

Linköping Studies in Science and Technology. Dissertations.
No. 1636

Optimal Predictive Control of Wheel Loader Transmissions

Tomas Nilsson



Linköpings universitet
INSTITUTE OF TECHNOLOGY

Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden

Linköping 2015

Linköping Studies in Science and Technology. Dissertations. No. 1636

This is a Swedish Doctoral Dissertation.

A Doctor's degree comprises 240 ECTS credits (4 years of full-time postgraduate studies),
of which at least 120 ECTS credits constitute a doctoral dissertation.

Optimal Predictive Control of Wheel Loader Transmissions

ISBN 978-91-7519-171-3

ISSN 0345-7524

© 2015 Tomas Nilsson, unless otherwise noted. All rights reserved.

Tomas Nilsson
tomas.nilsson@liu.se
www.vehicular.isy.liu.se
Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

The **cover** illustrations show wheel loaders at work.

©2014 Volvo Construction Equipment,

Reproduced with the permission of Volvo Construction Equipment.

<https://images.volvoce.com>

Typeset with L^AT_EX 2_ε

Printed by LiU-Tryck, Linköping, Sweden 2015

To my wife Malin and my son Vilhelm

Abstract

The transmissions of present heavy wheel loaders are in general based on torque converters. The characteristics of this component suits these machines, especially in that it enables thrust from zero vehicle speed without risk of stalling the engine, without active control. Unfortunately, the component also causes losses which might become large compared to the transmitted power. One approach for mitigating these losses is to switch to a continuously variable transmission. Changing to such a system greatly increases the possibility, and the need, for actively selecting the engine speed, and here a conflict emerges. A low engine speed is desired for high efficiency but a high speed is required for high power.

Heavy wheel loaders often operate according to a common repeating pattern known as the short loading cycle. This cycle is extremely transient, which makes the choice of engine operating point both important and difficult. At the same time, the repeating pattern in the operation enables a rough prediction of the future operation. One way to use the uncertain prediction is to use optimization techniques for selecting the best control actions. This requires a method for detecting the operational pattern and producing a prediction from this, to formulate a manageable optimization problem, and for solving this, and finally to actually control the machine according to the optimization results. This problem is treated in the four papers that are included in this dissertation.

The first paper describes a method for automatically detecting when the machine is operating according to any of several predefined patterns. The detector uses events and automata descriptions of the cycles, which makes the method simple yet powerful. In the evaluations over 90% of the actual cycles are detected and correctly identified. The detector also enables a quick analysis of large datasets. In several of the following papers this is used to condense measured data sequences into statistical cycles for the control optimization.

In the second paper dynamic programming and Pontryagin's maximum principle is applied to a simplified system consisting of a diesel engine and a generator. Methods are developed based on the maximum principle analysis, for finding the fuel optimal trajectories at output power steps, and the simplicity of the system enables a deeper analysis of these solutions. The methods are used to examine and visualize the mechanisms behind the solutions at power transients, and the models form the basis for the models in the following papers.

The third paper describes two different concepts for implementing dynamic programming based optimal control of a hydrostatic transmission. In this system one load component forms a stochastic state constraint, and the concepts present two different strategies for handling this constraint. The controller concepts are evaluated through simulations, in terms of implementability, robustness against uncertainties in the prediction and fuel savings.

The fourth paper describes the implementation and testing of a predictive controller, based on stochastic dynamic programming, for the engine and generator in a diesel electric powertrain. The controller is evaluated through both simulations and field tests, with several drivers, at a realistic work site, thus including all relevant disturbances and uncertainties. The evaluations indicate a $\sim 5\%$ fuel benefit of utilizing a cycle prediction in the controller.

Populärvetenskaplig sammanfattning

Precis som för andra fordon finns det en vilja att sänka bränsleförbrukningen för hjullastare genom att hitta och minska de energiförluster som finns i systemet. Drivlinan i dagens tyngre maskiner är uppbyggd kring en momentomvandlare. Denna komponent gör att momenten i drivlinan anpassas till fordonets och motorns hastigheter, så att om maskinen t.ex. kör fast så ökar det drivande momentet på hjulen samtidigt som lasten på motorn inte ökar, helt utan någon aktiv styrning. Detta gör drivlinan självreglerande och mekaniskt robust, och gör även att maskinen kan ha hög drivande kraft från stillastående. Tyvärr ger momentomvandlaren även förluster, som i vissa lägen kan vara mycket stora i förhållande till den effekt som används för att förflytta maskinen. Andra möjliga lösningar för drivlinan i dessa maskiner är att använda el- eller hydraulmotorer för framdrivningen, och att använda en dieselmotor kopplad till en generator eller hydraulpump för att driva dessa. Transmissionen i en sådan diesel-elektrisk eller diesel-hydraulisk drivlina fungerar då som en steglös växellåda. Med en sådan uppstår möjligheten, och kravet på, att aktivt välja motorvarvtal, och här uppstår en konflikt. Lågt varvtal ger låg bränsleförbrukning och högt varvtal ger bra respons, men det tar tid att byta varvtal. En genomtänkt strategi behövs därför för att välja varvtal, speciellt vid snabba effektförändringar.

Tyngre hjullastare arbetar ofta enligt ett vanligt mönster, där den korta lastarcykeln upprepas gång på gång. I denna cykel kör maskinen fram och fyller sin skopa från en källa, backar, kör fram och tömmer skopan i en mottagare, backar tillbaka och sedan upprepas cykeln. Detta mönster är väldigt transient; hastigheten och effektbehovet är sällan konstant, vilket gör valet av motorarbetspunkt extra svårt. Samtidigt innebär regelbundenheten att det finns en generell kunskap om hur maskinen kommer att användas i den nära framtiden, kunskap som bör användas i motorstyrningen. Ett sätt att använda den osäkra förutsägelsen är att använda optimeringsteknik för att i varje tidpunkt hitta den bästa styrsignalen. För att göra detta krävs en metod för att hitta mönstren i hur maskinen används och skapa en prediktion från detta, att formulera optimeringsproblemet så att detta blir hanterbart, att lösa detsamma, och slutligen att faktiskt styra maskinen enligt resultaten från optimeringen. Detta problem behandlas i denna avhandling, som i huvudsak är uppbyggd kring fyra artiklar.

Den första artikeln beskriver en detektor som automatiskt registrerar om maskinen arbetar enligt något av flera fördefinierade mönster. Denna gör det bland annat möjligt att snabbt analysera stora datamängder och kondensera dessa till statistiska mönster. Dessa kan, som i flera av de följande artiklarna, användas för att beräkna optimala styrlagar för varje mönster. Den andra artikeln undersöker de mekanismer som styr den bränsleoptimala lösningen för ett enklare system bestående av en dieselmotor kopplad till en generator, och visualiserar dessa. Den tredje artikeln beskriver och utvärderar två olika implementationer av optimal styrning av en hydrostatisk transmission, med avseende på implementerbarhet, robusthet mot prediktionsosäkerhet och bränslebesparing. Den fjärde artikeln beskriver optimal styrning av en diesel-elektrisk drivlina, med formuleringen av optimeringsproblemet, lösningen av detta, till implementation och tester utförda i maskin med flera förare på en realistisk arbetsplats.

Acknowledgment

Even though there is only one author of this dissertation, there are many persons that deserve acknowledgment for contributing, in many different ways, to the work presented, for the rewarding discussions which sometimes even were on the topics of this dissertation, and for the enjoyable atmosphere at the group.

I would like to start by thanking my colleagues at the Division of Vehicular Systems at Linköping University. First of all our professor and my co-supervisor Lars Nielsen for giving me the opportunity to work at his group with this rewarding project, and my first supervisor Jan Åslund for his supervision and assistance in my work on the project presented in this dissertation. I would like to acknowledge all the work my coauthors at the group; Peter Nyberg, Christofer Sundström, Erik Frisk, Mattias Krysander and Jan Åslund, have put into the papers which makes up the dissertation. Further, I would like to thank all my colleagues for the pleasant atmosphere at the group. I have had very interesting discussions on the different topics of our research and related to our undergraduate teaching, I have enjoyed the interesting fika-room discussions, and I will always carry with me the experiences from our conference trips.

I would like to thank all those at Volvo Construction Equipment who have contributed to or participated in the work presented here. First of all I would like to thank my supervisor Anders Fröberg for his invaluable support, from proposing research directions to providing the test vehicle for the final paper. I would like to thank Rickard Mäki and Jonas Larsson for their support of the projects that resulted in my papers. I would also like to thank all others who have contributed with material or their time, and especially mention my previous supervisor Gianantonio Bortolin, and Bobbie Frank and Mikael Skantz. This dissertation would not have been possible without your assistance.

Last but not least, I would like to thank my wife Malin for her endless support and for, together with my son Vilhelm, always reminding me that there is much more to life than writing a dissertation.

Contents

Introduction	1
1 Introduction	3
1.1 Background	3
1.2 Outline and Contributions	6
2 Optimal predictive control	9
2.1 Overview and definition	9
2.2 Prediction	11
2.3 Optimization	12
2.4 Optimal predictive control for wheel loaders	15
3 Wheel loader operation	17
3.1 Overview and the Short Loading Cycle	17
3.2 Repetition detection	19
References	23
Papers	29
1 Robust driving pattern detection and identification with a wheel loader application	31
1 Introduction	32
2 Problem formulation and challenges	33
3 Modeling	37
4 Method	40
5 Evaluation	44

6	Conclusions	48
	References	49
A	Nomenclature	51
2	Minimizing Fuel Use During Power Transients for Naturally Aspi- rated and Turbo Charged Diesel Engines	53
1	Introduction	54
2	System setup	55
3	Problem statement	58
4	Optimization Methods	59
5	Engine map and static optimal solution	62
6	DP derived optimal trajectories	63
7	PMP trajectory derivation	66
8	Suboptimal method development	74
9	Discussions and comments	76
	References	78
3	Development of look-ahead controller concepts for a wheel loader application	81
1	Introduction	82
2	Models	83
3	Methods	90
4	Evaluation	100
5	Summary and Conclusions	112
	References	114
4	Predictive control of a diesel electric wheel loader powertrain	117
1	Introduction	118
2	System and concept overview	119
3	Controller descriptions	126
4	Measurements	130
5	Results	132
6	Discussion and Conclusions	139
	References	140

Introduction

Introduction

1.1 Background

The wheel loader is a type of engineering vehicle mainly designed for short distance handling of bulk material, but also used in a wide range of other applications. An example of the vehicle type is shown in Figure 1.1. As an off-road engineering vehicle it is seldom used for longer transports, but more often in high force and/or high precision operations at relatively low vehicle speeds. For this reason, the transmission of the vehicle is designed for being mechanically robust and to have smooth and predictable low speed/high torque characteristics. In particular, the transmission has to be able to provide high wheel torque from standstill up to some speed without gear change jerks, and also have characteristics that prevents the engine from stalling if the driving resistance increases suddenly, as if the bucket hits an obstacle during a bucket filling. The most common solution for heavy loaders is to base the transmission on a torque converter [51, 52, 53, 54, 55, 56]. Basic descriptions and models of torque converters can be found in [27] and in [25], and in papers where the devices are used, such as in [37], [62] and [69]. The characteristics of a torque converter generally follows, and is usually presented as in, the example in Figure 1.2. In a hydrodynamic torque converter the propelling connection drives a flow of oil, and the flowing oil drives the propelled connection. The propelling connection is therefore denoted the pump side and the propelled connection is denoted the turbine side. Figure 1.2 shows the pump torque T_P , measured at a specific pump speed which in this case is $1000rpm$, and normalized with the torque at $\omega_T = 0$. It also shows the relation between pump torque and turbine torque T_T and the transmission efficiency η , which is the speed relation multiplied with the torque relation. All of these are drawn as functions of the output to input speed ratio

$\phi = \omega_T/\omega_P$ or the slip $s = 1 - \phi$. The pump torque is, for a fixed pump speed, relatively constant up to some speed ratio where it starts to drop and is close to zero at $\phi = 1$. This means that if the vehicle drives at high slip, a disturbance in vehicle speed or wheel torque will not increase the load on the engine much. Thus the engine is prevented from stalling if the vehicle experiences an increase in driving resistance, and at the same time such a disturbance will cause the propelling torque to increase according to the T_T/T_P relation. It is important to note that the torque converter also allows the vehicle speed to go to zero without any active action, such as the releasing of clutches. For wheel loaders these properties are particularly important, especially during the bucket filling where the vehicle drives at a low speed and experiences high and unpredictable forces from the bucket to ground contact, and this is one reason torque converters are so common in heavy wheel loaders. The drawback is that the transmission of torque requires some slip, and there is considerable power lost due to this slip. In many other applications the high losses can be mitigated by having a lockup function in the torque converter, as indicated e.g. in the models in [25], in which the input and output shafts are locked together by a clutch thus negating the slip and the related losses. This can however only be done if relatively constant vehicle speed is expected and the available gears places the engine speed in its allowed speed envelope. For wheel loaders this would only be possible in more static operations at higher speeds such as in transports, and not in the more common highly transient and low speed loading operations.



Figure 1.1: An example of a wheel loader, from [18].

Electric and hydraulic motors have very different characteristics as compared to internal combustion engines, especially in that these can produce torque from standstill. The electric or hydraulic power required to run the motor can come from a generator or hydraulic pump, which could be propelled by an internal combustion engine. For such a system the electric motor and generator, or hydraulic motor and pump, becomes an infinitely variable transmission (IVT), corresponding to a gearbox which has not only infinitely many gears, but also infinite gear ratios [48]. Transmissions of this type has for a long time been

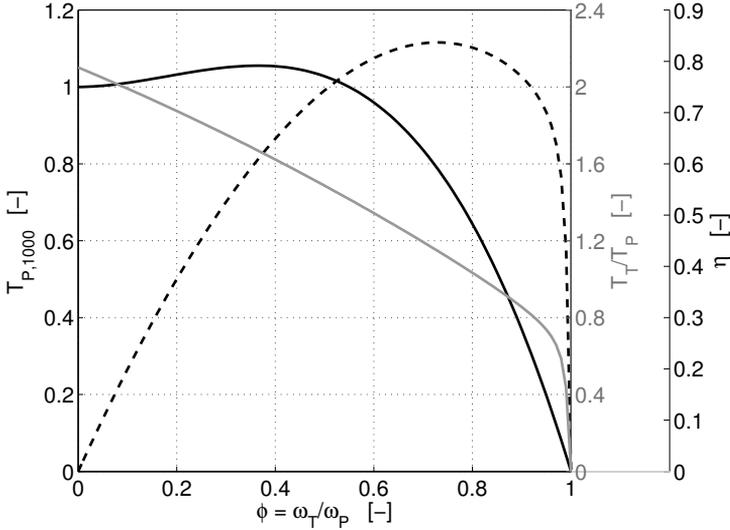


Figure 1.2: An illustration of torque converter characteristics. The black line shows the pump, or input, torque $T_{P,1000}$, the gray line shows the output to input torque ratio T_T/T_P and the dashed line shows the efficiency $\eta = \omega_T T_T / \omega_P T_P$.

used in some special applications, such as diesel-electric locomotives [28, 31] and hydraulic drives in excavators [97], and have more recently attracted some attention as transmissions in regular vehicles [13, 45, 46, 91]. A related transmission type is the continuously variable transmission (CVT), which corresponds to a gearbox with an infinite number of gears but with a limited span of gear ratios. These drives are more commonly mechanical solutions, such as the belt type CVTs in [90]. These types of transmissions can also be expanded with a significant energy storage, such as a battery [16], supercapacitor [17], pressure vessel [30] or flywheel [85], to form a series hybrid powertrain. Regardless of the type of transmission, the infinite number of gear ratios greatly increases the freedom in actively selecting the engine speed. In transients, when there is a non-zero engine speed derivative, the engine speed controller also has to take into account the power flow to or from the engine inertia. The problem of selecting the engine speed for this type of system in transient conditions has been studied e.g. in [44] and [76], and more recently in [67] and [86]. Since the engine speed is free, the main characteristic of the load is the power required in the electric/hydraulic/other connection. For static, or quasi static, load power the engine speed can be placed at the corresponding static fuel optimal operating point. These operating points, as a function of output power, are in general located near the maximum torque line of the engine, which means that there is little margin for handling sudden increases in the demanded output power. If the engine speed is low and there is a sudden increase in the desired output power, the engine will have to speed up before supplying this power, which causes a delay. In case there is a turbo this device will also have to speed

up, a process that can have a time constant of several tenths of a second [87]. Common delays of this magnitude would not be acceptable in such transient operation as that of wheel loaders. Another strategy that would not cause such delays could be to always maintain high engine and turbo speeds, though this would reduce the average efficiency of the engine. So, the engine speed should be low when the requested power is low and high when the power is high, but due to the time constants in the system this would require a load prediction.

In the most common operation of wheel loaders, which is short distance loading, the vehicle operation consists of repeating a distinct cycle known as the short loading cycle, as described in detail in [19]. The distinct and repetitive operation according to this cycle means that at the beginning of a cycle, the general appearance of the rest of the cycle can be predicted. In case the torque converter has been replaced with a CVT type transmission, the engine speed is actively controlled, and in this controller the information obtained through the cycle prediction should be utilized. There has been a substantial amount of work done on optimal control against predicted future operating conditions, in many fields both theoretical and applied. In most of the work with vehicle applications the prediction has however been deterministic. Some, like [76], of these has focused on pure dynamic optimization, without considering online implementations. Others, such as [29] and [36], have considered, or even implemented, online controllers but still assumed a deterministic prediction in their optimization. For the wheel loader however, even though the general appearance of the cycle can be predicted, the details of each individual cycle cannot, and a predictive controller would have to rely on a partly stochastic prediction. There are some vehicle related papers in which stochastic predictions are used; [35], [26] and [39] are just some. The controllers presented are in general focused on systems with dynamics that are considerably slower than the engine speed dynamics though, such as the battery state of charge in hybrid electric vehicles. The papers included in this dissertation describe work related to the development and implementation of a predictive controller for the engine dynamics in a CVT (or IVT) based wheel loader powertrain.

1.2 Outline and Contributions

This dissertation describes the development, implementation and testing of different algorithms necessary for utilizing the repetitiveness in the operation for optimal predictive control of a continuously variable wheel loader transmission. The dissertation starts with an introduction in three chapters, where the present chapter, Chapter 1, gives a background to the problem addressed and a summary of the papers included. Chapter 2 describes and defines the concept of optimal predictive control and its components, and Chapter 3 describes the particular properties of wheel loader operation that is an enabler for the prediction. This introduction is followed by the main part of the dissertation, which consists of the four papers [73], [70], [69] and [72]. In the rest of this section these papers are summarized, and the addressed problems are motivated in the context of the theme of this dissertation.

Paper 1: Robust driving pattern detection and identification with a wheel loader application, [73].

The first problem was to find a method for obtaining a prediction of future operation. Two general approaches were identified; to base the prediction on repetition detection or on pattern detection. While both of the approaches were promising from a prediction point of view, the second was considered more practical for optimization purposes. The repetition detection is described further in Section 3.2. More work was put into the pattern detection track, and the result is presented in the appended paper.

Paper 1 presents a framework for off- and online automatic detection and identification of operation according to a set of predefined wheel loader operational cycles. In this detector the cycles are constructed from a small set of discrete events, such as a change of driving direction or a bucket emptying, forming a unique sequence for each type of cycle. The paper is focused on the short loading cycle, and the detector is evaluated offline and online in handling of different materials by different drivers. The evaluation shows positive results of this simple but capable detector, with over 90% of all cycles detected, even in difficult operation such as handling of shot rock.

The detector described is highly useful for quickly analyzing and condensing large amounts of recorded data, by extracting the cycles and relevant information for these, so that this information can be summarized into a manageable cycle description suited for the optimization. In this role the cycle detector has been used in all the papers [65], [66], [69], [68], [71] and [72]. The substantial benefit of having the cycle detector when analyzing large sets of data is the main contribution of this paper.

Paper 2: Minimizing Fuel Use During Power Transients for Naturally Aspirated and Turbo Charged Diesel Engines, [70].

The next problem was to construct system models and selecting an optimization method suited to the problem at hand. The initial study focused on using simple models and trying to find underlying mechanisms in the optimal trajectories that could be utilized for simplifying the larger optimization problem.

Paper 2 is a technical report based on a section of the licentiate thesis [63], which in turn is an extension of the papers [64] and [67]. The paper analyzes the problem of finding the optimal operating point trajectories for a naturally aspirated or turbo charged engine connected to a generator, during output power transients. The problem is a simplified version of optimal control of the engine and generator in a diesel electric transmission, especially in that the power trajectory is deterministic. Instead of focusing on application of the solutions, the optimal engine operating point trajectories are studied in depth using both dynamic programming and Pontryagin's maximum principle.

The contribution of this paper is a deeper understanding of the mechanisms behind the fuel optimal solutions for power transients, and a pedagogical example of how Pontryagin's maximum principle can be used for finding the optimal trajectories for simple problems. The models introduced in this paper are used in the papers [65], [66], [69], [68], [71] and [72].

Paper 3: Development of look-ahead controller concepts for a wheel loader application, [69].

With the general prediction and optimization methods or concepts selected, the final problem was to implement and adapt these for use as an online controller in a wheel loader. In this implementation study a specific transmission was selected, a device presented in the patent [50]. The main problem in the implementation was how to handle the uncertainties in the prediction.

Paper 3 describes two dynamic programming based controller concepts for a wheel loader with a three mode hydrostatic continuously variable transmission. The two concepts present very different strategies for handling prediction uncertainties, especially in the requested hydraulic flow, since this flow introduces a stochastic state constraint. The first controller includes uncertainties in the load description, and is based on the papers [65] and [66]. The second concept uses the possibility of reducing the vehicle speed, and is based on the paper [68], supplemented with a small time delay in the hydraulics. The paper [69] focuses on the reformulations required for online implementation of the concepts, followed by an evaluation of these implementations.

The main contribution is the study into how the stochastic state constraint introduced by the hydraulic flow can be handled in the optimization. The paper also forms a feasibility study for predictive controllers for the transmission concept described, and in case a similarly equipped machine becomes available, the proposed strategies could form the basis of a predictive engine speed controller.

Paper 4: Predictive control of a diesel electric wheel loader powertrain, [72].

The final problem of implementing the selected prediction and optimization methods was approached again, but with a different transmission layout, based on the machine presented in [92]. The layout enabled a simpler load description, and since a machine was available for field tests, the controller concept could be implemented and tested.

Paper 4 treats the implementation and testing of a controller for the engine operation in a diesel electric wheel loader transmission. The main controller is based on stochastic dynamic programming, with a load probability distribution that depends on the distance driven. This controller is evaluated in both simulations and field tests against two simpler reference controllers. The optimized control schemes were first evaluated in [71], but only through simulations that did not include the influence of model errors, an uncertain environment and unpredictable driver feedback. In the included paper, the controllers are also field tested in a realistic operation scenario with three different drivers, thus including all relevant disturbances and uncertainties.

The main contribution of the paper [72] is an evaluation, which includes field tests, of the possible benefit of utilizing a cycle prediction in a continuously variable wheel loader transmission controller. Both simulations and field tests indicate a 5% fuel benefit of utilizing the cycle prediction in the engine controller. The paper is also a logical closing of the dissertation in that all the main parts developed are collected and tested together in a realistic implementation.

Optimal predictive control

2.1 Overview and definition

This dissertation treats optimal predictive control (OPC) of a continuously variable transmission in a wheel loader. Section 2.1 gives a broad definition of the concept of OPC and relates this to similar concepts such as model predictive control. Sections 2.2 and 2.3 describes the two components 'prediction' and 'optimization' required for this type of control, with brief introductions to the concepts, and some background and examples. Section 2.4 relates the described prediction and optimization methods to wheel loader operation, and motivates the choices that are made in this dissertation.

Optimal predictive control is here defined as a control method in which the control action at each moment is decided through optimization with respect to a prediction of future disturbance signals. Denoting the states of the system x , the control signals u and the disturbance signals w , the problem can be summarized according to Equation (2.1). It is assumed that there exists a prediction of the signals in w , although the prediction might include uncertainties, might have a limited time horizon t_θ and might not become available until the moment t_k when the control action $u(t_k)$ should be decided upon.

$$\lim_{T \rightarrow \infty} \min_{u \in \mathcal{U}} \frac{1}{T} \int_{t_k}^T G(x(t), u(t), w(t, x, u)) dt \quad (2.1a)$$

$$\frac{dx}{dt} = F(x(t), u(t), w(t, x, u)) \quad (2.1b)$$

$$0 = C(x(t), u(t), w(t, x, u)) \quad (2.1c)$$

$$x_0 = x(0) \quad (2.1d)$$

For practical reasons, the cost function (2.1a) is in general replaced with the truncated cost function (2.2), and the truncation is in general made inside the prediction horizon, so that $N \leq \vartheta$.

$$\min_{u \in U} \left\{ J_N(x(t_N)) + \int_{t_k}^{t_N} G(x(t), u(t), w(t, x, u)) dt \right\} \quad (2.2)$$

In the general control process, at each moment t_k , a prediction of $w(t_k, \dots, t_\vartheta)$ is obtained and an optimization is performed based on this prediction to find the minimizing $u(t_k, \dots, t_N)$. The first control action $u(t_k)$ is applied and the process is repeated at the next time step. This description is very similar to model predictive control (MPC), as described in detail in [14] and [22], in that the MPC is also based on a system model, which is used for making a prediction of the system behavior and finding the best control action. In its most classical form the MPC problem is formulated according to equation (2.3). In this formulation y are measurable outputs, r are references or targets for the outputs, u are control signals, λ are weighting factors, x are states, A , B , C and D are matrices and Z_u and Z_x are control and state constraints.

$$\min_u = \sum_{j=1}^N \{ (y_j - r_j)^2 + \lambda_j u_{j-1}^2 \} \quad (2.3a)$$

$$x_{j+1} = A_j x_j + B_j u_j \quad (2.3b)$$

$$y_j = C_j x_j + D_j u_j \quad (2.3c)$$

$$Z_{u,j,l} \leq u_j \leq Z_{u,j,h}, \quad Z_{x,j,l} \leq x_j \leq Z_{x,j,h} \quad (2.3d)$$

This control approach is an alternative to classical PID-type controllers, which is useful if the controlled system has many inputs, outputs or states, or if there are significant constraints. Further, the method can also utilize a prediction of reference state changes. There are also several mature software packages for MPC, such as [7]. Still, the classical formulation in Equation (2.3) is highly restrictive in that it requires a linear model, a cost that is a quadratic function of control action and deviation from known references, and is unable to utilize a prediction of future disturbances. There are however several extensions and modifications of this classical MPC formulation intended for removing these restrictions. Some of the more known extensions are generalized predictive control (GPC), robust MPC (RMPC) and nonlinear MPC (NMPC), each of which mainly address one specific deficiency in the classical MPC formulation. For these specific examples, GPC introduces a prediction of measurable disturbances, RMPC considers uncertainties in the models used and NMPC introduces nonlinearities in the models [14]. There are many more advanced formulations proposed or used, see for example [79], although there is less consistency in the notation for these formulations. In this work it was decided not to use MPC related notations even though the OPC problem could be formulated in the MPC framework. This was partly because there is no well established notation within the field of MPC for the type of problem studied here, but mostly because the MPC notation is associated with the classical formulation and control towards known, and often constant, reference values.

In the problem studied here, it is assumed that there exists a prediction of future operating conditions, that this prediction contains significant uncertainties, and that the prediction might change over time. It is also assumed that there exists a model of the controlled system. At each instant the available prediction and the system model should be used for solving the, possibly truncated, problem (2.1), so that the optimal control action can be applied. The following two sections describe some different methods for obtaining a prediction and for calculating the optimal control action.

2.2 Prediction

It is assumed that there are some external signals $w(t)$, here denoted disturbance signals, affecting the system and that there exist a prediction of these disturbance signals. The prediction is an enabler for optimal predictive control, and its properties are crucial for how the controller could be implemented. The main properties of the prediction are the prediction horizon t_θ and the level of uncertainty in the prediction, although there are other important properties such as the number of components in w . The prediction horizon affects the choice of optimization horizon t_N and the required complexity of the residual cost function $J_N(t_N)$, which is the cost associated to each state at the end of the optimization horizon. The uncertainty in the prediction affects which optimization methods that are suited to the problem. There are several papers in the field of vehicular control suggesting a wide range of different methods for obtaining a disturbance prediction, with specified or unspecified purposes. In the following a few different types of predictions, their properties and some examples, are listed. This list does by no means purport to be complete, but is only a selection of some interesting and illustrative examples.

A first category of predictions are the deterministic route based predictions, in which the external disturbances are known functions of some of the states of the system. One example of this type is the topography information from road maps used in the optimal predictive controller described in [29]. For this type of prediction, since the whole $w(x)$ map is known beforehand, there is in practice no prediction horizon that limits the optimization horizon and the optimal control actions as functions of the state could in theory be precalculated. In practice, precalculation of the optimal control actions as a function of the states is in general not possible though, primarily due to the size of the maps in the route database.

Another type of predictions are the observations based predictions, in which there are some type of sensors which are able to 'scan ahead' of the controlled system. Data, such as topography or the traffic situation, can be communicated from other vehicles; this is a novel field of research known as cooperative driving, where [40], [77] and [5] are just a few examples. Using forward looking radars for adaptive cruise control and collision avoidance or mitigation is becoming common in new cars, and has been thoroughly investigated [11, 98], and in later years there has been research and implementations where the radars are supplemented with, or replaced by, camera based systems [4, 8, 21, 89]. Common for this type of predictions are that the whole disturbance function is not known

beforehand, but there is a prediction horizon that gives an upper limit for the optimization horizon and the control actions can therefore not be optimized beforehand. These predictions in many cases contain some uncertainties that usually grow larger the further away the observations are, making the prediction horizon a somewhat murky concept.

A third type are predictions based on pattern recognition, in which the patterns can be predefined or evolving. In the predefined case there is a set of patterns and the predictor uses sensor data to select that which reflects the current operation and base the prediction on that pattern. This type includes identification or classification of driving type [34, 41] or situation [58, 74]. In the evolving pattern case the predictor tries to find recurring structures in the data, and base the prediction on these structures. This type is less common since the detector could produce unexpected results and because just loose enough assumptions has to be made of what form of recurring structures the detector should look for. In both of these types the predictions can be expected to contain relatively large uncertainties, which has to be considered in the optimization. Just as with the upstream observations, it can be expected that the uncertainties grow with the prediction horizon, especially in the evolving pattern prediction.

A fourth type of predictions are the stochastic Markov process disturbance models [24, 80], in which the disturbance signals are described by probability distributions, where the probability p of a specific disturbance at the instant t_k depends only on the state at the same instant, as described by $p(w_k|x(t_k))$. The disturbance at the previous instant w_{k-1} is however often included as a state. Since the process is memoryless, and the probability distributions are time invariant, a prediction of more than one step into the future becomes meaningless. Since the load descriptions are time invariant, the type should be considered semi-predictive; the models can describe and predict the evolution of a transient episode, but not the occurrence of a transient after stationary operation. Despite the simplicity of the models, from a prediction point of view, these have found use in several applications such as power split control in hybrid electric vehicles [42, 59, 81], route prediction [38, 83] and risk assessment [32, 82, 95]. Just as in the case of map- or database based predictions, since the whole $w(x)$ map is known beforehand, the optimal control actions as functions of the state can be precalculated. This approach is used in most implementations of stochastic dynamic programming, such as in [35], [57] and [39].

2.3 Optimization

Assuming that a prediction of some sort is available, the optimal predictive controller would at each stage select and apply the optimal control action. This implies that an optimization has been performed, either beforehand for all relevant state and load combinations, or as part of the selection of a control action at each stage. Both of these strategies have benefits and challenges. Performing the optimization beforehand requires first of all that the complete disturbance signal prediction is available before the process is started. This can be the case if the disturbances are either a limited number of deterministically known trajectories or can be formulated as uncontrollable states with stochastic tran-

sitions, such as Markov chains. Still, even if this holds, the number of state and load combinations that would require a precalculated control action might be unmanageable. Performing the optimization online on the other hand requires that the optimization is quick compared to the time constants of the controlled system, or the optimal control action will no longer be optimal when it has been calculated. Just as in the offline optimization, the load description highly affects the calculatory effort. A deterministic prediction would only require a single trajectory to be optimized, while a stochastic prediction would require a more complex load alternative handling or load description for managing the branching tree of possible load trajectories. Both the off- and online strategies have been implemented in different forms, with different load descriptions and optimizers. The rest of this section gives brief descriptions of the three main classes of optimization algorithms, and relates these to the deterministic or stochastic, offline or online, optimization problem in optimal predictive control. The three classes are the dynamic programming or Hamilton-Jacobi-Bellman approach, the indirect methods, and the direct methods.

Dynamic Programming

Dynamic programming (DP) is a structured method for making a complete search of the entire discretized state and control space. The recursion, as described in [6] and [9], used can be formulated according to Equation (2.4).

$$J_k(x_k) = \min_{u \in U} \{g(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k))\} \quad (2.4)$$

Since a complete search is made, this method guarantees global optimality. Furthermore, since the method is based on testing and not an iterative trajectory improvement, no derivatives are required. These properties together means that the method is not sensitive to non-convex or 'ugly' models containing switches, discontinuities, stochastic components, mixed discrete and continuous states and controls etc., in the same way as the indirect and direct methods are. The complete search however means that simulations has to be performed for all allowed controls from all allowed states for all possible disturbance signals, at all time steps, along with interpolations or other means required in the connecting of all the sub-trajectories. This ensures a high computational load, even for well posed problems, that grows exponentially with the number of states and controls. The method cannot manage problems with a combined number of more than about ten states plus control signals, and often problems with a far lower dimension become unmanageable. The method therefore in general suits problems where the calculation time is not critical, such as if the optimization can be done beforehand, with a low dimension and where there exist stochastic, nonconvex or nondifferentiable components. In [29] DP is used for control of vehicle speed and gear selection in a long haulage truck. The optimization is performed online with GPS data as input, and the method might be motivated by the discrete gears. There is a range of papers, such as [35], [57] and [39] that uses stochastic DP in vehicular controllers. In these cases the load is described by Markov models, and these stochastic predictions motivate the use of DP.

Indirect Methods

The indirect methods are based on Pontryagin's minimum principle (PMP) [78]. The optimality conditions are formulated using the Hamiltonian (2.5).

$$H = G(x(t), u(t), w(t)) + \lambda(t)F(x(t), u(t), w(t)) \quad (2.5)$$

PMP then states that the minimizing state and control trajectories also minimize this Hamiltonian. By differentiating the Hamiltonian with respect to the states and controls, a set of conditions necessary for optimality is obtained. The main conditions are (2.6), and the minimization problem is thus rephrased as a two point boundary value problem. The indirect methods are methods for solving this boundary value problem [10, 94].

$$\frac{\partial H}{\partial x} = -\frac{d\lambda}{dt} = -\dot{\lambda} \quad (2.6a)$$

$$\frac{\partial H}{\partial u} = 0 \quad (2.6b)$$

$$\frac{\partial J_N}{\partial x} = \lambda(t_N) \quad (2.6c)$$

The set of optimality conditions can be regarded as a generalization of the zero derivative condition in static one dimensional minimization. Just as in that simple case however, the conditions are only necessary but not sufficient conditions for optimality, unless the problem is convex, and if there are equality or inequality constraints the basic derivative conditions must be expanded with other terms. These other terms lead to state dependent switches and the system must therefore be continuously checked during the simulations to find any possible active constraint. Further, the extended system $\tilde{f} = [\dot{x}, \dot{\lambda}]$ forms a Hamiltonian system which means that, according to Liouville's theorem [43], the volume in the phase space of $[x, \lambda]$ is preserved [15, 61]. This means that unless all states are close to the instability border, the problem will become unstable, which complicates the solving of the two point boundary value problem. Last but not least, if there are stochastic components in the system and the cost in the objective function is written as an expected value, the optimality conditions might not be well defined. The primary benefit of the indirect approach is that the formulation of the optimality conditions might reveal underlying structures in the solution to the optimization problem. In some very special problems the conditions might even lead to an analytical solution. One of the most important examples in the vehicular control area is probably the ECMS approach for controlling the power split in hybrid electric vehicles, as described in [75] and [84]. Other problems suited for indirect methods are marginally stable systems with few state constraints, such as space flight trajectory planning [49]. The indirect methods are otherwise in general suited for the same types of problems as the direct methods, and in later years the direct methods has taken over as being the most popular.

Direct Methods

In the direct methods the original problem is discretized and the finite discrete problem is directly solved as a nonlinear optimization problem. In contrast to the indirect methods, there is no formulation of optimality conditions but an iterative search, using cost related derivatives, is performed for finding trajectories with increasingly low value of the cost function (2.2) [10, 94]. One drawback compared to the indirect methods is that the direct cannot give an explicit solution, regardless of the problem. The direct methods also require that the cost function, the state dynamics equations and the constraints can be differentiated, at least once, at least with respect to the controls, which means that the methods are not suited for 'ugly' models, with switches, discrete states, etc. Nor can the methods, just as with the indirect methods, guarantee global optimality or easily handle stochastic models. There are however several benefits with using direct methods, as compared to the indirect methods. The simulation of the system is simpler and more stable since the constraints are not formulated as switching conditions and a stable system is not made unstable by co-state dynamics. Since there is no need for formulating co-state dynamics or similar, setting up the problem is simpler than in the indirect methods. The effective solving of problems with these methods instead relies heavily on the effective calculation and handling of derivatives [93], for which there exist many mature software packages such as [3]. The increased simplicity, in general along with faster and more stable convergence, are reasons why the direct methods have become more popular than the indirect lately.

2.4 Optimal predictive control for wheel loaders

The characteristics of wheel loader operation opens up some different possibilities for prediction, related to the prediction types mentioned in Section 2.2. The nature of the operation, being off-road in a complex environment and with a human driver, does not provide for using route maps or similar for obtaining a prediction suited for optimization. Sensors such as cameras, radars or lidars can be added to the machine, as has been done as part of work focused on autonomous wheel loaders, e.g. in [2], [12], [23] and [47], to provide a close range awareness and prediction. The information obtained from such sensors would be of high reliability, but with a highly limited prediction horizon and only including information about the environment and not the intentions of the driver. The residual cost function $J_N(x)$ would be important, due to the relatively short prediction horizon as compared to the time constants of the turbo charged engine, but difficult to construct. Although this sensor based prediction possibility could be beneficial for component control, the available prediction horizon does not suit the intended application, while the uncertainties in the necessary time frame complicates the implementation of an optimizer.

As mentioned in Section 1.1, and described in greater detail in Section 3.1, the most common wheel loader operation consists of repeating a distinct cycle. This repetitive and pattern bound operation provides for some other means for obtaining a prediction. Two main approaches can be identified, where the

choice of approach has a major impact on the requirements on the optimization method used, and so that this largely decides which approach that can be used. The first approach is to utilize the repetitiveness of the operation, and in the predictor try to detect if there is a repeating pattern in the operation, and build the prediction on an assumption of the detected pattern continuing into the future. Since such a detector would try to find repetition, it could be able to make predictions in a wide range of operations, with relatively little data predefined, making the detector highly versatile. While the detector would often be able to provide a prediction, this prediction would only become available to the optimizer when repetition is detected, and it would contain significant uncertainties. The optimizer would therefore need to run online and be able to handle stochastic predictions. Despite the versatility of this prediction type, and because of the difficulty of setting up and solving the resulting optimization problem, it was decided that this approach would not be the focus of the work presented in this dissertation. The implementation of a simple repetition based predictor for wheel loader operation is however described in Section 3.2. The second approach is to utilize the pattern bound operation, and in the predictor try to detect if the wheel loader is operating according to any of a set of predefined patterns, and build the prediction on an assumption of the detected pattern continuing into the future. This type of detection and prediction relies on the assumption that there is a limited number of patterns, or cycles, which the machine usually operate according to, and that these are known beforehand. If this assumption holds true, and the patterns are specific enough for an optimization, it is not necessary to perform the optimization online but it can be performed beforehand. For wheel loaders there exist a few common operation types, such as the short loading cycle described in Section 3.1. While this very broad operation description is common, it is in itself not detailed enough for optimization. On the other hand, it is common for this type of machine to operate at the same site, in the same environment, with the same tasks over longer times. In such a situation, the prediction and corresponding optimized controller can be based on the details of the operation performed over the previous hours or days, and the online controller can use a pattern detector for selecting the proper optimized control strategy. The predicted operational patterns would contain significant uncertainties but these would also be available to the optimizer early, and the time available for controller optimization would be in the order of hours. Because of the significant uncertainties in this type of prediction and the availability of time for calculation, it was decided that the dynamic programming methods were the most appropriate tools for the controller optimization. This last concept of prediction according to predefined uncertain cycles and optimization using dynamic programming is the main approach studied in this dissertation, for utilizing the specifics of wheel loader operation for optimal predictive control.

Wheel loader operation

3.1 Overview and the Short Loading Cycle

The wheel loader is a versatile machine that comes in a wide variety of sizes, ranging from compact loaders at around 5 tonnes up to the heaviest machines at several hundred tonnes. The smaller machines are often a base tool used for a wide variety of tasks, such as handling of pallets, hay bales or other unit material, use as a snow plow or blower, or a carrier of various engineering tools such as hydraulic drills or brush cutters. The machines are however designed mainly for the purpose of bucket handling of bulk material, such as snow, soil, gravel, ore, wood chips or waste. For heavier loaders in particular, this is also the by far most common type of application, usually handling soil or granular minerals. Although bucket handling can involve dozing and moving of material, longer transports are less common due to the relatively low carrying capacity of these vehicles. The most common type of operation, and in particular the most common pattern of operation, is cyclic loading from a source to a nearby receiver, which is often next to the material source. Each cycle in this type of operation is denoted a 'short loading cycle'. This common cycle type forms the framework for the prediction in the optimal predictive controller, as defined in Section 2.4, that is the theme of this dissertation.

In the short loading cycle the loader moves material from a source to a nearby receiver. Figure 3.1 presents the general pattern of the cycle. Referring to the designations used in this figure, the cycle begins with the machine at the starting position at marker 4. The machine moves forward and picks up a load at the source at position 1, and reverses back to the starting position. Next the machine moves forward and leaves the load at the receiver at position 6, and finally it reverses back to the starting position where the cycle ends.

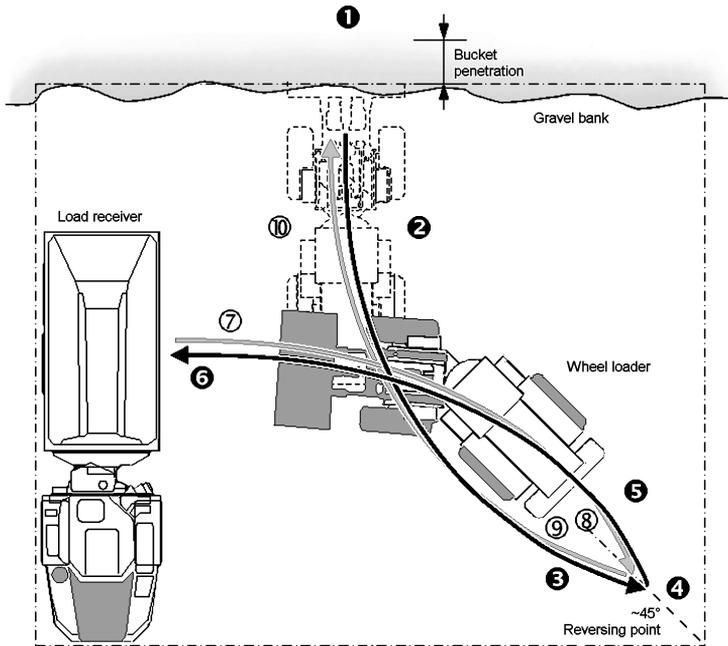


Figure 3.1: An overview of the short loading cycle, from [20].

The duration of, and distances in, the cycles vary with the type of material handled, the size of the machine and the experience of the driver, but for the loading cycle to be short, the two switches from backward to forward driving should be at roughly the same position and the machine should not travel a distance of more than about five times the circumference of the wheels in each leg of the cycle. A typical cycle for heavier loaders handling granular material can have a duration in the order of 5s for each leg and an additional 5s for filling the bucket at the source. The tool movement hydraulics, or bucket hydraulics, are separately controlled by the driver, but the pressure and required power trajectories generally follows a common pattern in the short loading cycle. During the bucket filling the bucket is usually both lifted and tilted while being pushed through the source pile. This produces high forces, both vertical and longitudinal, usually requiring very high powers both in the propulsion and the hydraulics. After the filling, the bucket is usually lifted more or less continuously during the second and third leg of the cycle, requiring significant power. The bucket emptying and the lowering of the same, during the fourth leg of the cycle, does in general not require significant power input. This cycle is repeated many times over, often with pauses or other operations between some of the cycles. The load receiver is often a dump truck or an articulated hauler, and the pauses occur when the truck has been filled and is replaced or emptied. The other operations that might occur primarily include operations for cleaning the working site from dropped material and preparing the source pile for faster

loading when a load receiver becomes available.

The intention in this work is to utilize the characteristics of this operation for optimal predictive control. This requires a method for analyzing the operation, that first of all finds the patterns in the operation, and second enables a prediction that is useful for predictive control. Two potential prediction types were described in Section 2.4; pattern detection and repetition detection. The main difference is whether the operational patterns are pre-specified or not. The pattern detection concept is investigated in the appended paper 1, [73], and the repetition detection concept is described further in Section 3.2.

3.2 Repetition detection

The intention with analyzing the operation is to find a method for obtaining a prediction for some use in some suitable optimization method. The operation is often highly repetitive, as illustrated by the example in Figure 3.2. There is therefore a real possibility of producing a prediction from such repetition, if the repetition can be detected. The simplest method for detecting repetition might be to compare a piece of the recent history of a signal to the earlier history of the same signal, e.g. by the sum of the square of the difference. In the rest of this section the approach is explained and demonstrated through two examples.

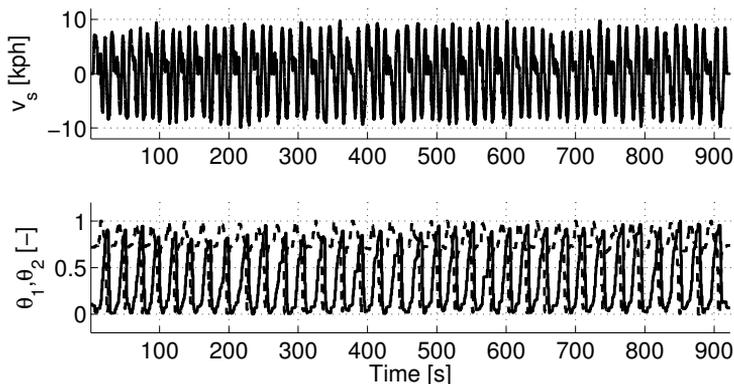


Figure 3.2: A set of data from wheel loader operation, represented by vehicle speed v_s , bucket lift angle θ_1 (solid) and tilt angle θ_2 (dashed). The set contains 34 short loading cycles and is an example of the highly repetitive operation.

Figure 3.3 shows a cutout of the bucket lift angle, θ_1 , data from Figure 3.2. In this example the process is at 590s, as marked with a vertical gray line, and a repetition detector is running. The detector picks the last fifteen seconds, as indicated by the darker line in the figure to the left, to use as a reference. This reference is then compared to the signal history. In the figure in the middle several such comparisons are made at about ten second intervals, and the best fit, which is at about $-27.5s$, is indicated in black. In the figure to the right a

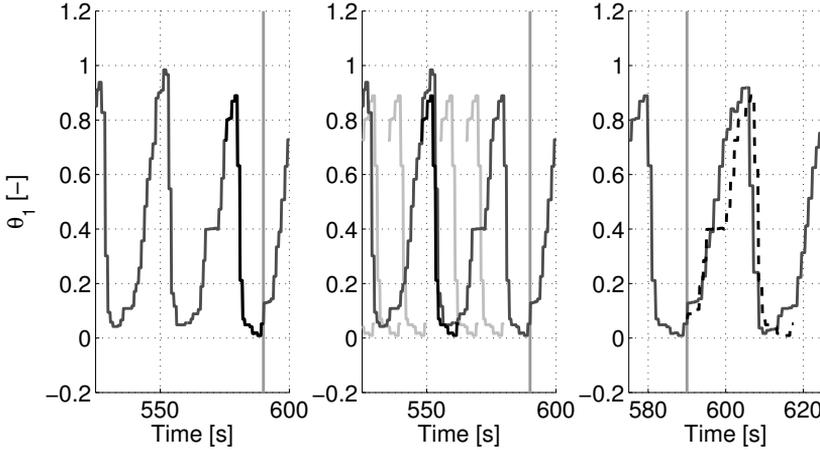


Figure 3.3: Comparison of the last 15s of the signal θ_1 to the earlier history of the signal, and a signal prediction based on the best fit, applied to data from the example in Figure 3.2.

prediction is constructed according to the dashed line, from the assumption that the operation is repeating with the detected 27.5s period, by shifting the latest signal history 27.5s into the future. Here the shifted signal is dashed and the actual signal is shown in gray. In this example the predicted signal trajectory agrees well to the actual trajectory. In the examples presented here, the fit in the comparisons are measured in the sum of the square of the difference between the two signals over the intervals, according to Equation (3.1).

$$J = \sum_{i=0}^H (y_{k-i-\tau} - y_{k-i})^2 \quad (3.1)$$

where y is the signal scanned for repetition, k is the instant of the scan, τ is the position in the signal history where the fit is evaluated, H is the size of the signal window used in the comparison and J is a measure of the fit. If J is low only for some values of τ and these τ -values are large enough, this indicates that the signal is repeating. Figure 3.4 shows the inverse of J from the data in Figure 3.2, where dark indicates a good fit, along with the output from a simple detector which marks the first peak in $1/J$ with a dashed gray line. In this example a time window of 15s is used and the data is scanned 120s back in time. The operation is highly repetitive and the time scales in the repeating cycle fits the size of the time window used in the detection, and therefore a clear, unambiguous and consistent detection of a repetition with ~ 25 s cycle time is achieved, all without any predefined information. Another example of a highly repetitive set of data with longer cycle times is presented in Figure 3.5. In the later part of this second example, some longer cycles are driven. In these cycles there is a short transportation between each bucket filling and emptying. The same detector with the same parameters is applied to this data, and the result

is presented in Figure 3.6. In this example correct and consistent detections are made in the first 1000s of the data. In this part the detector quickly adopts to changes in the period caused by switches between different cycle types, which in this case are the short loading cycle and some working site preparation through a cleaning cycle. In the later part of the data the detector is however unable to pick out a period, even though the signal is clearly repetitive, as shown in Figure 3.5. The reason for the bad result in the later part is that the signal window used in the signal used in the comparison in general contains little information, especially during the transportation phases and thus the fit will almost always be good. Since there is not a distinct fit, a clear cycle time will not be found, as shown in Figure 3.6.

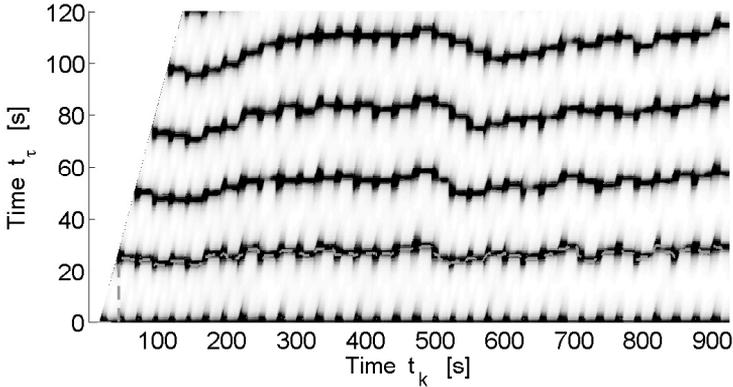


Figure 3.4: The output from a repetition detector applied to the data presented in Figure 3.2. Darker shade indicate similarity of historical data to recent data and the dashed line following the ~ 25 s dark band indicates the first repetition.

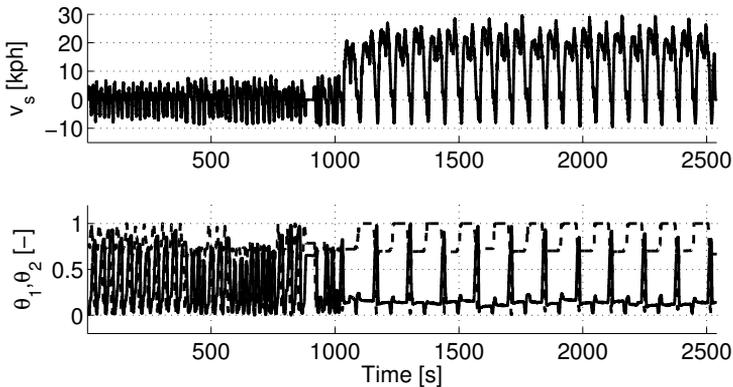


Figure 3.5: A set of data recorded in wheel loader operation. The set contains both short loading cycles, long loading cycles and some cleaning operation, and is an example of repetitive operation with longer time scales.

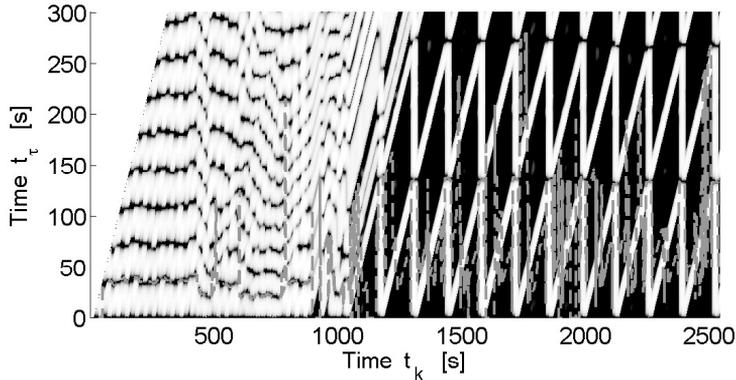


Figure 3.6: The output from a repetition detector applied to the data presented in Figure 3.5. Darker shade indicate similarity of historical data to recent data and the dashed line indicate the first repetition.

In the examples presented here a simple sum of squares fit is used, and only one signal is used in the fit calculation and repetition scan. Still, in some cases this very simple method produces a very clear and consistent detection of repetition. Figure 3.3 indicates that such a detection would also be useful for producing a prediction. The second example does however show that the simple least squares fit used here would not be sufficient for making predictions in all operating conditions. There are several more advanced methods for detecting repetition, such as those used in e.g. [1], [33], [60], [96] and [88], that could be expected to improve the detection. The positive results from the simple detector described here indicate that a practical repetition detector can be constructed. The implementation of such a detector would mean that a prediction would become available, although the accuracy of this would have to be thoroughly evaluated. The detector would be highly versatile since it could detect a wide range of driving cycles, with very little information specified beforehand. On the other hand, the resulting predictions would have uncertainties that would grow with the prediction horizon, and the prediction would only become available at the moment the detector finds a repetition. Any control optimization would therefore have to be performed online, and due to the relatively quick dynamics of the engine, especially in the engine speed, the optimization would have to be very fast. Any controller based on this type of prediction would also have to be able to handle changes in the repeating pattern. The combination of an uncertain prediction and a requirement for quick optimization pose a very difficult problem to solve, and this approach has therefore not been pursued further as part of the work presented in this dissertation. The approach followed is instead to, in advance, define one or a few operational patterns and perform control optimization against these patterns. The online controller then monitors the system for operation according to the predefined patterns, and when a detection is made, the controller can apply the corresponding control strategy. A framework for a cycle detector is presented in the appended paper 1, [73].

References

The “(Cited on p. 7,8,12)” of the reference list indicates the pages within the introductory chapters of this compilation thesis, where the reference is used.

- [1] M. Allmen and C.R. Dyer. Cyclic motion detection using spatiotemporal surfaces and curves. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, volume 1, pages 365–370. IEEE, 1990. (Cited on p. 22)
- [2] B.J. Alshaer, T.T. Darabseh, and M.A. Alhanouti. Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle. *Applied Mathematical Modelling*, 37(7):5315–5325, 2013. (Cited on p. 15)
- [3] J. Andersson, J. Åkesson, and M. Diehl. Casadi: A symbolic package for automatic differentiation and optimal control. In *Recent Advances in Algorithmic Differentiation*, pages 297–307. Springer, 2012. (Cited on p. 15)
- [4] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N.P. Papanikolopoulos. A vision-based approach to collision prediction at traffic intersections. *Intelligent Transportation Systems, IEEE Transactions on*, 6(4):416–423, 2005. (Cited on p. 11)
- [5] S. Behere, M. Törngren, and D-J. Chen. A reference architecture for cooperative driving. *Journal of systems architecture*, 59(10):1095–1112, 2013. (Cited on p. 11)
- [6] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957. (Cited on p. 13)
- [7] A. Bemporad, M. Morari, and N.L. Ricker. Model predictive control toolbox. Matlab User’s Guide, 1995. version 1. (Cited on p. 10)
- [8] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. *Proceedings of the IEEE*, 90(7):1258–1271, 2002. (Cited on p. 11)
- [9] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3 edition, 2005. (Cited on p. 13)
- [10] J.T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998. (Cited on p. 14, 15)
- [11] R. Bishop. A survey of intelligent vehicle applications worldwide. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 25–30. IEEE, 2000. (Cited on p. 11)
- [12] A. Bonchis, N. Hillier, J. Ryde, E. Duff, and C. Pradalier. Experiments in autonomous earth moving. In *IFAC World Congress*, pages 11588–11593. IFAC, 2011. (Cited on p. 15)
- [13] M. Burke. Powertrain efficiency optimization of the torotrak infinitely variable transmission

- (ivt). In *Transmission & Driveline Systems Symposium*, number 2003-01-0971. SAE, 2003. (Cited on p. 5)
- [14] E.F. Camacho and C. Bordons. *Model predictive control*. Springer, 2 edition, 2013. (Cited on p. 10)
- [15] M. Diehl. Script on numerical optimal control by m.diehl. <http://homes.esat.kuleuven.be/~mdiehl/TRENTO/numopticon.pdf>, 2011. (Cited on p. 14)
- [16] A. Emadi, K. Rajashekara, S.S. Williamson, and S.M. Lukic. Topological overview of hybrid electric and fuel cell vehicular power system architectures and configurations. *Vehicular Technology, IEEE Transactions on*, 54(3):763–770, 2005. (Cited on p. 5)
- [17] E. Faggioli, P. Rena, V. Danel, X. Andrieu, R. Mallant, and H. Kahlen. Supercapacitors for the energy management of electric vehicles. *Journal of Power Sources*, 84(2):261–269, 1999. (Cited on p. 5)
- [18] R. Filla. Alternative systems solutions for wheel loaders and other construction equipment. In *1st International CTI Forum Alternative and Hybrid Drive Trains*. CTI, 2008. (Cited on p. 4)
- [19] R. Filla. *Quantifying Operability of Working Machines*. dissertation, Linköping University, 2011. (Cited on p. 6)
- [20] R. Filla. Study of a method for assessing operability of working machines in physical and virtual testing. *International Journal of Vehicle Systems Modelling and Testing*, 7(3):209–234, 2011. (Cited on p. 18)
- [21] T. Gandhi and M.M. Trivedi. Pedestrian collision avoidance systems: A survey of computer vision based recent studies. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 976–981. IEEE, 2006. (Cited on p. 11)
- [22] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989. (Cited on p. 10)
- [23] R. Ghabelloo, M. Hyvönen, J. Uusitalo, O. Karhu, J. Jara, and K. Huhtala. Autonomous motion control of a wheel loader. In *Proceedings of the ASME 2009 Dynamic Systems and Control Conference*, pages 1339–1346. ASME, 2009. (Cited on p. 15)
- [24] D. Gillespie. *Markov Processes; An Introduction for Physical Scientists*. Academic Press, 1991. (Cited on p. 12)
- [25] Robert Bosch GmbH. *Bosch Automotive Handbook*. John Wiley & Sons Ltd., 7 edition, 2007. (Cited on p. 3, 4)
- [26] C. Guardiola, B. Pla, D. Blanco-Rodriguez, and A. Reig. Modelling driving behaviour and its impact on the energy management problem in hybrid electric vehicles. *International Journal of Computer Mathematics*, 91(1):147–156, 2014. (Cited on p. 6)
- [27] L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems*. Springer Verlag, 2 edition, 2007. (Cited on p. 3)
- [28] M.J. Hapeman, J. Long, and D.L. Plette. Diesel electric locomotive propulsion systems—a look into the future. *Industry Applications, IEEE Transactions on*, (3):495–501, 1986. (Cited on p. 5)
- [29] E. Hellström. *Look-ahead Control of Heavy Vehicles*. dissertation, Linköping University, 2010. (Cited on p. 6, 11, 13)
- [30] L.O. Hewko and T.R. Weber. Hydraulic energy storage based hybrid propulsion system for a terrestrial vehicle. In *Energy Conversion Engineering Conference, 1990. IECEC-90. Proceedings of the 25th Intersociety*, volume 4, pages 99–105. IEEE, 1990. (Cited on p. 5)
- [31] D.W. Hinde and M. Hinde. *Electric and diesel-electric locomotives*. Macmillan, 1948. (Cited on p. 5)
- [32] B. Holger, J. Emmert, and K. Dietmayer. Continuous driver intention recognition with hidden markov models. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 1189–1194. IEEE, 2008. (Cited on p. 12)
- [33] J-L. Hsu, C-C. Liu, and A.L.P. Chen. Discovering nontrivial repeating patterns in music data. *Multimedia, IEEE Transactions on*, 3(3):311–325, 2001. (Cited on p. 22)
- [34] S. Jeon, S. Jo, Y. Park, and J. Lee. Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition. *Journal of dynamic systems, measurement, and control*, 124(1):141–149, 2002. (Cited on p. 12)
- [35] L. Johannesson, M. Åsbogård, and B. Egardt. Assessing the potential of predictive control

- for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):71–83, 2007. (Cited on p. 6, 12, 13)
- [36] H. Khayyam, S. Nahavandi, and S. Davis. Adaptive cruise control look-ahead system for energy management of vehicles. *Expert Systems with Applications*, 39(3):3874–3885, 2012. (Cited on p. 6)
- [37] A.J. Kotwicki. Dynamic models for torque converter equipped vehicles. In *SAE technical papers*, number 820393. SAE, 1982. (Cited on p. 3)
- [38] J. Krumm. A markov model for driver turn prediction. In *SAE technical papers*, number 2008-01-0195. SAE, 2008. (Cited on p. 12)
- [39] T. Leroy, F. Vidal-Naquet, and P. Tona. Stochastic dynamic programming based energy management of hev’s: an experimental validation. In *IFAC World Congress*, pages 4813–4818. IFAC, 2014. (Cited on p. 6, 12, 13)
- [40] L. Li and F. Wang. Cooperative driving at blind crossings using intervehicle communication. *Vehicular Technology, IEEE Transactions on*, 55(6):1712–1724, 2006. (Cited on p. 11)
- [41] C. Lin, S. Jeon, H. Peng, and J. Moo. Driving pattern recognition for control of hybrid electric trucks. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 42(1-2):41–58, 2004. (Cited on p. 12)
- [42] C. Lin, H. Peng, and J.W. Grizzle. A stochastic control strategy for hybrid electric vehicles. In *American Control Conference, 2004. Proceedings of the 2004*, volume 5, pages 4710–4715. IEEE, 2004. (Cited on p. 12)
- [43] J. Liouville. Note sur la théorie de la variation des constantes arbitraires. *Journal de Mathématiques Pures et Appliquées*, 3:342–349, 1838. (Cited on p. 14)
- [44] S. Liu and B. Paden. A survey of today’s cvt controls. In *Proceedings of the 36th Conference on Decision and Control*, pages 4738–4743. IEEE, 1997. (Cited on p. 5)
- [45] S.E. Lyshevski. Energy conversion and optimal energy management in diesel-electric drivetrains of hybrid-electric vehicles. *Energy conversion and management*, 41(1):13–24, 2000. (Cited on p. 5)
- [46] G. Maggetto and J. Van Mierlo. Electric vehicles, hybrid electric vehicles and fuel cell electric vehicles: state of the art and perspectives. In *Annales de Chimie Science des Matériaux*, volume 26, pages 9–26. Elsevier, 2001. (Cited on p. 5)
- [47] M. Magnusson and H. Almqvist. Consistent pile-shape quantification for autonomous wheel loaders. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4078–4083. IEEE, 2011. (Cited on p. 15)
- [48] L. Mangialardi and G. Mantriota. Power flows and efficiency in infinitely variable transmissions. *Mechanism and machine theory*, 34(7):973–994, 1999. (Cited on p. 4)
- [49] J-P. Marec. *Optimal space trajectories*, volume 1. Elsevier Scientific Publishing Company, 1979. (Cited on p. 14)
- [50] P. Mattsson and M. Åkerblom. Continuously variable transmission and a working maching including a continuously variable transmission. Patent, 2012. WO 2012/008884 A1. (Cited on p. 8)
- [51] Caterpillar. Products: Wheel loaders, 2014. http://www.cat.com/en_US/products/new/equipment/wheel-loaders.html. (Cited on p. 3)
- [52] Hyundai Heavy Industries. Products: Wheel loaders, 2014. <http://constructionequipment.hyundai.eu/en/products/wheeled-loaders-loaders>. (Cited on p. 3)
- [53] Kawasaki Construction Machinery. Products: Wheel loaders, 2014. <http://www.kawasakiloaders.com/products/zv-2.aspx>. (Cited on p. 3)
- [54] Komatsu. Products: Wheel loaders, 2014. <http://www.komatsuamerica.com/equipment/wheelloaders>. (Cited on p. 3)
- [55] LiuGong. Products: Wheel loaders, 2014. http://www.liugong.com/en_me/products/loader.htm. (Cited on p. 3)
- [56] Volvo Construction Equipment. Products: Wheel loaders, 2014. <http://www.volvoce.com/constructionequipment/na/en-us/products/wheelloaders/wheelloaders/Pages/introduction.aspx>. (Cited on p. 3)
- [57] K. McDonough, I. Kolmanovsky, D. Filev, D. Yanakiev, S. Szwabowski, and J. Michelini. Stochastic dynamic programming control policies for fuel efficient in-traffic driving. In *American Control Conference (ACC), 2012*, pages 3986–3991. IEEE, 2012. (Cited on p. 12, 13)

- [58] D. Mitrovic. Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6:198–205, 2005. (Cited on p. 12)
- [59] S.J. Moura, H.K. Fathy, D.S. Callaway, and J.L. Stein. A stochastic optimal control approach for power management in plug-in hybrid electric vehicles. *Control Systems Technology, IEEE Transactions on*, 19(3):545–555, 2011. (Cited on p. 12)
- [60] K.B. Murray, D. Gorse, and J.M. Thornton. Wavelet transforms for the characterization and detection of repeating motifs. *Journal of molecular biology*, 316(2):341–363, 2002. (Cited on p. 22)
- [61] Y. Nambu. Generalized hamiltonian dynamics. *Physical Review D*, 7(8):2405, 1973. (Cited on p. 14)
- [62] V. Nezhadali and L. Eriksson. Modeling and optimal control of a wheel loader in the lift-transport section of the short loading cycle. In *7th IFAC Symposium on Advances in Automotive Control*, pages 195–200. IFAC, 2013. (Cited on p. 3)
- [63] T. Nilsson. Optimal engine operation in a multi-mode CVT wheel loader. Technical report, Linköping University, 2012. LiU-TEK-LIC-2012:32, Thesis No. 1547. (Cited on p. 7)
- [64] T. Nilsson, A. Fröberg, and J. Åslund. Optimized engine transients. In *7th IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, 2011. (Cited on p. 7)
- [65] T. Nilsson, A. Fröberg, and J. Åslund. Fuel potential and prediction sensitivity of a power-split cvt in a wheel loader. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, pages 49–56. IFAC, 2012. (Cited on p. 7, 8)
- [66] T. Nilsson, A. Fröberg, and J. Åslund. On the use of stochastic dynamic programming for evaluating a power-split cvt in a wheel loader. In *8th IEEE Vehicle Power and Propulsion Conference*, pages 840–845. IEEE, 2012. (Cited on p. 7, 8)
- [67] T. Nilsson, A. Fröberg, and J. Åslund. Optimal operation of a turbocharged diesel engine during transients. *SAE International Journal of Engines*, 5(2):571–578, 2012. (Cited on p. 5, 7)
- [68] T. Nilsson, A. Fröberg, and J. Åslund. Fuel and time minimization in a cvt wheel loader application. In *7th IFAC Symposium on Advances in Automotive Control*. IFAC, 2013. (Cited on p. 7, 8)
- [69] T. Nilsson, A. Fröberg, and J. Åslund. Development of look-ahead controller concepts for a wheel loader application. *Oil & Gas Science and Technology - Rev. IFP*, 2014. (Cited on p. 3, 6, 7, 8)
- [70] T. Nilsson, A. Fröberg, and J. Åslund. Minimizing fuel use during power transients for naturally aspirated and turbo charged diesel engines. Technical Report LiTH-R-3077, Department of Electrical Engineering, Linköpings Universitet, SE-581 83 Linköping, Sweden, 2014. (Cited on p. 6, 7)
- [71] T. Nilsson, A. Fröberg, and J. Åslund. Using stochastic dynamic programming for look-ahead control of a wheel loader diesel electric transmission. In *IFAC World Congress*, pages 6630–6635. IFAC, 2014. (Cited on p. 7, 8)
- [72] T. Nilsson, A. Fröberg, and J. Åslund. Predictive control of a diesel electric wheel loader powertrain. *Control Engineering Practice*, Submitted. (Cited on p. 6, 7, 8)
- [73] T. Nilsson, C. Sundström, P. Nyberg, E. Frisk, and M. Krysander. Robust driving pattern detection and identification with a wheel loader application. *International Journal of Vehicle Systems Modelling and Testing*, 9(1):56–76, 2014. (Cited on p. 6, 7, 19, 22)
- [74] E. Ohn-Bar, A. Tawari, S. Martin, and M.M. Trivedi. Predicting driver maneuvers by learning holistic features. In *Intelligent Vehicles Symposium Proceedings*, pages 719–724. IEEE, 2014. (Cited on p. 12)
- [75] G. Paganelli, T.M. Guerra, S. Delprat, J. Santin, M. Delhom, and 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*, 214:705–717, 2000. (Cited on p. 14)
- [76] R. Pfäffner. *Optimal Operation of CVT-Based Powertrains*. dissertation, ETH, Zurich, 2001. (Cited on p. 5, 6)
- [77] J. Ploeg, A. Serrarens, and G.J. Heijenk. Connect & drive: design and evaluation of cooperative adaptive cruise control for congestion reduction. *Journal of Modern Transportation*, 19(3):207–213, 2011. (Cited on p. 11)
- [78] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. *The Mathemat-*

- ical Theory of Optimal Processes*. Interscience Publishers, 1962. (Cited on p. 14)
- [79] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003. (Cited on p. 10)
- [80] D. Revuz. *Markov Chains*. Elsevier Science Publishers, 1975. (Cited on p. 12)
- [81] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky. A stochastic model predictive control approach for series hybrid electric vehicle power management. In *American Control Conference (ACC), 2010*, pages 5844–5849. IEEE, 2010. (Cited on p. 12)
- [82] H. Salmane, Y. Ruichek, and L. Khoudour. Using hidden markov model and dempster-shafer theory for evaluating and detecting dangerous situations in level crossing environments. In *Advances in Artificial Intelligence*, pages 131–145. Springer, 2013. (Cited on p. 12)
- [83] A. Sathyanarayana, P. Boyraz, and J. Hansen. Driver behavior analysis and route recognition by hidden markov models. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 276–281. IEEE, 2008. (Cited on p. 12)
- [84] A. Sciarretta and L. Guzzella. Control of hybrid electric vehicles. *Control Systems, IEEE*, 27:60–70, 2007. (Cited on p. 14)
- [85] S. Shen and F.E. Veldpaus. Analysis and control of a flywheel hybrid vehicular powertrain. *IEEE Transactions on Control Systems Technology*, 5:645–660, 2004. (Cited on p. 5)
- [86] M. Sivertsson and L. Eriksson. Time and fuel optimal power response of a diesel-electric powertrain. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, pages 262–269. IFAC, 2012. (Cited on p. 5)
- [87] M. Sivertsson and L. Eriksson. Optimal transient control trajectories in diesel-electric systems-part 1: Modeling, problem formulation and engine properties. *Journal of Engineering for Gas Turbines and Power*, 137(2), 2015. (Cited on p. 6)
- [88] E. Soei, C. Bellebaum, and I. Daum. Relational and non-relational memory-electrophysiological correlates of novelty detection, repetition detection and subsequent memory. *European Journal of Neuroscience*, 29(2):388–398, 2009. (Cited on p. 22)
- [89] N. Srinivasa. Vision-based vehicle detection and tracking method for forward collision warning in automobiles. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 626–631. IEEE, 2002. (Cited on p. 11)
- [90] N. Srivastava and I. Haque. A review on belt and chain continuously variable transmissions (cvt): dynamics and control. *Mechanism and Machine Theory*, 44:19–41, 2009. (Cited on p. 5)
- [91] J. Stecki and P. Matheson. Advances in automotive hydraulic hybrid drives. In *Proceedings of the 6th JFPS International Symposium on Fluid Power*, pages 664–669. JFPS, 2005. (Cited on p. 5)
- [92] G. Stein, A. Fröberg, J. Martinsson, B. Brattberg, R. Filla, and J. Unneback. Fuel efficiency in construction equipment - optimize the machine as one system. In *7th AVL International Commercial Powertrain Conference*. AVL & SAE, 2013. (Cited on p. 8)
- [93] R.F. Stengel. *Optimal control and estimation*. Courier Dover Publications, 2012. (Cited on p. 15)
- [94] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of operations research*, 37(1):357–373, 1992. (Cited on p. 14, 15)
- [95] J. Wang, W. Xu, and Y. Gong. Real-time driving danger-level prediction. *Engineering Applications of Artificial Intelligence*, 23(8):1247–1254, 2010. (Cited on p. 12)
- [96] L. Wang, E.S. Chng, and H. Li. Efficient sparse self-similarity matrix construction for repeating sequence detection. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 458–461. IEEE, 2009. (Cited on p. 22)
- [97] C.O. Weisenbach. Hydrostatic transmission applications. In *19th Annual Earthmoving Industry Conference*, number 680259. SAE, 1968. (Cited on p. 5)
- [98] J. Wenger. Automotive radar-status and perspectives. In *Compound Semiconductor Integrated Circuit Symposium, 2005. CSIC'05. IEEE*, pages 4–pp. IEEE, 2005. (Cited on p. 11)

Papers

Robust driving pattern detection and identification with a wheel loader application[†]

**Tomas Nilsson, Peter Nyberg, Christofer Sundström, Erik Frisk,
and Mattias Krysander**

*Vehicular Systems, Department of Electrical Engineering,
Linköping University, S-581 83 Linköping, Sweden.*

Abstract

Information about wheel loader usage can be used in several ways to optimize customer adaption. First, optimizing the configuration and component sizing of a wheel loader to customer needs can lead to a significant improvement in e.g. fuel efficiency and cost. Second, relevant driving cycles to be used in the development of wheel loaders can be extracted from usage data. Third, online usage identification opens up for the possibility of implementing advanced look-ahead control strategies for wheel loader operation.

The main objective of this paper is to develop an online algorithm that automatically, using production sensors only, can extract information about the usage of a machine. Two main challenges are that sensors are not located with respect to this task and that significant usage disturbances typically occur during operation. The proposed method is based on a combination of several individually simple techniques using signal processing, state automaton techniques, and parameter estimation algorithms. The approach is found to be robust when evaluated on measured data of wheel loaders loading gravel and shot rock.

[†]This is a formatted version of “Robust driving pattern detection and identification with a wheel loader application” by Tomas Nilsson, Peter Nyberg, Christofer Sundström, Erik Frisk, and Mattias Krysander, International Journal of Vehicle Systems Modelling and Testing, 2014 Volume 9, Number 1, pages 56-76. ©InderScience Publishers 2014. Reproduced with the permission of InderScience Publishers. The original paper can be found at <http://www.inderscience.com>, and by using the Digital Object Identifier (DOI): 10.1504/IJVSMT.2014.059156. The formatting is restricted to changing the article into a single-column format, adjusting sizes of figures and tables, and adjusting the referencing style.

1 Introduction

Wheel loaders are used for a wide variety of tasks, ranging from use as snow-plows to loading gravel or pallets onto trucks. This work concerns characterization of a specific customer's vehicle usage. For example, experience shows that proper matching of machine configuration, such as sizing of the thermal management system [1], to the customer profile can have significant influence on machine efficiency and reduce fuel consumption. Since many customers operate their wheel loaders mainly for specific tasks throughout the entire lifespan, there is a potential for significant efficiency improvement. Other motivating examples where knowing the driving behavior is beneficial are advanced predictive engine control and automatic gear shifting algorithms. Good estimates of future power trajectories can be utilized in controls to further improve efficiency resulting in lower purchase cost, higher productivity, and lower fuel consumption. A third area where knowing the customer's usage profile is beneficial is during development and evaluation of control algorithms. Good knowledge of customer usage makes it possible to simulate representative driving cycles and thereby obtaining more relevant evaluation results.

Today, a common situation is that only rough estimates, typically averaged quantities over long periods of time, about customer usage is available. Therefore, for example, customer adaption is based on qualified guesses and test drive experience and little adaptation to a particular customer's needs is possible. This situation is the main motivation for this work; to develop an algorithm that, using production sensors only, automatically extracts detailed information from customer's vehicles during operation. The output of the algorithm should support improved matching of vehicle configuration and customer usage, provide driving information for online adaption of engine control and automatic gear shifting, and provide data for generating relevant driving cycles to simulate during product development, such as ride comfort considerations [2]. For the control application, it must be possible to run the algorithm in real time on board the vehicle. Main challenges are to first define what information that is relevant and then to perform online usage identification robust against significant usage disturbances.

Related works for on-road automotive vehicles are for example [3, 4, 5], showing a potential to increase vehicle efficiency by using driving pattern knowledge. For construction machines this task is more complicated since, for example, the vehicle does not follow a given road. Control algorithms for hybrid electric vehicles based on pattern recognition are developed in [6] and [7]. A main difference is that the key objective of this work is to analyze usage patterns, not to design a control algorithm. The key contribution of this paper is an algorithm, seamlessly integrating techniques from automata theory [8, 9] and system identification [10], for detecting wheel loader operating patterns, which is robust against large usage disturbances that inherently affects vehicle operation.

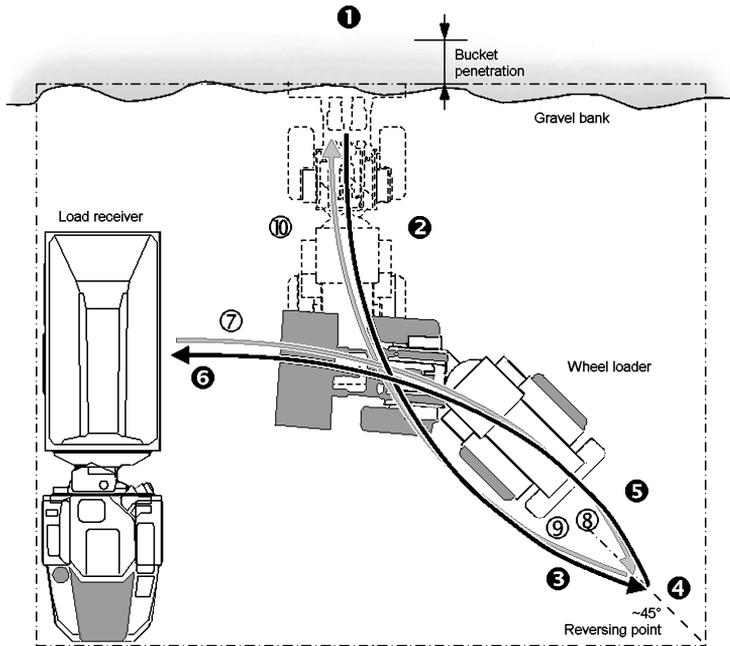


Figure 1: A view of a short loading cycle [11].

2 Problem formulation and challenges

Before the main problem is formulated together with some of the main challenges, a brief introduction to typical wheel loader usage and sensor configuration will be given.

2.1 Wheel loader usage

Figure 1 illustrates common usage of a wheel loader, where gravel is loaded onto an articulated hauler.

This loading mission consists of repeating the cycle of filling the bucket at the pile and emptying it at the receiving hauler. In the figure, the loader is starting from point 4, driving towards the pile for filling the bucket at point 1. The bucket is pushed into the pile, lifted and tilted up at loading. After reversing the loader to point 4 it approaches the receiver at point 6. The loaded bucket is lifted during the re-positioning from the pile to the hauler. After emptying the bucket the loader reverses to point 4 while lowering the bucket.

The load receiver is often, just as in Figure 1, a truck or a hauler. In this case, when the hauler is full, there will be a pause in the loading. During this pause the machine is often cleaning the working site or waiting in a dormant state for a new hauler to arrive for loading.

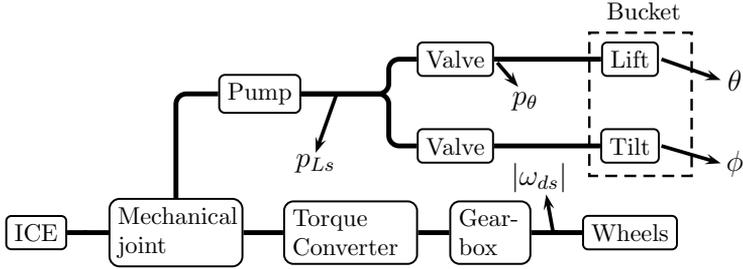


Figure 2: The configuration of the vehicle powered by the internal combustion engine, ICE. The pressure in the lift cylinder of the bucket p_θ and the Ls pressure p_{Ls} are two pressures in the system where only p_{Ls} is known in production loaders. The bucket lift and tilt angles, and the angular speed of the drive shaft are denoted θ , ϕ , and $|\omega_{ds}|$, respectively.

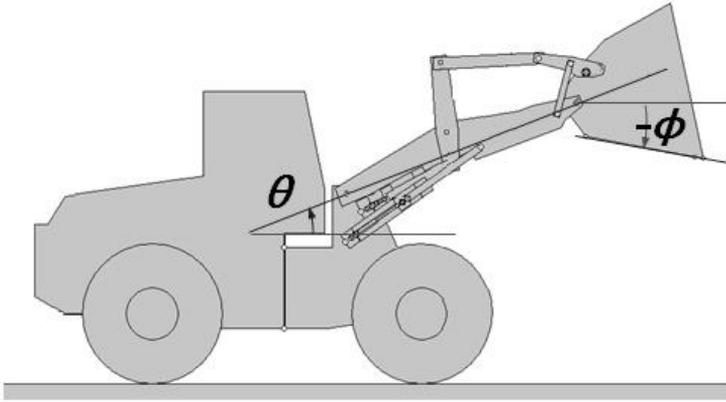


Figure 3: Side view of a wheel loader, indicating the lift angle θ and the bucket tilt angle ϕ .

2.2 Sensors configuration and measurement data

Figure 2 shows a schematic view of the vehicle where important measured signals are included. The production sensors used here are measuring the lift angle θ and the tilt angle ϕ of the bucket, as defined in Figure 3, the pressure p_{Ls} in the load sensing hydraulic pump and the angular speed $|\omega_{ds}|$ of the drive shaft. The vehicle has a four speed gearbox and a forward/reverse gear in series, and both selected gear and gear direction is known. Finally, the driver inputs, such as hydraulics controls usage, are available. The sensor measuring pressure p_θ in the lift cylinder of the bucket is not a production sensor and will therefore only be used as a reference.

Figure 4 shows an example of measured vehicle velocity and bucket lift and tilt angles during typical wheel loader operation. The data has been collected during a sequence of loading cycles, similar to the one described in Section 2.1,

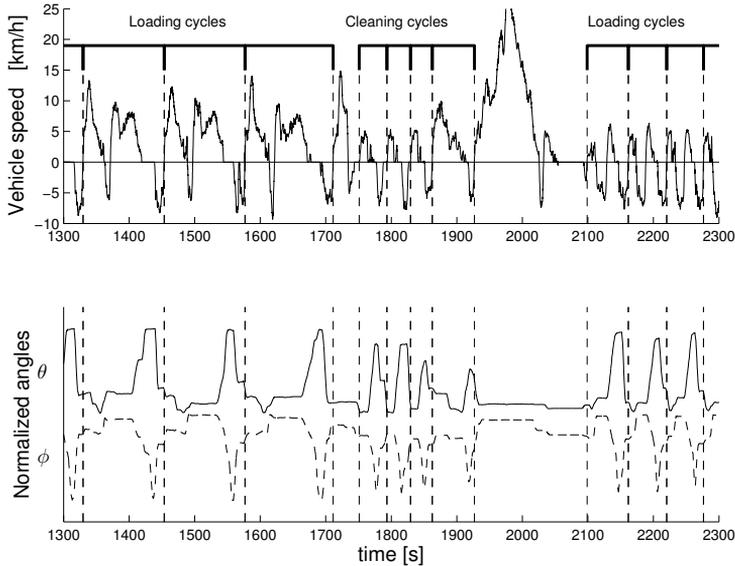


Figure 4: Data collected during typical wheel loader operation. in this case there are six loading cycles and four cleaning cycles.

and cleaning cycles in which dropped gravel is moved back to the pile. This type of data is the input to the cycle identification algorithm.

It is common that a significant part of the total energy consumption is required for lifting the bucket with its load and knowledge of load mass is therefore important for engine control. Also, since moving material is the purpose of the operation illustrated in Figure 1, relevant efficiency measures are the mass of the material moved divided by the time needed or the fuel used. It is therefore important to keep track of the amount of material handled. The bucket load is not measured directly, and estimating the load is not trivial, especially if no additional sensors can be introduced.

2.3 Problem formulation

This paper treats the development of an algorithm for online cycle detection and identification using only production sensors. The intended output of the algorithm is illustrated in Figure 4 where cycle detection has been performed. In addition to the identified cycles, important usage parameters such as the bucket load are also automatically estimated.

2.4 Challenges

The cycle identification problem is challenging because the specification of different types of cycles are based on what the driver think is done in a given mission. Figure 4 shows six loading cycles which all perfectly match the description, but have significantly different signal trajectories with respect to amplitudes and

lengths in parts of the cycle. In this case the differences are caused by the operator not driving fully repetitive, but in general also the driving style differs, different material handled such as shot rock or gravel requires different operations and the geometry of the site of operation differs. For successful cycle identification the algorithm must be robust against these types of disturbances.

The benefits of the cycle detector in parameter estimation will be exemplified with bucket load estimation. Figure 5 shows a loading cycle, according to Figure 1, with a 50s transportation between points 4 and 6. The first peak in p_{Ls} is caused by the bucket filling and the second peak is caused by the raising of the bucket before emptying. Estimating the bucket load from the pressure p_θ in the bucket lift cylinder when the machine carries a load would be straightforward, but the pressure p_θ is not measured in production vehicles which means that the estimation algorithm can only use the load sensing pressure p_{Ls} . This makes the estimation problem more challenging since p_{Ls} supplies pressure to all hydraulics including lifting and tilting. Furthermore, the pressures p_θ and p_{Ls} are only similar at specific operating modes, as illustrated in Figure 5. This indicates that intelligent partitioning of the measurements are needed to extract exactly those parts that are useful for bucket load estimation.

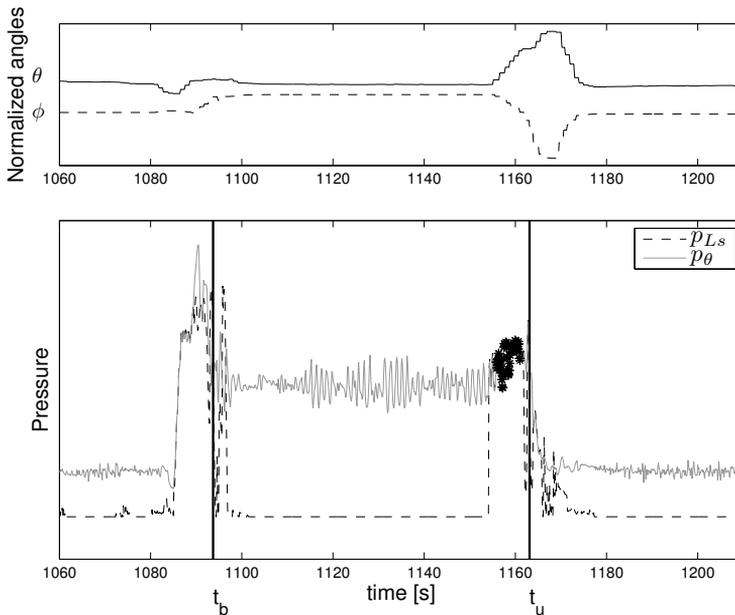


Figure 5: The load sensing pump and lift cylinder pressures, p_{Ls} and p_θ , during a transport. The bucket is loaded during the first pressure peak and unloaded at the second pressure peak. Time $t = t_b$ corresponds to where the wheel loader starts backward motion and $t = t_u$ when the bucket is unloaded.

3 Modeling

The objective of this section is to introduce a simple way of modeling operation of the wheel loader. Key properties of such a model are 1) it should be possible to model cyclic behavior and 2) support the extraction of important operation parameters such as bucket load mass and distance traveled etc. This paper propose modeling the cycles as sequences of discrete events, in which the events are at a low enough complexity for robustly being detected while diverse enough to uniquely describe the driving cycles. This structure of building cycles from events enable using standard automata theory [8] to devise cycle detection methods in Section 4. Automatic extraction of models directly from measured data would be interesting, for example using symbolic data mining techniques [12]. Here, it is assumed that the models are made by hand using engineering knowledge. Due to the high level of abstraction of the models, this has proven to be a relatively easy task. The models presented here are focused on bucket handling, but the methodology has also been used for other types of cycles, e.g., timber and pallet handling in [13], by additional modeling of events and cycles.

3.1 Events

The principal modeling object that is used is called an *event* which represents a specific occurrence in time. First a set of events need to be introduced. These should be simple enough to be robustly, with respect to usage disturbances, detected using measurement data and still diverse enough to describe the cycles.

The primary operation described in this paper is handling of gravel and shot rocks using a bucket. An analysis indicates that six events can be used to describe the driving cycles in this type of operation. These are:

- ★ transition from dormant to action - a
- ★ transition from action to dormant - d
- ★ transition from backward to forward motion - f
- ★ transition from forward to backward motion - b
- ★ bucket loading - l
- ★ bucket unloading - u

In the illustration shown in Figure 1, event f happens at point 4, and point 1 and 6 are positions where event b happens. Event l occurs near point 1, and event u near point 6. The events a and d usually occur when the receiver has been filled, and the wheel loader is dormant while waiting for a new receiver to arrive.

3.2 Event descriptions

For the formal statement of the models, the notation $z_k = z(t_k)$ is used, and the time intervals are denoted $\mathcal{I}_k^l = \{t_k, \dots, t_l\}$.

If the velocity v is 0 and if the lift angle θ and the tilt angle ϕ are constant during the $\epsilon+1$ last time samples $t_i \in \mathcal{I}_{k-\epsilon}^k$, then a transition $d_{k-\epsilon}$ from action to dormant operation is detected at time t_k . The parameter ϵ is a model parameter and will be discussed in Section 4.1. Formally, an event $d_{k-\epsilon}$ is generated if

$$(v_j = 0 \text{ and } \theta_i = \theta_j \text{ and } \phi_i = \phi_j) \forall t_i, t_j \in \mathcal{I}_{k-\epsilon}^k, \quad (1)$$

Note that the dormant event $d_{k-\epsilon}$ is detected in the end of the interval, i.e. at time t_k , but time stamped with the starting time $t_{k-\epsilon}$.

An activation event a_k is detected if the vehicle has been dormant according to the definition in (1) and starts to move, i.e. if (1) holds for $\mathcal{I}_{k-\epsilon-1}^{k-1}$ and

$$v_k \neq 0 \text{ or } \theta_{k-1} \neq \theta_k \text{ or } \phi_{k-1} \neq \phi_k. \quad (2)$$

The events f and b are straightforward to define since v is a processed signal that has no zero-crossing noise. An event f_k is generated if there exists an interval \mathcal{I}_j^k , where the velocity is negative at the start of the interval, positive at the end of the interval, and 0 in the, possibly empty, time in between. Formally, this translates into

$$v_j < 0 \text{ and } v_{j+1} = \dots = v_{k-1} = 0 \text{ and } v_k > 0, \quad (3)$$

where v is the vehicle velocity, computed from the drive shaft angular speed $|\omega_{ds}|$ and if a forward or reverse gear is selected. Note that the length of the interval is not fixed, but will depend on the number of consecutive time instances with 0 velocity. A corresponding condition for b_k is then

$$v_j > 0 \text{ and } v_{j+1} = \dots = v_{k-1} = 0 \text{ and } v_k < 0. \quad (4)$$

A bucket unloading event u is detected when the tilt angle ϕ is small enough, i.e.

$$u_k \text{ is generated if } \phi_{k-1} \geq \xi \text{ and } \phi_k < \xi, \quad (5)$$

where ξ is a model parameter. As can be seen in Figure 4, the tilt angle ϕ is given in discrete levels and not affected by noise which means that only a single unloading event is generated when the tilt angle is monotonously decreased.

The bucket loading event l is a bit more complex and is assumed to have happened if both the lift angle θ and the tilt angle ϕ has increased significantly over a time window while the machine is moving forward, i.e.

$$l_{k-L} \text{ is generated if } \theta_k - \theta_{k-L} > \alpha \text{ and } \phi_k - \phi_{k-L} > \beta \text{ and } v_{k-L} > 0, \quad (6)$$

where α and β are model constants and L the length of the time window. Note that this event is time-stamped at the start of the time window and not at the end. In contrast to the unloading event, the loading event can be generated multiple times during one bucket loading.

With the event descriptions (1)-(6) measurement data of velocity $v(t)$, lift angle $\theta(t)$, and tilt angle $\phi(t)$ can be transformed into a sequence of symbols from the alphabet $\Sigma = \{a, b, d, f, l, u\}$ with corresponding time stamps.

3.3 Cycles

As discussed in Section 2, repetitive behavior called cycles is of special importance. Here, cycles will be modeled using the events defined in Section 3.1 as a state automaton. The start time, $t_{c,s}$, of a cycle is determined by the first event and the end time, $t_{c,e}$, is determined by the event following the last event in the cycle.

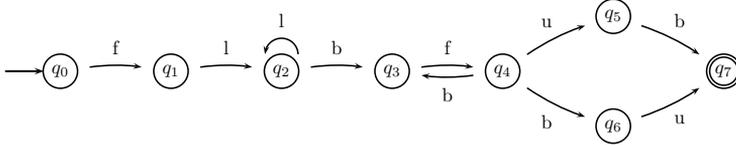


Figure 6: Transition diagram of the automata describing a loading cycle. The initial state is q_0 and the accepting state is q_7 .

It is possible to model any number of cycles using automata and here three common types of behavior will be modeled, *loading cycle*, *cleaning cycle*, and *dormant operation*. A loading cycle is intuitively described by the sequence of events $flbfub$ which is accepted by the automaton in Figure 6 by going through the states q_0 , q_1 , q_2 , q_3 , q_4 , q_5 , and q_7 . The rationale behind the model can be realized by going through the event sequence typically generated in the loading cycle shown in Figure 1. The event f is generated at point 4, l at point 1, b at point 1, f at point 4, u at point 6, and finally b at point 6. However if we only search for the ideal sequence $flbfub$, the order of events are crucial to get a fit. Due to minor changes in operator behavior, the events ub become bu and the reason is that these events occur near each other in time and small cycle-to-cycle variations affect the order of the events. For example, in Figure 1 this is common at point 6. Also possible multiple bucket loading events, l , generated at point 1 stresses that the patterns need to be robust against these disturbances. Due to the automaton modeling language used, regular expressions, it is straightforward to take such variations into account as is depicted in the full automata in Figure 6. The model for a cleaning cycle is slightly smaller but follows a similar structure as shown in Figure 7 and dormant operation is modeled as in Figure 8.

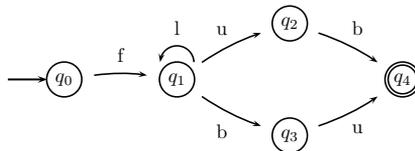


Figure 7: Transition diagram of the automata of the cleaning cycle. The initial state is q_0 and the accepting state is q_4 .

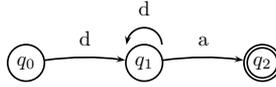


Figure 8: Transition diagram of the automata describing dormant operation. The initial state is q_0 and the accepting state is q_2 .

4 Method

In this section it is described how the models from the previous section can be used in an online algorithm for identifying wheel loader usage including cycle detection and usage parameter estimation. An overview of the different parts and the information flow of the algorithm is shown in Figure 9. The input to the algorithm is measurement data and the high-level cycle descriptions provided as automata like the ones given in Figures 6-8. The algorithm can be divided into three main parts: low-level event detection, cycle identification, and parameter estimation.

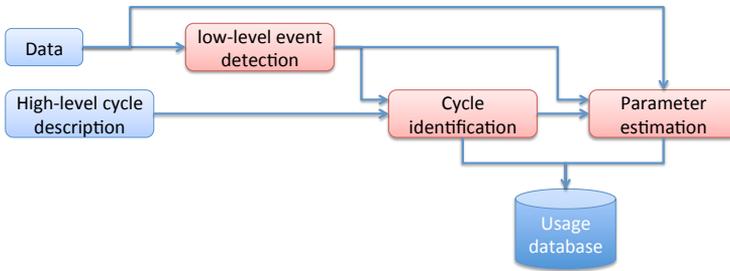


Figure 9: Data comes from sensors. The events are generated from the rules (1)-(6). The high-level description are the patterns that are fed to the cycle identification along with the generated events. Finally, parameters are estimated and stored together with cycle information in a usage database.

4.1 Event detection

The event detectors (1)-(6) take measurement data as input and generate a sequence of time-stamped events. The event sequence is then used, instead of whole data series of measurements, in the cycle identification to achieve robustness against user disturbances. In order to achieve reliable detection there are a few parameters in the event detectors that must be tuned. The choice of the parameter values is made with the aim of robustness in the detection of the events. The tuning is made by manually comparing video recorded sequences of wheel loader usage to the corresponding sensor data.

The time needed to robustly decide that the vehicle is dormant is determined by the parameter ϵ in (1). If ϵ is set low it means a risk of detecting dormant events in for example loading cycles and thus missing detection of these cycles.

A large value of ϵ implies that a short period of dormant operation would not be detected.

The value of ξ in (5) should be set at about the angle at which load would slide out of the bucket. The recorded data show no example of drivers tilting the bucket this low except when emptying and in that case the angle is usually much lower since this makes the emptying quicker. The sensitivity to the value of ξ is therefore low.

The values of α , β , and L , used in the bucket loading event as defined in (6), can be set according to common values in the recorded data. However, variations in these signals are large, which increase the risk of missed loading events. This is primarily handled by increasing the sensitivity of the loading event detector by decreasing α , β , and increasing L .

4.2 Cycle identification

The inputs to the cycle identification algorithm is an event sequence and a set of automata each describing a cycle type. The cycle identification is then performed each time a new event is detected.

Recall that events such as the loading event l and the dormant event d are not immediately detected. This means that when a new event is detected it is not sure that it will be the last one in the generated event sequence. In order to handle the non-causal behavior of incoming events all events after the latest matched cycle are the input to the cycle identification algorithm. The output is the set of states in the different cycles that is consistent with the considered event sequence. Consider as an example if the input event sequence is $flbfb$ then the output will be q_3 and q_6 in the loading cycle automaton in Figure 6 and q_3 in the cleaning cycle automaton in Figure 7. The string matching is done similar to the algorithms given in [14, 15].

There are words matching the loading cycle automaton such that the last part of the word also matches the cleaning cycle automaton and this causes a non-unique identification of cycles. To illustrate this, consider the word $flbfbub$ which matches the loading cycle automaton in Figure 6. The last part of this word, i.e. the three events fub also match the cleaning cycle automaton in Figure 7. To get a unique identification, the patterns are ordered according to a priority. Patterns with higher priority is matched first and for example, here the loading cycle has higher priority than the cleaning cycle, i.e., if the cleaning cycle is part of a longer sequence that can be interpreted as a loading cycle the latter interpretation is preferred. In this way matching coverage is maximized in this case.

User disturbances can lead to different event sequences, or variations of the sequences, for repetitions of the same type of cycle. To get a match even with cycle variation the automaton of the loading cycle in Figure 6 is made with this variation in mind. However, if the sequence of events does not match the automaton the algorithm regard this as a mismatch even if there is just a single difference between them. It works for this application but if there would be more and stochastic variations, approximate string matching techniques would be of interest [16, 17].

4.3 Parameter estimation

There are several parameters that are of interest in the analysis of how the vehicle is operated. Most of these are trivial, such as vehicle speed and gear selection, and need not be treated further. In this paper a suggested parameter for separation between short and long loading cycles and an estimation of the bucket load mass are presented since these require some additional analysis.

Separation between short and long loading cycles

The trivial way of estimating whether the loading cycle is short or long is to use the cycle time. If the time is above a threshold, the cycle is a long loading cycle, otherwise it is short. With this criteria a short loading cycle including a stop in the middle or with a toggle between the states q_3 and q_4 in Figure 6 could be classified as a long loading cycle. To get a more robust classification parameter for distinguishing between short and long cycles the longitudinal distance from the time t_l of the first loading event to the time t_u of the unloading event is used, i.e. if $v(t)$ is the velocity of the vehicle the classification parameter is

$$r = \int_{t_l}^{t_u} v(t) dt. \quad (7)$$

In a short loading cycle the loader reverses approximately the same distance as it is driving forward and r is close to zero. In a long loading cycle the wheel loader is driving longer distance forward than backward and r is a positive number. Therefore, r is calculated for every loading cycle and is compared to a threshold, ψ , and the classification of short and long cycles is determined according to

$$\begin{aligned} \text{short loading cycle: } & r \leq \psi \\ \text{long loading cycle: } & r > \psi. \end{aligned}$$

Here the threshold is set to $\psi = 30$ m.

Bucket load estimation

The bucket load mass is estimated once for each loading cycle. When estimating the load in the bucket, an affine relation between the pressure in the lift cylinder, p_θ , and the mass, m_{load} , relation is used

$$m_{load} = f(p_\theta) = a_\theta p_\theta + b_\theta, \quad (8)$$

where a_θ and b_θ are constants obtained by minimizing the least-squares error between the model (8) and a measured pressure-mass map. This simple relation between lift cylinder pressure and loaded mass has proven to work sufficiently well for our purposes but it is straightforward to use a more accurate description of the relation between pressure and load, e.g., as in [18]. If the machine operates at a significant ground slope angle, the accuracy of $f(p_\theta)$ increases if this angle is considered.

As stated previously, there is no sensor in series production loaders for measuring p_θ , but only the system pressure, p_{Ls} , at the hydraulic pump is known. The bucket load only affects p_{Ls} when the valve connecting the lifting cylinder to the pump is opened, i.e. during transient operation of the bucket. This can be seen in Figure 5, where these two pressures are given for a loading cycle, and at the two peaks in pressure, p_{Ls} is a good approximation of p_θ . However, it is beneficial to only base the load estimation on the second pressure peak where the bucket is lifted before it is unloaded. The reason is that when the bucket is lifted during loading, the pressure varies depending on if there are, e.g., roots in the pile or the material handled. The event and cycle identification algorithm is used to find the time interval to use in the estimation of the lift cylinder pressure, that later is used in the estimation of the bucket load. The time period of interest is stated to be after the first backward event after the loading event is generated in the loading cycle, t_b , and the unloading event is generated, t_u . The equality $p_{Ls} = p_\theta$ in the time interval $t \in [t_b, t_u]$, is assumed to be valid when two conditions are fulfilled. The first condition is based on that the driver lifts the bucket. The estimated angular velocity, $\hat{\theta}$, is used and the signal is smoothed by a low-pass Butterworth filter. The interval of interest is defined as

$$\mathcal{I}_1 = \left\{ t_i : \hat{\theta}(t_i) > \gamma \cdot \max_{t_k \in [t_b, t_u]} \{ \hat{\theta}(t_k) \} \right\}, \quad (9)$$

where $\gamma < 1$ is a tuning parameter. The condition uses a relative threshold to achieve as good performance as possible of the estimated pressure in the lift cylinder, \hat{p}_θ , for different driving situations. Due to robustness the second condition for the assumption $p_\theta = p_{Ls}$ is the interval

$$\mathcal{I}_2 = \left\{ t_i : p_{Ls}(t_i) > \delta \cdot \max_{t_k \in [t_b, t_u]} \{ p_{Ls}(t_k) \} \right\}, \quad (10)$$

where $\delta < 1$ is a tuning parameter. For time points in $\mathcal{I}_1 \cap \mathcal{I}_2$, the pressure p_θ is estimated according to

$$\hat{p}_\theta = \frac{1}{|\mathcal{I}_1 \cap \mathcal{I}_2|} \sum_{t_k \in \mathcal{I}_1 \cap \mathcal{I}_2} p_{Ls}(t_k). \quad (11)$$

The bucket load estimate is then

$$\hat{m}_{load} = f(\hat{p}_\theta) = a_\theta \hat{p}_\theta + b_\theta. \quad (12)$$

The samples where conditions \mathcal{I}_1 and \mathcal{I}_2 are fulfilled are marked with stars in Figure 5. Both γ and δ are used for removing data that is less accurate due to low signal amplitudes. Values close to 0 means most data is accepted and close to 1 that most data is rejected. As long as these extremes are avoided the sensitivity to the values is small. Here $\gamma = 0.65$ and $\delta = 0.50$ have been used.

5 Evaluation

It is hard to quantify how well the algorithm fulfills the requirements since it may be subjective which operation a wheel loader is performing in a particular situation. In addition, it is not obvious what should be considered to be usage disturbances within a specific cycle type and what should be considered to be completely different cycles. For example, it is common to adjust the position of the machine and/or shake the bucket during unloading, but the amount of deviation that should be allowed within a cycle is subjective. The proposed algorithm has been evaluated against real data where several drivers have used a machine, handling different materials, while being filmed. The resulting data and films have been used for evaluation of the algorithm. The drivers have been given a driving scenario, such as loading shot rock onto an artificial hauler, and instructed to drive as they would on a regular working day. The drivers have in most cases operated the vehicle in a cyclic behavior as described in Section 3, with occasional cleaning of the working site. Visual examination of these films has been used for creating a reference which can be compared to the result of the algorithm.

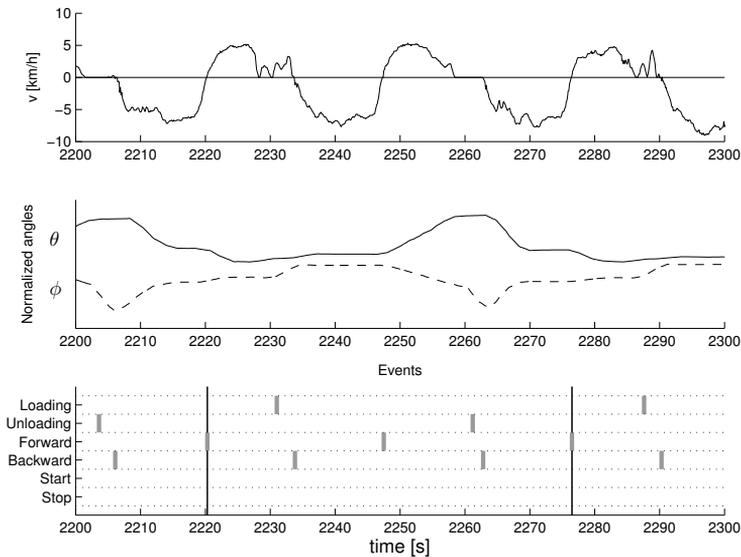


Figure 10: A data sequence with generated events and a detected loading cycle. The beginning and end of the loading cycle are marked with vertical lines in the lowest plot.

Figure 10 shows a close-up of one of these datasets. It shows the generated events and an identified loading cycle. In this particular case, the visible sequence of events is *ubflbfubflb*, and in this sequence a loading cycle *flbfub* has been identified. Figure 11 shows the full 3,600 seconds of the same dataset. The light-gray segments indicate identified loading cycles, the dark-gray segments indicate identified cleaning cycles, and the white segments correspond to the

parts that do not match any of the predefined cycles. The examination of this and the other datasets shows that the accuracy is in general very good. There are however some exceptions.

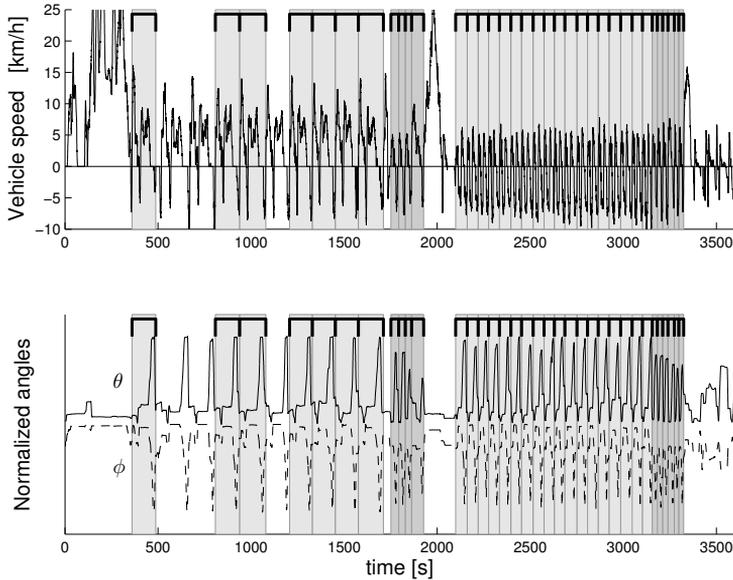


Figure 11: A cycle-partitioned dataset. White = no cycle, light gray = loading cycle, dark gray = cleaning cycle.

The first type is cases where the operator deliberately tries to drive in a peculiar way. The other type is at occasions, especially when handling shot rock, when the driver shakes the bucket at loading. It would be easy to adjust the loading cycle automata in Figure 6 to accept this behavior. Therefore, the main issue in these cases is to decide whether or not the cycle specifications are supposed to include the observed behavior as well. Without any adjustments of the models, the results are still valuable, since most of the operation is partitioned into cycles resulting in an accurate overview of the usage. The algorithm can therefore help directing attention to data regions with unusual operation.

5.1 Robustness of cycle identification algorithm

Nine datasets have been collected to evaluate the performance and robustness of the cycle identification algorithm to different drivers and driving missions. A summary of the analysis of these datasets is shown in Table 1. The loader is operated by three different drivers with different experiences; one is experienced, one is intermediate, and one is a beginner. Each driver performs the following three driving missions; one including short loading cycles handling gravel, one including short loading cycles handling shot rock, and one including long loading cycles handling gravel. The drivers were instructed to drive the loader as they would during a normal working day including cleaning of the working site. In

Table 1: The number of loading cycles the algorithm identifies for different drivers with different missions. The values in parenthesis represent the number of cycles in each case.

Material	Operator	Short loading cycles	Long loading cycles	Cleaning cycles	Detection ratio
gravel	experienced	14 (15)	0 (0)	1 (0)	93%
	intermediate	15 (15)	0 (0)	0 (0)	100%
	beginner	11 (12)	0 (0)	1 (0)	92%
shot rock	experienced	17 (18)	0 (0)	1 (0)	94%
	intermediate	11 (11)	0 (0)	6 (6)	100%
	beginner	10 (10)	0 (0)	0 (0)	100%
gravel	experienced	0 (0)	9 (13)	4 (0)	69%
	intermediate	0 (0)	15 (15)	0 (0)	100%
	beginner	0 (0)	5 (6)	1 (0)	83%
gravel	–	18 (18)	7 (10)	10 (10)	92%
		96 (99)	36 (44)	24 (16)	93%

addition to these cases, data is collected from a mission combining long and short loading cycles and cleaning cycles, all handling gravel and using a different wheel loader. The numbers of cycles detected by the algorithm are presented in the table, along with the actual numbers of cycles that the drivers have completed in parenthesis. The last row in the table summaries all missions in the evaluation.

The dataset on the second last line in the table is also presented in Figure 11, and in this dataset all cleaning and short loading cycles are detected, but three of the long loading cycles are not detected. These cycles can be seen in Figure 11 starting at 500 seconds, 650 seconds, and 1100 seconds. The reason for not being classified as loading cycles is that after the second forward event in each cycle, see Figure 6, a dormant d and an activation event a are generated. The generated sequence of events will then neither match the automata given in Figure 6 nor the one in Figure 7. In the first nine datasets presented in Table 1, the eight missed loading cycles have been wrongly classified as cleaning cycles. In seven of these cases a loading event is not generated, and in one case the transition from backward to forward motion is not detected. The reason for not detecting the direction change is that v is calculated from $|\omega_{ds}|$, as mentioned in Section 3.2, using the selected gear direction. In this case the loader is operating at a slope and starts to reverse with a forward gear selected. In all, a total of 93% of the 159 cycles in the evaluation datasets are correctly detected and classified. This corroborates that the algorithm is robust to different drivers, driving missions and wheel loaders.

5.2 Parameter estimation

As described in Section 4.3, a parameter r that separates short and long loading cycles is computed, and an estimation of the bucket load is performed. A

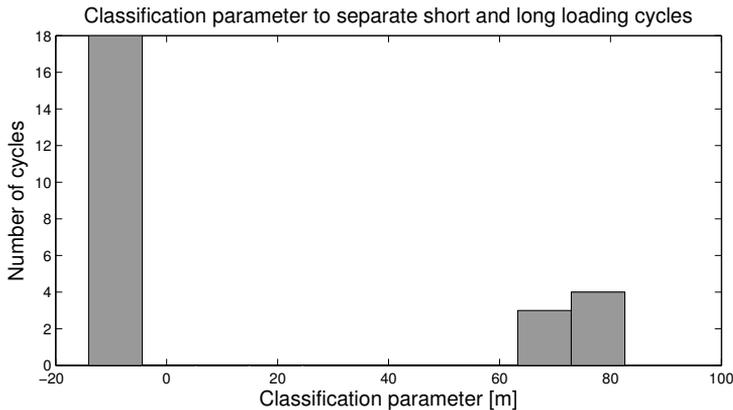


Figure 12: Histogram of r , as defined in (7), for the dataset presented in Figure 11.

histogram of the measure r that separate short and long loading cycles, as defined in (7), for the dataset presented in both Figure 11 and in the second last line in Table 1, is found in Figure 12. It can be seen that there is a clear distinction between the values from short loading cycles ($r < 0$ m), and long loading cycles ($r > 60$ m), which indicate that this classification parameter can be used to separate the two cycle types.

The mass estimator described in Section 4.3 gives an acceptable accuracy, even though an affine relation is used since the main objective was to investigate whether the event and cycle identification would aid the mass estimation. Figure 13 shows a comparison between the estimated and actual load mass normalized with the maximum load mass m_{max} of the loader. As stated in Section 2.2, the hydraulic pressure in the lift cylinder, p_θ , is not measured in production wheel loaders, but can be estimated from the pump pressure, p_{L_s} , when the driver lifts the bucket. In finding the time interval when p_{L_s} can be used as an approximation for p_θ , it is advantageous to use the results from the event and cycle identification algorithm. The load estimation is based on the averaged pressure from (11) and an affine relation between the bucket load and the pressure using (8). As discussed in Section 4.3, knowledge of the machine geometry and ground slope can be used to improve the accuracy of the function $f(p_\theta)$ in (8), and thereby increase the accuracy in the load estimation. Still, using the machine geometry would not reduce the importance of finding a good estimation of the pressure p_θ , and it is shown that the event and cycle identification algorithm gives valuable information to achieve this.

5.3 Summing up

Based on the cycle identification and the parameter estimation it is possible to summarize how the wheel loader is used for a long period of operation. In Figure 12 this is done by a histogram of the classification parameter stating

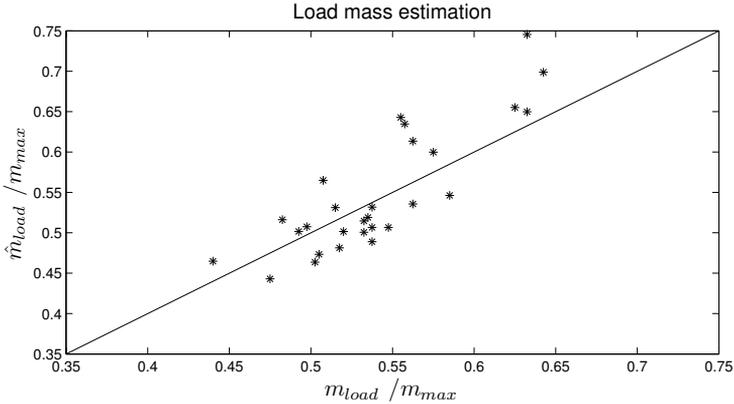


Figure 13: This figure shows the bucket load estimation, normalized with the maximum load of the wheel loader. The estimated normalized masses are given on the vertical axis and the sensor values on the horizontal axis.

whether the loading cycle is short or long, and in Figure 14 the time spent in each cycle type is shown. Both these two figures are based on the dataset presented in Figure 11. In the pie-chart there is a significant part of the time that is unspecified. The reason for this can be seen in Figure 11. The driver does not drive according to any of the modeled cycles in the beginning, at around 2000 seconds, and at the end of the driving mission. These parts are correctly classified as unspecified operation and represents more than half of the time spent in unspecified operations. Further, as stated earlier, there are three long loading cycles that are not classified in this dataset. Even though a significant part of the time is stated to be unspecified, 92% of the cycles in the dataset are correctly classified according to Table 1. This illustrates that in realistic operation, the proposed algorithm is successful in detecting and identifying the pre-defined cycles. Based on the table it is shown that the cycle identification algorithm is robust, since 148 of the 159 cycles in the evaluation datasets collected from different drivers, driving missions and wheel loaders are correctly classified.

Other values of interest to summarize over long time operation relates to fuel consumption and productivity, see Table 2 for an example. These values are based on the load mass estimation, that is shown to benefit in accuracy by using the event and cycle algorithm, and thereby avoid the use of additional sensors.

6 Conclusions

A framework for characterizing wheel loader operation has been developed. Two types of cycles, loading cycles and cleaning cycles, have been considered, but the framework is generic and it is possible to also include models for other cycles. It has been shown that the developed cycle identification algorithm is robust to

Table 2: The load handled by the loader per time unit and the load handled normalized with fuel consumption for the three drivers and the nine first datasets used in Table 1. The results are for the total of the three driving missions for each driver in Table 1.

Driver	Mass/time [ton/hr]	Mass/fuel [ton/l]
experienced	326	15.3
intermediate	269	14.3
beginner	138	10.2

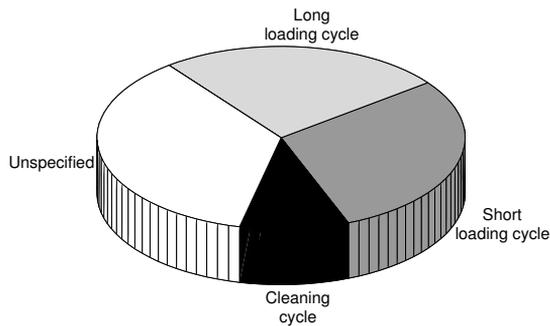


Figure 14: An example of a usage summary.

different drivers, machines, material handled, and the working site layout. Ten datasets were used in the evaluation, in which four different drivers perform sequences of different loading and cleaning cycles. In total the ten datasets contain 159 cycles, out of which the algorithm correctly detects and identifies 148 cycles (93%). This is a high detection rate, especially considering the low theoretical complexity of the algorithm and the diversity of the operation.

It is shown that a proposed classification parameter to separate short and long loading cycles works well, since there is a clear distinction in the value of the parameter in the two types of cycles. Further, it has been shown that the use of the cycle and event detection algorithm is beneficial in the estimation of the bucket load mass.

References

- [1] S. Park, A. Malikopoulos, M. Kokkolaras, and D. Jung. Thermal management system modeling and component sizing for heavy duty series hybrid electric vehicles. *International Journal of Heavy Vehicle Systems*, 18(3):272–287, 2011.
- [2] A. Rehnberg and L. Drugge. Ride comfort simulation of a wheel loader with suspended axles. *International Journal of Vehicle Systems Modelling*

- and Testing*, 3(3):168–188, 2008.
- [3] J. Engström and T. T. Victor. Real-time recognition of large-scale driving patterns. In *Proceedings of 2001 IEEE Intelligent Transportation Systems*, pages 1018–1023, 2001.
 - [4] L. Johannesson, S. Pettersson, and B. Egardt. Predictive energy management of a 4qt series-parallel hybrid electric bus. *Control Engineering Practice*, 17(12):1440–1453, 2009.
 - [5] C. C. Manzie, H. Watson, and S. Halgamuge. Fuel economy improvements for urban driving: hybrid vs. intelligent vehicles. *Transportation Research Part C: Emerging Technologies*, 15(1):1 – 16, 2007.
 - [6] C. Lin, S. Jeon, H. Peng, and J. Moo. Driving pattern recognition for control of hybrid electric trucks. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 42(1-2):41–58, 2004.
 - [7] S. Jeon, S. Jo, Y. Park, and J. Lee. Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition. *Journal of dynamic systems, measurement, and control*, 124(1):141–149, 2002.
 - [8] D. Kelley. *Automata and Formal Languages*. Prentice Hall, 1998.
 - [9] D. Mitrovic. Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6:198–205, 2005.
 - [10] L. Ljung. *System identification - theory for the user*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.
 - [11] R. Filla. Study of a method for assessing operability of working machines in physical and virtual testing. *International Journal of Vehicle Systems Modelling and Testing*, 7(3):209–234, 2011.
 - [12] H. Jiawei, C. Hong, X. Dong, and Y. Xifeng. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.
 - [13] K. Ohlsson-Öhman. Identifying operator usage of wheel loaders utilizing pattern recognition techniques. Master’s thesis, Linköping University, Linköping, 2012.
 - [14] H. Mannila, H. Toivonen, and I.A. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.
 - [15] G. Das, R. Fleischer, L. Gasieniec, D. Gunopulos, and J. K. Episode matching.
 - [16] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33:31–88, 2001.

- [17] P. Antoniou, J. Holub, C.S. Iliopoulos, B. Melichar, and P. Peterlongo. Finding common motifs with gaps using finite automata. In *11th International Conference on Implementation and Application of Automata*, pages 69–77, Taipei, Taiwan, 2006.
- [18] V. Nezhadali and L. Eriksson. Optimal control of wheel loader operation in the short loading cycle using two braking alternatives. In *IEEE VPPC 2013 - The 9th IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, 2013.

A Nomenclature

a	action event
a_θ, b_θ	parameters of the load mass model
b	backward event
d	dormant event
f	forward event
\mathcal{T}	a set of sampling times
l	bucket loading event
m_{load}	bucket load mass
m_{max}	maximum bucket load mass
p_{Ls}	pressure in load sensing hydraulic pump (Ls pressure)
p_θ	pressure in the lift cylinder of the bucket
q_i	state of an automaton
r	longitudinal distance with load
t	time
t_b	time at backward event
$t_{c,s}$	start time of a cycle
$t_{c,e}$	end time of a cycle
t_l	time at first loading event
t_u	time at bucket unloading event
u	bucket unloading event
v	wheel loader velocity
\hat{x}	estimated value of x
\dot{x}	time derivative of x
α, β, L	tuning parameters in the loading event detector
γ, δ	tuning parameters in the lift cylinder pressure estimator
ϵ	tuning parameter in the dormant event detector
θ	lift angle of the bucket
ξ	approximate angle at which load slides off
ϕ	tilt angle of the bucket
ψ	threshold for classifying short/long loading cycles
ω_{ds}	angular speed of the drive shaft

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

Tomas Nilsson, Anders Fröberg and Jan Åslund

*Vehicular Systems, Department of Electrical Engineering,
Linköping University, S-581 83 Linköping, Sweden.*

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.

[†]This is a formatted version of “Minimizing Fuel Use During Power Transients for Naturally Aspirated and Turbo Charged Diesel Engines” by Tomas Nilsson, Anders Fröberg and Jan Åslund, previously published as the technical report number “LiTH-R-3077”, which is based on a chapter from the Licentiate Thesis “Optimal Engine Operation in a Multi-Mode CVT Wheel Loader”, 2012, Tomas Nilsson. The formatting is restricted to changing the article into a single-column format, adjusting sizes of figures and tables, and adjusting the referencing style.

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.

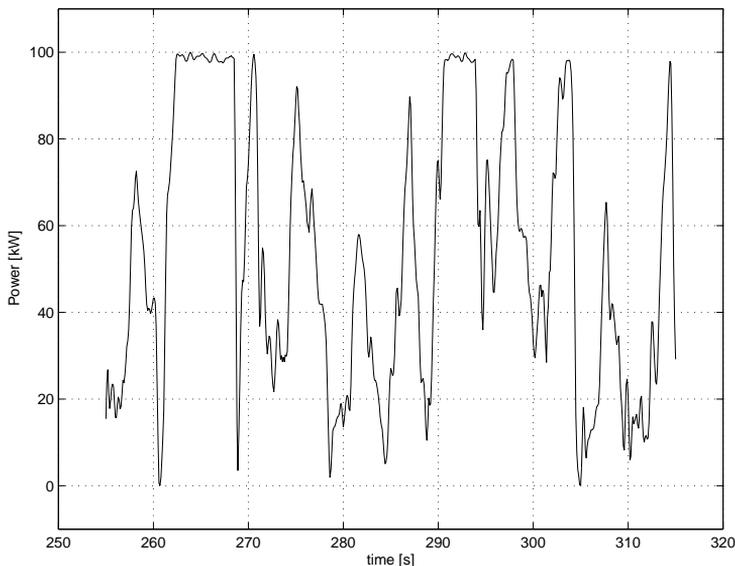


Figure 1: Power consumption of a wheel loader performing two short loading cycles

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.

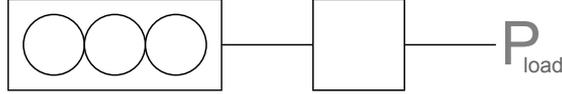


Figure 2: The system consists of an engine, the engine side of an infinitely variable transmission (e.g. an electric generator) and an output power.

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} .

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

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 $\eta_{e00}, \eta_{e01}, \eta_{e02}, \eta_{e10}, \eta_{e11}, \eta_{eL0}, \eta_{eL2}$ and define

$$A = \frac{q_{lhv} n_{cyl}}{2\pi n_r} \quad (2a)$$

$$\eta_e = \eta_{e0} - \eta_{e1} m_f \quad (2b)$$

$$\eta_{e0} = \eta_{e00} + \eta_{e01} \omega_e + \eta_{e02} \omega_e^2 \quad (2c)$$

$$\eta_{e1} = \eta_{e10} + \eta_{e11} \omega_e \quad (2d)$$

$$\eta_{eL} = \eta_{eL0} + \eta_{eL2} \omega_e^2 \quad (2e)$$

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 $T_t = 0$.

$$T_e = A \cdot \eta_e \cdot m_f - \eta_{eL} - T_t \quad (3)$$

The engine is also subject to the state and control restrictions

$$\begin{aligned} \omega_{e,min} &\leq \omega_e \\ 0 &\leq m_f \\ T_e &\leq T_{e,max}(\omega_e) \end{aligned} \quad (4)$$

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 efficiency parameters $\xi_{\tau 0}, \xi_{\tau 1}, \xi_{t1}, \xi_{t2}, \xi_{t3}, \eta_{t10}, \eta_{t11}, \eta_{t20}$ and η_{t21} and define

$$\tau_t = \xi_{\tau 0} + \xi_{\tau 1} \omega_e \quad (5a)$$

$$p_{t,set} = \xi_{t1} \omega_e + \xi_{t2} m_f + \xi_{t3} \quad (5b)$$

$$\eta_{t1} = \eta_{t10} + \eta_{t11} \omega_e \quad (5c)$$

$$\eta_{t2} = \eta_{t20} + \eta_{t21} \omega_e \quad (5d)$$

The pressure dynamics can then be described by

$$\frac{dp_t(t)}{dt} \cdot \tau_t(\omega_e) = p_{t,set}(\omega_e, m_f) - p_t(t) \quad (6)$$

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

$$T_t = \begin{cases} \eta_{t1}(\omega_e) \cdot p_{t,off}^2 + \eta_{t2}(\omega_e) \cdot p_{t,off} & \text{if } p_{t,off} > 0 \\ 0 & \text{if } p_{t,off} \leq 0 \end{cases} \quad (7)$$

2.3 Efficiency definitions

The quasi-static peak efficiency points Σ are defined as the (ω_e, T_e) that maximize (8a) as a function of P_{load} under the restrictions (4) and $\frac{d\omega_e}{dt} = \frac{dp_t}{dt} = 0$ as described by the Equations (8).

$$\eta_{e,static} = \frac{P_{load}}{P_{m_f}} = \frac{T_e \omega_e}{\omega_e A m_f} \quad (8a)$$

$$\omega_{e,\Sigma}(P_{load}) = \underset{\omega_e}{\operatorname{argmax}} \eta_{e,static}(P_{load}) \quad (8b)$$

$$m_{f,\Sigma}(P_{load}) = \underset{m_f}{\operatorname{argmax}} \eta_{e,static}(P_{load}, \omega_{e,\Sigma}) \quad (8c)$$

The Equations (8) 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 (9)

$$\min \int_0^T A\omega_e m_f dt \quad (9)$$

while fulfilling the engine dynamics Equation (1), the constraints (4) and, in case the engine is turbocharged, the turbo dynamics (5). 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.

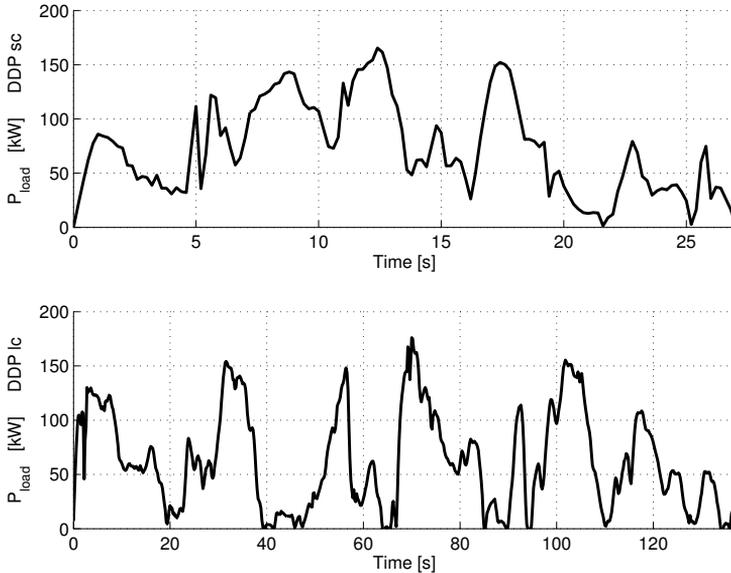


Figure 3: The output power trajectories in the load cases 'DDP *sc*' and 'DDP *lc*'.

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

$$\begin{aligned} \min_{u \in U} \{ & J_N(x(T)) + \int_0^T G(x, u, w) dt \} \\ \dot{x} = & F(x(t), u(t), w(t)) \\ x(0) = & x_0 \end{aligned} \quad (10)$$

along with possible 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

$$\begin{aligned} \min_{u \in U} \{ & J_N(x(T)) + \sum_{k=0}^{N-1} g_k(u_k, x_k, w_k) \} \\ x_{k+1} = & f(x_k, u_k, t), \quad k = 0, \dots, N-1 \end{aligned} \quad (11)$$

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

$$J_k(x_k) = \min_{u \in U} \{ g(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k)) \} \quad (12)$$

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 $\tilde{J}_{k+1}(x_{k+1}(x_k, u_k, w_k))$ 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

- 1: For $x_N \in X_N$, declare $J_N(x) = J_N$
- 2: **for** $k = N - 1, \dots, 1$ **do**
- 3: 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 $x_{k+1}(x_k, u, w_k)$
- 4: For each $x_k \in X_k$

$$J_k(x_k) = \min_{u \in U} (g(x_k, u, w_k) + \tilde{J}_{k+1}(x_{k+1}(x_k, u, w_k))) \quad (13)$$

with $\tilde{J}_{k+1}(x_{k+1})$ interpolated from $J_{k+1}(x_{k+1} \in X)$

- 5: **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

- 1: Select an initial state $x_0^* = x_0$
- 2: **for** $m = 1, \dots, N$ **do**
- 3: For x_{m-1}^* , simulate $\frac{dx}{dt}$ for t_{m-1} to t_m for all $u \in U$ to find $x_m(x_{m-1}^*, u)$
- 4: Select

$$u_{m-1}^* = \underset{u \in U}{\operatorname{argmin}} (g(x_{m-1}^*, u, w_{m-1}) dt + \dots + \tilde{J}_m(x_m(x_{m-1}^*, u, w_{m-1}))) \quad (14)$$

with $\tilde{J}_m(x_m)$ interpolated from $J_m(x_m \in X)$

- 5: $x_m^* = x_m(x_{m-1}^*, u_{m-1}^*, w_{m-1})$
- 6: **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

$$H = G(x(t), u(t), w(t)) + \lambda^T(t)F(x(t), u(t), w(t)) \quad (15)$$

in which G and F is the cost and dynamics functions from (10) 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

$$H(x^*, u^*, w, \lambda^*) \leq H(x^*, u, w, \lambda^*) \quad \forall u, t \in [t_0, T] \quad (16)$$

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 (10)

$$\frac{\partial H}{\partial u} = 0 \quad (17a)$$

$$\frac{\partial H}{\partial x} = -\dot{\lambda} \quad (17b)$$

$$\frac{\partial H}{\partial \lambda} = \dot{x} \quad (17c)$$

$$x(0) = x_0, \quad \lambda(T) = \frac{\partial J_N}{\partial x}(x^*(T)) \quad (17d)$$

must be fulfilled for x^*, u^* to be optimal. Condition (17c) is trivially fulfilled, as can be seen by differentiating (15). If the problem includes state or control constraints the Hamiltonian must be expanded, but the conditions (17) 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

$$J_N = \begin{cases} 0 & \text{for } x_N \geq \Omega \\ \infty & \text{else} \end{cases} \quad (18)$$

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.

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

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

$$H = A\omega_e m_f + \frac{\lambda_1}{I_e} (T_e - \frac{P_{load}}{\omega_e}) + \frac{\lambda_2}{\tau_t} (p_{t,set} - p_t) \quad (19)$$

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 (5). This gives the following conditions necessary for optimality

$$\frac{\partial H}{\partial m_f} = A\omega_e + \lambda_1 \frac{\partial}{\partial m_f} \frac{d\omega_e}{dt} + \lambda_2 \frac{\partial}{\partial m_f} \frac{dp_t}{dt} = 0 \quad (20a)$$

$$\frac{\partial H}{\partial \omega_e} = A m_f + \lambda_1 \frac{\partial}{\partial \omega_e} \frac{d\omega_e}{dt} + \lambda_2 \frac{\partial}{\partial \omega_e} \frac{dp_t}{dt} = -\frac{d\lambda_1}{dt} \quad (20b)$$

$$\frac{\partial H}{\partial p_t} = \lambda_1 \frac{\partial}{\partial p_t} \frac{d\omega_e}{dt} + \lambda_2 \frac{\partial}{\partial p_t} \frac{dp_t}{dt} = -\frac{d\lambda_2}{dt} \quad (20c)$$

The optimality conditions for the unconstrained NA-engine can be retrieved by using $\lambda_2 = 0$ and disregarding equation (20c).

5 Engine map and static optimal solution

The quasi-static optimal line Σ is defined in (8). 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 (8) 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.

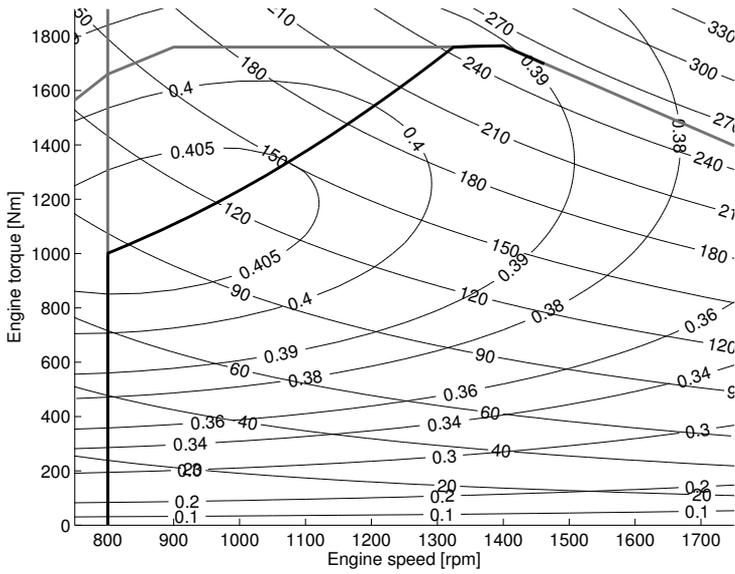


Figure 4: Engine map showing efficiency curves, output power lines with kW markings, state and control restrictions according to (4) and the quasi-static optimal line which for output powers below $\sim 85kW$ coincide with $\omega_{e,min}$ and above $\sim 240kW$ with $T_{e,max}$.

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.

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.

In Figure 7 the engine operation trajectories of the NA- and TC-engines are compared. Figure 3.7(a) shows the engine speed and torque during the first 10s of the slow pulse load case for the NA-engine and Figure 3.7(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

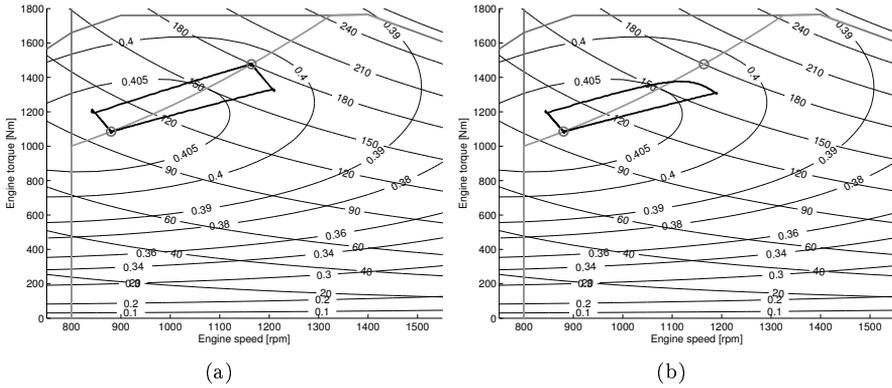


Figure 5: Engine map trajectories for the naturally aspirated engine in the slow (3.5(a)) and quick (3.5(b)) pulse load cases.

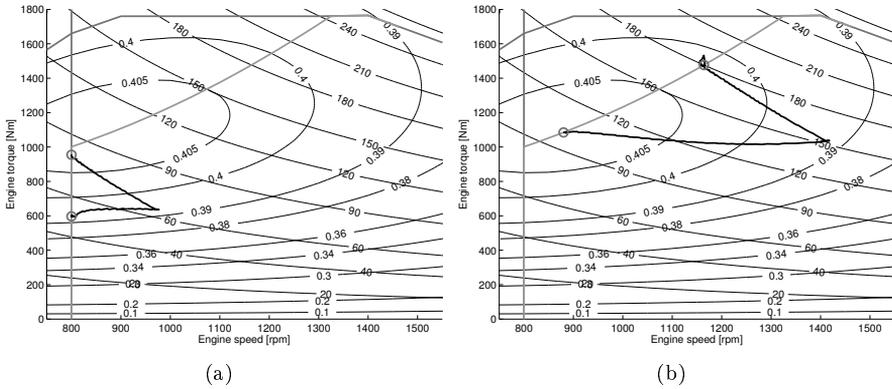
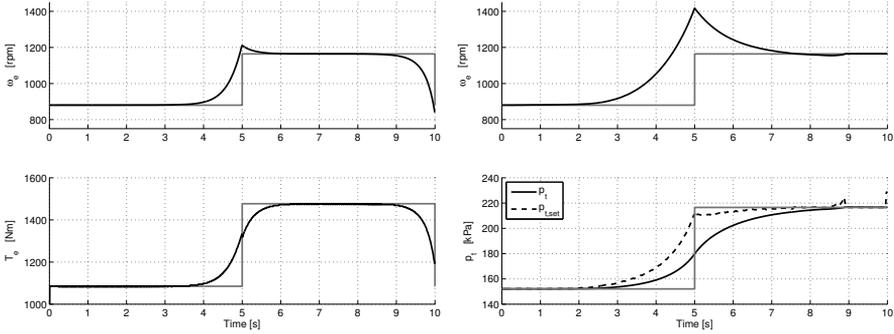


Figure 6: Engine map trajectories for the turbocharged engine in the low (3.6(a)) and high (3.6(b)) step load cases.

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 3.7(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.

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



(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.

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.

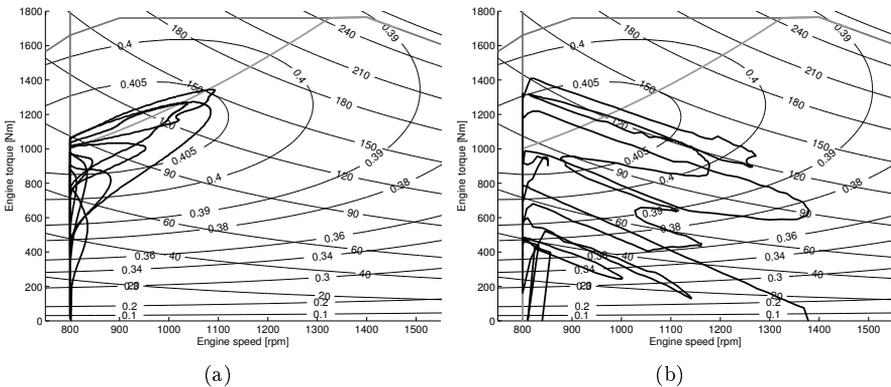


Figure 8: Engine map trajectories for the naturally aspirated (3.8(a)) and the turbocharged (3.8(b)) engine in the 'DDP sc ' cycle.

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 (4). A solution to the unconstrained problem (9) for a specific load $P_{load}(t)$ is optimal also for the constrained problem if and only if it does not violate the constraints (4). 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 3.5 (a), and the part used is presented again in Figure 3.9(a). Equation (20a) 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

$$\lambda_1 = \frac{\omega_e I_e}{2\eta_{e1} m_f - \eta_{e0}} \quad (21)$$

Figure 3.9(b) shows such a transformation of the map of Figure 3.9(a), including efficiency curves, output power lines with kW markings, the static optimal line Σ , the constraints (4) and the DP derived optimal operating point trajectory. In Figure 3.9(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 3.9(b) this translates to initial movement toward the lower right and a change of direction of motion at the instant of the step.

The dynamics of the adjoint variable $\lambda_1(t)$ is described by Equation (20b) (with $\lambda_2 = 0$). This equation can for the NA-engine be rewritten as

$$\dot{\lambda}_1 = -A m_f - \frac{\lambda_1}{I_e} \left(\frac{\partial T_e}{\partial \omega_e} + \frac{P_{load}}{\omega_e^2} \right) \quad (22)$$

in which

$$\frac{\partial T_e}{\partial \omega_e} = (\eta_{e01} + 2\eta_{e02}\omega_e - \eta_{e11}m_f)A m_f - 2\eta_{eL2}\omega_e \quad (23)$$

Since Equation (21) eliminates the only degree of freedom, all dynamics of the optimal solution is governed by Equations (1) (the engine speed) and (22) (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

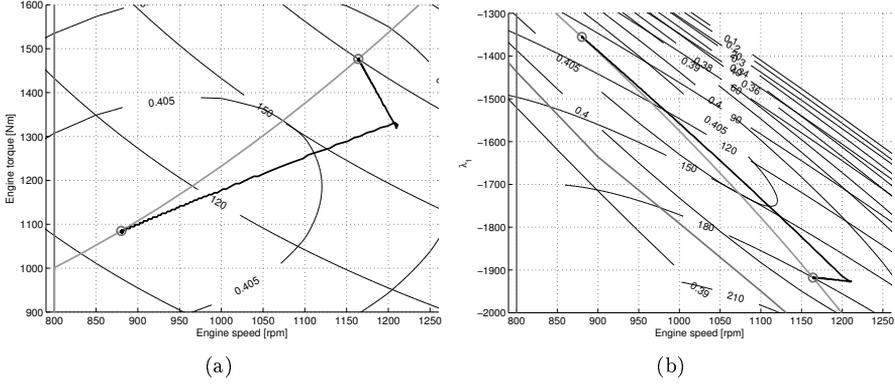


Figure 9: DP derived Optimal solution for the high step load case in ω_e-T_e (3.9(a)) and $\omega_e-\lambda_1$ (3.9(b)) engine maps.

planes for the system (1),(22) at the two output power levels of the high step load case are presented in Figure 10. The figure also shows the constraints (4), the static optimal line Σ and the DP-derived optimal trajectory, as shown in Figure 3.9(b).

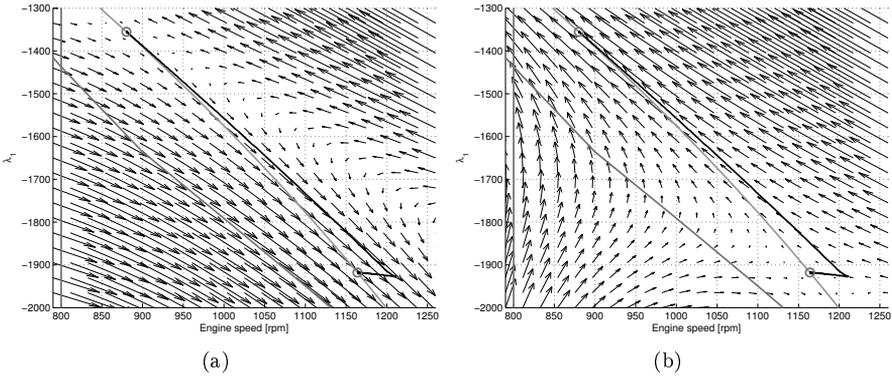


Figure 10: DP derived Optimal solution for the high step load case along with the 100kW (3.10(a)) and 180kW (3.10(b)) $\omega_e-\lambda_1$ phase planes.

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 3.10(a)), while the second segment, the approach of the second SOOP, is governed by the 180kW phase plane (Figure 3.10(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 (24). 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.

$$\frac{d}{dt}[\omega_e, \lambda_1]^T(P_{load}) \quad (24)$$

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 out-

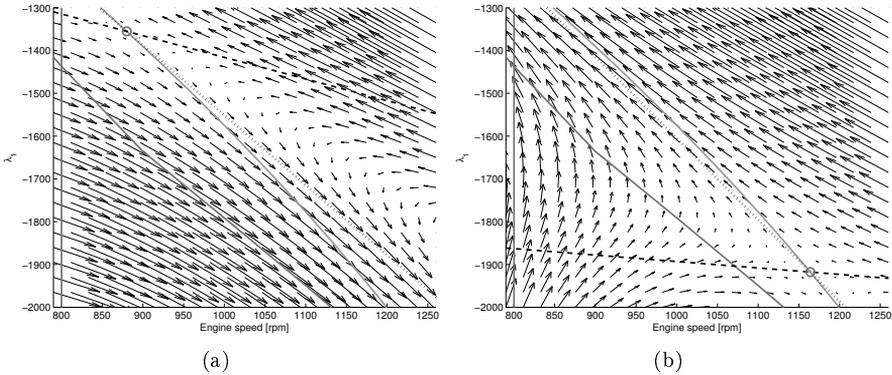


Figure 11: Phase planes along with stable (dashed) and unstable (dotted) eigenvectors of the Jacobian of the dynamic system (24) with $P_{load} = 100kW$ (3.11(a)) and $180kW$ (3.11(b)).

side 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.

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

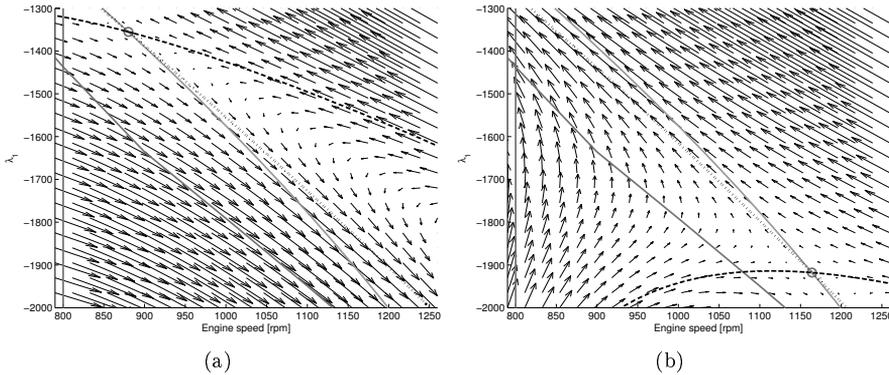


Figure 12: Simulation derived stable (dashed) and unstable (dotted) manifolds of the system (24) with $P_{load} = 100kW$ (3.12(a)) and $180kW$ (3.12(b)).

simulated paths. When the point of intersection is found the excess parts of the simulated paths are cropped 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 3.12(a) with Figure 3.12(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 (21).

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 3.14(a) by the gray line. In Figure 3.14(b) this solution (continuous) is translated to an ω_e, T_e map and compared to the solution derived with DP (dashed).

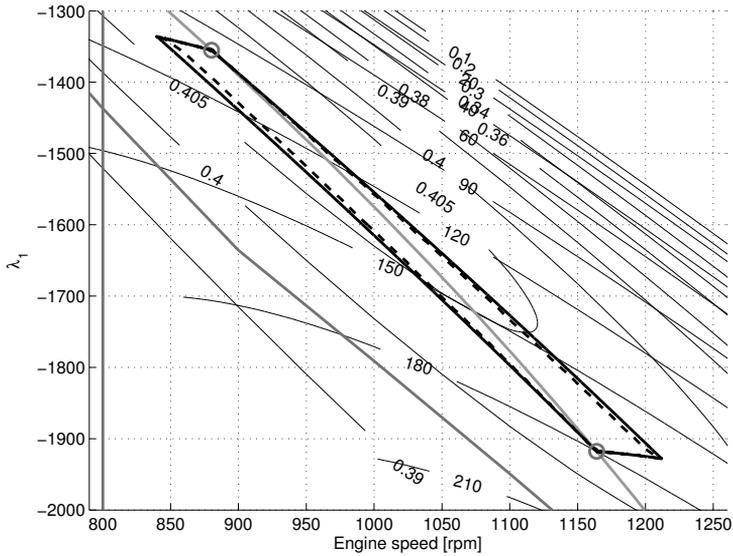


Figure 13: PMP (continuous) and DP (dashed) derived optimal solutions for the slow pulse load case.

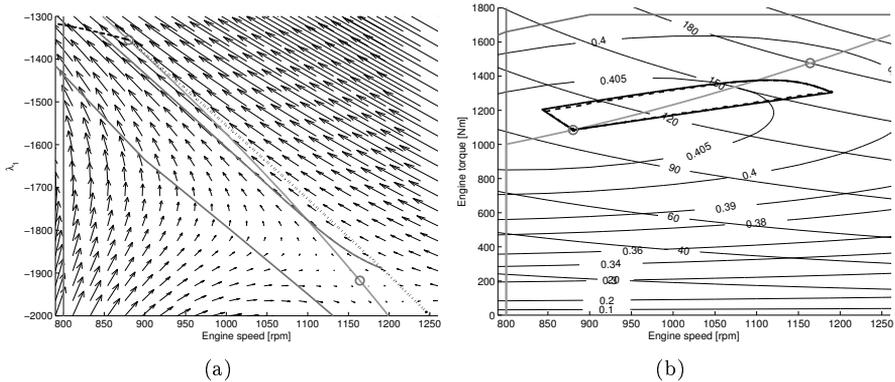


Figure 14: Illustration of the PMP-method for solving the quick pulse load case. Figure 3.14(a) shows the stable and unstable 100kW manifolds along with the 180kW phase plane and the 0.8s , 180kW transition path. In Figure 3.14(b) the PMP (continuous) and DP (dashed) derived optimal solutions are compared.

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 (25) and the static control relation (20a). The four dimensions of this problem means that phase planes can no longer be drawn

and the problem can therefore not be solved graphically.

$$\frac{d}{dt}[\omega_e, \lambda_1, p_t, \lambda_2](P_{load}) \quad (25)$$

The formulation of the torque loss T_t in Equation (6) may cause discontinuities in the optimality conditions (20) 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 (25) 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

$$\min_{s_1, t_1, s_2, t_2} \|X_1(P_1, t_1) - X_2(P_2, t_2)\|_2 \quad (26)$$

$$0 < [t_1, -t_2]^T, \quad 0 \leq [s_1, s_2]^T \leq 2\pi \quad (27)$$

in which

$$X_n = [\omega_e, \lambda_1, p_t, \lambda_2]^T(P_n, t_n), \quad n = 1, 2 \quad (28)$$

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

$$X_n(t_n = 0) = X_\Sigma(P_n) + \varepsilon(\sin(s_n)v_{n,1} + \cos(s_n)v_{n,2}), \quad n = 1, 2 \quad (29)$$

and the components of X_n in (26) 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 (26) 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.

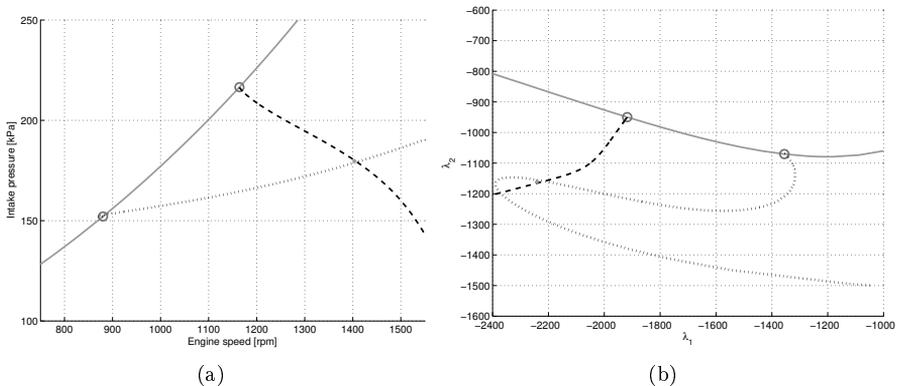


Figure 15: Intersecting stable (dashed) and unstable (dotted) trajectories for the high step load case in ω_e, p_t (Figure 3.15(a)) and λ_1, λ_2 (Figure 3.15(b)) maps. The minimum distance between the trajectories is marked with gray.

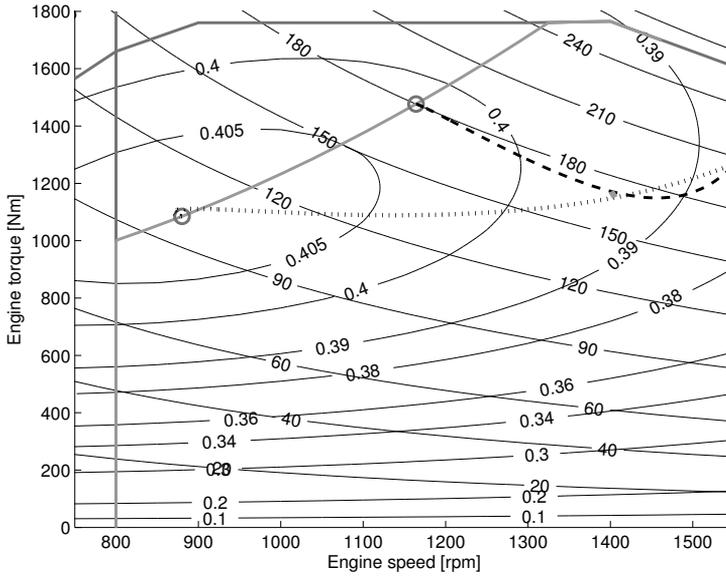


Figure 16: Intersecting stable (dashed) and unstable (dotted) trajectories for the high step case in an ω_e, T_e map. Note the minimum distance marker (gray).

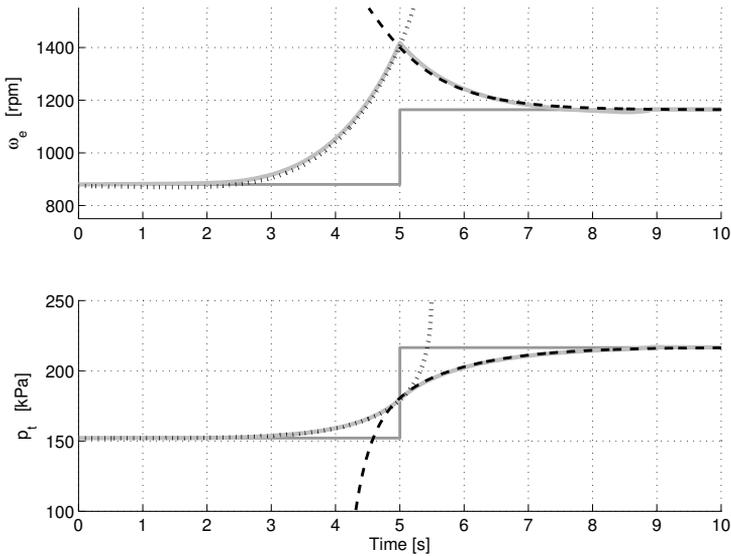


Figure 17: PMP-derived (dotted & dashed) compared to DP-derived (gray) solution for the high step load case. The dark gray lines indicate $X_\Sigma(P_{load}(t))$.

8 Suboptimal method development

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

$$T_e(t) = T_{e,\Sigma}(\omega_e(t)) \quad (30)$$

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 (30) 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 (31).

$$\omega_{e,k-1} = \omega_{e,k} - \left(\frac{T_{e,\Sigma}(\omega_{e,k})\omega_{e,k} - P_{load}}{\omega_{e,k}I_e} \right) dt \quad (31)$$

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	+0.086%	+0.099%	1 : 3340	1 : 3430

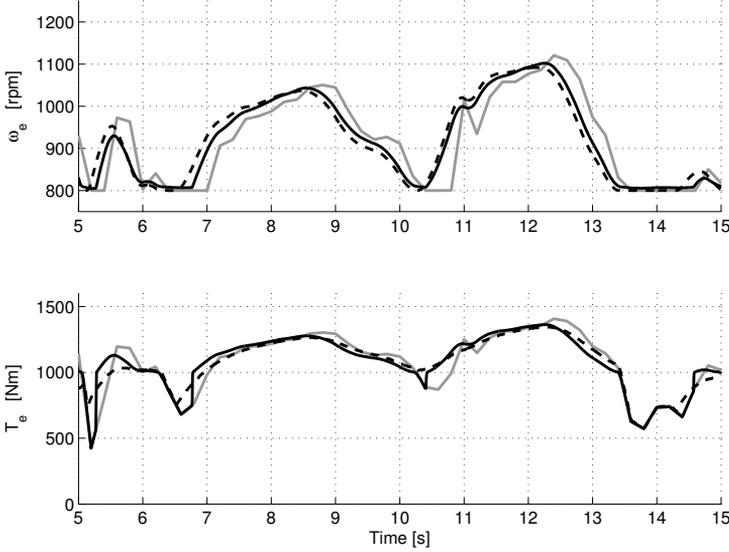


Figure 18: Engine speed and torque. Gray is static optimum ($\omega_{e,\Sigma}(P_{load}(t)), T_{e,\Sigma}(P_{load}(t))$), continuous is suboptimal and dashed is optimal.

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 (31). It is tempting to derive an $\omega_e(t), T_e(t)$ trajectory while disregarding $p_t(t)$, and then simulate (5) 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 (5)-(6) 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 (5) 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 from 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)-(5) for m_f so that $T_e = \frac{P_{load}}{\omega_e} - \frac{d\omega_e}{dt} I_e$.

If $I_e = \tau_t = 0$ 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 3.8(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

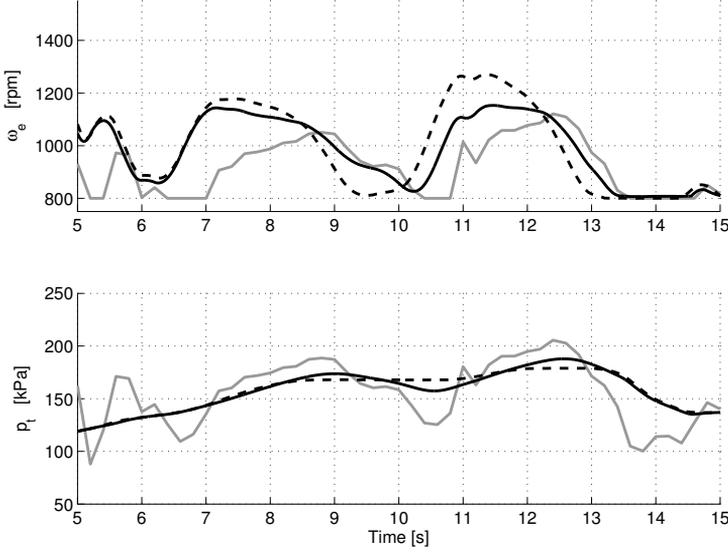


Figure 19: Engine speed and turbo pressure. Gray is static optimum $(\omega_{e,\Sigma}(P_{load}(t)), p_{t,\Sigma}(P_{load}(t)))$, continuous is suboptimal and dashed is optimal.

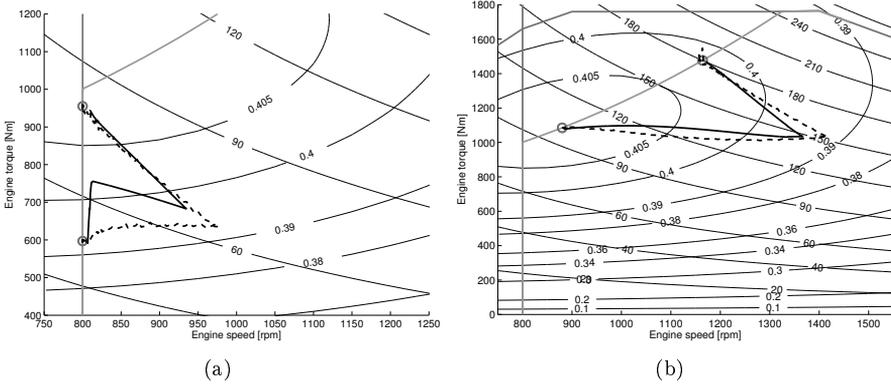


Figure 20: Suboptimal (continuous) and optimal (dashed) trajectories for the TC-engine in the low (3.20(a)) and high (3.20(b)) step load cases.

'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 the result presented in [31], in which the initial operating point movement is in a clockwise direction. The main operating point motions in [31] 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 (4) 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 (30) makes the system unstable.

References

- [1] F. Buscemi. Trends in automobile transmissions. *Gear Technology*, 23:24–26, 2006.
- [2] X.H. Zeng, Q.N. Wang, and D.F. Song. Direct statistical analyses of vehicle's fuel consumption based on driving cycles. *Journal of Hunan University Natural Sciences*, 37:35–40, 2010.
- [3] M.E. Starr, J.P. Buckingham, and C.C. Jackson. Development of transient test cycles for selected nonroad diesel engines. *American Society of Mechanical Engineers, ICE Division*, 32, 1999.

- [4] J. Lennevi. *Hydrostatic Transmission Control, Design Methodology for Vehicular Drivetrain Applications*. dissertation, Linköping University, 1995.
- [5] R. Ghabcheloo, M. Hyvönen, J. Uusitalo, O. Karhu, J. Jara, and K. Huh-tala. Autonomous motion control of a wheel loader. In *Proceedings of the ASME 2009 Dynamic Systems and Control Conference*, pages 1339–1346. ASME, 2009.
- [6] N. Koyachi and S. Sarata. Unmanned loading operation by autonomous wheel loader. In *ICCAS-SICE 2009*, pages 2221–2225. IEEE, 2009.
- [7] R. Zhang, D. Carter, and A. Alleyne. Multivariable control of an earth-moving vehicle powertrain, experimentally validated in an emulated working cycle. In *2003 ASME International Mechanical Engineering Congress*. ASME, 2003.
- [8] S. Grammatico, A. Balluchi, and E. Cosoli. A series-parallel hybrid electric powertrain for industrial vehicles. In *2010 IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, 2010.
- [9] S. Liu and B. Paden. A survey of today’s cvt controls. In *Proceedings of the 36th Conference on Decision and Control*, pages 4738–4743. IEEE, 1997.
- [10] N. Srivastava and I. Haque. A review on belt and chain continuously variable transmissions (cvt): dynamics and control. *Mechanism and Machine Theory*, 44:19–41, 2009.
- [11] A. Sciarretta and L. Guzzella. Control of hybrid electric vehicles. *Control Systems, IEEE*, 27:60–70, 2007.
- [12] J. Liu and H. Peng. Control optimization for a power-split hybrid vehicle. In *Proceedings of the 2006 American Control Conference*, pages 466–471. IEEE, 2006.
- [13] G. Paganelli, T.M. Guerra, S. Delprat, J. Santin, M. Delhom, and 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*, 214:705–717, 2000.
- [14] D. Ambuhl, O. Sundstrom, A. Sciarretta, and L. Guzzella. Explicit optimal control policy and its practical application for hybrid electric powertrains. *Control Engineering Practice*, 18:1429–1439, 2010.
- [15] R. Pfiffner. *Optimal Operation of CVT-Based Powertrains*. dissertation, ETH, Zurich, 2001.
- [16] P. Rutquist, C. Brietholtz, and T. Wik. An eigenvalue approach to infinite-horizon optimal control. In *Proceedings of the 16th IFAC World Congress*. IFAC, 2005.
- [17] B. Asadi and 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):707–714, 2011.
- [18] D. Mitrovic. Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6:198–205, 2005.

- [19] A. Pentland and L. Andrew. Modeling and prediction of human behavior. *Neural Computation*, 11:229–242, 1999.
- [20] T. Nilsson, A. Fröberg, and J. Åslund. Optimized engine transients. In *7th IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, 2011.
- [21] T. Nilsson, A. Fröberg, and J. Åslund. Optimal operation of a turbocharged diesel engine during transients. *SAE International Journal of Engines*, 5(2):571–578, 2012.
- [22] T. Nilsson. Optimal engine operation in a multi-mode CVT wheel loader. Technical report, Linköping University, 2012. LiU-TEK-LIC-2012:32, Thesis No. 1547.
- [23] G. Rizzoni, L. Guzzella, and B.M. Baumann. Unified modeling of hybrid electric vehicle drivetrains. *IEEE/ASME Transactions on Mechatronics*, 4:246–257, 1999.
- [24] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [25] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3 edition, 2005.
- [26] O. Sundström, D. Ambühl, and L. Guzzella. On implementation of dynamic programming for optimal control problems with final state constraints. *Oil & Gas Science and Technology - Rev. IFP*, 65(1):91–102, 2010.
- [27] L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems*. Springer Verlag, 2 edition, 2007.
- [28] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.
- [29] A.E. Bryson. *Applied Optimal Control; Optimization, Estimation and Control*. Taylor and Francis, 1975.
- [30] M. Dellnitz, O. Junge, and B. Thiere. The numerical detection of connecting orbits. *Discrete and Continuous Dynamical Systems - Series B*, 1:125–135, 2001.
- [31] R. Pfiffner and L. Guzzella. Optimal operation of cvt-based powertrains. *International Journal of Robust and Nonlinear Control*, 11(11):1003–1021, 2001.

Development of look-ahead controller concepts for a wheel loader application[†]

Tomas Nilsson, Anders Fröberg and Jan Åslund

*Vehicular Systems, Department of Electrical Engineering,
Linköping University, S-581 83 Linköping, Sweden.*

3

Abstract

This paper presents two conceptual methods, based on dynamic programming, for one step look-ahead control of a continuously variable transmission (CVT) in a wheel loader. The first method developed, designated Stochastic Dynamic Programming or SDP, uses a statistical load prediction and stochastic dynamic programming for minimizing fuel use. The second method developed, designated Free-Time Dynamic Programming or FTDP, has vehicle speed as a state and introduces a fixed 0.1s delay in the bucket controls in a combined minimization of fuel and time.

The methods are evaluated using a set of 34 measured loading cycles, used in a 'leave one out' maner. The evaluation shows that the SDP method requires about 1/10:th of the computational effort of FTDP and has a more transparent impact of differences in the cycle prediction. The FTDP method on the other hand shows a 10% lower fuel consumption, which is close to the actual optimum, at the same cycle times, and is able to complete a much larger part of the evaluation cycles.

[†]This is a formatted version of "Development of look-ahead controller concepts for a wheel loader application" by Tomas Nilsson, Anders Fröberg and Jan Åslund, Oil & Gas Science and Technology - Rev. IFP Energies nouvelles. ©IFP Energies nouvelles 2014. Reproduced with the permission of IFP Energies nouvelles. The original paper can be found at <http://ogst.ifpenergiesnouvelles.fr>, and by using the Digital Object Identifier (DOI): 10.2516/ogst/2014022. The formatting is restricted to changing the article into a single-column format, adjusting sizes of figures and tables, and adjusting the referencing style.

1 Introduction

1.1 Background

Wheel loader operation is often highly transient and contains episodes of low speed and high tractive effort, while the engine has to deliver power to both the transmission and the working hydraulics. The most common general transmission layout of heavy wheel loaders is presented in Figure 1. The engine is connected to the hydraulics through a variable displacement pump and to the drive shaft through a hydrodynamic torque converter and an automatic gearbox.

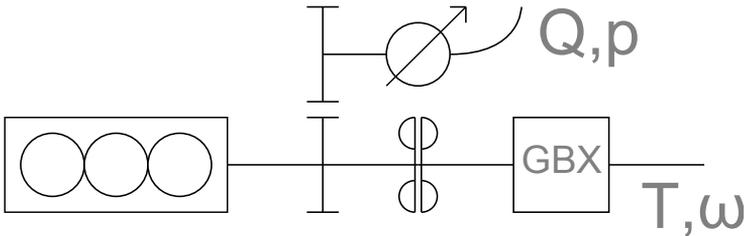


Figure 1: Reference vehicle drivetrain setup

In this setup the torque converter is a crucial component, since it provides some disconnection between the engine and vehicle speeds. This disconnection makes the system mechanically robust but the solution is also prone to high losses. High thrust is achieved by high torque converter slip, which produces losses. High hydraulic flow requires high engine speed, which also produces transmission torque which, if increased speed is not desired, is balanced by the brakes causing losses in both the torque converter and the brakes. This lack of efficiency is the reason for a desire to find other transmission concepts for wheel loaders.

1.2 On the choice of a hydraulic multi-mode CVT

Any alternative transmission has to enable increased efficiency at the typical operation conditions mentioned. The low speeds at which the machine often operate makes it impractical to use a stepped gearbox without a torque converter. One alternative is to consider infinitely variable transmissions, such as the diesel-electric used in [1] or the hydrostatic used in [2]. The drawback with this type of transmission is that the repeated power conversions reduce the peak efficiency. This is addressed by power-split constructions such as those described by [3] and by [4], in which some part of the power is mechanically transmitted. Multi-mode continuously variable transmissions (CVTs) are constructed so that several power-split layouts can be performed with the same device, thus enabling high efficiency at widely spaced gear ratios. In this paper, just as in [5], the transmission is based on a hydrostatic CVT since this solution has a favorable cost and torque rating.

1.3 CVT control in a wheel loader

The introduction of a CVT increases both the possibility for fuel saving and the risk of poor operability. The performance depends to a high degree on the implemented controller. Some work has been done on CVT control in wheel loader applications, e.g. in [6] and in [7]. The focus is often on actuator control though, and there is a lack of work on higher level control, including the choice of engine operating point. This choice is highly complicated by the operation often being extremely transient.

The most common operating pattern for wheel loaders is the short loading cycle. In this cycle the loader approaches a pile and fills the bucket, reverses, approaches a load receiver and empties the bucket, reverses and starts over. The operation is described in detail in [8] and in [9]. This easily described and highly repetitive operation may form the basis of a rough prediction of the future load. Because of the extremely transient operation, the benefits of utilizing the prediction in the controller can be expected to become high. Look-ahead control for on-road vehicles has been implemented e.g. in [10], [11] and [12]. In the wheel loader application, the potential benefit has been explored in [13], but so far there has been no implemented look-ahead controller for wheel loaders. The main difficulties, as compared to the on-road application, are the increase in system complexity and the uncertainties in the future load prediction. This paper introduces and evaluates two different conceptual look-ahead controller implementations for this system, both of which are based on dynamic programming.

1.4 Problem formulation

The goal of this paper is to develop and test, through simulations, conceptual dynamic programming based look-ahead controllers for use in a multi-mode CVT wheel loader. The controllers should be focused on the short loading cycle, and may therefore use future load predictions derived from data collected during measurements in a number of loading cycles. The aim should be to minimize, or at least to reduce, the fuel consumption without having a negative impact on drivability or performance of the machine.

2 Models

2.1 Machine operation

One of the most common operating patterns for wheel loaders is the short loading cycle (SLC), as described in [8] and in [9]. This cycle is also the basis for the prediction used in this work.

In the SLC definition used here, and referring to position designations in Figure 2, the cycle starts at position (2) and consists of four separate phases. In the first phase the machine drives forward to position (1), and during the final part of this phase the bucket is filled. The filling of the bucket often requires high tractive force combined with tilting and some lifting of the bucket. The

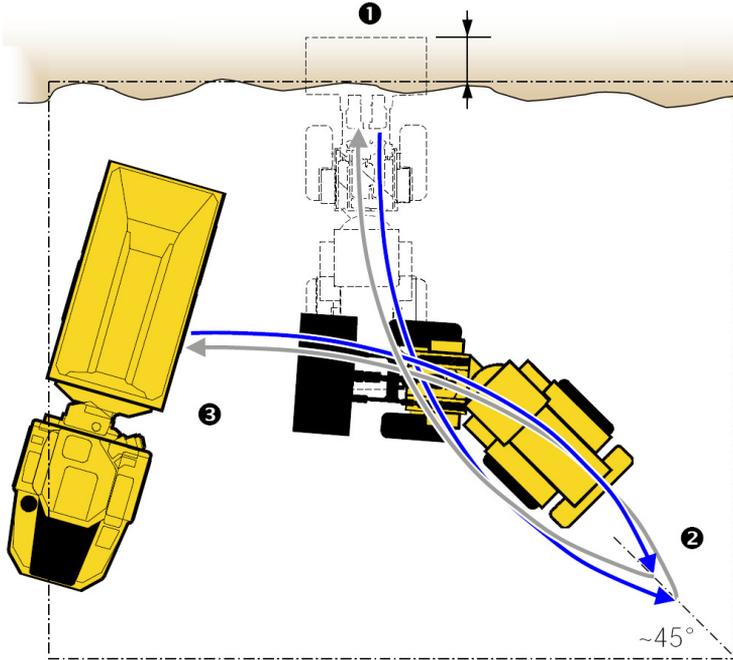


Figure 2: A view of a short loading cycle [8].

second phase is reversing back to position (2) and the third is forward driving to the load receiver at position (3). During these two phases, the bucket is raised, and at the end of the third phase it is emptied. The fourth and final phase is reversing back to position (2) while lowering the bucket. In a typical cycle the total duration is around 30s and the distances between the driving direction changes are around 10m.

In this paper a measurement sequence which includes 34 short loading cycles is used. The measurement setup is presented in Figure 3. The basic load components, related to the load components used in the system description in Figure 1, are vehicle speed v_w , tractive force F_w , hydraulic pressure p_H and hydraulic flow Q_H . The main difference from the description in Figure 1 is that F_w does not include inertia forces. These load components are derived as follows. The hydraulic pressure p_H is assumed to be equal to the measured hydraulic pump pressure p_{Ls} . The hydraulic flow Q_H is calculated from the volumes in the lift and tilt cylinders, which are calculated from the lift and tilt angles θ_1 and θ_2 . Lowering the bucket generally does not require pressurized hydraulic fluid, and this is therefore not supplied through the pump. The vehicle speed v_w is derived from the torque converter output speed ω_{ct} and the selected gear r_c , which include the selected driving direction. The tractive force F_w during the bucket filling is calculated from the torque converter output torque T_{ct} and the selected gear r_c . The torque converter output torque is calculated from the

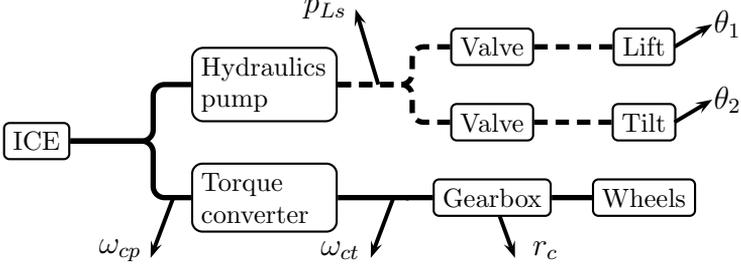


Figure 3: A view of the measurement setup indicating the signals available. Solid lines are mechanical connections and dashed lines are hydraulic connections. The system setup corresponds to that presented in Figure 1.

torque converter input and output speeds, ω_{cp} and ω_{ct} , according to

$$\nu_c = \frac{\omega_{ct}}{\omega_{cp}} \quad (1a)$$

$$T_{cp} = M_P(\nu_c) \left(\frac{\omega_{cp}}{\omega_{cp,ref}} \right)^2 \quad (1b)$$

$$T_{ct} = \mu(\nu_c) T_{cp} \quad (1c)$$

in which M_P and μ are scalable maps that have been measured at the reference speed $\omega_{cp,ref}$. The tractive force when not filling the bucket is modeled as a constant rolling resistance according to

$$F_w = \text{sign}(v_w) m g c_r \quad (2)$$

These basic load components are used in constructing the load cases $w(t)$ or $w(s)$, according to the requirements of each dynamic programming implementation. One of the measured SLCs, as described by the four presented load components, is displayed in Figure 4.

Due to adjustments made in the following load case creations, it is of interest to view the times and distances in the measured cycles. These are displayed in Figure 5. The average unadjusted cycle time is 26.5s and the average unadjusted distance driven is 35m.

2.2 Vehicle model and system layout

The vehicle is modeled as a mass m , for which the speed dynamics depend on the propulsive torque T_W , the brake torque T_b and the tractive force F_w . The factor r includes the final gear ratio and the wheel radius.

$$\frac{dv_w}{dt} \cdot m = r^{-1} T_W - r^{-1} T_b - F_w \quad (3)$$

The layout of the system is presented in Figure 6. The main components, which are described in the following sections, are the engine, the multi-mode CVT transmission and the variable displacement hydraulics pump.

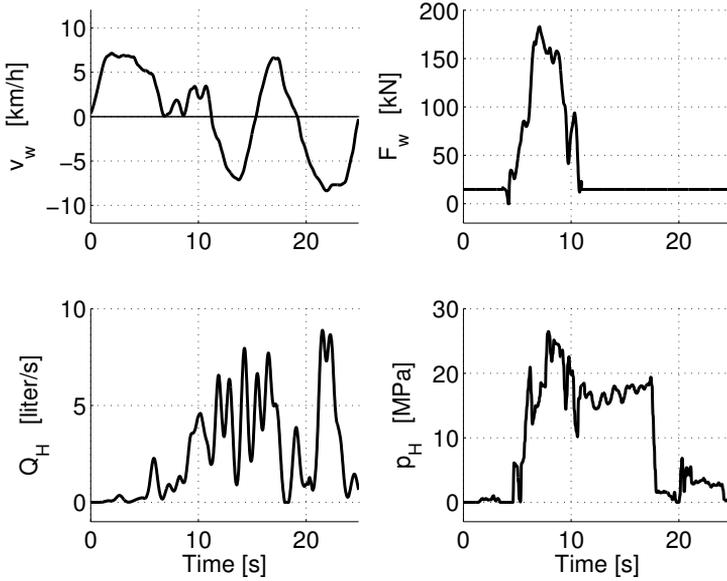


Figure 4: An example of a short loading cycle expressed in the four basic load components vehicle speed v_w , tractive force F_w , hydraulic flow Q_H and hydraulic pressure p_H .

2.3 Engine model

The engine is modeled as an inertia I_e which is affected by the engine torque T_e , the transmission torque T_T and the hydraulic pump torque T_H .

$$\frac{d\omega_e}{dt} \cdot I_e = T_e - T_T - T_H \quad (4)$$

The relation between fuel use and engine torque is described by a quadratic Willan's efficiency model, as presented in [14], expanded with a torque loss due to lack of intake manifold pressure

$$T_e = e(\omega_e, m_f) \cdot \frac{q_{lhv} n_{cyl}}{2\pi n_r} \cdot m_f - T_L(\omega_e) - T_{pt} \quad (5)$$

in which m_f is fuel mass per injection, ω_e is engine speed, e and T_L are efficiency functions, q_{lhv} , n_{cyl} and n_r are constants and T_{pt} is torque loss due to lack of air intake pressure $p_{off} = p_t - p_{set}(\omega_e, m_f)$. Here p_t is the actual pressure and p_{set} is a static setpoint map. The turbocharger speed dynamics is assumed to be a first order system. The dynamics model is expressed in the corresponding intake air pressure

$$\frac{dp_t}{dt} \cdot \tau(\omega_e) = -p_{off}(\omega_e, m_f) \quad (6)$$

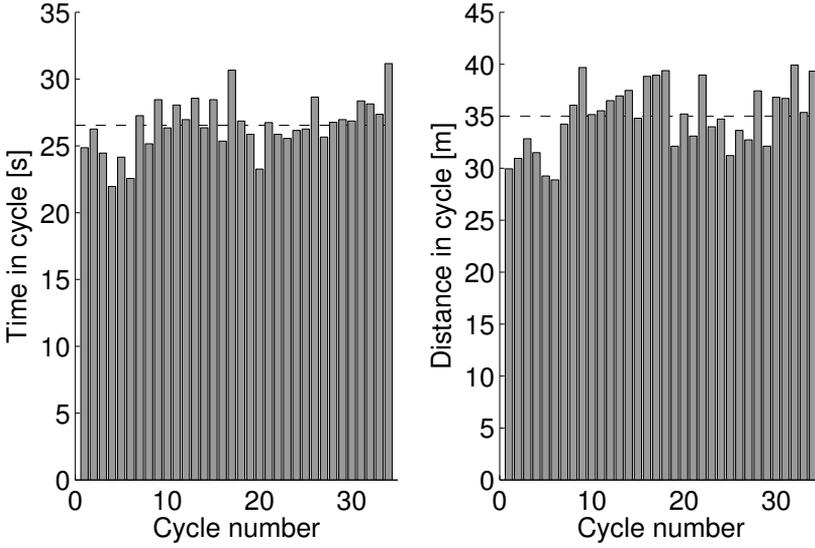


Figure 5: Overview of cycle durations expressed as time-in-cycle and as distance-in-cycle. The mean values are indicated by the dashed lines.

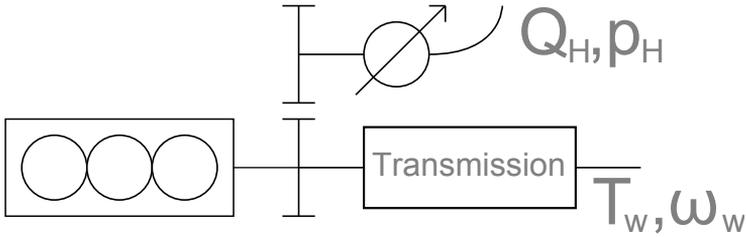


Figure 6: Overview of the layout of the system studied. The transmission is presented in detail in Figure 8.

and the torque loss from low pressure is described by

$$T_{pt} = \begin{cases} k_1(\omega_e) \cdot p_{off}^2 - k_2(\omega_e) \cdot p_{off} & \text{if } p_{off} < 0 \\ 0 & \text{if } p_{off} \geq 0 \end{cases} \quad (7)$$

The fuel per injection is related to the fuel flow according to

$$\frac{dM_f}{dt} = m_f \frac{n_{cyl}}{2\pi n_r} \omega_e \quad (8)$$

Figure 7 presents the efficiency map of the engine used. The gray lines indicate allowed operating region (minimum speed and maximum torque) and the black line indicates the static optimal operating points for each output power. The figure also shows efficiency levels and output power lines with kW markings.

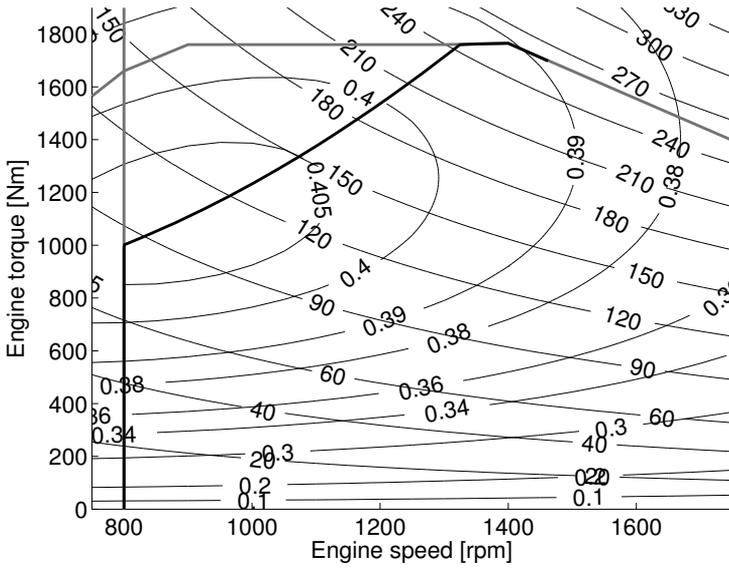


Figure 7: Engine map with static optimal operation line (black), speed and torque limits (thick gray), efficiency curves and output power lines with kW markings (thin gray).

2.4 Transmission model

The transmission used is the three mode ($m_T \in \pm[1, 2, 3]$) CVT described in the patent [15], and which has a structure similar to devices used in [5] and [16]. The layout is presented in Figure 8. In this figure the box to the left represents a Ravigneaux planetary gearset and the box to the right represents a regular planetary gearset. The driving direction and the transmission mode is selected by applying the corresponding clutches C_F or C_R , and C_1 , C_2 or C_3 . The CVT functionality is provided by the two hydraulic machines H_1 & H_2 , which together form a 'variator'. Changing gear ratio within a mode is done by altering the displacement ratio between the hydraulic machines. The engine side connection is marked with 'IN' and the wheel side connection is marked with 'OUT'. The transmission torque at the engine side is designated T_T and the torque at the wheel side is designated T_W .

The main source of losses in this concept is the variator, which is modeled according to Equations (9) and (10). This model is based on a model used in [6].

$$\psi_1 D_v \omega_1 \pm p_v (C_a + (\omega_1 + \omega_2) C_b) - \psi_2 D_v \omega_2 = C_v T \dot{p}_v \quad (9)$$

$$\psi_n D_v p_v - T_n \pm (C_c \omega_n + C_d p_v) = 0 \quad (10)$$

The index $n = 1, 2$ denotes the two machines, D_v is maximum displacement, $\psi_n \in (0, 1)$ is relative displacement, ω_n is axle speed, p_v is variator hydraulic pressure, T_n is torque and C_a, C_b, C_c and C_d are efficiency parameters. The signs in the equations depend on the power flow direction. Equation (9) describes

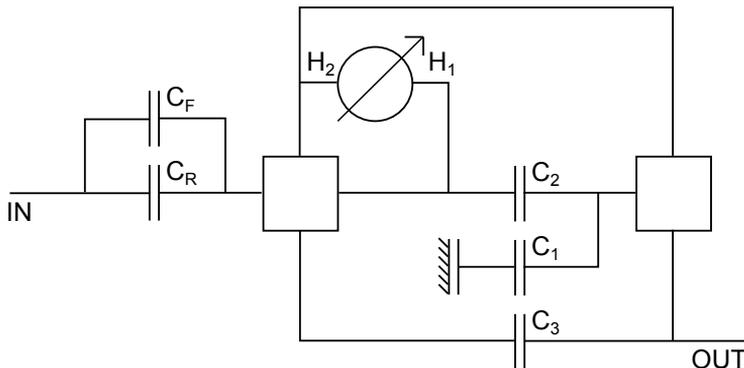


Figure 8: Layout of the multi-mode CVT. The box to the left is a Ravigneaux planetary gearset and the box to the right is a regular planetary gearset. H_1 and H_2 are hydraulic machines and the five C 's are clutches. Regular gears are not shown.

hydraulic fluid flow and Equation (10) describes torque at each machine. The variator is constructed so that $\psi_1 + \psi_2 = 1$. The variator pressure dynamics is assumed to be fast compared to other dynamics of the system, i.e. it is assumed that the time constant C_{vT} can be set to zero. Mode shifts are performed at the extremals of the variator displacement, and mode shifts at these points do not change the overall gear ratio for a lossless transmission. At mode shifts the speed differences over the involved clutches are close to zero, and the clutch losses are therefore small. This model can be summarized by the two functions

$$T_T(m_T, \psi_1, \omega_e, v_w) \quad (11)$$

$$T_W(m_T, \psi_1, \omega_e, v_w) \quad (12)$$

2.5 Hydraulics model

The bucket and boom are hydraulically driven. Pressure and flow of the hydraulic fluid are supplied by a hydraulic pump connected to the engine axle. This pump has variable displacement, so that the same pressure and flow can be provided at different engine speeds. Equations (13) and (14) describe the hydraulic pump

$$Q_H = \psi_H D_H \omega_e \quad (13)$$

$$Q_H p_H = \eta_H T_H \omega_e \quad (14)$$

D_H is maximum displacement, $\psi_H \in [0, 1]$ is relative displacement and $\eta_H(p_H, \psi_H)$ is pump efficiency. Lowering of the bucket does not require flow from the hydraulic pump.

3 Methods

3.1 Basic dynamic programming algorithm

Both of the control concepts to be presented are based on the dynamic programming recursion. This method description therefore starts with a recapitulation of this recursion, as used in the following methods. Denote the discretized flow variable $s \in s_k$ with $k = 0, \dots, N - 1$, states $x \in X$ and controls $u \in U$. The notation $x_k = x(s_k)$ is used. The optimization problem can be formulated, with E referring to the expected value if w_k is stochastic, as

$$\min_{u_k \in U} E \left\{ J_N(x_N) + \sum_{k=0}^{N-1} g_k(u_k, x_k, w_k) \right\}$$

$$x_{k+1} = f(x_k, u_k, t), \quad k = 0, \dots, N - 1 \quad (15)$$

along with equality and inequality constraints. According to [17] and [18] the dynamic programming recursion can, for this problem, be stated as

$$J_k(x_k) = \min_{u \in U} E \left\{ g(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k)) \right\} \quad (16)$$

$$J_N(x_N) = g_N(x_N) \quad (17)$$

This recursion is solved according to the following algorithm, expressed for a deterministic load w_k , as previously presented in [19].

- 1: For each $x_N \in X_N$, declare $J_N(x) = J_N$
- 2: **for** $k = N - 1, \dots, 1$ **do**
- 3: For each $x_k \in X_k$, simulate $\frac{dx}{dt}$ for s_k to s_{k+1} for all $u_k \in U$ to find $x_{k+1}(x_k, u_k, w_k)$
- 4: For each $x_k \in X_k$

$$J_k(x_k) = \min_{u_k \in U} (g(x_k, u_k, w_k) + \tilde{J}_{k+1}(x_{k+1}(x_k, u_k, w_k))) \quad (18)$$

with $\tilde{J}_{k+1}(x_{k+1})$ interpolated from $J_{k+1}(x_{k+1} \in X)$

- 5: **end for**

If the load is stochastic, step 3 is performed for each possible load combination $w_l \in W_k$, and Equation (18) is altered to

$$J_k(x_k) = \min_{u_k \in U} \sum_{w_l \in W_k} p(w_l) (g(x_k, u_k, w_l) + \tilde{J}_{k+1}(x_{k+1}(x_k, u_k, w_l))) \quad (19)$$

in which $p(w_l)$ is the probability of the load being w_l . This first part is used to establish a cost-to-go (CTG) map $J(x \in X, s)$. In the following part, this map is used for calculating the optimal trajectory $x^*(s), u^*(s)$.

- 1: Select an initial state $x_0^* = x_0$
- 2: **for** $m = 1, \dots, N$ **do**
- 3: For x_{m-1}^* , simulate $\frac{dx}{dt}$ for s_{m-1} to s_m for all $u \in U$ to find $x_m(x_{m-1}^*, u)$
- 4: Select

$$u_{m-1}^* = \underset{u \in U}{\operatorname{argmin}} (g(x_{m-1}^*, u, w_{m-1}) + \tilde{J}_m(x_m(x_{m-1}^*, u, w_{m-1}))) \quad (20)$$

in which $\tilde{J}_m(x_m)$ is interpolated from $J_m(x_m \in X)$

- 5: $x_m^* = x_m(x_{m-1}^*, u_{m-1}^*, w_{m-1})$
- 6: **end for**

3.2 Dynamic programming as a one step look-ahead controller

The second part of the DP algorithm presented in the previous section can be seen as a one step look-ahead simulation. In this case the load w_k , $k = 0, \dots, N - 1$, used in the second part is the actual load, which will differ from the load used in the cost-to-go map calculation, unless there is a perfect prediction of future loads. This type of control, assuming a perfect but limited horizon prediction, is used e.g. in [10] and in [20]. If there are differences in the loads in the two parts of the dynamic programming algorithm, the resulting state and control trajectories will in general not be optimal for the second load trajectory. It can however be expected that a well designed CTG-map will result in state and control trajectories with a low associated cost for a range of actual loads. In some controllers, such as the one presented in [21], a distance independent CTG-map can be created through assuming distance independent load probabilities. In the problem treated in this paper, a position dependent load prediction is available, but there are considerable uncertainties in this prediction. The problem is therefore translated to a problem of selecting states and control signals and constructing a load case for the CTG-map calculation, so that the look-ahead control in the second part gives low cost even when the load is altered. Two different concepts have been developed and these are presented in Sections 3.3 and 3.4.

This section discusses the implication of uncertainties, and the impact of disturbances, in each load component as compared to the values predicted in the CTG-map calculation. It is assumed that in the one step look-ahead simulation, the load components represent the desired trajectories derived from driver inputs and the resulting forces experienced by the machine. The components are vehicle speed v_w , longitudinal force F_w , hydraulic flow Q_H and hydraulic pressure p_H .

Component v_w : The load component v_w is part of Equations (3) (vehicle speed dynamics) and (11) and (12) (transmission input and output torque). Note that the vehicle speed dynamics limit the derivative of the possible disturbance. In Equation (3) the impact of changing v_w can be treated as an additional disturbance in F_w . In Equations (11) and (12), the CVT mode and

variator displacement ratio can be changed fast. Changes in v_w can therefore be transferred through T_w and T_T to the engine speed dynamics (1).

Component F_w : The load component F_w is part of Equation (3) (vehicle speed dynamics). This component includes longitudinal forces on the bucket, which can change rapidly, e.g. if the bucket hits a rock. According to the reasoning for the component v_w , a change in F_w can be transferred through T_w and T_T to the engine speed dynamics (1).

Component Q_H : The load component Q_H is part of Equations (13) (hydraulic flow) and (14) (hydraulic power). In the actual vehicle, the hydraulic flow is related to bucket lifting speed, so the bucket inertia should limit $\frac{dQ_H}{dt}$. This limitation however is lessened by the possibility of forces created through the vehicle pitch dynamics. Therefore, it is assumed that Q_H can change rapidly. Further, the desired hydraulic flow along with the maximum pump displacement $\psi_H D$ causes a lower limit for the engine speed $\omega_{e,H}$, according to Equation (13). It is not uncommon that $\omega_{e,H}(t_k) > \omega_{e,min}$, and during these instances the limit is often active.

Component p_H : The load component p_H is part of Equation (14) (hydraulic power). This component is related to vertical forces on the bucket, which can change rapidly, e.g. if the bucket hits a rock. Changes in this component are transferred through T_H to the engine speed dynamics (1).

The engine torque can be altered instantaneous, though the turbo speed may restrict the magnitude of the change. The component Q_H causes a limitation that is often active, and uncertainties in this load component is therefore the primary obstacle to using dynamic programming as a look-ahead controller. To recapitulate, the limit comes from the relation

$$Q_H = \frac{dV_H}{dt} = \frac{dV_H}{ds} v_s = \psi_H(t) D_H \omega_e(t) \quad (21)$$

which if $\frac{Q_H}{\omega_e}$ becomes high enough requires $\psi_H > 1$. Since this is not allowed, other solutions must be found. Since ψ_H is limited, the alternatives identified are to introduce margins through ω_e and v_w , allow for deviation from $V_H(s)$ or introduce a short horizon prediction. These three alternatives are discussed in the following part.

The inertias of the states ω_e and v_s can be seen as the cause of the problem. An instantaneous increase in Q_H would require an instant increase in ω_e or decrease in v_s , both of which are prevented by their inertias. The first alternative is therefore to keep ω_e , as a function of v_s , at such a level that ψ_H will never have to go above 1. Since the actual Q_H is not available a worst case scenario must be used in the CTG-map calculation. The drawback is that both the engine and the hydraulic pump are most efficient at low speeds, so using a preventive increase of the engine speed can be expected to increase fuel consumption. This approach is the motivation and foundation of the 'stochastic dynamic programming' method presented in Section 3.3.

In an actual vehicle, deviating from the desired bucket trajectory is a natural response to an unachievable desired trajectory. In the simulation however this approach becomes complicated by several factors. First, each of the measured cycles consists of a bucket trajectory along with corresponding forces. Deviat-

ing from the bucket trajectory would produce new forces, and calculating these would require a gravel pile model, which is not readily available. Second, allowing deviations in bucket height corresponds to introducing a freedom in $V_H(s)$, which would require at least one additional state in the system. This is highly undesired in dynamic programming. For these reasons, this approach is not studied further in this paper.

The availability of a short horizon prediction of future hydraulic flow might seem implausible. In an implementation though, the desired hydraulic flow would be an input from the driver. If a small delay is introduced between driver input and actual flow, this would be equivalent to a short horizon prediction of future hydraulic flow. If a constant time delay is used, no additional state is needed. This approach is the motivation and foundation of the 'free-time dynamic programming' method presented in Section 3.4.

A measurement sequence with 34 short driving cycles is available for the evaluation of the methods. In each evaluation, one cycle is used as the actual cycle in a simulation. In each case it is assumed that the other 33 cycles are available for the CTG-map creation. Further, in the second stage the present load is assumed to be known, so that in the simulation, at $s = s_k$, the load w_k is available.

3.3 Stochastic dynamic programming

The method presented here is an extension of an algorithm previously presented in [22] and in [23].

Concept description

This concept includes the prediction uncertainties in the load cases used in the CTG-map creation, by describing the load w_k as a Markov process. In this description there are at each stage some different alternatives for the load, along with a probability distribution. By assigning an infinite cost to states from which the vehicle cannot complete the cycle, and include a worst case scenario with a low probability, the cost-to-go map will correspond to a minimization of the cost under the condition that the vehicle always must be able to handle the worst case future load. This method is designated the Stochastic Dynamic Programming, or SDP, method.

Implementation

The problem is formulated as a minimization of expected total amount of fuel M_f required for performing a short loading cycle. This can be expressed as

$$\min E \{M_f(T)\} \quad (22)$$

and the cost function therefore becomes

$$g(x_k, u_k, w_k) = \sum_{w_l \in W_k} \left(p(w_l) \frac{dM_f}{dt} \right) \quad (23)$$

in which W_k is the set of possible loads w_k at $t = t_k$ and p is the probability of that load being w_l . The terminal cost is set to be $J_N = 0$ for all states x_N .

Since $\omega_e(\psi_1)$ is always invertible for this concept either ω_e or ψ , along with m_T , can be used as state. Since the speed will increase for one of the hydraulic machines when ψ_1 gets close to 0 or 1, the losses increase in these regions. Therefore it is desirable to have high state grid density near the extremes of ψ_1 , which implies using ψ_1 as state. The possibility of restrictions on $\frac{d\psi_1}{dt}$, especially during mode shifts, also points toward using ψ_1 as state. Since the dynamics are described in terms of ω_e this would imply the following computational scheme:

$$\psi_{1,k} \xrightarrow{W_k} \omega_{e,k} \xrightarrow{\frac{d\omega_e}{dt}} \omega_{e,k+1} \xrightarrow{W_{\kappa}} \psi_{1,k+1}$$

In the first and last steps the load is required, since $\omega_e(\psi_1)$ depends on the load. At the last step a choice has to be made whether to use $\kappa = k$ or $\kappa = k + 1$. Using $\kappa = k$ is equivalent to making a change of variables in Equation (1) from $\frac{d\omega_e}{dt}$ to $\frac{d\psi_1}{dt}$. This choice of κ does not guarantee continuity in ω_e , which makes it possible for the optimizer to draw a net power from the engine inertia. $\kappa = k + 1$ on the other hand guarantees continuous ω_e and works well for a deterministic load, but in the stochastic case this causes a quadratic increase in load combinations, since $\psi_{1,k+1}$ would have to be calculated for all combinations of W_k, W_{k+1} . This would cause an unacceptable increase in calculation time. This means that for SDP it is not practical to use ψ_1 as a state, and instead ω_e is used. $\omega_e(\psi, m_T)$ may only be non-invertible in small regions near $\psi_1 = \{0, 1\}$, so instead of using m_T as a state, the m_T which give highest efficiency is used in ambiguous cases.

The independent, or flow, variable in this calculation is the time t , the states are the engine speed ω_e and the turbo pressure p_t , and the sole control signal is the fuel mass per injection m_f , as summarized in Table 1. The same state and control signals are used in both the CTG-map calculation and the look-ahead control simulation.

Table 1: States and control signals in the SDP method.

Flow	States	Controls
t	ω_e, p_t	m_f

Load case creation for the SDP method

Using SDP in look-ahead control applications has been studied e.g. in [24] and [21]. In these papers the load has the Markov property and the probability distribution of the load is also independent of time. In the application at hand the load is modeled as a Markov process, but since the intention is to utilize the fact that the vehicle operates in a well known cycle, the probability distribution of the load does depend on the time, forming a probabilistic short loading cycle. As described in Section 2.1, the operation of a wheel loader can be described by the load components $\omega_w = v_w r^{-1}$, T_w , Q_H and p_H , which are also the components used here. The torque T_w can be calculated, using the measured vehicle speed, from Equation (3). Describing the vehicle speed ω_w as

a Markov process is deemed unrealistic, as discussed in Section 3.2, and this component is therefore regarded as deterministic. The load components are calculated from the set of measured loading cycles. First, the time scales in all cycles are adjusted so that the driving direction changes at the same instances in all cycles. The four driving phases are set to be 10s for the forward and loading phase, and 5s each for the other three phases. The vehicle speed v_w is adjusted so that the distances driven between each direction change agree with those specified in the FTDP method, for a fair comparison in the subsequent evaluation. All four load components are calculated for each cycle. The mean μ and standard deviation σ of each component over a set of cycles, as functions of time, are calculated. The load W_k for the CTG-map calculation consists of all load component combinations, according to Table 2, making a total of 36 possible loads at each instant t_k . This is repeated for all cycles in the measured sequence, producing 34 CTG-map load cases, each time excluding one of the basic loading cycles from the set of cycles used in the calculation of μ and σ .

Table 2: Load case components and corresponding probabilities for the SDP cost-to-go map calculation.

ω_w	T_w	Q_H	p_h
μ (1)	$\mu - \sigma$ (.25)	$\mu - \sigma$ (.25)	$\mu - \sigma$ (.25)
	μ (.5)	μ (.5)	μ (.5)
	$\mu + \sigma$ (.25)	$\mu + \sigma$ (.2)	$\mu + \sigma$ (.25)
		$\mu + 2\sigma$ (.05)	

The load case that was excluded in each CTG-map load case creation is later used as the load applied in the corresponding simulation, allowing for 34 method evaluations.

3.4 Free-Time Dynamic Programming

The CTG-map calculation in the method presented here is partly based on an algorithm previously presented in [25].

Concept

This method reduces the sensitivity to disturbances in Q_H by introducing a short horizon prediction of this load component, and to uncertainties in the prediction of F_w and p_H by introducing a freedom in time. The prediction of Q_H should prevent the vehicle from entering a situation in which the engine speed is too low to allow for the desired hydraulic flow. The freedom in time is introduced through a freedom in vehicle speed. This freedom allows for using the energy stored in the vehicle speed to compensate for temporary high F_w or p_H and for reducing the tractive and hydraulic power by slowing down the flow of time through reducing the vehicle speed. Since a freedom in time is introduced, the components of the load w are redefined as functions of the distances calculated from the vehicle speeds in the measured cycles. This method is designated the Free-Time Dynamic Programming, or FTDP, method.

Implementation

CTG-map calculation

Since a freedom in time is introduced, the problem is reformulated as a combined minimization of total amount of fuel M_f and time T for performing a short loading cycle. The factor β is introduced to weigh time to fuel in the minimization. This can be expressed as

$$\min \{M_f(T) + \beta T\} \quad (24)$$

and the cost function therefore becomes

$$g(x_k, u_k, w_k) = \frac{dM_f}{dt} + \beta \quad (25)$$

in which, introducing the vehicle speed $v_s = |v_w|$ and distance driven $s = \int v_s dt$, the time steps $\Delta t = \Delta s / v_{s,k}$ or, if $v_{s,k} \approx 0$, $\Delta t = 2\Delta s / (v_{s,k} + v_{s,k+1})$, are used. The terminal cost is set to be $J_N = 0$ for all states x_N .

By reformulating the cost function and system dynamics to depend on position rather than time, a freedom in time can be introduced without the need to have time as a state of the system. The dynamics for a state x is rewritten, using the chain rule, according to

$$\frac{dx}{dt} = \frac{dx}{ds} \frac{ds}{dt} = \frac{dx}{ds} v_s = f(x(s), u(s), w(s)) \Rightarrow \quad (26)$$

$$\frac{dx}{ds} = \frac{1}{v_s} f(x(s), u(s), w(s)) \quad (27)$$

During the general driving cycle, the vehicle changes driving direction several times. At these instances the vehicle speed v_s has to go to zero. The state derivatives will then, according to the formulation (27), not be well defined. For the vehicle speed dynamics this can be solved by changing the state from speed to kinetic energy according to the description in Section 3.5. Similar state changes would not solve the problem for the engine speed and turbo pressure dynamics though. Hence the approximation

$$\Delta s = \bar{v}_s \Delta t, \quad \bar{v}_s = \frac{v_{s,k} + v_{s,k+1}}{2} \quad (28)$$

is instead used when the initial vehicle speed is close to zero, just as in the cost function. In the engine dynamics this approximation is supplemented with a correction of T_T to assure that this approximation does not push the transmission efficiency to above 100%. When the approximation is active a constant transmission efficiency of $\alpha = 0.8$ is used. The reformulated minimization criterion becomes

$$\min \int_0^{S_f} \left(\frac{dM_f}{dt} + \beta \right) \frac{ds}{v_s} \quad (29)$$

The independent variable in the CTG-map calculation is the distance driven s and the states are the vehicle speed v_s ($= |v_w|$), the engine speed ω_e and the

turbo pressure p_t . The control signals are the fuel mass per injection m_f , the CVT-mode M_T , the variator displacement ratio ψ_1 and the brake torque T_b . The vehicle speed is forced to zero at the positions of the driving direction changes $s = s_m$ by assigning infinite cost to non-zero vehicle speeds at these instances $J(s_m, v_s > 0) = \infty$. For calculation effort reasons, zero speed is not allowed at any other instance. For the same reason, and since braking is a waste of energy and should be avoided, using non-zero brake torque is only considered if $T_b = 0$ gives infinite cost for all m_f, ψ_1 . The gain from the variator ratio ψ_1 to the torques T_T and T_W , according to the functions (11) and (12), is very high and a high density ψ_1 control signal grid must therefore be used. This would however have a severe effect on the calculation effort. For this reason, a ψ_1 with high grid density but a narrow range centered around $\psi_1(m_T, \omega_e, v_s)$ such that $T_T(m_T, \psi_1, \omega_e, v_s) = 0$, is used.

Look-ahead control simulation

In the one step look-ahead simulation, the time t is used as flow variable, and the time step corresponds to the hydraulic flow delay/short horizon prediction. In the evaluation, a 0.1s time step, and hence delay/prediction, is used. This way, an infinite cost can be assigned to controls which give $\psi_H > 1$ at t_{k+1} , and thus state - load combinations which would require $\psi_H > 1$ are avoided. This change of flow variable, from that used in the CTG-map calculation, means that the positions t_k will not correspond to positions in the grid s . The interpolations in the simulations must therefore be done also over the flow variable, which increases the dimension in the interpolation. This increases the computational load, but the most severe effect occurs at the driving direction changes.

The driving direction changes are included in the CTG-maps as infinite cost for all vehicle speeds $v_s > 0$ at the corresponding positions. Say that the vehicle speed must be zero at $s = s_m$. Interpolation will then render $\tilde{J}(s) = \infty$ for all $s_{m-1} < s < s_{m+1}$ except $s = s_m, v_s = 0$. The direction changes therefore need special treatment, both in approaching and in leaving these positions. The complete procedure of approaching and leaving a direction change position is illustrated by Figure 9.

Approaching a direction change is detected when $s_k < s_{m-1} - \kappa$ and $s_{m-1} - \kappa < s_{k+1}(u)$, with κ being a small value which acts as a minimum Δt for the next simulation step. When this detection occurs, Δt is adjusted for those u so that $s_{k+1}(u) = s_{m-1}$ and \tilde{J} is interpolated among $J(s_{m-1})$. In the next step those $\Delta t, u$ that give $s_{k+1} = s_m, v_{s,k+1} = 0$ are used and the \tilde{J} interpolation is performed among $J(s_m, v_s = 0)$. The vehicle has now reached the direction change position.

When the vehicle leaves the direction change position, that is, as long as $s_m < s(t_{k+1}) < s_{m+1}$, \tilde{J} is interpolated from the temporary CTG-map $[J(s_m, v_{s,1}), J(s_{m+1}, v_{s,n})]$, $n = 2, \dots, N$ in which N is the size of the v_s -grid.

Apart from the change of flow variable, the states and controls are the same in the look-ahead control simulation as in the CTG-map calculation. Also, the same calculation effort saving measures are taken for the control signals in the simulation. The flow, state and control signals in the two parts of the algorithm are summarized in Table 3.

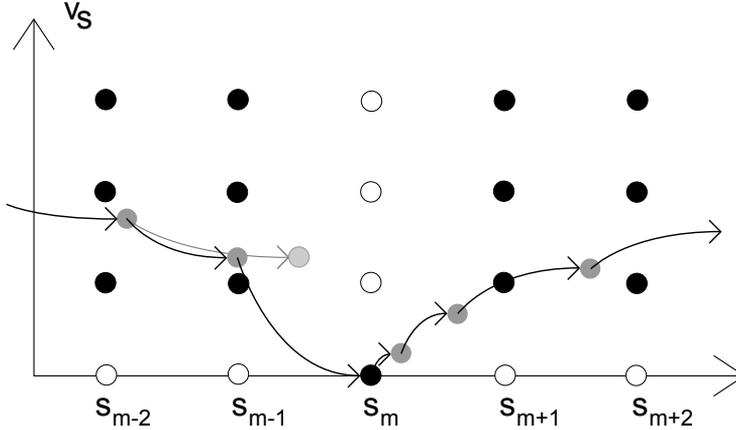


Figure 9: Illustration of the FTDP method simulation at a driving direction change. White nodes represent states with $J(x, s) = \infty$ and black represent states with $J(x, s) < \infty$. The arrows and gray nodes represent the simulated trajectory through the region. The light gray node is one for which Δt is reduced until $s(t_k + \Delta t) = s_{m-1}$.

Table 3: States and control signals in the FTDP method.

	Flow	States	Controls
CTG-calculation	s	v_s, ω_e, p_t	m_f, m_T, ψ_1, T_b
Simulation	t	v_s, ω_e, p_t	m_f, m_T, ψ_1, T_b

Load case creation for the FTDP method

The basic load components are functions of time, and these need to be reformulated into functions of distance driven. The distance driven must be monotonically increasing for these functions to be well defined. Since the machine drives in reverse direction part of the time, the velocity v_w is divided into speed $v_s = |v_w|$ and direction $d_s = \text{sign}v_w$ which enables the definition of the distance driven as $s = \int v_s dt$.

The positions at which the vehicle change the driving direction are specified by the driving cycle, which makes $d_s(s)$ a load component, while v_s is a state of the system. The driving direction changes must occur at the same positions in both the CTG-map calculation and the simulation. Therefore the distance scales are adjusted in all cycles so that each driving phase is $10m$. The tractive force $F_w(t)$ and the hydraulic pressure $p_H(t)$ can be directly shifted to depend on position rather than time $F_w(s)$, $p_H(s)$. The hydraulic flow Q_H , that is the hydraulic fluid volume per time, is transformed to a hydraulic volume per distance, or a hydraulic volume as function of distance driven $V_H(s)$.

$$V_H = \int_0^{s_N} \frac{Q_H}{v_s} ds \quad (30)$$

This hydraulic volume is the integrated flow of hydraulic fluid to the lift and tilt

cylinders as a function of distance driven while the force F_w and pressure p_H specify the wheel and bucket forces caused by this trajectory. This is repeated for each of the basic loading cycles, producing a total of 34 FTDP load cases. Each load case consists of the components direction of driving d_s , longitudinal force F_w , hydraulic volume V_H and hydraulic pressure p_H .

3.5 Simulations and energy balance

The choice of dynamic programming for optimization method, combined with the complexity of the system, makes efficient simulation of the functions $x_{k+1}(x_k, u_k, w_k)$ decisive. The Euler forward method is the simplest method for this simulation, and using this method is therefore desirable. Direct application of this method on the aforementioned states however does not preserve energy. In fact, using the engine speed dynamics as an example, the Euler step is

$$\omega_{e,k+1} = \omega_{e,k} + \Delta t \frac{T}{I_e} \quad (31)$$

and during this step the work performed by the torque is

$$W_1 = T\omega_{e,k}\Delta t \quad (32)$$

while the change in kinetic energy is

$$W_2 = \frac{I_e}{2}(\omega_{e,k+1}^2 - \omega_{e,k}^2) = T\omega_{e,k}\Delta t + \frac{(T\Delta t)^2}{2I_e} \quad (33)$$

and correspondingly for the vehicle speed dynamics, and also if formulated as functions of distance driven. There is obviously a discrepancy between the input and output energy. The optimization algorithm has been observed to exploit this discrepancy by fast switching between high positive and negative forces. Similar behavior has also been seen in e.g. [10] as oscillating controls in the solution. In the system at hand the gain from the control signal ψ_1 to the torques T_T and T_W is very strong, and the optimizer will therefore be highly inclined to using this shortcut by fast switching between high and low ψ_1 , especially in the FTDP method since the discrepancy can be exploited by moving kinetic energy between the engine and vehicle speeds with higher than 100% efficiency. In some cycles, the magnitude of the discrepancy became large enough for the vehicle to be propelled by this false input alone, requiring no fuel for completing an entire driving cycle. This problem can be prevented by using energy formulations for both vehicle and engine speed dynamics, according to

$$\frac{d}{dt} \frac{mv_s^2}{2} = v_s(r^{-1}T_W - r^{-1}T_b - F_w) \quad (34)$$

$$\frac{d}{dt} \frac{I_e\omega_e^2}{2} = \omega_e(T_e - T_T - T_H) \quad (35)$$

The Euler method simulation steps can be formulated as

$$v_{s,k+1} = \sqrt{v_{s,k}^2 + \frac{2v_s \Delta t}{m} \Sigma F} \quad (36)$$

$$\omega_{e,k+1} = \sqrt{\omega_{e,k}^2 + \frac{2\omega_e \Delta t}{I_e} \Sigma T} \quad (37)$$

and correspondingly when distance driven, s , is used as independent variable, which guarantee the balance of energy. This energy formulation is used in all simulation steps in both the CTG-map calculation and the look-ahead control simulation, in both of the methods.

4 Evaluation

Section 3 describes the one step look-ahead controller concepts and the corresponding load case creations. In this section, the controllers are evaluated by performing CTG-map calculations and subsequent simulations.

4.1 Stochastic Dynamic Programming

In each evaluation of the Stochastic Dynamic Programming (SDP) method, one loading cycle is used in the simulation and all other cycles from the measurement sequence are used in the CTG-map calculation. The measurement consists of 34 basic load cases, and the SDP method is therefore evaluated using 34 simulation loading cycles, each with a corresponding CTG-map calculated from the other 33 cycles, according to the description in Section 3.3.

Out of the 34 evaluations, 25 rendered a finite cost, which corresponds to 74% of the evaluations being successful. In three of the nine cases of infinite cost, this was caused by low engine speed compared to the minimum required by the hydraulic flow requirement. Most of the other six cases were caused by relatively high vehicle speed, related to the distance driven adjustment as described in Section 3.3. Figure 10 illustrates the fuel needed for performing each of the 34 cycles. The light gray bars represent the infinite-cost cycles, as the fuel used up until the encountering of the infinite cost, and the dashed line shows the average fuel use of 130g, in the cycles with a finite cost. The average optimal fuel use over the 34 cycles, that is the fuel required if the simulated cycle is also used for the corresponding CTG-map calculation, is 119g. The 4:th evaluation from the left is used as an example to illustrate the simulation results. Completing this particular cycle required 130g of fuel.

The CTG and simulation loading cycles for evaluation 4 are shown in Figure 11. The dotted lines are the load alternatives for the CTG-map calculation, as specified in Section 3.3, and the solid line is the load used in the simulation. This shows that there are significant differences in all components between the simulated cycle and the CTG cycle. In the vehicle speed, the positions of the driving direction changes coincide because of the design of the cycles. The load components in the simulated cycle are more transient than those in the CTG

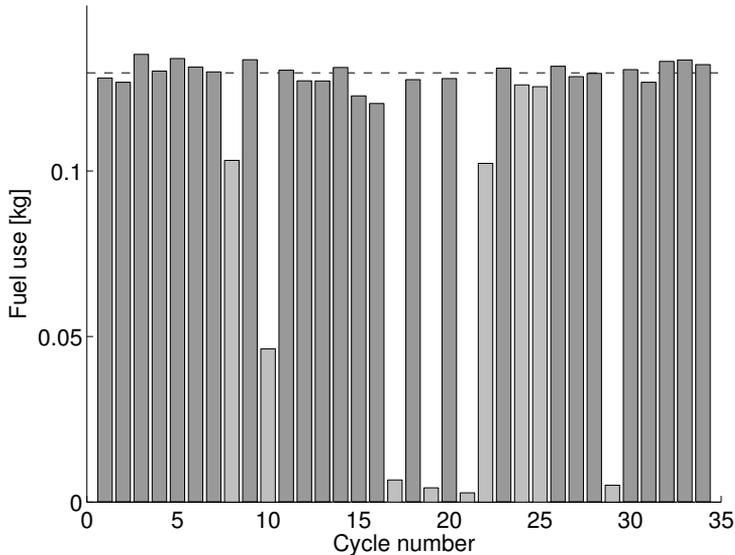


Figure 10: Fuel use in the SDP solutions. The light gray bars indicate cycles not completed due to infinite cost.

alternatives, since the CTG cycle has been constructed as an average over several cycles. Note that in this example, the hydraulic flow in the simulation is always lower than the highest alternative in the CTG cycle.

Figure 12 shows the state, ω_e and p_t , and control signal, m_f , trajectories from the simulation in evaluation 4. The ω_e state figure also shows the minimum engine speeds specified by the static limit (dotted line), and the hydraulic flow in the CTG load alternatives (four dotted curves) and in the simulation load (dashed curve). This shows that the engine speed is always higher than needed for the highest possible hydraulic flow in the CTG load. This keeps the engine speed higher than required by the actual desired hydraulic flow, which prevents infinite cost. In three of the simulation cases, this was not achieved, but the hydraulic flow in the simulated cycle was higher than the highest alternative in the CTG cycle at a time when the engine speed was close to this limit. This is illustrated by Figure 13 which shows the same signals as in Figure 12 but for the 8:th evaluation as referred to Figure 10, in which infinite cost is encountered at $t = 17s$, as indicated by a vertical gray line. In both figures, the intake pressure p_t is plotted along with the static pressure setpoint p_{set} (gray).

One of the main issues in using dynamic programming is the computational effort, especially when the number of states or control signals increases. Table 4 shows the experienced times needed for calculating the CTG-maps and for the look-ahead control simulations. The simulation times only include the cycles for which the cost is finite. All calculation times are highly dependent on the method implementation and state and control signal grid densities, and should therefore only be considered an indication and are only intended for comparison

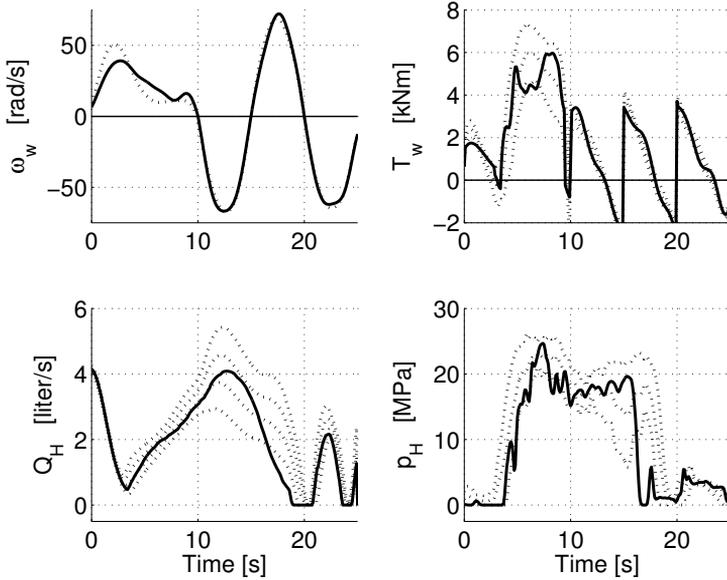


Figure 11: Illustration of the loads used for the 4:th evaluation. The dotted lines are the loads used in the CTG-map calculation (see also Table 2) and the solid lines are the cycle used in the simulation.

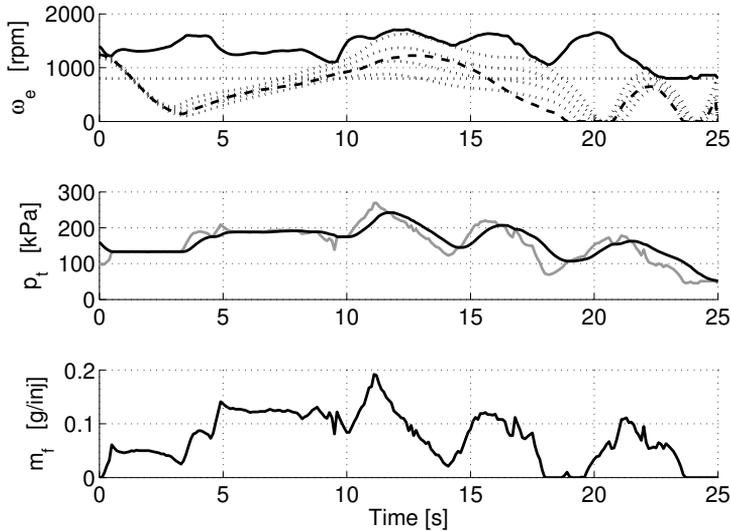


Figure 12: State and control signal trajectories in evaluation 4 (solid). The engine speed ω_e is plotted along with the minimum speeds given by the hydraulic flows in the CTG (dotted, see also Table 2) and simulation (dashed) cycles.

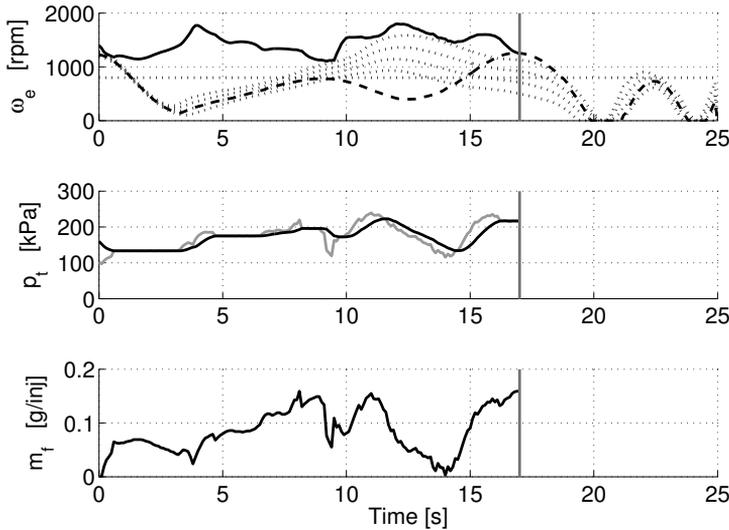


Figure 13: State and control signals in evaluation 8 (solid). The engine speed is plotted along with the minimum speeds given by the hydraulic flow in the CTG (dotted) and simulation (dashed) loads. Infinite cost is encountered at $t = 17s$

to the FTDP method. The discretizations have been made as sparse as possible without significantly affecting the optimization results.

Table 4: Experienced times for CTG-map calculation and look-ahead simulation, using the SDP method.

	$t_{min}[s]$	$t_{mean}[s]$	$t_{max}[s]$
CTG	877	974	1055
sim	0.30	0.30	0.32

4.2 Free-Time Dynamic Programming

In the Free-Time Dynamic Programming (FTDP) method, the creation of a load case for the CTG-map calculation only requires a single basic load case. An FTDP load case is therefore created from each of the 34 basic load cases, according to the description in Section 3.4. In the evaluation, the CTG-map is calculated using one FTDP load case and in the look-ahead control simulation any other FTDP load case can be used. The dataset contains a total of 34 cycles, making a total of 1122 combinations evaluated. The time to fuel weighting parameter is selected as $\beta = 0.5g/s$, since this gives cycle times similar to the 25s specified for the SDP method.

Figure 14 summarizes the result of these simulations. The gray markings indicate combinations that rendered a finite cost, while the black markings indicate combinations that rendered an infinite cost. In total, 1116 combinations

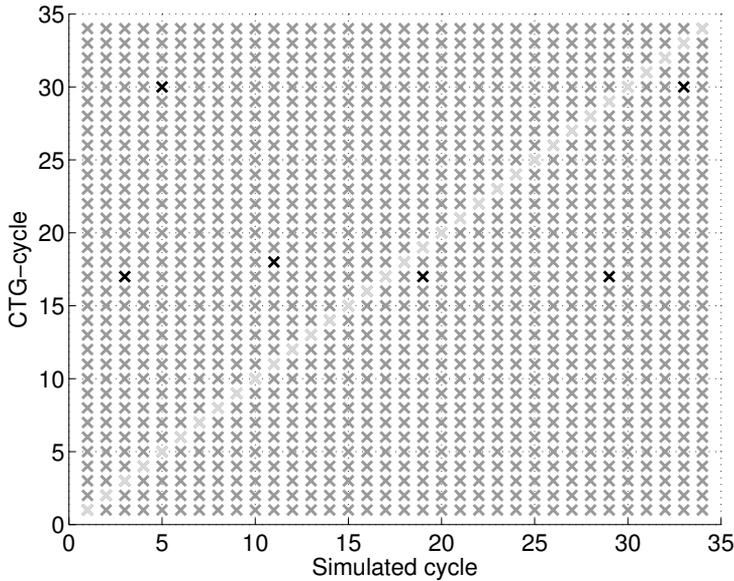


Figure 14: Illustration of success of the simulations in FTDP with $\beta = 0.5g/s$. The light gray markers indicate cycle combinations with perfect prediction, dark gray markers indicate successful simulations and black markers indicate cycle combinations which render infinite cost.

were successful while 6 rendered infinite cost, which translates to success in 99.5% of the combinations. Figure 15 shows the same result, but for $\beta = 10g/s$. In this case, 96.1% of the combinations were successful. It is clear though that some cycles were less suited for use in the CTG-map calculation. The most prominent of these are cycles 23 and 31. Disregarding these gives a total of 98.8% successful combinations.

Figure 16 shows the fuel and time required for completing each of the 1116 successfully simulated cycles. The average fuel use is $116g$ and the average time use is $24.7s$. The average optimal fuel use over the 34 cycles, which corresponds to the diagonal of Figure 14, is $115g$ and the corresponding average time use is $24.3s$. In the following part, the combination of the 4:th cycle for the CTG-map calculation and the 12:th cycle for the simulation, as referred to Figure 14, is used as an example to illustrate the simulation results. This combination will be referred to as evaluation 4-12. Completing this particular combination require $24.6s$ and $115g$ of fuel.

The CTG and simulation loadcases for this cycle are shown in Figure 17. The dotted lines are the CTG load, as specified in Section 3.4, and the solid lines are the load used in the subsequent simulation. The positions of the driving direction changes are the same in the two cycles, as specified in the design of the cycles. The other components are similar in appearance, though there are significant differences in amplitudes, durations and timing.

Figure 18 shows the state, v_s , ω_e and p_t , and time step Δt trajectories from

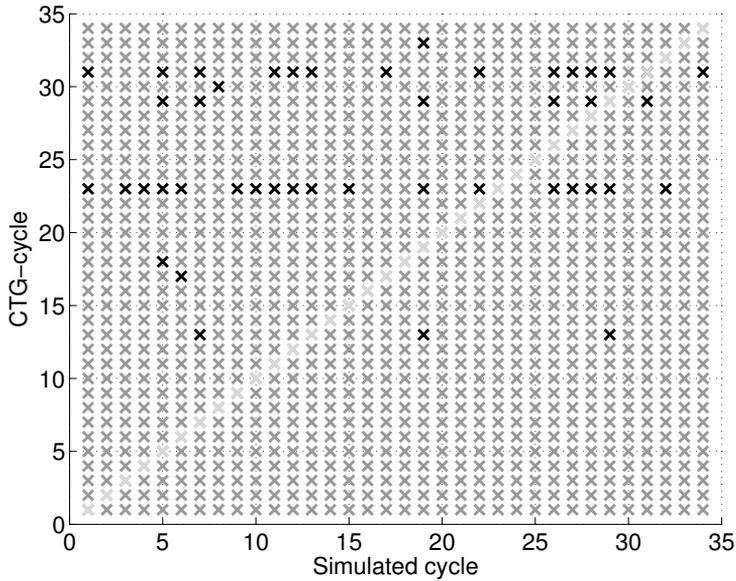


Figure 15: Illustration of success of the simulations in FTDP when time is prioritized by using $\beta = 10g/s$. See Figure 14 for explanation.

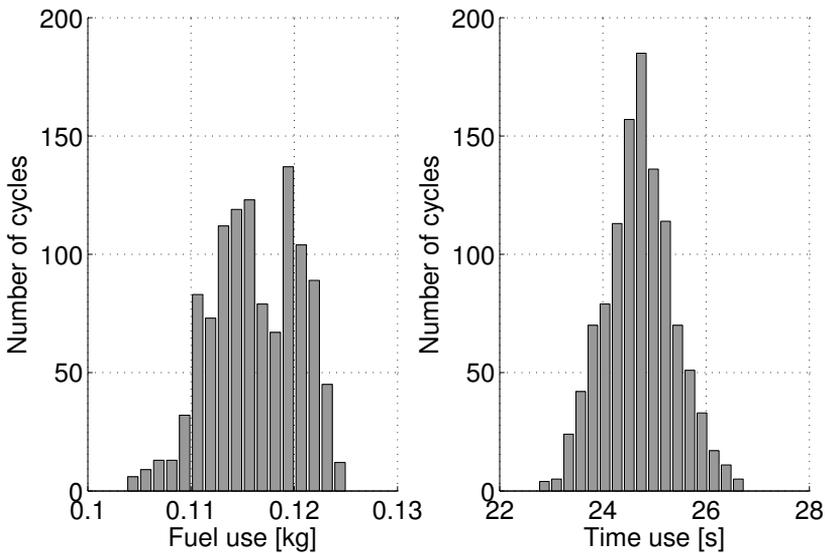


Figure 16: Summary of fuel and time use in the cycle combinations evaluated. The combinations with perfect prediction and those which rendered infinite cost are excluded.

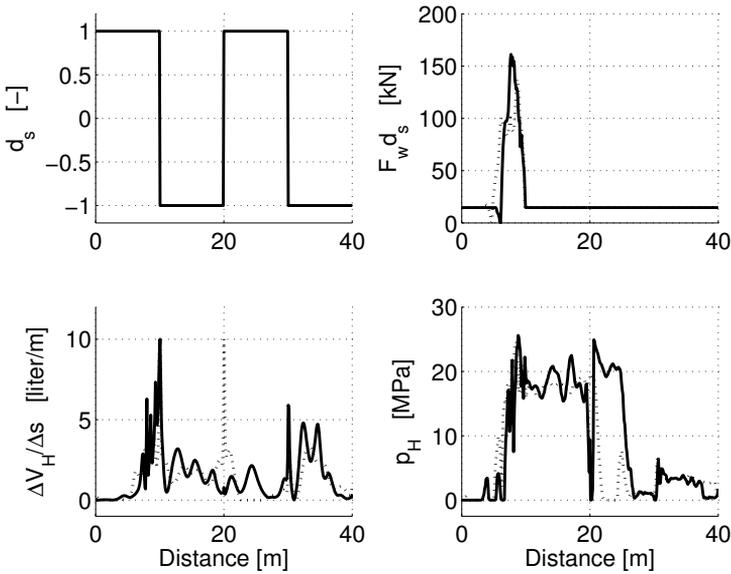


Figure 17: Illustration of the loads used in evaluation 4-12. The dotted lines are cycle 4, which is used for the CTG-map calculation, and the solid lines are cycle 12, which is used in the look-ahead control simulation.

evaluation 4-12. The ω_e figure also shows the minimum engine speed specified by the static limit (dotted line), and the hydraulic flow in the simulation load (dashed curve). This shows that the vehicle is able to keep the engine speed higher than required by the simulation hydraulic flow, which prevents infinite cost without maintaining a large ω_e margin. The intake pressure p_t is plotted along with the static pressure setpoint p_{set} (gray). The Δt figure clearly shows the adjustments of Δt made near the driving direction changes.

Figure 19 shows the control, m_f , m_T , ψ_1 and T_b , signal trajectories from evaluation 4-12. The m_f figure shows that the highest fuel flow is experienced during bucket filling. The m_T figure shows that around half the time is spent in CVT mode 1 and half of the time in CVT mode 2, while the ψ_1 figure shows that at mode changes, the variator displacement ratio is near its maximum or minimum, as required by the transmission model. The T_b figure shows that in this cycle, the vehicle never uses the brakes.

Figure 20 shows the state and control signals from an unsuccessful $\beta = 10g/s$ example. The example is the result from the combination of using the 23:rd cycle for the CTG-map calculation and the 12:th cycle in the subsequent simulation, as referred to Figure 15. Figure 21 shows the first part of the load case combination from this example, with the instant of the infinite cost marked. Typical for the cycles which produced CTG-maps which commonly rendered infinite cost in the subsequent simulations is that the increase in tractive force F_w related to the filling of the bucket is comparatively late and steep.

Table 5 shows seven examples of time and fuel use in evaluation 4-12, per-

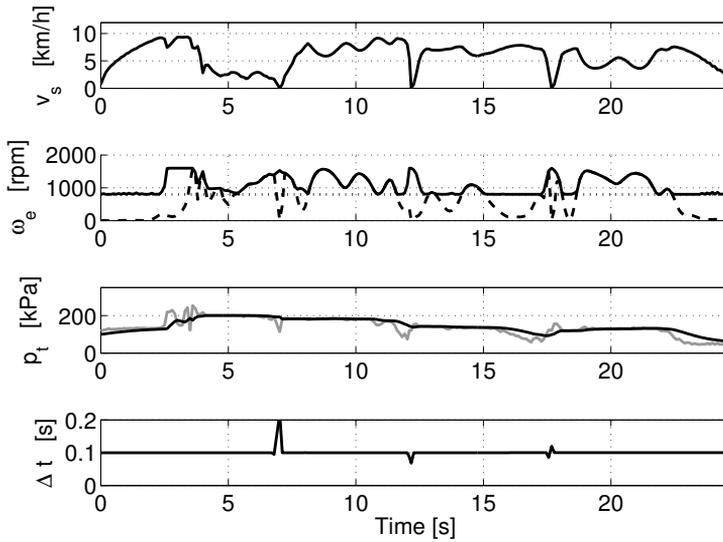


Figure 18: State trajectories in evaluation 4-12, along with the time steps used. The engine speed ω_e is plotted along with the minimum speed given by the hydraulic flow in the simulated cycle (dashed). The intake pressure p_t is plotted along with the set pressure p_{set} (gray).

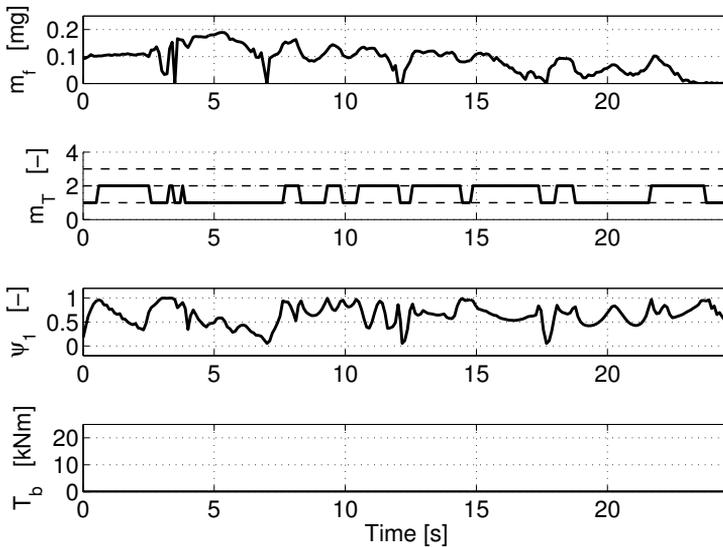


Figure 19: Control signal trajectories in evaluation 4-12.

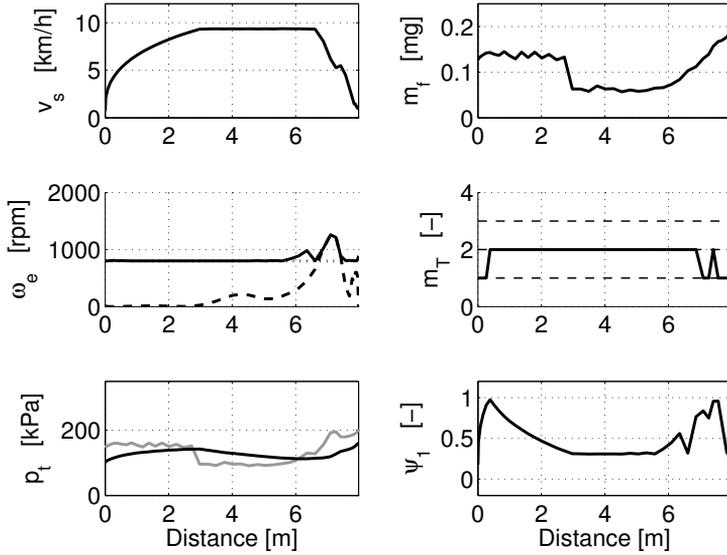


Figure 20: State and control signals before encountering infinite cost in evaluation 23-12 with $\beta = 10g/s$. See Figures 18 and 19 for explanations.

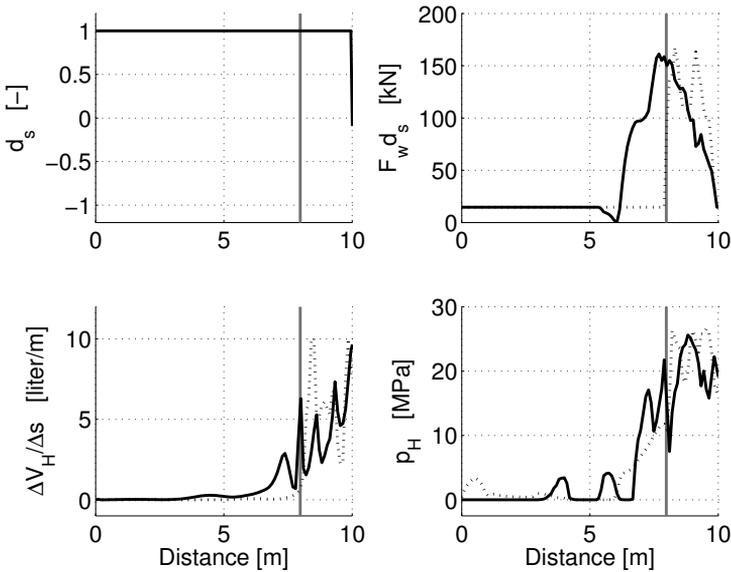


Figure 21: Illustration of the loads in the unsuccessful evaluation 23-12 with $\beta = 10g/s$. The dashed lines are the load used in the CTG-map calculation, the solid lines are the load used in the simulation and the vertical gray line shows the instant of infinite cost.

formed with different values of the time to fuel weighting parameter β . In the evaluations above, $\beta = 5 \cdot 10^{-4}$ was used, since this give an average cycle time similar to the cycle time specified for the SDP method.

Table 5: Impact of different β values in evaluation 4-12.

β [g/s]	T [s]	M_f [g]
5	20.9	124.7
2	22.1	119.6
1	23.2	116.9
0.5	24.6	115.0
0.2	25.9	114.7
0.1	26.3	114.4
0.05	26.4	114.2

Table 6 shows the experienced times needed for calculating the CTG-maps and for the look-ahead control simulations. The simulation times only include the cycles for which the cost is finite. All calculation times are highly dependent on the method implementation and state and control signal grid densities, and should therefore only be considered an indication and are only intended for comparison to the SDP method. The discretizations have been made as sparse as possible without significantly affecting the optimization results.

Table 6: Experienced times for CTG-map calculation and look-ahead simulation, using the FTDP method.

	t_{min} [s]	t_{mean} [s]	t_{max} [s]
CTG	13005	13365	14035
sim	8.28	9.09	10.6

4.3 Methods discussion and comparison

Two methods are created for using dynamic programming as a one step look-ahead controller in a wheel loader application. Each of these use a different approach for increasing the robustness of the look-ahead controller to deviations from the predicted load. This section discusses and compares the two methods, not only with regards to the performance as described in the previous section, but also properties that affect the possibilities of implementing any of the methods as an actual online controller.

CTG-map creation

In both of the methods, the first part of the algorithm is the creation of a cost-to-go (CTG) map $J(x, k)$. The appearance of this map will depend on the load used in the calculation.

In the SDP method, the creation of a load case for use in the CTG calculation can be automated. A dataset containing previously driven cycles, which

might e.g. be from the previous working day, is screened for cycles. A Markov probabilistic cycle is created from the detected cycles, using average and standard deviations of the load at each instance, along with assigned probabilities. The combination of possible loads and the corresponding probabilities are design parameters. The combination used in this evaluation is presented in Table 2. The impact of the combination and probabilities is quite transparent, especially in the impact on robustness to changes in actual hydraulic flow Q_H , while Section 3.2 states that uncertainties in this load component are the most important for the ability to complete the simulation. In the CTG-map calculation, engine speeds lower than required by the maximum possible Q_H , according to (21), renders infinite cost. In the simulation, the result is that the engine speed never drops below that which correspond to the highest predicted Q_H and maximum hydraulic pump displacement $\psi_H = 1$, as shown in Figure 12. Adding even higher possible alternatives for Q_H in the CTG-map calculation will therefore increase robustness to high hydraulic flow in the simulation, but will also increase fuel consumption through maintaining a high engine speed even if this is not required by the actual hydraulic flow.

In the FTDP method a single driving cycle is used as load in the CTG-map calculation. In the evaluation, each of the cycles used in the simulations were also used in the CTG-map calculation, but in a real application a particular, and perhaps designed, cycle would be used. When low cycle-time was prioritized, some cycles were less suited for use in the CTG-map calculation, which shows that the cycle used has a real impact on the performance in the subsequent simulation. Some care should therefore be put into the selection or creation of the CTG driving cycle. Unfortunately, the impact of the appearance of the CTG cycle is less transparent in the FTDP method than in the SDP method, since the hydraulic volume V_H gives a limit to the combination of the vehicle and engine speeds, rather than to the engine speed alone. The FTDP uses one state more than the SDP method, the vehicle speed, which is directly related to the time needed for completing a driving cycle. Completing a cycle faster generally requires more fuel, and the weighting of cycle time to fuel use is governed by the weighting parameter β . Increasing β increases the v_s dependency in $J(x, k)$, which pushes the vehicle toward higher speeds in the simulation. Predicting the impact of different β in the CTG-map calculation, on the simulation of a specific cycle, is not trivial and deciding upon a suitable value may require iteration of CTG-map calculations and look-ahead simulations.

Performance

In evaluating the performance of each of the methods, the first requirement is that the vehicle should be able to perform the specified driving cycles. This is fulfilled if the simulation is completed without the system violating any bound, deviating from the desired trajectory or going into an infinite cost region in the CTG-map. This requirement was not fulfilled in all cycles for any of the two methods, but the ratio of successful to unsuccessful simulations differs for the two, and can therefore be considered a first performance measure. In the SDP method 73.5% of the simulations were successful, compared to 99.5% in

the FTDP method. These numbers depend on the design parameters used in the CTG-map calculations though. Some of the failed SDP simulations did hit the minimum engine speed needed for fulfilling the hydraulic flow requirement while most fails were related to the vehicle speed adjustment made for obtaining the same distances driven in both method evaluations. The robustness to hydraulic flow uncertainties could have been increased through adding an even higher hydraulic flow component in the CTG load case. But as long as the highest hydraulic flow component in the CTG load case does not require maximum engine speed at all times, there will always be a possibility of a higher requirement in the simulation. In the FTDP method the cause of each unsuccessful simulation was less clear. The number of these simulations increases when speed is prioritized, and in this case the success of a simulation was more related to the cycle that was used in the CTG-map calculation than that used in the simulation. Many of the unsuccessful simulations occurred when cycles in which the longitudinal force increase related to bucket filling occurs late, were used in the CTG-map calculation.

Another requirement is low fuel consumption, and the fuel use is therefore the second performance measure. In the SDP method evaluations the average fuel use was $130g$ and the pre-specified cycle time was $25s$. If the robustness to hydraulic flow is increased through increasing the maximum predicted flow, the average fuel use can be expected to increase. In the example of Figure 10, the fuel use is already noticeably higher than the optimal ($119g$), because of the implemented margin towards high hydraulic flow. In the FTDP method evaluations, with $\beta = 0.5g/s$, the average fuel use was $116g$ and the average time use was $24.7s$. Changing β so that the FTDP average time increases to closer to $25s$ might reduce the average fuel consumption somewhat, but Table 5 indicate that this reduction would be small. The fuel use in the FTDP method is close to that achieved with perfect prediction ($115g$), though there is a difference in time use ($24.3s$). The optimal fuel use is lower for the FTDP method because of the addition of another state, or degree of freedom. The actual fuel use is also closer to the optimum for the FTDP method since this method does not need to maintain a power margin through the engine speed, causing a reduction of the efficiency.

Implementation

If any of the two proposed methods is to be used in an online application, there are a few issues still to be addressed.

Both methods rely on the actual position, as referred to the flow variable, being known and the length of the cycle, including the points of driving direction change, being fixed. In a real application this will not be the case, as illustrated by Figure 5. In both methods the position can be reset when a driving direction change occurs, but after these, disturbances in time or position, depending on method, must be handled.

In the simulations it is, at each instant t_k , assumed that the present load w_k is known and constant during each interval. Neither of these assumptions can be expected to hold in a real implementation, and an expected, and possibly

probabilistic, load must therefore be used in the one step look-ahead choice of control signals performed in the simulations. The need for preventing the states of moving into regions with infinite cost-to-go will require over-estimating the load, which can be expected to cause higher fuel consumption.

The time required for calculating a new CTG-map restricts the adaptability of the controllers, regardless of the method used. In the implementations evaluated here, the CTG-map calculation time corresponds to around 50 – 500 times the length of each loading cycle, depending on method used. It will therefore not be possible to quickly create a new CTG-map if the general driving cycle changes, but the CTG-map, or maps, must be created beforehand. This must be addressed if the working site changes from day to day or if different drivers operate the machine. The most critical calculation effort however is in the look-ahead control simulations. This part must be completed online and using a computer much less powerful. The simulations of the 25s cycles required around 0.3s using the SDP method and around 10s using the FTDP method. Despite requiring shorter time than the length of the cycle, the calculation effort is too high for an implementation. Improvements might be possible through approximation of the true CTG-map and improvement of the method for searching for the optimal control action in the one step look-ahead, or even by calculation of an $u^*(x, t, P)$ -map in advance.

5 Summary and Conclusions

Wheel loader operation is often highly repetitive. This repeating of similar motions may be used as the basis of a prediction of future operation. If a prediction of the future load trajectory is available, this can be used in an optimization of engine and transmission operation. In this paper a wheel loader with a three mode CVT is studied. Predictions of future loads have been used in actual control systems before, e.g. in [26], but only for on-road vehicles. A prediction based only on repetition however will become approximate and contain uncertainties. The complexity of the wheel loader system and its operation, along with the introduction of considerable uncertainties in the load prediction, makes it necessary to expand previously presented methods. Two conceptual methods, based on dynamic programming, for one step look-ahead control of a wheel loader transmission are developed and presented in this paper.

A wheel loader driving cycle can be represented by a bucket trajectory and the corresponding vertical and longitudinal forces. A measurement sequence which contains 34 short loading cycles, described by vehicle speed, hydraulic flow (change of bucket height), hydraulic pressure (vertical force) and tractive (longitudinal) force, is used throughout the paper. The most important prediction uncertainties are in the hydraulic flow. The two controller concepts are evaluated through their performance in each of these 34 cycles, in each case having the other 33 cycles available for use as a load prediction. Deviating from the desired trajectory is not allowed, since this would require introducing another state in the optimization and a gravel pile model for calculating new forces, a model which is not readily available.

The first method presented is based on stochastic dynamic programming and is designated SDP. In this method the 33 cycles available for the prediction are condensed into a statistical cycle with several possible loads at each instant in time, and an estimated cost-to-go (CTG) map is calculated from this cycle. The second method, designated FTDP, has vehicle speed as a state of the system, and introduce a fixed 0.1s delay from driver input to bucket movement, a delay equivalent to a prediction of bucket movement. Again, a cost-to-go map is calculated. The CTG maps are in both methods used in a one step look-ahead controller for, at each instant, selecting the control action that can be expected to minimize the cost for completing the driving cycle.

The SDP method implementation turns out to require about 1/10:th of the computational time of the FTDP method, both in the CTG-map calculations and in the subsequent simulations. The lower time is because the SDP method has two states while the FTDP method has three, and even though the SDP method has several load alternatives at each instant. The most important performance measure is the ratio of cycles for which the look-ahead simulation could be completed without violating any bound, deviating from the desired trajectory or going into an infinite cost region in the CTG-map. These simulations are regarded as successful. In the SDP method evaluation 74% of the 34 simulations were successful. In the FTDP method evaluation the ratio of successful simulations depends on the value of the time to fuel weighting parameter β . Using a β which gives cycle times similar to that specified in the SDP solving rendered 99.5% of 1122 evaluations successful. Increasing the weight on time in the CTG-map calculation increases the importance of the choice of cycle to use in the CTG-map calculation and reduces the ratio of successful simulations. The second performance measure is the fuel use. In the SDP method evaluations the average fuel use was 130g and the pre-specified cycle time was 25s. In the FTDP method evaluations, with $\beta = 5 \cdot 10^{-4}$, the average fuel use was 116g and the average time use was 24.7s.

The driving cycle used in the CTG-map creation affects the result of the one step look-ahead simulation. In the SDP method the impact is relatively transparent, especially with respect to robustness to different hydraulic flows. The CTG load can be used to trade increased robustness to hydraulic flow for higher fuel consumption. The FTDP method seems to be less sensitive to the load used in the CTG-map calculation, unless cycle time is prioritized. In any case, the impact of the CTG cycle is less transparent in FTDP than in SDP.

In all, this evaluation shows that both methods may have a potential for use in a one step look-ahead controller for a wheel loader transmission, but that there are still issues to be addressed before implementation, especially the treatment of uncertainties in the prediction of distance driven. In the evaluation, the SDP method required about 1/10:th of the computational effort of the FTDP method and has better transparency of the impact of the CTG load. On the other hand, the vehicle was unable to complete the cycle in 26% of the evaluations when using the SDP method, as compared to a fail rate of less than 1% for the FTDP method, while the FTDP method also showed a 10% lower fuel consumption.

References

- [1] R. Filla. Alternative systems solutions for wheel loaders and other construction equipment. In *1st International CTI Forum Alternative and Hybrid Drive Trains*. CTI, 2008.
- [2] K-E. Rydberg. Hydrostatic drives in heavy mobile machinery - new concepts and development trends. In *International Off-Highway & Powerplant Congress & Exposition*, number 981989. SAE, 1998.
- [3] B. Carl, M. Iivantysynova, and K. Williams. Comparison of operational characteristics in power split continuously variable transmissions. In *Commercial Vehicle Engineering Congress and Exhibition*, number 2006-01-3468. SAE, 2006.
- [4] S. Grammatico, A. Balluchi, and E. Cosoli. A series-parallel hybrid electric powertrain for industrial vehicles. In *2010 IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, 2010.
- [5] S. Savaresi, F. Taroni, F. Previdi, and S. Bittanti. Control system design on a power-split cvt for high-power agricultural tractors. *IEEE/ASME Transactions on Mechatronics*, 9(3):569–579, 2004.
- [6] J. Lennevi. *Hydrostatic Transmission Control, Design Methodology for Vehicular Drivetrain Applications*. dissertation, Linköping University, 1995.
- [7] R. Zhang. *Multivariable Robust Control of Nonlinear Systems with Application to an Electro-Hydraulic Powertrain*. dissertation, University of Illinois, 2002.
- [8] R. Filla. An event driven operator model for dynamic simulation of construction machinery. In *Proceedings from the Ninth Scandinavian International Conference on Fluid Power*. Linköping University, 2005.
- [9] F. Wang, J. Zhang, R. Sun, and F. Yu. Analysis on the performance of wheel loaders in typical work cycle. *Applied Mechanics and Materials*, 148:526–529, 2012.
- [10] E. Hellström, J. Åslund, and L. Nielsen. Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice*, 18(11):1318–1327, 2010.
- [11] B. Asadi and 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):707–714, 2011.
- [12] H. Khayyam, S. Nahavandi, and S. Davis. Adaptive cruise control look-ahead system for energy management of vehicles. *Expert Systems with Applications*, 39(3):3874–3885, 2012.
- [13] B. Frank, J. Pohl, and J-O. Palmberg. Estimation of the potential in predictive control in a hybrid wheel loader. In *SICFP'09 proceedings, The 11:th Scandinavian International Conference on Fluid Power*, 2009.
- [14] G. Rizzoni, L. Guzzella, and B.M. Baumann. Unified modeling of hybrid electric vehicle drivetrains. *IEEE/ASME Transactions on Mechatronics*, 4:246–257, 1999.

- [15] P. Mattsson and M. Åkerblom. Continuously variable transmission and a working machine including a continuously variable transmission. Patent, 2012. WO 2012/008884 A1.
- [16] C. Lauinger, A. Englisch, A. Gotz, A. Teubert, E. Muller, and A. Baumgartner. Cvt components for powersplit commercial vehicle transmissions. In *Proceedings of the 6th International CTI Symposium*. CTI, 2007.
- [17] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [18] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3 edition, 2005.
- [19] T. Nilsson, A. Fröberg, and J. Åslund. Optimal operation of a turbocharged diesel engine during transients. *SAE International Journal of Engines*, 5(2):571–578, 2012.
- [20] J.E. Silva and J.B. Sousa. Dynamic programming techniques for feedback control. In *Preprints of the 18th IFAC World Congress*, pages 6857–6862. IFAC, 2011.
- [21] K. McDonough, I. Kolmanovsky, D. Filev, D. Yanakiev, S. Szabowski, and J. Micheli. Stochastic dynamic programming control policies for fuel efficient in-traffic driving. In *American Control Conference (ACC), 2012*, pages 3986–3991. IEEE, 2012.
- [22] T. Nilsson, A. Fröberg, and J. Åslund. Fuel potential and prediction sensitivity of a power-split cvt in a wheel loader. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, pages 49–56. IFAC, 2012.
- [23] T. Nilsson, A. Fröberg, and J. Åslund. On the use of stochastic dynamic programming for evaluating a power-split cvt in a wheel loader. In *8th IEEE Vehicle Power and Propulsion Conference*, pages 840–845. IEEE, 2012.
- [24] T. Leroy, J. Malaize, and G. Corde. Towards real-time optimal energy management of hev powertrains using stochastic dynamic programming. In *8th IEEE Vehicle Power and Propulsion Conference*, pages 383–388. IEEE, 2012.
- [25] T. Nilsson, A. Fröberg, and J. Åslund. Fuel and time minimization in a cvt wheel loader application. In *7th IFAC Symposium on Advances in Automotive Control*. IFAC, 2013.
- [26] E. Hellström. *Look-ahead Control of Heavy Vehicles*. dissertation, Linköping University, 2010.

Predictive control of a diesel electric wheel loader powertrain[†]

Tomas Nilsson, Anders Fröberg and Jan Åslund

*Vehicular Systems, Department of Electrical Engineering,
Linköping University, S-581 83 Linköping, Sweden.*

4

Abstract

Wheel loaders often have a highly repetitive pattern of operation, which can be used for creating a rough prediction of future operation. As the present torque converter based transmission is replaced with an infinitely variable device, such as an electric or hydraulic transmission, a freedom in the choice of engine speed is introduced. This choice is far from trivial in the extremely transient operation of these machines, but the availability of a load prediction should be utilized.

In this paper, a predictive engine and generator controller, based on stochastic dynamic programming, is described, implemented and evaluated. The evaluation is performed against non-predictive controllers in the same system, to lift out any possible benefits of utilizing the repetition based prediction. Simulations and field tests show that the controllers are able to handle disturbances introduced from model errors, the machine environment and the human operator, and that the predictive controller gives around 5% lower fuel consumption than the non-predictive reference controllers.

[†]This is a formatted version of “Predictive control of a diesel electric wheel loader powertrain” by Tomas Nilsson, Anders Fröberg and Jan Åslund, submitted to Control Engineering Practice. The formatting is restricted to changing the article into a single-column format, adjusting sizes of figures and tables, and adjusting the referencing style.

1 Introduction

1.1 Background

Wheel loader operation is highly transient and repetitive, and contains periods of high tractive force at low speeds, while the engine delivers power to both the transmission and to the hydraulics. The most common layout of heavy wheel loader powertrains is presented in Figure 1. The engine is connected to a hydraulic pump and a torque converter. The torque converter is connected to an automatic gearbox, which connects to the drive shaft.

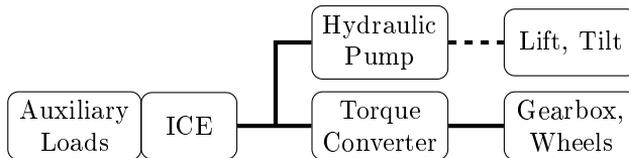


Figure 1: The reference vehicle powertrain setup. Solid lines are mechanical connections and dashed lines are hydraulic connections.

In this setup the torque converter is a crucial component since it provides some disconnection between the engine and vehicle speeds. This disconnection makes the system mechanically robust but it also causes high losses. Just as for most other vehicles, substantial work is ongoing for reducing emissions and increasing fuel efficiency. One approach for reducing fuel consumption is to replace the torque converter based transmission with another solution. The combination of low speeds, high forces and transient operation motivates the use of some type of continuously variable transmission (CVT).

The repetitiveness of the operation may form the basis of a prediction of future operation, which might be used in a predictive controller. This paper studies such repetition based predictive control of the engine and generator in a diesel electric wheel loader.

1.2 Problem formulation

The problem studied in this paper is the minimization of the expected amount of fuel needed for completing a series of loading cycles, using a CVT wheel loader. This problem was studied in the paper [1], though only through simulations. In this paper the developed control strategy is implemented and field tested, using the series hybrid electric vehicle described in [2].

The machine is operated by a human driver, which introduces major uncertainties in the power trajectories. The machine is also operated in an environment which is difficult to model. This is particularly evident in the pile of material from which the machine fills its bucket. The tests are designed to investigate the performance of the controller under such severe prediction uncertainties.

1.3 Previous work

There is a substantial amount of work done on different types of advanced drivetrains for construction equipment. An evaluation of the minimum fuel consumptions of the standard and a CVT vehicle is made in [3]. Several other papers, such as [4], [5], [6] and [7] study the use of hydrostatic CVTs in wheel loaders, though the focus is on component control. In [8] and [9] several advanced drivetrains are presented. In [2] a series electric hybrid and the hydrostatic device studied in [3] are presented. Neither [8], [9] nor [2] describe any controllers though. An electric transmission is used in this paper since hydrostatic transmissions has much slower component dynamics, which cannot be neglected. The series hybrid described in [2] is used for the online tests described in this paper.

An investigation into differences in driver performance and style in traditional machines can be found in [10].

A few heuristic CVT control strategies can be found in [11]. These control concepts however do not fully utilize the potential of the transmissions. Conversely, there are some papers, such as e.g. [12], [13] and [14], treating optimal transient engine operation. These, on the other hand, are based on perfect predictions and do not present solutions for handling disturbances. There are other areas in which optimal control without perfect prediction has been implemented. For on-road vehicles, and apart from ECMS, as described in [15], there have been several proposals, e.g. [16] and [17], for utilizing increased availability of information, such as road maps and GPS data, for predictive control. This type of information is in general not available for off-road applications. Wheel loader operation is often highly repetitive and often follows one of a few common patterns, which might enable a prediction based on pattern recognition, such as that presented in [18] or in [19]. In [20], [21], [22] and [23] stochastic dynamic programming (SDP) is used for optimal control based on an uncertain prediction, though these all treat on-road vehicles. The paper [1] analyses three different SDP implementations for use in a CVT wheel loader. This last paper forms the basis for the controllers implemented and tested in the work presented here.

2 System and concept overview

This paper investigates the benefit of utilizing the repetitiveness of wheel loader operation for making a load prediction, and controlling the engine speed in a diesel electric transmission based on this prediction. This section describes the operation, the vehicle and the transmission, and the optimization involved in building the controller. The main controller, the two reference controllers and some auxiliary controllers are described in Section 3.

2.1 The Short Loading Cycle

One of the most common operational patterns for wheel loaders is the short loading cycle (SLC). In this cycle, the machine loads material from a source, a pile, to a receiver, often a dump truck. Figure 2 gives an overview of a

short loading cycle. In the definition of the SLC used here the cycle consists of four legs which are defined by the driving direction changes. Referring to the notations in Figure 2, the first leg starts when the vehicle leaves point 4 and ends when the vehicle has filled the bucket at point 1. The second leg is the movement from point 1 to point 4, the third is the movement from point 4 to point 6 and the fourth is the movement from point 6 to point 4. The bucket is raised during leg two and three and emptied at point 6. Each leg commonly have a duration of 5–10s, with another 5–10s for filling the bucket. After some cycles, usually around four, the receiver is full and has to be replaced. During this the machine is usually resting in a dormant state. The dormant state is not included in this paper since it is easily detected and handled. Details about this operation can be found e.g. in [24], [25] and [26].

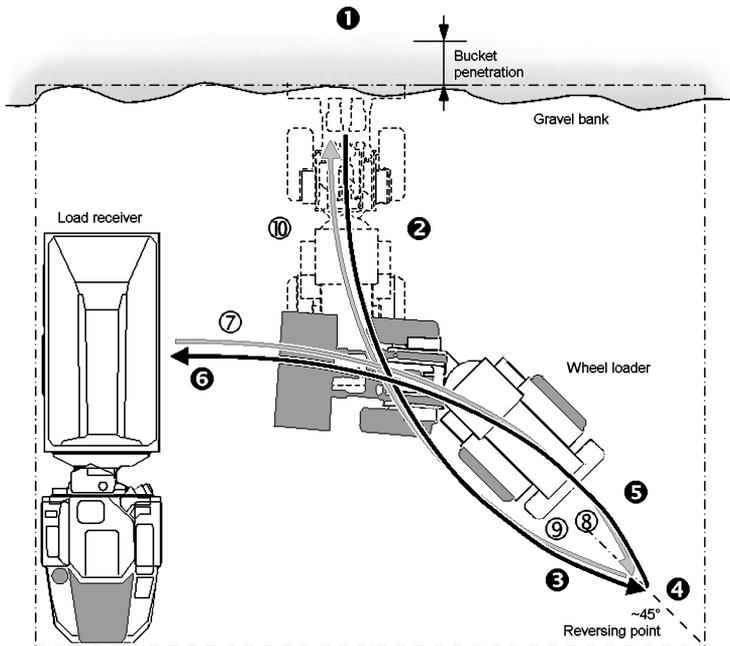


Figure 2: Overview of the short loading cycle, from [8]

2.2 Test vehicle description

The control strategies of this paper are designed for an arbitrary infinitely variable transmission (IVT), such as a diesel electric or hydrostatic system. The vehicle available for the testing was the supercapacitor series hybrid vehicle presented in [2], and which is based on a production Volvo wheel loader [27], and this is controlled so to avoid using the supercapacitor, making a diesel electric powertrain.

The layout of the test vehicle powertrain is presented in Figure 3. To the left is the engine-generator set, and to the right are the electrically powered

hydraulic pumps for the bucket and arm and the electric propulsion. The auxiliary loads consist mainly of the fan and hydraulic steering. The auxiliary loads, engine and generator are together denoted the 'genset'. The driver has superior control of the propulsion and the bucket lift and tilt, along with some of the auxiliary loads, especially the steering. In this project the right hand side of the system is seen as a set of power consumers, controlled by the driver to produce a requested power. Because of this, only the genset is modeled, and the right hand side is seen as an external power demand. There is however an important right hand side controller, that should be noted, known as the machine balance, which governs the partitioning of power between the hydraulics and the propulsion, especially when the available power is lower than the total requested power. This controller has the available power as one of its inputs and the actual power used as one of its outputs. The genset controller therefore has the instantaneous actual consumed power available as an input, and can have the maximum allowed power consumption as an output. The auxiliary loads are however not available for the controller.

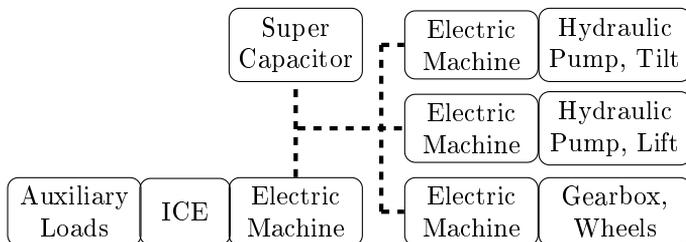


Figure 3: The test vehicle powertrain setup. Solid lines are mechanical connections and dashed lines are electric connections.

The vehicle is equipped with a dSpace Autobox which holds the genset controller. This is connected to a laptop running dSpace ControlDesk, from which the controller is loaded. The controller itself is implemented in Simulink, compiled and loaded into ControlDesk.

2.3 System models

The complete system studied is presented in Figure 3. Any relevant dynamics and efficiencies in the power consuming parts are included in the requested power of the driving cycle, since these devices are controlled by the driver. The auxiliary loads are neglected in the modelling since these cannot be measured in the evaluation vehicle. The following model, which describes the genset, is identical to that used in [13], apart from parameter values and that a smoke limiter and a generator efficiency model is added. The system consists of a turbo charged diesel engine and an electric machine. Apart from the inertia, the dynamics of the electric machine is assumed to be fast compared to the other dynamics of the system.

The genset is modeled as an inertia I_e which is affected by the engine torque

T_e and the generator torque T_g .

$$\frac{d\omega_e}{dt} I_e = T_e - T_g \quad (1)$$

The generator torque is modeled as the electric power from the generator, which is the same as the load power P_L , along with a quadratic loss function according to

$$P_{g,loss} = g_1 T_g^2 + g_2 |\omega_e| + g_3 \quad (2)$$

$$P_L = T_g \omega_e \pm P_{g,loss} \quad (3)$$

The relation between injected fuel and engine torque is described by a quadratic Willan's efficiency model, as presented in [28], expanded with a torque loss due to low intake manifold pressure

$$T_e = e(\omega_e, m_f) \frac{q_{lHV} n_{cyl}}{2\pi n_r} m_f - T_L(\omega_e) - T_{pt} \quad (4)$$

Here m_f is fuel mass per injection, ω_e is engine speed, e and T_L are efficiency functions. T_{pt} is torque loss due to low air intake pressure $p_{off} = p_t - p_{set}(\omega_e, m_f)$ caused by low turbo charger speed. The actual pressure is p_t and p_{set} is a static setpoint map. The turbo dynamics is modeled as a first order delay for the intake air pressure

$$\frac{dp_t}{dt} \tau(\omega_e) = -p_{off}(\omega_e, m_f) \quad (5)$$

The torque loss from low pressure is described by

$$T_{pt} = \begin{cases} k_1(\omega_e) p_{off}^2 - k_2(\omega_e) p_{off} & \text{if } p_{off} < 0 \\ 0 & \text{if } p_{off} \geq 0 \end{cases} \quad (6)$$

The fuel per injection is related to fuel flow according to

$$\frac{dM_f}{dt} = m_f \frac{n_{cyl}}{4\pi} \omega_e \quad (7)$$

and figure 4 presents the efficiency map of the engine used. The gray lines indicate allowed operating region (minimum speed and maximum torque) and the black line indicates the static optimal operating points for each output power. The figure also shows efficiency levels and output power lines with kW markings.

2.4 Formal problem formulation

The problem studied here is the minimization of the expected average amount of fuel needed for completing each cycle in a long series of loading cycles. For a fixed time or distance cycle, this can be formulated as a minimization of the expected average fuel flow. With U representing a set of control signals and M_f the total fuel used, this can be expressed as

$$\min_{U(t)} \lim_{T \rightarrow \infty} E \frac{1}{T} \int_0^T \frac{dM_f}{dt} dt \quad (8)$$

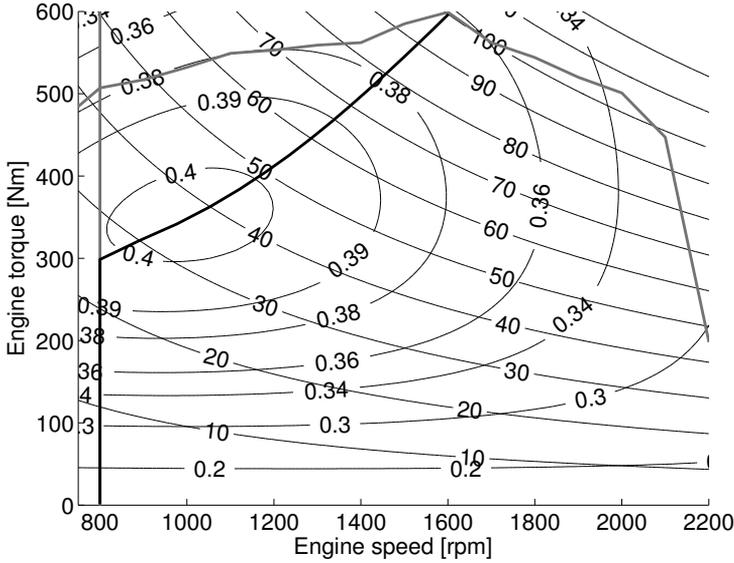


Figure 4: Engine map with efficiency curves, output power with kW markings, minimum speed and maximum torque bounds and the static optimal line.

while obeying the system dynamics and constraints, which includes the load power P_L . The driving cycles consist of a stochastic requested power P_R as a function of distance driven, and a deterministic vehicle speed v_s used to give a distance to time relation. If it would have been required that the genset should always be able to supply any possible requested power, even when the corresponding probability is low, the large uncertainties and spread in P_R would force the controller to always select high engine speeds. Since this would be detrimental to the efficiency, a limiter P_{max} for the actual electric load P_L is introduced according to

$$P_L = \min(P_R, P_{max}) \quad (9)$$

This limiter should only be used if the fuel gain of reducing P_{max} is large and the probability of $P_R > P_{max}$ is low. Since P_{max} does not hold any natural penalty, an artificial penalty function $G(P_{max})$ is introduced and added to the objective function. In the optimization it is also assumed that this limiter will be used sparsely and that the rest of the cycle will therefore not be affected by any usage.

2.5 Control concept overview

The objective for the genset controller is to minimize the fuel use while also minimizing the restriction imposed on the power consumers through the power limitation. The genset controller can affect the engine speed and the intake pressure through the injected fuel and the available power limiter. A set of

previously driven cycles is assumed to be available, and from these a probabilistic cycle is created. This cycle consists of a requested power probability distribution as a function of distance driven. Optimization performed against this probability distribution produces a state feedback control scheme which is used in the vehicle. The optimization is time consuming and cannot be done online, but rather on a day to day basis. Since the available vehicle has not been operated as an IVT vehicle, two reference control schemes are also created. The optimization is described in Section 2.6 and the controller implementations are described in Section 3. The choice of controller concepts is based on the investigation in [1].

2.6 Control optimization

The optimization problem is formulated as an average cost per stage problem. In this paper, stochastic dynamic programming (SDP) and the relative value iteration algorithm, as described in detail in [29] and in [30], is used for solving this problem. This approach is also used in [21] and [23]. The controller can be summarized as follows.

Denote the discretized state $x \in X$, control $u \in U$ and load $w \in W$ with the corresponding probability p , all with subscripts to denote the instant k . Select arbitrary values for J^1 , an arbitrary state x^* , and an optimality condition ε . Now, update J according to

- 1: **while** $\rho > \varepsilon$ **do**
- 2: For each $x \in X$, calculate

$$J^{i+1}(x) = \min_{u \in U} \sum_{w_l \in W} p(w_l)(g(x_k, u, w_l) + \dots + \tilde{J}^i x_{k+1}(x_k, u, w_l)) \quad (10)$$

in which \tilde{J}^i is interpolated from $J^i(x \in X)$

- 3: set $J^{i+1} \leftarrow J^{i+1} - J^{i+1}(x^*)$
- 4: calculate $\rho = \sup(J^i - J^{i+1})$
- 5: **end while**

Since selecting a proper ε value is not trivial, in this work the iterations are instead repeated until the resulting simulated state trajectories ceases to change. The result of the algorithm is a map which assigns a value to each state, $J(x \in X)$. This map can be directly used in a controller which at each instant selects the control action which minimizes the function

$$\hat{u}_k = \operatorname{argmin}_{u \in U} \sum_{w_l \in W_k} p(w_l)(g(\hat{x}_k, u_k, w_l) + \tilde{J} x_{k+1}(\hat{x}_k, u_k, w_l)) \quad (11)$$

in which \hat{x} and \hat{u} denote the actual state and control and \tilde{J} is interpolated among $J(x \in X)$. Since $p(w_l)$ depends on the state \hat{x} , the control \hat{u} will be a function of the state, and this function $\hat{u}(x \in X)$ can therefore be precalculated to save online computational complexity, as is done in this work. For states $x \notin X$ the control action is interpolated from $\hat{u}(x \in X)$. In this interpolation it

is assumed that $\hat{u}(x \in X)$ is sufficiently smooth. If this is not the case, and there are e.g. switching regions, some other solution should be found. The proper solution is to go back to solving (11) if \hat{x}_k appears to be in such a region, e.g. if $|\nabla(\hat{u}(\hat{x}_k))| > \kappa$. Another, simple, solution is to use nearest neighbor interpolation in such cases. In this work though, only linear interpolation is used even though there are switching regions, since speed and simplicity are essential and the nearest neighbor solution caused severe control signal oscillations at some occasions in simulations.

A major part of the design of an SDP controller is to select states to include in X and to formulate the probability functions. The states used here are the two essential engine states ω_e and p_t and the distance driven s . The load W consists of a constant P_R vector and state dependent probability distributions p . The probabilities could be based on e.g. previous power, as in [21], acceleration or a position related state. [20] implements two of these for on-road vehicles. This paper evaluates the benefit of utilizing rough cycle information in a predictive controller. The prediction is based on previously driven cycles and can best be formulated as a probabilistic load as a function of distance driven, $p(P_{R,k}|s_k)$. In this formulation the probabilities are assumed normally distributed due to the low number of measurements available at each distance s_k . The vehicle speed is described as deterministic, not only because this reduces the number of possible load combinations, but also because s can then be discretized so that s_{k+1} is always on the grid, even though a fixed time step of $0.1s$ is used in the optimization. The interpolations of \tilde{J} can therefore be made two dimensional in the J map calculation. Since there is a distinct flow direction in the $p(P_R|s)$ concept, \tilde{J}^i in (10) is replaced with \tilde{J}^{i+1} except in the last s state, and the calculation is proceeded in a backwards s -direction, thus making this more similar to classic dynamic programming. This has a profound impact on the number of iterations until the solution settles, reducing the number from > 25 to 2 or 3. To make a fair evaluation of this truly predictive control scheme, a non-predictive scheme is created from the same set of data from previously driven cycles. In this scheme the load probability is assumed independent of all states, $p(P_{R,k}|-)$. This later control scheme is selected as a reference based on the results presented in [1].

A natural choice of control signals for this system is fuel flow m_f , or engine torque T_e , along with P_{max} . The large span in P_R does however mean that for any given m_f , some of the $\omega_{e,k+1}(x_k, u, P_R)$ will always be outside the valid operating region, causing infinite expected cost. The only valid solution is for the controller to always keep P_{max} near zero. Since this is not acceptable, another control signal setup is proposed. It is assumed that there exist an engine speed controller which is fast compared to the update frequency of the SDP controller. By inverting the engine model, $\frac{d\omega_e}{dt}$ can then be used as control signal instead of m_f . The control signals in U are therefore selected to be $\frac{d\omega_e}{dt}$ and P_{max} , and the fuel flow is calculated from the engine model (4). The signal P_{max} does not carry any natural penalty, and artificial penalties $G_{P_{max}}$ are therefore introduced. These penalty functions are described with the corresponding controllers in Section 3.

2.7 Simulation models

The control schemes created through the optimization are first evaluated in basic simulations, which use the same models as used in the optimization. These simple simulations, along with directly studying the $\hat{u}(x)$ maps, can give an initial indication of the properties of a new control scheme, but they do not catch the impact of model errors, the auxiliary controllers nor the feedback from the driver.

A more complex Simulink simulation model was also available along with the test vehicle and is used here for more complete pre-evaluations of new control schemes. This model includes the actual controller that is later compiled and loaded into the dSpace Autobox in the machine. The Simulink model consists of more complete models of the supercapacitor and the components of the genset, including simple models of the auxiliary loads, as presented in Figure 3. It does not include models of any of the power consuming parts of the machine though, nor does it include models of the driver or the environment.

3 Controller descriptions

The optimization produces control schemes with desired engine speed derivative and maximum available power as outputs. The available controller framework does not agree with this description though. The framework, which includes the machine balance controller and lower level engine and generator controllers, requires engine and generator torque references and a maximum available power for the machine balance controller. Auxiliary controllers are therefore needed for controlling the engine speed, for suppressing the impact of the supercapacitor, and for making necessary adjustments to the maximum available power. These auxiliary controllers are identical in each of the three genset controllers.

The outputs from the genset controller are engine torque reference T_e , generator torque reference T_g and total available power P_{max} . The inputs used in the genset controller are actual engine and vehicle speeds, $\omega_{e,act}$ and $v_{s,act}$, actual engine torque $T_{e,act}$ (from an external model), actual output power P_L (complies with P_{max}) and the supercapacitor state of charge SOC . The intake manifold pressure is not available but is reconstructed from the turbocharger model (5), with the actual engine speed and torque as inputs. The layout of the control system, with the genset controller, turbocharger model and the auxiliary controllers, is presented in Figure 5.

This section describes the auxiliary controllers, followed by the implementations of the evaluated control schemes.

3.1 Auxiliary controllers

Generator controller

The generator should be controlled so that it produces the same electrical power as drawn by the power consumers. The lower level control of the generator is made for torque control, and not for electric power control though, and the

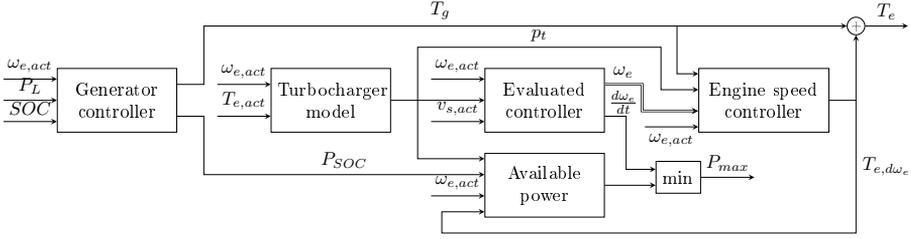


Figure 5: Overall genset control system layout. The 'evaluated controller' block is the location of the main controllers; the two reference controllers and the predictive controller, that are the subject of this paper.

produced and consumed powers will therefore not exactly match. Since the vehicle used in the tests has a supercapacitor fitted at the electrical connection between the genset and the power consumers, this device will absorb any difference as a change of state of charge, but since this paper treats IVTs this must be prevented.

A PI-feedback from the supercapacitor state of charge produces a power to charge or discharge the capacitor, and this power is added to the consumed power P_L . The sum of these is fed to the generator model (3), to calculate a generator reference torque which is an output from the genset controller. The charging power is also fed to the available power estimator since this reduces the power available for the power consumers.

Engine speed controller

The role of the engine speed controller is to decide for an engine torque reference so to produce a desired engine speed, in the first (constant speed) control concept, and a desired engine speed derivative in the other two (optimized) controllers. The engine speed is governed through applying generator and engine torque, according to (1). The desired engine torque is calculated from the load torque and desired speed derivative according to the same engine dynamics model (1). This model does not include the torques from the auxiliary loads since the actual torques from these auxiliary loads are not measured neither for the optimization nor in the online controller. A representative constant auxiliary load is however added to the generator torque when calculating the desired engine torque. In the case of an engine speed reference, which is the case mainly for the first controller, a PI-feedback part is added to the engine torque to hold the engine speed at its desired value. In case of an engine speed derivative reference, this derivative corresponds to the inertia torque of (1). Including a feedback component for the engine speed derivative has been deemed impractical due to the rapid changes in the loads and in the quick shifts in the derivative reference. Because of the relatively small inertia torque, compared to the model errors and the unmeasured auxiliary loads, it cannot be expected that the desired engine speed derivative will be achieved. Nevertheless, since these components are the only part that separates the different controllers, apart

from the maximum power, any difference in fuel consumption and/or driveability must be referred to this control output.

Available power

Since the controllers described above alter the output from the optimized control schemes, and there are highly relevant limitations in the system, the maximum available power must also be lowered if necessary. An actual limit evaluation is performed, in which the static maximum torques for the engine and generator and a smoke limiter are checked, the powers required for the engine speed control, except for the generator torque and supercapacitor control, are subtracted at the proper positions, and a constant generator efficiency is assumed. The remaining maximum power is assumed available for the driver controlled power consumers. This available power limit is compared to the output from the active control scheme and the lower limit is the output from the genset controller.

3.2 Controller 1, $\omega_e = k$, (reference)

The first controller which is evaluated uses a constant engine speed reference and imposes no artificial limitation on the available power. The engine speed reference value is selected based on simulations and there is no optimization involved. All of the auxiliary controllers described above are however used also here. This control scheme is included as a very simple reference for the optimized schemes.

3.3 Controller 2, $p(P_R| -)$, (reference)

The second control scheme which is evaluated consists of maps with engine speed derivative and maximum power as functions of engine speed and turbo pressure, along with the auxiliary controllers. The maps are calculated from the state independent probability function, $p(P_R| -)$, which is the P_R distribution from earlier measurements. The data only includes the power during active cycle driving.

Simple simulations were used to select proper state discretizations and find a proper cost function associated to the use of P_{max} , along with parameters for this function. The best results were found when using the function

$$G_{P_{max}} = \beta_2 P_{max} \quad (12)$$

with $\beta_2 = 3 \cdot 10^{-9}$.

In the controller, the desired engine speed derivative and maximum power are found by linear interpolation in the control maps $\hat{u}(x)$, from ω_e and p_t . The state signals are clipped at the limits of the maps. The engine speed derivative affects the desired engine torque both through the inertia torque and through the engine speed reference. The speed reference is set to be the actual speed plus the desired speed derivative, with a gain. This later component is added to increase the impact of the controller relative to the disturbances from the auxiliary loads. Adding a low pass filter on the actual engine speed, to provide

some stabilization, has also been considered and tested in simulations but is not used in the actual tests. Identical ω_e and $\frac{d\omega_e}{dt}$ reference signal calculations from the map outputs are used in the implementation of Controller 3.

3.4 Controller 3, $p(P_R|s)$, (predictive)

The third control scheme which is evaluated is identical to the second controller, apart from the maps with engine speed derivative and maximum power, especially in that these also depend on the distance driven. The maps are calculated from a distance dependent requested power probability function, $p(P_R|s)$. The probability function is calculated as a normal distribution adapted to the P_R distribution at each distance from a set of previous measurements. In the optimization data, the length of the legs in each cycle are normalized according to a pre-specified leg size, so that the driving direction changes occur at the same distance in each cycle. The average speed as a function of distance is used for discretizing the distance with a fixed time interval, as described in Section 2.6.

Simple simulations were used to select proper state discretizations and find a proper cost function associated to the use of P_{max} , along with parameters for this function. The best results were found when using the function

$$G_{P_{max}} = \alpha_3 p(P_{max} < P_R) + \beta_3 P_{max} \quad (13)$$

with $\alpha_3 = 10^{-3}$ and $\beta_3 = 10^{-12}$.

In the controller, the desired engine speed derivative and maximum power are found through linear interpolation in the control maps $\hat{u}(x)$, from s , ω_e and p_t . Just as in controller 2, the state signals ω_e and p_t are clipped at the limits of the maps. The distance driven in each cycle requires some more in depth description.

The distance input to the map has the range $[1,5)$, where the integer part is the active leg and the fractional part is the normalized distance driven in that leg. The active leg is incremented at each driving direction change until leg 4. The distance driven in the active leg is reset at each change of driving direction, integrated from the vehicle speed, normalized by the leg size used in the optimization, and clipped at $[0, 1]$.

Since there is no automatic detection of a cycle start, this had to be indicated by the driver. The cycle detector substitute is implemented as a push-button in the GUI of the dSpace ControlDesk controller. When this button is pushed, the detector becomes active, and when active the next change to forward driving is considered the start of a new driving cycle, and the leg counter is set to leg 1. Leg 2 of the cycle is used as a reset signal for the detector, making it inactive until the next pressing of the button. During continuous cycle operation the cycle detector button could therefore be pressed during leg 3 and 4.

Apart from the leg and distance counter, and the maps having this as a third input, all components and parameter values are identical to those of controller 2, especially including all auxiliary controllers.

4 Measurements

There are two general types of data used in this project. The first type is some data that was available before the start of the project and which is used in the optimizations and simulations. The second type are that collected in the controller field tests. This section describes the previously available data and the field tests measurement setup.

4.1 Previously available data

In the initial controller development, presented in [1], a set of 34 loading cycles are used. These were collected as a single sequence using a production Volvo L180E machine, with only production sensors. The power and position trajectories are calculated from the torque converter input and output speeds, the hydraulic pressure and the arm and bucket angle (θ_1 & θ_2) derivatives, and the powers are scaled according to the maximum engine power, to fit the series hybrid machine used in the tests presented in this paper. The procedure is presented in detail in [1]. Finally, the loading cycles are separated using the cycle detector presented in [19] and the distances in the cycles are adjusted to fit in the optimization in Controller 3 ($p(P_R|s)$), in particular the driving direction changes should occur at the same distances in each cycle. The resulting cycles were used in the optimization that produced the maps used in the field tests. The cycles were also used in the controller development, which included tuning of the P_{max} cost parameters, in the initial evaluation by simulating a single cycle from the separated set and in the Simulink evaluation, but in the last case using the entire sequence. The scaled but not separated sequence is presented in Figure 6.

Some datasets from earlier operation with the series hybrid were also made available. The set deemed to be the most representative of real operation consists of 20 loading cycles divided into four sets with pauses in between. The pauses are episodes where the load receiver is being replaced or emptied. This dataset was used as a more realistic scenario for the Simulink evaluation, before testing the controllers in the actual machine. For this reason, the cycles in this dataset are not separated. Observe though that these datasets have been collected running the vehicle as a series hybrid, utilizing the supercapacitor, thus not being restricted by the maximum torque or power of the engine or generator. The sequence is presented in Figure 7.

4.2 Collection of evaluation data

The main evaluation of the developed controllers is based on data collected in field tests. The operation performed in the tests are intended to represent real world operation, and especially be the result of driving performed by real drivers in a realistic external environment. For this reason, a specific site was made available for the tests.

The site consisted of a pile of coarse gravel positioned next to a ramp, and an articulated hauler capable of carrying 4-5 bucket loads. The hauler was

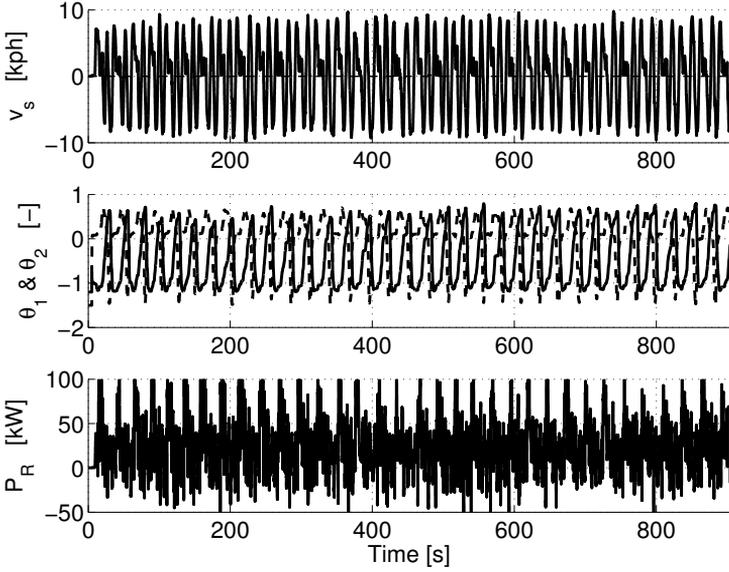


Figure 6: Previously available data, derived from data collected using a production Volvo L180E machine

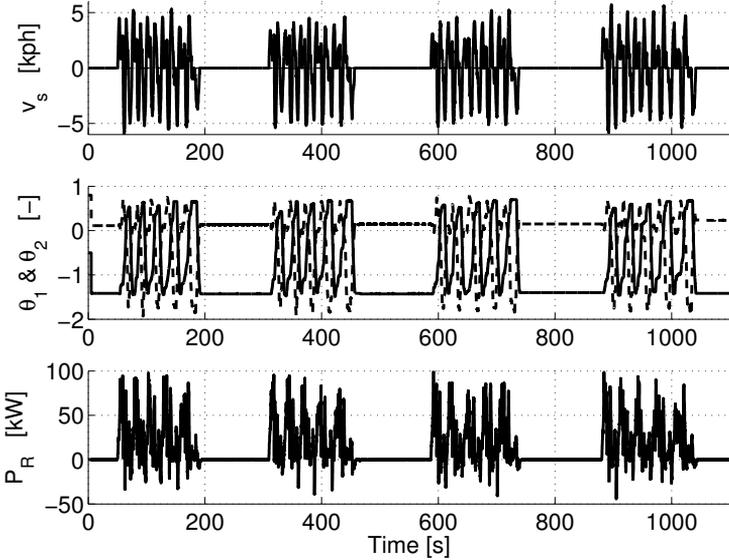


Figure 7: Previously available data, data collected using the series hybrid machine

positioned next to the pile, similar to the layout in Figure 2. When the hauler had been loaded with 4-5 bucket loads, it moved around the site onto the ramp, emptying its load back onto the original pile. This way, a realistic sequence

of similar cycles could be performed. The unusually long pauses between each subset of cycles are of less importance since the predictive controller is only to be used in active cycle driving, and the pauses can easily be detected and handled. The evaluation therefore includes all of the relevant disturbances or uncertainties imposed by the driver and the environment along with the model errors.

5 Results

5.1 Optimization

To make the evaluation as fair as possible, identical state and control signal discretizations were used in the $p(P_R|s)$ and $p(P_R|s)$ controllers. The leg sizes in the later was set to $4 \times 8m$. The experienced calculation times, using a standard desktop computer, have been in the order of $10s$ per iteration for the $p(P_R|s)$ method, with about 25 iterations required before the solution ceases to change, and in the order of $1h$ per iteration for the $p(P_R|s)$ method, with 2 iterations before the solution ceases to change. The results are cost to go maps $J(x)$, and a final iteration is required for finding the corresponding control maps $\hat{u}(x)$, making a total calculation time of about $5m$ for the $p(P_R|s)$ concept and $4h$ for the $p(P_R|s)$ concept.

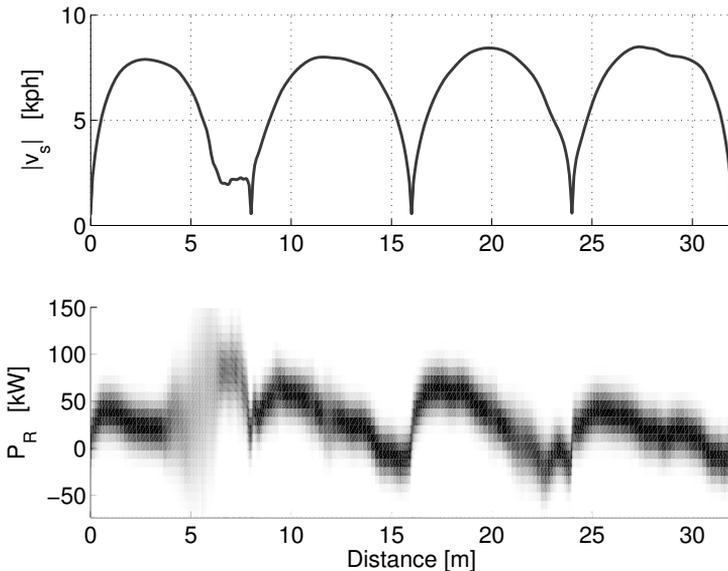


Figure 8: The probability function $p(P_R|s)$ used in the optimization for the predictive controller.

The velocity profile and probability function used in the optimization for the predictive controller are presented in Figure 8. The corresponding probability function $p(P_R|s)$ can be constructed from the time average of the probabilities

in Figure 8. The $\hat{u}(x)$ maps used in the $p(P_R| -)$ controller are presented in Figure 9 and slices, taken at $s \approx 5m$, of the maps used in the $p(P_R|s)$ controller are presented in Figure 10. As these figures show, there are switching regions in these maps, where linear interpolation might not produce the same results as a solving of (11), and nearest neighbor interpolation might be more adequate, as discussed in Section 2.6. Since the main result of using nearest neighbor interpolation was highly oscillatory control signals and a simple solution was required for the online implementations, linear interpolation was used throughout the $\hat{u}(x)$ maps. Figure 10 also indicates that the discontinuities in the $d\omega_e$ map are related to the discretization of P_{max} .

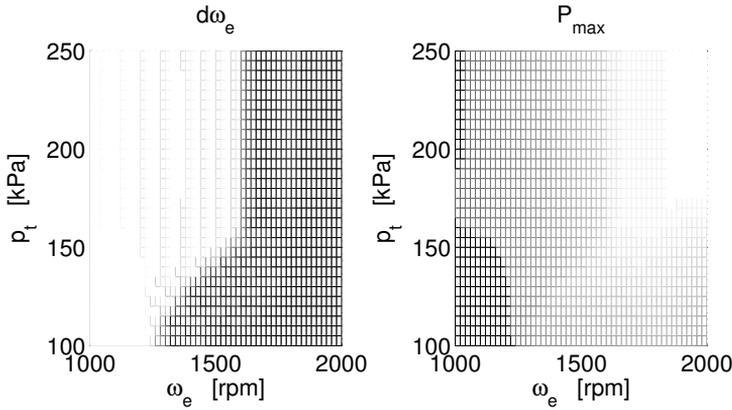


Figure 9: The optimized control maps $d\omega_e/dt$ and P_{max} for the $p(P| -)$ controller. Darker shades indicate lower values.

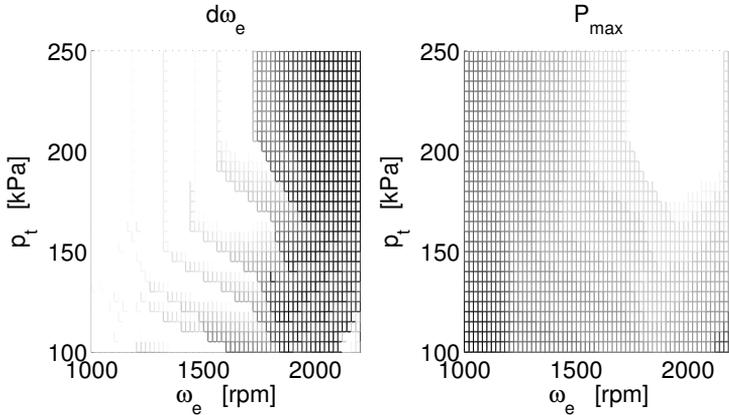


Figure 10: The optimized control maps $d\omega_e/dt$ and P_{max} for the $p(P|s)$ controller. In both figures darker shades indicate lower values. It can be seen that the dark bands in $d\omega_e/dt$ correspond to the changes in the levels in P_{max} .

5.2 Simulation results

The optimized control maps were first evaluated using simple simulations. An example of data from a simulation of a cycle from the L180E dataset using the $p(P_R|s)$ controller is presented in Figure 11. This figure shows the states ω_e and p_t , the requested power P_R and the control signal P_{max} . The control signal $d\omega_e/dt$ can be found from the ω_e figure, since there are no model errors in these simulations. In this cycle the leg sizes are set to $4 \times 8m$ and the leg limits are marked with gray lines. The fuel data from this cycle with the three controllers, described in Section 3, are presented in Table 1. The values are fuel use, efficiency η , which is fuel use divided by actual output power P_L , and energy not delivered, defined by $W_U = \int(P_R - P_L)$. The W_U value is now a measure for how big impact P_{max} has had on the operation, and by adjusting the parameters in the $G_{P_{max}}$ functions fuel can be traded for W_U . In these cycles, the fuel values are similar, but the energy not delivered falls with increasing controller complexity.

Table 1: Controller performance in the initial simulations.

	<i>const</i>	$p(P_R -)$	$p(P_R s)$
M_f [g]	60.9	63.1	60.9
η [%]	27.8	27.2	28.5
W_U [kWs]	61.3	50.1	40.1

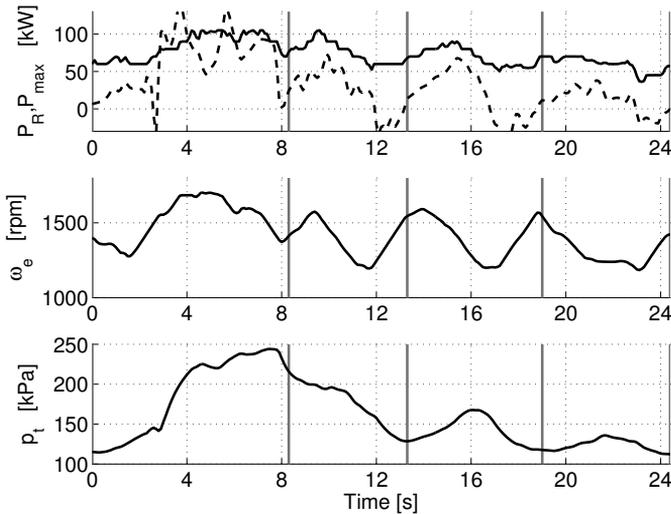


Figure 11: Illustration of the results from the initial simulations using the $p(P_R|s)$ -controller. The dashed line is P_R .

The controllers were further evaluated using Simulink simulations. Two examples of data from such simulations, using the $p(P_R|-)$ and $p(P_R|s)$ controllers, are presented in Figure 12. The example is a cutout from simulations using the series hybrid machine data presented in Figure 7. The driving direc-

tion changes are marked with gray lines. It can be seen that, despite the leg sizes being shorter than in the optimization data (the leg sizes are presented in Figure 13), the engine speed varies with the load, specially in the $p(P_R|s)$ case, as intended. The average fuel data from all cycles in both the L180E data (data 1) and the series hybrid data (data 2) simulations with the three controllers, described in Section 3, are presented in Table 2. The values are fuel use, efficiency η , which is fuel use divided by actual output power P_L , and energy not delivered, defined by $W_U = \int (P_R - P_L)$. In both of the datasets the fuel consumption values are similar although slightly lower for the $p(P_R|s)$ controller, but the undelivered energies differ. In the second case W_U is bigger for the $p(P_R|s)$ controller, and this is due to the legs being shorter in this dataset, thus leading to a mismatch between the locations of the predicted and simulated high power regions at the end of the first leg. In the L180E data, in which the leg sizes are closer to those in the optimization data, the W_U value is instead significantly smaller for the $p(P_R|s)$ controller.

Table 2: Controller performance in the Simulink simulations.

Data 1	<i>const</i>	$p(P_R -)$	$p(P_R s)$
M_f [g]	104.2	106.7	100.5
η [%]	23.7	23.1	25.1
W_U [kW s]	117.1	117.5	90.6
Data 2	<i>const</i>	$p(P_R -)$	$p(P_R s)$
M_f [g]	69.2	70.6	67.2
η [%]	24.1	23.6	24.4
W_U [kW s]	29.8	25.8	37.6

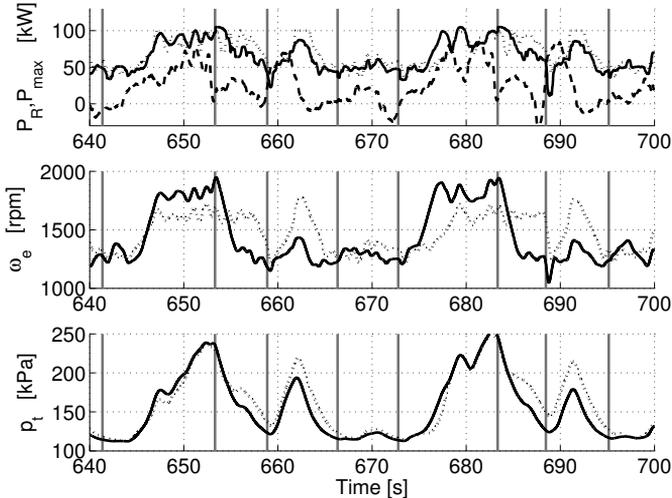


Figure 12: Illustration of the results from the Simulink simulations using the $p(P_R|-)$ (dotted) and $p(P_R|s)$ (solid) controllers in a cycle from the L180E data. The dashed line is P_R .

5.3 Field test results

The controller field tests were performed with three different drivers, each operating the machine for about an hour, producing a total of about twenty cycles using each of the three controllers.

Figure 13 shows the average leg sizes in the optimization, simulation and measurement datasets. The data is divided into the four legs, with the first leg at the bottom. This figure shows that the distances in the measurements are similar to those in the optimization data but that there are significant and realistic differences.

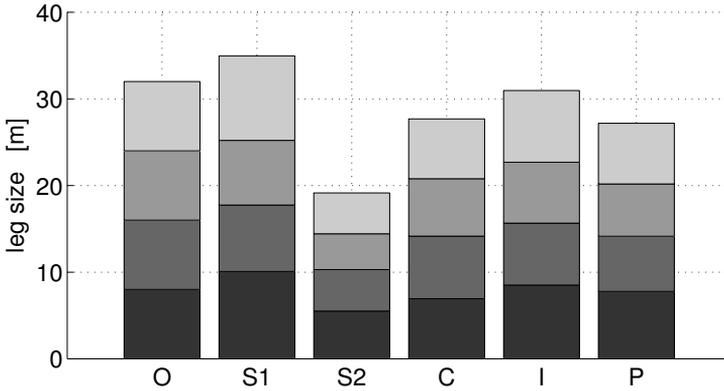


Figure 13: Average leg sizes in the datasets. Designations: O=optimization (adjusted L180E), S1=unadjusted L180E, S2=series hybrid, C=const, I= $p(P_R|-)$, P= $p(P_R|s)$.

The Figures 14, 15 and 16 each shows one short loading cycle and illustrates the data collected in the measurements, using the constant engine speed, $p(P_R|-)$ and $p(P_R|s)$ controllers, respectively. The first part shows vehicle speed v_s , which also gives the distance driven and especially the active leg. The second part shows maximum and requested power, P_{max} (solid) and P_L (dashed). Observe that unlike in the simulations, where the P_R signal was available, only the P_L signal which is an output from the machine balance controller is available, and defining a measure of the impact of P_{max} , such as W_U , is therefore nontrivial. The third part shows engine speed ω_e and turbo pressure p_t , where p_t is replaced by the state of charge in Figure 14. The vehicle speeds and desired powers are similar in these three cycles, though v_s is somewhat lower in the $p(P_R|s)$ cycle. The pause early in leg four in Figure 16 is an impact of the driver having to push the cycle detection button. The control computer was mounted beside the driver, so that he had to take his eyes and concentration off the driving to press the cycle detection button. Such pauses in or before the last leg were therefore common when operating with the $p(P_R|s)$ controller. P_{max} mainly follows ω_e but is also affected by p_t . The pressure p_t acts as a low pass filtered combination of the P_L and ω_e . The state of charge does start on a level off target in this example and has some ripple which is caused by

generator model errors. Initial offsets were common since one driver switched to a backup controller in the longer pauses, and the level of the ripple was similar in each measured cycle. The biggest difference between the outputs of the three controllers are in the engine speeds. In the first case, the speed should be constant. As Figure 14 shows, it varies, but this is to a lesser extent than in the other two cases and the variations can be regarded as random. In the second case, ω_e is controlled, but only from the actual ω_e and p_t . This can best be seen at 490 – 495s, where ω_e follows p_t , which in turn follows P_L . In the third case, ω_e increases mainly in the end of the first leg since there is a high probability for high P_R at that position, and there is less correlation between P_L and the engine speed.

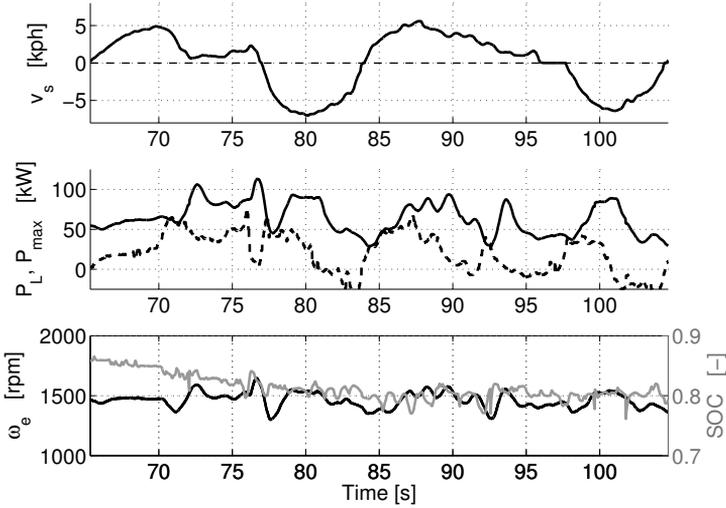


Figure 14: An example, from the measurement results, of a cycle operated with the constant engine speed reference controller.

Table 3 presents average data from the measurements. The values are fuel use per cycle, electric energy consumed including change of supercapacitor state of charge, efficiency η (fuel use divided by actual output power P_L), leg size, cycle time and number of cycles of each type. There are about 20 cycles of each type, which is few but still enough to make a first analysis. The cycle times are somewhat higher when using the optimized controllers, though the leg sizes are not. The energy consumed varies due to small differences in the operation, and are somewhat higher when using the optimized controllers. The energy from the Δ SOC is of course included when calculating the fuel to output power efficiency η . In the simulations, the P_R trajectory was the same for the controllers so that the fuel per cycle and efficiency are near equivalent. This is not the case in the field test data, and selecting a proper fuel measure for the controller comparison is not trivial. The fuel per cycle values are similar under the *const* and $p(P_R|s)$ controllers, and higher under the $p(P_R|-)$ controller. Due to the differences in cycle times and distances driven, partly caused by the pauses for pressing

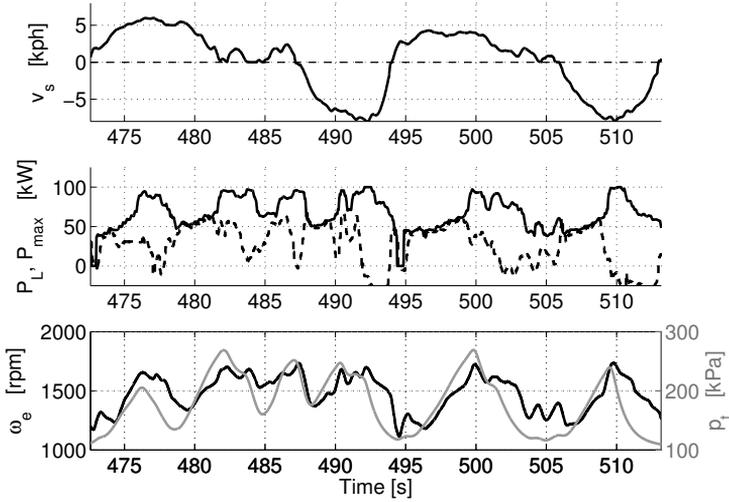


Figure 15: An example, from the measurement results, of a cycle operated with the $p(P_R|-)$ controller.

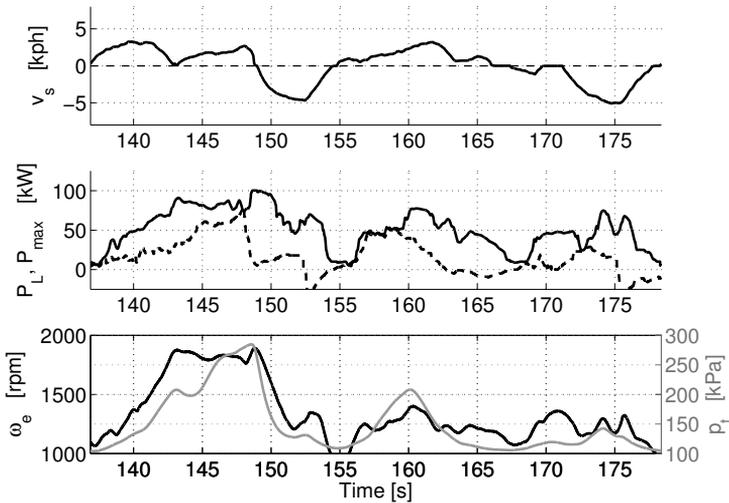


Figure 16: An example, from the measurement results, of a cycle operated with the $p(P_R|s)$ controller.

the cycle detection button in the $p(P_R|s)$ cycles, the fuel per cycle values is not an appropriate measure for a controller comparison. The most appropriate measure is instead the efficiency values η , which includes the differences in used energy and changes in supercapacitor state of charge. The efficiency increases slightly with the complexity of the controller, so that there is a 3 – 4% relative improvement when using the $p(P_R|s)$ controller, compared to the other two. Since the P_R signal is not available, a proper comparison of the impact of P_{max}

such as through W_U , is not trivial. However, since P_L and P_{max} only coincide when the power limiter is active, a comparison of these two signals, as can be done in Figures 14, 15 and 16, can give an indication of the impact of P_{max} . Such comparisons indicate similar impacts in the two optimized controllers and slightly lower impact in the constant engine speed controller. Observe though that this only indicate the frequency, and not the magnitude, of P_{max} having an impact.

Table 3: Controller performance in the measurements.

	<i>const</i>	$p(P_R -)$	$p(P_R s)$
M_f [g]	93.2	101.8	91.8
$\int P_L dt$ [kWs]	779	856	794
η [%]	19.4	19.5	20.2
avg leg size [m]	6.92	7.74	6.80
avg cycle time [s]	40.5	43.5	44.3
no. of cycles [-]	21	20	26

6 Discussion and Conclusions

The aim of the work presented in this paper was to investigate the possible benefit of utilizing the repetitiveness of wheel loader operation for optimal predictive control of of CVT based transmission in such a vehicle. The predictive controller was constructed using stochastic dynamic programming with a load power probability distribution dependent on distance driven, $p(P_R|s)$. This predictive controller was evaluated through both simulations and field tests, in which the machine was operated with the predictive and two less complicated controllers. The combined results measured in fuel use M_f and impact of the power restriction W_U , in simulations and measurements, show a slight advantage for the predictive $p(P_R|s)$ controller. In simulations the $p(P_R|s)$ controller in general gave both lower M_f and W_U values. The exception was when cycles with shorter legs were simulated. In this case the $p(P_R|s)$ controller gave higher W_U values, since the high power bucket filling occurs at a position that differs from that in the optimization data. In the measurements, the W_U value cannot easily be defined, and the cycles recorded when using the different controllers differ somewhat in both duration and average power. Due to the differences, the most proper fuel value to compare is the fuel to output power efficiency. This value indicate a fuel benefit of 3 – 4% of the $p(P_R|s)$ controller over the reference controllers, which agree well with values seen in simulations. The differences between the cycles from the different controllers are partly caused by the distraction of having to press a cycle detection button in the $p(P_R|s)$ controller, but are mainly differences that occur normally because of the human operator.

An important part of this evaluation, and the part that distinguishes this work from the previous paper [1] on which the controllers are based, is the introduction of real world disturbances. These include a real environment and,

most importantly, human drivers. All of these disturbances are present in the measurements. The machine operates on a realistic work site and there were three different drivers operating the machine. Despite severe model errors, especially in the lack of a model for the auxiliary loads, the main controller studied ($p(P_R|s)$) produced the expected general state trajectories, increasing the engine speed in anticipation of the high power during bucket filling and lowering of the speed during the rest of the cycle.

In all, this paper shows that the conceptual predictive controller can be used for controlling the genset in a diesel electric wheel loader. Tests are performed in which all relevant disturbances are present. The fuel improvement in simulations and field tests, based on the fuel to output power efficiency, is in the order of 3 – 4%, as compared to the results from the reference controllers.

References

- [1] T. Nilsson, A. Fröberg, and J. Åslund. Using stochastic dynamic programming for look-ahead control of a wheel loader diesel electric transmission. In *IFAC World Congress*, pages 6630–6635. IFAC, 2014.
- [2] G. Stein, A. Fröberg, J. Martinsson, B. Brattberg, R. Filla, and J. Unneback. Fuel efficiency in construction equipment - optimize the machine as one system. In *7th AVL International Commercial Powertrain Conference*. AVL & SAE, 2013.
- [3] T. Nilsson, A. Fröberg, and J. Åslund. Fuel potential and prediction sensitivity of a power-split cvt in a wheel loader. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, pages 49–56. IFAC, 2012.
- [4] J. Lennevi. *Hydrostatic Transmission Control, Design Methodology for Vehicular Drivetrain Applications*. dissertation, Linköping University, 1995.
- [5] K-E. Rydberg. Hydrostatic drives in heavy mobile machinery - new concepts and development trends. In *International Off-Highway & Powerplant Congress & Exposition*, number 981989. SAE, 1998.
- [6] R. Zhang, A. Alleyne, and E. Prasetyawan. Modeling and h_2/h_∞ mimo control of an earthmoving vehicle powertrain. *Journal of dynamic systems, measurement, and control*, 124(4):625–636, 2002.
- [7] R. Kumar, M. Ivantysynova, and K. Williams. Study of energetic characteristics in power split drives for on highway trucks and wheel loaders. In *SAE technical papers*, number 2007-01-4193. SAE, 2007.
- [8] R. Filla. Alternative systems solutions for wheel loaders and other construction equipment. In *1st International CTI Forum Alternative and Hybrid Drive Trains*. CTI, 2008.

- [9] T. Lin, Q. Wang, B. Hu, and W. Gong. Development of hybrid powered hydraulic construction machinery. *Automation in Construction*, 19(1):11–19, 2010.
- [10] B. Frank, L. Skogh, and M. Alaküla. On wheel loader fuel efficiency difference due to operator behaviour distribution. In *2nd International Commercial Vehicle Technology Symposium*, pages 329–346, 2012.
- [11] S. Liu and B. Paden. A survey of today’s cvt controls. In *Proceedings of the 36th Conference on Decision and Control*, pages 4738–4743. IEEE, 1997.
- [12] R. Pfiffner. *Optimal Operation of CVT-Based Powertrains*. dissertation, ETH, Zurich, 2001.
- [13] T. Nilsson, A. Fröberg, and J. Åslund. Optimal operation of a turbocharged diesel engine during transients. *SAE International Journal of Engines*, 5(2):571–578, 2012.
- [14] M. Sivertsson and L. Eriksson. Time and fuel optimal power response of a diesel-electric powertrain. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, pages 262–269. IFAC, 2012.
- [15] A. Sciarretta and L. Guzzella. Control of hybrid electric vehicles. *Control Systems, IEEE*, 27:60–70, 2007.
- [16] E. Hellström. *Look-ahead Control of Heavy Vehicles*. dissertation, Linköping University, 2010.
- [17] H. Khayyam, S. Nahavandi, and S. Davis. Adaptive cruise control look-ahead system for energy management of vehicles. *Expert Systems with Applications*, 39(3):3874–3885, 2012.
- [18] C. Lin, S. Jeon, H. Peng, and J. Moo. Driving pattern recognition for control of hybrid electric trucks. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 42(1-2):41–58, 2004.
- [19] T. Nilsson, C. Sundström, P. Nyberg, E. Frisk, and M. Krysander. Robust driving pattern detection and identification with a wheel loader application. *International Journal of Vehicle Systems Modelling and Testing*, 9(1):56–76, 2014.
- [20] L. Johannesson, M. Åsbogård, and B. Egardt. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):71–83, 2007.
- [21] I. Kolmanovsky and D. Filev. Terrain and traffic optimized vehicle speed control. In *6th IFAC Symposium on Advances in Automotive Control*, pages 378–383. IFAC, 2010.

- [22] T. Leroy, J. Malaize, and G. Corde. Towards real-time optimal energy management of hev powertrains using stochastic dynamic programming. In *8th IEEE Vehicle Power and Propulsion Conference*, pages 383–388. IEEE, 2012.
- [23] K. McDonough, I. Kolmanovsky, D. Filev, D. Yanakiev, S. Szwabowski, and J. Michelini. Stochastic dynamic programming control policies for fuel efficient in-traffic driving. In *American Control Conference (ACC), 2012*, pages 3986–3991. IEEE, 2012.
- [24] R. Filla. *Quantifying Operability of Working Machines*. dissertation, Linköping University, 2011.
- [25] F. Wang, J. Zhang, R. Sun, and F. Yu. Analysis on the performance of wheel loaders in typical work cycle. *Applied Mechanics and Materials*, 148:526–529, 2012.
- [26] V. Nezhadali and L. Eriksson. Modeling and optimal control of a wheel loader in the lift-transport section of the short loading cycle. In *7th IFAC Symposium on Advances in Automotive Control*, pages 195–200. IFAC, 2013.
- [27] Volvo Construction Equipment. Product history, 2006. <http://www.volvoce.com/CONSTRUCTIONEQUIPMENT/CORPORATE/EN-GB/ABOUTUS/HISTORY/PRODUCTS/WHEEL%20LOADERS/WHEEL%20LOADERS%20VOLVO/Pages/introduction.aspx>.
- [28] G. Rizzoni, L. Guzzella, and B.M. Baumann. Unified modeling of hybrid electric vehicle drivetrains. *IEEE/ASME Transactions on Mechatronics*, 4:246–257, 1999.
- [29] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3 edition, 2005.
- [30] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons Inc, Hoboken, NJ, USA, 2 edition, 2005.

Linköping studies in science and technology. Dissertations.
Division of Vehicular Systems
Department of Electrical Engineering
Linköping University

- No. 1 Magnus Pettersson *Driveline Modeling and Control*, 1997
- No. 2 Lars Eriksson *Spark Advance Modeling and Control*, 1999
- No. 3 Mattias Nyberg *Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications*, 1999
- No. 4 Erik Frisk *Residual Generation for Fault Diagnosis*, 2001
- No. 5 Per Andersson *Air Charge Estimation in Turbocharged Spark Ignition Engines*, 2005
- No. 6 Mattias Krysanter *Design and Analysis of Diagnosis Systems Using Structural Methods*, 2006
- No. 7 Jonas Biteus *Fault Isolation in Distributed Embedded Systems*, 2007
- No. 8 Ylva Nilsson *Modelling for Fuel Optimal Control of a Variable Compression Engine*, 2007
- No. 9 Markus Klein *Single-Zone Cylinder Pressure Modeling and Estimation for Heat Release Analysis of SI Engines*, 2007
- No. 10 Anders Fröberg *Efficient Simulation and Optimal Control for Vehicle Propulsion*, 2008
- No. 11 Per Öberg *A DAE Formulation for Multi-Zone Thermodynamic Models and its Application to CVCP Engines*, 2009
- No. 12 Johan Wahlström *Control of EGR and VGT for Emission Control and Pumping Work Minimization in Diesel Engines*, 2009
- No. 13 Anna Pernestål *Probabilistic Fault Diagnosis with Automotive Applications*, 2009
- No. 14 Erik Hellström *Look-ahead Control of Heavy Vehicles*, 2010
- No. 15 Erik Höckerdal *Model Error Compensation in ODE and DAE Estimators with Automotive Engine Applications*, 2011
- No. 16 Carl Svärd, *Methods for Automated Design of Fault Detection and Isolation Systems with Automotive Applications*, 2012
- No. 17 Oscar Leufven, *Modeling for control of centrifugal compressors*, 2013
- No. 18 Cristofer Sundström, *Model Based Vehicle Level Diagnosis for Hybrid Electric Vehicles*, 2014

- No. 19** Andreas Thomasson, *Modeling and control of actuators and co-surge in turbocharged engines*, 2014
- No. 20** Emil Larsson, *Model Based Diagnosis and Supervision of Industrial Gas Turbines*, 2014
- No. 21** Andreas Myklebust, *Dry Clutch Modeling, Estimation, and Control*, 2014