Offline driving pattern detection and identification under usage disturbances

Tomas Nilsson, Christofer Sundström, Peter Nyberg,
Erik Frisk, and Mattias Krysander
Dept. of Electrical Engineering, Linköping University, SE-581 83
Linköping, Sweden
Email: {tnilsson,csu,petny,frisk,matkr}@isy.liu.se
Report: LiTH-ISY-R-3054

November 26, 2012

Abstract

Optimizing the configuration of a wheel loader to customer needs can lead to a significant increase in efficiency with respect to fuel economy, cost, component dimensioning etc. Experience show that even modest customer adaptation can save around 20% of fuel cost. A key motivator for this work is that wheel loader manufacturers in general does not have full information about customer usage of the machine and the main objective here is to develop an algorithm that automatically, using only production sensors, extracts information about the usage of a machine at a specific customer site. Two main challenges are that sensors are not located with respect to this task and the significant usage disturbances that typically occur during operation. The proposed solution is a robust method, based on a mix of techniques using basic signal processing, state automaton techniques, and parameter estimation algorithms. A key property of the method is the method of combining, individually very simple, basic techniques in a scheme where robustness are introduced. The approach is evaluated on measured data of a wheel loader loading gravel and shot rock.

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. Experience shows that proper matching of machine configuration and customer profile can have significant influence on machine efficiency. For example up to 20% of fuel can be saved by basic machine reconfiguration. Since many customers operate their wheel loaders mainly for specific tasks throughout the entire lifespan, significant efficiency improvements can be achieved if the usage profile is known. With more detailed knowledge of the specific task, size of components can be adapted to customer needs. Also engine control and automatic gear shifting algorithms could be adapted to further improve efficiency resulting in lower purchase cost, higher productivity, and lower fuel consumption.

Today, a common situation is that only rough estimates, typically averaged quantities over long periods of time, about customer usage is available. Therefore, customer

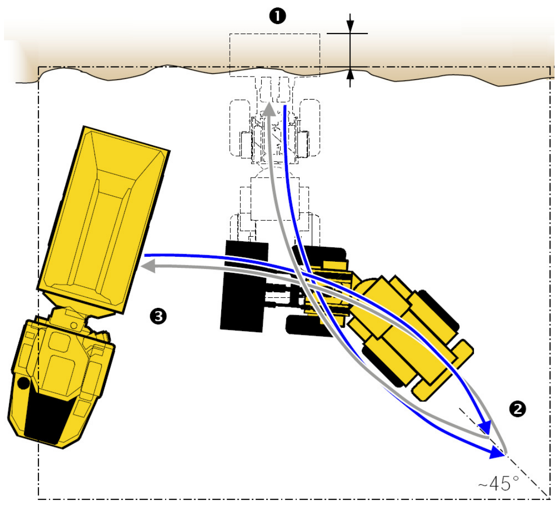


Figure 1: A view of a short loading cycle [4].

adaption is based on qualified guesses and test drive experience and no adaptation to a particular customers needs is possible. This situation is the main motivation for the objective here; to develop an algorithm that, using production sensors only, automatically extracts detailed information from customers vehicles during operation to support improved matching of vehicle configuration and customer usage. One possibility, to capture machine operation, would be to log all data and present it to sales and development engineers. However, the volume of data would easily become too vast and it is therefore important to condense measured data to the essentials. Main challenges are then to first define what information that is relevant and then to be robust against significant usage disturbances.

Related works for on-road automotive vehicles are for example [3] and [6, 11] showing a potential to increase vehicle efficiency by use of driving pattern knowledge. For construction machines this task is in a way 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 [8] and [5]. A main difference here is that the key objective is to analyze usage patterns, not to design a control algorithm. The key contribution of this report is an algorithm, that seamlessly integrates techniques from automata theory [7, 12] and system identification [9], to obtain an algorithm that is robust against large usage disturbances that inherently affects vehicle operation.

2 Problem formulation and challenges

Before the main problem formulation together with main challenges is stated, a brief introduction of typical wheel loader usage and sensor configuration is shown to illustrate the problem area.

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 onto the hauler. In the figure, the loader is starting from point 2, 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 2 it approaches the hauler at point 3. The loaded bucket is lifted during the displacement from the pile to the hauler. After emptying the bucket the loader reverses to point 2 while lowering the bucket.

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, the pressure p_{Ls} in the load sensing hydraulic pump, and the angular speed $|\omega_{ds}|$ of the drive shaft that is transformed to vehicular velocity v via information about the gear selection. 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.

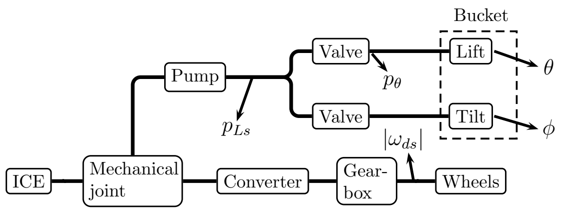


Figure 2: The configuration of the vehicle powered by the internal combustion engine, ICE. The pressure in the lift cylinder of the bucket p_θ and the Ls pressure p_{Ls} are two pressures in the system where only p_{Ls} is known in production loaders. The bucket lift and tilt angles, and the angular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

Figure 3 shows unfiltered measurements of vehicular velocity and the bucket lift and tilt angle during typical wheel loader operation. The data is collected during three consecutive loading cycles similar to the one described in Section 2.1 followed by some cleaning where dropped gravel is moved back to the pile. This type of unfiltered signals 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. The bucket load is not measured directly but must be estimated based on pressure measurement data typically looking as shown in Figure 4.

2.3 Problem formulation

The main problem studied in this report is illustrated in Figure 3 where operating cycles have been identified by hand. We propose an algorithm that automatically provides similar cycle detection, identification, and even partitioning, given only production sensor measurements. In addition to cycle identification and partitioning, different usage parameters, such as the bucket load weight, are estimated. The models developed here deals with bucket handling, but the methodology is applicable to also other types of cycles, e.g., pallet handling by additional modeling of events and cycles.

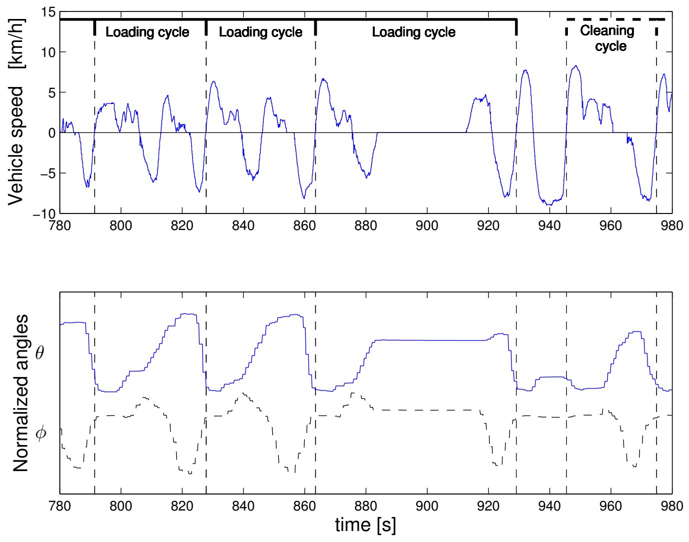


Figure 3: Data collected during typical wheel loader operation.

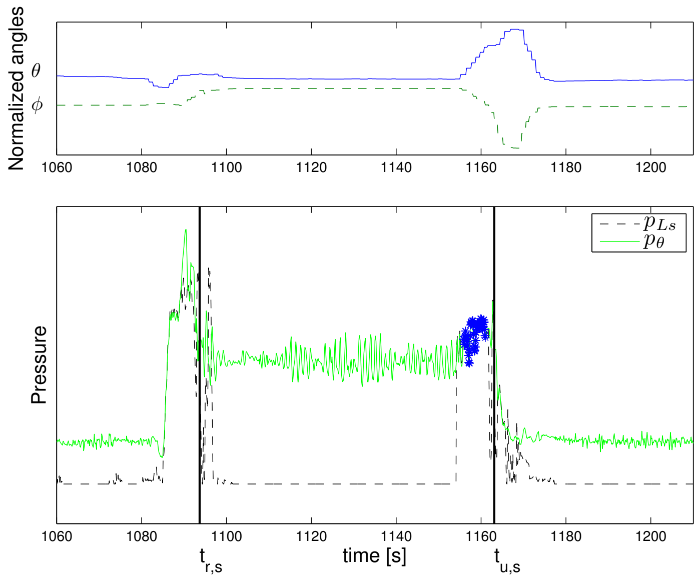


Figure 4: The load sensing pump and lift cylinder pressures, p_{Ls} and p_θ, during a transport. The bucket is loaded during the first pressure peak and unloaded at the second pressure peak.

2.4 Challenges

The cycle identification problem is challenging because the specification of different types of cycles are based on what the driver think he does for a given mission. The result is specifications similar to the description of the operation given in Section 2.1. Figure 3 shows three such loading cycles which all perfectly match the description, but have different signal trajectories such as different amplitudes and lengths in parts of the cycle. In this case the operator is not driving 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 type of disturbances.

Challenges of the parameter estimation will be exemplified with the bucket load estimation and Figure 4. A straightforward way of estimating the load is to use the pressure p_θ in the bucket lift cylinder when the machine carries a load and is clear off the pile i.e. between the times denoted by $t_{bl,s}$ and $t_{bl,e}$ in the figure. As said before, this pressure is not known in production vehicles which means that the estimation can only use the load sensing pressure p_{Ls}. This makes the estimation problem more challenging since p_{Ls} supplies all hydraulics including lifting and tilting. Furthermore, the bucket load only effects the load sensing pressure p_{Ls} when the valve connecting the lifting cylinder to the pump is opened, i.e. during transient operation, which is the main cause of the difference between p_θ and p_{Ls} in the figure. This reduces the relevant measurement data to the data points indicated with stars. This highlights the need for algorithms able to find episodes during which relevant data can be extracted.

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, i.e. the cycles mentioned in Section 2, and 2) support the extraction of important operation parameters such as bucket load weight and distance traveled etc. Further, since measurement data from the same operation cycle can look significantly different on a low enough level, it is important that the model is on a suitable level of abstraction, high enough to be able to robustly model operation, and detailed enough to capture relevant properties of specific cycles. Here, cycles will be modeled using events which will enable us to utilize standard automata theory [7] to devise cycle detection methods in Section 4.

3.1 Events

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

Analysis of the current problem indicates that four events are required to describe cycles of operation of the wheel loader:

- transition from backward to forward motion - f
- transition from forward to backward motion - b
- bucket unloading - u

- bucket loading - l

For example, as shown in Figure 1, event f happens at point 2, point 1 and 3 are positions where event b happens. Event l occurs at point 1, and event u at point 3. To formally state exactly the models for when events have happened, we use the notation x_k for an event x at time $t = t_k$ and let $z_k = z(t_k)$. 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

$$\mathcal{I} = [t_j, t_{j+1}, \ldots, t_k]$$

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

$$v_j < 0 \text{ and } v_{j+1} = \cdots = v_{k-1} = 0 \text{ and } v_k > 0 \tag{1}$$

where v is the vehicle velocity, computed from the drive shaft angular speed $|w_{ds}|$ 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

$$v_j > 0 \text{ and } v_{j+1} = \cdots = v_{k-1} = 0 \text{ and } v_k < 0 \tag{2}$$

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

$$u_k \text{ is generated if } \phi_{k-1} \geq \xi \text{ and } \phi_k < \xi \tag{3}$$

where ξ is a model parameter. As can be seen in Figure 3, 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.

$$\begin{aligned} l_{k-L} \text{ is generated if } \theta_k - \theta_{k-L} > \alpha \text{ and} \\ \phi_k - \phi_{k-L} > \beta \text{ and } v_{k-L} > 0 \end{aligned} \tag{4}$$

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 simple rules (1)-(4) measurement data of velocity $v(t)$, lift angle $\theta(t)$, and tilt angle $\phi(t)$ is transformed into a sequence of symbols from the alphabet $\Sigma = \{f, b, u, l\}$ with corresponding time stamps.

3.2 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 after to the last event in the cycle.

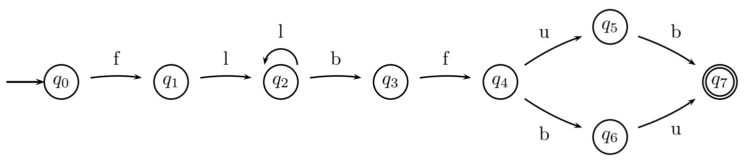


Figure 5: Transition diagram of the automata describing a loading cycle. The initial state is q_0 and the accepting states is q_7.

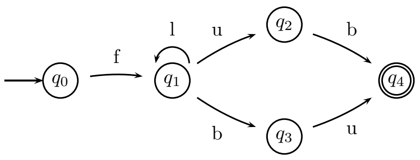


Figure 6: Transition diagram of the automata of the cleaning cycle.

One can model any number of cycles using automatons and here two common types of behavior will be modeled, first a *loading cycle* and second a *cleaning cycle*. A loading cycle with the ideal sequence of events is shown at the upper half of Figure 5, $flbfub$. 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 2, l at point 1, b at point 1, f at point 2, u at point 3, and finally b at point 3. 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 effect the order of the events. For example, in Figure 1 this is common at point 3. 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, it is straightforward to take such variations into account as is depicted in the full automata in Figure 5. The model for a cleaning cycle is slightly smaller but follows a similar structure as shown in Figure 6.

3.3 In-cycle phases

The modeling in Section 3.2 makes it possible to detect and identify different cyclic operation episodes. However, as discussed in Section 2, a finer partitioning of the data is desired. This is required for determining parameters such as transportation distance and bucket load weight. These shorter episodes of operation, parts of cycles, are called in-cycle phases or simply *phases*. Examples of phases are the bucket filling and forward motion. In contrast to the events, the phases are time intervals, defined by its start and end time. Another difference is that while the events are general and common to the entire data set, the phase partitioning is specific to each cycle type. This means that both the partitioning pattern and the conditions for each phase can be tailored to each cycle model.

Since the loading cycle is the main focus of this work, this has been partitioned into phases according to a predefined pattern. The phases in this cycle have been divided into two sequences: one containing the vehicle motion and the other containing the load handling. Within each of these parallel sequences there are no overlapping phases. The phases in the motion sequence are:

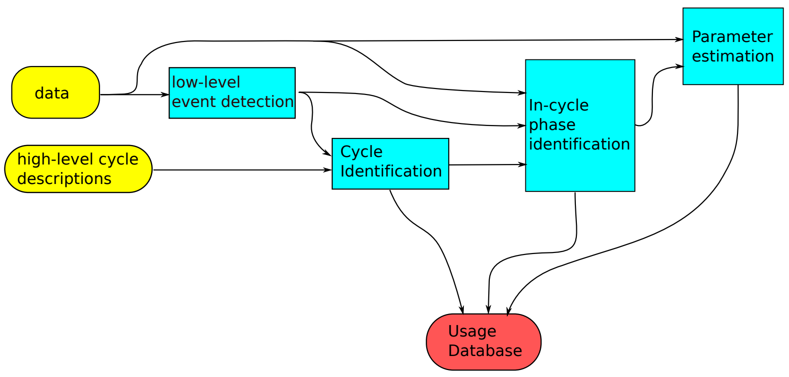

data
low-level event detection
high-level cycle descriptions
Cycle Identification
In-cycle phase identification
Parameter estimation
Usage Database

Figure 7: Data comes from sensors. The events are generated from the rules (1)-(4). The high-level description are the patterns that are fed to the cycle identification along with the generated events. The phase identification uses data and events to partition cycles.

- forward motion - fm
- backward motion - bm
- direction change - dc

And the phases in the load sequence are:

- bucket filling - bf
- bucket loaded - bl
- bucket emptying - be
- bucket unloaded - bu

The phase sequences in the loading cycle are:

- Motion: $fm_1 - dc_1 - bm_1 - dc_2 - fm_2 - dc_3 - bm_2 - dc_4$
- Load: $bu_1 - bf - bl - be - bu_2$

Here all phases that occur more than once during each cycle has been given an index number. Descriptions of each phase detector are given in Section 4.3.

4 Method

In this section we propose an 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 7. The input to the algorithm is measurement data and the high-level cycle descriptions provided as automata like the ones given in Figures 5 and 6. The algorithm can be divided into 4 parts: event detection, cycle identification, phase identification, and finally parameter estimation. Next, the different parts will be described in detail.

4.1 Event detection

The event detectors (1)-(4) 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 constants has been made with the aim of robustness in the detection of the events. The tuning was made by comparing filmed sequences of wheel loader usage to the corresponding sensor data.

The selected value of ξ in (3) is 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.

Common sizes of lift angle change, tilt angle change, and loading time have been determined from the filmed sequences of data. This can be used as α, β and L in (4). However, variation of these variables are large. This is primarily accounted for by increasing the sensitivity of loading detection by decreasing α, β, and increasing L. The high sensitivity of the detector leads to multiple alarms and this is considering when designing the cycle models shown in figures 5 and 6 to achieve robust cycle identification.

4.2 Cycle identification

The inputs to the cycle identification is an event sequence and a set of automata each describing a cycle type. The cycle identification is then performed similar to the string matching algorithms given in [10, 2].

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 5. The last part of this word, i.e. the 3 events fub also match the cleaning cycle automaton in Figure 6. 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 5 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 [13, 1].

4.3 Phase identification

The phase partitioning procedure is specific to each cycle model, therefore this is only described by the illustrative example of the loading cycle. The start and end time of a detected cycle will be denoted $t_{c,s}$ and $t_{c,e}$ respectively. The start and end time for a phase x will similarly be denoted by $t_{x,s}$ and $t_{x,e}$ respectively.

The vehicle motion phases are easily detected; the rules for the forward and backward events can be reused. In (1), t_j defines $t_{bm,e}$ and t_k defines $t_{fm,s}$. In (2), t_j defines $t_{fm,e}$ and t_k defines $t_{bm,s}$. In both (1) and (2), t_j and t_k define $t_{dc,s}$ and $t_{dc,e}$ respectively. Since the same conditions are used in the event detection this will necessarily produce the phase sequence defined in Section 3.3.

The *be* phase is equally easy to detect. The condition (3) is used for detecting $t_{be,s}$, and the opposite condition

$$\phi(t_k) \geq \xi \text{ and } \phi(t_{k-1}) < \xi \tag{5}$$

for $t_{be,e}$. In case (5) does not occur before the end of the cycle, then $t_{be,e} = t_{c,e}$.

Detection of the bucket filling, which in general is difficult, is substantially aided by the bounds set by the detected cycles. The bucket filling is known to occur once during fm_1. The first test is to search for usage of the first gear during fm_1, since the automatic gearbox is set so that the first gear is seldom used except when forcing the bucket into a pile. If the first gear is not used during fm_1, a search can be made in p_{Ls}, and define $t_{bf,s}$ as the first instant during fm_1 where $p_{Ls}(t) > \sigma \cdot \max_{fm_1} p_{Ls}$, with σ being a tunable parameter, and $t_{bf,e}$ as $t_{fm_1,e}$. Yet another criterion that can be used for finding $t_{bf,s}$ is the first bucket loading event in fm_1. In this case the first gear criterion proved to be the most accurate. The risk of having more than one first gear episode was countered by using the one that was the closest to the first bucket loading event.

The unloaded and loaded phases are defined by the filling and the emptying phases. bu_1 is defined by $t_{bu_1,s} = t_{c,s}$, $t_{bu_1,e} = t_{bf,s}$. *bl* is defined by $t_{bl,s} = t_{bf,e}$, $t_{bl,e} = t_{be,s}$ and bu_2 is defined by $t_{bu_2,s} = t_{be,e}$, $t_{bu_2,e} = t_{c,e}$.

4.4 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 report a suggested parameter for separation between short and long loading cycles and an estimation of the bucket load weight are presented since these require some additional analysis.

4.4.1 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 the third cycle in Figure 3 might be classified as a long loading cycle due to the stop in the middle, even though this cycle is similar to the first two cycles that are clearly short ones. To get a more robust classification parameter for distinguishing between short and long cycles the ratio

$$r = \frac{\int_{t_{c,s}}^{t_{c,e}} v(t)\,\mathrm{dt}}{\int_{t_{c,s}}^{t_{c,e}} |v(t)|\,\mathrm{dt}} \tag{6}$$

is used, where as before $t_{c,s}$ is the start time of the cycle, and $t_{c,e}$ is the end time. 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 loader is driving forward more than reversing and r is closer to 1. Therefore, r is calculated for every loading cycle and is compared to a threshold, ψ, and the cycle is deemed short or long according to

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

4.4.2 Bucket load estimation

The bucket load weight is estimated once for each loading cycle during the bucket loaded phase. The starting and ending times of the phase are denoted $t_{bl,s}$, and $t_{bl,e}$, respectively, as mentioned in Section 4.3. When estimating the load in the bucket, a model of the relation between the pressure p_θ in the lift cylinder and the mass m_{load}, using the geometry of the machine, is used

$$m_{load} = f(p_\theta) \tag{7}$$

Unfortunately, there is no sensor in production vehicles for measuring p_θ, but only the pressure p_{Ls} at the hydraulic pump is known. However, during the phase where the user lifts the bucket, the pressure p_{Ls} is close to the pressure p_θ. This can be seen in Figure 4. The equality $p_{Ls} = p_\theta$ 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 smoothened by a low-pass Butterworth filter. The interval of interest is defined as

$$\mathcal{I}_1 = \left\{ t_i : \hat{\dot{\theta}}(t_i) > \gamma \cdot \max_{t_k \in [t_{bl,s}, t_{bl,e}]} \left\{ \hat{\dot{\theta}}(t_k) \right\} \right\} \tag{8}$$

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

$$\mathcal{I}_2 = \left\{ t_i : p_{Ls}(t_i) > \delta \cdot \max_{t_k \in [t_{bl,s}, t_{bl,e}]} \{ p_{Ls}(t_k) \} \right\} \tag{9}$$

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

$$\hat{p}_\theta = \frac{1}{|\mathcal{I}_1 \cap \mathcal{I}_2|} \sum_{t_k \in \mathcal{I}_1 \cap \mathcal{I}_2} p_{Ls}(t_k) \tag{10}$$

The bucket load estimate is then

$$\hat{m}_{load} = f(\hat{p}_\theta) \tag{11}$$

The samples where conditions $\mathcal{I}_1$ and $\mathcal{I}_2$ are fulfilled are marked with stars in Figure 4. 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 $\delta = 0.50$ has been used.

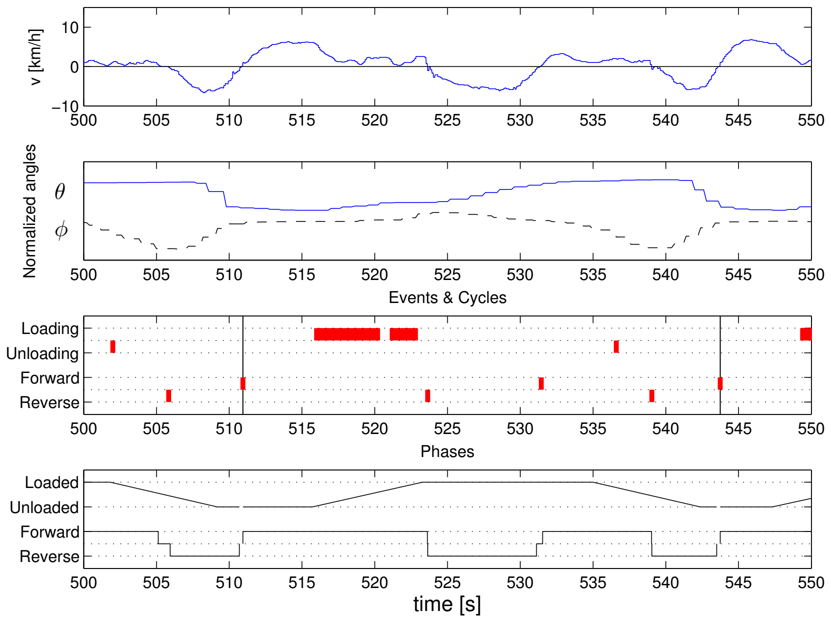


Figure 8: A data sequence with events, a detected loading cycle and phase partitioning

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, or operation. 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. Despite these difficulties, the proposed algorithm has been evaluated against real data. For this purpose 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 told 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 partitioning which is compared to the result of the algorithm.

Figure 8 shows a close-up of one of these data sets. It shows the detected events, an identified cycle, and computed in-cycle phases. In this particular case, the visible sequence of events is $ubfl^{32}bfubfl^{4}$, and in this sequence a loading cycle $fl^{32}bfub$ has been identified. Figure 9 shows longer time-series of the same data set. The light-gray segments indicate identified loading cycles, the dark-gray cleaning cycles, and the white segments correspond to the parts that did not match any predefined cycle. This output and the visual examination of the corresponding film gave the same result. The examination of the other data sets showed that the accuracy was in general equally good. There were however some exceptions.

The first type was cases where the operator deliberately tried to drive in a peculiar

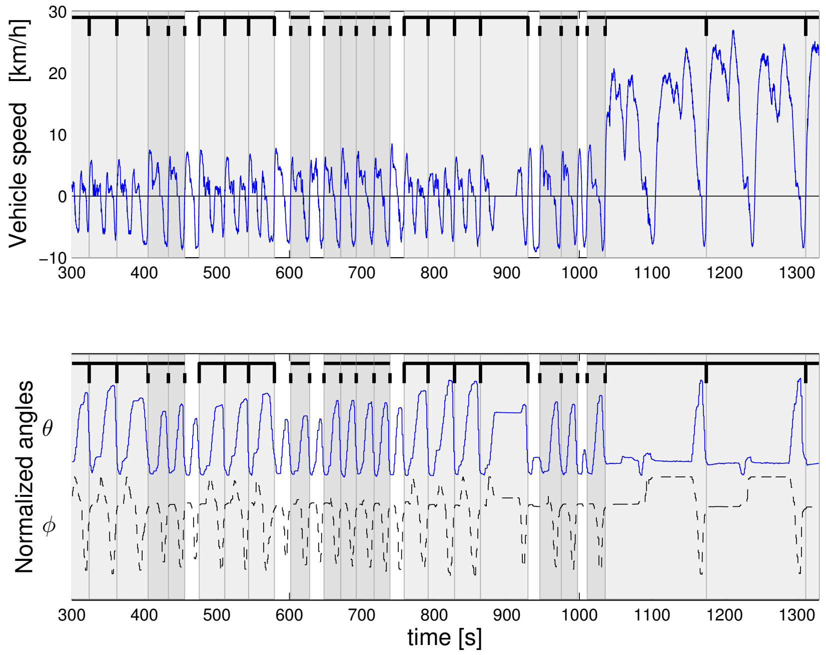


Figure 9: A cycle-partitioned dataset. White = No cycle, Light gray = Loading cycle, Dark gray = cleaning cycle

way. The other type occurred primarily in handling of shot rock, where the algorithm at some few instances missed what we visually judged as being loading cycles. At these occasions the driver either wiggled the bucket at loading or repositioned the vehicular before unloading. It would be easy to adjust the algorithm to accept this behavior by for example requiring a minimum distance or speed for a motion event. 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 an adjustment of the algorithm, the results are still valuable, since most of the operation is partitioned into cycles resulting in an accurate overview of the usage. Furthermore, where no cycle matches, the algorithm detects unusual behavior.

The weight estimator gives a reasonable accuracy, but the main objective was only to investigate whether phase partitioning would aid the weight estimation. Figure 10 shows a comparison between the estimated and actual load weight normalized with the maximum load weight m_{max} of the loader. Since the weight estimator is only used as an illustration of the advantage of having partitioned the data, it has not been optimally tuned. Knowledge of the machine geometry can be used to find the function $f(p_\theta)$ in (7). This has not been done in this work. Instead the pressure has been averaged according to (10) and a simplified function has been used for the corresponding load weight. Determining the actual lift and tilt angle dependency in the function would probably improve the accuracy significantly.

To conclude this section we give two examples of how wheel loader usage can be summarized over a long period of operation. In Figure 11a, a histogram shows the classification parameter r defined in (6), which is used for distinguishing between short and long loading cycles. The figure shows a clear separation between the short loading

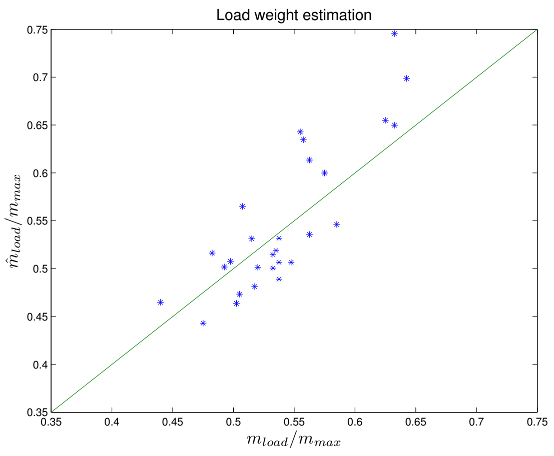
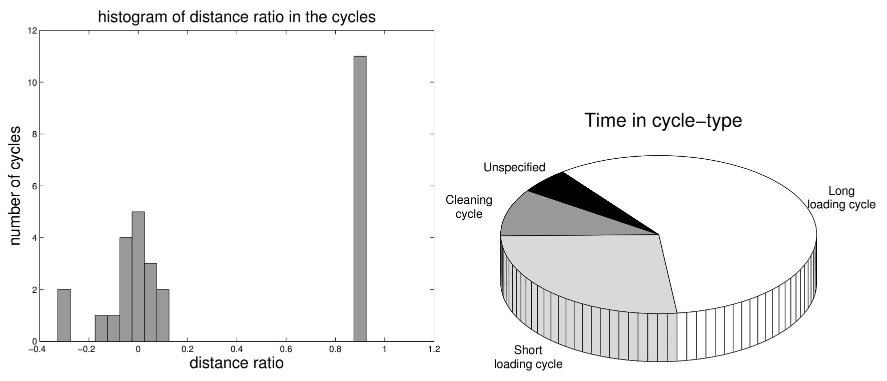


Figure 10: This figure shows the bucket load estimation, and the masses are normalized with the maximum load of the wheel loader. The estimated normalized masses (Section 4.4) are given on the vertical axis and the sensor values on the horizontal axis.



(a) Histogram of r, as defined in (6).

(b) Wheel loader usage.

Figure 11: Examples of usage summaries.

cycles with r-values around 0 and the long loading cycles with r-values significantly greater than 0. Figure 11b, a pie-diagram shows the relative time spent in different cycles for the same data set. In this case, $r = 0.5$ has been used as the limit between short and long loading cycles. The unspecified operation represents only about 5 % of the total time of operation. This illustrates that in realistic operation the proposed algorithm is successful in detecting and identifying the pre-defined cycles.

6 Conclusions

We have investigated the problem of characterizing wheel loader operation by automatic partitioning according to predefined high-level patterns. The detection and identification of these these patterns is robust against control signal noise and differences in driving style, the material that is handled and the working site layout. The implemented partitioning is also a useful tool for finding parameters that would otherwise be inaccessible.

The proposed algorithm has low theoretical complexity which makes it transparent, and requires no additional sensors which makes it easy to implement. The choice of detecting cycles before the phase partitioning is crucial for making a robust phase detection. It also makes it easy to make cycle or phase specific modifications and additions, such as the weight estimator that is specific to the bucket loaded phase of the loading cycle. The drawback is that it increases model size since separate partitioning rules must be developed for each cycle-model. The benefit of the phase partitioning has been illustrated with a load weight estimator which, even though it has not been perfected, shows a fairly good accuracy. The combination of simple detectors and a high-level automata description of the vehicle behavior is a key characteristic of the proposed method that introduces a sufficient level of robustness towards user induced disturbances. For example, robust load detection is achieved by using a sensitive event detector in combination with high-level models coping with multiple alarms.

The evaluation of the algorithm has been made by comparing the outputs to that of visual examination of films recorded during the data collection. In the cases where the visual result was clear, the algorithm was flawless. Some ambiguous cases however showed that the defining of the cycles has to be thoroughly thought through before implementation. In general these uncertain cases were uncommon, and the proposed algorithm works well in showing in which parts of the data the driver operates the vehicle according to the predefined patterns, and pointing out the periods where something unexpected happens. The evaluation of the algorithm has showed that even the few simple models that has been developed is sufficient for explaining in the region of 95% of the data recorded during realistic operation.

References

[1] P. Antoniou, J. Holub, C.S. Iliopoulos, B. Melichar, and Peterlongo P. Finding common motifs with gaps using finite automata. *Implementation and Application of Automata*, pages 69 –77, 2006.

[2] Gautam Das, Rudolf Fleischer, Leszek Gasieniec, Dimitris Gunopulos, and Juha Kärkkäinen. Episode matching. In *Combinatorial Pattern Matching*, volume 1264, pages 12–27. Springer Berlin / Heidelberg, 1997.

[3] J. Engstrom and T. Victor. Real-time recognition of large-scale driving patterns. In *Proceedings of 2001 IEEE Intelligent Transportation Systems*, pages 1018 –1023, Oakland, CA., USA, 2001.

[4] Reno Filla. Operator and machine models for dynamic simulation of construction machinery, 2005. Licentiate thesis, Fluid and Mechanical Engineering Systems, Linköping University, Sweden. LiU-Tek-Lic 2005:44.

[5] S. Jeon, S. Jo, Y. Park, and J. Lee. Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition. *Journal of Dynamic Systems, Measurement, and Control*, 124(1):141–149, 2002.

[6] L. Johannesson. *Predictive Control of Hybrid Electric Vehicles on Prescribed Routes*. PhD thesis, Chalmers University of Technology, Sweden, 2009.

[7] Dean Kelley. *Automata and Formal Languages*. Prentice Hall, 1998.

[8] C.-C. Lin, S. Jeon, and J.M. Peng, H.and Lee. Driving pattern recognition for control of hybrid electric trucks. *Vehicle System Dynamics*, 42(1-2):41–58, 2004.

[9] L. Ljung. *System Identification - Theory For the User*. Prentice Hall, 2nd edition, 1999.

[10] H. Mannila, H. Toivonen, and I.A. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowlegde Discovery*, 1:259–289, 1997.

[11] C. Manzie, H. Watson, and S. Halgamuge. Fuel economy improvements for urban driving: Hybrid vs. intelligent vehicles. *Transportation Research Part C: Emerging Technologies*, 15(1):1 – 16, 2007.

[12] D. Mitrovic. Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):198–205, 2005.

[13] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33:31–88, 2001.