Robust Driving Pattern Detection and Identification with a Wheel Loader Application

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

Original Publication:

Tomas Nilsson, Peter Nyberg, Christofer Sundström, Erik Frisk and Mattias Krysander, Robust Driving Pattern Detection and Identification with a Wheel Loader Application, 2014, International journal of vehicle systems modelling and testing, (9), 1, 56-76.
http://dx.doi.org/10.1504/IJVSMT.2014.059156

Robust driving pattern detection and identification with a wheel loader application

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

Dept. of Electrical Engineering
Linköping University
581 83 Linköping, Sweden
E-mail: tomas.nilsson@liu.se
E-mail: peter.nyberg@liu.se
E-mail: christofer.sundstrom@liu.se
E-mail: erik.frisk@liu.se
E-mail: mattias.krysander@liu.se
* Corresponding author

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.

Keywords: robust detection, driving cycle, driving pattern, pattern identification, driving cycle identification, wheel loader, usage detection, usage classification, state automaton

Reference to this paper should be made as follows: Author. (xxxx) 'Title', *Int. J.* , Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Tomas Nilsson is a PhD student at the Department of Electrical Engineering, Linköping University, Sweden. He received his MSc degree in Vehicle Engineering from the Royal Institute of Technology, Stockholm, Sweden, in 2009 and commenced his PhD studies later the same year. His research interests are in vehicular engine and transmission control and optimisation.

Peter Nyberg is a PhD Student at the Department of Electrical Engineering, Linköping University, Sweden. His research interests includes vehicle testing, optimal control, and driving cycles analysis. He received his MSc degree in Applied Physics and Electrical Engineering in 2009 from Linköping University, Sweden.

Christofer Sundström recieved his MSc degree in Engineering Physics from Lund University, Sweden, in 2008, and is currently working toward his PhD degree at the Department of Electrical Engineering, Linköping University, Sweden. His research interests are within model-based diagnosis and modelling of hybrid powertrains.

Erik Frisk is an Associate Professor at the Department of Electrical Engineering, Linköping University, Sweden. His research expertise is in signal processing, fault detection, and estimation theory.

Mattias Krysander received his MSc in Electrical Engineering in 2000, and PhD degree in 2006, both from Linköping University, Sweden. Currently, he has a position as an Associate Professor at the Department of Electrical Engineering at Linköping University. His current research interests in the field of model-based diagnosis and prognosis include fault isolation and sensor placement. Graph-theory and discrete mathematics are among his specialties.

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 (Park et al., 2011), 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 customers 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 customers 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 (Rehnberg and Drugge, 2008). 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 (Engström and Victor, 2001; Johannesson, 2009; Manzie et al., 2007), 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 (Lin et al., 2004) and (Jeon et al., 2002). 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 (Kelley, 1998; Mitrovic, 2005) and system identification (Ljung, 1999), for detecting wheel loader operating patterns, which is robust against large usage disturbances that inherently affects vehicle operation.

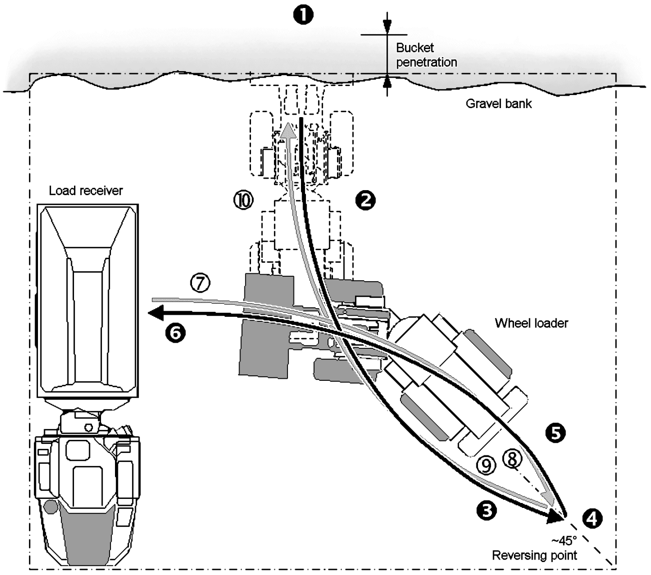

Figure 1: A view of a short loading cycle (Filla, 2012).

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.

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.

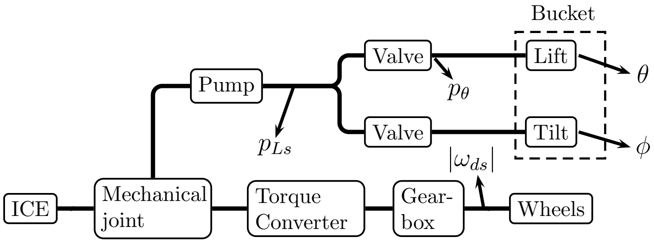

Figure 2: The configuration of the vehicle powered by the internal combustion engine, ICE. The pressure in the lift cylinder of the bucket p_θ and the Ls pressure p_{Ls} are two pressures in the system where only p_{Ls} is known in production loaders. The bucket lift and tilt angles, and the angular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

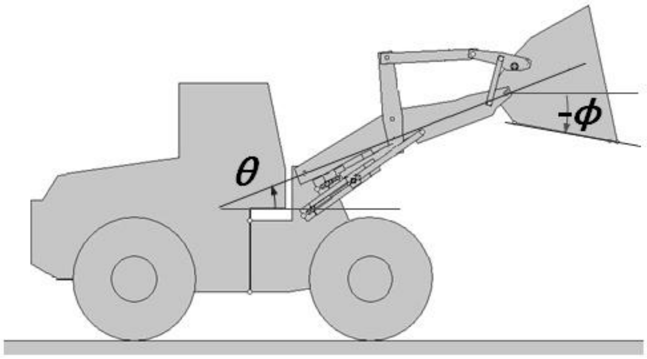

Figure 3: Side view of a wheel loader, indicating the lift angle θ and the bucket tilt angle ϕ.

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, 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.

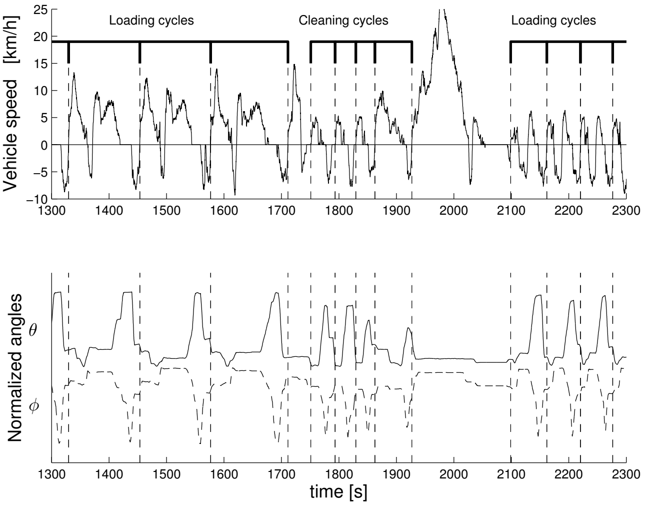

Figure 4: Data collected during typical wheel loader operation. in this case there are six loading cycles and four cleaning cycles.

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.

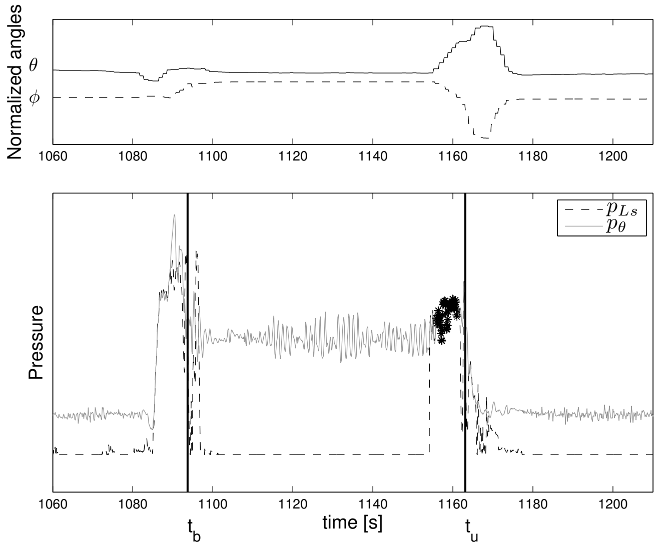

Figure 5: The load sensing pump and lift cylinder pressures, p_{Ls} and p_θ, during a transport. The bucket is loaded during the first pressure peak and unloaded at the second pressure peak. Time $t = t_b$ corresponds to where the wheel loader starts backward motion and $t = t_u$ when the bucket is unloaded.

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 (Kelley, 1998) 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 (Han et al., 2007). 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 (Ohlsson-Öhman, 2012), 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 $\mathcal{I}_k^l = \{t_k, \cdots, t_l\}$.

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

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

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

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

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.

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

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.

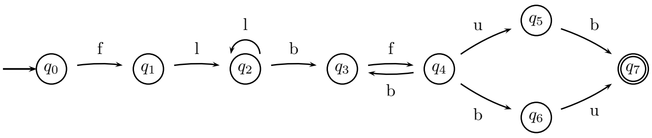

Figure 6: Transition diagram of the automata describing a loading cycle. The initial state is q_0 and the accepting state is q_7.

It is possible to model any number of cycles using automatons 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.

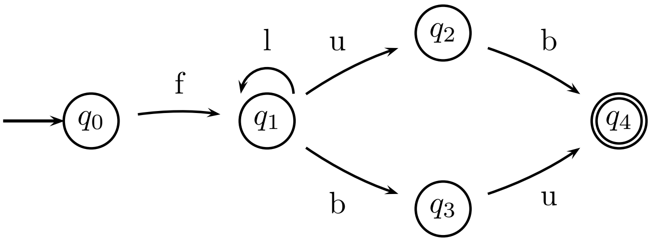

Figure 7: Transition diagram of the automata of the cleaning cycle. The initial state is q_0 and the accepting state is q_4.

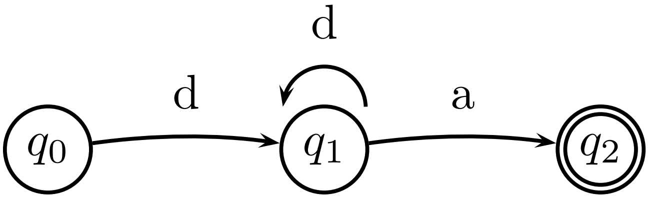

Figure 8: Transition diagram of the automata describing dormant operation. The initial state is q_0 and the accepting state is q_2.

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.

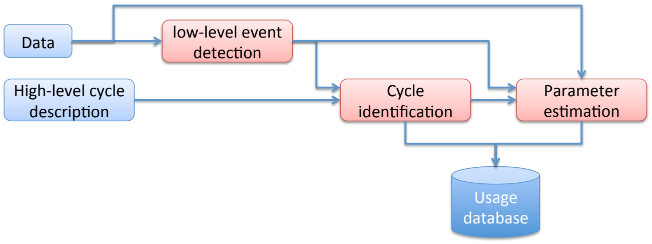

Figure 9: Data comes from sensors. The events are generated from the rules (1)-(6). The high-level description are the patterns that are fed to the cycle identification along with the generated events. Finally, parameters are estimated and stored together with cycle information in a usage database.

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 (Mannila et al., 1997; Das et al., 1997).

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 $flbfub$ 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 (Navarro, 2001; Antoniou et al., 2006).

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.

4.3.1 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

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

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

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

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

4.3.2 *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

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

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 (Nezhadali and Eriksson, 2013). 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, $\hat{\dot{\theta}}$, is used and the signal is smoothed by a low-pass Butterworth filter. The interval of interest is defined as

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

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, $\hat{p}_\theta$, for different driving situations. Due to robustness the second condition for the assumption $p_\theta = p_{Ls}$ is the interval

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

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

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

The bucket load estimate is then

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

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 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

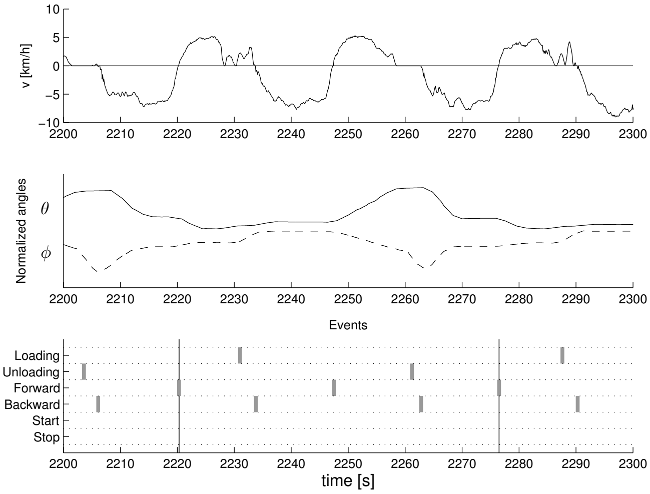

Figure 10: A data sequence with generated events and a detected loading cycle. The beginning and end of the loading cycle are marked with vertical lines in the lowest plot.

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.

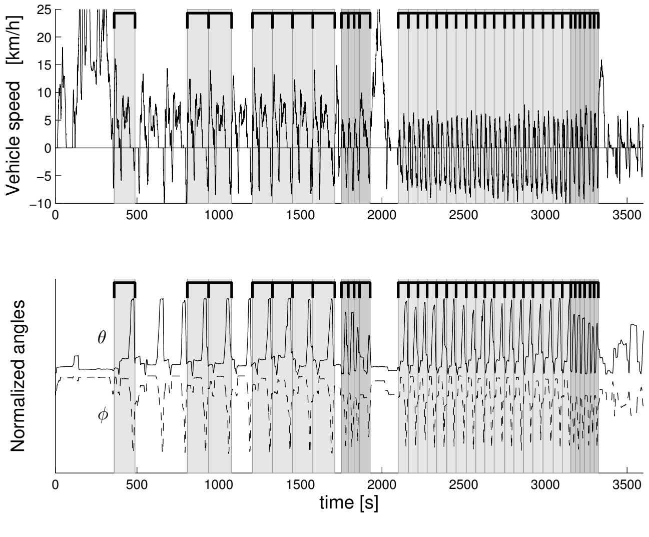

Figure 11: A cycle-partitioned dataset. White = no cycle, light gray = loading cycle, dark gray = cleaning cycle.

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.

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%**

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 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

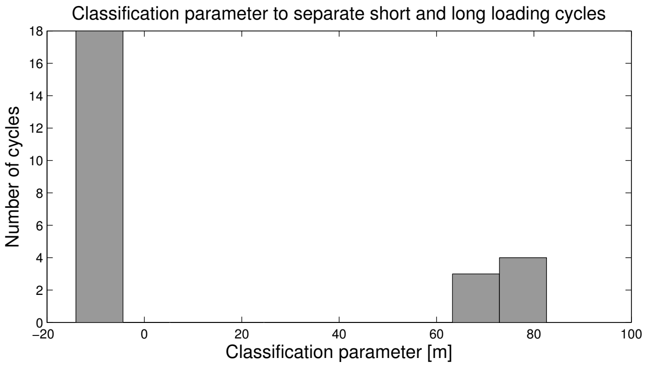

Figure 12: Histogram of r, as defined in (7), for the dataset presented in Figure 11.

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 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 4.3.2 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_{Ls}, when the driver lifts the bucket. In finding the time interval when p_{Ls} 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 4.3.2, 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.

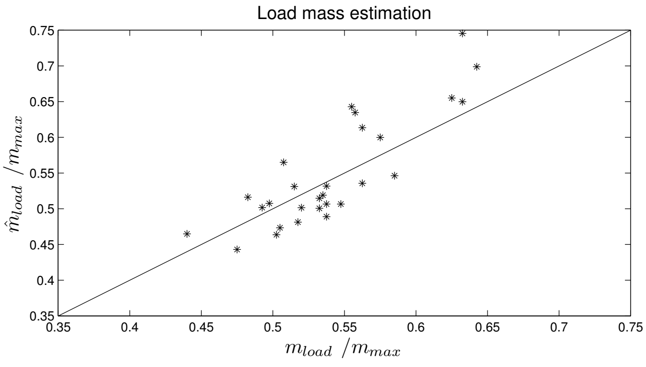

Figure 13: This figure shows the bucket load estimation, normalized with the maximum load of the wheel loader. The estimated normalized masses are given on the vertical axis and the sensor values on the horizontal axis.

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 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.

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

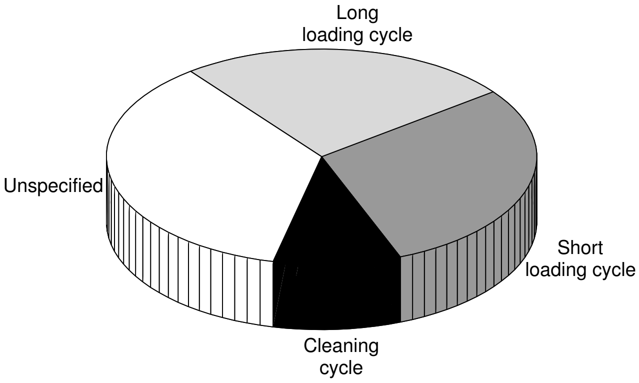

Figure 14: An example of a usage summary.

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 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

Antoniou, P., Holub, J., Iliopoulos, C., Melichar, B. and Peterlongo, P. (2006), Finding common motifs with gaps using finite automata, *in* O. Ibarra and H.-C. Yen, eds, ‘Implementation and Application of Automata’, Vol. 4094 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp.69–77.

Das, G., Fleischer, R., Gasieniec, L., Gunopulos, D. and Kärkkäinen, J. (1997), Episode matching, *in* ‘Combinatorial Pattern Matching’, Vol. 1264, Springer Berlin Heidelberg, pp.12–27.

Engström, J. and Victor, T. (2001), Real-time recognition of large-scale driving patterns, *in* ‘Proceedings of 2001 IEEE Intelligent Transportation Systems’, Oakland, CA., USA, pp.1018 –1023.

Filla, R. (2012), ‘Study of a method for assessing operability of working machines in physical and virtual testing’, *International Journal of Vehicle Systems Modelling and Testing*, Vol. 7 No. 3, pp.209–234.

Han, J., Cheng, H., Xin, D. and Yan, X. (2007), ‘Frequent pattern mining: current status and future directions’, *Data Mining and Knowledge Discovery*, Vol. 15 No. 1, pp.55–86.

Jeon, S., Jo, S., Park, Y. and Lee, J. (2002), ‘Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition’, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 124 No. 1, pp.141–149.

Johannesson, L. (2009), Predictive control of hybrid electric vehicles on prescribed routes, PhD thesis, Chalmers University of Technology, Sweden.

Kelley, D. (1998), *Automata and formal languages*, Prentice Hall, Upper Saddle River, NJ, USA.

Lin, C.-C., Jeon, S., Peng, H. and Lee, J. (2004), ‘Driving pattern recognition for control of hybrid electric trucks’, *Vehicle System Dynamics*, Vol. 42 No. 1-2, pp.41–58.

Ljung, L. (1999), *System identification - theory for the user*, 2nd edn, Prentice Hall, Upper Saddle River, NJ, USA.

Mannila, H., Toivonen, H. and Verkamo, I. (1997), ‘Discovery of frequent episodes in event sequences’, *Data Mining and Knowledge Discovery*, Vol. 1, pp.259–289.

Manzie, C., Watson, H. and Halgamuge, S. (2007), ‘Fuel economy improvements for urban driving: hybrid vs. intelligent vehicles’, *Transportation Research Part C: Emerging Technologies*, Vol. 15 No. 1, pp.1 – 16.

Mitrovic, D. (2005), ‘Reliable method for driving events recognition’, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6 No. 2, pp.198–205.

Navarro, G. (2001), ‘A guided tour to approximate string matching’, *ACM Computing Surveys*, Vol. 33, pp.31–88.

Nezhadali, V. and Eriksson, L. (2013), 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’, Beijing, China.

Ohlsson-Öhman, K. (2012), Identifying operator usage of wheel loaders utilizing pattern recognition techniques, Master's thesis, Linköping University, Linköping.

Park, S., Malikopoulos, A., Kokkolaras, M. and Jung, D. (2011), 'Thermal management system modeling and component sizing for heavy duty series hybrid electric vehicles', *International Journal of Heavy Vehicle Systems*, Vol. 18 No. 3, pp.272–287.

Rehnberg, A. and Drugge, L. (2008), 'Ride comfort simulation of a wheel loader with suspended axles', *International Journal of Vehicle Systems Modelling and Testing*, Vol. 3 No. 3, pp.168–188.

Nomenclature

a	action event
a_θ, b_θ	parameters of the load mass model
b	backward event
d	dormant event
f	forward event
$\mathcal{I}$	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