Development of Look-Ahead Controller Concepts for a Wheel Loader Application

Tomas Nilsson, Anders Fröberg, Jan Åslund

To cite this version:


HAL Id: hal-01931293
https://hal.archives-ouvertes.fr/hal-01931293
Submitted on 22 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
This paper is a part of the hereunder thematic dossier published in OGST Journal, Vol. 70, No. 1, pp. 3-211 and available online here.

Cet article fait partie du dossier thématique ci-dessous publié dans la revue OGST, Vol. 70, n°1, pp. 3-211 et téléchargeable ici.

IFP Energies nouvelles International Conference / Les Rencontres Scientifiques d’IFP Energies nouvelles E-COSM’12 — IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling E-COSM’12 — Séminaire de l’IFAC sur le contrôle, la simulation et la modélisation des moteurs et groupes moto-propulseurs

Oil & Gas Science and Technology – Rev. IFP Energies nouvelles, Vol. 70 (2015), No. 1, pp. 3-211
Copyright © 2015, IFP Energies nouvelles
Development of Look-Ahead Controller Concepts for a Wheel Loader Application

Tomas Nilsson¹*, Anders Fröberg² and Jan Åslund¹

¹ Department of Electrical Engineering, Linköping University - Sweden
² Volvo Construction Equipment, Eskilstuna - Sweden

e-mail: tnilsson@isy.liu.se - anders.froberg@volvo.com - jaasl@isy.liu.se

* Corresponding author

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 (SDP), uses a statistical load prediction and stochastic dynamic programming for minimizing fuel use. The second method developed, designated Free-Time Dynamic Programming (FTDP), has vehicle speed as a state and introduces a fixed 0.1 s 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' manner.

The evaluation shows that the SDP method requires about 1/10th 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.

Résumé — Développement de concepts d'une commande prédictive, destinée à une application pour chargeur sur pneus — Ce document présente deux méthodes de conception, basées sur la programmation dynamique, pour la commande à un pas de prédiction d’une transmission continûment variable (Continuously Variable Transmission, CVT) d’un chargeur sur pneus. La première méthode développée, appelée programmation dynamique stochastique (Stochastic Dynamic Programming, SDP) utilise une prédiction statistique de la charge et la programmation dynamique stochastique pour minimiser l’utilisation de carburant. La seconde méthode développée, appelée programmation dynamique à temps libre (Free-Time Dynamic Programming, FTDP), établit la vitesse du véhicule en tant qu’état et introduit un retard de 0,1 s dans les commandes du godet pour minimiser à la fois l’utilisation de carburant et le temps nécessaire à l’opération.

Les méthodes sont évaluées en s’appuyant sur 34 cycles de chargement mesurés, utilisés selon la méthode de validation croisée « leave-one-out ». L’évaluation montre que la méthode SDP requiert environ 1 dixième de l’effort de calcul de la méthode FTDP, et qu’elle a un impact plus transparent sur les écarts dans la prédiction du cycle. D’un autre côté, avec la méthode FTDP on obtient une réduction de 10 % de la consommation de carburant, ce qui est proche de l’optimum réel, pour les mêmes durées de cycle, et elle permet de réaliser une plus grande partie des cycles d’évaluation.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
INTRODUCTION

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.

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.

On the Choice of a Hydraulic Multi-Mode CVT

Any alternative transmission has to enable increased efficiency in the typical operation conditions mentioned. The low speeds at which the machine often operates 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, 4], in which some part of the power is mechanically transmitted. Multi-mode Continuously Variable Transmissions (CVT) 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.

CVT Control in a Wheel Loader

The introduction of a CVT increases both the possibility of 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 [6, 7]. The focus is often on actuator control though, and there is a lack of work on higher level control, including the choice of the 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, 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 [10-12]. In the wheel loader application, the potential benefit has been explored [13], but so far there has been no implemented look-ahead controller for wheel loaders. The main difficulties, as compared with on-road applications, 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.

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.
1 MODELS

1.1 Machine Operation

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

In the SLC definition used here, and referring to the 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 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 30 s and the distances between the driving direction changes are around 10 m.

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 $\theta_1$ and $\theta_2$. 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 $\omega_{ct}$ 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 torque converter output torque $T_{ct}$ and the selected gear $r_c$.

$F_w = \text{sign}(v_w) mg c_r$ (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.

---

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

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.
One of the measured SLC, 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 26.5 s and the average unadjusted distance driven is 35 m.

1.2 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:

\[
\frac{dv_w}{dt} \cdot m = r^{-1}T_W - r^{-1}T_b - F_w
\]

(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.

1.3 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 \( T_H \):

\[
\frac{d\omega_e}{dt} \cdot I_e = T_e - T_T - T_H
\]

(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:

\[
T_e = e(\omega_e, m_f) \cdot \frac{q_{inj}n_{cyl}}{2\pi n_r} \cdot m_f - T_L(\omega_e) - T_{pt}
\]

(5)

in which \( m_f \) is fuel mass per injection, \( \omega_e \) is engine speed, \( e \) and \( T_L \) are efficiency functions, \( q_{inj} \), \( n_{cyl} \) 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:

\[
\frac{dp_t}{dt} \cdot \tau(\omega_e) = -p_{off}(\omega_e, m_f)
\]

(6)
and the torque loss from low pressure is described by:

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

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

\[
\frac{dM_f}{dt} = m_f \frac{n_{cyl}}{2\pi n_r} o_e \quad (8)
\]

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.

1.4 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, 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 are 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 the 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]:

\[
\psi_1 D_v \omega_1 \pm p_v (C_a + (\omega_1 + \omega_2) C_b) - \psi_2 D_v \omega_2 = C_v p_v \quad (9)
\]

\[
\psi_n D_v p_v - T_n \pm (C_v \omega_n + C_d p_v) = 0 \quad (10)
\]

The index \(n = 1, 2\) denotes the two machines, \(D_v\) is maximum displacement, \(\psi_n \in (0, 1)\) is relative displacement, \(\omega_n\) 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 (11) describes hydraulic fluid flow and Equation (12) describes torque in each machine. The variator is constructed so that \(\psi_1 + \psi_2 = 1\). The variator pressure dynamics is assumed to be fast compared with other dynamics of the system, i.e. it is assumed that the time constant \(C_v p_v\) 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:
\[ T_T(m_T, \psi_1, \omega_e, \nu_w) \]  
(11)
\[ T_W(m_T, \psi_1, \omega_e, \nu_w) \]  
(12)

1.5 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:
\[ Q_H = \psi_H D_H \omega_e \]  
(13)
\[ Q_H \eta_H = \eta_H T_H \omega_e \]  
(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.

2 METHODS

2.1 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 \( k = 0, \ldots, N - 1 \), 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:
\[ \min_{u \in U} E \{ J_N(x_N) + \sum_{k=0}^{N-1} g_k(u_k, x_k, w_k) \} \]
\[ x_{k+1} = f(x_k, u_k, t), \quad k = 0, \ldots, N - 1 \]  
(15)
along with equality and inequality constraints. According to [17, 18] the dynamic programming recursion can, for this problem, be stated as:
\[ J_k(x_k) = \min_{u \in U} E \{ g(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k)) \} \]  
(16)
\[ J_N(x_N) = g_N(x_N) \]  
(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 \( k = N - 1, \ldots, 1 \) do
3. For each \( x_k \in X_k \), simulate \( \frac{dx_k}{dt} \) 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 \( x_k \in X_k \)
\[ J_k(x_k) = \min_{u \in U} \sum_{w \in W_k} p(w_k)(g(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k))) \]  
(18)

with \( J_{k+1}(x_{k+1}(x_k, u_k, w_k)) \) interpolated from \( J_{k+1}(x_{k+1}(x_k, u_k, w_k)) \)
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:
\[ J_k(x_k) = \min_{u_l \in U} \sum_{w \in W_k} p(w_k)(g(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k))) \]
(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 \( m = 1, \ldots, N \) do
3. For \( x_{m-1}^* \), simulate \( \frac{dx_m}{dt} \) for \( s_{m-1} \) to \( s_m \) for all \( u \in U \) to find \( x_m(x_{m-1}^*, u) \)
4. Select
\[ u_{m-1}^* = \arg\min_{u \in U} (g(x_{m-1}^*, u, w_{m-1})) + J_m(x_m(x_{m-1}^*, u, w_{m-1})) \]  
(20)
in which \( J_m(x_m) \) 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

2.2 Dynamic Programming as a One-Step Look-Ahead Controller

The second part of the algorithm presented in the previous section can be seen as a one-step look-ahead simulation. In this case, the load \( w_k \), \( k = 0, \ldots, N - 1 \) used in the second part is the actual load, which will differ from the load used in the CTG map calculation, unless there is a perfect prediction of future loads. This type of control, assuming a perfect but limited horizon prediction,
is used [10 and 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 2.3 and 2.4.

This section discusses the implication of uncertainties, and the impact of disturbances, in each load component as compared with 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 (4).

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 (4).

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 the bucket lifting speed, so the bucket inertia should limit \( \frac{\Delta Q_H}{\Delta t} \). 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 \( \omega_e \), according to Equation (13). It is not uncommon that \( \omega_e > \omega_e \), 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 (4).

The engine torque can be altered instantaneously, though the turbo speed may restrict the magnitude of the change. The component \( Q_H \) causes a limitation that is often active, and uncertainty 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:

\[
Q_H = \frac{dV_H}{dt} = \frac{dV_H}{ds} v_s = \psi_H(t) D_H \omega_e(t)
\]

which if \( \frac{\Delta Q_H}{\Delta t} \) becomes high enough requires \( \psi_H > 1 \). Since this is not allowed, other solutions must be found. Since \( \psi_H \) is limited, the alternatives identified are to introduce margins through \( \omega_e \) and \( v_s \), 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 \( \omega_e \) and \( v_s \) can be seen as the cause of the problem. An instantaneous increase in \( Q_H \) would require an instant increase in \( \omega_e \) or decrease in \( v_s \), both of which are prevented by their inertias. The first alternative is therefore to keep \( \omega_e \), as a function of \( v_s \), at such a level that \( \psi_H \) 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 in 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 2.3.

In an actual vehicle, deviating from the desired bucket trajectory is a natural response to an unachieviable 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. Deviating 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 undesirable 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 2.4.

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.

### 2.3 Stochastic Dynamic Programming

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

#### 2.3.1 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 including a worst-case scenario with a low probability, the CTG map will correspond to a minimization of the cost under the condition that the vehicle must always be able to handle the worst case future load. This method is designated the Stochastic Dynamic Programming (SDP), method.

#### 2.3.2 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:

\[
\min E\{M_f(T)\} \tag{22}
\]

and the cost function therefore becomes:

\[
g(x_k, u_k, w_k) = \sum_{w_l \in W_k} (p(w_l) \frac{dM_f}{dt}) \tag{23}
\]

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 \( \omega_c(\psi_1) \) is always invertible for this concept either \( \omega_c \) or \( \psi \), along with \( m_T \), can be used as state. Since the speed will increase for one of the hydraulic machines when \( \psi_1 \) 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 \( \psi_1 \), which implies using \( \psi_1 \) as state. The possibility of restrictions on \( \frac{d\psi_1}{dt} \), especially during mode shifts, also points toward using \( \psi_1 \) as state. Since the dynamics are described in terms of \( \omega_c \), this would imply the following computational scheme:

\[
\psi_{1,k} \stackrel{W_k}{\rightarrow} \omega_{e,k} \stackrel{\frac{d\omega_c}{dt}}{\rightarrow} \omega_{e,k+1} \stackrel{W_k}{\rightarrow} \psi_{1,k+1}
\]

In the first and last steps, the load is required, since \( \omega_c(\psi_1) \) 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 (4) from \( \frac{d\omega_c}{dt} \) to \( \frac{d\psi_1}{dt} \). This choice of \( \kappa \) does not guarantee continuity in \( \omega_c \), 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 \( \omega_c \) 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 \( \psi_1 \) as a state, and instead \( \omega_c \) is used. \( \omega_c(\psi, m_T) \) 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 gives the 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 \( \omega_c \) 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>
<thead>
<tr>
<th>Flow</th>
<th>States</th>
<th>Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>( \omega_c, p_t )</td>
<td>( m_f )</td>
</tr>
</tbody>
</table>

#### 2.3.3 Load Case Creation for the SDP Method

Using SDP in look-ahead control applications has been studied [21, 24]. 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 1.1, 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 $\omega_w$ as a Markov process is deemed unrealistic, as discussed in Section 2.2, 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 10 s for the forward and loading phases, and 5 s 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 $\mu$ and standard deviation $\sigma$ 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 $\mu$ and $\sigma$.

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.

### 2.4 Free-Time Dynamic Programming

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

<table>
<thead>
<tr>
<th>$\omega_w$</th>
<th>$T_w$</th>
<th>$Q_H$</th>
<th>$p_H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$ (1)</td>
<td>$\mu - \sigma$ (.25)</td>
<td>$\mu - \sigma$ (.25)</td>
<td>$\mu - \sigma$ (.25)</td>
</tr>
<tr>
<td>$\mu$ (5)</td>
<td>$\mu$ (5)</td>
<td>$\mu$ (5)</td>
<td></td>
</tr>
<tr>
<td>$\mu + \sigma$ (.25)</td>
<td>$\mu + \sigma$ (.25)</td>
<td>$\mu + \sigma$ (.25)</td>
<td></td>
</tr>
<tr>
<td>$\mu + 2\sigma$ (.05)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2.4.1 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 (FTDP) method.

### 2.4.2 Implementation

#### CTG Map Calculation

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

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

and the cost function therefore becomes:

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

in which, introducing the vehicle speed $v_k = |v_w|$ and distance driven $s = \int v_k dt$, the time steps $\Delta t = \Delta s/v_k$ or, if $v_k \approx 0$, $\Delta t = 2\Delta s/(v_{k,k} + v_{k,k+1})$, 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:

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

$$\frac{dx}{ds} = \frac{1}{v_k} f(x(s), u(s), w(s))$$
During the general driving cycle, the vehicle changes driving direction several times. In these instances the vehicle speed $v_s$ has to go to zero. The state derivatives will then, according to Equation (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 2.5. Similar state changes would not solve the problem for the engine speed and turbo pressure dynamics, though. Hence the approximation:

$$As = \bar{v}_s At, \quad \bar{v}_s = \frac{v_{s,k} + v_{s,k+1}}{2}$$

is used instead 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 ensure 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:

$$\min \int_0^{s_f} \left( \frac{dM_T}{dt} + \beta \right) \frac{ds}{v_s}$$

(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 $\omega_e$ 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 $\psi_1$ 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 in these instances $J(s_m, v_s > 0) = \infty$. For calculation effort reasons, zero speed is not allowed in 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 $m_f, \psi_1$. The gain from the variator ratio $\psi_1$ to the torques $T_T$ and $T_w$, according to the functions (11) and (12), is very high and a high density $\psi_1$ control signal grid must therefore be used. This would, however, have a severe effect on the calculation effort. For this reason, a $\psi_1$ 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 the flow variable, and the time step corresponds to the hydraulic flow delay/short horizon prediction. In the evaluation, a 0.1 s 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 also be done over the flow variable, which increases the dimension in the interpolation. This increases the computational load, but the most severe effect occurs in 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 $s_k < s_m - \kappa$ and $s_m - 1 < s_k + 1(u)$, with $\kappa$ being a small value which acts as a minimum $At$ for the next simulation step. When this detection occurs, $At$ 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 $At, u$ 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 $s_m < s < s_{k+1} < s_{m+1}$, $\tilde{J}$ is interpolated

![Figure 9](image-url)

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 $At$ is reduced until $s(t_k + At) = s_{m-1}$. 
This hydraulic volume is the integrated flow of hydraulic fluid to the lift and tilt cylinders as a function of the 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 $p_H$.

### 2.5 Simulations and Energy Balance

The choice of dynamic programming for the 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:

$$\omega_{e,k+1} = \omega_{e,k} + \frac{T}{I_e} dt$$

and during this step the work performed by the torque is:

$$W_1 = T \omega_{e,k} dt$$

while the change in kinetic energy is:

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

and correspondingly for the vehicle speed dynamics, and also if formulated as functions of the 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 $\psi_1$ to the torques $T_T$ and $T_W$ is very strong, and the optimizer will therefore be highly inclined to use this shortcut by fast switching between high and low $\psi_1$, 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 to complete an entire driving cycle. This problem can be prevented by using energy formulations for both vehicle and engine speed dynamics, according to:

$$\frac{d}{dt} \frac{m_v v^2}{2} = v_s(r^{-1} T_W - r^{-1} T_b - F_w)$$

$$\frac{d}{dt} \frac{1}{2} I_e \omega_e^2 = \omega_e(T_e - T_T - T_H)$$
The Euler method simulation steps can be formulated as:

\[ v_{s,k+1} = \sqrt{v_{s,k}^2 + \frac{2v_{s} \Delta t}{m} \Sigma F} \]  \hspace{1cm} (36)

\[ \omega_{e,k+1} = \sqrt{\omega_{e,k}^2 + \frac{2\omega_{e} \Delta t}{I_e} \Sigma T} \]  \hspace{1cm} (37)

and correspondingly when the distance driven, \( s \), is used as the independent variable, which guarantees 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.

3 EVALUATION

Section 2 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.

3.1 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 2.3.3.

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 with 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 2.3.3. 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 130 g, 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 119 g. The 4th evaluation from the left is used as an example to illustrate the simulation results. Completing this particular cycle required 130 g of fuel.

The CTG and simulation loading cycles for evaluation 4 are shown in Figure 11. The dotted lines are the loads used in the CTG map calculation (see also Tab. 2) and the solid lines are the cycle used in the simulation.
simulated cycle are more transient than those in the CTG 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, \( x_e \) and \( p_t \), and control signal, \( m_f \), trajectories from the simulation in evaluation 4. The \( x_e \) 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 8th evaluation, referring to Figure 10, in which infinite cost is encountered at \( t = 17 \text{ s} \), 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 with the FTDP method. The discretizations have been made as sparse as possible without significantly affecting the optimization results.

### 3.2 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 2.4.3. 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 1 122 combinations evaluated. The time to fuel weighting parameter is selected as \( \beta = 0.5 \text{ g/s} \), since this gives cycle times similar to the 25 s specified for the SDP method.

![Figure 12](image1.png)

**Figure 12**

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

![Figure 13](image2.png)

**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 \( t = 17 \text{ s} \).

<table>
<thead>
<tr>
<th></th>
<th>( t_{\text{min}} ) (s)</th>
<th>( t_{\text{mean}} ) (s)</th>
<th>( t_{\text{max}} ) (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTG</td>
<td>877</td>
<td>974</td>
<td>1055</td>
</tr>
<tr>
<td>Sim</td>
<td>0.30</td>
<td>0.30</td>
<td>0.32</td>
</tr>
</tbody>
</table>

**Table 4**

Experienced times for CTG map calculation and look-ahead simulation, using 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 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 $\beta = 10\,\text{g/s}$. 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 116 g and the average time use is 24.7 s. The average optimal fuel use over the 34 cycles, which corresponds to the diagonal of Figure 14, is 115 g, and the corresponding average time use is 24.3 s. In the following part, the combination of the 4th cycle for the CTG-map calculation and the 12th cycle for the simulation, referring 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 requires 24.6 s and 115 g of fuel.

The CTG and simulation load cases for this cycle are shown in Figure 17. The dotted lines are the CTG load, as specified in Section 2.4.3, 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, $v_x$, $\omega_x$ and $p_x$, and time step $\Delta t$ trajectories from evaluation 4-12. The $\omega_x$ 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 $\omega_x$ margin. The intake
pressure $p_t$ is plotted along with the static pressure set-point $p_{\text{set}}$ (gray). The $\Delta t$ figure clearly shows the adjustments of $\Delta t$ made near the driving direction changes.

**Figure 19** shows the control, $m_f$, $m_T$, $\psi_1$ 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 $\psi_1$ 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 before encountering infinite cost in evaluation 23-12 with $\beta = 10 \text{g/s}$. See Figures 18 and 19 for explanations.
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, performed with different values of the time to fuel weighting parameter $\beta$. In the evaluations above, $\beta = 5 \cdot 10^{-4}$ was used, since this gives an average cycle time similar to the cycle time specified for the SDP method.

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

3.3 Method 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 uses 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.

3.3.1 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 in 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 2.2 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), render infinite cost. In the simulation, the result is that the engine speed never drops below that which corresponds to the highest predicted

<table>
<thead>
<tr>
<th>$\beta$ (g/s)</th>
<th>$T$ (s)</th>
<th>$M_f$ (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>20.9</td>
<td>124.7</td>
</tr>
<tr>
<td>2</td>
<td>22.1</td>
<td>119.6</td>
</tr>
<tr>
<td>1</td>
<td>23.2</td>
<td>116.9</td>
</tr>
<tr>
<td>0.5</td>
<td>24.6</td>
<td>115.0</td>
</tr>
<tr>
<td>0.2</td>
<td>25.9</td>
<td>114.7</td>
</tr>
<tr>
<td>0.1</td>
<td>26.3</td>
<td>114.4</td>
</tr>
<tr>
<td>0.05</td>
<td>26.4</td>
<td>114.2</td>
</tr>
</tbody>
</table>

Table 6

<table>
<thead>
<tr>
<th></th>
<th>$t_{\text{min}}$ (s)</th>
<th>$t_{\text{mean}}$ (s)</th>
<th>$t_{\text{max}}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTG</td>
<td>13005</td>
<td>13365</td>
<td>14035</td>
</tr>
<tr>
<td>Sim</td>
<td>8.28</td>
<td>9.09</td>
<td>10.6</td>
</tr>
</tbody>
</table>

Figure 21

Illustration of the loads in the unsuccessful evaluation 23-12 with $\beta = 10$ g/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.
$Q_H$ 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 the 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 $\beta$. Increasing $\beta$ increases the $v_s$ dependency in $J(x,k)$, which pushes the vehicle toward higher speeds in the simulation. Predicting the impact of different $\beta$ in the CTG map calculation, in 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.

### 3.3.2 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 measurement. In the SDP method, 73.5\% of the simulations were successful, compared with 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.

However, 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 measurement. In the SDP method evaluations the average fuel use was $130\ g$ and the pre-specified cycle time was $25\ s$. 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 ($119\ g$), because of the implemented margin towards high hydraulic flow. In the FTDP method evaluations, with $\beta = 0.5\ g/s$, the average fuel use was $116\ g$ and the average time use was $24.7\ s$. Changing $\beta$ so that the FTDP average time increases to closer to $25\ s$ might reduce the average fuel consumption somewhat, but Table 5 indicates that this reduction would be small. The fuel use in the FTDP method is close to that achieved with perfect prediction ($115\ g$), though there is a difference in time use ($24.3\ s$). 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.

### 3.3.3 Implementation

If either 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 the 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 CTG will require overestimating 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 the 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 much less powerful computer. The simulations of the 25 s cycles required around 0.3 s using the SDP method and around 10 s using the FTDP method. Despite requiring a 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 $u^t(x, t, P)$ map in advance.

**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 CTG map is calculated from this cycle. The second method, designated FTDP, has vehicle speed as a state of the system, and introduces a fixed 0.1 s delay from driver input to bucket movement, a delay equivalent to a prediction of bucket movement. Again, a CTG map is calculated. The CTG maps are in both methods used in a one-step look-ahead controller for, in 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 1/10th 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 this persists even though the SDP method has several load alternatives in each instant. The most important performance measurement 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 $\beta$. Using a $\beta$ which gives cycle times similar to the one 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 measurement is the fuel use. In the SDP method evaluations, the average fuel use was 130 g and the pre-specified cycle time was 25 s. In the FTDP method evaluations, with $\beta = 5 \cdot 10^{-4}$, the average fuel use was 116 g and the average time use was 24.7 s.

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 1/10th 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 with a fail rate of less than 1% for the FTDP method, while the FTDP method also showed a 10% lower fuel consumption.

REFERENCES


Manuscript accepted in March 2014
Published online in August 2014