Recognition of Operating States of a Medium-Sized Mobile Machine

T. Krogerus, M. Hyvönen, K. Raivio* and K. Huhtala

Department of Intelligent Hydraulics and Automation, Tampere University of Technology, Tampere, Finland
E-mail: tomi.krogerus@tut.fi, mika.hyvonen@tut.fi, kalevi.huhtala@tut.fi
*Department of Information and Computer Science, Aalto University School of Science, Espoo, Finland

Abstract

In this study the main goal was to study the operating states of a medium-sized mobile machine. The measured time series data were analysed to find frequent episodes (sequences of operating states) to which the conditional probabilities were then calculated. The time series data were first segmented to find events. One or more segments build up an event which can be interpreted to be an operating state. The segments were then clustered and classified. The segment class labels were interpreted as events. As a result, a list of rules was established. The rules describe causal connections between consecutive operating states and transition probabilities from 1st state to 2nd state. The recognized operating states were further analysed to be used in diagnosis of the operation of the machine and focusing the diagnostics on certain operating states.

Keywords: Mobile machine, analysis, diagnostics, operating state, hydraulics, time series, segmentation, episode, event, piecewise linear regression, clustering, classification, association rules, quantization error

1 Introduction

Technological development has produced mobile machines that are remotely operated [1] directly or work autonomously [2] while only being supervised remotely. These machines demand increased monitoring and analysis [3] of the performance and operating states of these machines when there is no person present inside the machine. If the operating states of the machine can be recognized during work tasks, causal connections between consecutive operating states can be determined and furthermore, diagnostics can be focused on specific operating states.

Typically, in a modern mobile machine there is already a lot of information available about the operation of the machine, e.g. process and control data through communication buses, which can be used in analysis. In addition to this, condition monitoring specific sensors can also be added to the system. When all this information from communication busses and additional sensors are recorded, it leads to the generation of a huge amount of data. High dimensionality complicates the processing of time series data especially from the pattern recognition point of view [4]. A time series is defined as a collection of observations made sequentially in time [5].

In order to analyse the time series data of mobile machines, detectors of events to find frequent episodes need to be first created, and after that some higher level description, for example probability distributions or quantization error method [3] should be used. Finding frequent episodes from the measured time series data of a mobile machine requires segmentation of the time series data to find the events. Time series segmentation is often used as a pre-processing step in time series analysis applications. An episode is again defined as a collection of events that occur relatively close to each other in a given partial order [6]. Causal connections are then searched for by analysing the consecutive episodes.

The operation of the machine can be further analysed using the recognized operating states. Diagnostics can be focused on certain operating states. Usually these are the ones that have the biggest changes in the analysed variables. Most information in regard to the detection of anomalies is obtained from these operating states.

2 Description of the studied mobile machine

The studied mobile machine, called IHA-machine [7], was engineered at the Department of Intelligent Hydraulics and Automation at Tampere University of Technology. It was designed to serve as a platform where different types of research could be conducted concurrently. The research platform is shown in fig. 1.

An overview of the hydraulic systems of IHA-machine, which are related to the analysis performed in this study, is given here. More details are presented in [7]. Figure 2 shows a simplified hydraulic circuit of the closed loop hydrostatic transmission (HST) of IHA-machine. It also shows the added sensors and most important auxiliary components that provide safety and maintenance functions.

Figure 1: Studied mobile machine [7].

The main source of power is a 100 kW four-cylinder diesel engine. The HST pump has a displacement of 100 cm^3/r and contains various integrated hydraulic components, sensors and electronics to implement the closed-loop control of the swivel angle and the data communication.

Both the diesel engine and the pump are connected to the main PLC (Programmable Logic Controller) of the machine via the CAN bus. They also have separate control units, which are connected to the CAN bus, offering data from integrated sensors. Therefore, via the CAN bus several parameters related to the operation of these components, e.g. diesel load and HST pump angle, can be monitored and recorded for later analysis.

Every wheel of the machine is equipped with a slow speed hydraulic hub motor, with a displacement of 470 cm^3/r. Each motor has a pressure controlled holding brake and an integrated sensor for measuring rotational speed. A separate hydraulic gear pump provides the power needed by the steering system. The steering of the machine can be controlled using proportional flow control valve and two symmetrically placed hydraulic cylinders. The work hydraulics of the IHA-machine are based on digital hydraulics [7,8], but this part of the machine is not studied here.
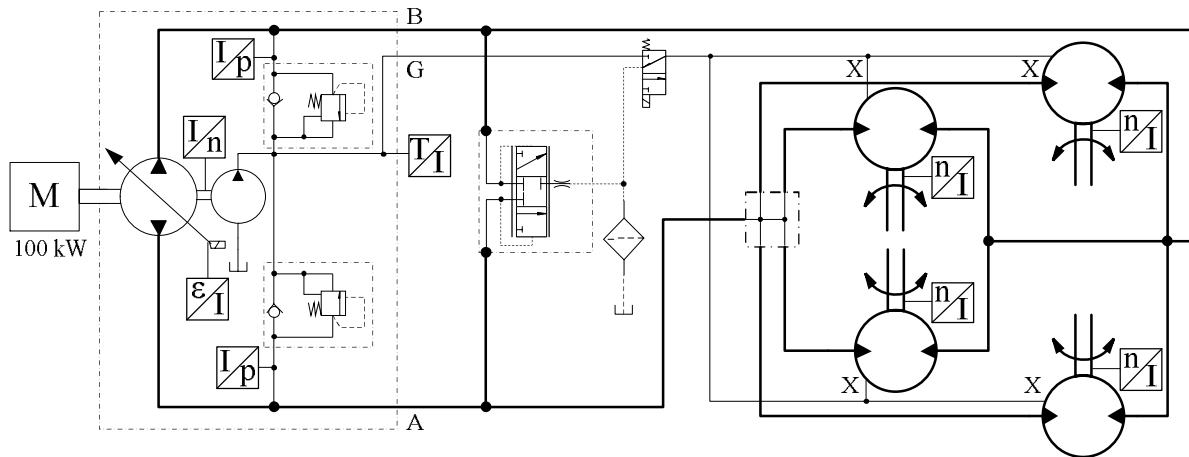

Figure 2: Hydrostatic transmission of studied mobile machine [7].

3 Events and causal connections from time series

In the method used in this study to find causal connections from the time series data, the data have been segmented to extract operating states. One or more segments build up an event which can be interpreted to be an operating state. A state is defined as a combination of the patterns of the selected variables. The sequences of operating states can then be further analysed to discover association rules from the sub-sequences. The method depicted in fig. 3 will be explained in more detail in the following sections.

3.1 Pre-processing

The data have to be scaled before segmentation. This guarantees that all variables have an equal effect on the segmentation result. Representative statistics of the data are needed in order to be able to scale the measurements. To do that, all the variables of the data should have enough variation. The data range and distribution used to compute the statistics should correspond to those of the whole data set. Pre-processing also includes the elimination of outliers

and noise reduction and possibly down sampling using moving average filtering for each measurement.
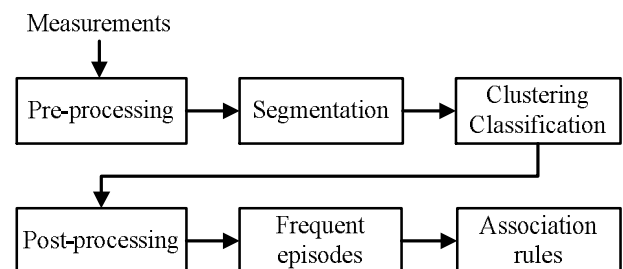

Figure 3: Events and causal connections from time series.

3.2 Segmentation

In segmentation, the time series data are transformed into piecewise linear representation. A segment is a contiguous subset of a time series. In this study, the time series data is segmented using the sliding window method with piecewise linear regression [5]. In this method, curves are approximated with lines: see fig. 4.

New samples are added to the segment until cumulative squared estimation error exceeds a predefined threshold. Another possible criterion is the maximum of squared error in the segment. Several measurements can be segmented together. Thus, they have common cost function and the cost is computed by summing all the measurements. The measurements define together the edges of segments.

The sliding window segmentation method is well suited to online segmentation, because only the preceding samples are needed to define the edges. However, the method can be quite slow if the sampling rate is high. There are several modifications of the sliding window segmentation algorithm and there are other more efficient segmentation methods, but unfortunately most of them are not very suitable for online analysis. This is because the methods expect that the whole sequence is available at the time of segmentation.

3.3 Classification of segments

Pre-processed segments are then clustered and classified. Clustering is the process of organizing objects into groups whose members are similar in some way. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering can be performed using hierarchical methods or k-means algorithm [9].

The parameters of piecewise linear regression lines are used as features. One choice is to use the slopes of lines as features. Also, the offset and the length of segment can be used as additional features. The number of segment clusters has to be defined, or a validation index should be used to select a suitable amount of clusters [10]. It is not necessary to cluster all the available segments, as some of them can be classified using cluster prototype vectors defined using a representative set of segments. Each cluster is presented by a d-dimensional prototype vector (also known as: weight, codebook, model, reference) $m = [m_1, ..., m_d]$, where d is equal to the dimension of the input vectors. It is useful to save these prototype vectors, to classify another set of segments or to analyse classes or classification results later.

3.4 Post-processing segmentation results

Segment classes can be analysed using their prototype vectors. In case slope coefficients are used as segment features, prototypes can easily be visualized to find interpretation for classes or states: see fig. 5.

When the classified segments are converted to events, it is possible to remove non-interesting segments or events. Events describe the behaviour and actions of the system. For example, a segment class with around zero slope coefficients for all measurements can be considered unusable. Events built from these segments can be removed before further analysis, because later analysis will be focused on those operating states where changes in the analysed variables are at their biggest which means that the slopes of the feature vectors are steep. It is also possible to combine subsequent segments with the same class labels if a lower number of detected states are wanted or a high probability of transitions from states to themselves is

undesirable. Of course, the above post-processing methods will change single state distributions and transition probabilities.

The sequence of segment classes will be interpreted as a sequence of states. Each state has a discrete label and a time stamp. Sequences from multiple sources can be combined into one sequence. Possible sources for sequences are the segmentation results of two separate measurement groups. Different methods or features can be used in each segmentation.

3.5 Search for frequent episodes

Frequent episodes can be searched from the sequences using the Apriori algorithm [6,11]. The algorithm finds the most probable sub-sequences from a sequence starting from the sub-sequence of length one. The length of the sequence is increased by one every round. The effectiveness of method is based on the principle that only those episodes which have had all their sub-episodes frequent enough in the previous round are possible candidates for frequent episodes.

The algorithm has two possibilities for initial search of length of two episodes. It is possible to search for serial or parallel episodes from the sequence. In the case of serial episodes, the order in which the states have occurred on the time scale is important, but in the case of parallel episodes the primary interest is to find out if the states occurred in the same time window or not. When longer episodes are searched for, it is, of course, possible to search for more complicated combinations of parallel and serial episodes if desired.

The algorithm has several parameters, which have to be defined. These are the minimum frequency for an episode to be considered as interesting, the length of window used in the search, maximum length of episode and minimum confidence. The last one is actually a threshold for the conditional probability of a sequence.

The sequence occurrence rates are converted to probabilities by dividing the rate by the number of all search windows. The sequence probabilities $P(A, B)$ are further converted to conditional probabilities $P(B|A)$ using single state probabilities: see eq. 1.

$$P(B|A) = \frac{P(A, B)}{P(A)} \qquad (1)$$

Here, state A precedes state B, and $P(A)$ is the probability of state A. The target is to find causal connections between states which occur frequently enough compared to other sequences in which either one of the two states occur.

4 Measured data for analysis

Different test cases were defined and 79 different test drives were carried out to obtain measurement data from different operating states, which were then analysed. These test cases consist of a combination of different work tasks: driving straight, turning, going uphill, downhill, over obstacles, emergency braking, reversing, different driving speeds, and

slow and fast rise time of driving speed. Two different ground types were used: asphalt and gravel. The machine was driven with either a test person on-board, or remotely using a laptop computer. An added load of 1000 kg was also used in some of the test drives. Some combinations of task scenarios were driven several times. Table 1 shows cases of these test drives on gravel. Table 2 shows the measured variables during these test drives which were then used in the analysis. Measured time series data were analysed using two different data sets, namely variables of machine driving behaviour (velocity/turnings) and hydrostatic transmission.

Table 1: Cases of test drives on gravel.

Case id	Description	Speed	Control	Load	Special definitions
6-1	Uphill	40 %	Person	X	
6-2	Downhill	20 %	Person	X	
6-3	Uphill-stop-uphill start	40 %	Person	X	
6-4	Uphill	40 %	Person	-	
6-5	Downhill	20 %	Person	-	
6-6	Uphill-stop-uphill start	40 %	Person	-	
7-1	Turn right - over obstacle	20 %	Computer	-	Obstacle: block of wood (right tyre)
7-2	Turn right - over obstacle	40 %	Computer	-	Obstacle: block of wood (right tyre)
7-3	Turn right - over obstacle	20 %	Computer	-	Obstacle: block of wood (left tyre)
7-4	Turn right - over obstacle	40 %	Computer	-	Obstacle: block of wood (left tyre)
8-1	Driving straight	40 %	Computer	X	Slow: rise time 1.5 s to max. speed
8-2	Driving straight	40 %	Computer	X	Fast: rise time 0.5 s to max speed
8-3	Driving straight	40 %	Person	X	Slow: rise time 1.5 s to max. speed
8-4	Driving straight	40 %	Person	X	Fast: rise time 0.5 s to max speed
8-5	Driving straight	40 %	Computer	-	Slow: rise time 1.5 s to max. speed
8-6	Driving straight	40 %	Computer	-	Fast: rise time 0.5 s to max speed
8-7	Driving straight	40 %	Person	-	Slow: rise time 1.5 s to max. speed
8-8	Driving straight	40 %	Person	-	Fast: rise time 0.5 s to max speed
9-1	Driving straight	40 %	Person	-	Emergency braking
10-1	Straight-right-straight-left-straight	20 %	Person	-	
10-2	Straight-right-straight-left-straight	40 %	Person	-	
10-3	Straight-right-straight-left-straight	20 %	Person	X	
10-4	Straight-right-straight-left-straight	40 %	Person	X	
10-5	Straight-right-straight-left-straight	20 %	Computer	X	
10-6	Straight-right-straight-left-straight	40 %	Computer	X	
10-7	Straight-right-straight-left-straight	20 %	Computer	-	
10-8	Straight-right-straight-left-straight	40 %	Computer	-	
11-1	Directly reversing	40 %	Computer	-	Slow: rise time 1.5 s to max. speed
11-2	Directly reversing	40 %	Computer	-	Fast: rise time 0.5 s to max speed
12-1	Simulation of normal operation		Person	-	

Table 2: Measured variables during test drives.

	Signal	Range	Unit
	Steering reference	-16384…16383	-
	Frame angle	193,1…117,1	°
Driving behaviour	Rotational speed of front left tyre	-∞...∞	r/s
	Rotational speed of front right tyre	-∞...∞	r/s
	Rotational speed of rear left tyre	-∞...∞	r/s
	Rotational speed of rear right tyre	-∞...∞	r/s
	Diesel rotational speed reference	0…2200	rpm
	HST pump angle reference	-1000…1000	‰
Hydrostatic transmission	Diesel rotational speed	0…2200	rpm
	Diesel load	0…100	%
	HST pump measured angle	-1000…1000	‰
	Pressure at port A	0…500	bar
	Pressure at port B	0…500	bar

5 Recognition of operating states from measurement data

Measurements were scaled to zero mean and unit variance before the analysis. For this, measurement statistics from eleven representative test drives were collected. Moving average filtering was also used to get rid of situations when segment is intercepted by noise in measured variables.

Next, the pre-processed measurements were segmented using sliding window segmentation with piecewise linear regression. Cumulative squared error function with the threshold 0.3 was used to define the edges of segments. Data from all the test drives were segmented similarly using two groups of measurements. An example of segmentation of one test drive in the case of driving behaviour related measurements (frame angle and rotational speed of front left) is shown in fig. 4.
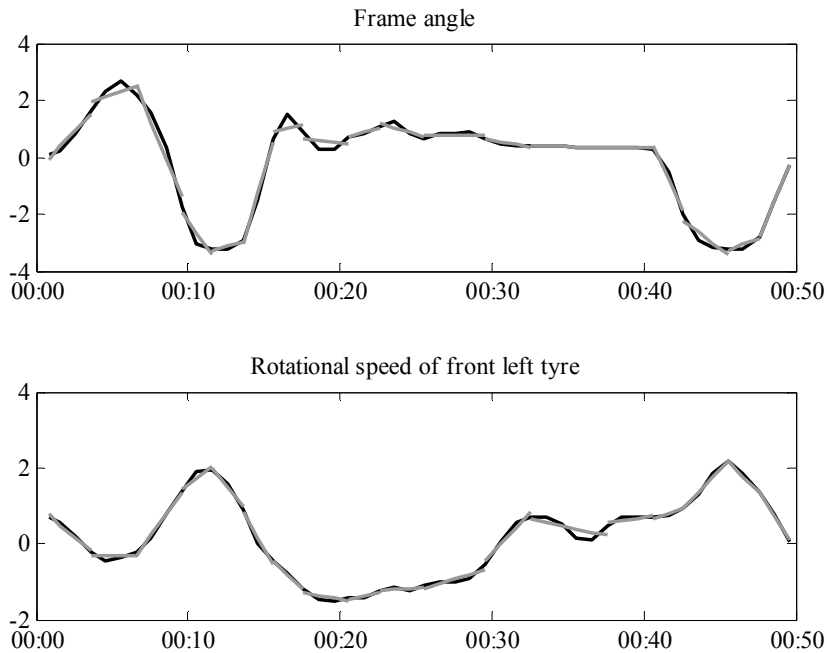

Figure 4: Time series segmentation of case 12-1. Frame angle and rotational speed of front left tyre are shown as examples. The measurements are normalized to zero mean and unit variance for segmentation and the length of time series is 50 s.

The segments of eleven representative test drives were clustered using k-means clustering with Davies-Bouldin validation index [10] and segment slopes as features. In tab. 3 is defined the basic steps of k-means algorithm.

Table 3: Steps of K-means clustering.

K-means clustering
1. Initialize: select a set of K candidate cluster centres
2. Assign each data point to the closest cluster centre
3. Set the cluster centres to the mean value of the points in each cluster
4. Repeat Steps 2 and 3 for a fixed number of iterations or until there is no change in cluster assignments

The grouping in k-means algorithm is done by minimizing the sum of squares of distances between data and the corresponding cluster centres according eq. 2, where K is the number of groups and N is the number of sample/feature vectors, x_n is sample/feature vector and m_i is the mean of the data points in set i, i.e. prototype of cluster i and it is calculated according eq. 3.

Clustering was repeated five times for every number of clusters and the result with minimum quantization error was selected. The number of clusters was varied from 10 to 30

clusters and the one giving minimum validation index value was selected. In this study, the optimal number of clusters for driving behaviour variables was found to be 27, and for hydrostatic transmission variables 29.

As a result of clustering, the segments of representative test drives were clustered and a prototype vector for each cluster was computed. These models were used to classify the segments of other test drives. Prototype vectors of clusters for driving behaviour measurements are shown in fig. 5. The lines in the subplots describe the behaviour of the signal in a particular state. The signal is constant or increases or decreases at a certain rate. A state is a combination of the patterns of the selected variables. For example, in state 27 (the far right column) the values of the 1st and 6th variable increase and the others decrease.

$$J = \sum_{i=1}^{K} \sum_{n=1}^{N_i} \|x_n - m_i\|^2 \qquad (2)$$

$$m_i = \frac{1}{N_i} \sum_{n=1}^{N_i} x_n \qquad (3)$$

The segmentation results were further processed by combining consecutive segments belonging to the same class. The segment class labels generate an event sequence. For each event a time stamp is computed by averaging sample time stamps of the segment.
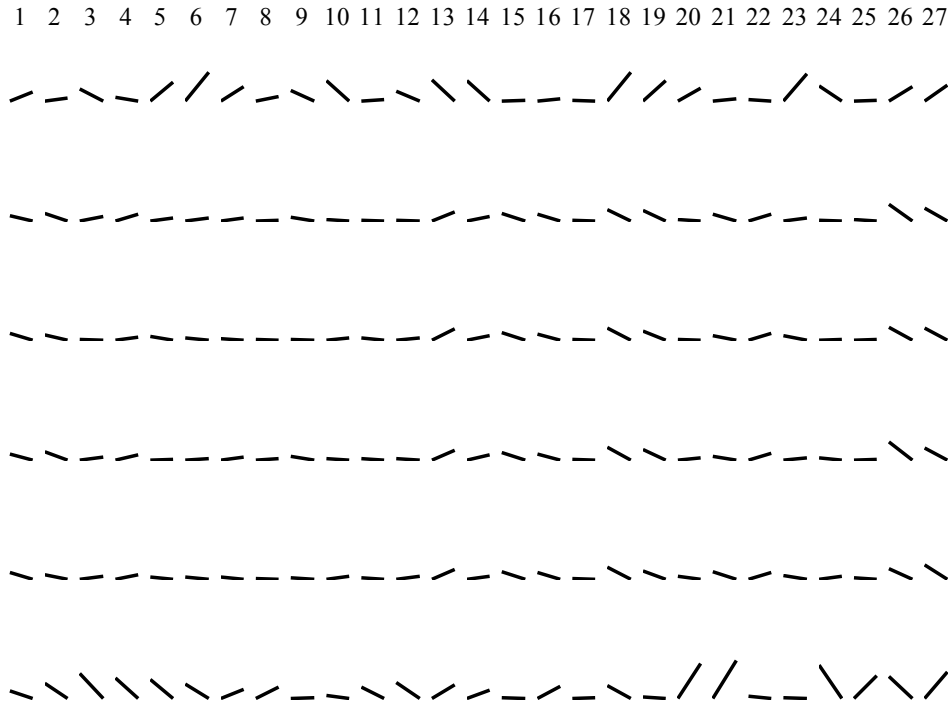

Figure 5: Prototype segments for 1st variable set. The rows correspond to variables and columns to operating states.

Two event sequences were built for each test drive. Event labels generated using hydrostatic transmission related measurements are shifted by the number of segment classes for driving behaviour measurements to obtain event labels

which are unique over both measurement sets. As a result, hydrostatic transmission related events are labelled between 28 and 56. In fig. 6 classified segments with the event labels of one test drive are shown.
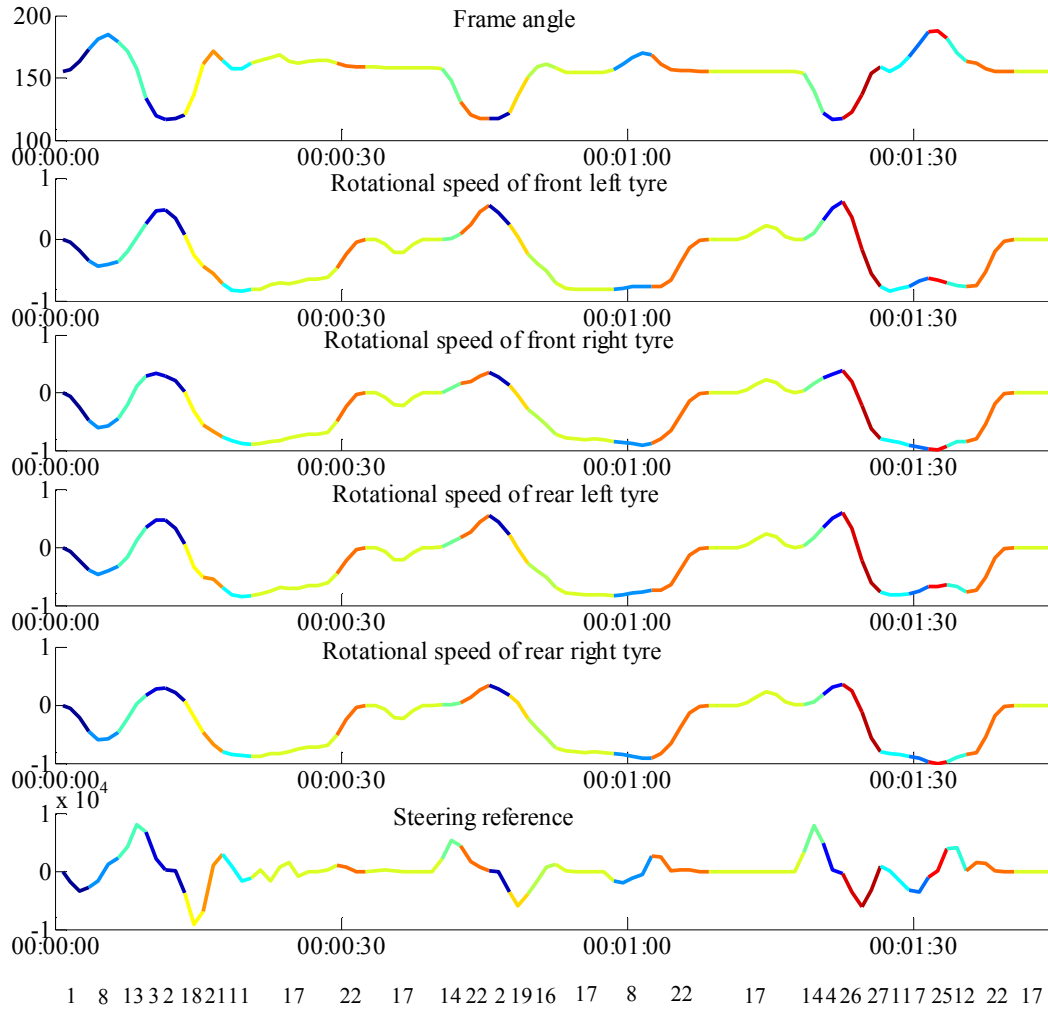
Figure 6: Time series (case 12-1) of length 1min 45s are plotted using different colours for each event. The labels of events are shown on the bottom row.

The probabilities of the events which occur in the test drives are shown in fig. 7. Some of the segment classes are more probable than others. Usually, the slopes of segments in such classes or states are near zero. Thus, the classes correspond to constant movement or operation without any changes.

For each test drive two sequences corresponding to the measurement sets were combined. The new sequence has events with labels from 1 to 56.

Frequent serial episodes of length 2 were searched for from these sequences. The time stamps were converted to discrete values, because the original algorithm is designed to process these.

Serial episodes with a certain probability were found. The probabilities were further converted to conditional probabilities using eq. 1 to find out how probable it is that one state occurs when a certain state has already occurred. In tab. 4 serial episodes with the highest conditional probability are shown. The probability of an episode in the studied sequences is also shown, as well as the overall probability of the first events of episodes.

According to the first rule in the table, state 45 occurs nine times and there is an over 88% probability that state 15 will follow.
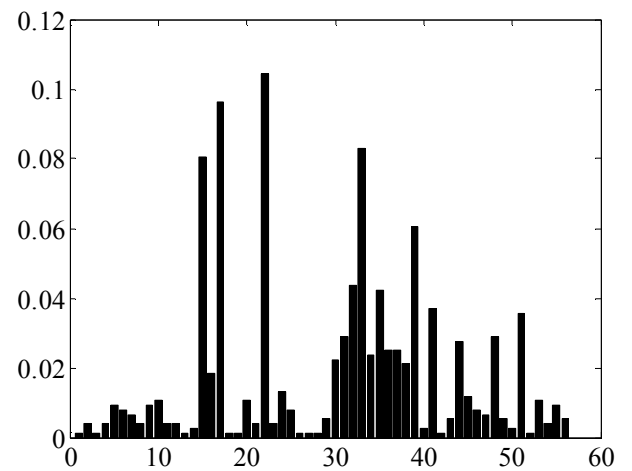

Figure 7: Probabilities of all events over all cases. There are total 56 different events: 27 for 1st variable sets and 29 for 2nd variable sets.

385

Table 4: Serial episodes and the event probabilities used to compute episode conditional probabilities. There are in total 758 events. Episodes with first event occurring three times or less are left out as well as episodes with conditional probability of less than 0.3.

A → B	P(A, B)	P(A)	n(A)	P(B|A)
45 → 15	0.01055	0.01187	9	0.8889
46 → 15	0.00660	0.00792	6	0.8333
31 → 22	0.02243	0.02902	22	0.7727
44 → 15	0.02111	0.02770	21	0.7619
53 → 22	0.00792	0.01055	8	0.7500
43 → 22	0.00396	0.00528	4	0.7500
55 → 22	0.00660	0.00923	7	0.7143
30 → 22	0.01583	0.02243	17	0.7059
51 → 15	0.02243	0.03562	27	0.6296
20 → 10	0.00660	0.01055	8	0.6250
48 → 17	0.01715	0.02902	22	0.5909
41 → 22	0.02111	0.03694	28	0.5714
5 → 33	0.00528	0.00923	7	0.5714
6 → 20	0.00396	0.00792	6	0.5000
6 → 33	0.00396	0.00792	6	0.5000
34 → 37	0.01187	0.02375	18	0.5000
22 → 39	0.04881	0.10422	79	0.4684
37 → 31	0.01055	0.02507	19	0.4211
24 → 33	0.00528	0.01319	10	0.4000
39 → 17	0.02375	0.06069	46	0.3913
10 → 41	0.00396	0.01055	8	0.3750
17 → 33	0.03430	0.09631	73	0.3562
35 → 17	0.01451	0.04222	32	0.3438
25 → 32	0.00264	0.00792	6	0.3333
35 → 32	0.01319	0.04222	32	0.3125
15 → 48	0.02507	0.08047	61	0.3115

6 Analysis of operation using recognized states

After recognition of the operating states, these can be further analysed to study the operation of the machine. On the basis of earlier research results [3] it was noticed that the biggest effects of different fault states were at the transient stages of the hydraulic system in which the changes in the pressure and volume flows were at their highest. Therefore analysis will be focused on operating states where the slopes of the feature vectors are steep and when changes in the analysed variables are at their biggest.

The quantization error method [3,12,13] was used to study the operation of the machine, and it is based on the distance calculation, Euclidian distance, which is shown in eq. 4.

Here, the neuron whose weight vector is closest to the input sample vector x is called the BMU, denoted by c.

$$e_q = \|x - m_c\| = \min_i\{\|x - m_i\|\} \qquad (4)$$

After this, a certain threshold can be set which determines the greatest distance on which recognition occurs. So when the method is tested the distances between the sample vectors from the testing data and all the cluster prototype vectors are calculated. If the minimum distance is bigger than the threshold value set beforehand, then this sample vector is treated as an anomaly, which can be either a new operating or a fault state.

After anomalies have been detected and proved to be faulty operating states, data from these states can be used to detect and identify them in the future. However, in this study there were no data available from faulty states.

Figure 8 shows the quantization error of all the classified segments of the 1st data set. There are altogether 302 segments. In turn, fig. 9 shows the quantization error of the 2nd data set. There are altogether 377 segments. Based on these quantization errors, thresholds can be set to separate possible anomalies from normal states. In these figures few larger quantization errors can be seen, which complicates the detection of anomalies. In this case, some of the anomalies could be mixed with normal states. This could perhaps still be improved by selecting a different number of clusters or even another clustering method. Only the k-means clustering method was used in this study. But also longer and more versatile test drives than those which are used here would improve the situation and reduce the number of spikes in the quantization error.

Figure 10 shows the quantization error of state number 22, which is the most probable state in the test drives: see fig. 7. 84 segments were classified as state 22. Here, only a few of the quantization error values are significantly larger than the rest of the values. Thus, in case of state number 22 the definition of the threshold to detect anomalies is more straightforward.
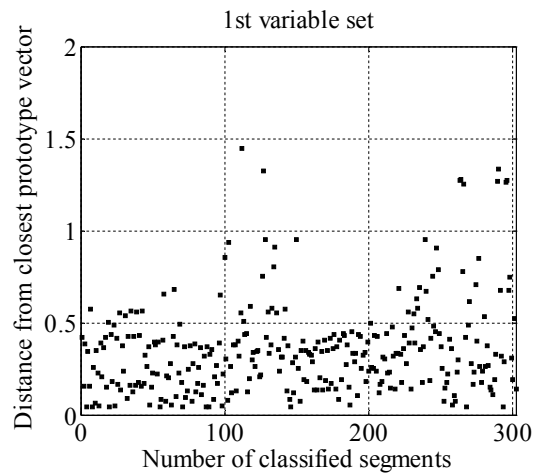

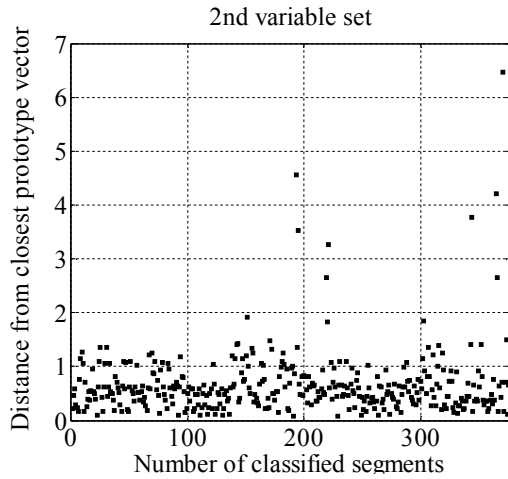*Figure 8: Quantization error of all classified segments of 1st data set.*

Figure 9: Quantization error of all classified segments of 2nd data set.
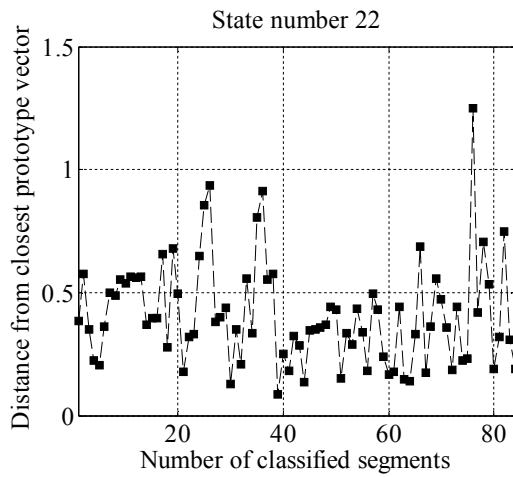

Figure 10: Quantization error of state number 22.

7 Conclusions

The operating states of a medium-sized mobile machine were studied to find causal connections between consecutive operating states and transition probabilities from 1st state to 2nd state, and to focus the analysis of operation on certain operating states.

Altogether 56 different operating states were found from measurement data using sliding window method with piecewise linear regression for time series segmentation and k-means algorithm for clustering and classification of pre-processed segments. The analysed data were comprised of two different data sets (machine driving behaviour and hydrostatic transmission).

Causal connections between consecutive operating states were found, using the Apriori algorithm, to which transition probabilities were then calculated. In this study consecutive serial operating states of length 2 were searched for from the recognized operating states. If desired, also longer and/or more complicated sequences can be searched for.

The operating states that have the steepest slopes of the feature vectors were found and these can be further analysed

to study the operation of the machine using the quantization error method. It is noted that this is an on-going work.

The data-driven methods described in this study can be implemented to different kinds of mobile machines. Insufficient sensor information may limit the number of applications, but the analysis methods in general are not restricted to a specific machine type. The thresholds in segmentation and anomaly detection need be defined based on specific applications and need specific knowledge about the operation of the system. The selection of critical measurement signals describing the operation of the machine also requires knowledge about the machine.

Nomenclature

Designation	Denotation	Unit
A	1st state	[-]
B	2nd state	[-]
c	Best-matching unit (BMU)	[-]
e_q	Quantization error	[-]
J	Sum of squares clustering function	[-]
K	Number of groups/clusters	[-]
m	Prototype vector	[-]
$\boldsymbol{m}_c$	Prototype vector chosen as BMU	[-]
N	Number of sample vectors	[-]
n(A)	Number of state A	[-]
P(A)	Probability of state A	[-]
P(A, B)	Sequence probability	[-]
P(B|A)	Conditional probability	[-]
$\boldsymbol{x}$	Data/feature vector	[-]

References

[1] Uusisalo, J. 2011. A Case Study on Effects of Remote Control and Control System Distribution in Hydraulic Mobile Machines. Dissertation. Tampere, Finland. Tampere University of Technology. Publication 960. 111 p.

[2] Durrant-Whyte, H. 2005. Autonomous Land Vehicles. Proc. IMechE., *Part I: Journal of Systems and Control Engineering*, 1, vol. 219, no. 1, pp. 77-98.

[3] Krogerus, T. 2011. *Feature Extraction and Self-Organizing Maps in Condition Monitoring of Hydraulic Systems*. Dissertation. Tampere, Finland. Tampere University of Technology. Publication 949. 126 p.

[4] Ruotsalainen, M., Jylhä, J., Vihonen, J. and Visa, A. 2009. A Novel Algorithm for Identifying Patterns from Multisensor Time Series. In Proceedings of the *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009*, Los Angeles, California, USA, 31 March - 2 April, 2009, vol. 5, pp. 100-105.

[5] Keogh, E., Chu, S., Hart, D. and Pazzani, M. 2011. An Online Algorithm for Segmenting Time Series. In Proceedings of *IEEE International Conference on Data Mining*, 2001, pp. 289-296.

[6] Mannila, H., Toivonen, H. and Verkamo, I. 1997. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 259-289.

[7] Backas, J., Ahopelto, M., Huova, M., Vuohijoki, A., Karhu, O., Ghabcheloo, R. and Huhtala, K. 2011. IHA-Machine: A Future Mobile Machine. In Proceedings of *The Twelfth Scandinavian International Conference on Fluid Power, SICFP'11*, Tampere, Finland, May 18-20, 2011, Vol. 1, No. 4, pp. 161-176.

[8] Huova, M., Karvonen, M., Ahola, V., Linjama, M. and Vilenius, M. 2010. Energy Efficient Control of Multiactuator Digital Hydraulic Mobile Machine. In Proceedings of 7th *International Fluid Power Conference Aachen,* Aachen, Germany, March 22 -24, 2010, vol. 1, 12 p.

[9] Theoridis, S. and Koutroumbas, K. 1999. Pattern Recognition, Academic Press, USA.

[10] Davies, D. and Bouldin, D. 1979. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 2, pp. 224-227.

[11] Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In Proceedings of the *20th VLDB conference*, 1994, pp. 487–499.

[12] Ahola, J., Alhoniemi, E. and Simula, O. 1999. Monitoring Industrial Processed Using the Self-Organizing Maps. In *Proceedings of the 1999 IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications, Kuusamo, Finland*, June, 1999. pp. 22-27.

[13] Alhoniemi, E., Hollmén, J., Simula, O. and Vesanto, J. 1999. Process Monitoring and Modeling Using the Self-Organizing Map. *Integrated Computer-Aided Engineering*, vol. 6, no. 1, pp. 3-14.