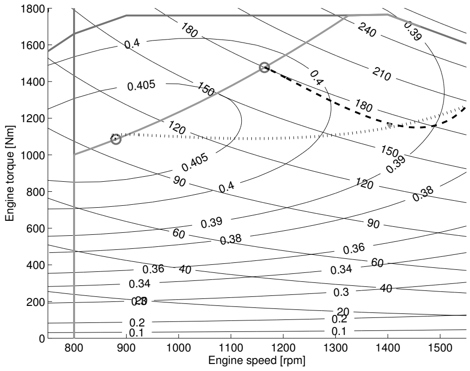


Figure 16: Intersecting stable (dashed) and unstable (dotted) trajectories for the high step case in an ω_e,T_e map. Note the minimum distance marker (gray).

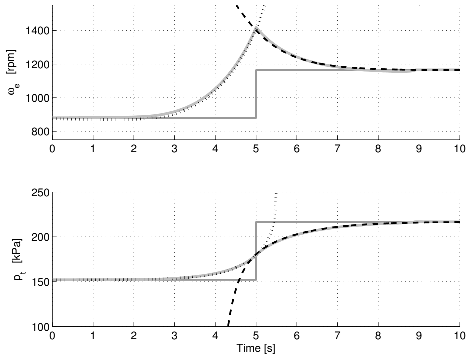


Figure 17: PMP-derived (dotted & dashed) compared to DP-derived (gray) solution for the high step load case. The dark gray lines indicate $X_\Sigma(P_{load}(t))$.

8. Suboptimal method development

8.0.1. Method for the NA-engine:

As mentioned, DP has several advantages but is slow while the PMP methods presented above are fast but very restrictive in which load cases can be treated. Another method which is fast and works for all load cases is desired, even if the resulting trajectories become suboptimal. Using $\omega_e(t) = \omega_{e,\Sigma}(P_{load}(t))$ is not possible, since output power steps would then imply engine speed steps. Inspiration for a method can instead be found in the optimal trajectories, for example in Figure 8. The operating point of the NA-engine seldom move far from the static optimal line Σ. A natural suboptimal strategy is to keep the operating point exactly on the line Σ at all times. Such a trajectory can be found by adding a large cost for deviation from this line to the DP algorithm, but solving this problem would be as computationally costly as solving the original problem. Instead start by redefining the static optimal line by introducing a small inclination in the minimum engine speed, so that at high torque the minimum speed is somewhat higher, to make $T_{e,\Sigma}(\omega_e)$ well defined. The rule

$$T_e(t) = T_{e,\Sigma}(\omega_e(t)) \tag{31}$$

then define the control signal, and thereby eliminate the only degree of freedom. The problem is therefore reduced from an optimization problem to finding the state and control trajectories that correspond to a set of admissible boundary conditions. Observe that as long as $T_{e,\Sigma}(\omega_e) \cdot \omega_e$ increase with increasing ω_e applying (31) will make the system unstable. This means that at the instant of an output power step the engine must already have exactly reached the terminal stationary operating point by a preceding divergence from the initial stationary operating point, initiated by a small disturbance. Since the system is always unstable it can easily be simulated backward in time from an arbitrary terminal engine speed, for example using the Euler method according to Equation (32).

$$\omega_{e,k-1} = \omega_{e,k} - \left(\frac{T_{e,\Sigma}(\omega_{e,k})\omega_{e,k} - P_{load}}{\omega_{e,k} I_e}\right) \mathrm{d}t \tag{32}$$

This method works well, as illustrated by Table 3, for all cases tested. The table shows fuel usage in the solutions derived with DP and the suboptimal method, along with typical calculation times experienced. The same $x(T)$ is used in both methods and the $x(0)$ from the suboptimal method is used as initial condition for the DP solving. The last row shows the relative increase in fuel consumption and reduction of calculation time for the suboptimal method compared to DP. Figure 18 shows the suboptimal and optimal engine speed and torque trajectories. The $\omega_{e,\Sigma}(P_{load}(t))$, $T_{e,\Sigma}(P_{load}(t))$ trajectories that would have been applicable and indeed optimal for an engine with zero inertia I_e are included as a reference. The figure shows that the engine speed reacts somewhat later to upcoming load changes in the suboptimal solution than in the optimal. The example is a cutout from the 'DDP *sc*' load case.

Table 3: Calculation effort and fuel usage with the suboptimal method.

	Fuel usage $[ml]$		Calculation time $[s]$	
	DDP '*sc*'	DDP '*lc*'	DDP '*sc*'	DDP '*lc*'
DP	152.8	675.9	1270	6480
Suboptimal	152.9	676.5	0.38	1.89
Relation	+.086%	+.099%	1 : 3340	1 : 3430

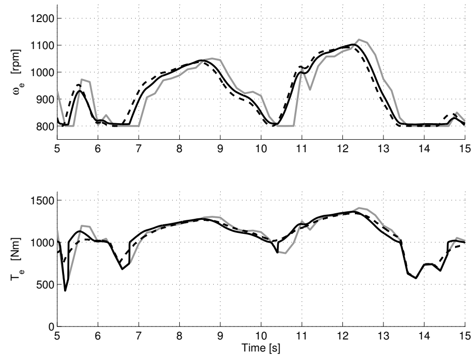


Figure 18: Engine speed and torque. Gray is static optimum $(\omega_{e,\Sigma}(P_{load}(t)),T_{e,\Sigma}(P_{load}(t)))$, continuous is suboptimal and dashed is optimal.

8.0.2. Method for the TC-engine:

The expansion to the TC-engine is not trivial. The turbocharger stable in the forward direction, so it appears unstable in the backward direction and cannot be included in the simulation (32). It is tempting to derive an $\omega_e(t), T_e(t)$ trajectory while disregarding $p_t(t)$, and then simulate (7) forward in time while compensating for T_t with increased m_f. Unfortunately this is not possible for a general load case for this engine. This is most obvious for an upward step between two SOOPs with $\omega_{e,\Sigma} = \omega_{e,min}$. With this method, and with a neglectable minimum speed inclination, a step in P_{load} requires a step in T_e, and thereby in m_f. Equations (6)-(8) indicate that the p_t dynamics prevents making arbitrarily big steps in T_e simply by steps in m_f. It is therefore necessary to increase p_t in preparation for upcoming output power steps and/or to use power from the engine inertia I_e. Preparatory increasing of p_t has to be done by altering the engine speed and torque trajectories, possibly deviating from the static optimal line. The following algorithm is therefore proposed:

1) Find $\omega_e(t)$,$m_f(t)$ either by backward simulation of (1) assuming $p_{t,off} = 0$ or by assuming $I_e = \tau_t = 0 \Rightarrow \omega_e T_e = P_{load}, p_{t,off} = 0$ with $T_e = T_{e,\Sigma}(\omega_e)$.

2) Using $\omega_e(t)$,$m_f(t)$ from 1), simulate (7) forward in time to find a first estimate of $p_t(t)$, and thereby also of $T_t(t)$.

3) Update $\omega_e(t)$,$T_e(t)$ by simulating (1) backward in time while adding the result form 2) to the load; $T_e(t) = T_{e,\Sigma}(\omega_e) - T_t(t) = \frac{P_{load}}{\omega_e} - \frac{d\omega_e}{dt} I_e$.

4) Update $m_f(t)$,$p_t(t)$,$T_t(t)$ by simulating p_t forward in time, in each step solving Equations (1)-(6) for m_f so that $T_e = \frac{P_{load}}{\omega_e} - \frac{d\omega_e}{dt} I_e$.

If I_e=τ_t= 0 is assumed in step 1), this step can be performed inside step 2). After step 4) a feasible $\omega_e(t)$,$p_t(t)$,$m_f(t)$ trajectory has been found. This method works well for all cases tested, as illustrated by Table 4. The table shows the fuel usage in the trajectories derived with DP and the suboptimal method, along with typical calculation times experienced. The same $x(T)$ is used in both methods and the $x(0)$ from the suboptimal method is used as initial condition for the DP solving. This is also the cause of the high initial engine speed in Figure 8(b). The last row shows the relative increase in fuel consumption and reduction of calculation time for the suboptimal method compared to DP.

Table 4: Calculation effort and fuel usage with the suboptimal method.

	Fuel usage $[ml]$		Calculation time $[s]$	
	DDP '*sc*'	DDP '*lc*'	DDP '*sc*'	DDP '*lc*'
DP	154.8	701.0	6800	38500
Suboptimal	157.2	725.2	2.10	10.2
Relation	+1.54%	+3.46%	1 : 3240	1 : 3800

An example of resulting engine speed and turbo pressure trajectories are compared to the optimal in Figure 19. The example is a cutout from the 'DDP *sc*' load case. The figure shows that while the suboptimal engine speed differs significantly from the optimal, the suboptimal turbo pressure trajectory is close to the optimal. Since the operating point is forced to leave the static optimal line, the engine map trajectories for the low and high steps load cases are also presented in Figure 20. In the high step load case the suboptimal and optimal trajectories are close. In the low step load case, just as in the 'DDP *sc*' case, the engine speed reacts later in preparation for upcoming loads in the suboptimal solution.

9. Discussions and comments

9.1. Dynamic programming

The dynamic programming optimization in this report is fairly straight-forward. The result for the naturally aspirated engine is a bit unexpected though; before output power steps it is optimal to accelerate or decelerate past the upcoming static optimal engine speed, and approach the new static optimum from the 'wrong' direction after the step. The motion of the engine operating point is counter clockwise in all cases studied, so that it travels toward higher engine speeds below the static optimal line and toward lower speeds above this line. This differs from

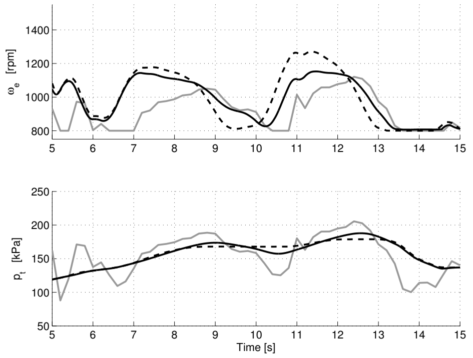
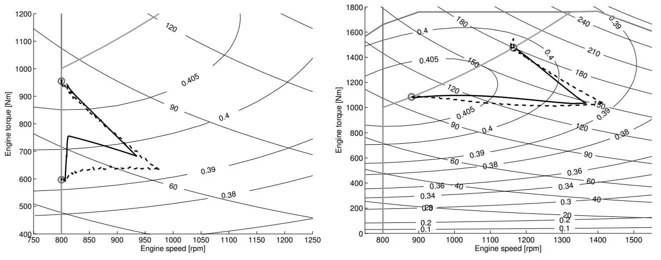


Figure 19: Engine speed and turbo pressure. Gray is static optimum $(\omega_{e,\Sigma}(P_{load}(t)), p_{t,\Sigma}(P_{load}(t)))$, continuous is suboptimal and dashed is optimal.



(a)

(b)

Figure 20: Suboptimal (continuous) and optimal (dashed) trajectories for the TC-engine in the low (20(a)) and high (20(b)) step load cases.

the result presented in [15], in which the initial operating point movement is in a clockwise direction. The main operating point motions in [15] however seems to be caused by a bad choice of initial and terminal states. In this paper the engine is forced to start and finish at the static optimal points corresponding to the initial and terminal output powers, and given sufficient time to move between these so that the trajectories would not change if more time were added to the beginning or the end of the load cases. The primary problem with DP, which is encountered in both engine setups but especially for the turbocharged engine, is the high calculatory effort. The most obvious way of countering this is to reduce the discretization grid densities, though care has to be taken to avoid large simulation errors and faulty infinite-cost spread (as mentioned in Section 4.2).

9.2. PMP based methods

The phase planes in Section 7.1 is used to validate the results derived with dynamic programming and to provide insight into the the mechanisms behind the trajectories. This insight is enhanced by the actual derivation of optimal trajectories in Section 7.2, and the expansion in Section 7.3 which show that the reasoning is valid also for the TC-engine. The actual solving of the dynamic optimization problems in this section is also fast, compared to dynamic programming. The treatment therefore provide an excellent pedagogic example of optimization with Pontryagin's maximum principle. The methods are however highly restrictive in the load cases which can be treated. The PMP formulation used does not include the state and control constraints (5) and the methods are only practically usable for output power steps or, for the NA-engine, slightly more complicated cases.

9.3. Suboptimal methods

The developed methods for finding suboptimal solutions works well for both of the engine setups. In both cases the time for finding a solution is reduced by a factor > 3000, while the amount of fuel required only increase by $< 0.1\%$ for the NA-engine and $< 5\%$ for the TC-engine. It should be noted that in both cases, and in particular for the TC-engine, finding even a feasible solution is not a trivial problem. The developed methods does not require analytic expressions neither for the engine efficiency nor for the static optimal line. The only requirements for the NA-engine are that $T_{e,\Sigma}(\omega_e)$ is well defined for all ω_e and that $T_{e,\Sigma}(\omega_e) \cdot \omega_e$ is strictly increasing with increasing ω_e, so that the rule (31) makes the system unstable.

References

[1] F. Buscemi, Trends in automobile transmissions, Gear Technology 23 (2006) 24–26.

[2] X. Zeng, Q. Wang, D. Song, Direct statistical analyses of vehicle's fuel consumption based on driving cycles, Journal of Hunan University Natural Sciences 37 (2010) 35–40.

[3] M. Starr, J. Buckingham, C. Jackson, Development of transient test cycles for selected nonroad diesel engines, American Society of Mechanical Engineers, ICE Division 32.

[4] J. Lennevi, Hydrostatic transmission control, design methodology for vehicular drivetrain applications, dissertation, Linköping University (1995).

[5] R. Ghabcheloo, M. Hyvönen, J. Uusisalo, O. Karhu, J. Jara, K. Huhtala, Autonomous motion control of a wheel loader, in: Proceedings of the ASME 2009 Dynamic Systems and Control Conference, ASME, 2009, pp. 1339–1346.

[6] N. Koyachi, S. Sarata, Unmanned loading operation by autonomous wheel loader, in: ICCAS-SICE 2009, IEEE, 2009, pp. 2221–2225.

[7] R. Zhang, D. Carter, A. Alleyne, Multivariable control of an earthmoving vehicle powertrain, experimentaly validated in an emulated working cycle, in: 2003 ASME International Mechanical Engineering Congress, ASME, 2003.

[8] S. Grammatico, A. Balluchi, E. Cosoli, A series-parallel hybrid electric powertrain for industrial vehicles, in: 2010 IEEE Vehicle Power and Propulsion Conference, IEEE, 2010, pp. 1–6.

[9] S. Liu, B. Paden, A survey of today's cvt controls, in: Proceedings of the 36th Conference on Decision and Control, IEEE, 1997, pp. 4738–4743.

[10] N. Srivastava, I. Haque, A review on belt and chain continuously variable transmissions (cvt): dynamics and control, Mechanism and Machine Theory 44 (2009) 19–41.

[11] A. Sciarretta, L. Guzzella, Control of hybrid electric vehicles, Control Systems, IEEE 27 (2007) 60–70.

[12] J. Liu, H. Peng, Control optimization for a power-split hybrid vehicle, in: Proceedings of the 2006 American Control Conference, IEEE, 2006, pp. 466–471.

[13] G. Paganelli, T. Guerra, S. Delprat, J. Santin, M. Delhom, E. Combes, Simulation and assessment of power control strategies for a parallel hybrid car, Proceedings of the Institution of

Mechanical Engineers, part D: Journal of Automobile Engineering 214 (2000) 705–717.

[14] D. Ambuhl, O. Sundstrom, A. Sciarretta, L. Guzzella, Explicit optimal control policy and its practical application for hybrid electric powertrains, Control Engineering Practice 18 (2010) 1429–1439.

[15] R. Pfiffner, L. Guzzella, Optimal operation of cvt-based powertrains, International Journal of Robust and Nonlinear Control 11 (11) (2001) 1003–1021.

[16] P. Rutquist, C. Brietholtz, T. Wik, An eigenvalue approach to infinite-horizon optimal control, in: Proceedings of the 16th IFAC World Congress, IFAC, 2005.

[17] B. Asadi, A. Vahidi, Predictive cruise control: utilizing upcoming traffic signal information for improved fuel economy and reduced trip time, IEEE Transactions on Control Systems Technology 19 (3) (2011) 707–714.

[18] D. Mitrovic, Reliable method for driving events recognition, IEEE Transactions on Intelligent Transportation Systems 6 (2005) 198–205.

[19] A. Pentland, L. Andrew, Modeling and prediction of human behavior, Neural Computation 11 (1999) 229–242.

[20] T. Nilsson, A. Fröberg, J. Åslund, Optimized engine transients, in: 7th IEEE Vehicle Power and Propulsion Conference, IEEE, 2011, pp. 1–6.

[21] T. Nilsson, A. Fröberg, J. Åslund, Optimal operation of a turbocharged diesel engine during transients, SAE International Journal of Engines 5 (2) (2012) 571–578.

[22] T. Nilsson, Optimal engine operation in a multi-mode CVT wheel loader, Tech. rep., Linköping University, liU-TEK-LIC-2012:32, Thesis No. 1547 (2012).

[23] G. Rizzoni, L. Guzzella, B. Baumann, Unified modeling of hybrid electric vehicle drivetrains, IEEE/ASME Transactions on Mechatronics 4 (1999) 246–257.

[24] R. Bellman, Dynamic Programming, Princeton University Press, 1957.

[25] D. Bertsekas, Dynamic Programming and Optimal Control, 3rd Edition, Vol. 1, Athena Scientific, 2005.

[26] O. Sundström, D. Ambühl, L. Guzzella, On implementation of dynamic programming for optimal control problems with final state constraints, Oil & Gas Science and Technology Rev. IFP 65 (1) (2010) 91–102.

[27] L. Guzzella, A. Sciarretta, Vehicle Propulsion Systems, 2nd Edition, Springer Verlag, 2007.

[28] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, E. Mishchenko, The Mathematical Theory of Optimal Processes, Interscience Publishers, 1962.

[29] A. Bryson, Applied Optimal Control; Optimization, Estimation and Control, Taylor and Francis, 1975.

[30] M. Dellnitz, O. Junge, B. Thiere, The numerical detection of connecting orbits, Discrete and Continuous Dynamical Systems - Series B 1 (2001) 125–135.