Haul Truck Payload Modelling Using Strut Pressures

by

Joshua C. Henze

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Mining Engineering

Department of Civil and Environmental Engineering

University of Alberta

Abstract

Haul trucks are commonly used by surface mines to transport ore and waste material out of the mine. They account for a significant portion of the total equipment fleet and the maintenance budget. Their payloads are an important consideration when trying to improve truck reliability. Unbalanced payloads cause increased rack, pitch and roll events, resulting in increased failures and lost production.

Excavator operators have a restricted visibility of the truck body during loading and limited aids to assist in balancing these loads. In order to provide a more accurate view of payload distribution, payload modelling software was developed based on previous work by Chamanara & Joseph [1]. Key goals were to increase the efficiency of the algorithms and identify future research required in order to facilitate real time use. Haul truck strut pressures were used to estimate and display the location and shape of the payload within the body of the truck. Two algorithms were used to determine shape and location respectively. Various methods were considered for each algorithm and evaluated based on processing speed and accuracy.

To verify the software, data was gathered from an operating Caterpillar 785C and lab tests using a scale model of a Caterpillar 797B. This data was used to estimate payload distribution through the software model. Comparing the generated results with actual distributions, it was determined that this software could be suitable for use on material that has a relatively uniform distribution. It is expected that its accuracy will degrade for materials that clump and do not flow freely. Based on the evaluation of the software, it was found to be useful for field implementation but required further research and development before it could be implemented for real time use. Required areas of further research and development were identified with solutions suggested to address them.

Acknowledgements

The author would like to acknowledge the support and advice of his supervisors Tim Joseph and Robert Hall, as well as the encouragement of friends and family.

Table of Contents

Table of Tables

Table of Figures

Glossary of Terms

Truck Body - The portion of the truck used to hold material. Also referred to as truck box

Payload - Material held within the truck body

Strut - Suspension cylinder

Centroid - Coordinate referring to the center of volume of a pile of material

Rack - Twisting motion on the truck structure

Pitch - Front to back motion on the truck structure

Roll - Side to side motion on the truck structure

r - Radius (m)

h - Height (m)

V - Volume (m)

θ - Angle of Repose

ρ - Material Density (kg/m^3)

d - Truck Body Depth (mm)

l - Truck Body Length (mm)

w - Truck Body Width (mm)

β - Angle from horizontal to front body slope

α - Angle from horizontal to rear body slope

$\mathbf{x_f}$ - x direction distance from edge of body to front strut (mm)

$\mathbf{y_f}$ - y direction distance from edge of body to front strut (mm)

$\mathbf{x_r}$ - x direction distance from edge of body to rear strut (mm)

$\mathbf{y_r}$ - y direction distance from edge of body to rear strut (mm)

s - Grid Spacing (mm)

FL - Front Left

FR - Front Right

RL - Rear Left

RR - Rear Right

1 Introduction

1.1 Goal

Develop a system to provide additional decision making information to a shovel operator, or shovel control system, in order to assist in creating balanced payloads within truck bodies. It is desired to do this using only the currently available on-board sensors and information, and must be able to function in real time. This system must also be more efficient than previous methods and not reliant on external software. Providing more balanced payloads will increase haul truck reliability by reducing excess stress on the frame and tires, in addition to reducing haul road wear.

1.2 Background and Motivation for Work

The mining industry is always striving to increase productivity and lower operating costs. A major component of operating costs is the maintenance of equipment. In order to lower these maintenance costs, there has been a push to increase the reliability of equipment and reduce the frequency of failures. Haul trucks are a significant portion of the average equipment fleet and as such are major contributors to the costs of maintenance.

Truck payloads play an important role in haul truck reliability. Improper loading can cause excessive wear to both the frame and tires. With ultra-class tires costing well over $100,000 each, this is a very large expense. In addition to direct wear on the truck, improper loading can cause increased deterioration of haul roads, which in turn cause further truck wear through rack, pitch and roll events.

Rack, pitch and roll events caused by unbalanced loading can cause excessive wear to the haul truck and haul routes as well as having a negative impact on the truck operator's health. Reducing the frequency and severity of these events will aid in managing maintenance costs through a reduction in frame cracking and increased tire life. Preventing unnecessary haul road deterioration will result in lower upkeep costs and indirectly reduce maintenance costs for trucks, as uneven hauling surfaces cause an increase in rack, pitch and roll events. Additionally this will reduce the whole body vibration faced by operators and reduce damage to their spine resulting from prolonged exposure.

Available payload distribution information is currently limited to total weight of the load, visible inspection of the load by the shovel operator, and post load scanning and weighting. As shovel operator visibility is often limited, it would be very beneficial to provide operators with a better view of the payload. By giving the operator more information on how the truck is loaded, the operator will be able to better distribute material within the truck box.

Previous research by Chamanara and Joseph [1] suggested theoretical methods to provide additional information to the operator based on haul truck strut pressures. This method will be explored with a focus on increasing efficiency and identifying further research required to for real time field implementation.

1.3 Potential Challenges

As prior research has proven it possible to estimate payload distribution through haul truck strut pressures alone, challenges for this research will be primarily in regards to implementing such a system that has the capacity to function in real time. To achieve this goal, it will be very important to consider processing speed and software efficiency, as previous methods could not

provide results quickly enough for field use. Additionally, this system should be developed without the use mathematical modelling software not traditionally found in industry.

The software makes use of root finding analyses, which can be time consuming. In using such analyses, three things need to be considered in order to optimize processing speed: definition of upper and lower bounds, accuracy of the initial estimate, and determination of subsequent estimates. Of additional concern, is the loop which implements the root finding analysis for correcting the center of gravity of the load. The center of gravity is affected by multiple variables which can complicate the root finding analysis.

Additionally, if the software is to monitor data streams in real time, it must be able to do so without interference or delay from the modelling algorithms.

2 Literature Review

2.1 Introduction

Haul trucks are a crucial element in the operation of a mine or any large scale earthmoving operation. As they constitute a significant portion of an equipment fleet, they also account for a considerable portion of the cost of operating. Payloads can be up to almost 400 tonnes and often account for over half of the total truck weight. This means that payloads can have a significant effect on the condition of the equipment if they are not distributed according to the equipment's designed capacity. As payloads are completely defined by the judgment of the excavator operator, it is critical that they are provided with accurate and useful information to assist them in achieving optimal loading.

Current research on the estimation and measurement of haul truck payloads has been compiled and presented here. Truck rated capacities and load distributions will be discussed as well as the effects of improper loading. Current and proposed sources of information on payloads and volume estimation will be discussed as well as any shortcomings associated with each method.

2.2 Capacity and Payload Distribution

Given the important role that payloads play in equipment productivity, it is important that the correct capacity of a haul truck be known and adhered to. There are two capacities that must be considered when determining how much material a truck will be able to carry: weight and volumetric capacity. Both of these capacities are generally listed in manufacturer specification documents. For highly dense materials the weight capacity tends to be the limiting capacity and truck bodies may have to be left partially filled. Lighter materials, on the other hand, can fill a

truck body before the rated weight capacity has been reached. Under these conditions, the rated volumetric capacity will be the limiting factor.

It has been suggested by Hagenbuch [2] & [3] that this rated capacity is often overestimated by manufacturers. Equipment manufacturers define volumetric capacity based on SAE Standard J1363 [4]. J1363 defines body capacity by dividing the volume into two types: struck capacity (volume below the body line) and heaped capacity (includes volume piled above the body line). The total SAE capacity is shown in Figure 2-1.

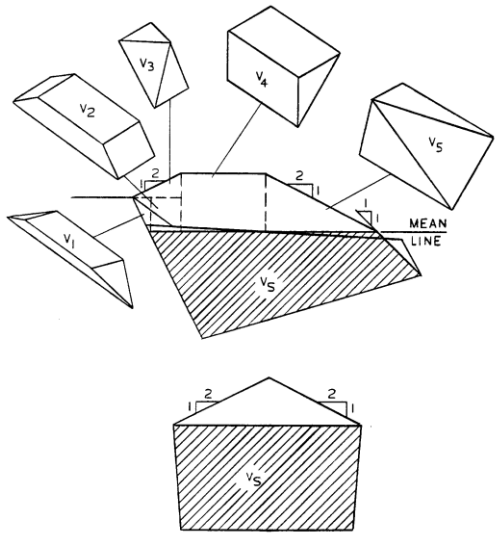


Figure 2-1 SAE J1363 Volumetric Capacity [4]

Struck capacity is defined by the body geometry. In the case of exposed sides, a 1:1 slope is to be extended up and inwards to the top edge, or mean line of the top edge, of the body [4]. However, "few materials will hold a 1:1 slope even momentarily" [2]. Figure 2-2 demonstrates how using this assumption overestimates capacity.

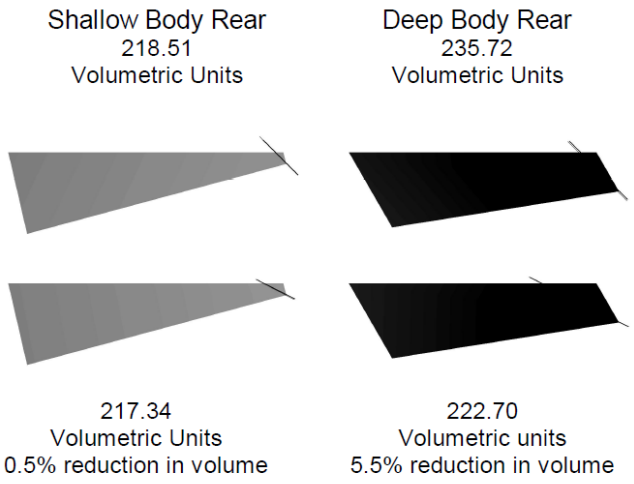


Figure 2-2 The Difference in Body Volumetric Ratings When Going From a Rear 1:1 Slope to a 2:1 Slope [2]

Heaped capacity is calculated by combining the struck capacity with the volume heaped above the box line. Heaped volume is determined by extending a 2:1 slope plane upwards and inwards from the edges of the body. The volume contained within the shape formed by the intersection of these planes is the heaped volume. Total heaped capacity is then the sum of this volume and the struck capacity [4].

There are two problems with this standard, according to Hagenbuch. The transition from 1:1 to 2:1 is unreasonable as granular materials are known to heap at a relatively constant angle. Additionally, the use of flat planes extending from the truck body further overestimates volume, as freely placed material tends to form a conical shape at the material's angle of repose [2]. Hagenbuch provides guidelines on how to calculate the true capacity, which forms a shape similar to Figure 2-3.

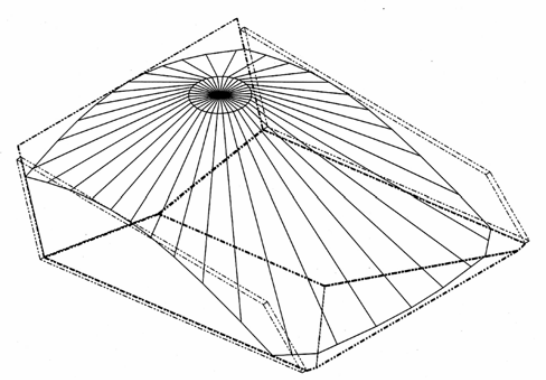


Figure 2-3 Actual Volumetric Truck Capacity [3]

Joseph and Chamanara further explored the capacity of haul trucks, suggesting that, since trucks are loaded in multiple passes, the shape of a payload is actually the combination of a series of intersecting cones [5]. Figure 2-4 shows an image of these.

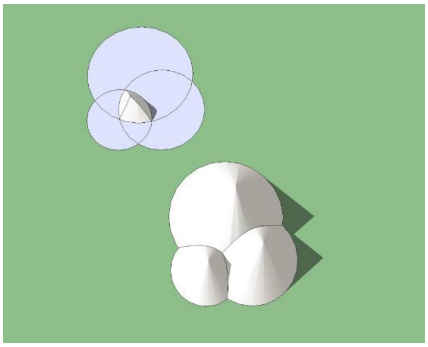


Figure 2-4 Intersecting Cones [6]

Since the first principles method of calculating such a volume is quite complex, Joseph and Chamanara developed a simplified mathematical model, which required computer simulation, to determine the shape of such a payload. This algorithm used MatLab to generate mesh networks representing the truck body and load shapes. The shape of the first load was defined using an assumed loose material density and angle of repose, as well as, a known load location and weight. The volume and centroid of this load could then be determined by summing the load volumes associated with each grid point. The locations of subsequent cones were determined

"by moving the combined first and second pass cones' center of gravity to a common center of gravity location, while honouring the overall load distribution by employing a numerical optimization function" [5]. This distribution could then be used to evaluate the balance of the payload distribution.

Haul trucks are designed to run optimally when their loaded vehicle weight is evenly distributed between all six tires. Since there are two tires on the front axle and four on the rear, this equates to a loaded weight distribution of 33.3% on the front axle and 66.7% on the rear, as shown in Table 2-1.

WEIGHT DISTRIBUTIONS - APPROXIMATE	
Front Axle - Loaded	33.3 %
Rear Axle - Loaded	66.7 %
Front Axle - Empty	47.2 %
Rear Axle - Empty	52.8 %

Table 2-1 Caterpillar 797F Weight Distributions [7]

In order to achieve this distribution, trucks are designed such that the empty weight is split so that the front axle supports 47.2% and the rear axle supports 52.8%. A truck's payload is a defining factor on whether the truck is in balance. Payload constitutes over half of the gross vehicle weight (GVW) of a haul truck. For example, a Cat 797F's payload is 58.3% of its GVW (based on a reported 623 tonne GVW and 363 tonne payload capacity) [7].

2.3 Static vs Dynamic Loading

Joseph [8] explored the difference between static and dynamic truck loading. It has been observed that suspension cylinder, or strut, pressure varies when the truck is in motion. This is a

result of the cylinders extending or compressing as the tires encounter undulations in the running surface. As payload is re-measured from these strut pressures during the transition between the 1st and 2nd gear, it may not give an accurate reading.

Joseph explains that when the truck is not in motion the weight can be determined through a simple sum of the strut loadings. However, when the truck is in motion, there may be additional accelerations that must be considered besides gravity. As such dynamic force must be calculated as follows:

$$F_{Dynamic} = \sum_{i=1}^{4} M_i(g + a_i) \quad (N)$$

Where M_i is the mass associated with an individual strut, g is the acceleration due to gravity ($9.81 m/s^2$) and a_i is the varying acceleration that influences measurements under dynamic conditions.

In addition, Joseph suggested that a reasonable way to consider strut loading would be to express it in terms of gravity, or g, units. The number of g's at any given strut can then be calculated as:

$$\#g_{STRUT_i} = \frac{F_{Dynamic_i}}{F_{Static_i}} = \frac{g + a_i}{g}$$

Using these concepts, Joseph defined rack, pitch and roll as described below.

2.4 Rack, Pitch and Roll

2.4.1 Suspension Cylinders

When calculating rack, pitch and roll, the loading measured at each strut is used. Strut loadings will be abbreviated as follows:

- Front Left = FL
- Front Right = FR
- Rear Left = RL
- Rear Right = RR

As described by Joseph, the front struts are "of a slightly larger diameter than the rear to facilitate greater steering control of the unit". Front struts are designed to each support one sixth of the total load. Rear struts are designed to each support one third of the total load [8].

2.4.2 Rack

Rack is the torsional or twisting motion on the machine. It is defined as:

$$Rack = (FL + RR) - (FR + RL) \quad (kg)$$

Or if using g units:

$$Rack = \frac{(a_{FL} + a_{RR}) - (a_{FR} + a_{RL})}{g}$$

Figure 2-5 shows an exaggerated image of the effect of rack on the frame of a truck.

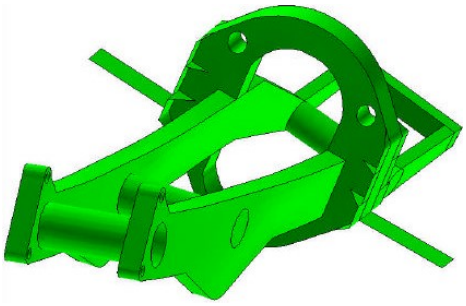


Figure 2-5 Frame Racking [9]

Chamanara studied the effect of payload balance on recorded strut pressures. He performed field tests and examined the resulting strut pressure data. In doing so, he concluded that "The more a truck payload is out of balance, the greater the number of rack events are evident." [6].

2.4.3 Pitch

Pitch is the longitudinal motion on the machine. It is a result of deviation from the recommended 33.3% front and 66.7% rear loading distribution. Pitch is defined as:

$$Pitch = (FL + FR) - (RL + RR) \quad (kg)$$

Or if using g units:

$$Pitch = \frac{(a_{FL} + a_{FR}) - (a_{RL} + a_{RR})}{g}$$

2.4.4 Roll

Roll, also called bias, is the lateral motion on the machine. Ideally trucks are loaded with an even split between left and right sides. Roll is defined as:

$$Roll = (FL + RL) - (FR + RR) \quad (kg)$$

Or if using g units:

$$Roll = \frac{(a_{FL} + a_{RL}) - (a_{FR} + a_{RR})}{g}$$

2.5 Results of Improper Loading

Improper loading can cause rack, pitch, and roll events, with even a 0.5m offset from ideal loading generating 1.3g events while the truck is in motion on level ground [8]. Rack, pitch, and

roll events can then cause many adverse effects including frame failure, tire wear, running surface deterioration and operator injury.

2.5.1 Frame Failure

Whalen and Obaia [9] studied the effects of frame torsion, or racking, on haul truck frames and dump bodies in an effort to design a more resilient dump body. In doing so, they noted that "frame torsion is the limiting factor governing the expected life of the frame structure in most, if not all, mining applications." [9].

Joseph [8] expanded on this statement by examining the data from 44 haulage cycles. During all of these cycles the trucks were under what was considered, "adverse operating conditions, such that it was suspected that the frame life may become compromised." [8]. To demonstrate how much this reduced the life of the frame, Joseph presented an example in which 1 million rack events above 1.5g's would cause a frame failure. Given these values and the haulage data, the frame life would be reduced by an estimated 62%.

2.5.2 Tire Wear

In reviewing tire performance research, Zhou et al [10] found that payload was "a very important factor in tire-truck interactions." [10]. They noted that mines trying to optimize productivity tend to overload their trucks and in doing so need to increase tire pressures. Increased tire pressures then increase heat buildup resulting in shortened tire life. The effect of inflation on tire life is shown in Figure 2-6.

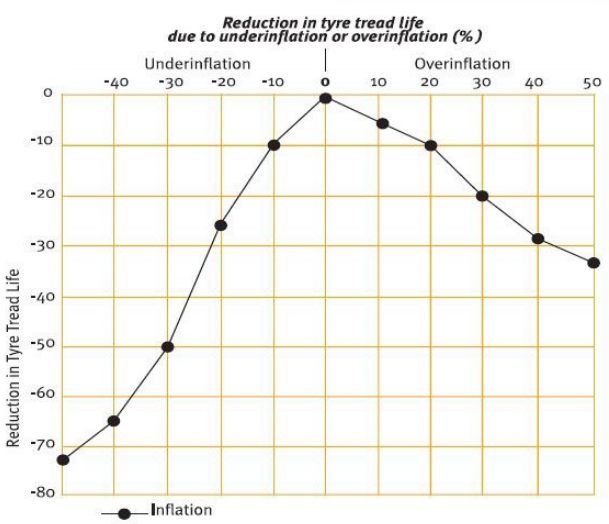


Figure 2-6 Tire Life vs Inflation [11]

Additionally, rack, pitch and roll events can cause tires to be instantaneously overloaded by creating large forces on the tires as a result.

2.5.3 Running Surface Deterioration

Joseph [12] examined the deterioration of oil sands haul roads as a result of equipment interactions. Oil sands are a "strain softening material" [12]. This means that, unlike some materials which compact and get harder, oil sands get softer as equipment drives over it. He found that truck strut pressure could be correlated with passive seismic ground responses. Rack events were defined as those exceeding ±1.5g's. These responses then result in the formation of ruts and hummocks in the road which cause further rack events.

2.5.4 Operator Health (Whole body Vibration)

Berezan compiled research into whole body vibration (WBV) in order to develop a warning system for operators. It has been shown that within the mining industry operators can be exposed to high levels of vibration which exceed the guidelines set by ISO 2631-1. The severity of the exposure is dependent on the magnitude and duration of the WBV. Berezan compiled a

list of health problems which have been found to be connected to WBV, this list is shown in Table 2-2.

Health Problems Related to Whole Body Vibration	
Physiological	**Psychological**
Lower back pain	Chronic fatigue
Lumbago	Nervous irritability
Sciatica	
Generalized back pain	
Intervertebral disc herniation	
Intervertebral disc degeneration	
Cardiovascular disorders	
Osteoarthritis	
Gynecological disorders	
Male sexual disorders	

Table 2-2 Health Effects Encountered in WBV Investigations [13]

Berezan also notes that haul roads which have formed ruts and hummocks can increase the vibration exposure of the operator.

2.5.5 Productivity

Schexnayder, Weber and Brooks [14] performed a study to determine the connection between increasing payload and productivity. They collected data from 54,300 haul truck cycles between October 1996 and May 1997 and covering 14,419 operating hours. This data was collected using Caterpillar's VIMS system. All trucks included in the study were newly purchased Cat 785B trucks.

The nominal payload of the trucks was determined by the mine to be 136 metric tonnes. While normally the manufacturer's rated payload would be used, the mine had installed sideboards on the trucks to increase their volumetric capacity. Due to the weight of the sideboards, the gravimetric payload capacity of the trucks had been reduced.

Schednayder, Weber and Brooks explained that it is common for mines to overload trucks in an attempt to increase productivity. To determine the value of this practice, they plotted productivities measured from their study against the payload weights, as shown in Figure 2-7.

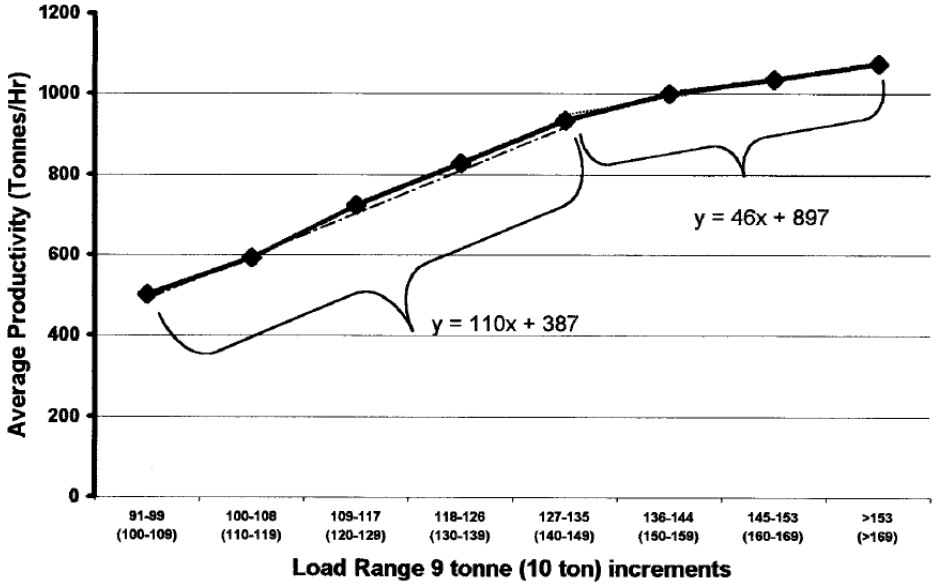


Figure 2-7 Productivity vs Payload [14]

The rate at which productivity increases as a result of increased payload was reduced as payloads exceeded the nominal capacity. Schexnayder, Weber and Brooks speculated that, "the reduction in the rate of production increase with increased payload could be attributed to longer load times and a decrease in the haul unit's loaded travel speed." [14]

2.6 Frequency of Improper Loading

Zhou, Hall & Huntingford [15] used weigh scale studies to collect data from three different mine sites over six separate time periods. The data included three different truck fleets of: Komatsu 830Es, Euclid R260s, and Caterpillar 785Cs. Empty weight, payload weight and gross weight distributions were compared. It was found that imbalanced loads were very frequent, with loads being more likely to be imbalanced between the right and left sides than from front to back.

They noted that, “Loading patterns are a major reason for payload differences between the left and the right side.” [15].

2.7 Measuring Volume of Granular Materials

Research available on the measurement or estimation of the volume of granular materials in mining is fairly limited. For this reason, research and practices done by other industries were also examined. Measurement or estimation of the volume of granular materials is commonly done in many industries. In the construction, mining, and shipping industries the volume of material contained within stockpiles can be an important piece of information. These stockpiles are usually measured by surveyors using GPS or other surveying equipment. Alternatively, stockpiles can be measured using scanners of various types (including laser), aerial photography or LiDAR (Light Imaging Detection and Ranging).

The use of scanners has grown in popularity in many industries. An example includes the use of acoustic 3D scanners for the measurement of granular material level and distribution within storage silos. Imbalanced loads within silos can cause failure, even toppling. As such, acoustic scanners designed specifically to monitor the distribution and level of material are readily available for purchase. Figure 2-8 shows a diagram of a scanner used to monitor silo inventory.

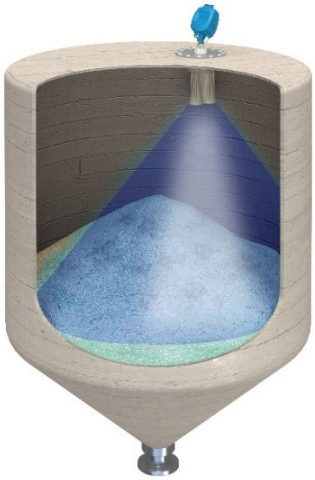


Figure 2-8: Acoustic 3D Scanning of Silo Inventory [16]

The amount of bulk material contained in an oceangoing freight ships is often calculated through draught surveys. The process for these surveys is described in detail in the Code of Uniform Standards and Procedures for the Performance of Draught Surveys of Coal Cargoes [17]. In summary, a draught survey involves measuring the change in displacement of the ship after loading in order to calculate the weight of the material loaded, based on Archimedes' law of buoyancy. If the material has a relatively uniform density the volume can then be estimated.

While useful in their own right, most of these methods cannot be easily translated to mining. Scanners could be mounted on equipment to measure volume, but this method has limitations as discussed in section 2.9.

2.8 Current Sources of Payload Information

2.8.1 Vehicle Monitoring

2.8.1.1 Original Equipment Manufacturer Systems

Equipment manufacturers, OEMs, often provide information systems that record and interpret vehicle sensors to provide valuable information to mine personnel.

Caterpillar's Vital Information Management System or VIMS for short, is described by Caterpillar as, “a powerful tool for machine management that provides operators, service personnel and managers information on a wide range of vital machine functions.” [18]. This is done by recording data from sensors installed on the equipment. VIMS is also set up to monitor these sensors and provide alerts to help detect or prevent mechanical issues with the vehicle.

KOMTRAX Komatsu’s proprietary system, which stands for Komatsu Tracking System, “has made it possible even for people in the office to gain access to and use machine data that was formerly accessible only in the field, such as the current machine position, service meter (hour meter) and fuel gauge readings, machine trouble indications, and consumable parts replacement timings.” [19]. Like VIMS, KOMTRAX makes use of on board sensors to wirelessly provide information to remote users. Figure 2-9 shows an overview of how the KOMTRAX system works.

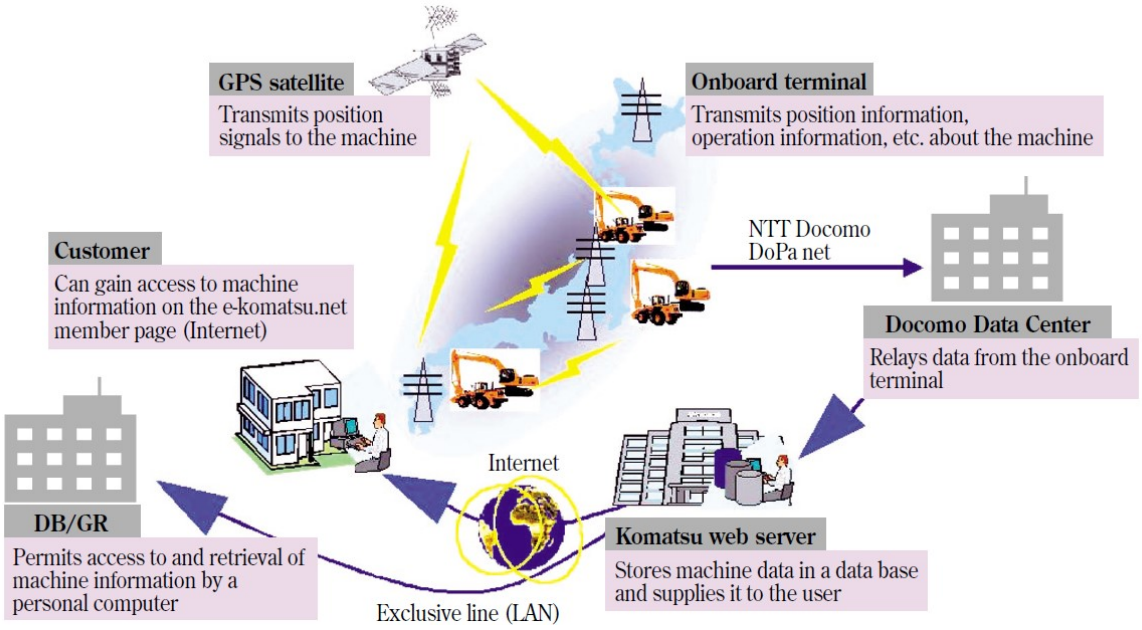


Figure 2-9 KOMTRAX System Configuration [19]

2.8.1.2 Excavator Payload Monitoring

While most systems rely on truck sensors to determine payload, it can also be estimated by determining the payload of individual excavator bucket loads. It has been suggested by Slob [20] that it would be more accurate to estimate shovel payloads. Slob states that shovel measurements can be taken during the swing motion to provide reasonable results, while truck measurements are only considered accurate at the second gear reweigh.

Lipsett [21] suggested that there are two methods to determining shovel bucket payload, "instrument the bucket and assume that motion errors are small, or instrument the machine and calibrate its measurements for standard motions" [21]. Electric cable excavator payloads can be estimated by calculating rope forces based on hoist drive torque. For hydraulic excavators, cylinder pressures can be used to determine payload. For either type of excavator, the output is the mass of material contained within the bucket.

2.8.2 Weigh Scale Studies

Weigh Scale studies are a common industry practice for analyzing truck payload trends and verifying the accuracy of on board equipment weighing sensors. Scales are brought in to a mine site and set up in the field, as shown in Figure 2-10. Once trucks have been loaded they are driven on to the scale and weighed at each tire. Using these weights the payload and weight distribution can be calculated by subtracting the weight of an empty truck. An example of how weigh scale data can be used was presented by Zhou, Hall and Huntingford [15] as described in section 2.6.

Figure 2-10 Weigh Scale [22]

2.9 Proposed Sources of Payload Information

2.9.1 Laser Scanners

Duff proposed the use of laser scanners, placed above haul roads to scan the payloads of passing trucks. His system used two scanners, one to scan the width and the other the length of the truck. However, use of these lasers required the truck to remain under the laser for a minimum of 4.8 seconds, assuming a 12m truck length. This equates to a maximum speed while passing the scanner of 9km/h. [23] Once data had been obtained a surface was generated and volume was calculated by subtracting a surface representative of the empty box. This system has since been implemented by industry as shown in Figure 2-11.

Figure 2-11 Walz Load Scanner [24]

The use of laser scanners was also suggested by Dunbabin & Corke [25] and [26]. They developed a scale model of an electric rope shovel in order to research excavator automation. Dunbabin and Corke installed a laser scanner on the model to provide some visual aid to the shovel. The scanner was used both as a means to map the digging surface and to locate haul trucks and generate an image of their payload. As described in US Patent No 8903689 B2, the laser would scan the truck's tray as the shovel swung into location. This information would then be used to simulate ideal load distributions to be executed by the shovel.

2.9.2 Stereovision

Borthwick [27] developed a system to aid shovels in avoiding collisions with trucks by providing warnings using a stereographic camera. Stereographic cameras acquire data at a significantly faster rate than laser scanners, limited only by shutter speed. A representation of the truck body is created by the stereovision image, which is presented in a collection of x, y and z coordinates. Borthwick also explored the use of these cameras as a means to estimate material volume within

body of the truck. He found that they provided an effective means of generating a payload profile.

2.9.3 Strut Pressures

Modular Mining's patent on a "Load Distribution System for Haulage Trucks" [28] describes the use of haul truck strut pressure sensors to determine the relative position of the center of gravity of the payload. This center of gravity would then be displayed showing the current center of gravity's position relative to the ideal center of gravity. The intent of this display would be to give excavator operators an indicator to aid in balancing payloads.

Chamanara and Joseph [5] [1] and Chamanara [6] expanded on this concept. They used truck strut pressures to calculate the center of gravity of the payload as Modular Mining's patent describes. The shape of the payload was then estimated based on an assumed volume placed per load at the location of the center of gravity. This system is based on the idea that granular materials tend to form a conical shape at an angle of repose based on internal friction angles of the material. The center of gravity was re-measured after each bucket load had been placed and a new load was then generated. Multiple loading passes were taken into account by intersecting cones with a surface representative of the truck body and any previously placed loads.

2.10 Uses of Payload Information

2.10.1 Shovel Decision Making

Excavators both dig material and load the trucks which transport it.

> Shovel operators visually identify the proper location for the shovel load passes in the truck body to achieve a balanced final payload. They do not benefit from any

> loading assist system to create a balanced payload. They mainly rely on their visual interpretation of the payload position within the truck body. [1].

Based on this statement, it is entirely up to the operator's judgement whether a truck will be correctly loaded. Compounding this issue, operator visibility from an excavator cab is often limited making it even more difficult to achieve a balanced payload.

It is even more critical for an automated excavator to receive an accurate representation of the truck's current payload as they do not have an operator to provide the visual interpretation. This is demonstrated by Rowe and Stentz's [29] proposed system for planning and executing the motions of a hydraulic excavator. Their work was focused on the motions that take place after the digging portion of the cycle since, "Most of the research on autonomous excavation has focused on digging without much attention given to the completion of the rest of the task." [29]. They described what information would be required to make such a system functional. One such condition was the ability to determine how the truck is loaded so that their scripts could decide where to place a load.

2.10.2 Planning and Dispatch

2.10.2.1 Fleet Management Systems

Fleet Management Systems (FMS) are available that can provide a wide variety of information based on equipment sensors. FMS are commonly used as aids for mine dispatchers. Dispatchers are provided with location, status and condition information about the equipment. This information can then be used to make decisions on where to direct trucks. Once a dispatcher has sent a command to a truck, the equipment operator is shown that command. They can additionally be used by planners as a means to provide excavator operators with instructions on

how to proceed with digging. These instructions appear on an operator's screen, which they can then use as a guideline to ensure that they are proceeding as intended.

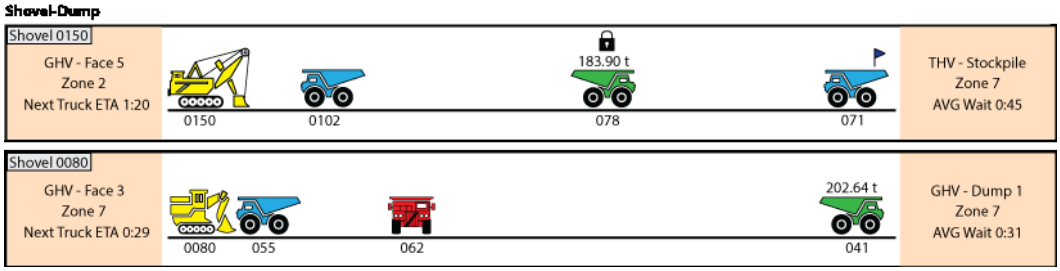


Figure 2-12 Wenco Shovel-Dump [30]

Figure 2-12 shows an example of a FMS display. This particular display is a simple presentation of relative truck position on a haul route, including the current payload.

2.10.2.2 Data Mining

Golosinski & Hu [31] suggested that systems such as VIMS and KOMTRAX could be data mined in order to determine trends regarding equipment condition and performance. Data mining is described by Golosinski & Hu as an "iterative process that involves setting the objectives of the search, selecting and cleaning input data, transforming it, running a mining function and interpreting the results." [31]. This would allow mines to get additional information from manufacturer systems.

While this method could be used for any number of purposes, it could potentially be useful for planners trying to determine issues with payload balance. Algorithms could be defined to monitor sensors for readings that would suggest an improper payload distribution. These flagged events could then be analyzed to determine trends or connections to operator loading tendencies.

2.11 Other Solutions to Load Imbalance

Hardy [32] suggested that the number of passes an excavator uses to load a truck should be considered more carefully. Current industry practice tends to favor four pass loading as an unofficial standard. However, based on Hardy's research, while under some circumstances 4 pass loading may be favorable, in other cases additional passes may be more beneficial. In specific, Hardy compared four pass with six pass loading. One of the deciding factors was haul distance, or the portion of a truck cycle's time spend that is spent being loaded. Additionally, he hypothesized that an increase in loading passes would reduce the frequency of overloading and increase the frequency of balanced payloads [32].

2.12 Weaknesses of Current & Proposed Methods

2.12.1 Vehicle Monitoring & FMS

Information systems such as Caterpillar's VIMS, Komatsu's KOMTRAX, or third party Fleet Management Systems such as Wenco, Minestar or Modular Mining Dispatch provide valuable information for productivity analysis. However in their current form, they only provide limited information. In regards to payload, these systems only provide the current total weight, but can do so in real time. In addition, weight measurements rely on the accuracy of the suspension cylinder pressure sensors.

Excavator payload monitoring systems have similar limitations if used to evaluate truck payload. In addition, any spillover that occurs during dumping will reduce the accuracy of any truck payloads estimated in this manner.

2.12.2 Data Mining & Weigh Scale Studies

While data mining OEM systems for trends and correlations certainly has value, it is time consuming and can only be done on large collections of previously recorded data. It provides no immediate assistance to shovel operators.

Weigh scale studies have similar issues as they only provide data after the payload has been placed. They are however very useful for calibration of onboard sensors and examining loading trends.

2.12.3 Laser Scanners

Laser scanners take time to complete their analysis of a surface. As demonstrated by Duff, passing trucks would be required to limit their speed to 9km/h [23], in doing so productivity is reduced. Walz Scale confirms this speed stating that the scanning process "can be operated at speeds of 25+ mph, however the ideal speed for achieving the greatest accuracy is under 5mph" [33]. Even if this process was quick enough not to require a reduction in speed, it only provides information after the payload has been placed.

The use of laser scanners installed on an excavator has been proven possible by Dunbabin and Corke. However, the slow rate of data collection is a more significant issue with this implementation, as any delays in a shovel's loading cycle will result in a productivity loss. Visibility can be an issue for laser scanners, however newer systems have implemented algorithms that can compensate for issues such and rain and dust. This system also requires the laser scanner to be installed on the excavator. This would require an additional cost to purchase the equipment, as well as additional maintenance costs. Finally laser scanners are fairly fragile pieces of equipment and could be easily damaged in a mining environment.

2.12.4 Stereovision

Stereovision cameras, while very quick to gather data, are known to be limited by visibility. Specifically they tend to encounter issues under conditions such as fog, rain or dust. Additionally stereovision systems require good lighting, which can prove problematic during night shifts. Stereovision cameras would have to be installed on excavators and face the same issues regarding cost, maintenance, and durability as laser scanners.

2.12.5 Loading Patterns

As suggested by Hardy, increasing the number of passes may increase the likelihood of payload balance. This could prove to be an improvement, but again does not directly provide additional information to operators in order to better load trucks.

2.12.6 Strut Pressures

Using strut pressures to obtain payload data does not require the installation of any additional hardware, only currently on board equipment sensors. This means that no additional cost or maintenance will be required. In addition, data collection is not limited by scanning speed or visibility. Strut sensors are very durable, but can require recalibration to ensure reading accuracy. This would not be an issue for payload modelling, as a system would rely on the difference between the current and tare readings. However, this method only provides an estimate of payload distribution not an actual image.

3 Methodology

In order to provide shovel operators with more information regarding truck payloads, modelling software will be developed. This software will be developed to as a standalone system, not requiring the use of external mathematical modelling software such as MatLab. It will be designed with a focus on efficiency and processing speed in order to facilitate real time use. As such, various algorithms will be considered and compared to determine the best solution.

Lab tests will be performed on a scale model of a Cat 797B haul truck body in order to obtain data sets useful for testing the modelling software. Field data from a Cat 785C truck will also be used for model testing. Based on these tests, any limitations of the software will be highlighted. To address these limitations, recommendations will be made to direct future research and development in order to achieve a system suitable for real time field implementation.

4 Modelling Payload

4.1 Introduction

In order to model the shape of the payload based on strut pressure data, modelling software was developed. Based on strut pressure data, known truck dimensions and material properties, the location of the payloads centroid and its expected volume can be determined. Using these values an algorithm was determined to find the shape of the payload. However, the shape of a payload is not a true cone, and therefore its centroid is not located at the peak as it would be for a simple cone. This is a due to the shape of haul truck body floors, which are not simply a flat plane but generally the combination of two inclined planes. For this reason a search algorithm was also developed to determine the location of a payload's peak such that its centroid was as close as possible to the measured load centroid. An example of this deviation is shown in Figure 4-1. Several variations of these algorithms were evaluated and compared.

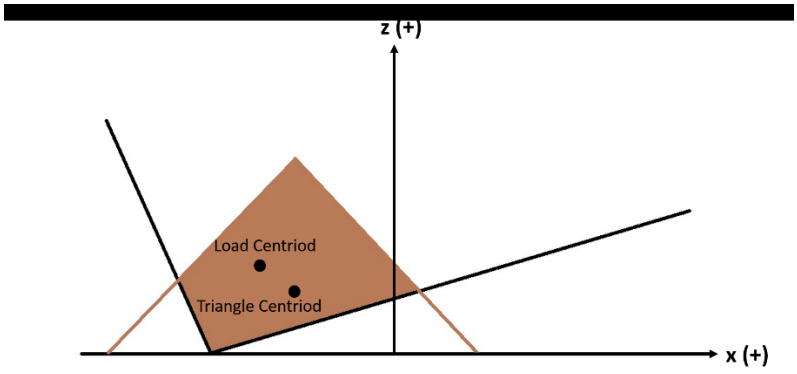


Figure 4-1 Centroid Deviation

4.2 Programming Language and Interface

The programming language used to develop the modelling software was C#. C# was chosen for ease of programming. WPF (Windows Presentation Foundation) was used to create the user interface for the software. WPF was chosen as it helps to simplify the development of Windows

Applications by providing many pre-set interface tools. An image of the interface as well as pseudocode for the software is available in Appendix III.

4.3 Modelling

Payloads were modelled by creating a mesh network over a grid area representing the truck body. Using these grid points a 3D mesh model was built of the truck body. An example model is shown in Figure 4-2.

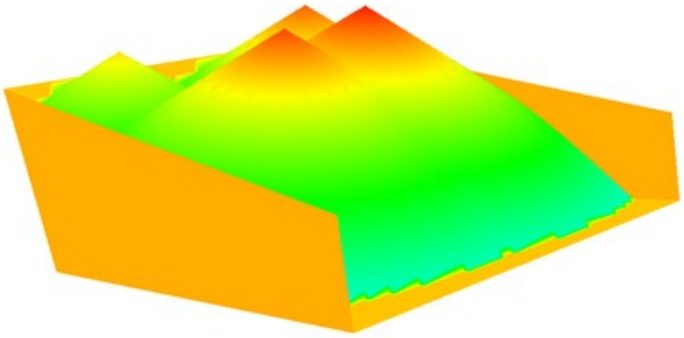


Figure 4-2 3D Model

4.4 Inputs

4.4.1 Material Properties

The following material properties were required as inputs to model the payload model:

ρ = Average Material Density (kg/m^3)

θ = Angle of Repose

For the purposes of the modelling software, the density was assumed to be constant throughout the material. This assumption is known to be invalid for many materials. However, density was used by the model only to estimate the volume placed. Problems may occur if the pre-set density is lower than the actual density of the material which would cause the volume estimate to be too

large. This could potentially cause an error, likely during later load passes, if the modelling software was unable to find any potential shapes that could fit such a volume within the remaining space. Overestimated densities could cause the load shape to be too small, but would not cause the same error. The ability to handle unique densities for each load pass could be included in future versions of the software.

Angle of repose is defined as the angle between the horizontal and the slope of a pile of granular material. The angle of repose is known to be constant for any given granular material. Any variance from the assumed angle of repose could cause the generated payload shape to be inaccurate. An overestimation would cause the shape to be too steep and compact, while an underestimation would cause the slope to be too small and the load to look too spread out.

4.4.2 Truck Dimensions

The following truck dimensions were required as inputs to model the payload:

d = Body Depth (mm)

l = Body Length (mm)

w = Body Width (mm)

β = Angle from horizontal to front body slope

α = Angle from horizontal to rear body slope

x_f = x direction distance from edge of body to front strut (mm)

y_f = y direction distance from edge of body to front strut (mm)

x_r = x direction distance from edge of body to rear strut (mm)

y_r = y direction distance from edge of body to rear strut (mm)

All of the listed dimensions are available from manufacturer specification sheets. A diagram corresponding to the listed dimensions in shown in Figure 4-3.

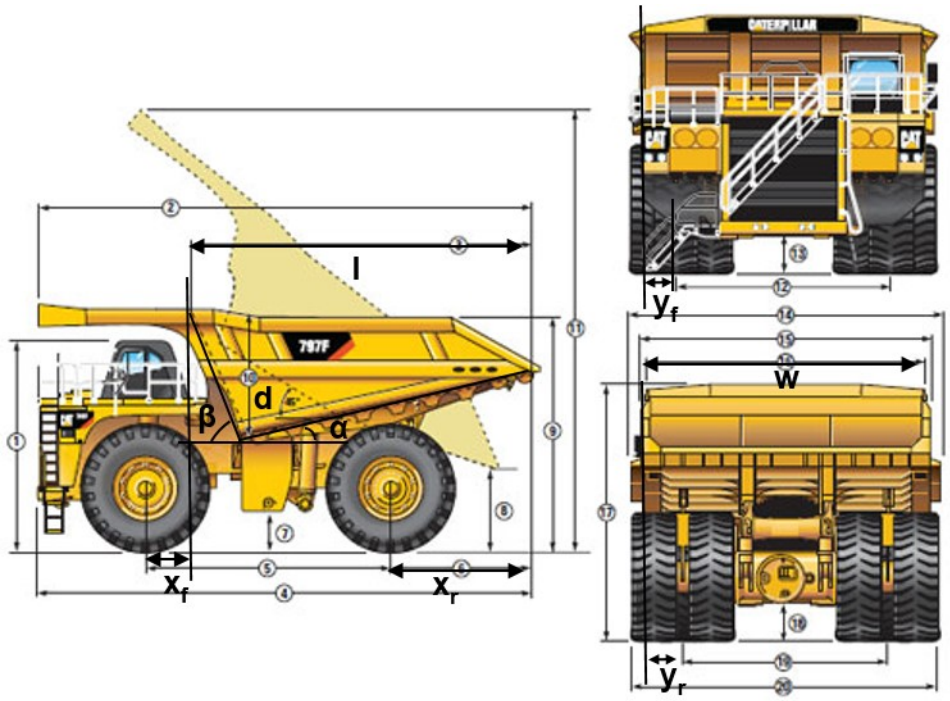


Figure 4-3 Truck Dimensions. Adapted from [34]

For the purposes of testing the modelling software, the dimensions of a Caterpillar 797B were used as listed by Caterpillar [35].

4.5 Determining the Centroid

Figure 4-4 shows a top down diagram of a truck body and strut locations. Struts will be abbreviated as:

FL = Front Left

FR = Front Right

RL = Rear Left

RR = Rear Right

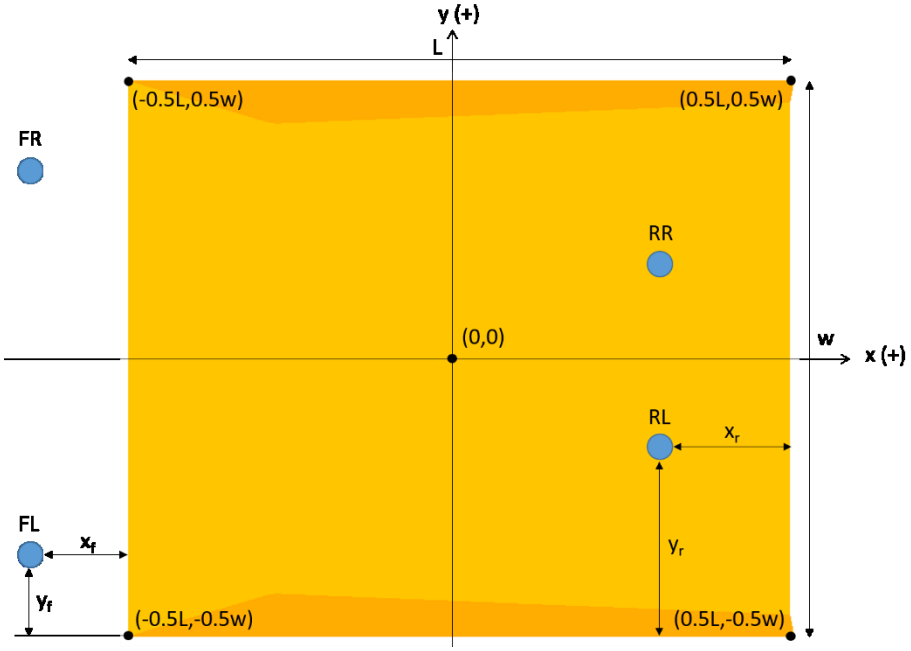


Figure 4-4 Approximate Strut Locations

Sensors installed in the suspensions cylinders output a pressure reading. This pressure can be converted to a weight given the area of the cylinder. To determine the total payload, the weights measured at each of the four struts is summed. The percent of the total load carried by an individual strut can be calculated as:

$$\%Loading = \frac{Strut\ Loading}{Total\ Payload} * 100\%$$

In addition, the split between front and rear or left and right can be determined by:

$$\%Front\ Loading = \frac{FL\ Load + FR\ Load}{Total\ Payload} * 100\%$$

$$\%Left\ Loading = \frac{FL\ Load + RL\ Load}{Total\ Payload} * 100\%$$

From these the centroid of the payload, using the coordinate system shown in Figure 4-4, can be determined as:

$$x_{centroid} = \frac{-(0.5L + x_f) * (FL\ Load + FR\ Load) + (0.5L - x_r) * (RL\ Load + RR\ Load)}{Total\ Payload} \quad (mm)$$

$y_{centroid}$

$$= \frac{-(0.5w - y_f) * FL\ Load - (0.5w - y_r) * RL\ Load + (0.5w - y_f) * FR\ Load + (0.5w - y_r) * RR\ Load}{Total\ Payload} \quad (mm)$$

This centroid provides a target point for the software model to match when searching for the most likely payload location.

4.6 Estimating Volume

The shape of a payload within a truck body is estimated by a series of intersecting cones. The volume of a cone intersected with a flat plane is defined as:

$$Volume = \frac{\pi r^2 h}{3} \quad (m^3)$$

Where r is the radius of the cone and h is defined as the height of the cone. Radius can also be determined based on a relationship between the height and the angle between the slope of the cone and the horizontal, such that:

$$r = \frac{h}{tan\theta} \quad (m)$$

Where θ is the angle of repose. Combining these two formulas results in a Volume function based only on height and a constant angle of repose. Therefore:

$$V = \frac{\pi h^3}{3tan^2\theta} \quad (m^3)$$

However, when the plane of intersection is no longer flat, such as in the case of a haul truck body, the calculation becomes much too complex to define simply. The volume above the truck body is shown in Figure 4-5.

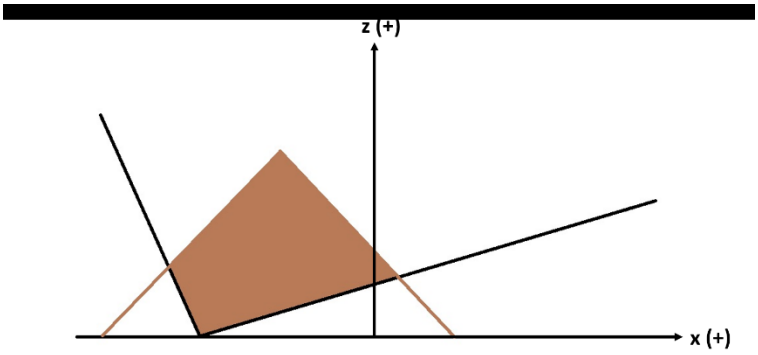


Figure 4-5 Volume of Material Above Truck Body

Instead, the volume can be estimated by summing incremental volumes. For a grid with a spacing of s this volume can then be described as:

$$V = s^2 * \sum_{i=1}^{n} h_i \quad (m^3)$$

Where n is the total number of grid points and h_i is the height above the truck body at point i. The height above the truck body can be determined by subtracting the z coordinate of the body from the z coordinate of the cone at all points where the cone is above the body.

This is most easily done by a computer algorithm which can progressively evaluate each point on the grid to determine if h_i will be above zero, then to calculate its value or assign a value of 0 to any points that would otherwise result in a negative value.

4.7 Payload Shape Algorithm

When determining payload shape based on strut pressures, the only known values are the location of the centroid and the expected volume. The height of the cone's peak relative to the truck body is unknown which poses a problem when modelling the shape of the payload.

At any given point on the truck body there is only one height at which the volume of a cone will result in the expected volume. This height can be determined through the use of a root finding algorithm.

Load shapes were generated as a cone of a height as determined by the root finding algorithm and at a given point centered at the peak. This cone was then intersected with the truck body to eliminate all parts of the cone that fell below the body.

4.7.1 Root Finding Algorithms

Root finding algorithms are designed to find a value of a variable such that a given function of that variable is equal to zero. There are many root finding algorithms available. Three methods were chosen for consideration: Newton's Method, the Bisection Method and Ridder's Method.

4.7.2 Newton's Method

Newton's method, as described by Stewart [36], relies on finding the tangent line at the point indicated by the first guess. The intersection of the tangent and the axis should result in a point closer to the solution. A diagram of the process is shown in Figure 4-6.

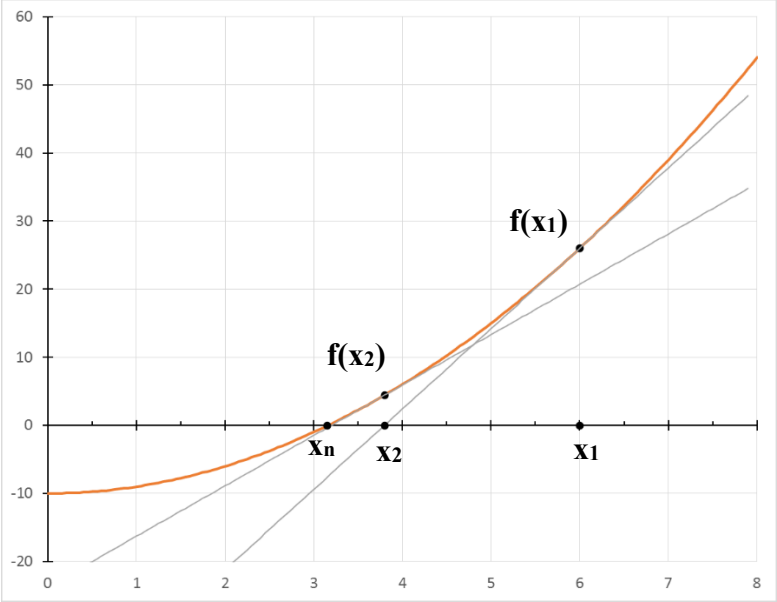


Figure 4-6 Newton's Method

In the diagram, x_1 refers to the first estimate, x_2 is the next estimate, and x_n is the final solution. Using this method, the next estimate can be calculated as:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Where, x_{n+1} is the next estimate, x_n is the current estimate, $f(x_n)$ is the value of the function at x_n and $f'(x_n)$ is the value of the derivative of the function f at x_n.

Newton's method is one of the fastest root finding methods, however it relies on the calculation of a derivative. As the function for calculating the volume of a cone in the modelling algorithm requires a logical statement, it is discontinuous and a derivative cannot be defined..

4.7.3 Bisection Method

The bisection method is a bracketed root finding algorithm. It requires an upper and lower bounds to be set. The bisection method is considered to be a robust and simple method, but is slower than most other root finding algorithms.

Based on the predefined boundaries the algorithms tests the midpoint, then determines a new boundary based on whether the result is above or below zero. The search bracket is halved with each iteration, eventually bringing the estimate closer to the solution until the predetermined error condition is met.

4.7.4 Ridder's Method

Ridder's method, as described by Ridder [37], as a "linearizing the original function whereafter the regula falsi is applied to this modified function" [37] is a modified form of the false position method. The first step is to choose an interval such that the lower and upper bounds are on opposing sides of the root. Figure 4-7 shows a diagram of this process.

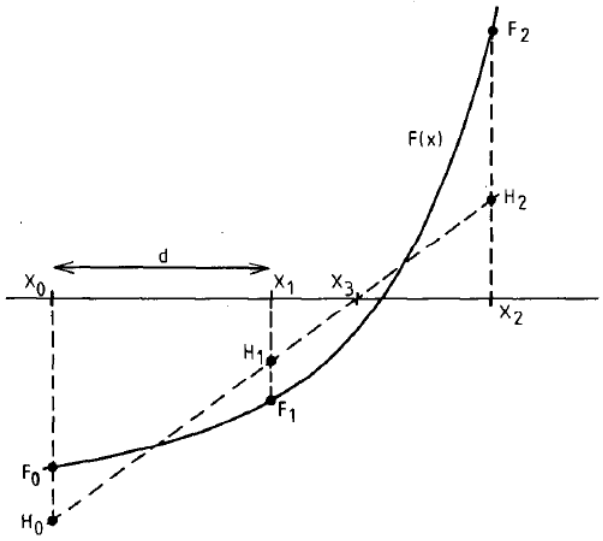


Figure 4-7 Ridder's Method [37]

Based on this diagram, x_3 is calculated as:

$$x_3 = x_1 + (x_1 - x_0) * \frac{F_1/F_0}{\sqrt{(F_1/F_0)^2 - F_2/F_0}}$$

x_3 is then used to define the next search interval. The sign of x_3 is compared to the sign of x_1, x_0 and x_2. The search interval is then set to be between x_3 and one of the other values such that each boundary is on opposite sides of the root. This process is repeated until the result is within a predetermined acceptable error range.

4.8 Evaluation of Root Finding Methods.

An evaluation was performed for the Bisection Method and Ridder's Method. Newton's method was not evaluated as a derivative could not be defined. Based on these evaluations a single root finding method was chosen for use in the payload shape determination algorithm.

4.8.1 Bisection Method

The bisection method requires a target result for a given function to be set as well as a pre-determined initial estimate, and upper and lower bounds on the adjusted variable. For the purposes of the modelling software, the target was set such that the difference between the volume of the modelled load and the expected volume was within $10mm^3$. The target function was then:

$$\Delta V(h) = s * \sum_{i=1}^{n} (h_i) - V_{Expected} = 0 \pm 10mm^3$$

Where h is the height of a simple cone measured from the base of the truck body. The height at which the above condition was met was considered to be the solution of the analysis.

The initial cone height estimate was calculated by determining the height of a cone on a flat plane at the height of the truck body at the x and y coordinates of the cone peak. The lower boundary (h2) was then set to the height of a cone on a flat plane located at the bottom of the truck body. The upper boundary was set to be 5 times the height of the lower boundary. This

value was selected based on tests run on the algorithm to ensure the highest possible speed while ensuring that the algorithm always found a solution. The results of this test are shown in Table 4-1.

Comparison of Iterations Required to Find a Solution for 21060 Loads					
Upper Boundary	**20h2**	**10h2**	**5h2**	**2.5h2**	**1.25h2**
Average Number of Iterations Required	36	35	27	42*	92*
Number of Indeterminate Shapes	0	0	0	3079	18,444
**indeterminate shapes are given a value of 100 iterations*					

Table 4-1 Average Iterations for Varying Upper Bounds Using the Bisection Method

Based on these results it can be seen that 5h2 was the best option as it always found a solution in the lowest number of iterations.

Once an estimated height had been tested and the volume determined to be above or below the expected volume, a new upper or lower boundary was set as the last tested height. The next estimate was then the midpoint between the new upper and lower bounds. This process was repeated until the difference between the two volumes was less than or equal to $10mm^3$. Figure 4-8 shows an example of how the search bracket was defined after each iteration.

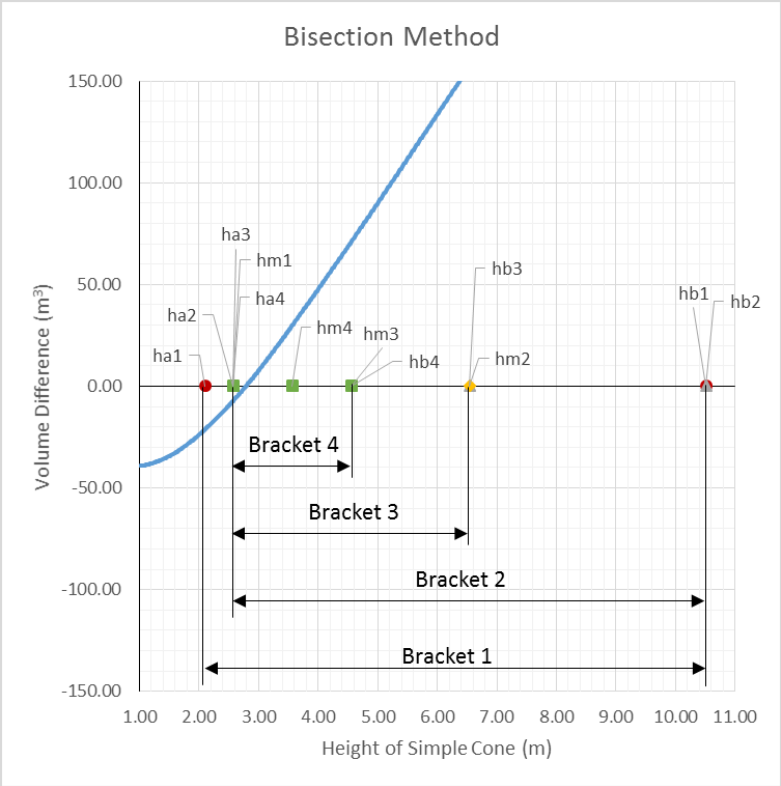


Figure 4-8 Example of Bisection Method

4.8.2 Application of Ridder's Method to Load Estimations

Like the bisection method, Ridder's requires a definition of a target and an upper and lower bound. The target and lower bound were set to the same value as in the bisection method. The same values were also considered for the upper bound. The results of the various upper boundaries are shown in Table 4-2.

Comparison of Iterations Required to Find a Solution for 21,060 Loads					
Upper Boundary	**20h2**	**10h2**	**5h2**	**2.5h2**	**1.25h2**
Average Number of Iterations Required	8	7	6	5*	29*
Number of Indeterminate Shapes	0	0	0	1	5,016
**indeterminate shapes are given a value of 100 iterations*					

Table 4-2 Average Iterations for Varying Upper Bounds Using Ridder's Method

Based on these results it can be seen that 5h2 was the best option as it always found a solution in the lowest number of iterations.

Once an estimated height had been tested and the volume determined to be above or below expected, a new upper or lower boundary was set. Ridder's method checks the signs of the last estimate, next estimate, and upper and lower bounds to determine how to set the new search bracket. If the sign of the last tested value and the next estimate are opposite, then the new bracket is between the two. Then it checks if the next estimate and the lower boundary are opposing signs. If so the new bracket is set to be between the old upper boundary and the next estimate. If not the new bracket is set to be between the old lower boundary and the new estimate. Figure 4-9 shows an example of how the search bracket is defined after each iteration.

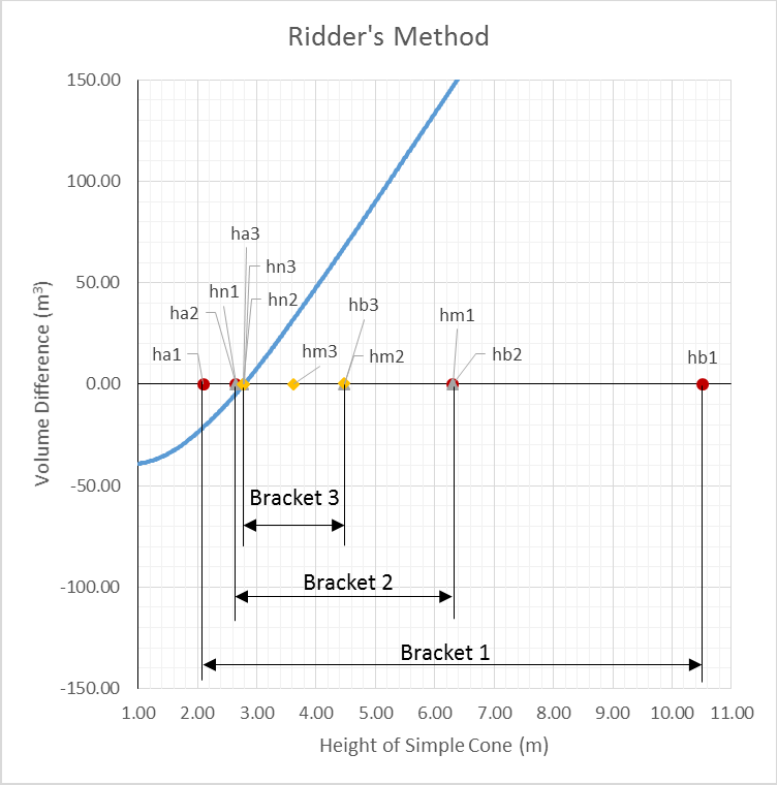


Figure 4-9 Example of Ridder's Method

4.9 Search Algorithm

While the centroid for a cone on a flat plane will be located at the x and y coordinate of the peak, this no longer holds true for a cone on a more complex surface. As such, a search algorithm was designed in order to determine the correct location to place the cone peak in order to minimize the distance between the calculated and measured centroids.

4.9.1 Acceptable Load Definition

An acceptable payload was defined as one that was found to meet the following conditions:

- The calculated volume was equal to the expected volume within $10mm^3$.
- The estimated payload shape must fit within the body geometry. No points on the edge of the body may be above the body sides.
- The calculated centroid of the payload is based on an acceptable cone shape, with lowest error (horizontal distance) from the measured centroid.

4.9.2 Assumptions

In order for the algorithm to function, it is assumed that there is only one shape at any given point on the box that meets the above requirements. It is also assumed that the angle of repose and material density remain constant.

4.9.3 Method

In order to determine the best solution, a search algorithm was developed. This algorithm first generates an array of potential loads. A load was generated with its peak located at each point on the grid using the payload shape algorithm. The centroid of this shape and its height from the base of the truck body were then calculated and saved in an array.

While the centroid was being calculated, the validity of the potential load shape was also evaluated. This was done by checking that all points on the load were at or below the sides of the body along the edges of the grid. If a point was detected to be above the allowable geometry the centroid was set to be well outside the limits of the grid such that its distance from the measured centroid would be much higher than that of any valid shapes.

Once an array of potential load shapes had been generated the algorithm then searched through every potential load shape and determined which had the minimum distance from the measured centroid. Initially, an arbitrarily large distance was initially set as the minimum distance. This distance was large enough that invalid shapes would be ignored while still evaluating valid ones. As the algorithm searched through the points it would replace the minimum distance with any value that was less than the previous minimum, thereby setting a new minimum. This process was repeated until all points were evaluated, at which point the key values required to recreate the best fit load were output to the modelling portion of the program. These key values included the height of the cone relative to the base of the truck body, as well as the x and y coordinates of the peak. This process is shown in Figure 4-10.

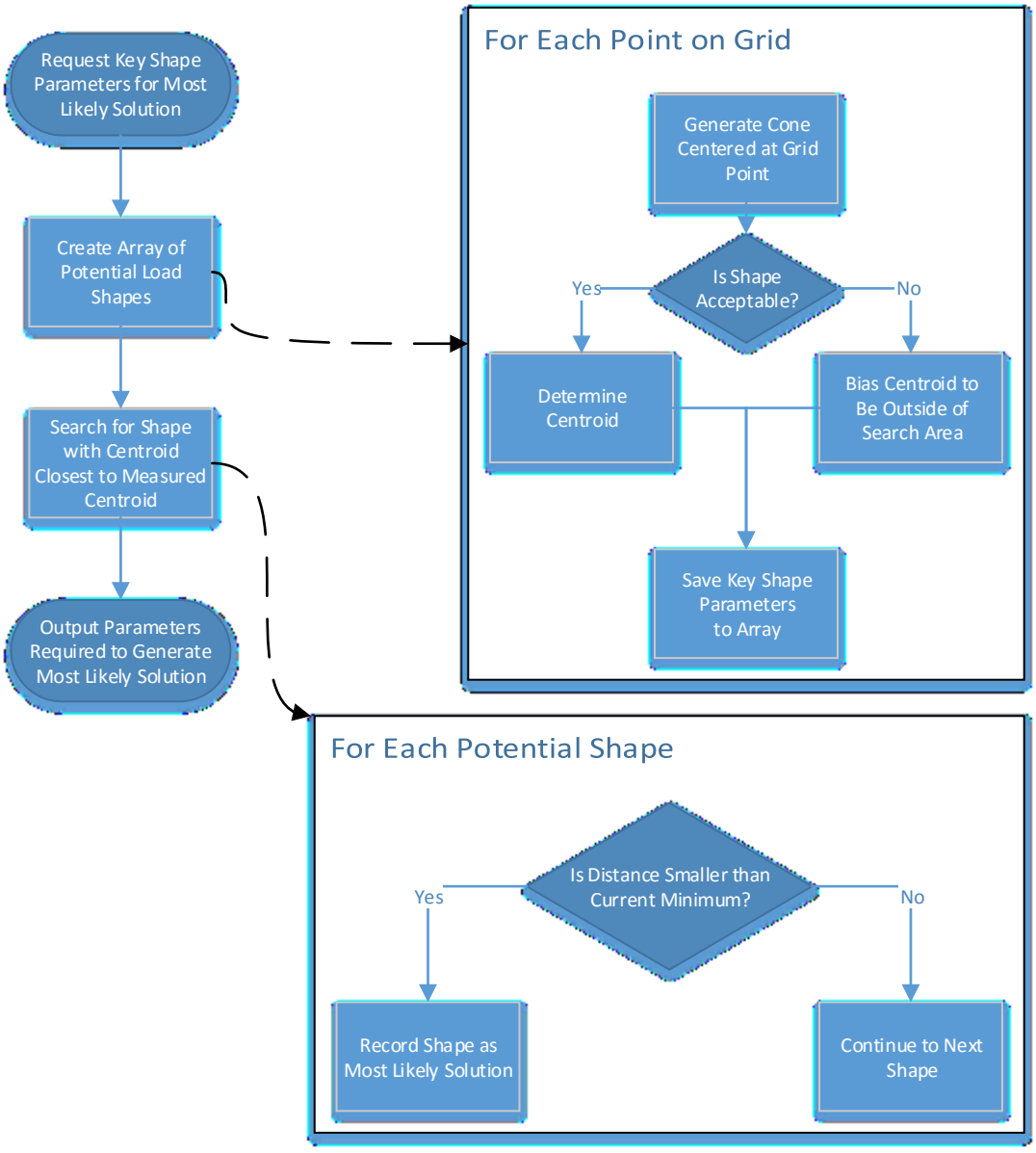


Figure 4-10 Search Algorithm

4.10 Optimizing the Search Algorithm

While the previously described search algorithm is very effective at finding the best possible location of the cone to meet the predefined constraints, it takes a long time to do so. In order for the program to be useful as a real-time solution it must be able to generate results between when a shovel operator places a load and before they need to place the next (approximately 25

seconds. As such, various methods were considered in order to speed up the search while still finding the best solution.

4.10.1 Search Rectangle

The payload peak is unlikely to be on the opposite end of the truck body as the measured centroid. Because of this the search area could likely be limited. As the search area for the full search algorithm was essentially a rectangle the size of the truck body, it was proposed that this rectangle could be scaled down and centered on the measured centroid. Doing so would considerably reduce the processing time as it would require less potential loads to be generated before being evaluated.

The concern with this method is that if the search area is set to be too small, it will miss the best result and provide a less accurate one instead. On the other hand, if the search area is too large there will be little to no benefit in terms of processing speed.

4.10.2 Progressively Narrowing Search

Another method considered, was to begin with a rough search of the truck body. The results of this search would be checked to determine which was closest to the measured centroid. Next a second rough search would be performed over the section of the box represented by the first result. The results of this search would then be evaluated in the same way as the first rough search, in order to select a final search area. A full search would be performed over this area to determine the best solution.

This method was implemented as follows:

- Phase 1 – Calculate twelve points spaced evenly through body and determine the closest point.

- Phase 2 – Calculate four points spaced evenly through the section represented by the previously selected point and determine the new closest point and section.
- Phase 3 – Perform a full search of the section selected in Phase 2 and determine the closest payload shape.

This process is shown visually in Figure 4-11.

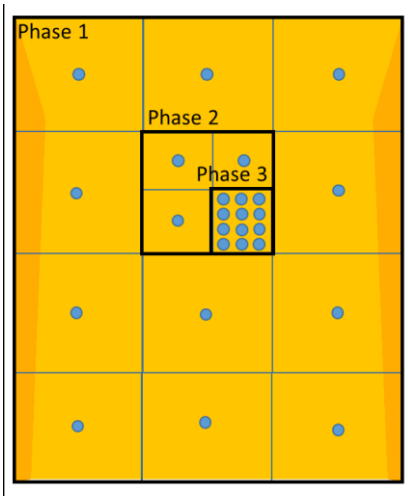


Figure 4-11 Progressively Narrowing Search

This system allowed the whole body to be scanned at a much faster rate than the simple full search. If the search area is defined correctly it should be able to provide a similar accuracy to the full search.

4.10.3 Comparison of Search Methods

The three different search methods were compared for both the Bisection and Ridder's methods. Results were generated on a computer system with the following hardware installed:

- Processor: Intel Core i5-4590 Processor which runs at 3.30GHz
- RAM: 2x8GB DDR3 Dual Channel
- Video Card: NVIDIA GeForce GTX 970

After running tests for each of the methods on the same sample data, the results shown in Table 4-3 were obtained.

Processing Time (s)		Full Search	Search Rectangle	Narrowing Search
Load Pass 1	*Bisection Method*	704	112	21
	Ridder's Method	502	81	15
Load Pass 2	*Bisection Method*	790	121	21
	Ridder's Method	475	87	14
Load Pass 3	*Bisection Method*	792	122	21
	Ridder's Method	454	83	13
Load Pass 4	*Bisection Method*	710	94	20
	Ridder's Method	378	58	12
Average	*Bisection Method*	749	112	21
	Ridder's Method	452	77	14

Table 4-3 Processing Speed Comparison (all times in seconds)

Based on these results, it is clear that using the Progressively Narrowing Search and Ridder's Method provide the fastest results.

The payload shapes were also compared, based on the assumption that the full search provided the best possible result. These shapes are shown in Figure 4-12.

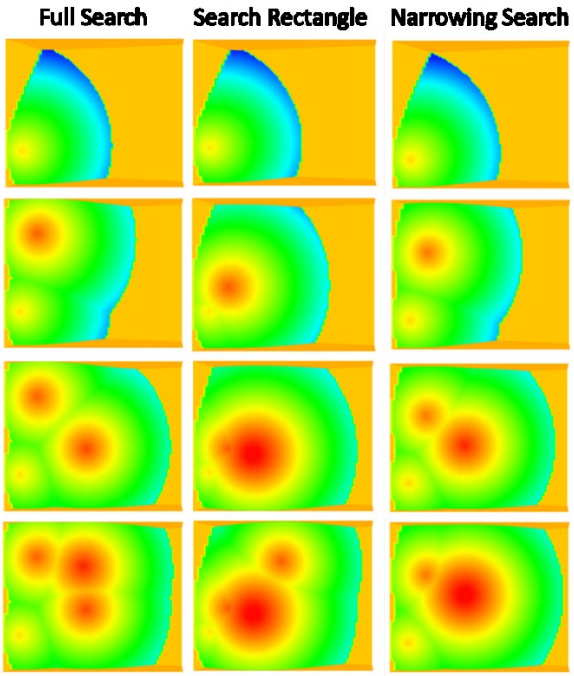


Figure 4-12 Payload Shape using Different Search Methods

Based on these generated shapes it is clear that while the other search methods are significantly faster, they do not always generate the same result as the full search. The narrowing search appears to have gotten closer to the full search results but there was still fairly significant deviation.

4.11 Real Time Considerations

In order for the modelling software to be utilized in the field, the ability to monitor live, real-time data must be implemented. This could be done by adding algorithms to monitor the data stream and sample readings when required. This process is described further in Section 0.

In addition, the full search algorithm, which provides the most reliably accurate results runs much too slow, other search methods would need to be used. Recommendations on how to improve this algorithm are available in Section 8.1.

5 Lab Tests

5.1 Purpose of Tests

The purpose of these lab tests were to create data sets based on scaled to actual loads that could be used to verify the modelling software. Images of the loads were taken in order to allow visual comparison between the software model and the test results.

5.2 Scaling Methodology

A 1:25 scale model of a 797B haul truck box was built for prior research by Chamanara [6]. As described in detail by Chamanara [6], this model was built from welded sheets of 3mm steel with screws welded to locations corresponding to a haul truck's struts. Linear dimensions were determined by scaling down directly scaled directly. Volume and weight capacities were scaled down using a cube root approach based on the reported capacities on the Caterpillar 797B spec sheet [38]. In order to provide an attachment point for the front struts, a metal plate was welded to the front.

The box has a theoretical volumetric heaped capacity of 14,080cm^3, which was scaled from the equivalent 797B capacity of 220m^3. Given this capacity, loading in four passes would require a scoop of 3,520cm^3 per load, scaled from the equivalent 797B capacity of 55m^3. The scaled weight capacity was calculated to be 23.04kg from the equivalent capacity of 360 metric tonnes. The scale model is shown in Figure 5-1.

Figure 5-1 Scale Model

Attached to the box, at each of the four strut locations**,** was an Artech 20210- 50lb S-Shaped load cell. These load cells are made of nickel/chrome plated alloy steel and can handle a maximum load of 50lb each. Data was output as a voltage. [39]

Each load cell had a unique calibration constant used to convert the voltage output and convert it to imperial pounds through the data acquisition system.

A one gallon plastic ice cream pail was used to represent a shovel bucket, of a size that would approximately load the box in four passes. The approximate volume of this pail was 3,480cm^3.

5.3 Data Acquisition System

A HBM MGCPlus was used to read the load cells and create spreadsheets of loading data for each test. While this hardware is capable of much higher measurement frequencies, a frequency of 2Hz or 2 readings per second was used for the tests. This frequency provides a reasonable number of samples, while keeping processing speed for the modelling software high. The MGCPlus runs using HBM's proprietary Catman software [40].

The load cells calibration was confirmed against a known weight. This software also accepted the load cell conversion parameters required to interpret the voltage data, which converted the output to imperial pounds. These loading were then converted to kilograms for use with the modelling software. Figure 5-2 shows an image of the data acquisition hardware (DAQ).

Figure 5-2 Data Acquisition Hardware

5.4 Material

Two different materials were used for the lab tests, sand and crushed limestone. The sand was found to have an average loose density of 1,647kg/m^3 and an angle of repose of 32.2^o. The second material was a crushed limestone, which represents scaled blasted rock material. The crushed rock was found to have an average loose density of 1,519kg/m^3 and an angle of repose of 37.4^o.

The angle of repose was calculated by dumping a load of material in a pile on a flat surface. The width and height of the piles were then measured in order to calculate the angle.

5.5 Procedure

Once the DAQ was set to begin recording, loads were placed into the box in pre-determined patterns of three or four sequential loads. Three different patterns were used. Patterns were chosen in order to provide a variety of payload distributions for use in verifying the modelling software.

Load Pattern A is illustrated in Figure 5-3. This pattern involved placing loads subsequently into the Front Left, Front Right, Rear Left, and then Rear Right quadrants. Pattern A was chosen in order to generate payloads that were biased toward the rear.

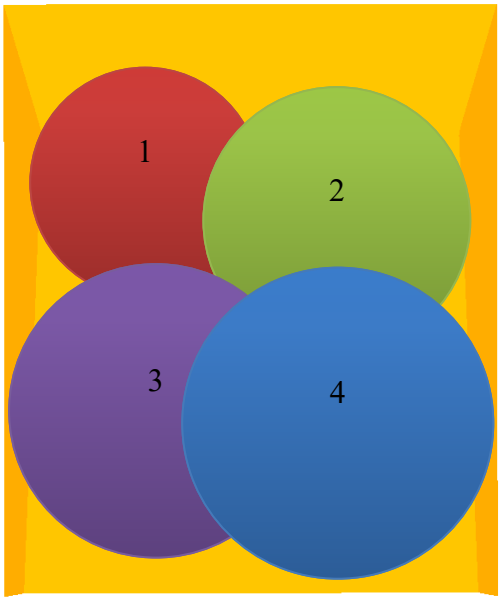


Figure 5-3 Load Pattern A

Load Pattern B is shown in Figure 5-4. This pattern required loads to be placed in the Front Left, Rear Left, Front Right, and then Rear right quadrants. Pattern B was chosen in order to generate payloads that were biased to the right.

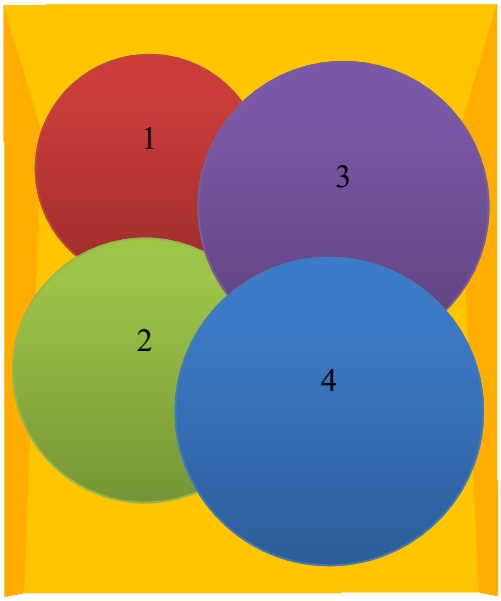


Figure 5-4 Load Pattern B

Load Pattern C, illustrated in Figure 5-5, was performed by placing 3 loads in the center of the box. Only three loads were placed in this pattern as any further loads would have overflowed the box. Pattern C was chosen in order to verify that the software functioned correctly when subsequent load peaks were placed in the same location, causing the new load to overlap the previous one.

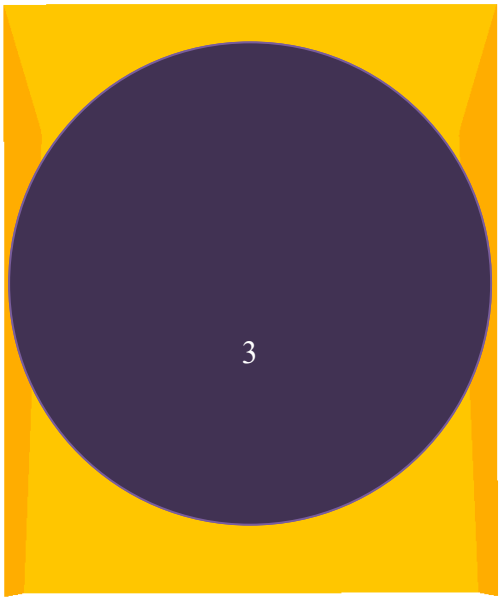


Figure 5-5 Load Pattern C

Tests run with sand used all three patterns, but only Pattern A and Pattern B were used with the crushed rock material. Pattern C was also run for the crushed rock but the data was corrupt due to an experiment error. This test was not repeated as Pattern C was found to have minimal use for verification after testing the Sand Pattern C results in the software.

Tests were labelled by combining a letter to indicate the material used and the pattern letter. For example, "C-A" corresponds to a test using the crushed rock material with Pattern A. A complete list of the tests, patterns and material used can be found in Table 5-1.

Test	Material	Pattern
C-A	Crushed Rock	A
C-B	Crushed Rock	B
S-A	Sand	A
S-B	Sand	B
S-C	Sand	C

Table 5-1 Tests

After each payload had been placed, they were poured into a bucket from which a depth measurement was taken. Using this measurement, as well as the dimensions of the bucket, a volume was calculated. These measured volumes were compared with the volumes estimated using only the mass readings and previously calculated densities, in Table 5-2.

Test	Depth in Bucket (cm)	Mass (kg)	V Measured (cm3)	V Estimated (cm3)	Error (cm3)
C-A	21.27	18.05	11,277	11,887	610
C-B	22.86	18.10	12,186	11,919	(267)
S-A	22.23	18.27	11,821	11,093	(728)
S-B	22.86	18.63	12,186	11,315	(871)
S-C	19.05	16.11	10,019	9,779	(240)

Table 5-2 Lab Test Payload Volumes (Not Scaled)

The deviation between measured and estimated volume is likely in part due to a measurement error. This was likely due to spilled material resulting from overflow during loading. This spillage was included in the volume measurement, but not in the mass measurement from the load cells.

5.6 Results

Table 5-3 shows a summary of the final payload distribution for each of the five tests. These results were calculated based on strut readings for the final payload with the initial loading subtracted in order to isolate the payload.

Test	FL	FR	RL	RR	Left	Right	Front	Rear
C-A	20.4%	15.5%	32.2%	31.9%	52.7%	47.3%	35.9%	64.1%
C-B	18.3%	19.7%	31.2%	30.8%	49.5%	50.5%	37.9%	62.1%
S-A	17.3%	22.0%	27.8%	32.9%	45.2%	54.8%	39.3%	60.7%
S-B	18.6%	21.6%	19.4%	40.4%	38.0%	62.0%	40.2%	59.8%
S-C	21.5%	19.9%	32.1%	26.4%	53.6%	46.4%	41.4%	58.6%

Table 5-3 Payload Distribution

A detailed summary of the results of Pattern B loading for both the crushed rock material and the sand are presented below with additional test results available in Appendix I.

5.6.1 Crush Pattern B Test

This test resulted in a scaled total load of 282 metric tonnes. As shown in Table 5-3, the payload was fairly balanced between left and right, with only a 0.5% overload on the right. The front portion of the box held 37.9% of the load, while the rear held 62.1%. Figure 5-6 shows the strut loading as each of the four loads were placed. An image of the final payload can be seen in Figure 5-7.

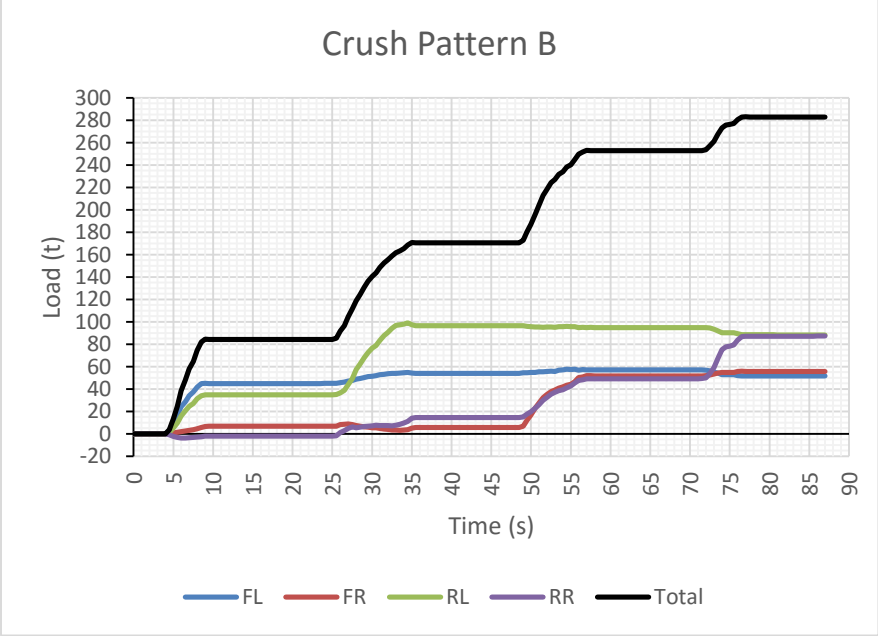


Figure 5-6 Crush Pattern B Results

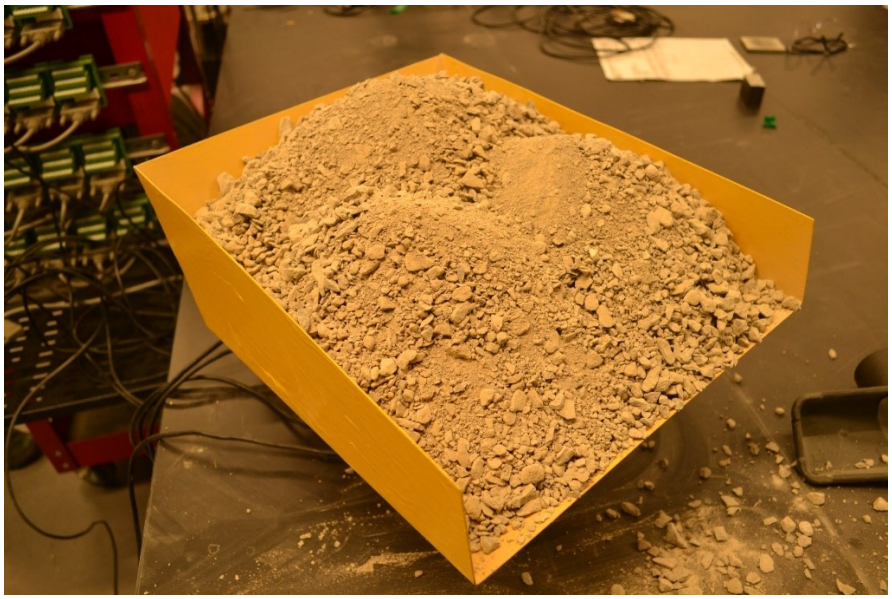


Figure 5-7 Crush Pattern B Final Payload

5.6.2 Sand Pattern B Test

This test resulted in a total load of 291 metric tonnes, as shown in Figure 5-8.the payload was overloaded on the right side, with 62% of the load held by the right struts. The front portion of the box held 40.2% of the load, while the rear held 59.8%. Figure 5-8 shows the strut loading as each of the four loads were placed. From this information it can be seen that the RR strut was already overloaded after the third load was placed, with the fourth load serving to increase the overload. An image of the final payload can be seen in Figure 5-9.

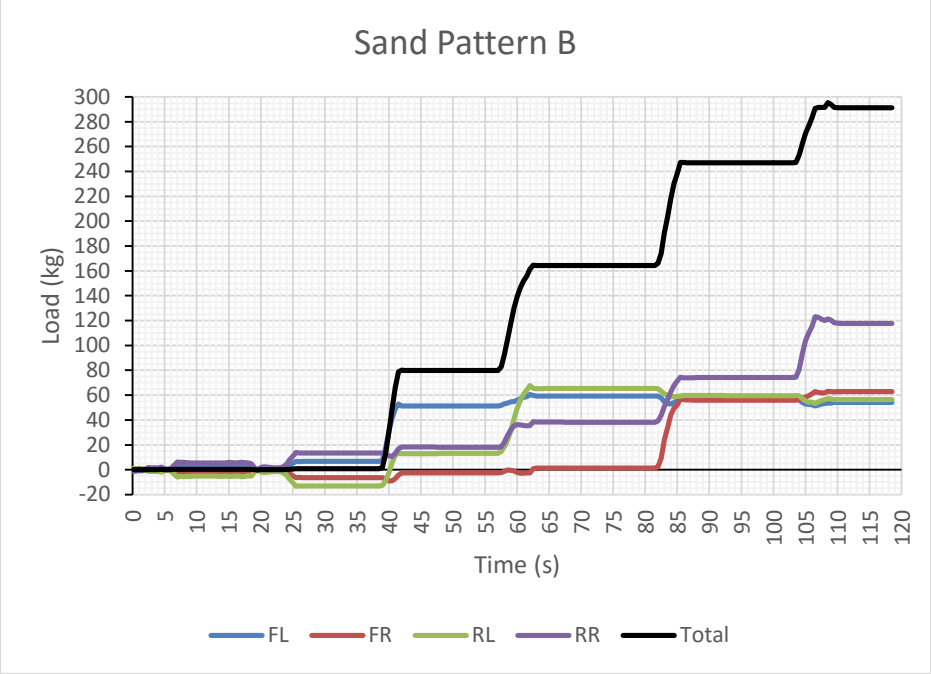


Figure 5-8 Sand Pattern B Results

Figure 5-9 Sand Pattern B Final Payload

5.7 Application of Model

The load cell data collected from the lab tests was used to test the modelling software after being scaled to be equivalent to a Cat 797B haul truck, as described in Section 5.

5.7.1 Results

As the modelling software is designed to match the expected volume in order to generate a payload shape, all calculated volumes were equal to the expected volumes. This is shown in Table 5-4.

	S-A		S-B		S-C		C-A		C-B	
Volume Comparison	*Expected Volume*	*Calculated Volume*	*Expected Volume*	*Calculated Volume*	*Expected Volume*	*Calculated Volume*	*Expected Volume*	*Calculated Volume*	*Expected Volume*	*Calculated Volume*
Pass 1	51.16	51.16	48.57	48.57	51.73	51.73	54.96	54.96	55.53	55.53
Pass 2	54.85	54.85	51.13	51.13	53.91	53.91	55.36	55.36	56.70	56.70
Pass 3	50.16	50.16	50.28	50.28	47.15	47.15	45.96	45.96	54.20	54.20
Pass 4	17.14	17.14	26.75	26.75			29.40	29.40	19.76	19.76
Total	***173.31***	***173.31***	***176.73***	***176.73***	***152.79***	***152.79***	***185.68***	***185.68***	***186.19***	***186.19***

Table 5-4 Expected vs Calculated Volumes (all volumes measured in m^3)

The distance between the modelled load and measured centroids was calculated and summarized in Table 5-5.

Distance from Measured Load Centroid				
Test	***Pass 1***	***Pass 2***	***Pass 3***	***Pass 4***
S-A	0.90	0.96	0.09	0.40
S-B	0.67	0.05	1.41	0.36
S-C	0.04	0.08	0.98	
C-A	0.80	0.60	0.09	0.08
C-B	0.74	0.02	1.08	0.14
Average	***0.50***			

Table 5-5 Distance from Measured Centroid (all distances measured in m)

In addition, the distance between the modelled and measured overall payload centroid was calculated after each pass. These results are shown in Table 5-6.

Distance from Measured Payload Centroid				
Test	***Pass 1***	***Pass 2***	***Pass 3***	***Pass 4***
S-A	0.90	0.93	0.62	0.55
S-B	0.67	0.32	0.66	0.62
S-C	0.04	0.04	0.33	
C-A	0.80	0.68	0.47	0.40
C-B	0.74	0.37	0.60	0.53
Average	***0.54***			

Table 5-6 Distance from Measured Payload Centroid (all distances measured in m)

5.7.2 Sand Pattern B Detailed Results

These results were examined in further detail for test S-B as an example. Results for the other lab tests are available in Appendix I.

Centroid locations for both measured and calculated load pass centroids were plotted in Figure 5-10.

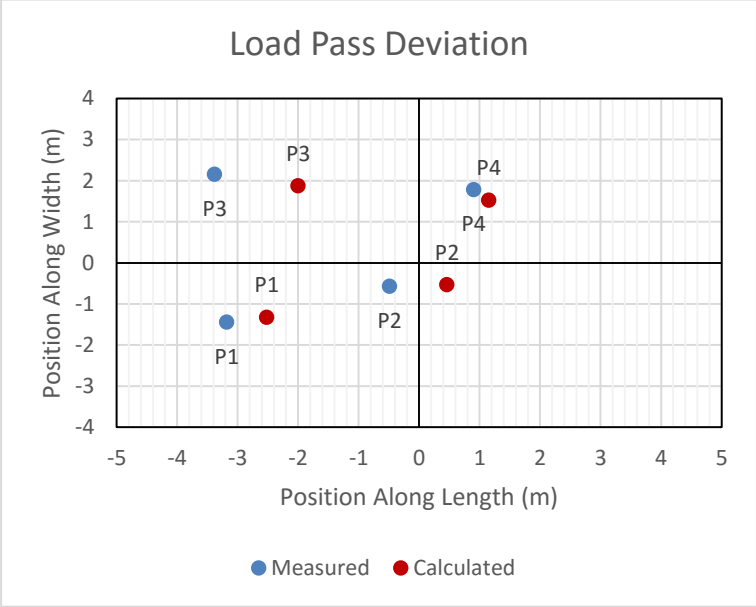


Figure 5-10 Load Pass Centroid Deviation

The overall centroid location after each pass, for both measured and calculated, were plotted in Figure 5-11.

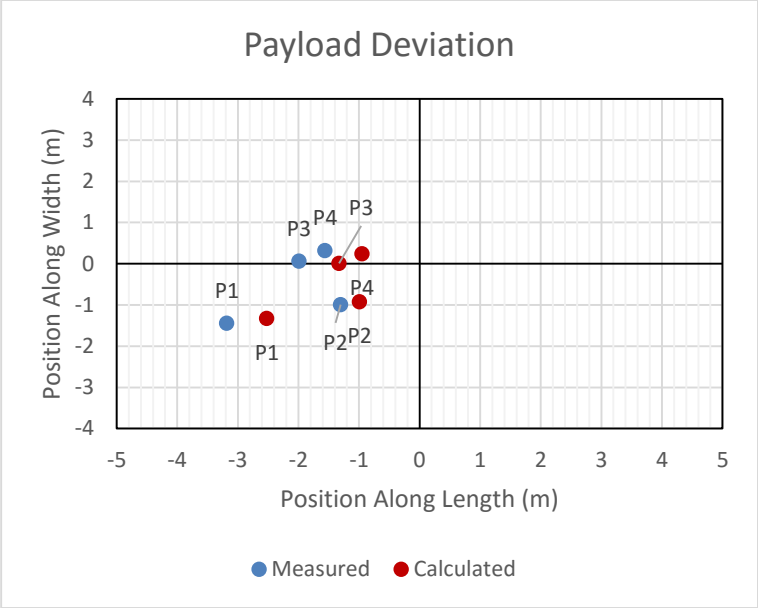


Figure 5-11 Overall Payload Centroid Deviation

Figure 5-12 shows a visual comparison between the actual lab test payloads and the payloads from the modelling program.

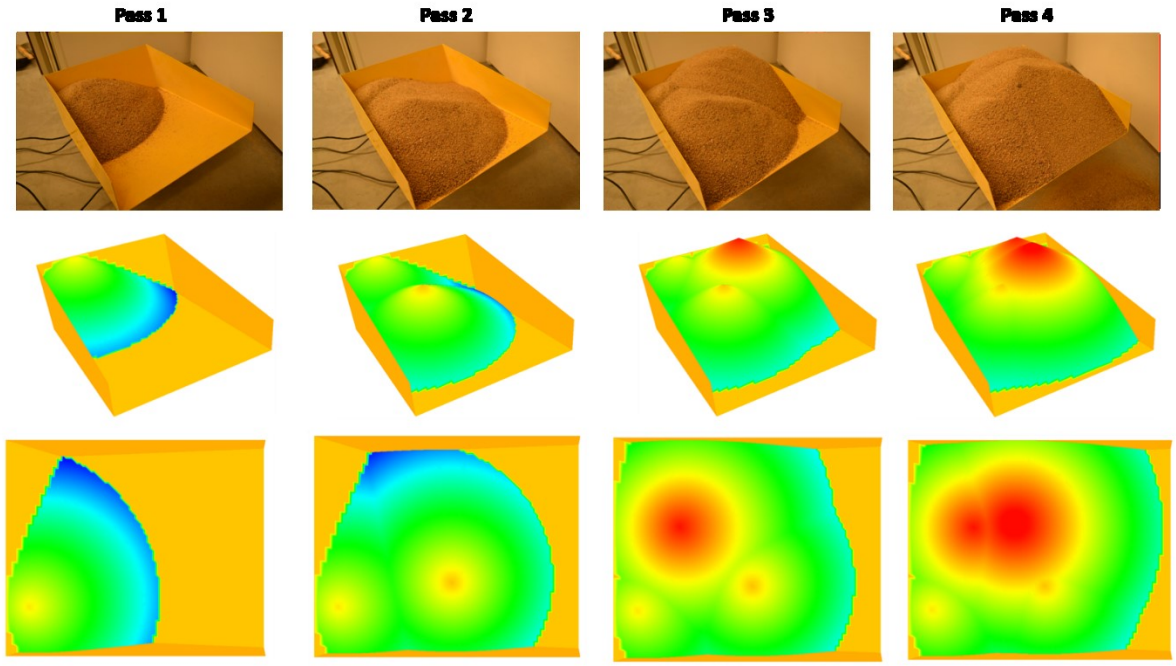


Figure 5-12 Pass by Pass Comparison of Modelled to Actual Payload Shape

As shown, the modelling software provided a reasonable estimation of the actual payload distribution. There are, however, a few areas of concern.

Passes 1 and 3 show the modelled payload having a clearly defined cone, while the actual payload was more spread out. This was likely due to the load being spread out more than intended during testing. If placed by an excavator the boom can be held in a constant position while dumping which would reduce this effect. Unfortunately, from strut pressures alone, there is no way to detect such a behavior of the load.

While a load can form a sharp peak as shown by Pass 4, it appears to be more likely that the peak is more smoothed out than modelled by the software. This could be fixed by altering the modelling algorithms to smooth out the cone near the peak.

It is also very important that the material density and angle of repose be as accurate as possible as these inputs have a major impact on the definition of the payload shape. Any inaccuracies in the measurement of these values will directly affect the accuracy of the modelled payload. While varying density could be an issue contributing to inaccuracies, it is unlikely for the materials used in the lab tests. The sand showed a uniform density and particle size distribution. The limestone crush showed a less uniform distribution but did not appear to deviate significantly.

6 Field Data

Strut pressure data was collected for one loading cycle of a Cat 785C haul truck. Pressures were converted to kilograms by multiplying the pressure by the area of a suspension cylinder. The raw data was then analyzed and readings selected to represent each of the eight load passes. Figure 6-1 shows the points at which readings were taken for each of the six passes.

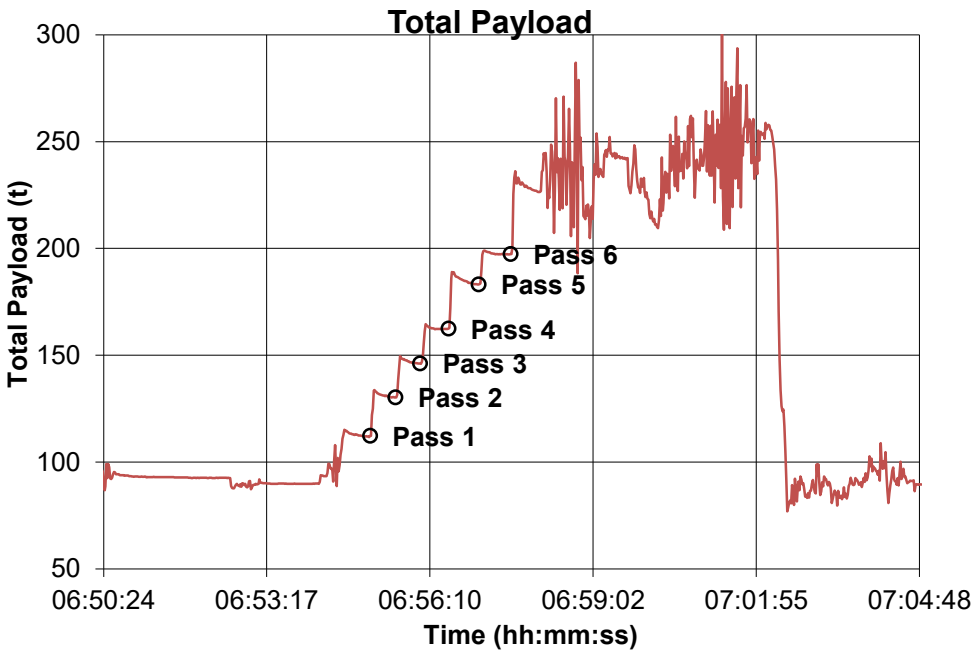


Figure 6-1 Total Field Data Reading Selection

Readings were selected at points which best represented the payload. This was done by selecting points after the truck had stopped or slowed its rocking motion resulting from load placement. A material density was 1972kg/m^3 and angle of repose was 26.25º. Using these parameters and the Cat 785C dimensions the following results were obtained.

6.1 Application of Model

The modelling software successfully found the payload shape for all six load passes. Centroid locations for both measured and calculated load pass centroids were plotted in Figure 5-10. Distances from the measured overall payload centroid were plotted in Figure 5-11.

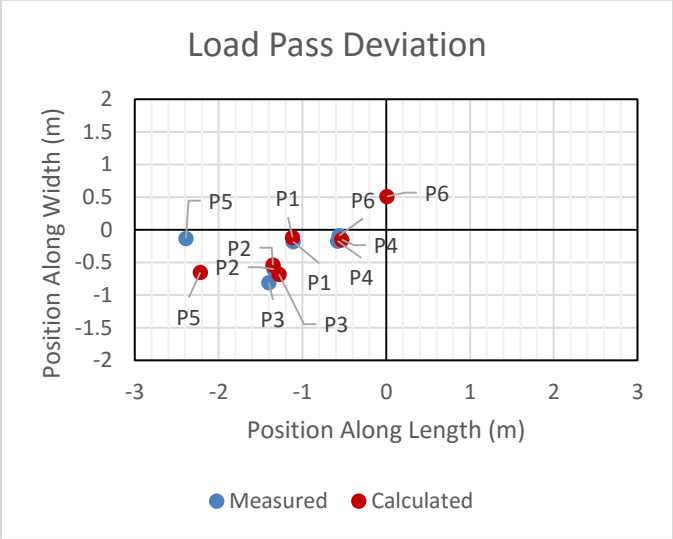


Figure 6-2 S1324 Load Pass Centroid Deviation

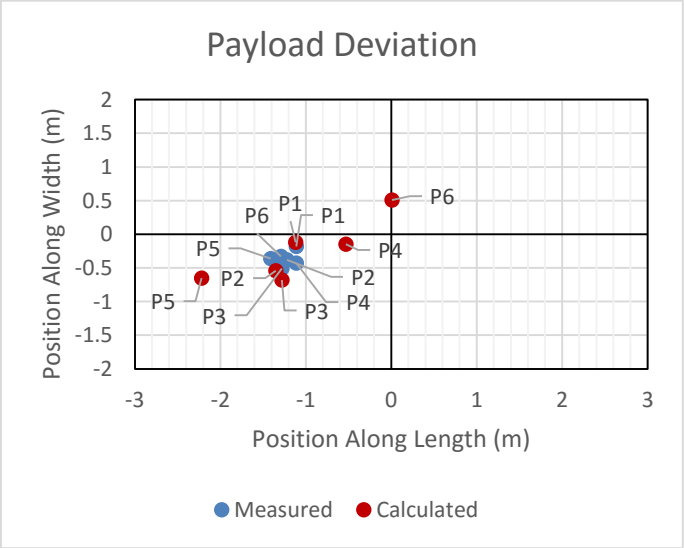


Figure 6-3 Overall Payload Centroid Deviation

The distance between the modelled load centroid and measured centroid was calculated and summarized in Table 5-5.

Distance from Measured Load Centroid					
Pass 1	***Pass 2***	***Pass 3***	***Pass 4***	***Pass 5***	***Pass 6***
0.06	0.06	0.18	0.05	0.80	0.83

Table 6-1 Distance from Measured Centroid (all distances measured in m)

In addition, the distance between the modelled and measured overall payload centroid was calculated after each pass. These results are shown in Table 5-6.

Distance from Measured Payload Centroid					
Pass 1	***Pass 2***	***Pass 3***	***Pass 4***	***Pass 5***	***Pass 6***
0.06	0.05	0.09	0.07	0.24	0.31

Table 6-2 Distance from Measured Payload Centroid (all distances measured in m)

Figure 6-4 shows a plot of the distance between the calculated and measured centroids for each individual load and the overall payload.

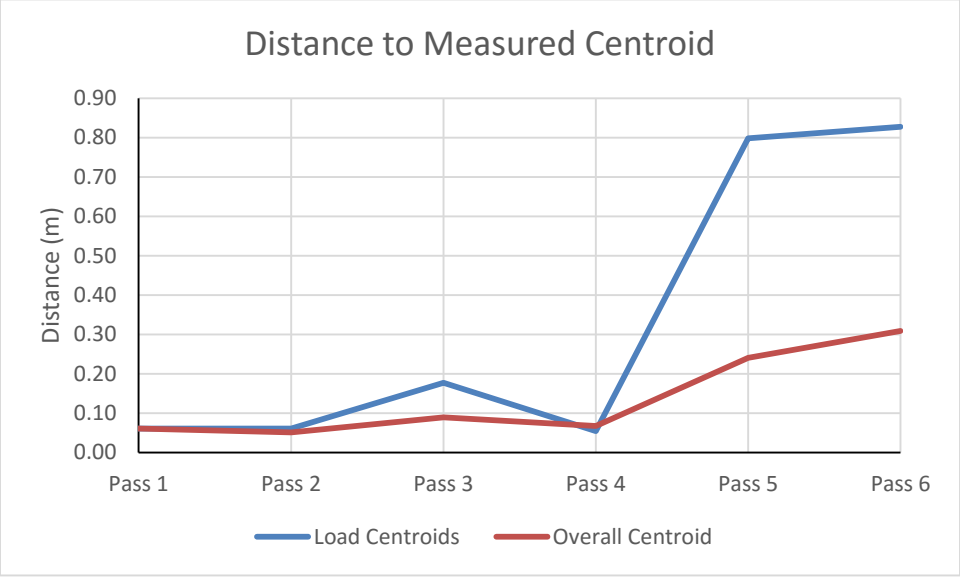


Figure 6-4 Distance to Measured Centroid

The distance remained below 0.2m for the first four passes, after which the deviation from the measured centroids increased significantly. Pass 5 shows the greatest increase in error, which is determined by the distance from the measured centroid. The most likely reason for this is that the assumption that material density remains constant throughout the material was incorrect. If this particular pass had a different density it would change the shape and resultant centroid. However, in order to determine the exact cause, further information would be required. Figure 6-5 shows the modelled surface after each pass.

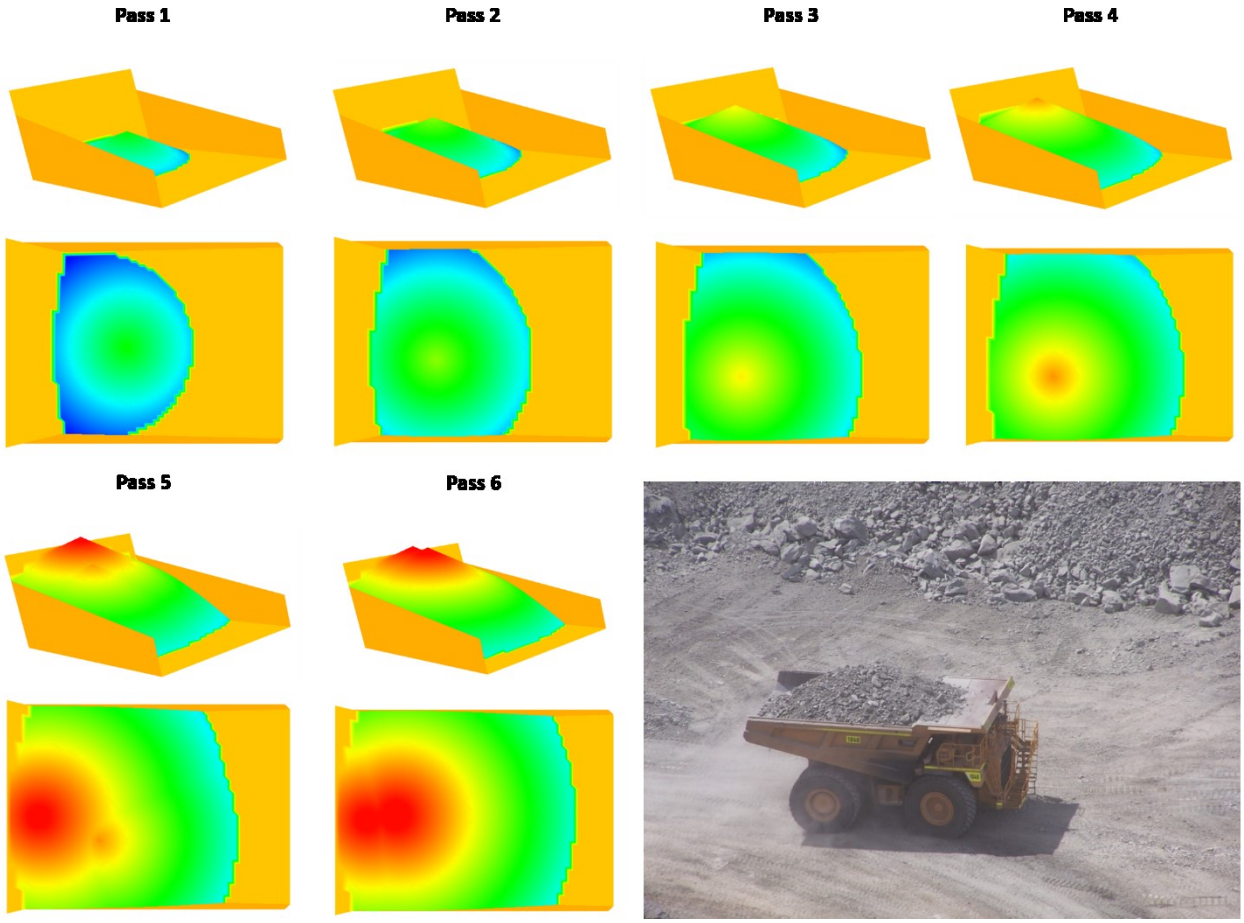


Figure 6-5 Pass by Pass Model of Payload

Figure 6-5 indicates that while the model followed the shape of the payload, as shown in the photo, it did not fill the truck body as completely. It was hypothesized that this issue may have been due to a variable material density, and is further discussed in Section 8.3. Actual truck loading pass by pass images were unavailable for this data set, so further analysis of the shape accuracy was not possible.

7 Conclusions

Unbalanced haul truck payloads can cause increased frequency and intensity of rack, pitch and roll events. These events, in turn, can cause excess equipment wear, running surface deterioration and operator health issues. Excavator operators play a key role in balancing truck payloads. These operators have minimal aids to determine if a payload is balanced, instead relying on visual inspection alone. Operator visibility is often limited by the excavator structure. It is important that these operators be given additional aids so that they can better judge the balance of payloads. Such information must be reasonably accurate and provided within one cycle of the excavator.

In order to provide excavator operators with additional information on truck payload balance, modelling software was developed using truck strut pressures. This software determined the location of the load centroid and estimated the volume of the load. It assumed that granular material forms a conical shape at a constant angle of repose and the material density remained constant throughout the payload. Ridder's method of root finding was used to determine the shape of the load such that the volume was equal to the initial estimate. A search algorithm was used to generate a potential load for all points on a grid and calculate their deviation from the measured centroid. The minimum deviation was then determined and the corresponding load generated on the visual model.

While this method was effective at finding a solution, it took well over one excavator cycle to do so. For this reason other search methods were considered. Suggested options included a search rectangle centered on the measured centroid and a progressively narrowing search that scanned the entire truck body at a much faster rate. Of the methods only the progressively narrowing search was able to meet the speed requirements, but it was unable to correctly match the full

search results on every load. It is suggested that this method be researched further in order to improve its accuracy.

Strut data was then collected from lab tests and modelled with the software. The modelled payload appeared to be similar to the actual payload for most of the tests. Deviations from the actual load shape occurred when the load had not been placed from a constant point resulting in a more spread out shape. This error is unavoidable using only strut data as there is no way to determine that the load had been spread out.

Field data was also modelled with the software for one loading cycle. The modelling software successfully modelled all six passes with what appeared to be reasonable accuracy. However, the model centroid began to deviate from the measured centroid for the last two loads. In addition, the modelled payload appeared to have underestimated the volume of the final payload. It is hypothesized that this could have been due to a variable density, but further field testing would be required to determine the actual cause of the error.

An additional limitation of the software is related to material that does not fit the assumption of a conical shape. This can occur when material is sticky and forms clumps. Under these conditions the modelling software would not be able to provide an accurate shape, instead providing an equivalent cone shape based on the strut pressures.

8 Future Work

8.1 Search Algorithm

The current search method used by the modelling software, while effective, has limited use due to its slow processing speed. Two other search methods were also suggested but were found to have limited accuracy as currently defined. It is hypothesized that with further research, these search methods could determine the correct load shape with a similar accuracy to the full search.

The search rectangle method, while faster than the full search did not perform quickly enough for real time use using the parameters defined. It could be improved by determining the maximum distances that a calculated centroid can deviate from the location of the cone peak in both the x and y directions. Using these distances a search area could be defined at the minimum size required to always include the correct shape. The processing speed at this size could then be tested to determine if it met the requirements for real time operation.

The progressively narrowing search method was shown to be capable of providing results within the time constraints required for real time use. For this reason, this search method has the best potential for implementation in the field. This method could also be improved by determining the maximum deviation between peak and centroid. However, it is likely of greater benefit to this method to determine any trends between this deviation and the position of the cone peak within the truck body. Using these trends the rough searches performed by the first two phases could likely be better defined to provide greater accuracy.

8.2 Further Field Testing

Since this work only included the field results of one loading cycle, it is recommended that further tests be performed to determine any additional issues or limitations. Further testing may provide the information required to help determine and solve issues such as the modelling software's deviation from the measured centroid. Of particular use, would be strut pressure data paired with pass by pass photos of the payload so that the modelled surface could be visually compared to the actual payload.

8.3 Variable Material Density

It was hypothesized that variable material density may have contributed to the modelling software's underestimation of the payload volume. The software could be adjusted to model each pass with a unique density. In order to do so, an outside source of density would be required.

Variations in material density can cause a change in the required digging power of an excavator. The power used to dig a load could be compared to the digging power required for a load at the average material density. Based on this comparison a factor could likely be determined and applied to increase or decrease the material density. If these factors were determined correctly it could result in a more robust model.

8.4 Data Input

In order to function as a real time operator assistance tool, the software model would need to be able to monitor live data in order to detect when a load has been placed and when to take a pressure reading. This could be done by sampling the recorded data at a frequency equal to the rate at which data is recorded. These samples could then be collected in a buffer.

The most recent values could be compared with previous values within the buffer, until a significant enough change is detected to indicate a load had been placed. Once a load was detected, the data buffer would be further monitored to determine when to take a reading to use in modelling the shape. It is important that his reading be taken at the correct time, as the suspension cylinders are often put in motion by the impact of a load placement. The most accurate readings will be taken once the motion has stopped, but must also be taken early enough that the software can determine a shape before the next load is ready to be placed.

It is critical that the sampling of data be in sync with it being recorded. Therefore, it is important that the other functions of the software not cause any delays in the sampling process. To do this the sampling could be done in a separate processing thread from the modelling functions. A flow chart of how this process would work is shown in Figure 8-1.

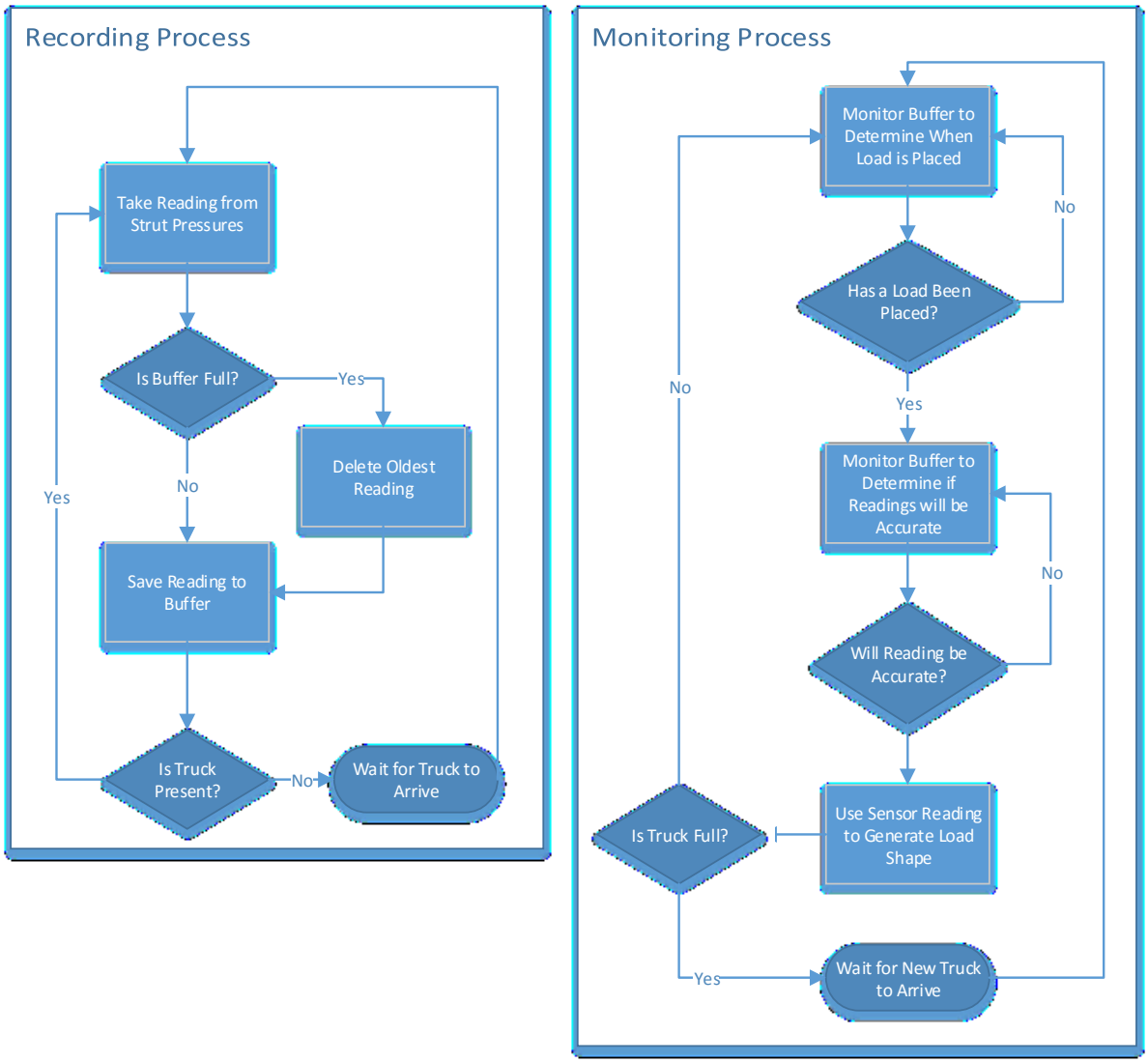


Figure 8-1 Data Input Algorithms

8.5 Automation

If required, the modelling software could be modified for use with an automated excavator. An automated piece of equipment's motions are often determined by predefined scripts. These scripts require information in order to make correct decisions. In the case of excavators, the scripts require a model of the payload shape within the truck body in order to decide where to place the next pass. The modelling software could be adjusted to provide such a model instead of the visual output it currently provides.

9 References

[1] A. Chamanara and T. G. Joseph, "Adverse Effect of Unbalanced Payloads on Mining Haul Truck KPIs," Canadian Institute of Mining, Metallurgy and Petroleum, 2013.

[2] L. G. Hagenbuch, "Adapting the Off-Highway Truck Body Volumetric Process to Real World Conditions," *SAE Technical Paper 2000-01-2652,* pp. doi:10.4271/2000-01-2652, 2000.

[3] L. Hagenbuch, "Off-Highway Truck Body True Capacity... Why can't I get rated payload on my off-highway trucks without hungry boards and tail extensions?," in *SME Annual Meeting*, Phoenix, 2002.

[4] SAE, "SAE J1363," SAE International, 2003.

[5] T. G. Joseph and A. Chamanara, "Hauler Body Payload Balance," Canadian Institute of Mining, Metallurgy and Petroleum, 2012.

[6] A. Chamanara, "Enhancing Mine Haul Truck KPIs via Payload Balance," University of Alberta, Edmonton, 2013.

[7] Caterpillar, "797F," 8 May 2015. [Online]. Available: http://www.cat.com/en_US/products/new/equipment/off-highway-trucks/mining-trucks/18093014.html.

[8] T. G. Joseph, "Large Mobile Mining Equipment Operating on Soft Ground," in *International Mining Congress and Exhibition of Turkey*, 2003.

[9] D. Whalen and K. Obaia, "Effect of Oil Sand Mining Operations on Haul Truck Dump Body Design," in *Operational Challenges in the Canadian Oilsands*, Montreal, 2003.

[10] J. Zhou, R. A. Hall, G. Fowler and K. Huntingford, "Applications of Engineering Analysis to Improve Tire Management," Canadian Institute of Mining, Metallurgy and Petroleum, Vancover, 2006.

[11] Michelin, "Maintenance Guide for Earthmover Tyres: Factors affecting tyre wear life," 11 May 2015. [Online]. Available: http://www.miningsafety.co.za/dynamiccontent/42/Maintenance-Guide-for-Earthmover-Tyres-Factors-affecting-tyre-wear-life.

[12] T. G. Joseph, "OsEIP: The Oil Sands - Equipment Interactions Program," *Canadian Institute of Mining, Metallurgy and Petroleum Bulletin,* pp. 58-61, 2002.

[13] J. J. Berezan, T. G. Joseph and V. D. del Valle, "Monitoring whole body vibration effects on ultra-class haulers," *CIM Bulletin,* September 2004.

[14] C. Schexnayder, S. L. Weber and B. T. Brooks, "Effect of Truck Payload Weight on Production," *Journal of Construction Engineering and Management 1999.125,* pp. 1-7, 1999.

[15] J. Zhou, R. A. Hall and K. Huntingford, "Engineering Analyses of Truck Weight Distribution Variances to Improve Tire Management," in *MPES 2012*, New Delhi, 2012.

[16] Rosemount, "Products: 3D Solids Scanners," 7 May 2015. [Online]. Available: http://www2.emersonprocess.com/en-us/brands/rosemount/level/solids-measurement/3d-solids-scanners/rosemount-series-5708s/pages/index.aspx.

[17] Economic Commission for Europe, "Code of Uniform Standards and Procedures for the Performance of Draught Surveys of Coal Cargoes," United Nations - Economic and Social Council, 1992.

[18] Caterpillar, "VITAL INFORMATION MANAGEMENT SYSTEM," 4 May 2015. [Online]. Available: http://www.cat.com/en_US/support/operations/technology/fleet-management-solutions/vims.html.

[19] S. Arakawa, "Development and Deployment of KOMTRAX STEP 2," Komatsu, 2002.

[20] J. J. Slob, "Payload Estimation for Electric Mining Shovels using a Load Sensing Pin," University of Queensland, Brisbane, 2007.

[21] M. G. Lipsett, "Methods for Assessing Dynamic Performance of Shovels," in *18th International Symposium on Mine Planning & Equipment Selection*, Banff, 2009.

[22] Walz Scale, "Walz Mining," 10 May 2015. [Online]. Available: http://www.walzscale.com/walz-mining.

[23] E. Duff, "Automated Volume Estimation ofHaul-Truck Loads," CSIRO Manufacturing Science and Technology - Queensland Centre for Advanced Technologies, 2000.

[2 Walz Scale, "Walz Load Scanner," 3 May 2015. [Online]. Available:
4] http://loadscanner.com/.

[2 M. Dunbabin and P. Corke, "Autonomous Excavation Using a Rope Shovel," in *Journal of*
5] *Field Robotics, 23*, Wiley Periodicals, Inc., 2006, p. 379–394.

[2 M. Dunbabin and K. Usher.US Patent 8903689 B2, 2014.
6]

[2 J. Borthwick, "Mining Haul Truck Pose Estimation and Load Profiling Using Stereo
7] Vision," University of British Columbia, Vancouver, 2009.

[2 M. Baker.US Patent 6,157,889, 2000.
8]

[2 P. Rowe and A. Stentz, "Parameterized Scripts for Motion Planning," Robotics Institute -
9] Camegie Mellon University, Pittsburgh, 1997.

[3 Wenco, "Production Management," 11 May 2015. [Online]. Available:
0] http://www.wencomine.com/core-system/production-management/.

[3 T. S. Golosinski and H. Hu, "Data Mining Uses in Mining," in *APCOM 2001*, Beijing, 2001.
1]

[3 R. J. Hardy, "Four-Pass Loading - Must-Have or Myth," in *Fifth Large Open Pit Mining*
2] *Conference*, Kalgoorlie, 2003.

[3 Walz Scale, "Load Scanner System Operation," 14 May 2015. [Online]. Available:
3] http://loadscanner.com/otr-tire-optimization.

[3 Caterpillar, "How Big Is The Vehicle That Uses Those Tires?," 25 May 2015. [Online].
4] Available: http://robertkaplinsky.com/wp-content/uploads/2012/12/dimensions_small.jpg.

[3 Caterpillar, "Caterpillar 797B Rock Truck," 17 April 2015. [Online]. Available:
5] http://www.ritchiespecs.com/specification?type=Con&category=Rock+Truck&make=Caterpillar&model=797B&modelid=93274#.

[3 J. Stewart, "Newton's Method," in *Calculus Early Transcendentals 6E*, Scarborough, Nelson
6] Education Limited, 2007, pp. 334-340.

[3 C. J. Ridder, "New Algorithm for Computing a Single Root of a Real Continuous Function,"
7] *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, VOL. CAS-26, NO. 11,* pp. 979-980, 1979.

[3 Caterpillar, "Caterpillar 797B Rock Truck," 17 April 2015. [Online]. Available:
8] http://www.ritchiespecs.com/specification?type=Con&category=Rock+Truck&make=Caterpillar&model=797B&modelid=93274#.

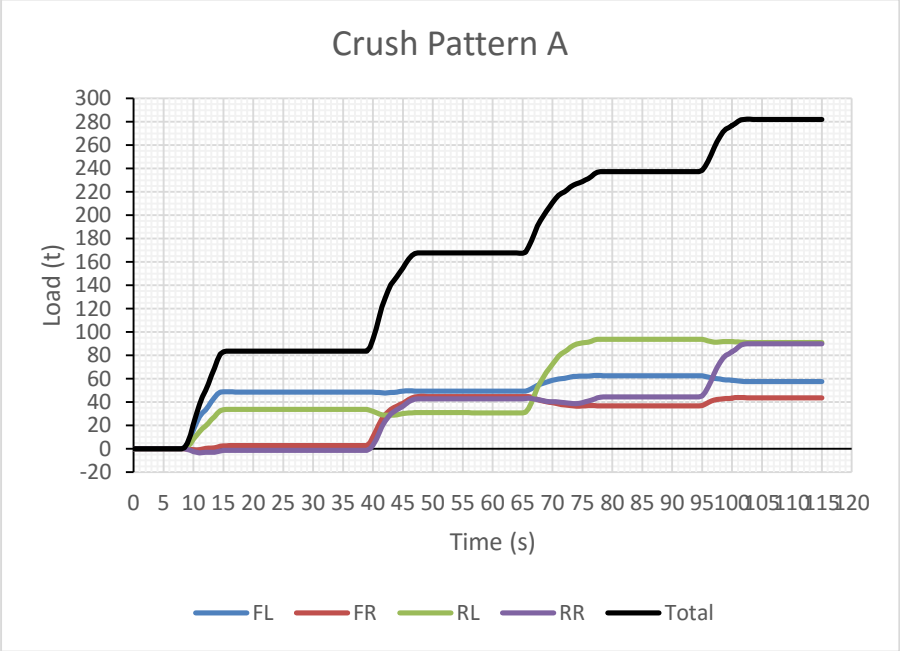
[3 Artech Industries Inc, "S-Beam Load Cells: Model 20210," 17 April 2015. [Online].
9] Available: http://www.artech-loadcell.com/20210.htm.

[40] HBM Canada, "MGCplus - Laboratory & Test Stand DAQ," 17 April 2015. [Online]. Available: http://www.hbm.com/en/menu/products/measurement-electronics-software/laboratory-test-stand/.

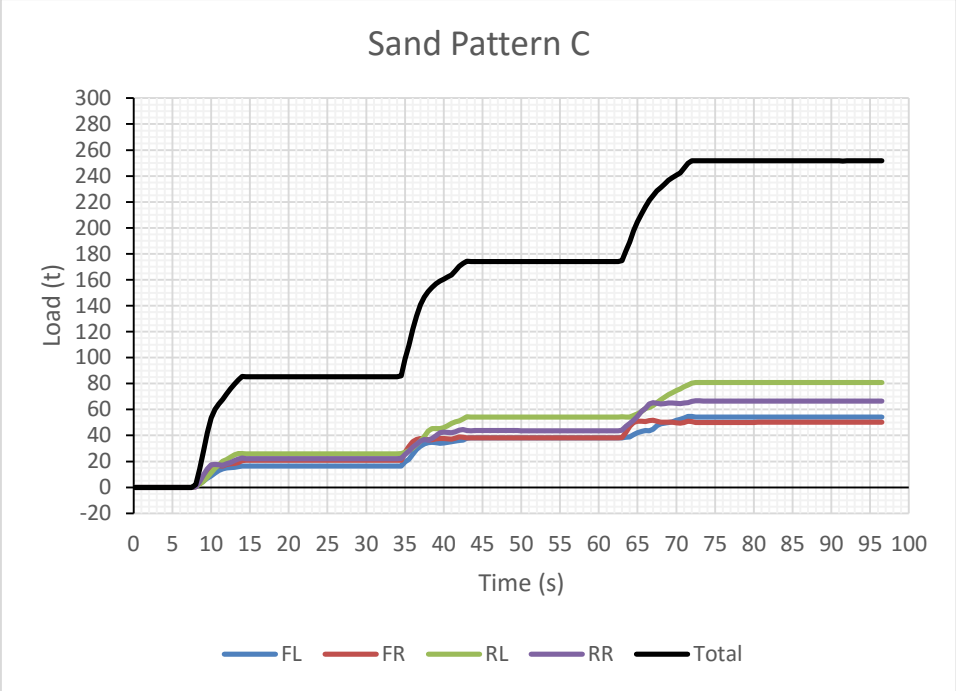
Appendices

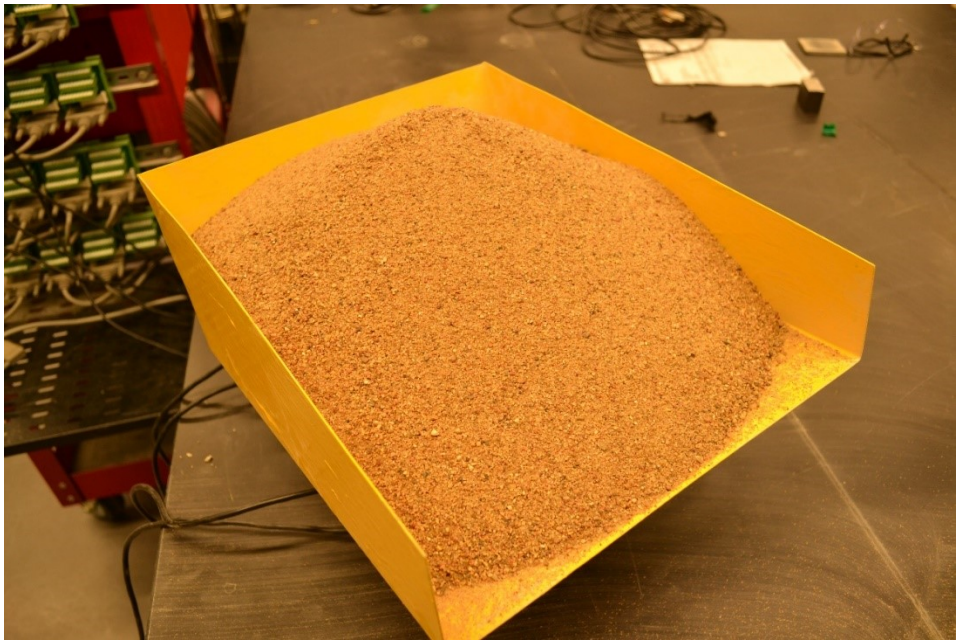
Appendix I

Crush Pattern A

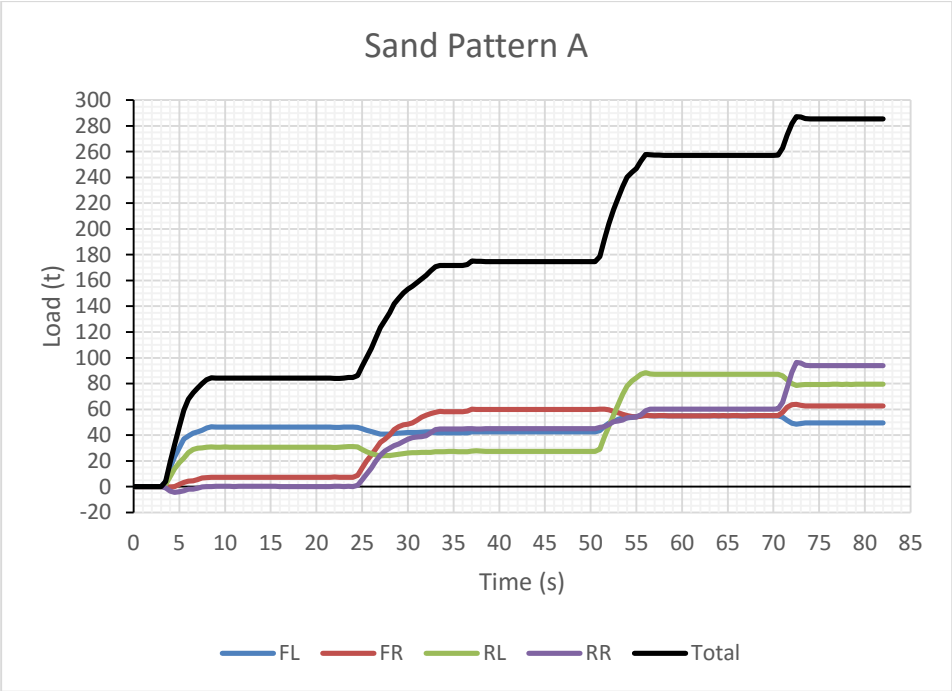


Sand Pattern C





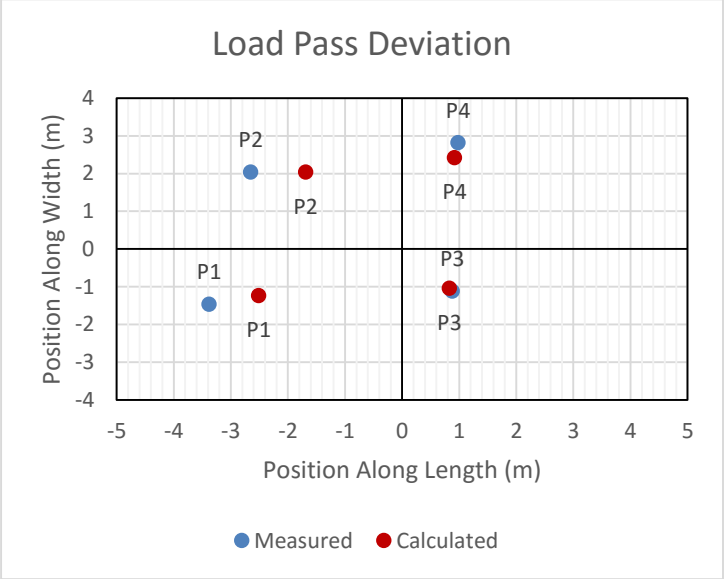
Sand Pattern A



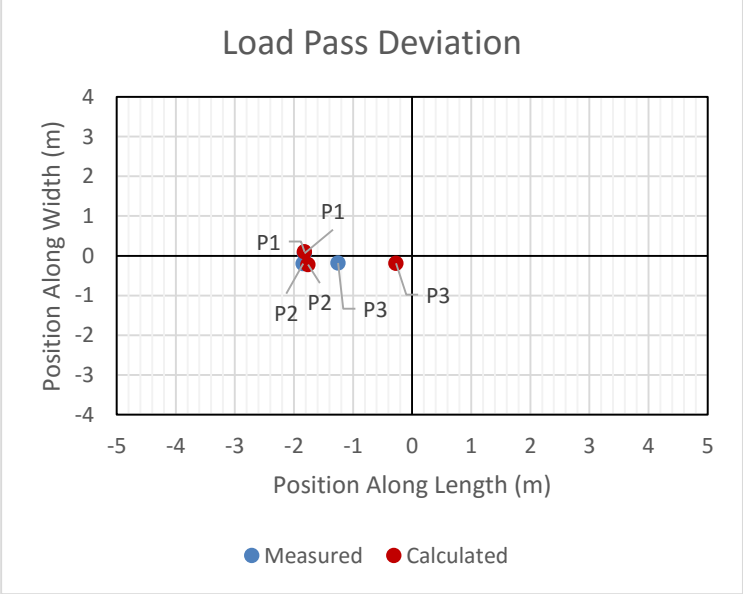
Appendix II

Individual Load Centroid Comparison

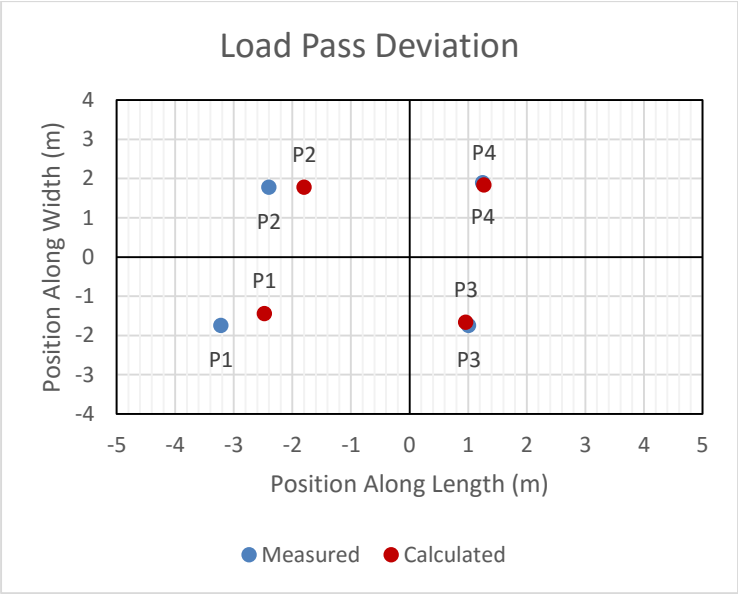
Sand Pattern A



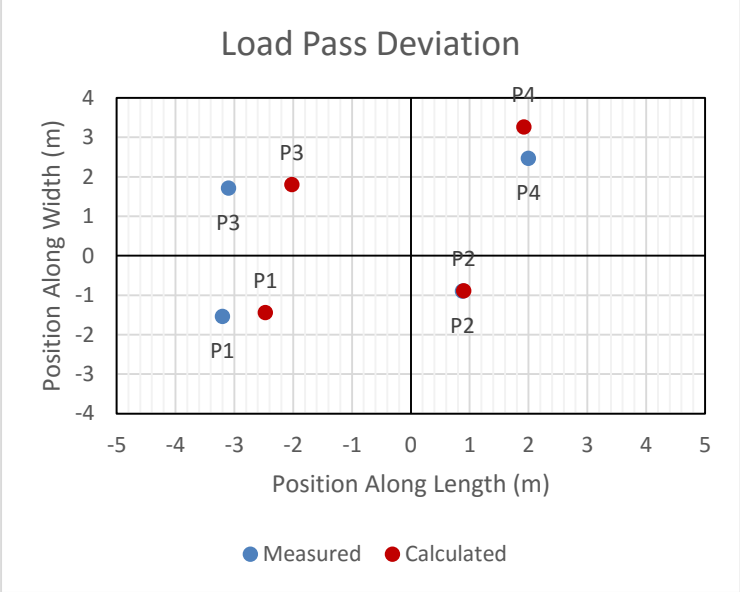
Sand Pattern C



Crush Pattern A

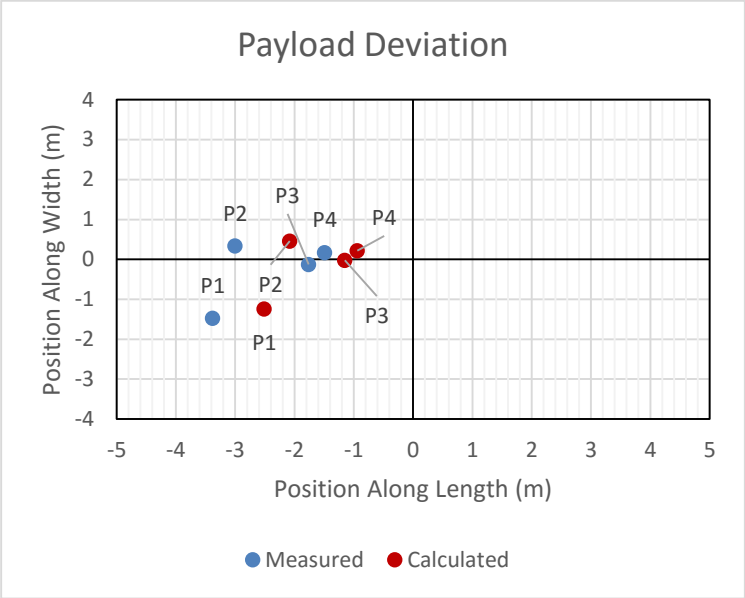


Crush Pattern B

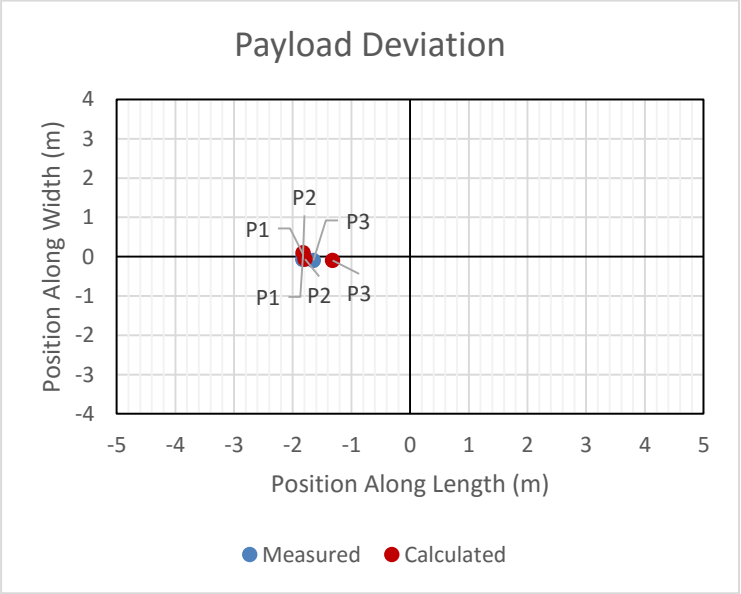


Total Payload Centroid Comparison

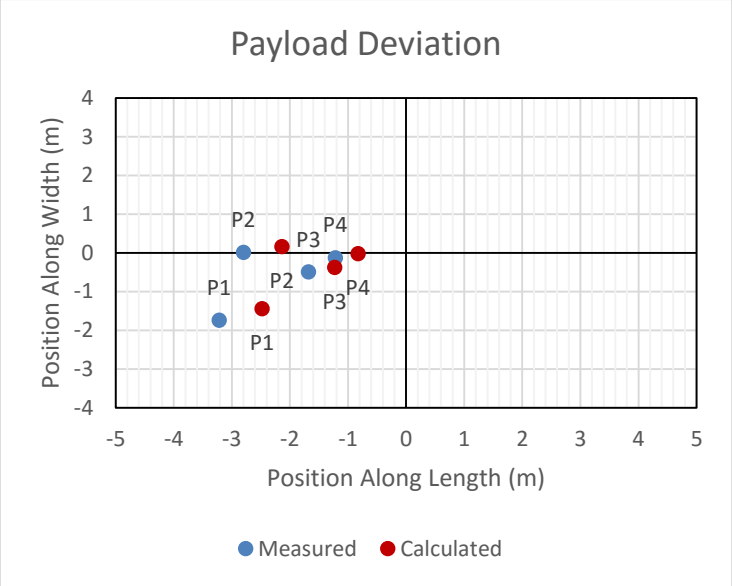
Sand Pattern A



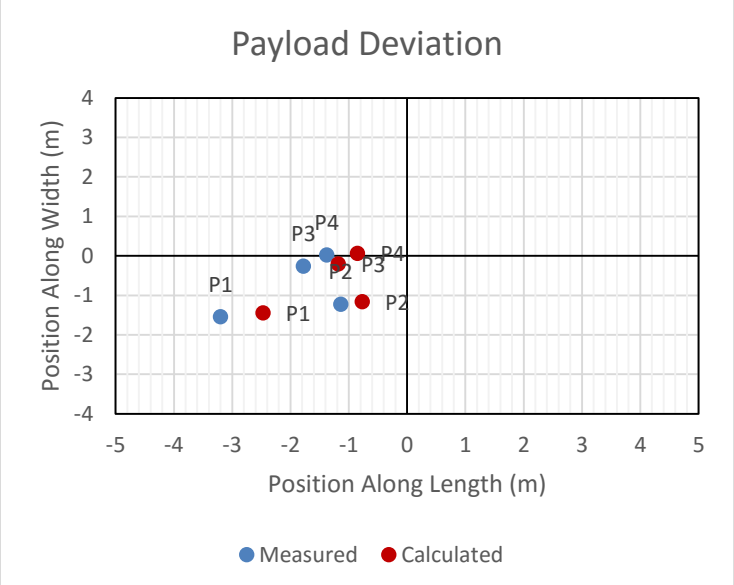
Sand Pattern C



Crush Pattern A

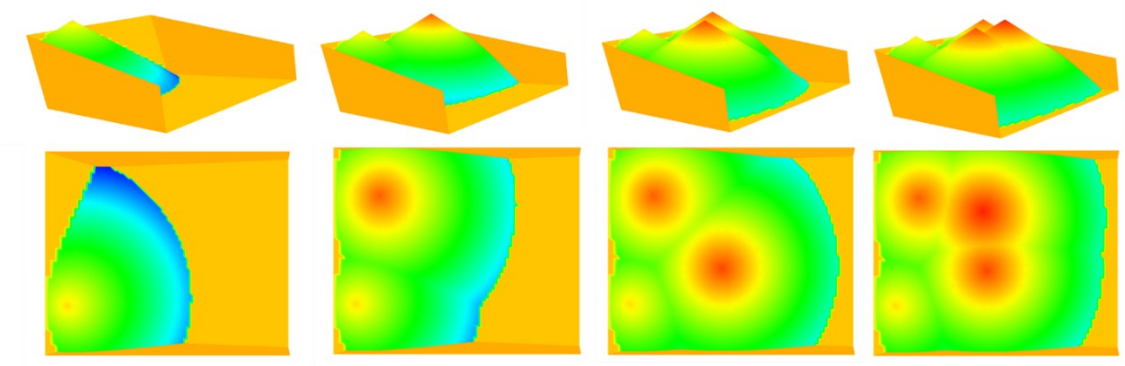


Crush Pattern B

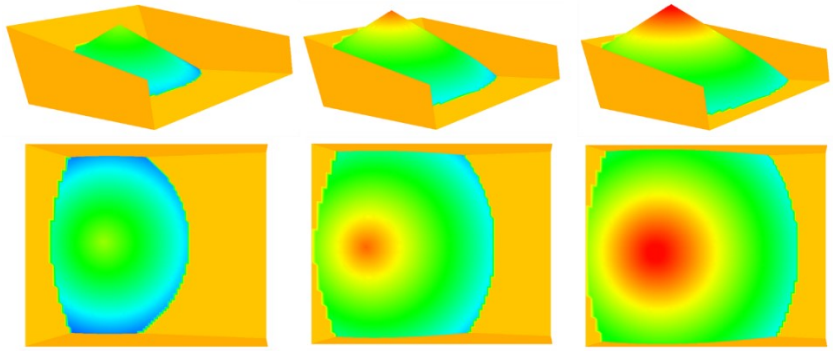


Payload Models

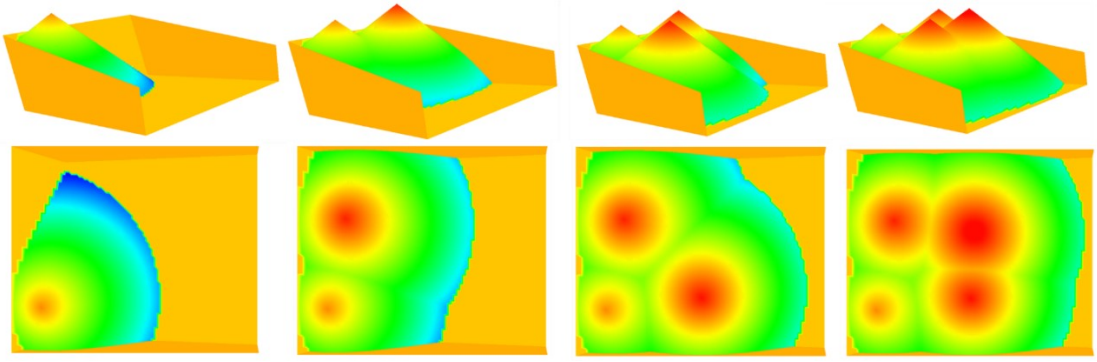
Sand Pattern A



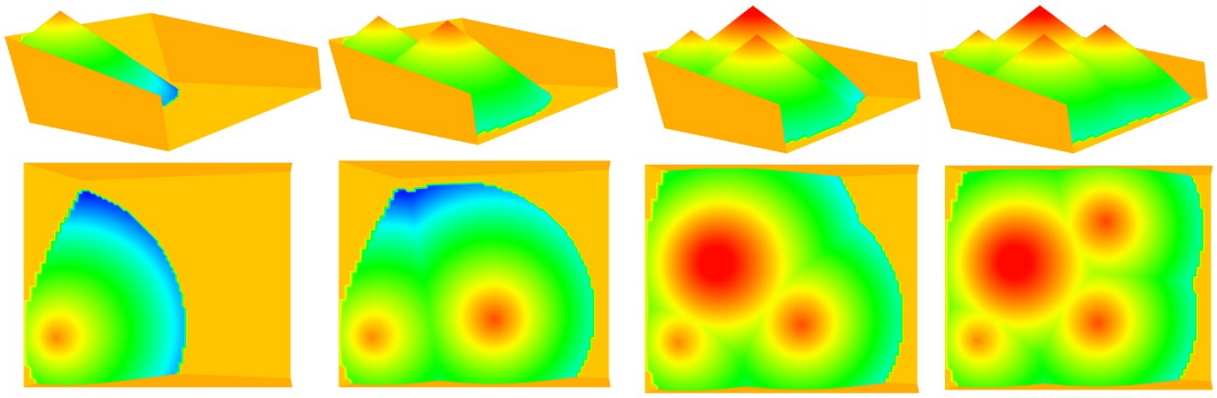
Sand Pattern C



Crush Pattern A

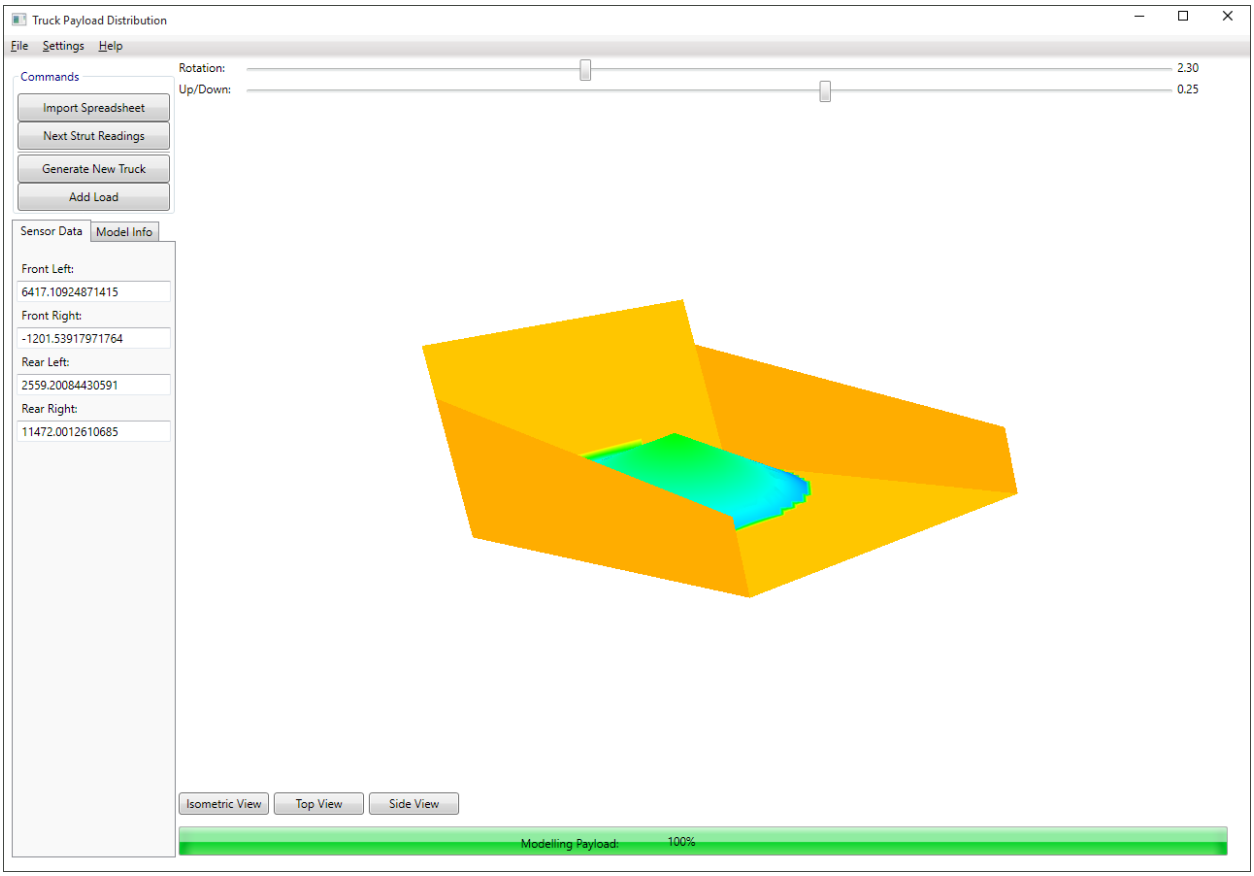


Crush Pattern B



Appendix III

User Interface



Main Flow

1. Import Strut Data

User clicks “Import Spreadsheet” button

User selects suitable spreadsheet from popup file browser

Initialize Workbook variable equal to selected Excel Workbook

Initialize Worksheet variable equal to selected Excel Worksheet

2. Create & Display Truck Model

User clicks “Generate New Truck” button

Round pre-set Body Length and Width to nearest integer

Initialize BodyFloorCoordinates Surface3D array
    Use rounded Body Length and Width, and Grid Size to determine array length

Determine total number of grid points

For each point in BodyFloorCoordinates
    Call *CalculateBodyFloorCoordinates* to determine z-coordinate
End for loop

Save BodyFloorCoordinates to CurrentShell property in MainWindowViewModel

Create triangle mesh of BodyFloorCoordinates

Create model of body sides
    Call *BodySides* for lab tests (or *AltBodySides* for field data) to determine coordinates
    Use points to create a triangle mesh

Add floor and side models to viewport

Move and scale model to be centered in viewport
    Find center of x, y and z ranges
    Translate model to center of viewport
    Scale model to fit within viewport

Assign translation and scaling parameters to MainWindowViewModel ViewMatrix property
Display models

Clear any variables saved from last truck

Reset results to zero

3. Collect Strut Reading

User clicks “Next Strut Readings” button

Initialize Excel Range variable r

Determine how many load passes have been placed

Select the spreadsheet values corresponding to the load pass strut pressures

Assign values to MainWindowViewModel SensorReadings property

4. Determine and Display Load Shape and Position

User Clicks “Add Load Button”

Create new background thread

Get input SettingsData and SensorReadings from MainWindowViewModel

Round pre-set Body Length and Width to nearest integer

Initialize Surface3D arrays for BodyFloorCoordinates and ConeCoordinates
    Use rounded Body Length and Width, and Grid Size to determine array length

Calculate change in sensor readings from previous load or empty state

Calculate weight and estimate volume of most recently placed load pass

Calculate centroid of placed load using sensor readings

For each point in BodyFloorCoordinates
    Call *CalculateBodyFloorCoordinates* to determine z-coordinate

End For

Determine shape of simple cone with an above body volume that best meets Acceptable Shape requirements
    For each point on the grid
        Calculate shape which meets estimated load volume
            Call *CalculateCentroids*
    End For

    Search all shapes to determine which has a centroid closest to measured centroid and output coordinate of cone peak

Create a Surface3D array to represent a simple cone with peak located at previously output coordinate

Initialize Surface3D CurrentCoordinates to MainWindowViewModel.CurrentShell

For each point in ConeCoordinates
    If ConeCoordinates[i].z is below the truck body or last placed load,
        CurrentCoordinates[i].z = CurrentCoordinates[i].z
    Else
        CurrentCoordinates[i].z = ConeCoordinates[i].z
    EndIf
End For

Set MainWindowViewModel.CurrentShell equal to CurrentCoordinates

Assign colors to each point in the array based on the z-coordinate
    Color gradient goes from blue to yellow to red

Create triangle mesh model of array

End background thread

Apply ViewMatrix to translate and scale the model to match the body model

Display mesh model

5. Repeat Steps 3 & 4 until all load passes have been modelled

Functions

CalculateBodyFloorCoordinates(x, y, xMin, yMin, SettingsData)

Retrieve FrontSlope, RearSlope, and BodyDepth from SettingsData

If x < (xMin + (BodyDepth / Tan(FrontSlope)))
    z-coordinate = BodyDepth – (x - xMin) * Tan(FrontSlope)

Else
    z-coordinate = (x – xMin – BodyDepth / Tan(FrontSlope)) * Tan(RearSlope)

End If

Return z-coordinate

BodySides(SettingsData)

Retrieve BodyLength, BodyDepth, BodyWidth, FrontSlope and RearSlope from SettingsData

Initialize BodyPoints Point3D array of size 12

Assign:

$$BodyPoints[0].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[0].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[0].z = 0$$

$$BodyPoints[1].x = -\frac{BodyLength}{2}$$
$$BodyPoints[1].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[1].z = BodyDepth$$

$$BodyPoints[2].x = -\frac{BodyLength}{2} + BodyLength - \frac{BodyDepth - \left(BoxLength - \frac{BodyDepth}{tan(FrontSlope)}\right) * \tan(RearSlope)}{\tan(90 - RearSlope)}$$

$$BodyPoints[2].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[2].z = BodyDepth$$

$$BodyPoints[3].x = -\frac{BodyLength}{2} + BodyLength$$
$$BodyPoints[3].y = -\frac{BodyWidth}{2}$$

$$BodyPoints[3].z = \left(BodyLength - \frac{BodyDepth}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

$$BodyPoints[4].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[4].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[4].z = \text{BodyDepth}$$

$$BodyPoints[5].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[5].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[5].z = \left(BodyLength - \frac{BodyDepth}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

$$BodyPoints[6].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[6].y = \frac{BodyWidth}{2}$$
$$BodyPoints[6].z = 0$$

$$BodyPoints[7].x = -\frac{BodyLength}{2}$$
$$BodyPoints[7].y = \frac{BodyWidth}{2}$$
$$BodyPoints[7].z = BodyDepth$$

$$BodyPoints[8].x = -\frac{BodyLength}{2} + BodyLength - \frac{BodyDepth - \left(BoxLength - \frac{BodyDepth}{tan(FrontSlope)}\right) * \tan(RearSlope)}{\tan(90 - RearSlope)}$$
$$BodyPoints[8].y = \frac{BodyWidth}{2}$$
$$BodyPoints[8].z = BodyDepth$$

$$BodyPoints[9].x = -\frac{BodyLength}{2} + BodyLength$$
$$BodyPoints[9].y = \frac{BodyWidth}{2}$$
$$BodyPoints[9].z = \left(BodyLength - \frac{BodyDepth}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

$$BodyPoints[10].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[10].y = \frac{BodyWidth}{2}$$
$$BodyPoints[10].z = \text{BodyDepth}$$

$$BodyPoints[11].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$

$$BodyPoints[11].y = \frac{BodyWidth}{2}$$
$$BodyPoints[11].z = \left(BodyLength - \frac{BodyDepth}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

Return BodyPoints

AltBodySides(SettingsData)

Retrieve BodyLength, BodyDepth, BodyWidth, FrontSlope and RearSlope from settings

Initialize BodyPoints Point3D array of size 12

Assign:

$$BodyPoints[0].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[0].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[0].z = 0$$

$$BodyPoints[1].x = -\frac{BodyLength}{2} + \frac{801}{\tan(FrontSlope)}$$
$$BodyPoints[1].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[1].z = BodyDepth - 801$$

$$BodyPoints[2].x = -\frac{BodyLength}{2} + BodyLength - \frac{BodyDepth - \left(BoxLength - \frac{BodyDepth}{tan(FrontSlope)}\right) * \tan(RearSlope)}{\tan(90 - RearSlope)}$$

$$BodyPoints[2].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[2].z = BodyDepth - 801$$

$$BodyPoints[3].x = -\frac{BodyLength}{2} + BodyLength$$
$$BodyPoints[3].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[3].z = \left(BodyLength - \frac{BodyDepth}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

$$BodyPoints[4].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[4].y = -\frac{BodyWidth}{2}$$
$$BodyPoints[4].z = BodyDepth - 801$$

$$BodyPoints[5].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[5].y = -\frac{BodyWidth}{2}$$

$$BodyPoints[5].z = \left(BodyLength - \frac{BodyDepth - 801}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

$$BodyPoints[6].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[6].y = \frac{BodyWidth}{2}$$
$$BodyPoints[6].z = 0$$

$$BodyPoints[7].x = -\frac{BodyLength}{2} + \frac{801}{\tan(FrontSlope)}$$
$$BodyPoints[7].y = \frac{BodyWidth}{2}$$
$$BodyPoints[7].z = BodyDepth - 801$$

$$BodyPoints[8].x = -\frac{BodyLength}{2} + BodyLength - \frac{BodyDepth - \left(BoxLength - \frac{BodyDepth}{tan(FrontSlope)}\right) * \tan(RearSlope)}{\tan(90 - RearSlope)}$$

$$BodyPoints[8].y = \frac{BodyWidth}{2}$$
$$BodyPoints[8].z = BodyDepth - 801$$

$$BodyPoints[9].x = -\frac{BodyLength}{2} + BodyLength$$
$$BodyPoints[9].y = \frac{BodyWidth}{2}$$
$$BodyPoints[9].z = \left(BodyLength - \frac{BodyDepth}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

$$BodyPoints[10].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[10].y = \frac{BodyWidth}{2}$$
$$BodyPoints[10].z = BodyDepth - 801$$

$$BodyPoints[11].x = -\frac{BodyLength}{2} + \frac{BodyDepth}{\tan(FrontSlope)}$$
$$BodyPoints[11].y = \frac{BodyWidth}{2}$$
$$BodyPoints[11].z = \left(BodyLength - \frac{BodyDepth - 801}{\tan(FrontSlope)}\right) * \tan(RearSlope)$$

Return BodyPoints

CalculateCentroids(vEst, nX, nY, x, y, currentProfile, BodyCoordinates, SettingsData)

Initialize ConePoints Array centroids

For i = 0 to i = nX - 1

```
        For j = 0 to j = nY - 1
                Initialize ConePoints cone variable
                Calculate the peak of the shape that meets requirements at point (i , j)
                        Call CalculateShape
                centroids[i,j] = cone
        End For
End For

Return centroids
```

CalculateShape(vEst, xCenter, yCenter, currentProfile, BodyCoordinates, nX, nY, SettingsData)

```
Initialize ConePoints centroid
Retrieve AngleRepose from SettingsData
Initialize Surface3D heightAboveBody
Initialize Surface3D coneCoordinates
Initialize double he = (3 * (vEst) * (tan(AngleRepose) ^ 2) /π) ^ (1 / 3)
Initialize double hb = he
Initialize double hn
Initialize double fn
Initialize double xWeighted
Initialize double yWeighted
Initialize double totalVolume
Initialize int n = currentProfile.ArrayLength()
Initialize int count = 0

Do
        Initialize double hm = (ha + hb) / 2
       Initialize double Array volumes

        Assign values to volumes based on points ha, hb and hm for Ridder's Method
                Call CalculateRiddersCones

        Initialize double fa

        Assign value of function at ha to fa
                fa = volumes[0] – vEst

       Initialize double fb

        Assign value of function at hb to fb
                fb = volumes[1] – vEst

       Initialize double fm
```

```
        Assign value of function at hm to fm
                fm = volumes[2] – vEst

        Assign:
                hn = hm + (hm - ha) * (Sign(fa - fb) * fm) / (Sqrt( (fm ^ 2) - (fa * fb)))
                xWeighted = 0
                yWeighted = 0
                totalVolume = 0

        Initialize LoadInfo loadn

        Assign value to loadn
                Call Calculate Cone for lab data (or CalculateConeAlt for field data)

        Assign:
                fn = loadn.Volume - vEst
                totalVolume = loadn.Volume
                xWeighted = loadn.XWeighted
                yWeighted = loadn.YWeighted

        If fn and fm are different signs
              ha = hm;
              hb = hn;
       Else If fn and fa are different signs
              hb = hn;
        Else
               ha = hn;
        End If

       count += 1;
While |fn| > 10 & count <= 100

centroid.X = xCenter
centroid.Y = yCenter
centroid.CentroidX = xWeighted / totalVolume
centroid.CentroidY = yWeighted / totalVolume
centroid.H = hn

Return centroid
```

CalculateRiddersCones(ha, hb, hm, n, xCenter, yCenter, coneCoordinates, BodyCoordinates, currentProfile, SettingsData, nX, nY)

```
Retrieve GridSize from SettingsData
Initialize double totalVolumeA = 0
Initialize double totalVolumeB = 0
```

```
Initialize double totalVolumeM = 0
Initialize Surface3D aConeCoord
Initialize Surface3D bConeCoord
Initialize Surface3D mConeCoord
Initialize Surface3D heightAboveBodyA
Initialize Surface3D heightAboveBodyB
Initialize Surface3D heightAboveBodyM

For i = 0 to i = n-1
        Initialize ColoredPoint3D point = coneCoordinates[i];

        Assign value to aConeCoord[i].z
                Call CalculateConeCoordinate
        Assign value to bConeCoord[i].z
                Call CalculateConeCoordinate
        Assign value to mConeCoord[i].z
                Call CalculateConeCoordinate

        If aConeCoord[i].z is at Or above currentProfile[i].z
                heightAboveBodyA[i].z = aConeCoord[i].z - currentProfile[i].z
        Else
                heightAboveBodyA[i].z = 0
        End If

        If bConeCoord[i].z is at Or above currentProfile[i].z
                heightAboveBodyB[i].z = bConeCoord[i].z - currentProfile[i].z
        Else
               heightAboveBodyB[i].z = 0
        End If

        If mConeCoord[i].z is at Or above currentProfile[i].z
                heightAboveBodyM[i].z = mConeCoord[i].z - currentProfile[i].z
        Else
                heightAboveBodyM[i].z = 0
        End If

        Add (heightAboveBodyA[i].z * GridSize ^ 2) to totalVolumeA
        Add (heightAboveBodyB[i].z * GridSize ^ 2) to totalVolumeB
        Add (heightAboveBodyM[i].z * GridSize ^ 2) to totalVolumeM
End For

Initialize double Array volume
        volume[0] = totalVolumeA
        volume[1] = totalVolumeB
        volume[2] = totalVolumeM
```

Return volume

CalculateConeCoordinate(x, y, xMin, yMin, SettingsData, cGX, cGY, h)
Retrieve Angle Repose from SettingsData

Initialize z

z = h - Sqrt((cGX - x) ^ 2 + (cGY - y) ^ 2) * tan(AngleRepose)

Return z

CalculateCone(h, n, xCenter, yCenter, xWeighted, yWeighted, totalVolume, coneCoordinates, BodyCoordinates, currentProfile, heightAboveBody, SettingsData settings, nX, nY)

```
For i = 0 to n -1
        Retrieve GridSize, BodyLength, Body Depth, FrontAngle from SettingsData
        Initialize ColoredPoint3D point

        point = coneCoordinates[i]

        Assign z-coordinate to coneCoordinates
                Call CalculateConeCoordinate

        If coneCoordinates[i].z is above currentProfile[i].z
             heightAboveBody[i].z = coneCoordinates[i].z - currentProfile[i].z
        Else
             heightAboveBody[i].z = 0
        End If

        If coneCoordinates[i].x is at the front or rear edge of the body
             If coneCoordinates[i].z is above the top edge of the body
                Bias centroid so that it is located outside of grid to mark shape as invalid
             Else
                Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^ 2) xWeighted
                Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^ 2) to
yWeighted
             End If
        End If

        Initialize double bodyLengthBeforeSlope
        Initialize double tanTheta

        bodyLengthBeforeSlope = -BodyLength / 2 + BodyLength - ((BodyDepth - (BodyLength
        - (BodyDepth / tan(FrontAngle))) * tan(RearAngle)) / tan(90 - RearAngle))
```

```
tanTheta = (BodyDepth - (BodyLength - BodyDepth / tan(FrontAngle)) *
tan(RearAngle)) / (BodyLength - bodyLengthBeforeSlope);

If coneCoordinates[i].y is at the left or right edge of the body
        If coneCoordinates[i].x is between the front And the point before the top edge of
        the body slopes
                If coneCoordinates[i].z is above the top edge of the body
                        Bias centroid so that it is located outside of grid to mark shape as
                        invalid
                Else
                        Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^
                        2) to xWeighted
                        Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^
                        2) to yWeighted
                End If
        Else
                If coneCoordinates[i].z is above sloping portion of top body edge
                        Bias centroid so that it is located outside of grid to mark shape as
                        invalid
                Else
                        Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^
                        2) to xWeighted
                        Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^
                        2) to yWeighted
                End If
        End If

        Add heightAboveBody[i].z * (gridSize ^ 2) to totalVolume
        Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^ 2) to
        xWeighted
        Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^ 2) to
        yWeighted
End If

Initialize LoadInfo load

Assign:
        load.Volume = totalVolume;
        load.XWeighted = xWeighted;
        load.YWeighted = yWeighted;

Return load
```

CacluateAltCone(x, y, xMin, yMin, SettingsData, cGX, cGY, h)

```
For i = 0 to n -1
        Retrieve GridSize, BodyLength, Body Depth, FrontAngle from SettingsData
```

```
Initialize ColoredPoint3D point

point = coneCoordinates[i]

Assign z-coordinate to coneCoordinates
        Call CalculateConeCoordinate

If coneCoordinates[i].z is above currentProfile[i].z
      heightAboveBody[i].z = coneCoordinates[i].z - currentProfile[i].z
Else
      heightAboveBody[i].z = 0
End If

If coneCoordinates[i].x is at the front or rear edge of the body
      If coneCoordinates[i].z is above the top edge of the body
         Bias centroid so that it is located outside of grid to mark shape as invalid
      Else
         Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^ 2) xWeighted
         Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^ 2) to
         yWeighted
      End If
End If

Initialize double bodyLengthBeforeSlope
Initialize double tanTheta

bodyLengthBeforeSlope = -BodyLength / 2 + BodyLength - ((BodyDepth - (BodyLength
- (BodyDepth / tan(FrontAngle))) * tan(RearAngle)) / tan(90 - RearAngle))

tanTheta = (BodyDepth - (BodyLength - BodyDepth / tan(FrontAngle)) *
tan(RearAngle)) / (BodyLength - bodyLengthBeforeSlope);

If coneCoordinates[i].y is at the left or right edge of the body
        If coneCoordinates[i].x is on the front slope
                If coneCoordinates[i].z is above the top edge of the body
                        Bias centroid so that it is located outside of grid to mark shape as
                        invalid
                Else
                        Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^
                        2) to xWeighted
                        Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^
                        2) to yWeighted
                End If
        Else If coneCoordintes[i].x is between the front and rear sloping portions of the
        top edge
                If coneCoordinates[i].z is above the top edge of the body
```

```
                    Bias centroid so that it is located outside of grid to mark shape as
                    invalid
                Else
                    Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^
                    2) to xWeighted
                    Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^
                    2) to yWeighted
                End If
        Else
                If coneCoordinates[i].z is above sloping portion of top body edge
                    Bias centroid so that it is located outside of grid to mark shape as
                    invalid
                Else
                    Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^
                    2) to xWeighted
                    Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^
                    2) to yWeighted
                End If
        End If

        Add heightAboveBody[i].z * (gridSize ^ 2) to totalVolume
        Add heightAboveBody[i].x * heightAboveBody[i].z * (gridSize ^ 2) to
        xWeighted
        Add heightAboveBody[i].y * heightAboveBody[i].z * (gridSize ^ 2) to
        yWeighted
End If

Initialize LoadInfo load

Assign:
        load.Volume = totalVolume;
        load.XWeighted = xWeighted;
        load.YWeighted = yWeighted;

Return load
```

Classes

MainWindowViewModel

Constructor
Assign:
    Truck dimension and material properties pre-set values to SettingsData
    LoadCount = 0
    CurrentVolume = 0
    Initial camera position and facing to CameraViewModel
    Initial sensor readings to SensorViewModel

Properties
SettingsData SettingsData
double CurrentVolume
double ConeHeight
double CentroidX
double CentroidY
int LoadCount
Surface3D CurrentShell
Matrix3D ViewMatrix
CameraVM CameraViewModel
SensorVM SensorViewModel

SensorReadings

Properties
double FL
double FR
double RL
double RR

Methods
TotalWeight(SensorReadings)
    Initialize total double variable
    total = FL + FR + RL + RR
    Return total

SettingsData

Properties
double BodyLength
double BodyWidth
double BodyDepth
double FrontSlope
double RearSlope
double Xo
double Yo
double Xi

double Yi
double MaterialDensity
double AngleRepose
double GridSize

Surface3D

Constructor
Calculate incremental distances, dx and dy between points

```
For each x point (i = 0 to nX)
        For each y point (j = 0 to nY)
                Initialize double xI = xMin + dx * i
                Initialize double yI = yMin + dy * j

                Initialize int nI = j + I * nY

                points[nI].x = xI
                points[nI].y = yI
                points[nI].z = 0
        End For
End For
```

Properties
ColoredPoint3D Array points
int nX
int nY

Methods
ArrayLength()
        Return length of points

GetPoint(i, j)
        Return point coordinates at (i, j)

SetColor(i, j, color)
        Set points.color at point (i, j) to color

ConvertToMesh()
        Converts surface into a triangle mesh by creating triangles between adjacent points

ColoredPoint3D

Properties
Color color
Point3D point

ConePoints

Properties
double X
double Y
double H
double CentroidX
double CentroidY

LoadInfo

Properties
double Volume
double XWeighted
double YWeighted