MONTAN
UNIVERSITÄT
LEOBEN

Chair of Mining Engineering and Mineral Economics

Master's Thesis

Loading and Crushing Assessment of a Blasted Muck Pile

Robert Schleicher, BSc

May 2019

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 13.05.2019

Unterschrift Verfasser/in
Robert, Schleicher
Matrikelnummer: 00735166

Master's Thesis

Loading and Crushing Assessment of a Blasted Muck Pile

A best practice manual for SLIM WP 6

Robert Schleicher

05.06.2019

Lehrstuhl für Bergbaukunde, Bergtechnik und Bergwirtschaft
Department Mineral Resources Engineering
Montanuniversitaet Leoben

A-8700 LEOBEN, Franz Josef Straße 18
Tel.Nr.: +43 3842-402-2001
Fax: +43 3842-402-2002
bergbau@unileoben.ac.at

Ehrenwörtliche Erklärung

„Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.“

”

Aus Gründen der leichteren Lesbarkeit wird auf eine geschlechtsspezifische Differenzierung in dieser Arbeit, wie z.B. TeilnehmerInnen, verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter.

Preface, Dedication and Acknowledgements

Firstly, I would like to thank the Head of the Chair of Mining Engineering and Mineral Economics, Univ.-Prof. Dipl.-Ing. Dr. mont. Peter Moser for the opportunity to develop this Master thesis.

Furthermore, I would like to thank Dipl.-Ing. Dr. mont. Philipp Hartlieb and Dipl.-Ing. Thomas Seidl for their guidance and excellent advice during this Master thesis and their scientific review. Both have given a substantial contribution to this thesis.

Zusammenfassung

Der stets steigende Bedarf an Rohstoffen, insbesondere in den osteuropäischen Ländern, hat zur Folge, dass immer mehr Bergbaubetriebe eröffnet werden müssen bzw. bestehende Betriebe ihre Produktion erhöhen. Gebunden an die Lagerstättenvorkommen, sowie von den umwelttechnischen und wirtschaftlichen Rahmenbedingungen abhängig, ist die Situierung des Bergbaubetriebes oftmals nicht fernab der urbanen Bebauung möglich. Dies führt oft aufgrund der mit dem Abbau einhergehenden Emissionen zu Spannungen zwischen dem Betrieb und der Bevölkerung. Aus diesem Grund wurde ein EU - Projekt unter dem Titel SLIM (Sustainable low impact mining solution) lanciert, welches das Thema der Emissionen und des Energieverbrauchs in den betroffenen Betrieben behandelt. Als ein Teil dieses Projektes, gilt es Einfluss Faktoren und Prozessinteraktionen, definiert als Key Performance Indikators, zu finden, welche es ermöglicht jene Prozesse bzw. die Änderungen daran zu klassifizieren und hinsichtlich ihrer Emissionsreduktion zu beurteilen. Der Erzberg in Eisenerz dient hier als Testbergbau, an dem ein mehrwöchiger Messzyklus durchgeführt wird. Dabei wird der Einfluss von veränderbaren Parametern wie Sprenggeometrie, Sprengstofftyp, o.ä auf die nachgeschalteten Prozesse wie Korngrößenverteilung oder Ladezeiten untersucht.

Um einen Rahmen um diese Diplomarbeit zu legen, wird zu Beginn ein Einblick in international existierende Studien gegeben, um im Anschluss darauf die Vorgehensweise und Ergebnisse einer ersten Test-Messkampagne zu beschreiben. Als Bestandteil dieses Messprogramms und Aufgabe dieser Arbeit ist es die Ladetätigkeit am Hauwerk in Verbindung mit den Brechern zu analysieren. Hierbei wird untersucht ob der sehr allgemein gefasste Begriff der „Ladbarkeit (eines gesprengten Hauwerks)“ in Zahlen gefasst bzw. näher quantifiziert werden kann. Basis hierfür dienen Telemetriedaten der am jeweiligen Hauwerk zugeordneten Mulden und Radlader, sowie die Energiekurven der Brecher.

Um diese Fülle der bei diesen Tätigkeiten erhaltenen Datensätze möglichst effektiv zu verarbeiten, wurde im Zuge dieser Arbeit eine semi-automatisierte Daten-Auswertsystematik entwickelt. Um die Telemetriedaten um eine weitere Datenquelle zu ergänzen wurde in dieser mehrtägigen Test-Messkampagne eine Kamera am Radlader installiert. Aus dieser Kamera sind nun detailliertere Erkenntnisse aus dem Ladezyklus erhältlich, wobei ergänzende Aussagen hinsichtlich der individuellen Einflüsse der jeweiligen Radladerfahrer möglich sind.

Abstract

The constantly increasing demand for raw materials, especially in the Eastern European countries, means that more and more mining companies have to be opened or that existing companies increase their production. Bound to the deposits and dependent on the environmental and economic conditions, it is often not possible to situate mining operations beyond urban development. Due to the emissions associated with mining, this often leads to tensions between the company and the population. For this reason, an EU project entitled SLIM (Sustainable low impact mining solution) was launched, which addresses the issue of emissions and energy consumption in the affected companies. As a part of this project, it is important to find influencing factors and process interactions, defined as Key Performance Indicators, which allow to classify and assess these processes and their changes with respect to their emission reduction. The Erzberg in Eisenerz serves here as a test mine where a several weeks lasting measuring campaign is carried out. Hereby, the influence of variable parameters such as blast site geometry or explosives type on downstream processes such as particle size distribution or loading times will be investigated.

As a framework for this thesis, an insight into existing international studies will be given at the beginning, followed by a description of the methodology and results of a first test measurement campaign. Part of this measuring program and task of this work it is to analyse the loading activity at the muck pile in connection with the primary crushers. In this context, it is examined whether the rather general term "loadability (of a blasted muck pile)" can be expressed in figures or quantified in a more precise form. This is derived from telemetry data of the trucks and wheel loaders assigned to the respective muck pile as well as the energy curves of the crushers.

In order to process this abundance of data sets obtained during these activities as effectively as possible, a semi-automated data evaluation system was developed in the course of this work. Within the scope of a several days lasting test measurement campaign, a camera was installed on the wheel loader as a further data source to supplement the telemetry data. From this camera, more detailed findings of the loading cycle are now available, whereby additional statements regarding the individual influences of the respective wheel loader operators are possible.

Table of Content

Ehrenwörtliche Erklärung I

Preface, Dedication and Acknowledgements II

Zusammenfassung III

Abstract IV

Table of Content V

1 Introduction 1
1.1 Open Pit Erzberg 3
1.2 Scope and Objectives 5

2 Literature review 6
2.1 On-site Observations 6
2.2 Loader Performance 9
2.3 Key Performance Indicators 16
2.4 Match factor 17

3 Methodology 18
3.1 Data collection 19
3.1.1 On-site observations 21
3.1.2 Video camera 23
3.1.3 Telemetry data - truck 26
3.1.4 Telemetry data - crusher 27
3.1.5 Telemetry data - Loader 28
3.2 Data processing 29
3.2.1 Data from video camera 29
3.2.2 Crusher telemetry 32
3.2.3 Truck telemetry 33
3.3 Combined data sets in Matlab 34
3.3.1 Script: EnergyCalculationCrusher 36
3.3.2 Script: MatchCrusherDataInputTXT.m 38
3.3.3 Script: MergeTable 41

4 Results 42
4.1 Semi-automated data analysis 42
4.2 Influence of the operator on the digging process 44
4.3 Correlation between digging and overall cycle times 47
4.3.1 Influence of the spatial variability 50
4.3.2 KPI assessment 52

5 Conclusion 55

6 Publication bibliography 56

7 List of Figures 58

8 List of Tables 60

10 Appendix A 61

11 Appendix B 62

12 Appendix C 64
12.1 ImportCrusherData.xlsm: 64
12.2 Matlab Scripts: 65
12.2.1 EnergyCalculationCrusher: 65
12.2.2 MatchCrusherDataInputTXT: 76
12.2.3 MergeTables: 88

1 Introduction

Compared to largest mining areas and producers worldwide, the economic exploitation of mineral deposits in the European Union constitutes a great challenge for each producing enterprise in the EU. Increasing population and rising prosperity leads to a rapid surge in raw materials demand. From an economic point of view and considering raw-material policy, it is indispensable to cover the raw material needs as far as possible from countries' own resources. The assurance of raw material supply within the EU is stipulated in the 2nd pillar of the European raw material initiative.

As it can be seen in Figure 1, small and middle-sized enterprises account for the major share of raw material supply in the EU. According to Eurostat 2018, 77% (14.662) of all enterprises in the mining and quarrying industry employed in 2016 0 to 9 persons per enterprise. 11% or 2.075 enterprises employed 10 to 19 persons, 8% or 1.435 enterprises employed 20 to 49 persons, 4% or 675 enterprises employed 50-249 persons and only 1 % or 216 out of 19.063 listed enterprises in the mining and quarrying industry, employed more than 249 persons in 2016. This means 99% of all enterprises falls in the category small and medium sizes enterprises (SMEs), which are defined by the European Commission as having less than 250 persons employed per enterprise.

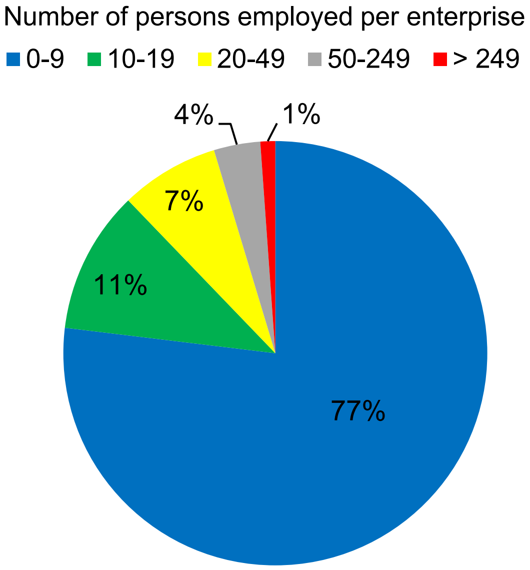


Figure 1: Share of enterprises in the mining and quarrying industry by employment size class in EU-26 (without Malta, Cyprus) in 2016 – Detail see Appendix - (Eurostat 2018)

Especially such SMEs are often close to inhabited areas, whereas environmental impacts like dust and noise often cause tensions between the population and the stakeholders of the mine. Furthermore, small and middle-sized enterprises are very sensitive to price fluctuations on the raw materials market.

To support these SMEs, including mines with chemically complex ore-forming phases, the EU launched a project under the title SLIM (Sustainable low impact mining solution) which shall have the objective to mitigate the Emissions and improve each of the exploitation process steps to ensure the supply of raw materials occurring in the EU.

The consortium consists out of 4 Universities (School of Mines and Energy of the Technical University of Madrid, acting as coordinator, Luleå Tekniska Universitet, Montanuniversitaet Leoben, Technische Universitaet Graz), 3 mines (Granada, Toledo (both Spain) and Eisenerz (Austria)) and 7 research institutions (3GSM, Benito Arno e Hijos, Bureau de Recherches Géologiques et Minières, Gate2Growth, Maxam Corp. International, Minpol GmbH, ZABALA Innovation Consulting).

The project itself consists out of 11 Work packages, shown in Table 1 where the Montanuniversitaet is involved in every Work package, except in WP 2.

WP 1	Coordination and Management
WP 2	Rock / Explosive Interaction
WP 3	Environmental Impact and Safety
WP 4	New Explosive Application Technologies for small Mineral Deposits
WP 5	Rock Mass Modelling Software
WP 6	Upstream System Validation and Plant Monitoring
WP 7	Economic and Environmental Assessment
WP 8	Dissemination and Communication
WP 9	Cluster with other projects and networking
WP 10	Public Awareness, Acceptance, Trust and Communication
WP 11	Ethic requirements

Table 1: Work packages of the SLIM project

1.1 Open Pit Erzberg

The Erzberg mine, located in the Alps of Eisenerz (Figure 2), about 30 km from Leoben, situated directly near the city Eisenerz, is the largest open-pit iron-ore mine in middle Europe and the world-largest Siderite ($FeCO_3$) deposit. The mine is operated by VA Erzberg GmbH, which generates around 50 mio. EUR a year by 230 employees.

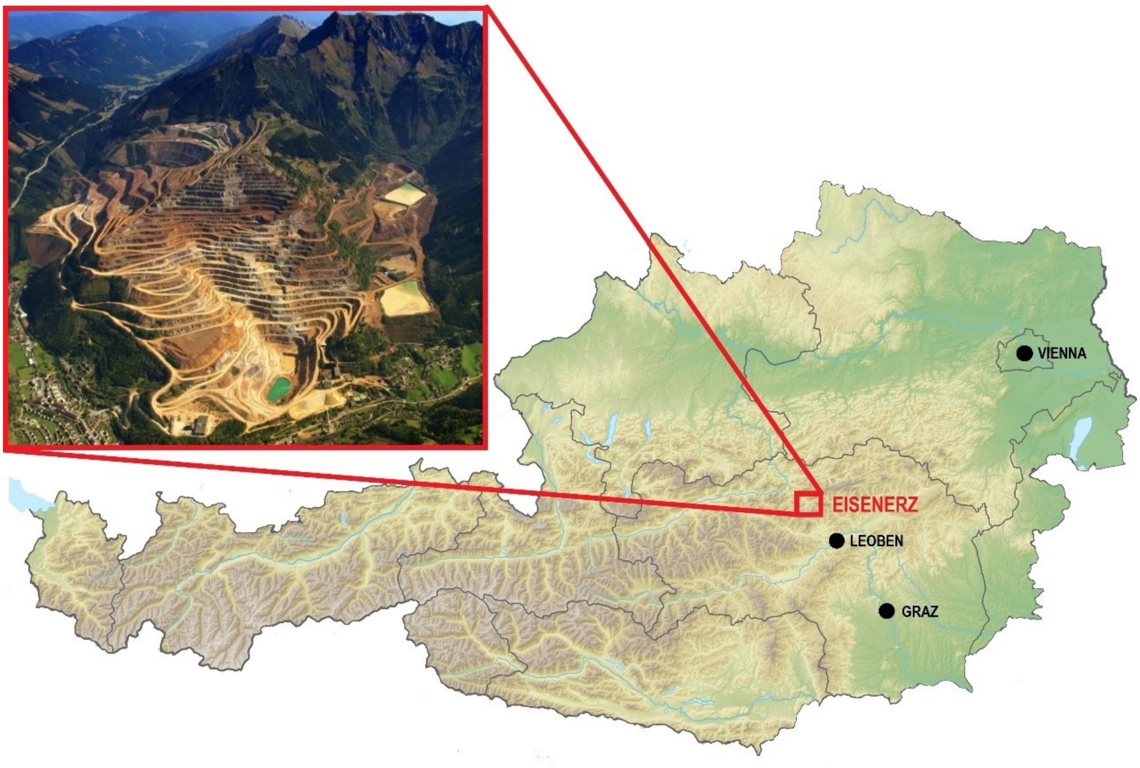


Figure 2: Location of Eisenerz and aerial view of Erzberg (adopted from: NASA Shuttle Radar Topography Mission 2018; BMVIT 2015)

The open-pit consists out of 30 benches, reaching from 700 meter above sea level to 1464 meter above level. Bench heights vary from 24 to 30 meter. 13 Trucks (Komatsu HD 985-5 & Komatsu HD 785-7) with a payload of 100 tonnes and 3 Komatsu wheel loaders (WA-800-3) with a 13m³ bucket volume, mine 12 Mio. tonnes rock a year, whereas 3 Mio. tonnes of recovered ore are generated.

The deposit, as illustrated in Figure 3, can be divided into 3 main tectonically zones. Firstly, the foot wall (coloured in green), which includes the green layers of volcanic porpyhroid. Interbedded strata containing carbonate and porphyroid (coloured in cyan) divides the hanging wall from the second layer, the iron-ore layers itself. These cloud shaped ankerite and siderite formations (coloured in orange) are embedded between massive carbonate formations, designated in “Sauberger Kalke” (coloured in yellow). The so called “Werfener Layers” (violet) represents the transition to the third geological zone, the hanging wall, which consists out of green and violet, mica-rich sandstone and shale layers.

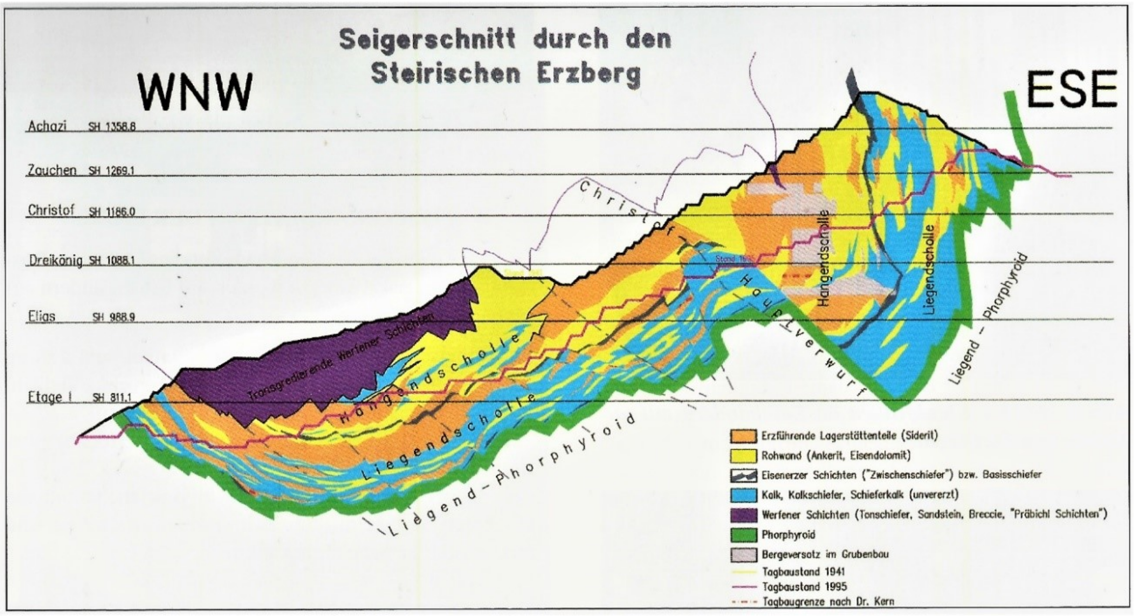


Figure 3: Geological section through the Styrian Erzberg (Montanhistorischer Verein Österreich 2012)

The ore deposit has striking lengths up to 600 m and dipping angels from 60-70 degrees. Siderite formations contain 38 – 41 % Fe, whereas Fe-concentrations in the ankerite formations reaches from 10-17%. The cut-off grade lies around 22% Fe which accounts approximately 25 % Siderite in the Siderite/Ankerite composition (Sideroplesit).

The run of mine product can be distinguished in 3 main categories. Ore with more than 30% Fe content (55% Siderite content in the blasted material) is considered as “high quality ore” (“Wascherz”-abbrev.: “WE”) which is crushed and directly transported to the dispatch. Ore containing 22 – 30 % Fe is designated as medium quality ore (“Zwischengut” abbrev.: “ZG”), which is enriched in the heavy medium separation plant to 34% Fe Content. Low quality ore, as third product, which contains less than 20% of Fe is considered as waste material (“Berge” abbrev.:”BE”) and dumped directly from the blasted muck pile to the waste-stockpile. (VA Erzberg GmbH 2018).

Despite the low iron ore content compared to other large iron ore mines worldwide, the mine guarantees a certain degree of independency from foreign ore imports for the domestic smelter “Voestalpine, so Erzberg is still of big interest for the steel-company.

1.2 Scope and Objectives

This thesis takes place within the scope of SLIM work package 6: “Upstream System Validation and Plant Operation monitoring”. The aim of this work package is to qualify the outcomes of work package 2 and 3 by defining Key performance Indicators (KPI). To guarantee its further proper examination, the KPIs are classified in Downstream and Upstream KPIs. Upstream KPIs like, Rock Characterisation, Explosive Characterisation, Blast Damage, Vibrations and Environmental Impacts should then combined with Downstream KPIs, as they are, the loadability of the muck pile, the crusher throughput, the normalised crushing energy, or the Loading cycle times.

The foundation of this KPIs constitutes a several weeks ongoing measuring campaign, whereas a numerous amount of field tests will be executed. In these field tests, existing process designs, such as blasting geometries or the use of explosives, are documented, and then analysed to see whether a change in these designs would have a positive or negative effect on ensuing operating processes. KPIs serve here as measuring instrument in order to classify the received output.

The superordinate objective of this thesis is to identify in which pre-condition the amount and type of existing datasets can be collected from operating mining units, in order to develop a best practice manual for the imminent measuring campaign.

For this purpose, a pre-measuring campaign was carried out, whereby such datasets of the ongoing loading and crushing processes are collected. These datasets are filtered and prepared in such a way that findings can be compared and reasonable interpreted.

As a result of this data collection and as a further key element of this work, it should be evaluated if it is possible to use mucking (hereinafter also called *digging*) times of the loader as a parameter to draw conclusions on the characteristics of the blasted material muck pile. It will be examined if the term *“loadability (of a muck pile)”* can be quantified in numbers and figures. As side effect of the data collection, the analysis may give insights into the loading and hauling process of VA Erzberg GmbH; such as the individual loading behaviour of the operators.

The outcome of this research conducts the basis of the next crucial part of the SLIM work package, the linking of obtained results to upstream mining procedures and characteristics, respectively, such as the blasting pattern, initial rock characteristics, as well as blast output or explosive characterisation. But this further examination should not be part of this work.

2 Literature review

The constant strive to optimize the production processes in mining can be an important factor when it comes to survive on the market. Thus, the consistent work on improvements on every small step, even if it seems not to affect the production cycle at the first sight, means a noticeable enhancement on the long run.

The characterising of the muck pile and the ensuing loading processes as a basis for improvement analyses, remains highly challenging. Various researchers who have conducted studies in modern mines, analysed these processes to gain more detailed knowledge of their production cycle and control their processes in a more efficient way.

Some of this literature is described in this section to explain the framework conditions under which this work is carried out. First, a general overview of the literature with similar content and tasks as in the SLIM project, is presented, followed by a more detailed description of the performance of the operators.

2.1 On-site Observations

Commencing by a work carried out by Beyglou (2016), who analysed the main mining activities (drilling, blasting, loading and crushing) within the STRIM project (Strategic innovation programme for the Swedish mining and metal producing industry), this summary should provide an overview of the tasks addressed here. According to the aim of this study, namely, *to develop a more efficient, environment-friendly, and less energy-intensive operations in the foreseeable future*, the STRIM project shows a strong similarity to the SLIM project, hence this research is presented shortly below.

The Author combines this study in 4 main research questions:

1) *How can rock mass characteristics be implemented in open pit operations to improve fragmentation and efficiency?*
2) *How can stress wave propagation theories be utilized to improve fragmentation and efficiency through the timing of blast holes?*
3) *What are the effects of fragmentation on the efficiency of downstream tasks and what fragmentation is most favourable?*
4) *How can fragmentation measurement techniques be utilized to improve efficiency?*

All tests were conducted in the Aitik Copper mine in Sweden, operated by Boliden Minerals AG.

Starting by the rock mass characterisation, photogrammetric mapping of specific bench faces where executed. Data gained out of the measurements (coordinates, dip, dip direction) is used to form structural domains, which are correlated to different initiation directions, and subsequently correlated to bucket weights to get an efficiency assessment. Two main insights where gained from these investigations. On the one hand certain initiation directions towards certain discontinuities should be avoided. On the other hand, blasts initiated a certain direction shows statistically a slightly better performance, which results, according to the authors, in a significant improvement on the long-run productivity.

Secondly, the impact of changes on delay times on stress wave interactions were investigated. To keep this complex topic to short findings, the authors concluded that the *effect of delay time on fragmentation was easily overshadowed by slight variations of specific charge. Some improvements were observed, […], but improvements were marginal and delay times did not show any significant effect on fragmentation. (Beyglou 2016)*

With the aid of Downstream Indicators, as **3rd** major aspect, the upstream indicators can be classified and determined how these processes, like rock characteristics or blast agents, influence the efficiency of mining activities. The author characterised these indicators by the muck pile swell, the loading activity and the crushing process. The loading activities were filtered in terms of certain exclusion criteria to get comparable datasets with no outliers and then analysed regarding the operator efficiency. The crushing cycle, as another subchapter of Downstream Indicators, was studied by linking the power draw curve to the telemetry data. Thus, the energy consumption per tonnage crushed material (kWh / tonnes) and the actual crusher throughput (to/s) are further main Key performance indicators in the downstream process.

Fragmentation as the 4th Important issue to be dealt with, is measured by two techniques. 2D image analysis as first technique measures the truck-bed, from which a size distribution curve can be created. Secondly, a 5 class-qualitative assessment system, based on visual comparison of the blasted material is used to correlate the fragmentation classes with several Key performance Indicators like the crusher throughput, the shovel fill factor or the crushing energy. Some dependencies were observed, to cite a few examples: the crushing energy consumption increases from very fine material to very coarse material, harder ore consumes more energy than softer ore, the fill factor decreases from fine material to coarse material.

Despite the fact that shock wave propagation, comminution and other parameters are also measured in the course of the SLIM project, the following literature now focuses on the loading cycle and the characteristics of the muck pile.

The performance of the loading cycle relies on several properties as they are also described in similar type in (Singh and Narendrula 2006). The parameters can be classified in 2 main groups, muck pile properties and operating properties. The first group specifies the characteristics and the spatial properties of the muck pile, which represent the result of the blast design and the geological properties. The second group describes the loader related properties, whereby machine design and operator proficiencies have an direct influence on the loading process.

Muck pile properties:

- **Muck pile characteristics:**
  - Physical properties:
    - Geometry (Height, Distance of throw, spread, loosening)
    - Fragmentation (Grain size distribution, Uniformity Index)
  - Mechanical properties:
    - Hardness
    - Density
    - Abrasiveness
    - Anisotropy
    - Cohesion
- **Spatial properties:**
  - Muck pile orientation
  - Manoeuvring Space
  - Floor conditions

Operating properties

- **Machine performance:**
  - Bucket size, shape and design
  - Machine size, capacity and power (Breakout force)
    -
- **Operator skills:**
  - Bucket fill
  - Cycle pattern (time, trajectories, strategy for muck pile attack)
  - Special skills (comminution of boulders, performance on wet floor, handling of trickling material)

Considering all these factors, fragmentation accounts as one of the most important parameter, as it is proved by several researchers (Ouchterlony 2003, Thurley et al.; D. Brunton et al. 2011; Ivanova 2015, Schimek 2015), although the degree of influence of fragmentation on the loading process is not consistent throughout the scientific community.

Fragmentation of the muck pile, as a descriptive term of the grain size distribution, can be interpreted in different ways. In most cases, the expression is used to describe the extent of the comminution. However, an important factor which must not be overlooked in the process is the uniformity of the grainsize distribution, summarised and well put to the point by (Danielsson 2016): *“A finer fragmentation with an unfavourable distribution might cause a more problematic digging scenario, since the muck pile characteristics might not be suitable (compacted)”* as well as “*the median fragment size should be kept low and the uniformity high […]*.”. In contrast, fines between the larger grain sizes can act as a lubricant, facilitating shovel penetration into the muck pile (Singh and Narendrula 2006).

Workman and Eloranta (2003) examined the impact of blast setups on energy consumption in the downstream comminution steps, with the aim of reducing energy consumption and thus costs. It is shown that a higher powder factor can lead to better downstream effects, but with reference to the consideration of influencing framework conditions such as drilling costs, labour costs, or blast agent costs. The authors conclude from this work that the *greatest energy saving available are in grinding due to the large change in particle size achieved,* as well as *the use of greater energy input in the blasting unit operation will often be less costly than expending the energy downstream.*

2.2 Loader Performance

Hendricks (1990) began already to record and analyse the work cycle of excavation machines. For this purpose, a complete monitoring system for test blasts was created, whereby the energy consumption of the excavation unit - in that case a mining shovel - the digging trajectories, the load weight and the cycle times were recorded. The fragmentation was assessed by a firstly introduced semi-automated image analysis system. Based on a microprocessor-aided monitoring equipment installed on the shovel, the author developed a diggability index for mining shovels. He concluded on the perception that “*data on dipper trajectory has demonstrated that variations in digging practices do exist arnongst an experienced group of shovel operators*.” Trying to define diggability in terms of operator related digging effort he stated, that “*Dig cycle times are not specifically related to digging effort* and as another useful conclusion*, that operator's mean dig cycle time varied only slightly between easy and hard digging situations.*

Additionally Kanchibotla et al. (1998) concluded that diggability depends also on muck pile shape an looseness. That means, if the muck pile shows good properties in terms of fragment size and the uniformity index, these advantages may be eliminated by adverse characteristics of the muck pile (compacted, confined).

Danielsson (2016) examined how variations in blast hole dimensions has an influence on fragmentation and LHD operation. The author investigates the operation at the Malmberget underground iron ore mine, situated in the northern part of Sweden mined by the method of

Sublevel caving. The analysis of logged production data and on-site filming of the loading cycle constitutes the same procedure as being carried out in this thesis. Muck transportation is utilized by LHDs. Like mentioned above, also the issue of fragmentation is discussed, however it is to be noted that the muck pile and also the mucking process takes place in a confined area, that renders obsolete the comparison of muck pile properties to the open pit operation at Erzberg.

In order to accurately capture the exact cycle times for classifying the loadability of the muck pile, a camera system was installed at the loading point. Together with the available telemetry data some correlations and insights have been obtained: The author concluded that "*Loading times, and in particular time spent at the muck pile is heavily governed by operator efficiency".* Considering the correlation between digging and cycle *times "[…] there is a substantial difference amongst operator in the digging phase although fragmentation and compaction factor of the muck pile is comparable".* Comparing operator proficiencies among each other the author noted that *"it is "[…] a multi-dimensional difference where different operators are differently efficient at different stages of the LHD operations".* Summarizing his thesis, he concluded that *"cycle time is not a very reliable indicator of neither loadability or fragmentation"*. The author argues this by the *"significant time difference amongst operators which is difficult to account for."*

Coming to the loader performance, Filla (2013) did a lot of studies on the influence or operators on loading procedures. He states that two major phases of loading influence the process the most. At the one hand the driving pattern constitutes a major influence on the cycle time and on the other hand the digging curve, both patterns strongly depend on operators' proficiencies. In terms of efficiency, the driven pattern can give insights to the capabilities of the operator. As shown in Figure 4 below, the characteristic driving pattern from end of the digging phase to dumping the loaded material onto the truck usually describes a V or in some cases a Y. This V/Y shaped pattern should take as long as it requires to lift the bucket from the filled position in the muck pile, reversing towards the turning point and to lift the bucket above the truck bed. To keep the cycle time as short as possible the operator should move backwards straight-line till the turning point and then move forwards with 1,5 tire revolutions, so that the truck is positioned to the muck pile to 120° – 135°.

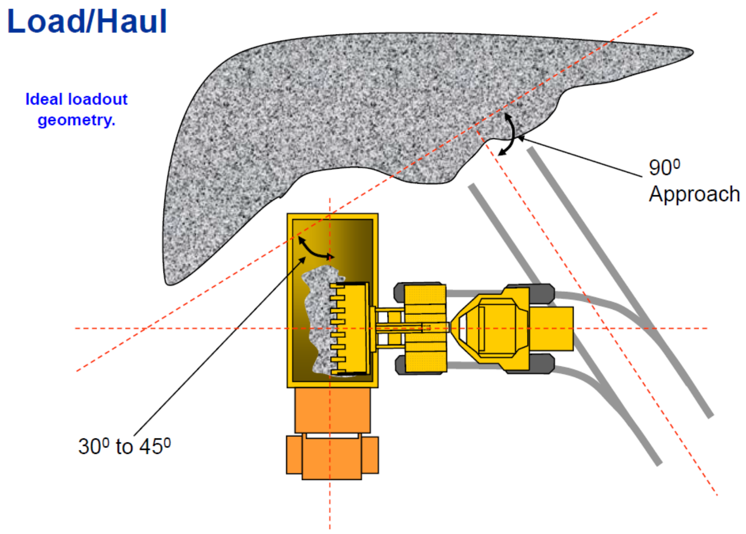


Figure 4: Ideal loading geometry (Petr and Hissem 2015)

Of course this ideal manoeuvring pattern is often not possible, so many variations can occur. These sub-variants are described in detail in Figure 5, where the preferred V-shaped pattern is replaced by a S-shaped leg.

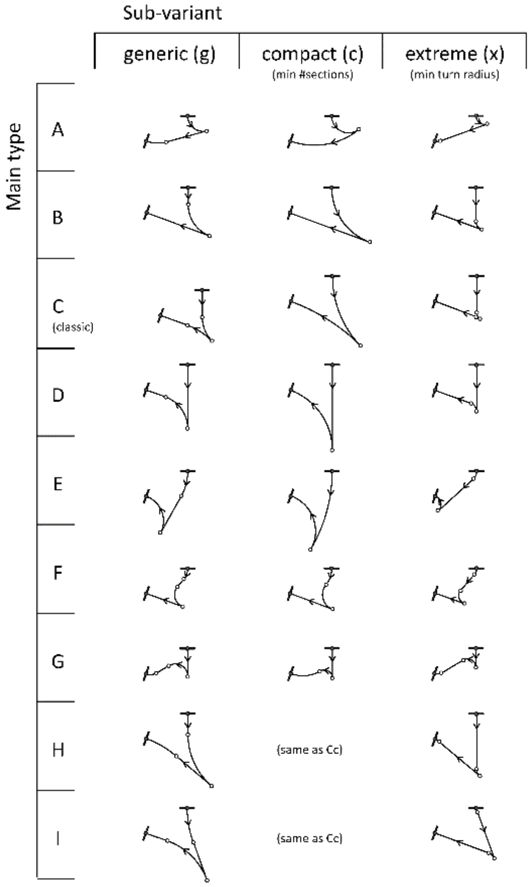


Figure 5: Main cycle types and sub-variants (Depicted load receiver position and orientation is just an example.) (Filla 2013)

Each time when positioning the bucket edge to the foot of the muck pile, the operator has different possibilities to fill the bucket. These different approaches stem from the numerous arrangement possibilities of the control elements, like the turning wheel, the joy sticks or the pedals, which leads to different digging behaviour.

Filla et al. (2014) describes the components which affect the bucket filling and how these components, such as digging forces, hydraulic pressures and the loader geometry, are correlated to each other. The whole loading operation of the machine and finally the digging process consists of 3 main movements, which must be coordinated by the operator in the best possible way:

- Forward movement of the machine
- upward movement of the boom
- Rotation of the bucket

The complex kinematic and kinetic relations of these individual control elements to each other are aptly formulated by the author:

The accelerator pedal controls engine speed, which affects both the machine's longitudinal motion and via the hydraulic pumps also affects the speeds of the lift and tilt cylinders. The linkage between the hydraulic cylinders and the bucket acts as a non-linear planar transmission and due to its design a lift movement will also change the bucket's tilt angle and a tilt movement affects the bucket edge's height above the ground.

In Figure 6 the just described interaction operator - machine is represented graphically and shows that the operator should have a good knowledge of the machine in order to coordinate each of these control elements in such a way that the loading cycle runs as effectively as possible. The coordination and the proper handling of all these factors can be summarised under the term *operator skills*, which can be expressed either by the fuel consumption, or by the time it takes to fill a shovel. The analysis of these two performance parameters is expressed by the author's finding that especially the period of bucket filling accounts for 35-40% of the total fuel consumption per cycle, although the scooping operation takes just 25% of the cycle time.

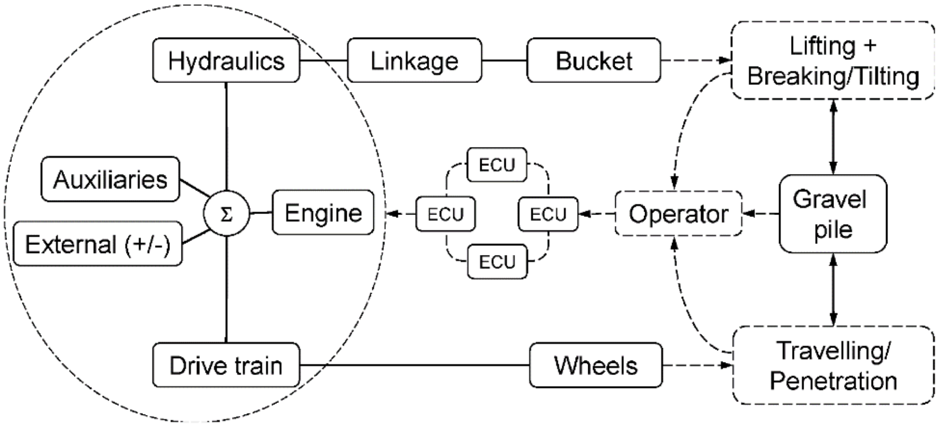


Figure 6: Simplified power transfer and control scheme of a wheel loader (Filla et al. 2014)

Furthermore, the path of the bucket edge in the muck pile during filling determines the efficiency during this process. This path, also called trajectory, strongly influences the energy consumption and scooping time and can be described in 4 main types of such trajectories which are described as follows.

- **Type A:**
  Developed by professional machine instructors, this is an idealised strategy, where the bucket is moved parallel to the slope. At reaching the maximum height, which is governed by the length of the arms, the bucket is tilted back and retracted:

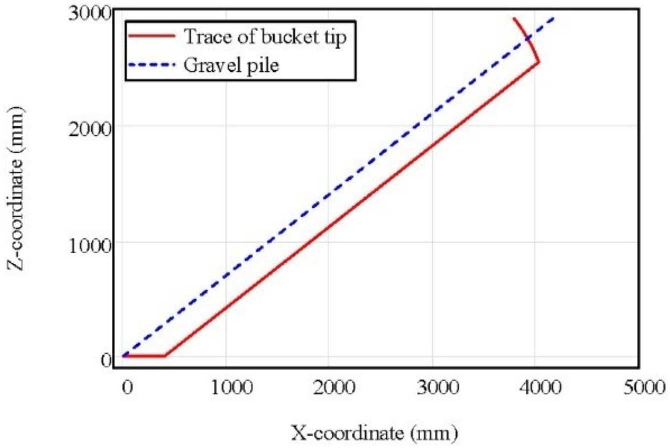


Figure 7: Trajectory Generation: Slice of constant thickness (Filla et al. 2014)

- **Type B:**
  As the most simplified and not often executable, this pattern describes a straight forward push into the muck pile, reaching its maximum traction, then tilting back and retracting.

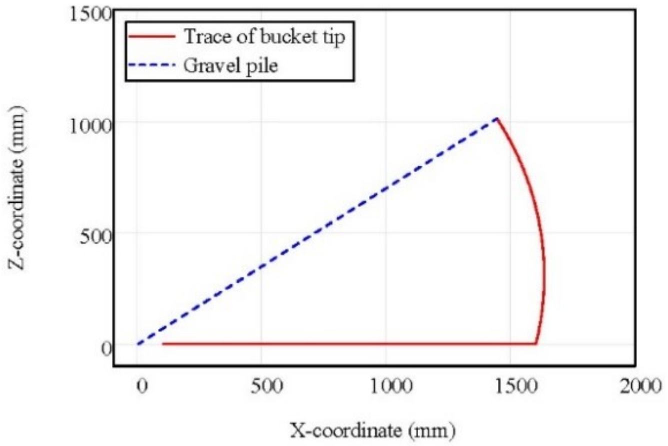


Figure 8: Trajectory Generation: horizontal penetration and tilting (Filla et al. 2014)

- **Type C:**
  Similar to type A, this trajectory is characterized by a parabolic curve. This pattern constitutes the most used one beside Type D.

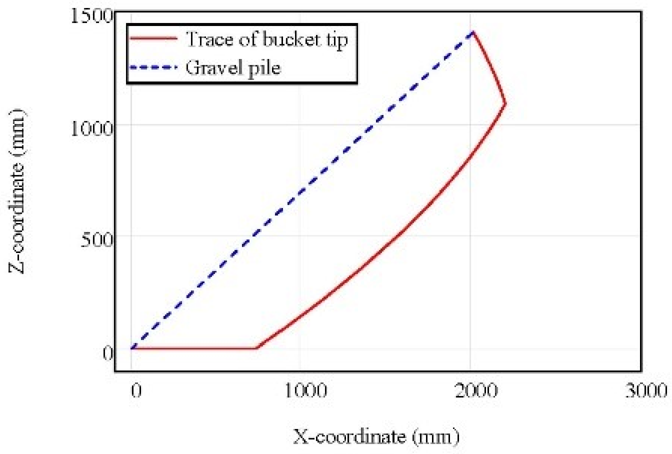


Figure 9: Trajectory Generation: Parametric parabola (Filla et al. 2014)

- **Type D:**
  As shown below this pattern is characterized by a stepwise curve. While scooping each step the wheel loader does not move, only the bucket is tilted and lifted. A new step is induced by driving forward moving into the muck pile.

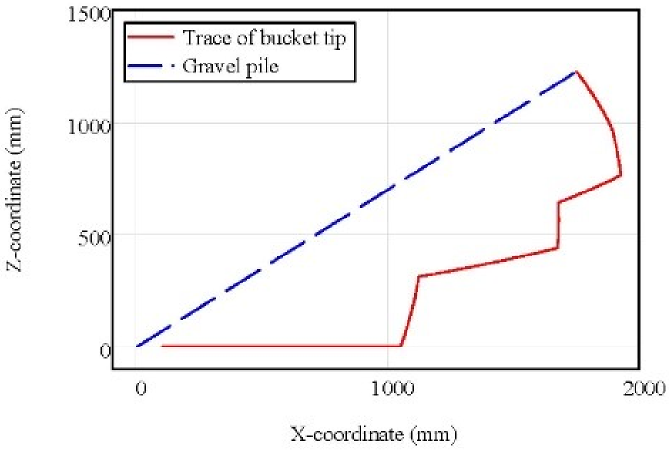


Figure 10: Trajectory Generation: stepwise filling of the bucket (Filla et al. 2014)

All these different approaches and influencing factors which directly influence the loading cycle and thus the performance are very strongly dependent on the operator's ability.

Also (Bogunovic and Kecojevic 2011) summarised in the field of mining shovels, that especially the mucking process is strongly influenced by operator performance.

Literature research shows clearly that the loading process, especially the digging phase is not that easy to evaluate and to classify as it seems at the first sight. Differences in the loading cycle are likely to occur more by the human factor, which can be expressed rather in operators experience, than in muck pile characteristics, spatial properties or machine performance, although these factors cannot be neglected.

Thus, the question is, whether it is possible to indicate a difference in operator experience in terms of numbers and figures.

2.3 Key Performance Indicators

Gaining lots of datasets constitutes nowadays not a big challenge anymore. Modern equipped hauling and loading units send in second-intervals nearly every action of the affected unit to data bases. Such big data sets must now be analysed and transformed to meaningful parameters, so called key performance indicators (KPI), which further can be used for a process interpretation at multiple levels and for several activities, like health and safety, the environment, and coming into effect in this thesis, the production and costs. Depending on the type of process and on the required outcome, it is crucial to define the correct and for the process suitable KPIs. By just defining arbitrary parameters out of the enormous datasets without gaining meaningful conclusions out of it, gathering big data sets is futile effort. Furthermore, it is vital to understand that KPIs are not the product of a closed system, as it seems at first sight, but rather as a result of an interlocked process chain, hence the interpretation and specification of KPIs should be done in a more global scale. Therefore the definition for KPIs*: "Measures that ensure you're staying on track to make the money you planned to make; sustainably, in the short, medium and long term"* (James Stoddart 2016) is in agreement with the assigned target of the local mine. This means a KPI which is outside of a determined boundary does not necessarily mean a loss in productivity, it is more an indicator to have a closer look at the concerned process step.

2.4 Match factor

KPIs as a performance measurement tool can be supplemented by the so-called *Match factor,* which represents another performance indicator of the whole transportation and loading fleet. Although this parameter is not suitable for assessing the loadability and the exact performance of a single loader, this factor can be useful for assessing the utilization of the loader and to show a tendency if the operator is experiencing a high workload.

In (Burt and Caccetta 2018) *the match factor (MF)* is defined as *the actual truck arrival rate to loader service time.* As it can be seen in Figure 11 a MF smaller 1 indicates that the loader is underutilized, whereas a MF greater 1 indicates that the trucks must queue until they are getting loaded.

The match factor is derived by following equation:

$$\boldsymbol{MF = \frac{t_T \times n_T}{t_L \times n_L}}$$

Whereas:

t_T … time taken to load the truck

t_T … time taken to load the truck

n_T … Number of Truck

n_L … Number of Loaders

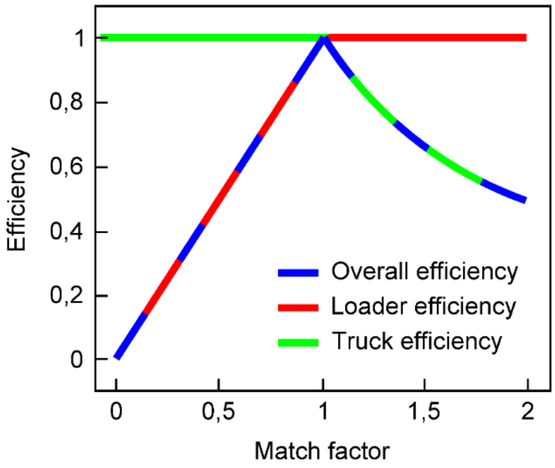


Figure 11: Match factor (Burt and Caccetta 2018)

This means subsequently, if the Match factor is smaller than 1, the production can be increased by adding a truck into to loading cycle, so that the utilization of the loader increases. On the other hand, a match factor greater than 1 indicates that trucks must queue for getting loaded - the loader is operating on its full capacity. Nevertheless, production can be increased to a certain amount by detailed analysing of the loading sequence to obtain potential improvements.

3 Methodology

The main mining activities like blasting, loading and transport and comminution represents the controllable component of a mine, which is in contrast to the only uncontrollable component of a mine, namely the rock mass. The task is now to analyse these controllable process steps on its efficiency to gain the greatest possible benefit of the contemplated cycle.

This chapter elucidates how the raw-data was collected from these controllable process units, how this raw data was prepared, and which approach was chosen to analyse several process units. From these units, like the loader, the truck and the crusher, datasets of different extent can be extracted. Several filters and criteria were used and applied to prepare these datasets, till they can be further processed together with other data.

To obtain equal datasets and guarantee a useful comparison between the loading cycles only one truck-type was observed, since several models are in use. In order to facilitate a reasonable comparison between trucks, only telemetry data of the same truck-type were extracted and further processed.

By acknowledging the fact that fragmentation is a key part in assessing the digging effort, at this point it should be noted that the detailed analysis of fragmentation is on one hand not the focus of this thesis and secondly, the photogrammetric method of measuring the grainsize distribution has not been implemented so far at this stage of the project, hence, this factor is left aside in this thesis.

3.1 Data collection

As already mentioned in the introduction, the extracted material consists of 3 quality levels, rich ore, medium ore and waste rock. The waste product is classified according to the deposit model before blasting and then transported directly from the muck pile to the waste dump, as it can be seen in Figure 12. Thus, only downstream data of material > 20% iron ore content is available. In order to guarantee the daily production and also the required quality, several rounds are always active simultaneously, which are either mined sequentially or at the same time, in coordination with the processing plant.

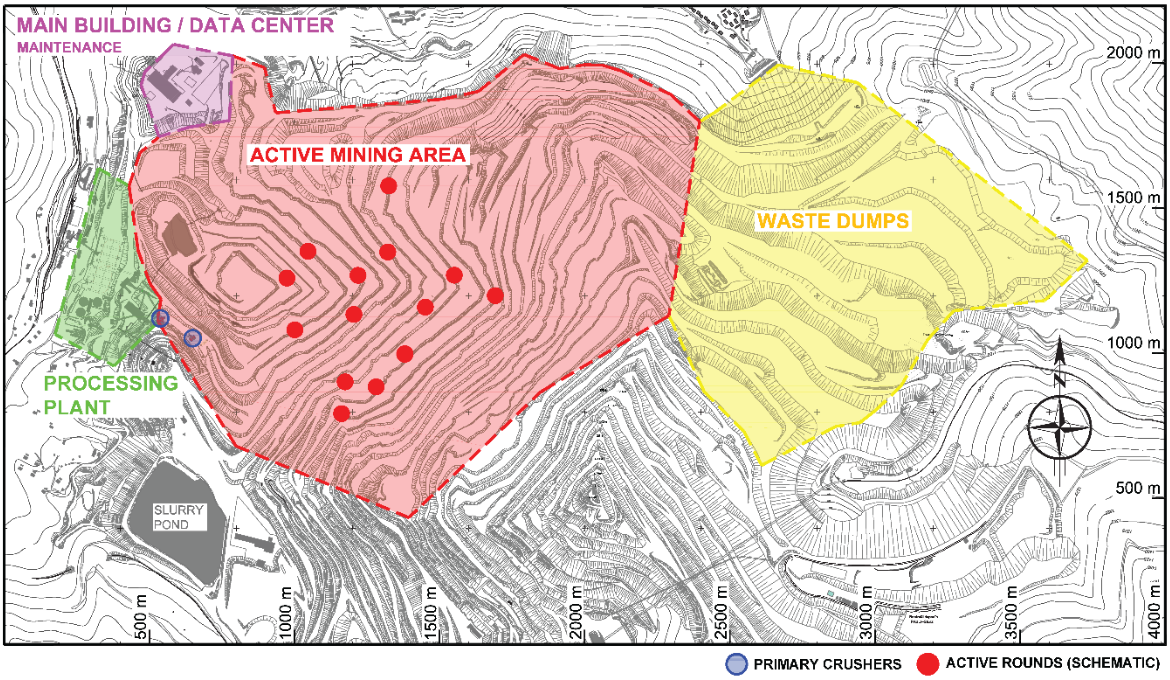


Figure 12: Operational areas with active rounds (schematic)

Depending on the observed process, different datasets in a different quality and quantity can be gained out of the machinery used in the mine.

The trucks are equipped with a vehicle telemetry system which automatically and continuously sends the current operating status to the datacentre. The wheel loaders in the mine are also equipped with such a vehicle telemetry system, but here the data is recorded only by the operator by activating a signal when the truck is fully loaded. In contrast to the trucks, this does not happen automatically.

Furthermore, the 2 primary crushers have data loggers which record the energy consumption over time.

These data sets will be described in more detail in the following subchapters, but first the general loading cycle will be described.

The present loading cycle which is observed in open pit operations can be generally subdivided in varying loader work and in constant loader work, as illustrated in Figure 13 Varying loader work occurs from dumping the last bucket onto the truck (D4) till dumping the 1st bucket onto to next empty truck (D1). Usually there is no truck waiting between. these actions, unless the wheel loader is busy with other tasks, such as refuelling or maintenance, which take longer than this intermediate interval.

In this thesis, the loading cycle of the wheel loader begins as soon as the shovel is attached to the muck pile and fills the first shovel for the arriving or already waiting empty truck. This activity belongs to the varying activities, since this first filling occurs in the interval where the operator decides individually which activity to perform. Before or after the first filling of the bucket, the operator has various possibilities, as they are illustrated in Figure 12:

As **alternative 1**, the operator directly manoeuvres to the muck pile and starts filling the bucket. Then the filled bucket is kept on the muck pile and the operator waits till the next truck comes to his visibility so that he can position himself for spotting the truck.

In **alternative 2**, the operator starts to prepare the muck pile for the next cycle in form of loosening the blasted rock, cleaning the floor, comminute or separate boulders aside. Afterwards the operator starts filling the bucket and waits, like in alternative 1, for the next truck.

In **alternative 3** the operator waits idle till the next empty truck comes to his visibility, from then he starts filling the bucket and positioning himself for spotting the truck.

The following filling and dumping operations up to the complete loading of the truck (D1 till D4) are carried out continuously and without interruptions. As a rule, a truck is fully loaded with 4 dumps. However, in some cases a fifth or even sixth loading cycle may be required to fully load the truck. A loading cycle ends as soon the truck is completely loaded, from then onwards it takes some time till the next truck is on site and ready to be filled. During this interval the operator has the possibility to perform several different actions, like cleaning the road, preparing the muck pile for further loading cycles, establishing safety measurements along the benches, or just waiting till the next truck arrives at the loading spot.

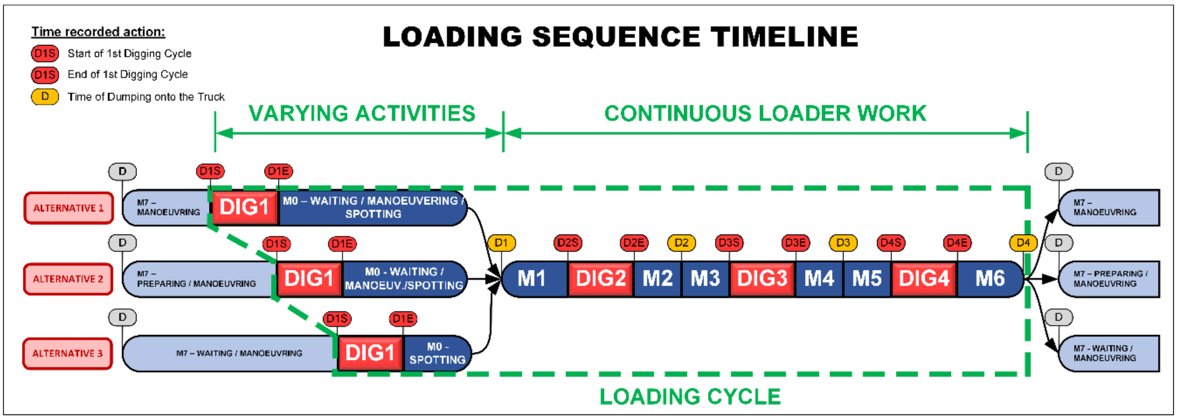


Figure 13: Loading sequence – 4 dumps onto truck

Every operator usually has his own working habit, in terms of acting during the idle times, which is described above under the term of “alternatives”. They even change their habits during their shift. Also, every quick-inspection by mechanics or refuelling is happening in the irregular interval between D4 and the next dump.

In order to create comparability between the individual operators, only the effective dig time for assessing diggability can be taken into account.

This means that the digging and manoeuvring time depends only on interaction operator / machine / muck pile, which is described in the following chapters.

3.1.1 On-site observations

Beside the introduction of a semi-automated KPI assessment, the observation of the loader activities, in conjunction with the blasted muck pile is one of the main objectives in this work. As already described in literature the interaction loader / muck pile can be determined in sufficient detail, provided that the loader is equipped with sensors which are linked to a data logger. By capturing the pressure of the boom-, or shovel-cylinders, together with the position of the control levers, the amount of acceleration and the coordinated position of the shovel, the loading cycle as illustrated in Figure 13, could be automatically determined and analysed in detail by using machine learning technique. These automatically assessed datasets could be further used to find parameters which classifies or characterises the behaviour of the wheel loader, thus the experience of the operator.

Operating Loaders at Erzberg mine do not have such sensors or data loggers, so the loading cycle must be recorded in a different way. In a first approach the loading cycle was recorded manually by filling a prepared form. Situated on the bench above the muck pile which was currently loaded, observing the ongoing loading process, the time for each individual loading step, according to Figure 13, was noted on the form. Additionally, some muck pile properties like, point of muck pile attack or spatial variability were noted.

Figure 14: Loading process observed from a bench above

During this observation technique the observer gets good insights in the whole loading chain and the loadability of the muck pile, even differences in operator proficiencies could be noticed. However, this method of observation is associated with some disadvantages. On the one hand, one is very strongly bound to a specific location, a frequent change of location of the wheel loader relates to the same, whereby the correct repositioning above the loading point is often complicated. Generally stated, the observation cycle can only be carried out at daytime and under suitable weather conditions, making continuous observation practically impossible. But highlighting the most important one: to guarantee good visibility on the loading process the observer must position himself quite close to the edge of the bench, where several cracks, induced by blasting, make the bench very unstable. This safety aspect makes this method void.

3.1.2 Video camera

In order to eliminate this safety aspect as mentioned in the previous chapter, a tripod with a camera attached to it was placed on the upper bench, which now records the loading cycle. This tripod can be positioned very close to the edge of the bench and when the muck pile is directly underneath, a good view of the loading process is possible.

When setting up the tripod, however, during evaluating the process on the computer it turned out that due to topographical possibilities it is often not possible to position the camera in such a way as to achieve a clear view of the digging process. Also, if the wheel loader changes its position, the tripod has to be repositioned again, which in turn requires a certain amount of work and time and also means that some loading cycles are not recorded. Furthermore, it turned out that it is often not possible to record the loading cycle at night due to the difficult night light conditions, especially when the camera is not directly near the mucking point.

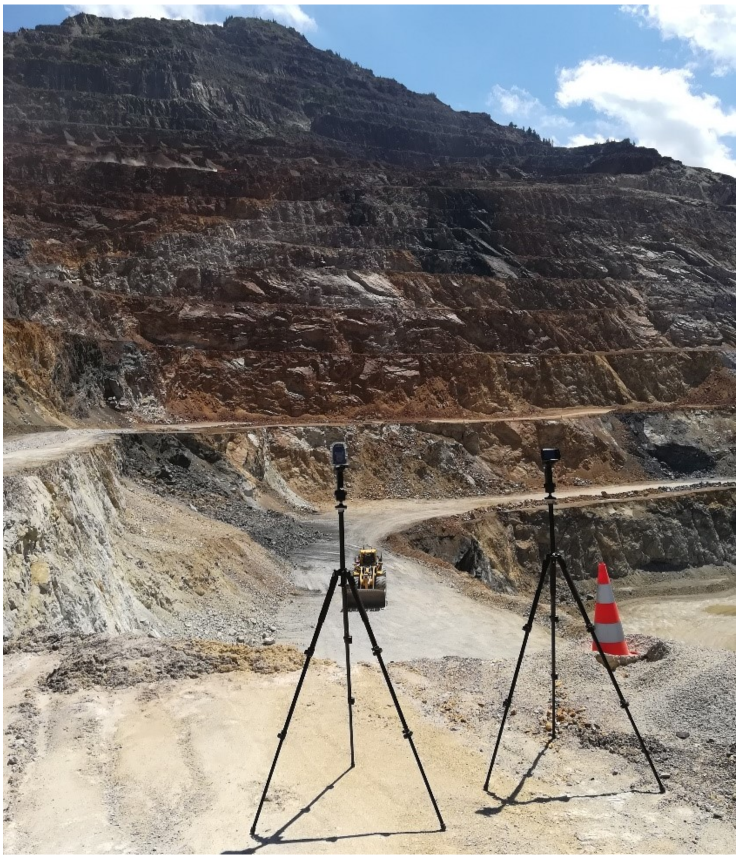


Figure 15: Loading process observed from a bench above

In addressing this problem of location dependency, the camera was mounted to the front of the driver's cab using a prefabricated bracket. In this position there is always a good visibility on the digging process, which allows the determination of the start of the digging phase very well.

With this camera position, the loading cycle at night can also be observed very well, as the wheel loader's headlights provide sufficient exposure for the camera. Even the number of the truck, which is placed on the side of the driver's cabin, is clearly visible. Due to the good accessibility to the camera, the good perspective on the muck pile and above all the continuous recording of the wheel loader's activity, this method was chosen for the future measurement campaign.

The camera, as it can be seen at Figure 16, is a commercially available time lapse camera by the brand name Brinno of the model TLC200PRO. With a shutter release interval of 1 seconds, a frame rate of 10 frames per second, the usage of a 32 GB memory storage SD together with lithium batteries, a continuous recording time of at least 3 full days could be achieved during the test run. For more detailed specifications and setting see Appendix B.

Figure 16: Loader with camera mounted on top of operator’s cabin

The start of the digging cycle begins when the bucket teeth touches the muck pile as shown in Figure 17 a). Although the exact point of contact of the shovel to the muck pile is not visible with this type of camera mounting, as shown in Figure 17 b), the start of muck pile penetration can still be determined very well when viewed on the computer.

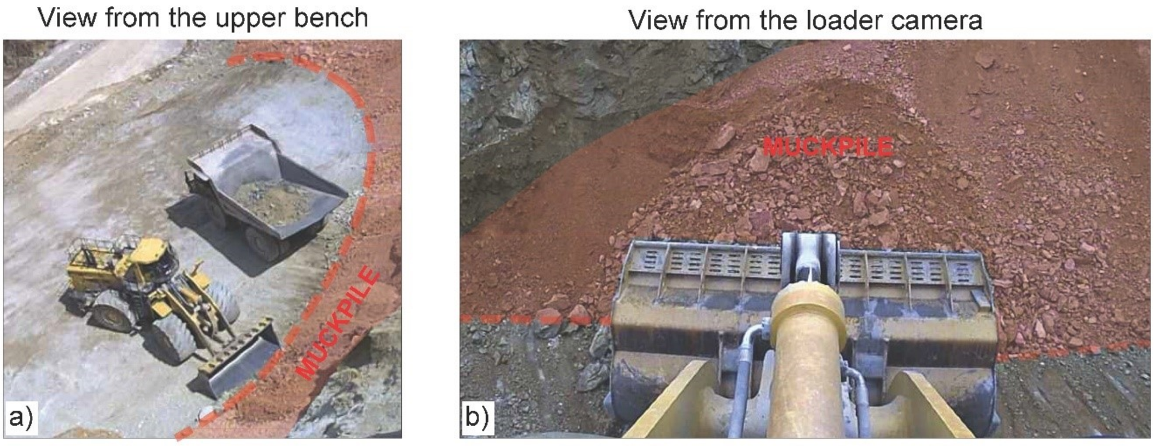


Figure 17: Start of digging phase

During the digging phase the operator moves the shovel, as shown in Figure 7 - Figure 10 through the muck pile. The end of the digging phase is characterized by tilting the bucket backwards sweepingly to get the highest possible bucket fill. This can be seen in the video evaluations by a slight tear-off edge on the muck pile, as shown in Figure 18 b). Furthermore, the end is also characterized by the fact that the wheel loader rotates at the swivel joint to start the first backward movement of the V-shapes, as also seen in the illustrations below.

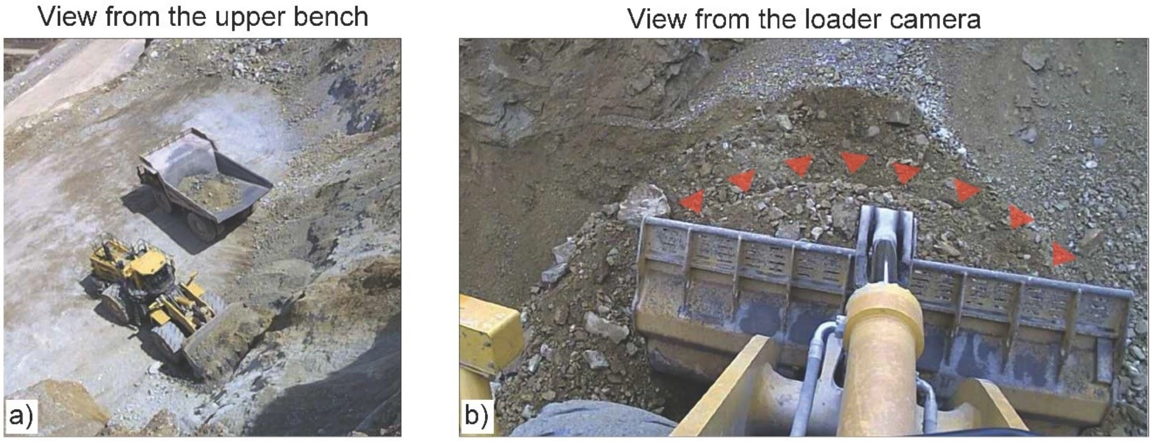


Figure 18: End of digging phase

3.1.3 Telemetry data - truck

In order to have an overview of the machines used in open pit mine as well as their transport routes, besides the already existing telemetry system VHMS (Vehicle Monitoring System) by the loader manufacturer Komatsu, the company implemented an additional customised system. This system uses passive monitoring to record all available information and transmits it in real time directly to the driver's cab (bunker filling levels, tire pressure, tire temperature, etc.) and to the control centre in the main building (position, loading, fuel gauge, status, etc.). This data is recorded around the clock at intervals of 2 seconds as TXT-file and stored in a database, from which the data can be extracted manually. Depending on the file information stored, the file size is approx. 37 - 117 MB.

Thus, one file per day and truck is usually obtained, regardless of location, material and operator code, which provides the following information from this file:

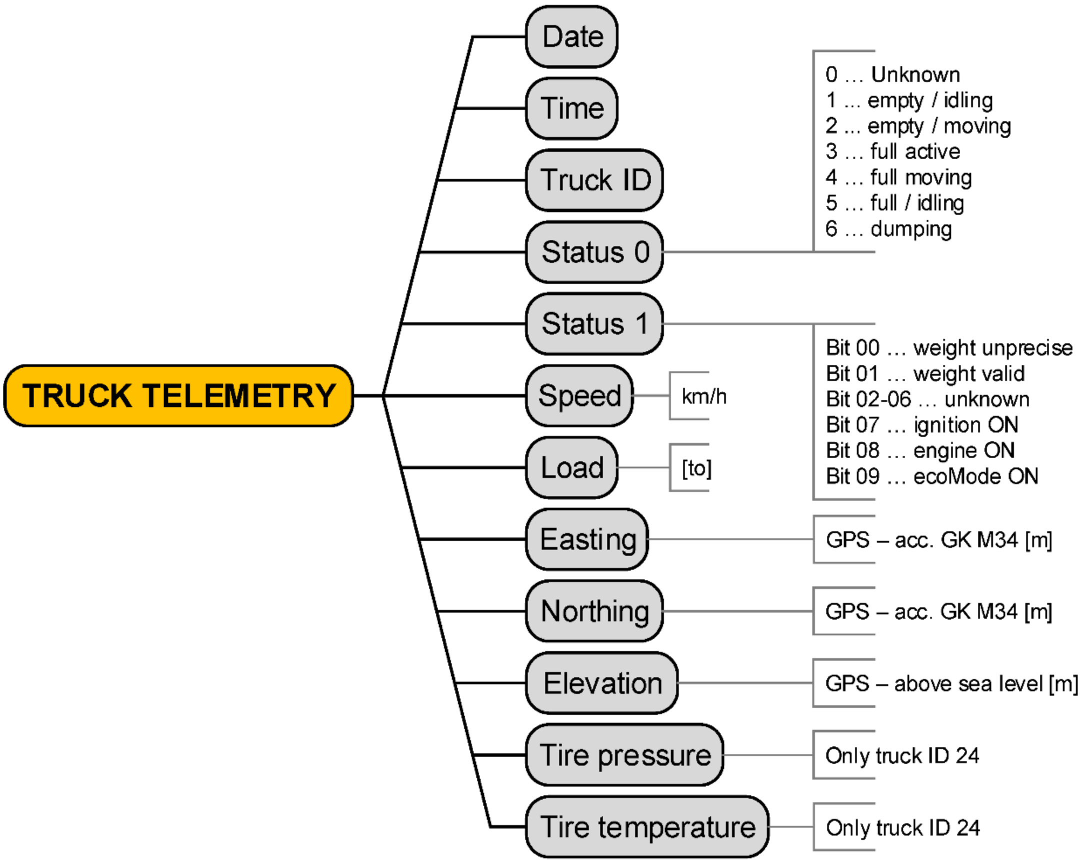


Figure 19: Present available data of each truck unit

3.1.4 Telemetry data - crusher

Just as the telemetry data of the trucks is recorded at regular intervals, the entire crushing cycle is also recorded at 15 second intervals. There are 2 primary crushers at the edge of the active mining area, which are loaded with ore, as already shown in Figure 12. Both crusher no. 2 and crusher no. 3 are fed, whereby crusher no. 2 is preferred. These two crushers have a base load of approx. 24 KW and peak loads of up to approx. 250 KW, as visualized in Figure 20. If a truck load is dumped into the crusher before the load curve reaches the base load again, so-called multi dumps occurs. A single crushing cycle can no longer be clearly defined here. For the single dumps, on the other hand, the curve falls back to the base load. The data records are manually extracted from the database for processing.

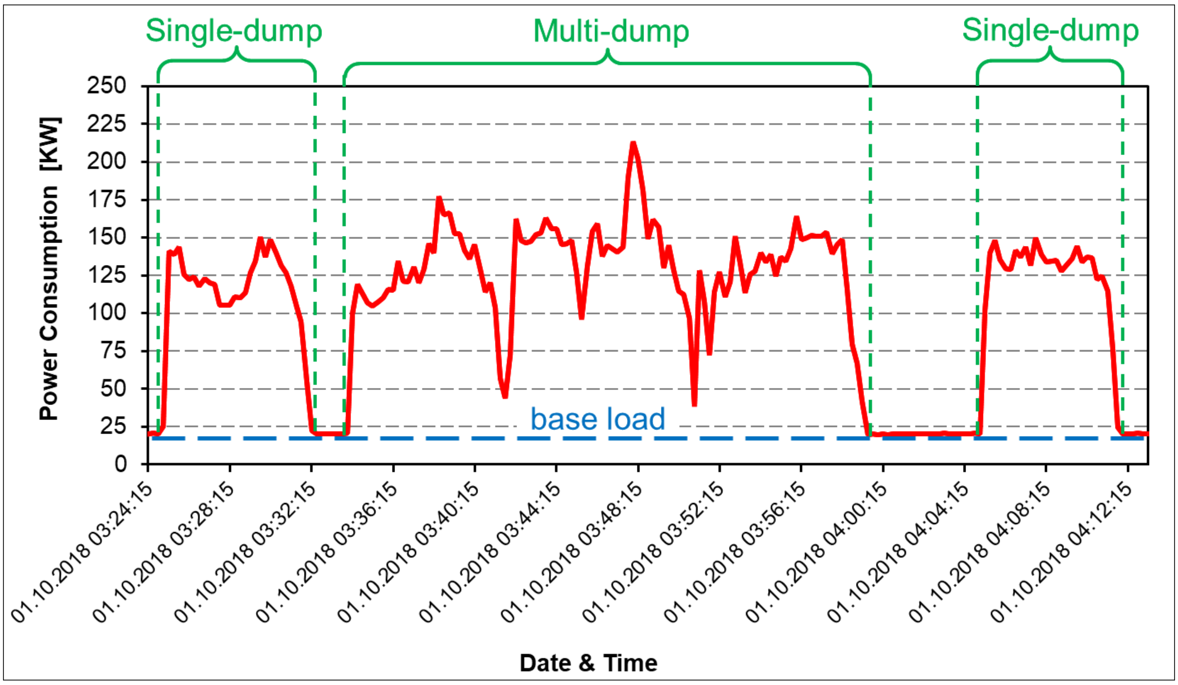


Figure 20: Original power curve of the crusher

3.1.5 Telemetry data - Loader

The wheel loaders are also equipped with a telemetry system. However, the structure and format of the data sets received, differ from those of the trucks. A key factor here is that the recording is not done continuously, rather manually by the operator. As soon as the truck has been fully loaded by the wheel loader, the operator activates a signal which documents the end of the loading process. This in turn means that all other activities apart from truck loading, such as levelling the floor, heading to another muck pile or establishing safety bunds, are not documented.

Another important difference to the telemetry data described above is the configuration of the data sets. Those of the trucks as well as the crusher are based on a daily (24h) data set structure, whereas the wheel loader telemetry is oriented on the shifts, as shown in Figure 21. Shift 1 starts at 06:00 and lasts until 14:00. Shift 2 starts at 14:00 and ends at 22:00. Shift 3 starts at 22:00 of the same day and ends at 06:00 of the next day. However, in the PROmis internal database system, this change in date of the third shift occurring between midnight and 6 a.m. is stored with the date of the previous day.

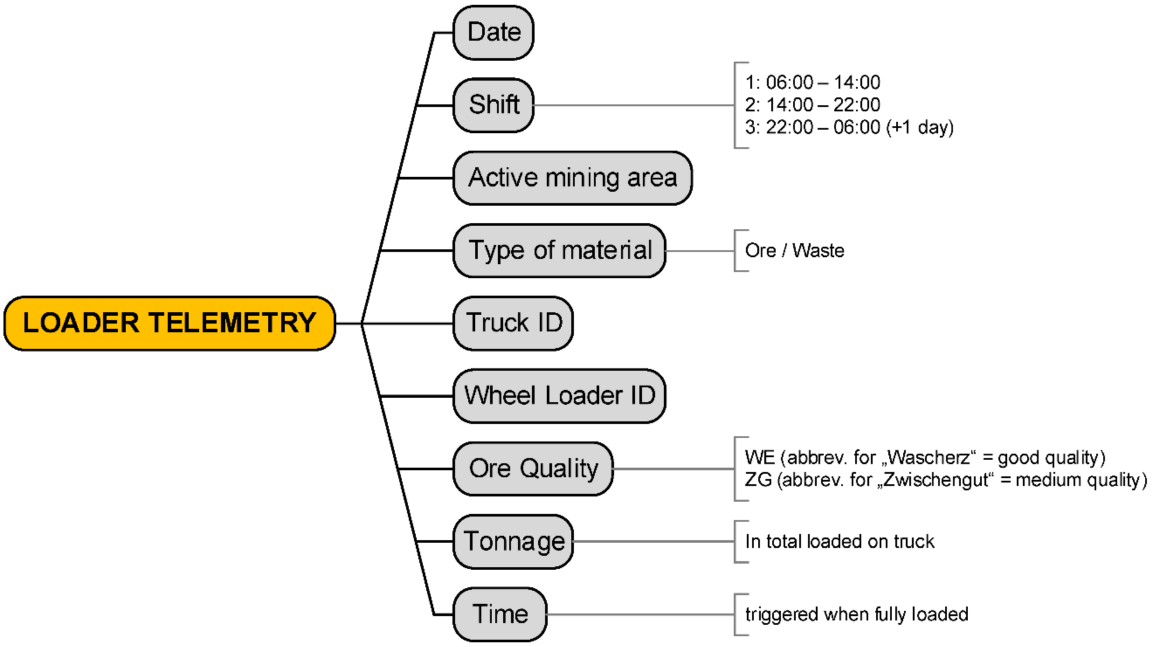


Figure 21: Present available data of each loader unit

3.2 Data processing

The datasets described in the previous chapter, however, do not come from a central database with the same characteristics, but from 3 different sources (loader camera, truck telemetry, crusher telemetry) with different quantity and quality. These data sets have not yet been compared together in an interactive manner, although each data set of every mining unit is already used to display live information in the control room. Therefore, a semi-automated system has been developed to associate these loosely available data sets with each other in order to present them in a central database and thus link them with each other in order to obtain the desired KPIs.

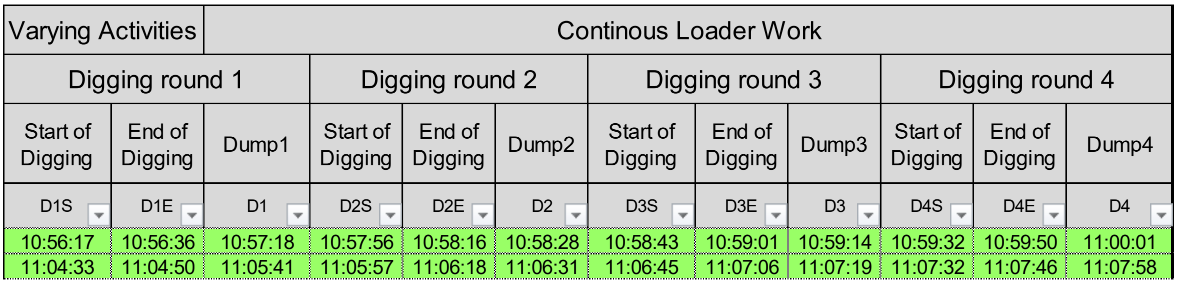
The Matlab software was chosen in order to obtain fast processing of the unstructured and various data sets. Out of this program, the combined data is then listed in a table and can be read into any program or database.

As already mentioned before, the extracted data set of the wheel loader is not taken into account in the following data evaluation because the data content is incompatible with other data sets with regard to structure and data information.

3.2.1 Data from video camera

Commencing with the video data, a tabular form was set up in MS EXCEL in such a way that the loading cycle times, taken from the integrated time stamp in the video, can be manually filled in, as shown in Table 2. The structure of the table is done according to the procedure on site. A more detailed description of the data recording from the video can be found in the Appendix B.

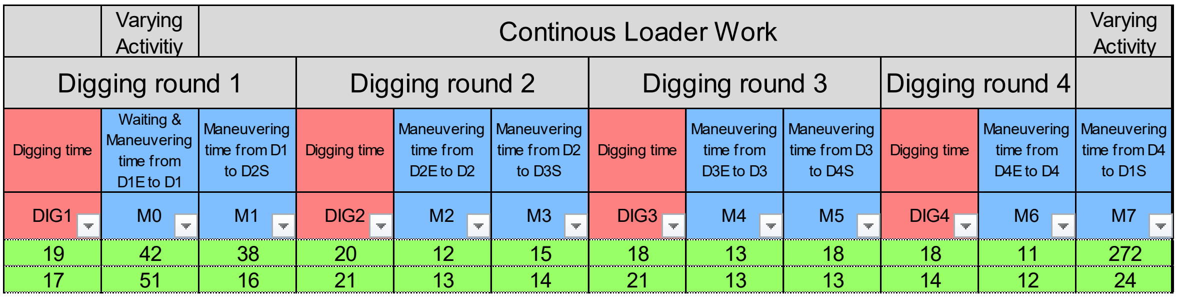
One row in the spreadsheet corresponds to a digging cycle, which consists of 4 digging rounds (DIG1, DIG2, DIG3, DIG4). As already mentioned at the beginning, the truck is usually filled with 4 buckets, but under certain circumstances it may take more than 4 loading operations to fully load a truck. But those cycles with more than 4 loads are not used and excluded with regard to a consistent processing scheme.



Varying Activities		Continous Loader Work									
Digging round 1			Digging round 2			Digging round 3			Digging round 4		
Start of Digging	End of Digging	Dump1	Start of Digging	End of Digging	Dump2	Start of Digging	End of Digging	Dump3	Start of Digging	End of Digging	Dump4
D1S	D1E	D1	D2S	D2E	D2	D3S	D3E	D3	D4S	D4E	D4
10:56:17	10:56:36	10:57:18	10:57:56	10:58:16	10:58:28	10:58:43	10:59:01	10:59:14	10:59:32	10:59:50	11:00:01
11:04:33	11:04:50	11:05:41	11:05:57	11:06:18	11:06:31	11:06:45	11:07:06	11:07:19	11:07:32	11:07:46	11:07:58

Table 2: Timestamps according to time lapse camera

Additionally, to each row, a date, a truck number, a wheel loader ID and an operator ID is assigned in order to be able to trace the loading cycle. At the end of each cycle, it is possible to record notes on the manoeuvring conditions, such as the spatial conditions, point of attack or any preparation works. Out of these manually entered data, net times of each loader activity in the loading cycle can now be determined and used for further analysis. As shown above, the start (D1S) and end (D1E) of the first digging phase result in the net digging time (DIG1), wherein the end of the digging phase until the first dump on the truck represents the first manoeuvring cycle.



	Varying Activitiy	Continous Loader Work									Varying Activity
Digging round 1			Digging round 2			Digging round 3			Digging round 4		
Digging time	Waiting & Maneuvering time from D1E to D1	Maneuvering time from D1 to D2S	Digging time	Maneuvering time from D2E to D2	Maneuvering time from D2 to D3S	Digging time	Maneuvering time from D3E to D3	Maneuvering time from D3 to D4S	Digging time	Maneuvering time from D4E to D4	Maneuvering time from D4 to D1S
DIG1	M0	M1	DIG2	M2	M3	DIG3	M4	M5	DIG4	M6	M7
19	42	38	20	12	15	18	13	18	18	11	272
17	51	16	21	13	14	21	13	13	14	12	24

Table 3: Net digging rounds

As soon as the first digging process ends (Figure 18) the first manoeuvring path (M0) begins until the first dumping onto the truck (D1), which is characterized by the first visual recognition of the bucket tilting. The graphic below shows very clearly that the first manoeuvring distance M0 in particular is very different from the other manoeuvring rounds in the considered loading cycle.

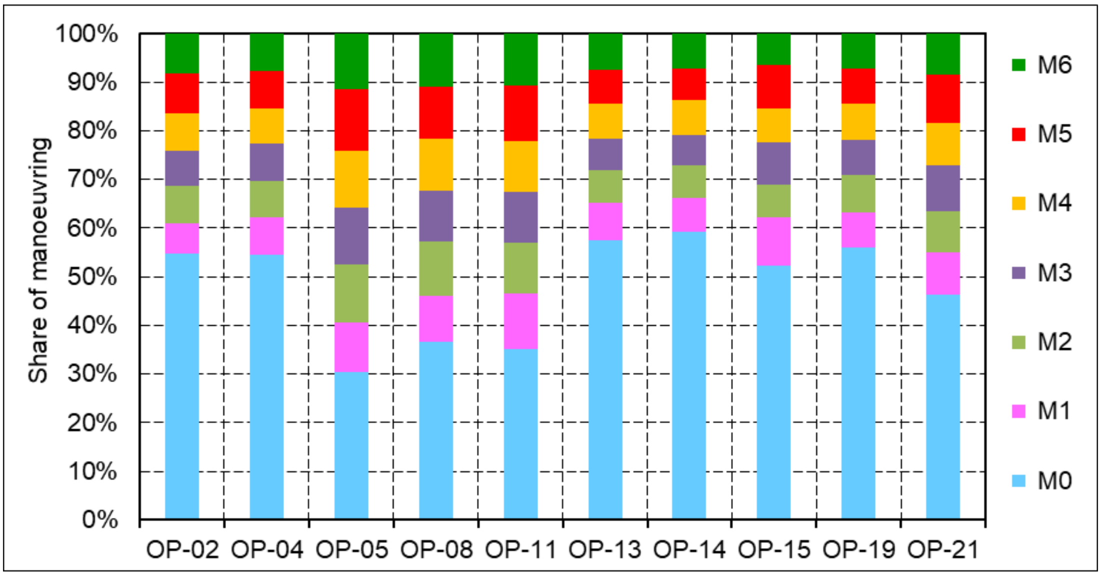


Figure 22: Share of total manoeuvring times per operator

The relatively long duration of the M0 cycle is due to the fact that in the logged cases no truck has yet waited for the load and the wheel loader then has to wait as well. As shown in Figure 13 as alternative 1, here the operator starts filling the first shovel directly after unloading the preceding truck. Afterwards the operator has no possibility to make other preparations because the bucket is already filled.

In contrast to alternative 1, the operator of the 3rd alternative waits with the filling of the first bucket until the next truck appears near the wheel loader. This is done according to the operators as soon as the first truck is seen or after an individual assessment of the sense of time. In this round, the operator has a longer time span (M7) to perform preparatory activities. Depending on the number of trucks assigned to a blast and the length of the hauling route, this cycle can be longer.

It must be mentioned, that long idle times without any preparation work, like it is in the case of just waiting for the next truck, are considered as not ideal. But in many cases preparation after every loading cycle is simply not necessary, because either the material trickles down by itself or the floor is already cleaned up.

Short intervals as shown in the 2nd line can be traced back to waiting trucks, which was rather rare in the observed period.

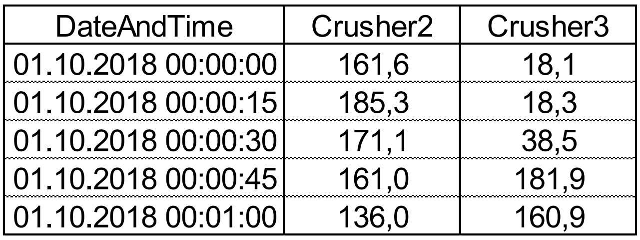
Since the phases M0 and M7, i.e. the phases up to the spotting of the truck, last considerably longer than the other manoeuvring phases, it can be assumed that a match factor of less than 1 is obtained and thus a higher utilisation of the wheel loader operator could be achieved as described in chapter 2.4. In addition, Alternative 2 is a combination of the alternatives described above.

3.2.2 Crusher telemetry

The dataset from the crusher telemetry is made available by the company as MS Excel file format. But these files are not presented in chronological order and must be formatted additionally. Hence, the data records are previously processed in an Excel VBA code in order to be flexible with regard to changes in the nomenclature and file structure and thus avoid any error code messages in the automatized code. Furthermore, these prepared raw data sets (in an MS Excel file format) are then available in a concise form and can be used for further processing. Detailed instructions on how to do this can be found in the appendix.

On the part of the company it is possible to export the crusher raw data in daily steps, which means 5760 rows of data information per day per file, taking into account a recording rate every 15 seconds. This is of particular importance because when importing data into the programming interface as described in the chapter below, the increase of the data set by days or weeks increases the required computing power and thus the computing time. This issue was solved in Matlab by the daily import of the datasets. Furthermore, the daily import of the datasets results in a clear and structured output, as described in the following chapter.

During the pre-measurement campaign, data sets of 4 days or 23040 lines were extracted, leading to reasonable computation times. The data set which is generated from the VBA code and then imported into Matlab is structured as follows, whereby the second and third columns describe the energy consumption, in kilowatts (KW), of the crushers 2 and 3 respectively recorded in 15 seconds time intervals:



DateAndTime	Crusher2	Crusher3
01.10.2018 00:00:00	161,6	18,1
01.10.2018 00:00:15	185,3	18,3
01.10.2018 00:00:30	171,1	38,5
01.10.2018 00:00:45	161,0	181,9
01.10.2018 00:01:00	136,0	160,9

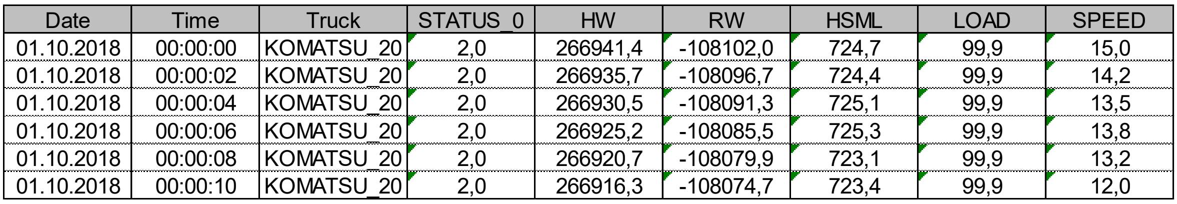
Figure 23: Excerpt of the first 5 rows from a prepared crusher raw file

In order to link this formatted data record with the trucks, these telemetry data must first be prepared for further processing.

3.2.3 Truck telemetry

As described in the data collection, the data provided by the company in the TXT- format have information every 2 seconds. This information, however, is not listed chronologically and in tabular form, but starts every hour with a new variable. So, the recording of the first variable starts at "00:00:00" and lasts until "00:59:58", whereby the next variable starts at "00:00:00" and is recorded until "00:59:58". As soon as all variables have been run through, the same procedure starts from the beginning, but now starting at "01:00:00" and ending at "01:59:58". This structure continues until the end of the day "23:59:58". As a result, the information contained in the original TXT file consists out of 302400 lines. This unsorted type of file structure and its size makes it inevitable to prepare the raw-data semiautomatically.

There have now been developed 2 types of data processing. As first variant the respective TXT-file is prepared by means of a VBA code (“TRUCK_ImportTXT_ExportXLSX.xlsm”) in the spreadsheet program MS EXCEL and stored in tabular form in a new EXCEL file. Figure 24 shows an excerpt from such a prepared file, where the file begins at the time "00:00:00" and ends at 23:59:58", resulting in 43200 lines of file information per day:



Date	Time	Truck	STATUS_0	HW	RW	HSML	LOAD	SPEED
01.10.2018	00:00:00	KOMATSU_20	2,0	266941,4	-108102,0	724,7	99,9	15,0
01.10.2018	00:00:02	KOMATSU_20	2,0	266935,7	-108096,7	724,4	99,9	14,2
01.10.2018	00:00:04	KOMATSU_20	2,0	266930,5	-108091,3	725,1	99,9	13,5
01.10.2018	00:00:06	KOMATSU_20	2,0	266925,2	-108085,5	725,3	99,9	13,8
01.10.2018	00:00:08	KOMATSU_20	2,0	266920,7	-108079,9	723,1	99,9	13,2
01.10.2018	00:00:10	KOMATSU_20	2,0	266916,3	-108074,7	723,4	99,9	12,0

Figure 24: Excerpt from a prepared truck telemetry file

The advantage of this procedure is a fast and clear presentation of the contained datasets in a user-friendly data format. This preprocessed file is now imported into the Matlab programming interface and then linked to the other data sets.

In order to avoid this intermediate step of the file processing with VBA code, a second possibility was developed, which reads, sorts and formats the existing TXT raw file directly in Matlab. This procedure, which is described in the next chapter, provides the same table structure as shown in Figure 24.

3.3 Combined data sets in Matlab

Matlab as an interactive data processing and evaluation program is suitable due to its common and intuitive user interface to collect and quickly execute large amounts of data and calculation processes where other spreadsheet software such as MS Excel comes up against its limits.

Following the subtitle of this work, the subsequent section is described as a guideline for the further procedure of data evaluation of the upcoming measuring campaign.

To obtain a concise data evaluation, a folder structure was created which is explained briefly below. A detailed explanation of the files contained therein will be given later.

- **Folder: „Code“**
  This folder contains all required scripts, which are executed from the programming interface in Matlab.
- **Folder: „Data“**
  Here, the prepared crusher file as well as the reformatted "TXT" files of the trucks are stored in the corresponding subfolders.
- **Folder: „SavedCrusherData“**
  Here the program automatically saves an auxiliary file with the ending ".mat" which contains the necessary information of the crusher data.
- **Folder: „SavedTruckData“**
  In this folder the program automatically saves auxiliary files with the extension ".mat" which contain all necessary information of the loaded TXT- files of the trucks.

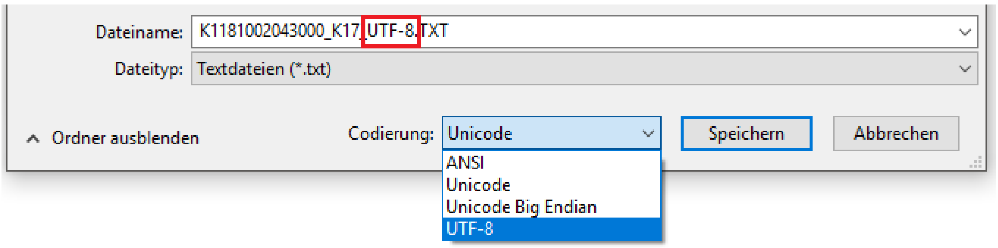
Prior to executing the codes from the "Code" folder, the raw data received from the company, namely the crushers and trucks, must be prepared for import into Matlab.

Starting by the raw data of the trucks, these are provided by the company with the following file name:

K1	**181001**	**043000**	**_K17**	**.TXT**
	Date of recording	Time of saving	ID	
	yy.mm.dd	hh:mm:ss		

Table 4: Filename explanation of truck telemetry

These files are delivered in the standard unicode encoding, but in order to be able to process them, they must be encoded in "UTF-8" as shown in Figure 25. These files must be renamed and saved in the "Data" directory in its respective subdirectories.



Dateiname: K1181002043000_K17_UTF-8.TXT

Dateityp: Textdateien (*.txt)

Codierung: Unicode

Figure 25: UTF-8 Encoding of truck raw data

The raw crusher data received from the company must be prepared in accordance with the instructions in Appendix C: ImportCrusherData.xlsm: and also stored in the "Data" directory.

In order to make data processing as clear and structured as possible, the code was divided in three scripts.

The first script "EnergyCalculationCrusher.m", subdivided in the editor as "Section A", deals with the calculation of the crusher energy. The second script "MatchCrusherTruckTXT.m", subdivided as “Section B”, processes the truck telemetry including the subsequent assignment of the trucks to the energy curves. The third script matches all single data information which is automatically stored in the folder “SavedTruckData” into an entire output table.

For a detailed description of the individual code sections, see Appendix C, Section “2) Matlab Script”, where the entire code including the corresponding comments is listed.

When starting the first script it has to be considered that first the whole folder is loaded into the interface to avoid an error message, as illustrated in Figure 26.

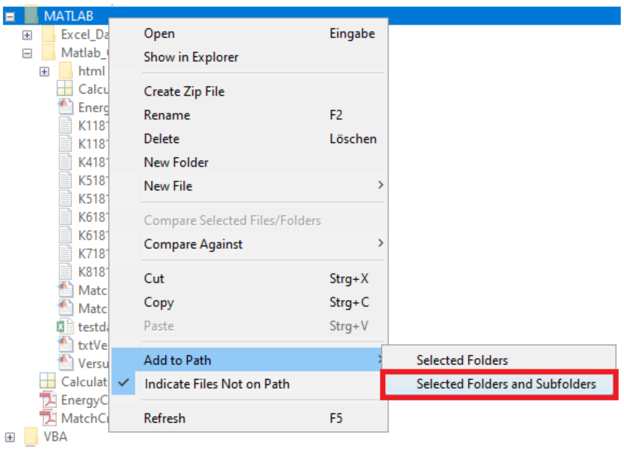


Figure 26: Add folders and subfolders

3.3.1 Script: EnergyCalculationCrusher

Once all raw data has been saved in the appropriate directories and subdirectories and loaded into the programming interface, the execution of this script can be launched. To accomplish this, open the "Code" directory and open the "EnergyCalculationCrusher.m" file in the Matlab directory tree by double-clicking.

The data structure of the trucks and thus also the programming structure is based on a daily basis. However, the crusher data can be extracted by the company over several days. In order to compare a day of operation of the trucks with the same day of operation of the two crushers, the respective range (= day) to be imported in Section A.1, marked with "edit_range" in

```
.„T = readtable(filePath, 'Range', ' edit_range');‟
```

must be updated accordingly, before the code is executed within the "EnergyCalculationCrusher" script.

Furthermore, before executing the code in section A.7, the path marked with "edit_path" in

```
savePath = 'C:\edit path\MATLAB\SavedCrusherData\'
```

must be adjusted accordingly.

The day ("edit_range") in Section A.1 must be updated as soon as all TXT-data has been loaded using the second script and the next day shall be analysed. The path in Section A.7 must only be updated if the location of the entire code package changes, e.g. by transferring it to another workstation. The script is then executed by pressing the "Run" button in the "Editor" tab.

This opens a window in which the prepared crusher file is selected from the "Data" directory and executed with the confirmation of the code.

When the script has been completed, 2 figures are displayed which depicts the respective energy curves of the crushers 2 and 3, as shown for one day in Figure 27 and Figure 28.

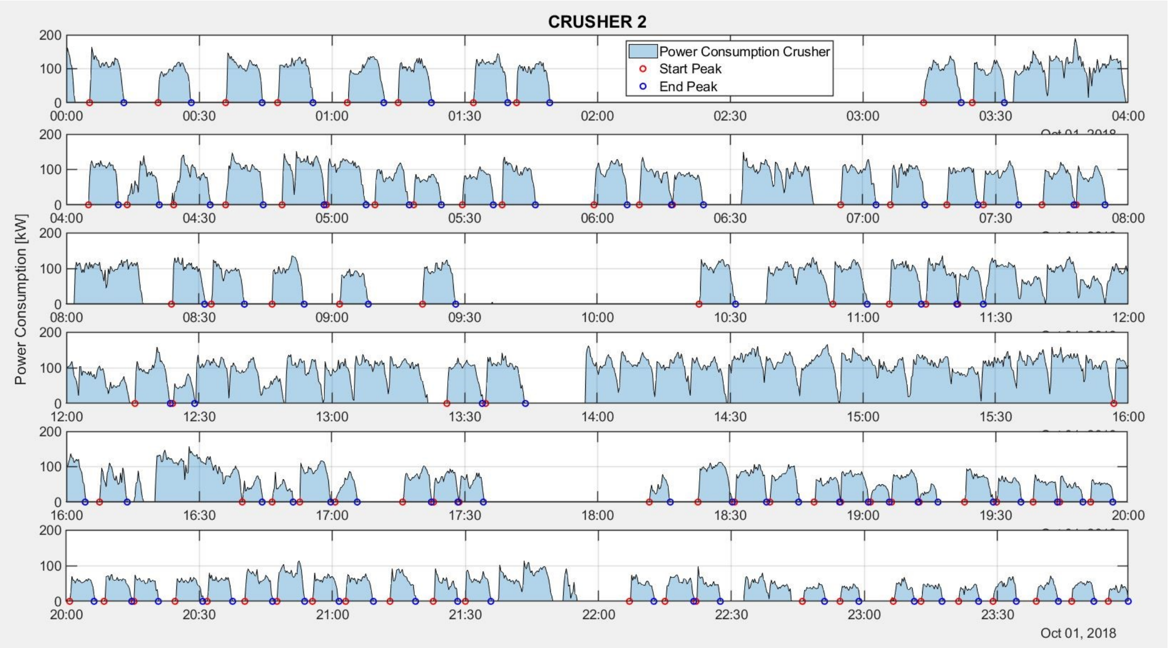


Figure 27: Energycurve with marked single dumps for crusher 2

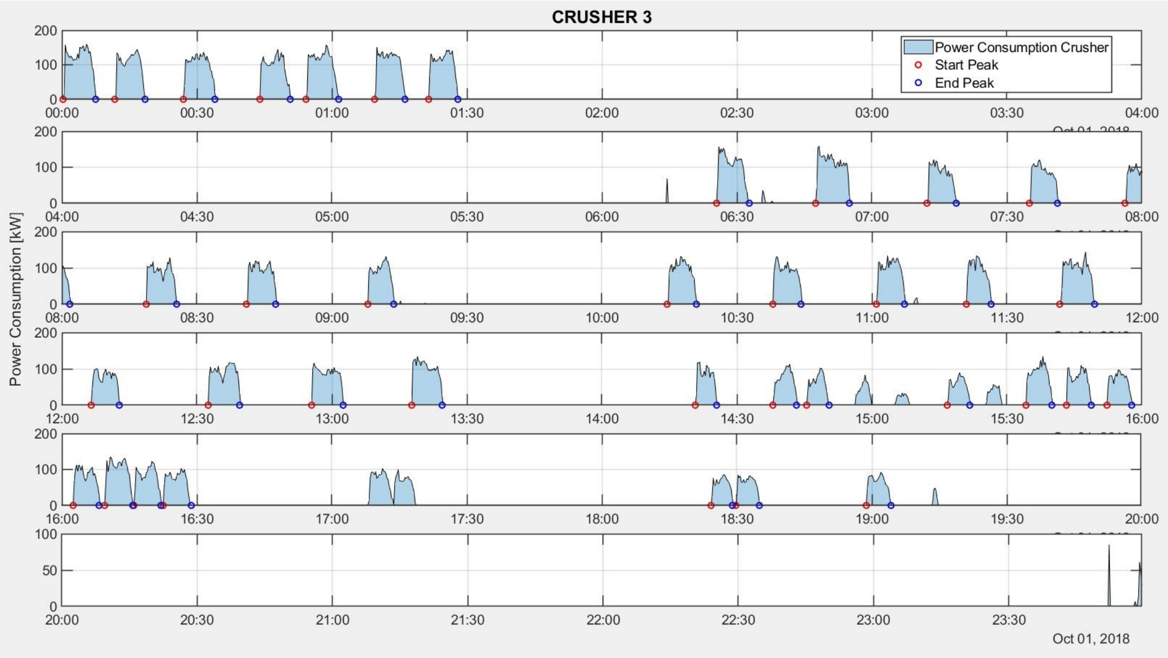


Figure 28: Energycurve with marked single dumps for crusher 3

Only those crushing cycles are considered which correspond to a single dump because they are clearly defined and can therefore be analysed precisely. Multi dumps are not considered in this thesis. The base load of the crushing unit is subtracted from the power of crushers 2 and 3, which was assumed to be 24 kilowatts in this work and can be adjusted if necessary. If this results in a negative numerical value, then this value is set to 0.

In a further step, the energy consumption of the identified single dumps is calculated, which relates to the area under the respective load curve. This means that for each comminution cycle concerned, the respective energy is determined and assigned to a time stamp in tabular form. These single dumps are marked in Figure 27 for crusher 2 and in Figure 28 for crusher 3 with "Start Peak" and "End Peak".

All determined data, which are required for further processing, are stored in a ".mat" file in the folder "SavedCrusherData", which contains the day as file name. This file is then accessed in the second script.

3.3.2 Script: MatchCrusherDataInputTXT.m

This script is opened like the previous one. Before executing this, the path in Section B.11 ("edit_path") must be changed once and the day in question, must be adjusted accordingly, as marked with "DDMMYYYY" in

```
load('C:\edit_path\MATLAB\SavedCrusherData\DDMMYYYYCrusherData.mat').
```

Once this has been done the RUN button can be pressed.

A file input window appears in which the UTF-8 encoded TXT- file of the truck must be selected. By confirming the code will be run through. When the execution is finished, a table appears in the command window in which all information of the crushers is matched with the data from the truck file.

In the following, the individual sections of the code are briefly described. For a more detailed explanation, reference is made to the comments in the code in 1.1

Because the raw data of the truck telemetry is not available in a usable form, and thus to continue chapter 3.2.3, they are processed in the second script from Section B.1 to Section B.7.

In section B.8 the newly prepared file is purged and filtered with regard to the required information. Only those status information are retained which correspond to loading (status_0 = 3) and dumping (status_0 = 6). Since the status information "loading active" in the raw file is time-delayed and therefore not exact, those lines are removed where the speed is not equal to 0 to ensure that only loading events on the muck pile are detected.

In section B.8, a loading event is assigned to each dumping cycle. Since the truck data only record one day, it is possible that the logged data file starts with status 6 and the corresponding loading event takes place the day before and was therefore saved in another file. Since this LoadDump cycle is thus incomplete, it is not taken into account.

Section B.9 identifies all possible loading and unloading event. It should be noted that the number determined here does not necessarily have to correspond to the number of crushing cycles at the crusher as the truck can also dump elsewhere.

Section B.10 examines, where the truck is dumping its load. This is done by checking, whether the truck is in the close perimeter of either the "crusher 2" or "crusher 3". With the aid of the coordinates of the both the truck and the crushers, the Euclidean distance is calculated by:

$$Euklidean\ distance = \sqrt{(RWtruck - RWcrusher)^2 + (HWtruck - HWcrusher)^2}$$

<u>Where:</u>

RWtruck … RechtsWert truck ≙ Easting coordinate of truck

RWcrusher … RechtsWert crusher ≙ Easting coordinate of crusher

HWtruck … HochWert truck ≙ Northing coordinate of truck

HWcrusher … HochWert crusher ≙ Northing coordinate of crusher

To determine the exact dumping place, it is checked whether the calculated Euclidian distance is less than 50m as illustrated in Figure 29. Depending on which criterion is reached, the truck dumps at crusher 2 or crusher 3.

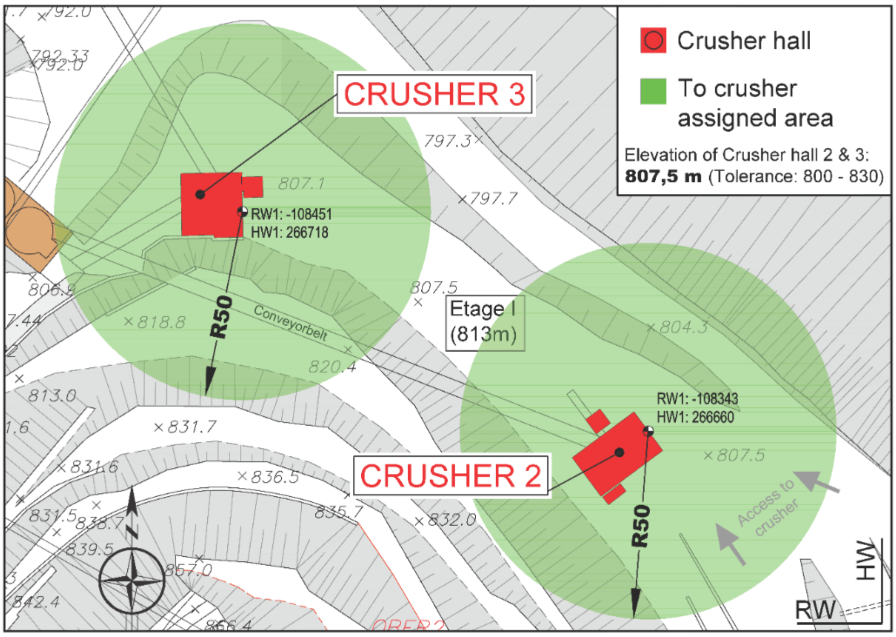


Figure 29: Coordinative area allocation of crushers 2 and 3

Section B.11 loads the ".mat" file with the crusher data out of the directory "SavedCrusherData" to match this data with the truck data. Here the path and the respective date are to be adapted as already described above.

Section B.12 now merges all data.

Section B.13 combines all merged data into one table and saves the table in the directory "SavedTruckData". For each truck one file is created with an individual file name in the format "DDMMYYYY_KOMATSU_XX.mat".

As soon as all TXT files of the trucks for a particular day have been processed and different individual truck tables have been created, they can be consolidated by the third script.

3.3.3 Script: MergeTable

Prior to executing this script, the path in Section C.1 and in Section C.2 must also be adjusted once. Again, this only needs to be done once and only if the code package changes the directory.

Afterwards the script can be executed by using the "Run" button This will merge all files saved in the directory "SavedCrusherData" into one table, termed "CumulativeOutputTable". This table consists of the following information:

Header	**Content**
Date	Date of recorded action
LoadTime	Time when the truck was loaded at the muck pile
DumpTime	Time at which the truck dumps at one of the crushers
HWload	Northing coordinate of loading place at the muck pile
RWload	Easting coordinate of loading place at the muck pile
HSMload	Elevation (Bench) of loading place at the muck pile
Truck	Truck ID
DumpPlace	„Crusher 2“ or „Crusher 3“
Tonnage	Tonnage dumped into one of the crushers
Energyconsumption	KWh per crushing cycle
KWhperTonnage	KWh per tonnes
TotalCrushingTime	In seconds
TotalCrushingTimePerTon	In seconds

Table 5: Cumulative Output table

Since this data, together with the other collected data from the measurement campaign, will ultimately be stored in a database, this table can now be exported in section C.2 to the spreadsheet program MS excel or retained in the Matlab data structure.

4 Results

4.1 Semi-automated data analysis

Using the Matlab scripts developed in this way, 2 data sets, which were originally stored in separate databases, can be linked to each other and correlations can be found. Out of the single TXT-, or EXCEL files it is now possible to combine the specific crusher energy, the position where the truck has been loaded and the required crushing time per tonnage load, per truck in one table.

The assignment of the truck loads to the load cycles is possible either graphically, as shown in Figure 30 for a specific date and a specific Truck ID, or in tabular form, as obtained by the table “CumulativeOutputTable” in Appendix C.

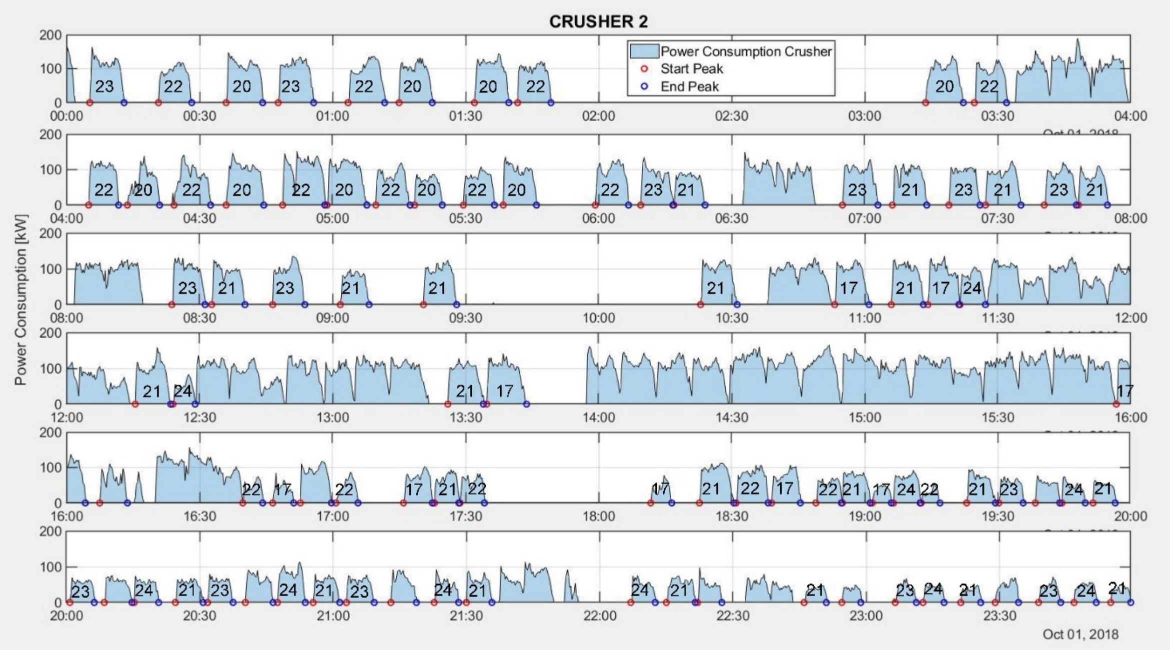


Figure 30: Load cycles with the assigned trucks

When analysing the crushing energy, it can be seen that the specific energy is between 0.02 kWh / tonnes and 0.16 kWh / tonnes. Since there are no grain distribution curves of the muck pile so far, these energy values cannot yet be linked to the composition of the muck pile, in particular the grain size or the uniformity. Nevertheless, during the dumping the material of the trucks into the cone crusher, it was observed that fine material passes relatively quickly through the crushing cone or through the closed side setting. Hence, as already known in the scientific community, the first insight that can be made is, the finer the material from the muck pile, the lower the specific energy consumption.

Although this is a semi-automated procedure, it is still advisable to view the raw data records, since inconsistent data, such as weight, coordinates or status variables, can occur repeatedly in the data.

The data processing and thus the code refers to a data structure obtained from the first pre-measurement campaign. If this data structure changes for any reason, this can lead to problems and error messages in the programming interface.

4.2 Influence of the operator on the digging process

During the data recording on site and through the video recordings, it became apparent that the observed operators scoop the blasted rock differently in time and type and unload it onto the truck also differently. This led to the consideration of elaborating differences between the individual operators and prepare these differences for further processes.

The theoretical assumption that every operator uses the same amount of work and know how in the digging process leads to a homogenization of the data basis and would come closer to a statistical significance, but this would not correspond to reality, as also described in (Danielsson 2016).

Every operator has his personal preferences, his personal work motivation and conviction to be able to actively intervene in the loading process. The personal experience and the cognitive abilities of an operator are extremely difficult to grasp in figures and cannot be judged with pure statistical time evaluations.

But in order to obtain an initial assessment and a quantitative representation, the digging times per operator and per digging round were examined. It turned out that some operators have a greater spread of digging times than others with smaller ranges.

Looking at all 4 rounds in Figure 32 - Figure 35, it can be seen that especially operators 13, 15 and 19 show a larger dispersion of the quantiles.

Figure 31 shows the sum of the observed digging cycles in order to get a statistical significance from the analyses. Only operator 40 has only 40 digging rounds, which corresponds to ten filling operations.

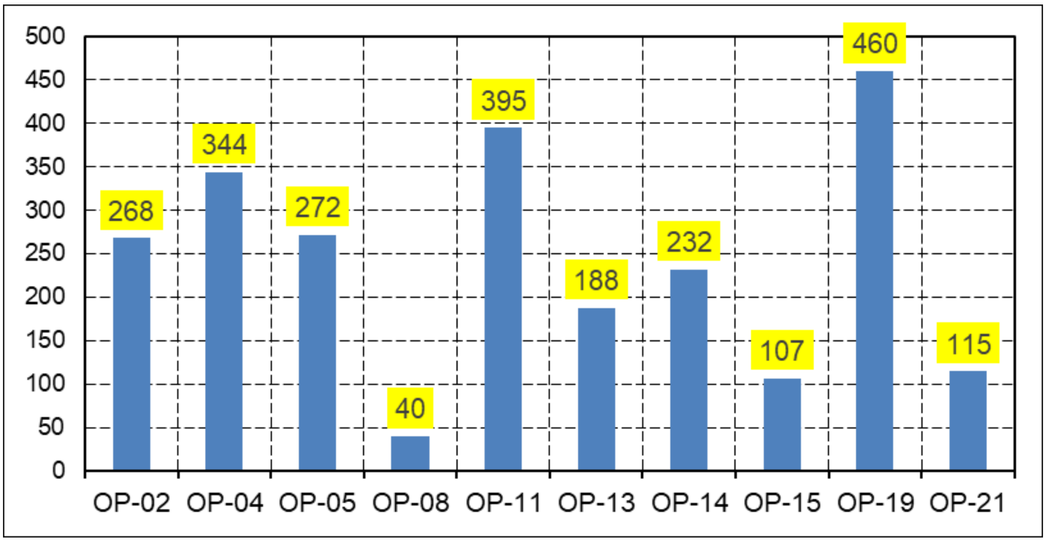


Figure 31 : Amount of digging rounds per operator

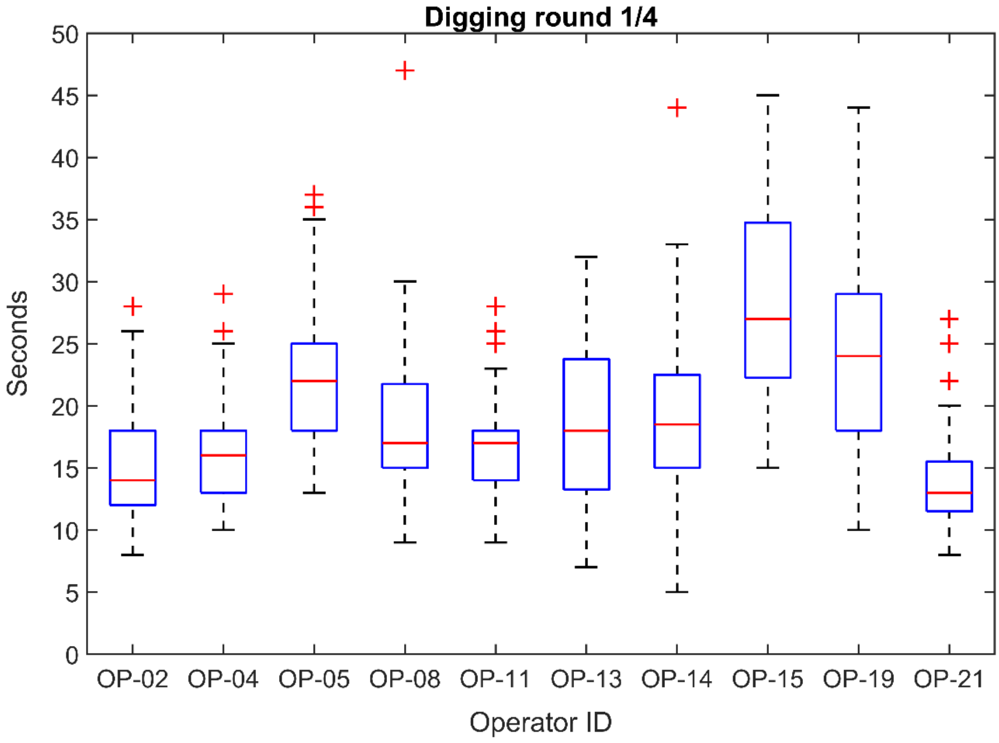


Figure 32: Digging times per operator – DIG1

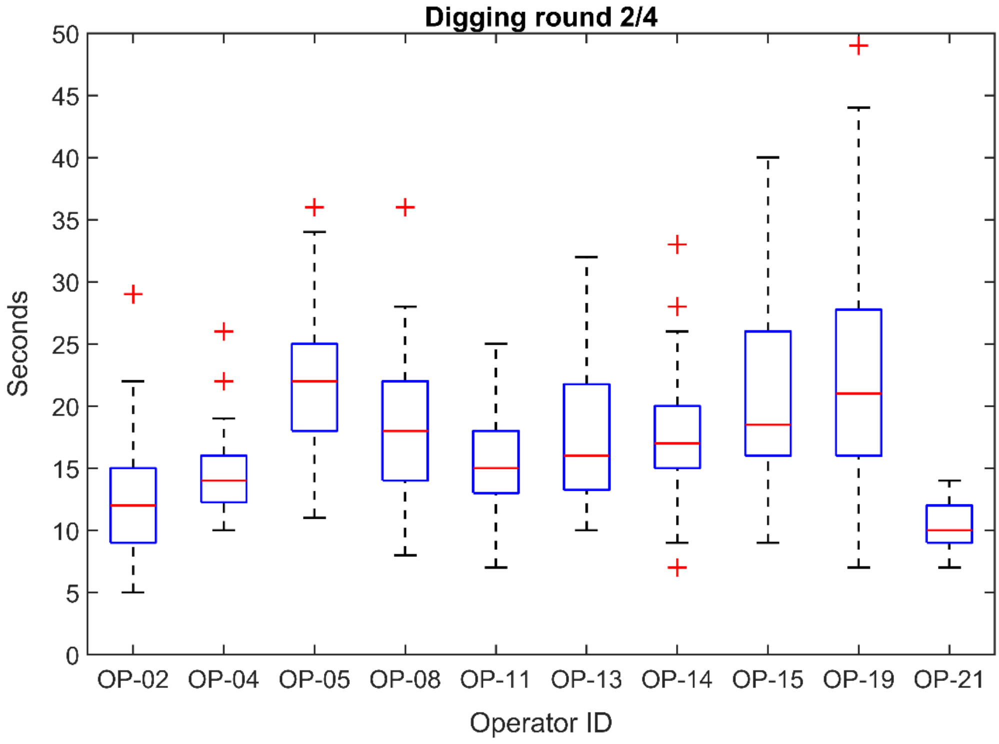


Figure 33: Digging times per operator – DIG2

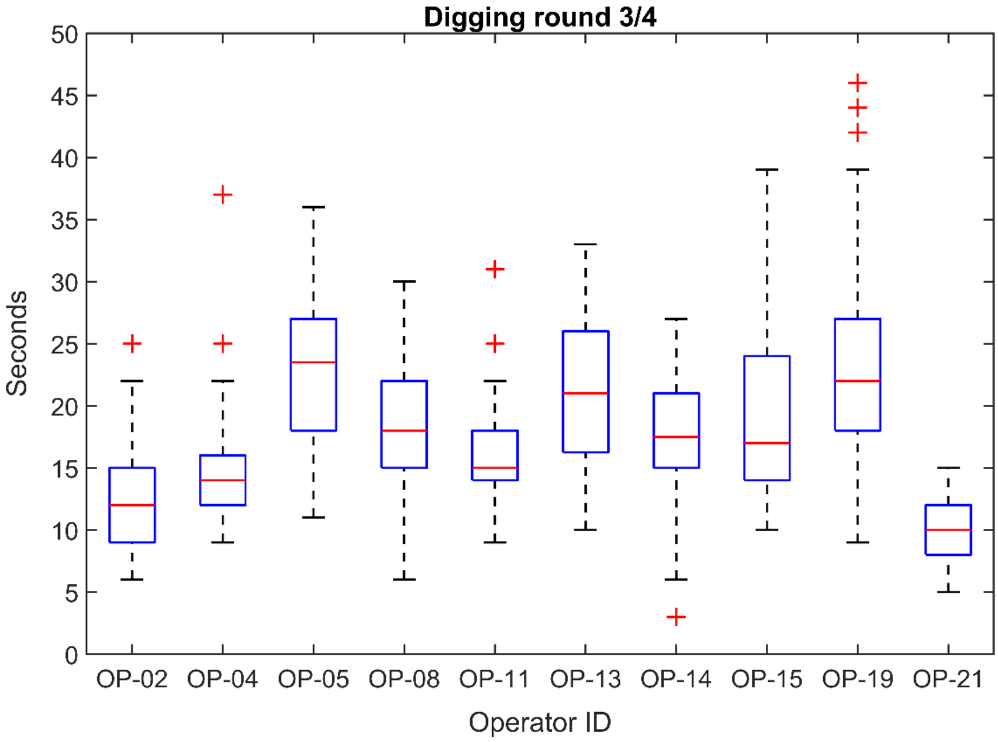


Figure 34: Digging times per operator – DIG3

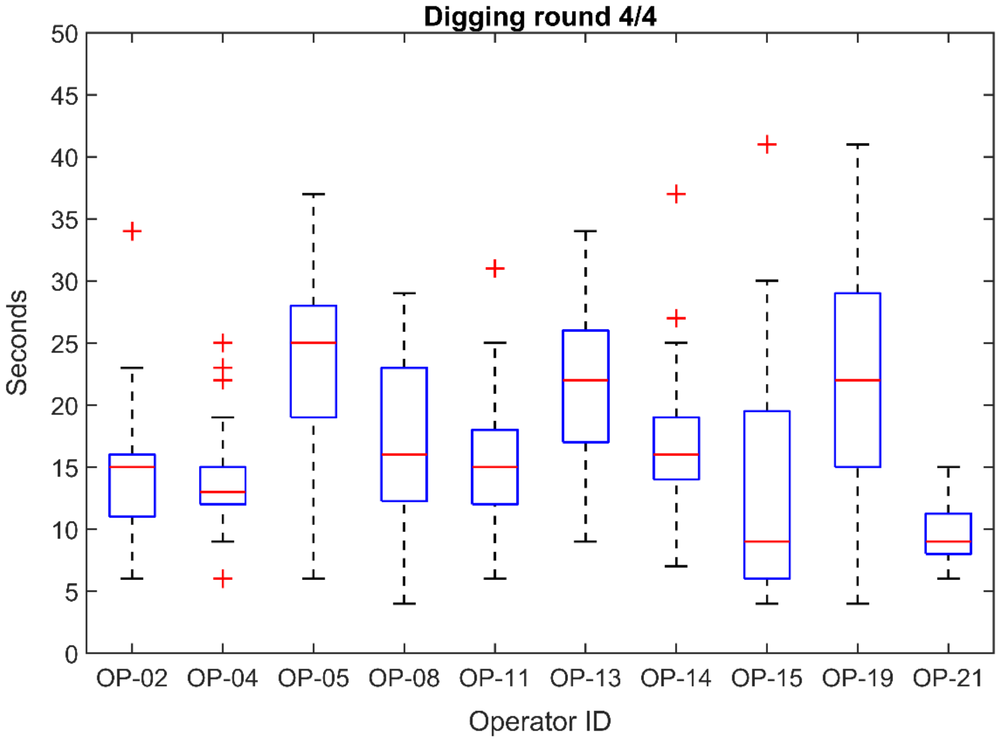


Figure 35: Digging times per operator – DIG4

Though these differences are discernible, the individual reasons for them are difficult to work out, as Filla tried to do in 2011.

The reasons for these deviations were assumed to be the spatial availability and the character of the muck pile. These factors are, however, superordinated by the operational capabilities and experience of the respective operators. The muck pile composition, like the grain size analysis, was not yet recorded at the time of this work. Thus, only the influence of spatial availability, which is described in chapter 4.3.1 could be investigated.

4.3 Correlation between digging and overall cycle times

One of the tasks of the telemetry data evaluation was to determine whether the digging times on the muck pile can be deduced from the telemetry data of the trucks, which only records the dumping times and the loaded tonnage of the wheel loader. Although digging and manoeuvring processes directly follow each other, they are completely different in manner. Nevertheless, these successive operating processes should be investigated for correlation in order to substantiate the assumption of independence from each other.

If the muck pile is very favourable in its condition regarding the bucket penetration, this would result in a very short digging time. However, if the truck is badly positioned, which is caused by spotting by the wheel loader operator unfavourably, the manoeuvring time can increase significantly. This can also happen if the spatial position for the muck pile attack requires long travelling distances. But if spotting is better in the next loading cycle, resulting in shorter manoeuvring times, two different manoeuvring times have been obtained for the same muck pile with the same attack point, leading to neither or a poor correlation.

In order to obtain an initial estimate of these dependencies, two analyses of the recorded digging and manoeuvring times were carried out.

First, it was examined whether the digging times obtained (DIG) correlate with the overall cycle time (M_{BEFORE}+DIG+$M_{AFTER)}$) of a digging procedure. The results of these correlations are shown in Table 6 and visualized in Figure 36.

In this context it can be seen that except for operator 15, none of the loading times listed show significant and constant correlations with the total cycle times. It should also be noted that the respective digging time is included in the total cycle time. If there is a correlation with the individual digging time contained therein, a value is compared with itself to a certain extent, which in turn increases the overall correlation.

R^2 R	DIG2 & M1+DIG2+M2	DIG3 & M3+DIG3+M4	DIG4 & M5+DIG4+M6	Amount of Datasets
Operator 02	0,48	0,54	0,26	67
	0,69	0,73	0,51	
Operator 04	0,27	0,70	0,44	87
	0,52	0,84	0,66	
Operator 05	0,53	0,56	0,49	68
	0,73	0,75	0,70	
Operator 08	0,03	0,02	0,00	10
	0,18	0,15	0,06	
Operator 11	0,18	0,21	0,37	99
	0,42	0,46	0,60	
Operator 13	0,05	0,64	0,50	47
	0,22	0,80	0,71	
Operator 14	0,12	0,20	0,28	58
	0,35	0,45	0,53	
Operator 15	0,73	0,74	0,72	27
	0,85	0,86	0,85	
Operator 19	0,49	0,54	0,64	115
	0,70	0,73	0,80	
Operator 21	0,59	0,47	0,16	29
	0,77	0,69	0,40	

Table 6: Correlation of digging times with overall cycling times per operator

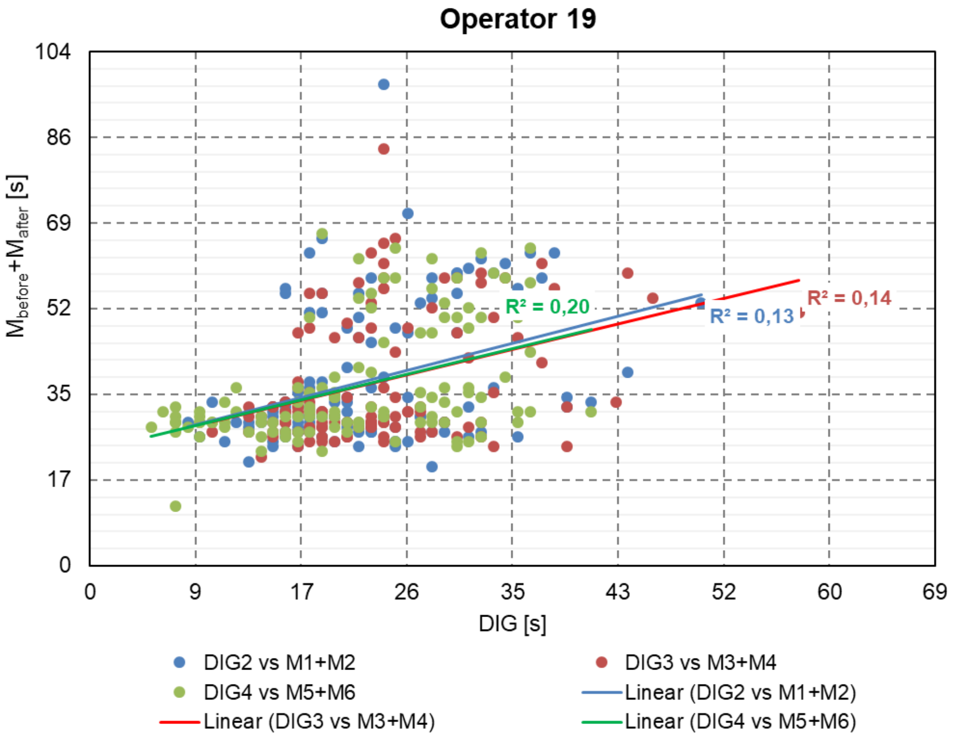


Figure 36: Correlation of digging times with overall cycling times per operator

To complete and confirm these results, the net digging times (DIG) from the video data were correlated with the preceding and subsequent manoeuvring cycle (M_{BEFORE}+M_{AFTER})). The results of these correlations are shown Table 7 and visualized in Figure 37.

R^2 R	DIG2 & M1+M2	DIG3 & M3+M4	DIG4 & M5+M6	Amount of Datasets
Operator 02	0,17 0,41	0,20 0,44	0,07 0,26	67
Operator 04	0,01 -0,12	0,05 0,22	0,01 0,11	87
Operator 05	0,00 0,07	0,02 0,13	0,01 0,10	68
Operator 08	0,59 -0,77	0,10 -0,32	0,19 -0,44	10
Operator 11	0,01 0,11	0,00 0,00	0,07 0,26	98
Operator 13	0,00 0,02	0,00 0,02	0,03 -0,18	47
Operator 14	0,00 0,00	0,00 0,05	0,03 0,16	58
Operator 15	0,05 0,23	0,03 -0,17	0,41 0,64	27
Operator 19	0,13 0,35	0,14 0,37	0,20 0,45	115
Operator 21	0,08 0,28	0,13 0,35	0,05 -0,22	29

Table 7: Correlation of digging times with manoeuvring times per operator

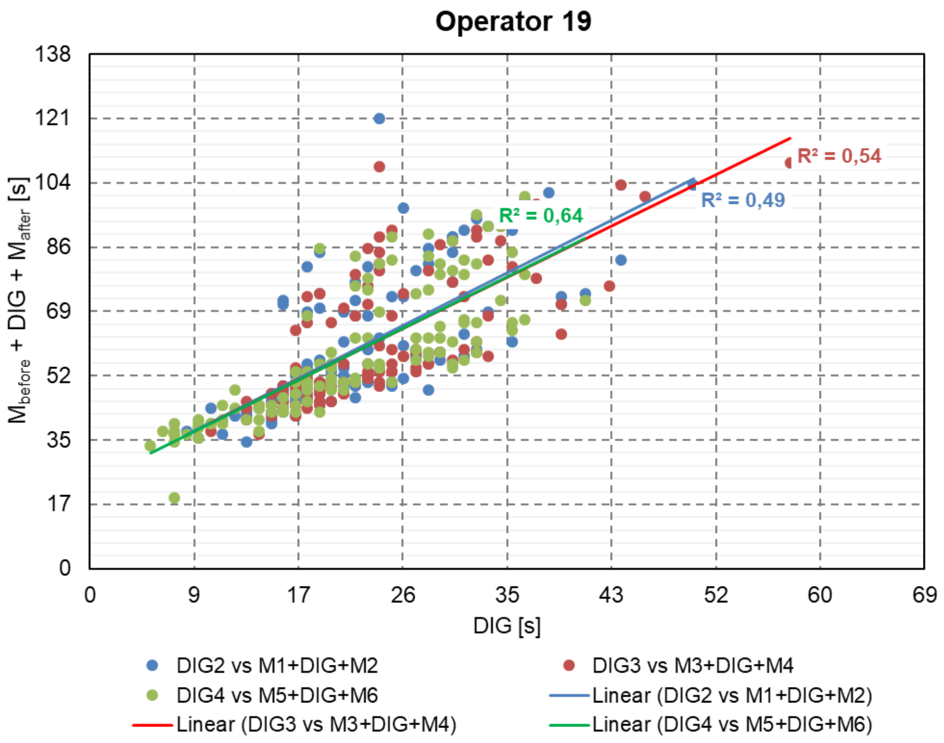


Figure 37: Correlation of digging times with manoeuvring times per operator

In this connection, it is clearly evident that none of the digging rounds shows a significant correlation with any of the observed operators.

Thus, using the telemetry data of the trucks, or specifically the detected dumping times, it is not possible to deduce the digging times. At this point it should be noted that even if the digging and manoeuvring processes would correlate, this would not necessarily indicate a direct relationship. Correlation alone is not proof of a causal relationship. In other words, the correlation coefficient can only enhance the result of a content-related question, but not prove it. (Fernuniversität Hagen)

4.3.1 Influence of the spatial variability

In the course of observing the loading process at the muck pile, the perception occurred that the spatial conditions had an influence on the loadability.

Therefore, the muck pile was divided into 3 areas, as shown in Figure 38. The classification and assignment of the point of muck pile attack is subjective, whereby a source of error is introduced here.

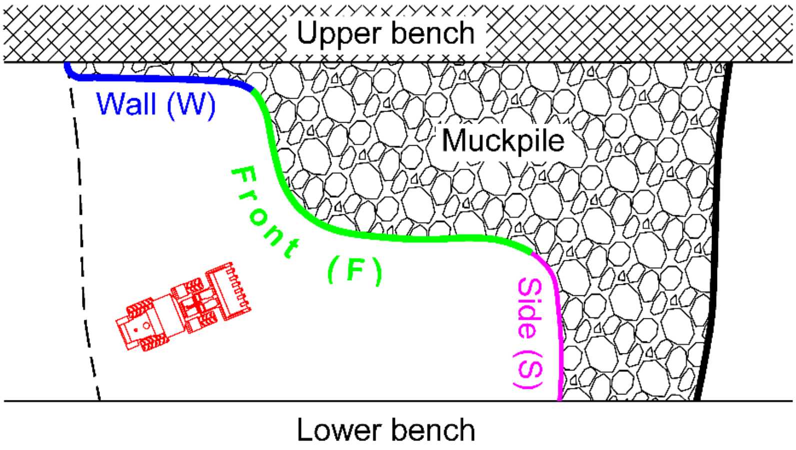


Figure 38: Geometrical muck pile situation

However, during the video evaluation it turned out that the position of the muck pile attack and the scooping duration are constantly changing. The loading times on the wall are slightly shorter, but as can be seen in Figure 39, the data here is too inconsistent to evaluate all loading times equally. Where it is possible, as with Operator 4, 5, 13, 14 and 19 it can only be stated that the frontal and sideways loading times alternate depending on the operator.

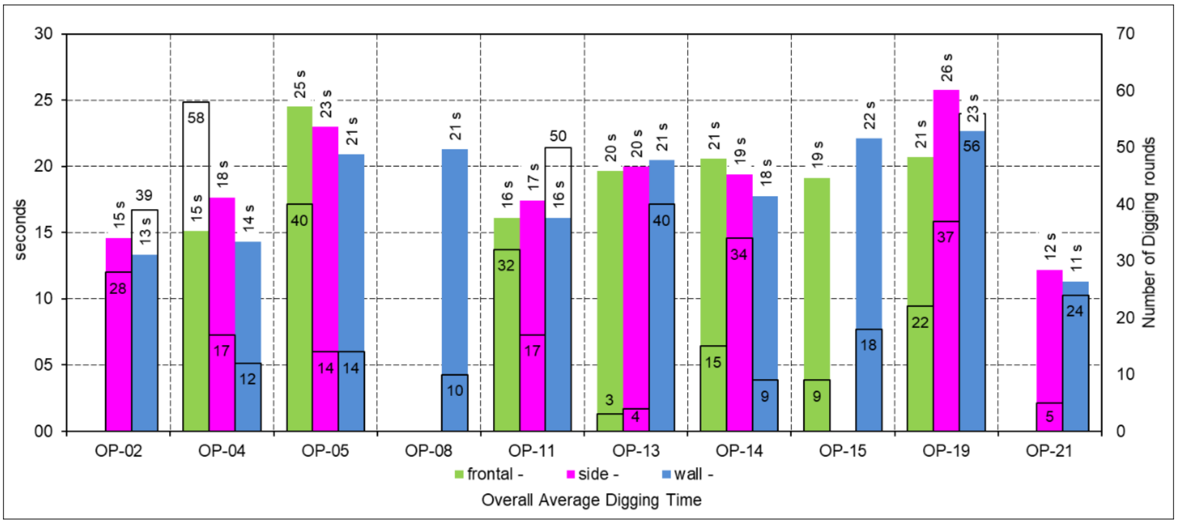


Figure 39: Overall average digging time per operator and point of muck pile attack

4.3.2 KPI assessment

In order to assess the emission reduction as effectively as possible by changing factors in the upstream process chain, as illustrated by Figure 40, it is now necessary to develop key performance indicators. As already emphasized in chapter 2.3, it is particularly important to ensure that useful and above all meaningful KPIs are developed. The future measurement campaign, which will last several weeks, will record a wealth of data that will serve as the basis for the KPIs.

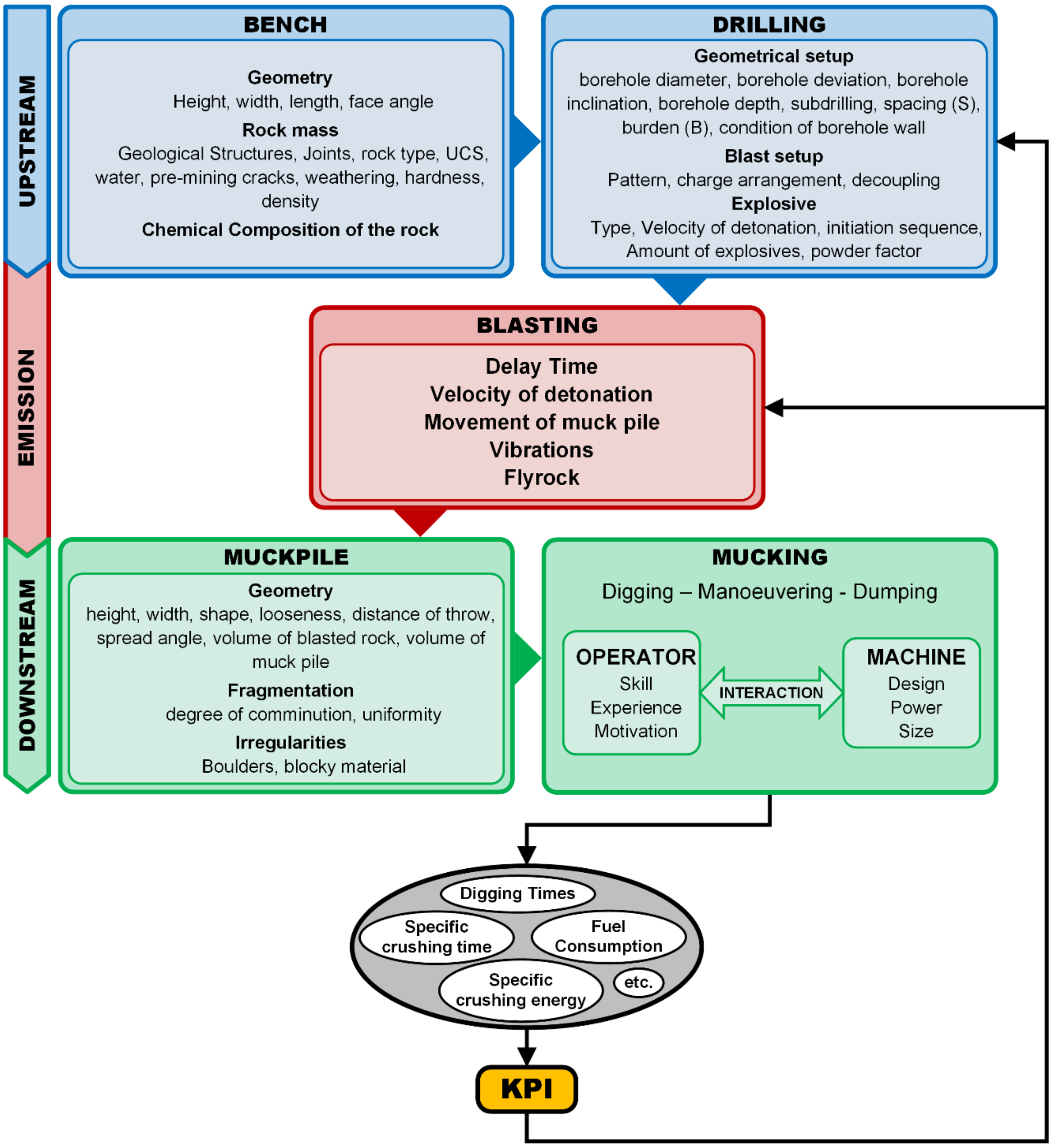


Figure 40: Measured parameters for KPI development

The data sets received can be divided into modifiable and unmodifiable factors. Unmodifiable factors include the bench geometry, such as height, width, length, face angle, and the numerous rock mass properties. Modifiable parameters can also be classified as directly modifiable or indirectly modifiable parameters, whereby directly modifiable parameters are primarily referred to as drilling and blasting setup. The muck pile properties result from the drilling & blasting setup and can therefore be classified as indirectly modifiable parameters.

Thereby it has to be considered that only those parameters are connected with each other which have a causal association. To keep the Figure 40 as it is, the combination of the bench geometry with the crusher energy can lead to correlations, but it is also important here to question the significance and the resulting value of this correlation. The loading activity at the muck pile represents here an exceptional position in the data acquisition and evaluability, because here the human factor cannot be classified, as is the case with the use of explosives, for example, where exact properties and behaviour is known. Thus, all measured data which are bound to the operative abilities of the operator, as it is the case for the loading activity at the muck pile, can only be used to a limited extent. Here the combination of digging work with the muck pile composition, such as the grain size distribution, would be rather advantageous, although, as described in the literature above, the grain size distribution alone should not be the main attribute of the muck pile, since other factors, such as the loosening or the presence of boulders, must also be taken into account. For direct qualitative statements on the loadability of the muck piles, it would be preferable to compare the machine data of the wheel loader, such as fuel consumption, or the cylinder pressures with the digging times. But at the moment these machine data are not available.

Another possibility to estimate the influence of the wheel loader operator on loadability would be a comprehensive survey of the operators with questionnaires, but this does not give accurate individual information about the digging times at the muck pile. The important influence of the operator on the loading process is even highlighted in the automation industry, as this describes (Dadhich 2015):

> *Operator experience makes a big difference in remote control performance. In manual operation, drivers use their vision, hearing and balance-detecting capacities to judge and make decisions in real-time.*

And it is precisely this classification and evaluation of this vision, hearing and balance-detecting capacities that makes it difficult to link the resulting loading times to the muck pile, as it would be unclear whether a change in the KPIs is due to an actual change in the upstream chain or whether the operator has changed his digging behaviour at the muck pile.

In any case, on the basis of the literature research and the lack of a causal relationship, the use of KPIs that can be attributed to human skills are not recommended in this thesis. The parameters mentioned above give much more reliability, since they are unmodifiable.

Figure 41 shows an outline of combinable parameters which are recommended as KPIs which can be used to assess the emission reduction.

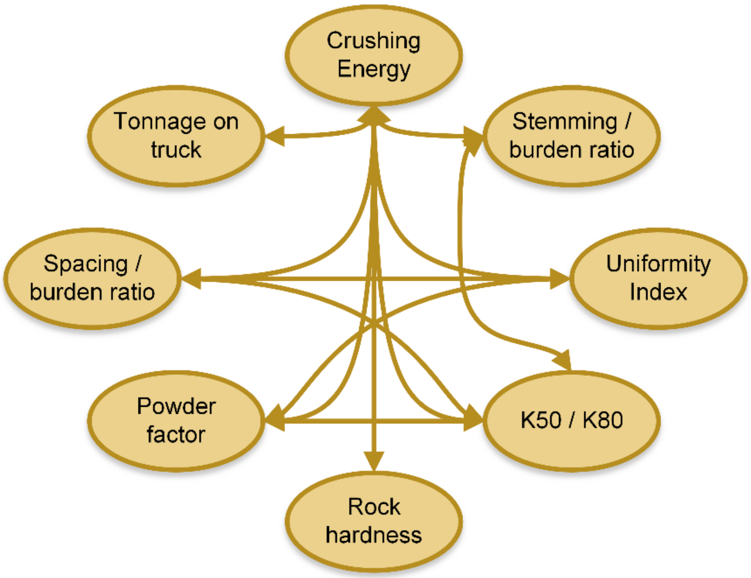


Figure 41: Recommended key performance indicators

5 Conclusion

This thesis examined how to document and evaluate the loading process on blasted muck pile as effectively as possible. Furthermore, a guideline was developed in order to prepare the data obtained during a measurement cycle of several weeks in a structured way.

For the analysis of the loading process it turned out that the use of a camera mounted on the wheel loader and the evaluation of the existing telemetry data are best suited. The loading process was captured characteristically by observing the loading process on site and by evaluating it by means of a video camera. The insights gained from these investigations can be summarized in the statement that the loading process of a wheel loader depends strongly on the operator skills. The different machine handling, the alternating approach of loading the material, the condition of the material itself and finally the personal motivation of the wheel loader drivers make the interpretation of the loading times very difficult, even impossible. An assessment of the loadability of the muck pile should therefore not be based exclusively on the loading times, but rather only in combination with other factors such as fuel consumption and muck pile characteristics. Thereby it has to be emphasized that just from the fluctuating influences of the operator the obtained loading performance cannot be assumed as constant and exact, but only as a tendency. In addition to the video recording of the loading times, the aim of this work was to process the existing telemetry data. This was done against the background of developing an iterative assessment system in order to be able to assess the process sequence "upstream - blasting - downstream" with regard to its influences on each other and ultimately its emission reduction. Since the two telemetry datasets "crusher" and "truck" log independently of each other, it was necessary to connect these partially inconsistent datasets with each other by means of a code specially created for them. A database can thus be set up around the linking of the two process locations "crusher" and "loading point" in which all mineralogical, geological, geotechnical and metrological information can be stored.

The code developed and described here provides the basis for clearly processing the wealth of data gained from the SLIM project's measurement cycle.

6 Publication bibliography

6.1 Academic theses - Papers - Books

- Beyglou, Ali (2016): On the Operational Efficiency in Open Pit Mines. Licentiate thesis, comprehensive summary. Luleå University of Technology, Luleå. Department of Civil, Environmental and Natural Resources Engineering. Available online at http://ltu.diva-portal.org/smash/get/diva2:1033097/FULLTEXT01.pdf.
- Bogunovic, Dragan; Kecojevic, Vladislav (2011): Impact of Fill Factor on Dragline Production Rate and Energy Consumption. In *Mining Engineering* 63.
- Burt, Christina N.; Caccetta, Louis (2018): Equipment Selection for Mining. With Case Studies. Cham: Springer International Publishing (Studies in Systems, Decision and Control, 150). Available online at http://dx.doi.org/10.1007/978-3-319-76255-5.
- D. Brunton, I.; M. Thornton, D.; Hodson, R.; Sprott, D. (2011): Impact of blast fragmentation on hydraulic excavator dig time.
- Dadhich, Siddharth (2015): A Survey in Automation of Earth-moving Machines. Luleå: Luleå tekniska universitet (Forskningsrapport / Luleå tekniska universitet).
- Danielsson, Markus (2016): Borehole Dimension Impact on LHD Operation in Malmberget Mine. Master thesis. Lulea University of Technology. Department of Civil, Environmental and Natural Resources Engineering.
- Filla, Reno (2011): Quantifying Operability of Working Machines.
- Filla, Reno (2013): Optimizing the trajectory of a wheel loader working in short loading cycles.
- Filla, Reno; Obermayr, Martin; Bobbie, Frank (2014): A study to compare trajectory generation algorithms for automatic bucket filling in wheel loaders. CVT 2014 Commercial Vehicle Technology Symposium at Kaiseralutern, Germany.
- Hendricks, Carl F. B. (1990): Performance monitoring of electric mining shovels. PhD. McGill University, Montreal, Montreal, Ottawa. Department of Mining and Metallurgical Engineering.
- Ivanova, Radoslava (2015): Investigation on Fragmention by Blasting. The influence of distorted blasthole patterns on fragmentation, roughness of the remaining bench face and blast damage behind it in model scale blasting. Dissertation. Montanuniversitaet Leoben. Chair of Mining ENgineering and Mineral Economics.
- Kanchibotla, Seshadri; Morrell, S.; Valery, Walter; O Loughlin, P. (1998): Exploring the Effect of Blast Design on SAG Mill Throughput at KCGM. Conference: Mine to Mill Conference, At Brisbane, Australia.

- Montanhistorischer Verein Österreich (Ed.) (2012): 1300 Jahre Erzabbau am Steirischen Erzberg 712 - 2012. der Steirische Erzberg - seine wirtschaftliche, soziale und kulturelle Bedeutung. Montanhistorischer Verein Österreich. Leoben.
- Ouchterlony, Finn (2003): Influence of blasting on the size distribution and properties of muckpile fragments, a state-of-the-art review. Lulea University of Technology.
- Schimek, Peter (2015): Improvement of Fragmentation by Blasting. Investigation of the influence of delay-times on the crack development in the surrounding rock, the characteristics of blasted bench face and the fragmentation of the further rows. Dissertation. Montanuniversitaet Leoben, Leoben. Chair of Mining Engineering and Mineral Economics.
- Singh, S. P.; Narendrula, R. (2006): Factors affecting the productivity of loaders in surface mines. In *International Journal of Mining, Reclamation and Environment* 20 (1), pp. 20–32. DOI: 10.1080/13895260500261574.
- Thurley, Matthew J.; Wimmer, Michael; Nordqvist, Andreas: Blast fragmentation measurement based on 3 D imaging in sublevel caving draw-points and LHD buckets at LKAB. In *11th International Symposium on Rock Fragmentation by Blasting* (161).
- Workman, Lyall; Eloranta, Jack (2003): The Effects of Blasting on Crushing and Grinding Efficiency and Energy Consumption. In *Proceedings of the Annual Conference on Explosives and Blasting Technique*.

6.2 Internetdocuments

- BMVIT (2015): Ein Spielplatz für Forschung. Edited by Bundesministerium für Verkehr, Innovation und Technologie. Infothek. Available online at https://infothek.bmvit.gv.at/forschungszentrum-zentrum-am-berg-spielplatz-fuer-forschung/, updated on 11/4/2018.
- Eurostat (2018): Mining and quarrying statistics - NACE Rev. 2. Eurostat. Available online at https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Mining_and_quarrying_statistics_-_NACE_Rev._2, updated on 11/3/2018.
- Fernuniversität Hagen: Kurs: Deskriptive Statistik. Available online at https://www.fernuni-hagen.de/KSW/neuestatistik/content/MOD_23196/html/comp_23464.html, checked on 4/2/2019.
- James Stoddart (2016): Best Practice in Data Management. Available online at https://www.miningmagazine.com/innovation/news/1263889/practice-management, updated on 12/8/2016.
- NASA Shuttle Radar Topography Mission (2018): Reliefkarte_Österreich.png. Available online at https://upload.wikimedia.org/wikipedia/commons/4/45/Reliefkarte_%C3%96sterreich.png, updated on 5/8/2018, checked on 1/8/2019.

- Petr, Vilim; Hissem, Bill (2015): Loading and Hauling. Edited by Quarry Academy. Available online at https://www.911metallurgist.com/blog/wp-content/uploads/2015/01/Loading-Hauling.pdf, updated on 9/11/2018, checked on 9/11/2018.
- VA Erzberg GmbH (2018): Geologie - Erzführende Formationen. Available online at http://www.vaerzberg.at/erzproduktion/geologie.html, updated on 9/30/2018, checked on 9/30/2018.

7 List of Figures

Figure 1: Share of enterprises in the mining and quarrying industry by employment size class in EU-26 (without Malta, Cyprus) in 2016 – Detail see Appendix - (Eurostat 2018) 1

Figure 2: Location of Eisenerz and aerial view of Erzberg (adopted from: NASA Shuttle Radar Topography Mission 2018; BMVIT 2015).. 3

Figure 3: Geological section through the Styrian Erzberg (Montanhistorischer Verein Österreich 2012).. 4

Figure 4: Ideal loading geometry (Petr and Hissem 2015)... 11

Figure 5: Main cycle types and sub-variants (Depicted load receiver position and orientation is just an example.) (Filla 2013).. 11

Figure 6: Simplified power transfer and control scheme of a wheel loader (Filla et al. 2014). 13

Figure 7: Trajectory Generation: Slice of constant thickness (Filla et al. 2014)...................... 14

Figure 8: Trajectory Generation: horizontal penetration and tilting (Filla et al. 2014) 14

Figure 9: Trajectory Generation: Parametric parabola (Filla et al. 2014)............................... 15

Figure 10: Trajectory Generation: stepwise filling of the bucket (Filla et al. 2014).................. 15

Figure 11: Match factor (Burt and Caccetta 2018)... 17

Figure 12: Operational areas with active rounds (schematic)... 19

Figure 13: Loading sequence – 4 dumps onto truck.. 21

Figure 14: Loading process observed from a bench above.. 22

Figure 15: Loading process observed from a bench above.. 23

Figure 16: Loader with camera mounted on top of operator’s cabin....................................... 24

Figure 17: Start of digging phase... 25

Figure 18: End of digging phase.. 25

Figure 19: Present available data of each truck unit.. 26

Figure 20: Original power curve of the crusher.. 27

Figure 21: Present available data of each loader unit 28
Figure 22: Share of total manoeuvring times per operator 31
Figure 23: Excerpt of the first 5 rows from a prepared crusher raw file 32
Figure 24: Excerpt from a prepared truck telemetry file 33
Figure 25: UTF-8 Encoding of truck raw data 35
Figure 26: Add folders and subfolders 35
Figure 27: Energycurve with marked single dumps for crusher 2 37
Figure 28: Energycurve with marked single dumps for crusher 3 37
Figure 29: Coordinative area allocation of crushers 2 and 3 39
Figure 30: Load cycles with the assigned trucks 42
Figure 31 : Amount of digging rounds per operator 44
Figure 32: Digging times per operator – DIG1 45
Figure 33: Digging times per operator – DIG2 45
Figure 34: Digging times per operator – DIG3 46
Figure 35: Digging times per operator – DIG4 46
Figure 36: Correlation of digging times with overall cycling times per operator 48
Figure 37: Correlation of digging times with manoeuvring times per operator 49
Figure 38: Geometrical muck pile situation 50
Figure 39: Overall average digging time per operator and point of muck pile attack 51
Figure 40: Measured parameters for KPI development 52
Figure 41: Recommended key performance indicators 54
Figure 42: Recommended computer arrangement for a convenient video analysis 63

8 List of Tables

Table 1: Work packages of the SLIM project 2

Table 2: Timestamps according to time lapse camera 30

Table 3: Net digging rounds 30

Table 4: Filename explanation of truck telemetry 34

Table 5: Cumulative Output table 41

Table 6: Correlation of digging times with overall cycling times per operator 48

Table 7: Correlation of digging times with manoeuvring times per operator 49

Table 8: Number of enterprises in the mining and quarrying industry by size classes in EU-26 (without Malta, Cyprus) in 2016 - (Eurostat 2018) 61

Table 9: Time lapse camera settings 63

10 Appendix A

	Number of persons employed					Total
	0-9	10-19	20-49	50-249	> 249	
Austria	226	49	51	17	5	348
Belgium	144	:	16	7	:	167
Bulgaria	224	47	38	34	14	357
Croatia	170	26	23	7	2	228
Czechia	281	43	33	28	14	399
Denmark	176	14	14	8	2	214
Estonia	102	16	16	10	2	146
Finland	734	61	18	11	5	829
France	1.335	168	128	36	8	1.675
Germany	966	361	288	120	19	1.754
Greece	626	25	15	11	4	681
Hungary	328	39	30	:	:	397
Ireland	351	34	18	:	:	403
Italy	1.686	288	117	45	4	2.140
Latvia	211	21	23	14	1	270
Lithuania	75	15	22	14	0	126
Luxembourg	3	3	2	2	0	10
Netherlands	391	18	20	29	9	467
Poland	1.673	137	98	104	44	2.056
Portugal	830	123	74	14	4	1.045
Romania	745	173	95	47	16	1.076
Slovakia	166	25	19	:	:	210
Slovenia	69	19	6	4	1	99
Spain	1.614	239	139	42	12	2.046
Sweden	667	24	21	6	3	721
United Kingdom	869	107	111	65	47	1.199
Number	**14.662**	**2.075**	**1.435**	**675**	**216**	**19.063**
Share	**77%**	**11%**	**8%**	**4%**	**1%**	**100%**

Table 8: Number of enterprises in the mining and quarrying industry by size classes in EU-26 (without Malta, Cyprus) in 2016 - (Eurostat 2018)

11 Appendix B

The VLC player was utilised for the video technical evaluation. The following adjustments were made to the keyboard shortcuts. The German terms are used because the computers used for this thesis were running German language software

Action	Shortcut	Description
Abspielen / Pause	Space	
Normale rate	=	
Schneller (fein)	+	0,1 stepwise faster
Langsamer	-	0,1 stepwise slower
Sehr kurzer Sprung zurück	Shift + Left	30 sec. (Timestamp related) backward
Sehr kurzer Sprung vorwärts	Shift + Right	30 sec. (Timestamp related) forward
Kurzer Sprung zurück	Alt + Left	100 sec (00:01:40) (Timestamp related) backward
Kurzer Sprung vorwärts	Alt + Right	100 sec (00:01:40) (Timestamp related) forward
Nächstes Bild	e	One frame ahead (frame back not possible)

In order to determine the loading times from the TLC videos as time-saving as possible, a recommendation for fast execution is given here.

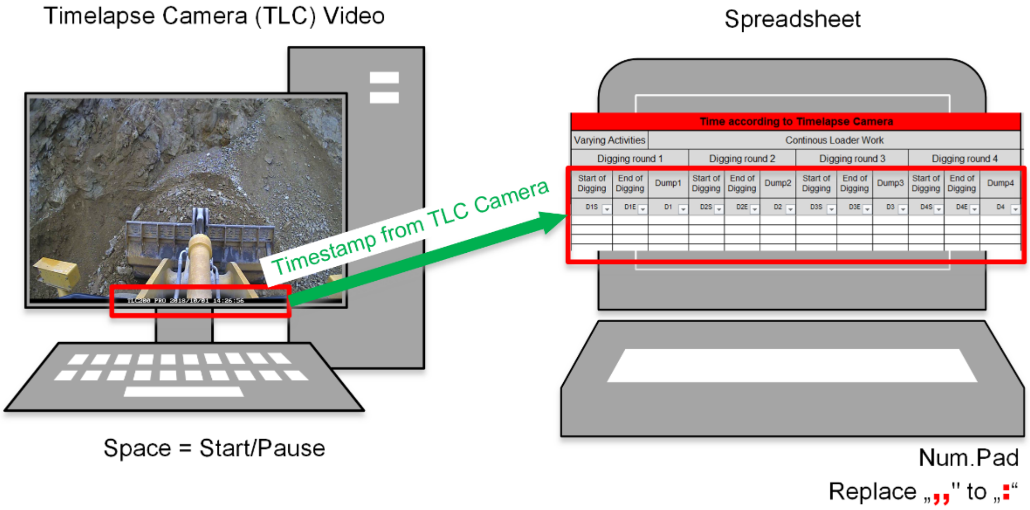
If possible there should be 2 different computers which can be operated independently from each other. On one computer the video recording is running, on the other the timestamp is written to the prepared EXCEL table (Figure 42).

Due to the fact that the video recording is a time lapse, the playback runs at increased speed. During the evaluation, a speed of 0.5-0.6 times of the original speed was found to be sufficient to clearly identify the delimitation of a process. To jump to the next cell on the right instead of one cell down in the spreadsheet when you press the enter key, this setting has also been changed in EXCEL:

1) Click on Excel Options and move to the Advanced tab. (erweitert)
2) Go to the "Editing options" section. (Bearbeitunsgoptionen)
3) Check the "After pressing Enter,move selection" box.
4) In the Direction tab: expand the options and select "Right"

In order to facilitate the input of the times, an adaption in the MS excel was also used here. The colon is entered on the numeric keypad of the keyboard by entering a double comma. For this purpose the following adjustments must be made.

1) Click on Files (Datei)
2) Excel Options
3) Go to “Proofing” (Dokumentprüfung)
4) Go to Auto Correct Options (AutokorrekturOptionen)
5) Replace “,,” (=2 * comma) by “:”



Time according to Timelapse Camera											
Varying Activities		Continous Loader Work									
Digging round 1			Digging round 2			Digging round 3			Digging round 4		
Start of Digging	End of Digging	Dump1	Start of Digging	End of Digging	Dump2	Start of Digging	End of Digging	Dump3	Start of Digging	End of Digging	Dump4
D1S	D1E	D1	D2S	D2E	D2	D3S	D3E	D3	D4S	D4E	D4

Figure 42: Recommended computer arrangement for a convenient video analysis

Time Lapse Camera Settings:

Time Intervall	1 sec
Frame rate	10 fps
White Balance Mode	Auto
Image Quality	Good
Scene	Daylight
HDR Range	Medium
Exposure	Centered
Low Light recording	ON
Time stamp	ON

Table 9: Time lapse camera settings

12 Appendix C

12.1 ImportCrusherData.xlsm:

Instructions for the preparation of the crusher data for the Matlab code:

1.1. Open the file "ImportCrusherData.xlsm" as well as the original raw file from the crusher database received by the VA Erzberg GmbH.

1.2. Raw file: Sheet1 (=Crusher2):
Select A2:B2 (DateTime column & Crusher2 column) →Strg+Shift+Down → Copy

1.3. File: ImportCrusherData.xlsm:
Select cell A2 → Paste

1.4. Raw file: Sheet2 (=Crusher3):
Select B2 (Crusher3 column) →Strg+Shift+Down → Copy

1.5. File: ImportCrusherData.xlsm
Select cell C2 → Paste

1.6. Click FORMAT Button

1.7. Click EXPORT Button: Rename last occurring date in predefined filename

1.8. In order to create a blank template again click on the delete button.

12.2 Matlab Scripts:

12.2.1 EnergyCalculationCrusher:

```
% XXXXXXXXXXXXXXXXXXXXXXXXXX
% EnergyCalculationCrusher.m
% XXXXXXXXXXXXXXXXXXXXXXXXXX

% Calculating the energy-consumption of the crushers (crusher 2 and crusher
% 3)

% This code plots the load of the crusher data over the time and
% calculates the overall energy usage (except the energy of the idling
% crusher)

% (c) Robert Schleicher, 2019

% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
% BEFORE RUNNING THE SCRIPT, UPDATE RANGE IN LINE A.1
% BEFORE RUNNING THE SCRIPT, UPDATE PATH IN LINE A.7
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Preparing the workspace

```
clc;
clear;
close;
```

SECTION A.1:

Getting all the needed data from an Excel sheet into the workspace

```
% Here, the command "readtable" was used, because it is easier to handle
% than "xlsread"

% The data gets stored in a table (only column A to C are needed in this
% code script)

%T = readtable('Crusher_20181001-20181004.xlsx', 'Range', 'A1:C5761','ReadVariableNames',true);
[fileName, filePath] = uigetfile({'*.xlsx', 'Text Files'}, 'Select Data File');

cd(filePath);
```

```
filePath = fileName;
%filePath = 'K1181002043000_K17_UTF-8.TXT';

T = readtable(filePath, 'Range', 'A1:C5761','ReadVariableNames',true);
% First column: date + time (Format: "dd.MM.yyyy HH:mm:ss")
% Second column: crusher2
% Third column: crusher3
% Range definition: (One Day has 5760 rows)
%   Day 1: A1:C5761
%   Day 2: A5761:C11521
%   Day 3: A11521:C17281
%   Day 4: A17281:C23041
%   Day 5: A23041:C28801
%   Day 6: A28801:C34561
%   Day 7: A34561:C40321
%Notice: be shure to use "HH:mm:ss" for the time format instead of
%"hh:mm:ss", otherwise Matlab interprets "00:00" as "12:00" (only important
%when using xlsread, though)
```

SECTION A.2:

Subtract the baseload that the crusher uses when idling

```
% During idling, the crusher uses a load of around 24[Kw], which has to be
% subtracted from each value in the corresponding column of the crusher
% If the value falls below 0 after this subtraction, it is set to 0.

% Get the number of rows of the imported table
numRowT = height(T);

% Loop through the table "T".
% Notice: Arrays start with "1" in MATLAB!
% T{i,2} stands for crusher 2
% T{i,3} stands for crusher 2

for i=1:numRowT
    if T{i,2}-24 >0
        T{i,2} = T{i,2} - 24;
    else
        T{i,2}=0;
    end

    if T{i,3}-24> 0
        T{i,3} = T{i,3} - 24;
    else
        T{i,3} =0;
```

```
    end
end
```

SECTION A.3:

```
% Extraction from Row 1 Seconds, Minutes, Hours

T{:,4} = second(T{:,1});
T{:,5} = minute(T{:,1});
T{:,6} = hour(T{:,1});

% calculate the time increments:
t_increm =abs( T{1,4} - T{2,4});

% T{a,7} cumulates the seconds
% T{a,8} formats the date
% T{a,9} formats the time

for a = 1:numRowT

   T{a,7} = T{a,4} + 60*T{a,5} + 3600 * T{a,6};
   T{a,8} = datetime(T{a,1},'InputFormat', 'dd.MM.yyyy HH:mm:ss',
'Format','dd.MM.yyyy');
   T{a,9} = datetime(T{a,1},'InputFormat', 'dd.MM.yyyy HH:mm:ss', 'Format',
'HH:mm:ss');

end
```

SECTION A.4:

```
% Rename the columns of the table T, in order to get a nice overview and
% thus a better understanding

T.Properties.VariableNames([1 2 3 4 5 6 7 8 9]) = {'DateAndTime' 'Crusher2'
'Crusher3' 'Seconds' 'Minutes' 'Hours' 'CumulativeSeconds' 'Date' 'Time'};

%area_Energy2 = t_increm * trapz(t(20:56),1);
```

SECTION A.5:

Analysing the character of the crushing curve

```
% Determines all ascending edges of crusher 2 & 3
xvalascending2 = [];
xvalascending3=[];
```

```
% indexes the row-numbers of all ascending edges of crusher 2 & 3
indexascending2 = [];
indexascending3 = [];

% Determines all descending edges of crusher 2 & 3
xvaldescending2 = [];
xvaldescending3 = [];

% indexes the row-numbers of all descending edges of crusher 2 & 3
indexdescending2 = [];
indexdescending3 = [];

% assigns a time and an index to each ascending/descending energy value
% for crusher 2 and 3

for b = 1:numRowT-1
   if T{b,2} == 0 && T{b+1,2} >0
      xvalascending2 =[xvalascending2 T{b,9}];
      indexascending2 = [indexascending2 b];

   end

   if T{b,2} > 0 && T{b+1,2} == 0
      xvaldescending2 = [xvaldescending2 T{b,9}];
      indexdescending2 = [indexdescending2 b];

   end

   if T{b,3} == 0 && T{b+1,3} > 0
       xvalascending3 =[xvalascending3 T{b,9}];
       indexascending3 = [indexascending3 b];

   end

   if T{b,3} > 0 && T{b+1,3} == 0
      xvaldescending3 = [xvaldescending3 T{b,9}];
      indexdescending3 = [indexdescending3 b];

   end
end

% The following lines of code in this section, that contain the if
% statements are mandatory, since the given data in the .excel files may
% start or end with an ascending or descending part of an energy-consumption
```

```
% curve. In other words, the following if-statements should deal with an
% incomplete data set and remove the parts of the energy-calculation-curve
% that are not complete
% NOTE: number of ascending x-values should be equal to the
% number of descending x-values!!
if xvalascending2(1) > xvaldescending2(1)
   xvaldescending2(1) = [];
   indexdescending2(1) =[];
end

if xvalascending2(length(xvalascending2))> xvaldescending2(length(xvaldescending2))
   xvalascending2(length(xvalascending2)) = [];
   indexascending2(length(xvalascending2))=[];
end

if xvalascending3(1) > xvaldescending3(1)
   xvaldescending3(1) = [];
   indexdescending3(1) =[];
end

if xvalascending3(length(xvalascending3))> xvaldescending3(length(xvaldescending3))
   xvalascending3(length(xvalascending3)) = [];
   indexascending3(length(xvalascending3))=[];
end
```

SECTION A.6

Calculation the Area under the crusher curve Identifying all the singledmps with predefined time intervalls

```
%---Crusher 2

% preallocate an empty array containing the computed values for the areas
% under the data-curve (these areas are then equal to the energy consumption
% Note: preallocation is a better coding practice, that saves memory, which
% makes the programme run faster (since the array does not have to be
% reallocated in each iteration of the for-loop

areasEnergyCrusher2 = [];
plotxAsc2=[];
plotxDesc2=[];
T{:,10} =0;
T{:,11} =0;

for z = 1: length(xvalascending2)
```

```
   % Decide, whether the curve should be used for the calculation of the
   % energy (everything under 00:04:00 (=240[s]) and over 00:10:00 (=600[s])
   % should not be considered and hence, omitted)
   if (T{indexdescending2(z),7}-T{indexascending2(z),7} > 240) &&
(T{indexdescending2(z),7}-T{indexascending2(z),7} < 600)
    plotxAsc2 = [plotxAsc2; xvalascending2(z)];
    plotxDesc2 = [plotxDesc2; xvaldescending2(z)];

   % Calculating the area here gives back the area in the unit [kWs]
   % However, since Energy has the unit [kWh], there has to be another
   % computation to get from [kWs] to [kWh]
   % The conversion factor is the following: 1 [kWh] = (1/3600) [kWs]
   areasEnergyCrusher2 =[areasEnergyCrusher2 (t_increm/3600) *
trapz(T{indexascending2(z):indexdescending2(z),2})];
   T{indexascending2(z)+1, 10} = (t_increm/3600) *
trapz(T{indexascending2(z):indexdescending2(z),2});
   T{indexascending2(z)+1,11} = T{indexdescending2(z),7} - T{indexascending2(z),7};

   % Note: the numerically calculated integral (which equals the area under
   % the curve has to be multiplied by the time increment (t_increm), which
   % is 15 [s] in this case (since 15[s] is the interval between the
   % measurements

   end

end

%---Crusher 3

areasEnergyCrusher3 = [];
plotxAsc3=[];
plotxDesc3=[];
T{:,12} = 0;
T{:,13} =0;

for c = 1: length(xvalascending3)

   % Decide, whether the curve should be used for the calculation of the
   % energy (everything under 00:04:00 (=240[s]) and over 00:10:00 (=600[s])
   % should not be considered and hence, omitted)
   if (T{indexdescending3(c),7}-T{indexascending3(c),7} > 240) &&
(T{indexdescending3(c),7}-T{indexascending3(c),7} < 600)
    plotxAsc3 = [plotxAsc3; xvalascending3(c)];
    plotxDesc3 = [plotxDesc3; xvaldescending3(c)];
```

```
   % Calculating the area here gives back the area in the unit [kWs]
   % However, since Energy has the unit [kWh], there has to be another
   % computation to get from [kWs] to [kWh]
   % The conversion factor is the following: 1 [kWh] = (1/3600) [kWs]
   areasEnergyCrusher3 =[areasEnergyCrusher3 (t_increm/3600) *
trapz(T{indexascending3(c):indexdescending3(c),3})];
   T{indexascending3(c)+1, 12} = (t_increm/3600) *
trapz(T{indexascending3(c):indexdescending3(c),3});
   T{indexascending3(c)+1,13} = T{indexdescending3(c),7} - T{indexascending3(c),7};

   % Note: the numerically calculated integral (which equals the area under
   % the curve has to be multiplied by the time increment (t_increm), which
   % is 15 [s] in this case (since 15[s] is the interval between the
   % measurements
   end

end

% T.Properties.VariableNames([10 11]) = {'EnergyConsumptCrush2',
'EnergyConsumptionCrush3'};
```

SECTION A.7

Save the table T to a "*.mat" file for further usage in the next script

```
% Create a filename
% The idea is to put the name of the truck together with the time (when the
% code was run) and use it as the distinctive filename
datetme = datetime(T{1,8}, 'InputFormat', 'dd.mm.yyyy', 'Format', 'ddMMyyyy');
% Convert the datetime to a string
datestring = string(datetme);
% put together the date and the char 'CrusherData'
saveFileName = strcat(datestring,'CrusherData');
% Set the path, where the .mat-file (table T) should be stored
savePath =
'C:\Users\Robert\Documents\Uni\1_Master\Bergbaukunde\02_Diplomarbeit\MATLAB\SavedCru
sherData\';
% concatenate the string of the path and the savefilename
saveFileNameWPath = strcat(savePath,saveFileName);
%Save the output
save(strcat(saveFileNameWPath, '.mat'), 'T');

%save('CalculatedData.mat', 'T');
```

SECTION A.8:

Plot and show the peaks, that are beeing considered in the calculation

```
%----- Crusher 2
%PLOTS optimised for a resolution of 1980 x 1080 pixel

% Subplot helps to create several single plots in one figure
fig1 = figure;
subplot(5,1,1);
area1=area(T{:,9}, T{:,2});
area1.FaceAlpha = 0.3; % transparency of area
xlim([datetime(T{1,1}) datetime(T{961,1})]);
hold on;
plot(plotxAsc2, zeros(1,length(plotxAsc2)),'ro');
plot(plotxDesc2, zeros(1,length(plotxDesc2)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

title('CRUSHER 2','fontsize', 18);
% xlabel('Time');
y=ylabel('Power Consumption [kW]');
set(y, 'Units', 'Normalized', 'Position', [-0.035, -2.9, 0], 'FontSize',16);
grid on;
legend({'Power Consumption Crusher', 'Start Peak', 'End Peak'},'FontSize',14,
'Location', 'best');

subplot(6,1,2);
area2=area(T{:,9}, T{:,2});
area2.FaceAlpha = 0.3;
xlim([datetime(T{961,1}) datetime(T{1921,1})]);
grid on;
hold on;
plot(plotxAsc2, zeros(1,length(plotxAsc2)),'ro');
plot(plotxDesc2, zeros(1,length(plotxDesc2)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,3);
area3=area(T{:,9}, T{:,2});
area3.FaceAlpha = 0.3;
xlim([datetime(T{1921,1}) datetime(T{2881,1})]);
grid on;
hold on;
plot(plotxAsc2, zeros(1,length(plotxAsc2)),'ro');
plot(plotxDesc2, zeros(1,length(plotxDesc2)), 'bo');
```

```
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,4);
area4=area(T{:,9}, T{:,2});
area4.FaceAlpha = 0.3;
xlim([datetime(T{2881,1}) datetime(T{3841,1})]);
grid on;
hold on;
plot(plotxAsc2, zeros(1,length(plotxAsc2)),'ro');
plot(plotxDesc2, zeros(1,length(plotxDesc2)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,5);
area5=area(T{:,9}, T{:,2});
area5.FaceAlpha = 0.3;
xlim([datetime(T{3841,1}) datetime(T{4801,1})]);
grid on;
hold on;
plot(plotxAsc2, zeros(1,length(plotxAsc2)),'ro');
plot(plotxDesc2, zeros(1,length(plotxDesc2)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,6);
area6=area(T{:,9}, T{:,2});
area6.FaceAlpha = 0.3;
xlim([datetime(T{4801,1}) datetime(T{numRowT,1})]);
grid on;
hold on;
plot(plotxAsc2, zeros(1,length(plotxAsc2)),'ro');
plot(plotxDesc2, zeros(1,length(plotxDesc2)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

%----Crusher3
% PLOTS optimised for a resolution of 1980 x 1080 pixel

% Subplot helps to create several single plots in one figure
fig2 = figure
subplot(5,1,1);
area1=area(T{:,9}, T{:,3});
area1.FaceAlpha = 0.3; % transparency of area
xlim([datetime(T{1,1}) datetime(T{961,1})]);
```

```
hold on;
plot(plotxAsc3, zeros(1,length(plotxAsc3)),'ro');
plot(plotxDesc3, zeros(1,length(plotxDesc3)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

title('CRUSHER 3','fontsize', 18);
% xlabel('Time');
y=ylabel('Power Consumption [kW]');
set(y, 'Units', 'Normalized', 'Position', [-0.035, -2.9, 0], 'FontSize',16);
grid on;
legend({'Power Consumption Crusher', 'Start Peak', 'End Peak'},'FontSize',14,
'Location', 'best');

subplot(6,1,2);
area2=area(T{:,9}, T{:,3});
area2.FaceAlpha = 0.3;
xlim([datetime(T{961,1}) datetime(T{1921,1})]);
grid on;
hold on;
plot(plotxAsc3, zeros(1,length(plotxAsc3)),'ro');
plot(plotxDesc3, zeros(1,length(plotxDesc3)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,3);
area3=area(T{:,9}, T{:,3});
area3.FaceAlpha = 0.3;
xlim([datetime(T{1921,1}) datetime(T{2881,1})]);
grid on;
hold on;
plot(plotxAsc3, zeros(1,length(plotxAsc3)),'ro');
plot(plotxDesc3, zeros(1,length(plotxDesc3)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)



subplot(6,1,4);
area4=area(T{:,9}, T{:,3});
area4.FaceAlpha = 0.3;
xlim([datetime(T{2881,1}) datetime(T{3841,1})]);
grid on;
hold on;
plot(plotxAsc3, zeros(1,length(plotxAsc3)),'ro');
plot(plotxDesc3, zeros(1,length(plotxDesc3)), 'bo');
```

```
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,5);
area5=area(T{:,9}, T{:,3});
area5.FaceAlpha = 0.3;
xlim([datetime(T{3841,1}) datetime(T{4801,1})]);
grid on;
hold on;
plot(plotxAsc3, zeros(1,length(plotxAsc3)),'ro');
plot(plotxDesc3, zeros(1,length(plotxDesc3)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)

subplot(6,1,6);
area6=area(T{:,9}, T{:,3});
area6.FaceAlpha = 0.3;
xlim([datetime(T{4801,1}) datetime(T{numRowT,1})]);
grid on;
hold on;
plot(plotxAsc3, zeros(1,length(plotxAsc3)),'ro');
plot(plotxDesc3, zeros(1,length(plotxDesc3)), 'bo');
set(gca,'FontSize',14, 'linewidth', 1.2)
set(findobj(gcf,'type','line'),'linew',1.5)
```

Just Plot and show the results (without showing the used peaks)

%--------Crusher2-------------- % Plot the curve of the crusher2

%Divide the whole plotted data into 5 equally sized smaller plots, to give %a better overview (to make the peaks easier to distinguish from each %other) plotrange = numRowT/5;

%Subplot helps to create several single plots in one figure subplot(5,1,1); plot(T{1:plotrange,9}, T{1:plotrange,2}); title('Crusher 2'); xlabel('Time'); ylabel('Power Consumption [kW]'); grid on; %hold on; %plot(T{indexascending,9}, zeros(1,length(indexascending)), 'r.');

subplot(5,1,2); plot(T{plotrange+1:2*plotrange,9}, T{plotrange+1:2*plotrange,2});

grid on;

subplot(5,1,3); plot(T{2*plotrange+1:3*plotrange,9}, T{2*plotrange+1:3*plotrange,2}); grid on;

subplot(5,1,4); plot(T{3*plotrange+1:4*plotrange,9}, T{3*plotrange+1:4*plotrange,2}); grid on;

subplot(5,1,5); plot(T{4*plotrange+1:5*plotrange,9}, T{4*plotrange+1:5*plotrange,2}); grid on;

plot(T{:,9},T{:,2}); hold on; plot(xvalascending2, zeros(1, length(xvalascending2)), 'r.'); plot(xvaldescending2, zeros(1, length(xvaldescending2)), 'ro'); grid on; title('Crusher 2'); xlabel('Time'); ylabel('Power Consumption [kW]');

%--------Crusher3--------------

% Plot the curve of the crusher2

%Divide the whole plotted data into 5 equally sized smaller plots, to give %a better overview (to make the peaks easier to distinguish from each %other) plotrange = numRowT/5;

%Subplot helps to create several single plots in one figure fig3 = figure; subplot(5,1,1); plot(T{1:plotrange,9}, T{1:plotrange,3}); title('Crusher 3'); xlabel('Time'); ylabel('Power Consumption [kW]'); grid on; %hold on; %plot(T{indexascending,9}, zeros(1,length(indexascending)), 'r.');

subplot(5,1,2); plot(T{plotrange+1:2*plotrange,9}, T{plotrange+1:2*plotrange,3});

grid on;

subplot(5,1,3); plot(T{2*plotrange+1:3*plotrange,9}, T{2*plotrange+1:3*plotrange,3}); grid on;

subplot(5,1,4); plot(T{3*plotrange+1:4*plotrange,9}, T{3*plotrange+1:4*plotrange,3}); grid on;

subplot(5,1,5); plot(T{4*plotrange+1:5*plotrange,9}, T{4*plotrange+1:5*plotrange,3}); grid on;

12.2.2 MatchCrusherDataInputTXT:

```
% XXXXXXXXXXXXXXXXXXXXXXXXXX
% MatchCrusherDataInputTXT.m
% XXXXXXXXXXXXXXXXXXXXXXXXXX

% This code matches the crusher data to the corresponding truck, in order
% to compute the overall tonnage and energy consumption of each truck
```

Preparing the workspace

```
clc;
clear;
close;
```

SECTION B.1

Getting all the needed data from a .txt-file into the workspace

```
% The given .txt-file contains 5 columns with:
% 1) The same date in each line
% 2) The time (the time-increment between two lines is 2s)
% 3) The name of the corresponding truck and its:
%     - Status
%     - HW
%     - RW
%     - Speed
```

```
%     - HSML
%     - Load
% 4) The following value of the (Status, HW, RW, Speed, HSML or Load)
% 5) A column with "SPONT T_INTERN", this will not be used here

% Unfortunately, the data provided in the .txt are not sorted
% and also inconsistent, which means, that sorting them first has to be
% done in this script

% Load the data via a graphical user interface. This approach has been
% choosen, since it is faster and more hands-on than loading more than one
% .txt file into the workspace and process them; this would take same time

[fileName, filePath] = uigetfile({'*.txt', 'Text Files'}, 'Select Data File');

cd(filePath);
filePath = fileName;
%filePath = 'K1181002043000_K17_UTF-8.TXT';
fileID = fopen(filePath, 'r');
```

SECTION B.2

Formating Step 1:

```
% the function textscan reads the data from a .txt file (in this case)
% and stores them in
aC:\Users\Robert\Documents\Uni\1_Master\Bergbaukunde\02_Diplomarbeit\MATLAB\Matlab_Codecell
% start with line 7, because the previous ones are some sort of
% "Headerlines", which means, they are not used (they simply contain no
% usable data and only depict what the following data looks like)
date = textscan(fileID,'%s %*s%*s%*s%*s', 'Delimiter', ';', 'HeaderLines', 7);
fclose(fileID);

fileID = fopen(filePath, 'r');
time = textscan(fileID,'%*s %s %*s%*s%*s', 'Delimiter', ';', 'HeaderLines', 7);
fclose(fileID);

fileID = fopen(filePath, 'r');
truckNametxt = textscan(fileID, '%*s%*s%s%*s%*s', 'Delimiter', ';', 'Headerlines', 7);
fclose(fileID);

fileID = fopen(filePath, 'r');
```

```
parameter = textscan(fileID,'%*s%*s%*s%s%*s', 'Delimiter', ';', 'HeaderLines', 7);
fclose(fileID);
```

SECTION B.3

Formating Step 2. Extract cells for further sorting and processing

```
% extract the date cell
date = date{1,1};
% get a single date char
singledate = date{1,1};
% extract the time cell
time = time{1,1};
% extract the parameter cell (which contains the 'STATUS_0' or 'STATUS_1'
% char AND the delimiter ';' AND the corresponding status, this has to be
% separated afterwards
parameter = parameter{1,1};
% Extract the truckname
truckNametxt = truckNametxt{1,1};
```

SECTION B.4

Separating the parameter cell to get the status char and the status

```
count = 0;
% separate the information in the parameter cell
for i=1:numel(truckNametxt)
    % the functions exractBefore and extractAfter extracts information
    % before or after a given delimiter
    parameter(i,2) =  extractAfter(truckNametxt(i,1),'.');
    truckNametxt(i,1)  = extractBefore(truckNametxt(i,1), '.');

end
```

SECTION B.5

Create a table with time, indicator and its value

```
% the idea is to create a table, sort its value (since the time is not
% ordered in the .txt file and then delete useless information to form a
% final table in the end (which contains the needed information)
T1(:,1) = cell2table(time);
T1(:,2) = cell2table(parameter(:,2));
T1(:,3) = cell2table(parameter(:,1));
truckNametxt = cell2table(truckNametxt);

% Headerline of the table
```

```
T1.Properties.VariableNames={'Time','Indicator','Value'};

% Here,the table gets sorted, so that the time is in ascending order.
t = sortrows(T1,1);
t{:,4} = datetime(t{:,1}, 'InputFormat', 'HH:mm:ss.S', 'Format', 'HH:mm:ss');
clear T1;

% Remove unneeded information

%remove all lines with the entry 'STATUS_1' (char), since they will not be
%needed afterwards

t(strcmp(t.Indicator, 'STATUS_1') ,:) = [];
```

SECTION B.6

Extract information that is later merged into a final table

```
% this has to be done because the txt file is in a chronologically
% unusable structure
% here, the corresponding lines are extracted, and will be merged
% afterwards, to obtain a tidy table
status0 = t(strcmp(t.Indicator, 'STATUS_0'),3);
hw = t(strcmp(t.Indicator, 'HW'),3);
rw = t(strcmp(t.Indicator, 'RW'),3);
hsml = t(strcmp(t.Indicator, 'HSML'),3);
loadtruck = t(strcmp(t.Indicator, 'LOAD'),3);
speed = t(strcmp(t.Indicator, 'SPEED'),3:4);
```

SECTION B.7

Merge the previously extracted tables to a final table

```
% this table now is ordered correctly, and can be used for further
% computing
T1(1:numel(status0),1) = cell2table(date(1:numel(status0)));
T1{:,2} = speed{:,2};
T1{:,3} = truckNametxt{1:height(T1),1};
T1{:,4} = status0{:,1};
T1{:,5} = str2double(hw{:,1})-5000000;
T1{:,6} = str2double(rw{:,1});
T1{:,7} = str2double(hsml{:,1});
T1{:,8} = str2double(loadtruck{:,1});
T1{:,9} = str2double(speed{:,1});

% Create a "headerline" for the table T1
```

```
T1.Properties.VariableNames={'Date','Time', 'Truck', 'STATUS_0',
'HW','RW','HSML','Load','Speed'};
```

SECTION B.8

In this section, a few corrections of the obtained data in the .txt files have to be done. One major issue of the data in the .txt files is that the loading of the truck is NOT done after the "Status" changes from 3.0 to 4.0, it is in the last row (where the status changes from 3.0 to 4.0) AND the speed of the truck is equal to ZERO! If one was tempted to think, that the last row (where status 3.0 changes to another value) can be taken to get the coordinates of the loading place of the truck, he might get wrong coordinates. This issue is bypassed by considering the last row, where status =3.0 AND the speed = 0.

```
% Get all the rows of table "T1" that contain STATUS_0 == '3.0'
% Status_0 = getting loaded
B = T1(strcmp(T1.STATUS_0, '3.0'),:);

% Delete the rows of table "B", where the speed != 0

B(B.Speed ~= 0,:)= [];

% Get all the rows of table "T1" that contain STATUS_0 == '6.0'
% Status_0 = dumping
B =[B; T1(strcmp(T1.STATUS_0, '6.0'),:)];

% Sort the table, so that the time is in an ascending order
B = sortrows(B, 2);

% Detect the row-number, where status '3.0' changes to '6.0'. All the
% previos rows with STATUS_0 == '3.0' can be deleted
% This is used for identifying the loading point on the bench so that
% the allocation to the dumping point at the crusher can take place

rowsToDel = [];
for a = 1:height(B)-1
   if(strcmp(B{a,4}, '3.0') && strcmp(B{a+1,4}, '6.0') || strcmp(B{a,4},'6.0'))

   else
      rowsToDel = [rowsToDel; a];
   end

end

% Finally delete the mentioned rows
B(rowsToDel,:) = [];
```

```
% Delete the column, that contained the speed, since it is not needed in
% the furhter computations

B(:,9) = [];
```

SECTION B.9

In order to match the coordinates of the loading event with the dumping event, the now obtained data will be separated into LoadDump-cycles. This means, that the truck is loaded (if STATUS_0 == '3.0') and dumps its load (if STATUS_0 == '6.0'), which can be considered as a single dumpLoad-cycle

```
dumpLoadCycleCounter = 0;

for b = 1:height(B)

    if(strcmp(B{b,4},'3.0') == 1)
       dumpLoadCycleCounter = dumpLoadCycleCounter + 1;
    end
    B{b,9} = dumpLoadCycleCounter;

end

% Delete the rows in B, that have dumpLoadCycleCounter =0
% This has to be done, since dumpLoadCycleCounter =0 means, that the data
% starts with Status 6.0. Consequently, this truck has been loaded the day
% before.
% This means in further consequence, if the dumpLoad CycleCounter =1,
% starts the raw file with a loading event and thus a "loading and a
% dumping event" is complete.

B(B.Var9 ==0,:) = [];
```

Section B.10

Examine, WHERE the truck is dumping its load

```
% This is done by checking, wheter the truck is in the close perimeter of
% either the "crusher 2" or "crusher 3"
% With the aid of the coordinates of the both the truck and the crushers,
% the euklidian distance is calculated:
% eukldist = sqrt((xCoordTruck - xCoordCrush)^2 + (yCoordTruck -
% - yCoordCrush)^2)

% The coordinates of the crusher 2 are:
% HW = 266660
```

```
% RW = -108343
hwCrush2 = 266660;
rwCrush2 = -108343;

% The coordinates of the crusher 3 are:
% HW = 266718
% RW = -108451
hwCrush3 = 266718;
rwCrush3 = -108451;

numRowB = height(B);

% Compute the euklidian distance of the truck to crusher2 and crusher3:

for a = 1:numRowB

if(strcmp(B{a,4}, '6.0'))
% Compute the euklidian distance of the truck to crusher2:
B{a,10} = sqrt((B{a,5}-hwCrush2)^2 + ((B{a,6})-rwCrush2)^2);

% Compute the euklidian distance of the truck to crusher3
B{a,11} = sqrt((B{a,5}-hwCrush3)^2 + ((B{a,6})-rwCrush3)^2);
    % Tell, where the load is dumped (crusher2 or crusher3), 50 is a random
    % threshold value
        if B{a,10} < 50
            B{a,12} = "Crusher2";
        elseif B{a,11} < 50
            B{a,12} = "Crusher3";
        else
            B{a,12} = "unknown dumping";

        end
end
end

% Create new columns in the table (for the euklidian distances to the
% crusher2 and crusher3 and name them accordingly
B.Properties.VariableNames([10 11 12]) = {'euklidDistC2' 'euklidDistC3'
'DumpLoadAt'};
```

SECTION B.11

Load the Previously Calculated Table from EnergyCalculationCrusher

```
% In this section, the data from the script "EnergyCalculationCrusher" gets
% loaded, in order to match the data from the trucks with the corresponding
```

```
% crusher
% For the sake of a better understanding, the energy calculation and the
% matching of the crusher data are separately programmed. If it is
% necessary, the code could also be put into a single script.


load('C:\Users\Robert\Documents\Uni\1_Master\Bergbaukunde\02_Diplomarbeit\MATLAB\Sav
edCrusherData\01102018CrusherData.mat');

% Since only the entries of T where the calculated energy is not
% equal to 0 are needed, all the other entries are deleted here

rowsToDelete = [];
for i=1:height(T)

   if(T{i,10} ==0  && T{i,12}== 0)
     rowsToDelete = [rowsToDelete; i];

   end

end

T(rowsToDelete,:) = [];
```

SECTION B.12

```
% Convert the time from "string" to a usable format "time"
B{:,13} = datetime(B{:,2}, 'InputFormat', 'HH:mm:ss', 'Format', 'HH:mm:ss');
% Calculate the cumulative seconds
B{:,14} = second(B{:,13}) + 60*minute(B{:,13}) +3600*hour(B{:,13});

% When inspecting the given data from the crusher and the truck, one might
% notice, that the dumping of the material is first "detected" by the
% crusher. In other words, the data of the truck lags approximately 15-20s
% behind. However, this has to be considered here

% Uncomment these variables, if the tonnage and energy consumption have to
% be calculated cummulatively
% cumTonnageKomatsu = 0;
% cumEnergyConsumptKomatsu = 0;

% set these columns to 0. They are necessary in the nested for loop below.
% the arbitrary value of "0" is then set to "1" only if this line has been
% examined. With this methode, it is guaranteed, that no value gets
% processed two times!
```

```
T{:,14} = 0;
T{:,15} =0;

tonnage = [];
energycon = [];
dumptime = [];
dumpplace = [];
crushingtime = [];



% check, whether the truck has been dumping its load at one of the crushers

% This for-loop calculates the tonnage of each load-dump cycle, the
% energyconsumption of the crusher after the truck has dumped, the time of
% the truck's loading and it's dumping, the place (crusher 2 or 3) of the
% truck's dumping and the total crushingtime
% These computed values are then needed for the output-table
for i=1:height(B)

   % Check, where the truck is dumping its load (crusher 2 or 3)
   if B{i,12} == "Crusher2"
        for c = 1:height(T)
           % The value 120[s] (tolerance) is needed, since the data of the crusher 
is
           % not fully synchronized (ther is a small time-lag) with the
           % data of the truck
            if(B{i,14} - T{c,7}) < 120 && (B{i,14} - T{c,7}) > 0 && T{c,14} ~= 1 && 
T{c,10} > 0
               T{c,14} = 1;
              % Store each value of the calculated tonnage in an array
               tonnage = [tonnage; B{i,8}];
               % Store each value of the crusher's energy consumption in an
               % array
               energycon = [energycon; T{c,10}];
               % Store each value of the time, when the truck dumps in an
               % array
               dumptime = [dumptime; B{i,13}];
               % Store the String "Crusher2" in an array
               dumpplace = [dumpplace; B{i,12}];
               % Store the duration of the crushing-process in an array
               crushingtime = [crushingtime; T{c,11}];
               B{i,15} = 1;

            end
        end
```

```
   end
   % This part does exactly the same as the one corresponding to "Crusher
   % 2"

   if B{i,12} == "Crusher3"
        for b = 1:height(T)
           if(B{i,14} - T{b,7}) < 120 && (B{i,14} - T{b,7}) > 0 && T{b,15} ~=1  &&
T{b,12} >0
            T{b,15} = 1;
            tonnage = [tonnage; B{i,8}];
            energycon = [energycon; T{b,12}];
            dumptime = [dumptime; B{i,13}];
            dumpplace = [dumpplace; B{i,12}];
            crushingtime = [crushingtime; T{b,13}];
            B{i,15} = 1;
        end

    end
    end

end

% Match the data from the truck with the one from the crushers

% Match the Coordinates of the Loading place(HW, RW, HSML)
% with the data from the output table

% Get all the data with STATUS_0 == '3.0'
B_stat3 = B(strcmp(B.STATUS_0, '3.0'),:);
% Get all the data with STATUS_0 == '6.0'
B_stat6 = B(strcmp(B.STATUS_0, '6.0'),:);
% Remove the data from the table B_stat6 that are not considered for
% dumping, this shall only be done if B has more than 14 columns.
[m, n] = size(B);
if(n>14)
B_stat6 = B_stat6(B_stat6{:,15} == 1,:);
end

% Remove all the data from the table B_stat3, that are not considered for
% dumping. Remember, data from B_stat3 contains all the data about the
% location where the truck has been LOADED. Now this data has to be matched
% with the table B_stat6 (the data about the tonnage, kWh/tonnage, energy
% consumption,...) of the corresponding truck
```

```
% Extract the dumploadCycles from B_stat6. These dumploadCycles are the
% cycles that are considered for calculating the energyconsumption of the
% crusher.
% Important note: Komatsu_17 has approx. 20 loadings + dumpings (= 20
% dumploadCycles) on one day, but only 9 of these dumps are considered when
% it comes to the energy calculation of the crusher. This means, that the
% majority of the dumps remain unconsidered. This might be due to the
% multidumps, as it can be seen in the script file
% "EnergyCalculationCrusher".
dumpLoadCyc = B_stat6{:,9};
B_stat3(ismember(B_stat3.Var9,dumpLoadCyc) ~=1,:) = [];
```

SECTION B.13

Create a table that will be presented on the output of this program

```
OutpTableTitles = {'Date', 'LoadTime', 'DumpTime', 'HWload',
'RWload','HSMLload','Truck', 'DumpPlace','Tonnage', 'EnergyConsumption',
'KWhPerTonnage', 'TotalCrushingTime', 'TotalCrushingTimePerTon' };

% If the arrays energycon and therefore tonnage are empty, the truck has
% not been dumping and therefore the program outputs the following message:

if (numel(tonnage) ~=0)
    % if the truck has dumped something: tonnage, energy and energy/tonnage
    % are beeing displayed in a table together with date, time and name of
    % the truck
OutpTable = table(T{1:numel(energycon),8},B_stat3{:,2},
dumptime(1:numel(dumptime),1),B_stat3{:,5},B_stat3{:,6},B_stat3{:,7},
T1{1:numel(energycon),3}, dumpplace(1:numel(dumpplace), 1),
tonnage(1:numel(tonnage),1),
energycon(1:numel(tonnage),1),energycon(1:numel(tonnage),1)./
tonnage(1:numel(tonnage),1),crushingtime(1:numel(crushingtime),1),crushingtime(1:num
el(crushingtime),1)./tonnage(1:numel(tonnage),1),'VariableNames', OutpTableTitles)

% Save the output table

% In this section, the "output-table" OutpTable is saved. The data will be
% stored into a separate folder
% This method has been chosen, since the output tables are stored
% separately and can be accessed with another script in Matlab. This keeps
% the code and the data easier to maintain and to understand

% Create a filename
% The idea is to put the name of the truck together with the time (when the
% code was run) and use it as the distinctive filename
```

```
datetme = datetime(T{1,8}, 'InputFormat', 'dd.mm.yyyy', 'Format', 'ddMMyyyy');

% Convert the datetime to a string
datestring = string(datetme);

% put together the date and the name of the truck with the function strcat,
% that concatenates two strings
saveFileName = strcat(datestring,B{1,3});

% Set the path, where the .mat-file (output-table) should be stored
savePath =
'C:\Users\Robert\Documents\Uni\1_Master\Bergbaukunde\02_Diplomarbeit\MATLAB\SavedTru
ckData\';

% concatenate the string of the path and the savefilename
saveFileNameWPath = strcat(savePath,saveFileName);

%Save the output
save(strcat(saveFileNameWPath, '.mat'), 'OutpTable');

% create an excel file with the combined table
writetable(OutpTable,strcat(saveFileName, '.xlsx'),'Sheet',1);

else
    disp("This truck has neither been dumping into Crusher 2 nor into Crusher 3 on
this day!");
end
```

12.2.3 MergeTables:

Merge the saved tables together and save them as an excel file

```
% The aim of this script is to fetch all the previously calculated output
% tables from the function "MatchCrusherData" and put them together.
```

Prepare the workspace

```
clc;
clear;
```

SECTION C.1

```
% Get a struct, that contains information about the name, folder,etc. about
% the saved file in the folder "SavedData"
a =
dir('C:\Users\Robert\Documents\Uni\1_Master\Bergbaukunde\02_Diplomarbeit\MATLAB\Save
dTruckData\*.mat');

% get the size of the struct, this will be needed to get the number of
% required iterations in the following for loop
[m, n] = size(a);

% Create an empty struct
P = struct([]);

% loop through the struct a
for i=1:m
    %get the folderName out of the folder column in a
   folderName = a(i).folder;
    %get the fileName out of the name column in a
   fileName = a(i).name;
   %merge the folderName and the file name
   fullPath = [folderName, '\', fileName];
   %load the tables and store them in the struct P
   P =[P; load(fullPath)];

end

% Vertically concatenate the obtained tables to a single table
CumulativeOutputTable = struct2cell(P);
CumulativeOutputTable =vertcat(CumulativeOutputTable{:})
```

SECTION C.2

Store the concatenated table in a .xlsx-file

```
savePath = 'C:\Users\Robert\Documents\Uni\1_Master\Bergbaukunde\02_Diplomarbeit\MATLAB\SavedExcelFiles\';
saveFileName = strcat(savePath, 'CumulativeOutputTable');
writetable(CumulativeOutputTable,strcat(saveFileName, '.xlsx'),'Sheet',1);
```