Using Discrete Element Modelling (DEM) and Breakage Experiments To Model The Comminution Action in a Tumbling Mill

by

Chisenga Kulya

BEng, UNZA

A thesis submitted to the

Department of Mechanical Engineering

University of Cape Town

for the degree of

Master of Science (MSc)

March 2008

Statement of originality

I know the meaning of plagiarism and declare that the work presented in the thesis is to the best of my knowledge and belief original and my own work except as acknowledged in the text and that the material has not been submitted either in whole or in part for a degree at this or any other university.

Signed by candidate

Chisenga Kulya

Acknowledgments

The successful completion of this thesis would not have been realised without the valuable contributions of a lot of people. I would like to thank my Almighty God for giving me the grace and opportunity to do this Masters study. My heartfelt thanks goes to my Father and Mother for their love and always being there for me. My sister, Grace and brothers have always been good to me and believed in me. My Wife, Aggie and son, Mwansa were my main pillar and source of comfort, encouragement and drive throughout my thesis work.

My special thanks go to my supervisors, Dr Malcolm Powell, Dr Aubrey Mainza, Dr Indresan Govender and Dr Robert Sarracino. I was lucky to have such supervisors who believed more in me than myself. Andrew McBride is also especially thanked for the "C++" course that I enjoyed so much and for paving so much of the way for my DEM work and MATLAB post-processing procedures. Dr Manoj Kanal is acknowledged for carrying out the DEM/DWT simulations.

I thank the Comminution and CERECAM groups of the University of Cape Town for providing me with so many friends and guidance.

Abstract

The Discrete Element Method (DEM) is a powerful modelling tool that characterises the system at the individual particle level. This makes it particularly well suited for simulating tumbling mills whose charge is principally individual particles (steel balls, rocks and fines). The use of DEM to simulate tumbling mills has proliferated since the early 1990s and been successfully employed to predict important milling parameters such as charge motion, power draw, liner wear and impact energy distribution.

The ultimate aim of any model of the tumbling mill is to predict the product of the milling process. Current DEM simulations of the tumbling mill however do not simulate the breakage of the particles and as such can not directly predict the product. In order to predict the performance of industrial-scale tumbling mills, laboratory-scale mills are used to experimentally obtain data, which is then scaled up using black box mathematical models.

In this thesis a tumbling mill model that utilises the power of DEM to provide the mechanical environment and the energies available for breakage is proposed. The incorporation of DEM eliminates the need to scale up because DEM is able to simulate the actual industrial-scale device. Data from breakage experiments on the ore being treated is also incorporated into the model to determine the breakage functions. Population balance techniques are applied in the mathematical framework of the model to predict the product of the comminution process.

In order to test the proposed tumbling mill model, DEM simulations of a 1.695m diameter pilot SAG mill using charge based on actual operation data were performed and analysed. Results from the DEM simulation and Drop Weight Tester breakage experiments were then used in the proposed tumbling mill model to predict the evolution of the product size distribution.

Contents

List of Figures

List of Tables

List of Abbreviations

3D	-	Three-Dimensional
CAD	-	Computational Aided Design
CFD	-	Computational Fluid Dynamics
CMR	-	Centre for Minerals Research
CoC	-	Centre of Circulation
CoM	-	Centre of Mass
CSV	-	Comma-Separated Values
DEM	-	Discrete Element Method (Discrete Element Modelling)
DWT	-	Drop Weight Tester
HFS	-	High Fidelity Simulations
HPGR	-	High Pressure Grinding Rolls
IGES	-	Initial Graphics Exchange Specification
JKMRC	-	Julius Kruttschitt Mineral Research Centre
PFC	-	Particle Flow Code
RBT	-	Rotary Breakage Tester
SAG	-	Semi-Autogenous Grinding Mill
SHPB	-	Split Hopkinson Pressure Bar
STEP	-	Standard for Exchange of Product data
UCM	-	Unified Comminution Model
UCT	-	University of Cape Town
UFLC	-	Ultra-Fast Load Cell

Nomenclature

μ	-	Coefficient of friction
π	-	Pi (3.142)
υ	-	Poisson's ratio
ζ	-	Arbitrary parameter
ε	-	Coefficient of restitution
$\triangle t$	-	Time step
λ	-	Breakage rate
Λ	-	Number of collisions
ρ	-	Density
ω	-	Mill rotational speed

Chapter 1

INTRODUCTION

The Motivation, Hypothesis and Objectives of this thesis are presented in this chapter. This chapter also gives the layout of the thesis.

1.1 Motivation

The mining industry routinely uses tumbling mills for processing ores. The tumbling mills which are mainly ball and semi-autogenous are used to reduce large rock particles to fine sizes. A tumbling mill is a cylindrical drum fitted with conical or flat end plates on both sides. The inside is filled with steel balls (for ball and semi-autogenous mills) which is called the grinding media. The grinding media and ore are collectively referred to as the charge. The mill is rotated and the charge tumbles, which results in the ore being broken up in smaller particles. In continuous operations, the ore is continuously fed into the mill from one end and the ground particles are discharged from the other.

The process of size reduction, termed *comminution* is very energy intensive. Although the comminution process consumes a large portion of energy in a mineral processing plant, it is known that the process is not particularly energy efficient. In fact, it is believed that only 20% of the energy is utilized

in comminution, while the rest is wasted (Flavel and Rimmer (1981)). There is therefore potential to make significant energy savings in the milling process. The best way to discover and identify ways of saving energy in milling is by understanding the mode and mechanisms of energy utilization inside the mill. This requires a method that models the basic interactions of the individual particles as they collide in the charge.

The Discrete Element Method (DEM) is one of the numerical modelling methods that is used to model the internal dynamics of particle based systems. The originators of the DEM, Cundall and Strack (1979), described the method as being "based on the use of a numerical scheme in which the interaction of particles is monitored contact by contact and the motion of the particles modelled particle by particle". DEM simulates the motion and interaction of individual distinct particles. Therefore, DEM is suited for investigating the tumbling mill, in which the balls and ore form distinct particles that interact as they tumble.

The design, optimisation and performance of comminution devices constitute an important aspect of mineral processing. Conventionally experimental data based on laboratory to full-scale mills is scaled using black box models to predict the performance of industrial mills. This is an entirely empirical procedure which does not incorporate the actual breakage regime in the mill. Although the current empirical models work well if the design is in the same regime as where the experimental data was collected, the scale-up procedure is believed to result in inaccuracies due to differences in charge motion behaviour between laboratory-scale mills and industrial-scale mills (Datta and Rajamani (2002)).

The DEM on the other hand, is able to simulate directly the motion of the charge for industrial-scale mills as demonstrated by Cleary (2004). The configuration parameters and operating conditions of the DEM simulations can be closely matched with the actual configurations and operating con-

ditions, that control the throughput and product of the mills (Powell and McBride (2006)). In order to overcome the inherent shortcomings of the conventional tumbling mill models, outputs from DEM simulations can be used as one of the principal inputs into models for tumbling mills. Since DEM does not predict breakage, data from breakage experiments needs to be integrated into DEM based tumbling mill models.

The intention of this thesis is to demonstrate the use of DEM simulations and breakage experiments as the primary inputs into tumbling mill modelling. A pilot-scale SAG mill with actual operating conditions and particle size distributions is used for this demonstration.

1.2 Hypotheses

In this thesis, the following hypotheses will be tested:

1. Outputs from a DEM simulation of the tumbling mill can be used to inform tumbling mill models.
2. The distribution of the collision energies in the tumbling mill obtained from DEM simulations can be combined with data from breakage experiments in a population balance framework, that can be used to predict the breakage of the coarser particles in the tumbling mill.
3. DEM's prediction of the collision energy distribution and Bond work index laboratory tests can be used to predict the breakage of fine particles in the tumbling mill.

1.3 Objectives

The principal aim of this theses is to develop methodologies for using a combination of DEM simulations outputs and breakage experiments data as the key inputs into tumbling mill modelling. In order to achieve this, the main objectives of this thesis are:

1. Develop methodologies for extracting and analysing the data from DEM simulations, in such a way that it is useful to the prediction of the collision energy distribution in the tumbling mill.
2. Develop a methodology of using and analysing data from current breakage experiments to inform a DEM based tumbling mill model.
3. Develop a tumbling mill model that utilises inputs from DEM simulations, breakage experiments and Bond work index laboratory tests.
4. Test the developed tumbling mill model on a representative tumbling mill.

1.4 Thesis Layout

This thesis is divided into five chapters as follows:

The introductory chapter (1) gives the background of the thesis work, hypotheses, main objectives and thesis layout.

A review of the relevant literature that is concerned with the work in this thesis is given in chapter (2). In the literature review the DEM is reviewed in general, highlighting its particular application to tumbling mills. This is followed by a general review of tumbling mill modelling. An in-depth review

of selected articles that were of particular use and interest to this thesis work is given.

Chapter (3) is concerned with the theory behind the proposed tumbling mill model. This chapter details the mathematical framework of proposed model. Also in this chapter, the methodology for obtaining and analysing pertinent data from breakage experiments is laid out.

Chapter (4) presents the methodology employed to carry out the DEM simulations. Results obtained for the DEM simulation of a pilot-scale SAG mill, which is used to demonstrate the proposed tumbling model, are also presented. In presenting the DEM simulation methodology, a pilot SAG mill example is used.

A description of method that was employed to post-process results from Drop Weight Tester experiments for use the proposed tumbling mill model is presented in chapter (5).

The results obtained from the proposed tumbling models, based on the DEM simulations and breakage experiments of the pilot SAG mill are presented in chapter (6).

Conclusions and recommendations drawn from the work in this thesis are given in chapter (7).

Chapter 2

REVIEW OF LITERATURE

In this chapter, a review of the literature that was of importance to this thesis work is presented. A general review of DEM is initially given, followed by the application of DEM in comminution. A short general review of tumbling mill models is then given. An in-depth review of selected articles that were pertinent to this thesis work is also presented.

2.1 Introduction

When Cundall and Strack (1979), first proposed DEM in 1979, it was used to model the behaviour of soil particles, subjected to dynamic loading conditions. DEM has since been adapted to suit other applications involving granular media, including the following:

- Powders, fragmented and loose solids such as in powder mixing, powder flow, chutes, hoppers, tumbling mills .
- Soils which include soil stability, creep, avalanching.
- Brittle solids as in stability of tunnels, dams, foundations, stone structures, fracture and fragmentation and heave and throw in blasting.

The use of DEM in simulating tumbling mills was mainly pioneered by Mishra and Rajamani (1992). With the continuing increase in computing power and decrease in the cost of processors, DEM in milling has over the last one-and-half decades, grown from two-dimensional codes with hundreds of particles -represented as discs (Mishra and Rajamani (1992), Misra and Cheung (1999)) to three-dimensional codes capable of simulating hundreds of thousands of particles (Cleary (2001), Morrison and Cleary (2004)).

2.2 Overview Of The Discrete Element Method (DEM)

The DEM is a numerical modeling method that represents individual particles in a system, separately. In the DEM, the particles are modelled as discrete bodies that are allowed to displace and rotate independently (Cundall and Strack (1979)). The discrete elements only interact with each other at the contact points. When particles interact, they are allowed to overlap each other at the contact point. This is referred to as the soft contact approach (Govender et al. (2004), Sarracino et al. (2004), McBride et al. (2004a)) and is used to model plastic deformation at the interaction points.

Particles in DEM are usually modelled as either circular discs (two-dimensions) or spheres (three-dimensions). However, other researchers have managed to successfully model other shapes like ellipsoids, (Ting (1992)) and polygons (Ghabousi and Barbosa (1990)). Irregular shaped super particles can also be created by bonding several spherical or circular particles (Itasca Consulting Group (1999)). Also recently, Wang et al. (2005), used X-ray tomography imaging to represent real particles. Herbst and Nordell (2001), have showed that 3D DEM simulations give more accurate predictions than 2D DEM simulations.

The DEM simulation algorithm that is employed by many DEM researchers (Datta et al. (1999), Govender et al. (2004), Mishra and Rajamani (1992),

Morrison and Cleary (2004)), among others, generally involves the following steps that are repeated for every time step:

1. A contact search is carried out to determine the particles that are in contact.
2. Once the contacts are determined, the amount of overlap and relative velocities are calculated.
3. From the overlap, relative velocities and material properties, the net forces acting on the contacting forces are calculated using a suitable contact model.
4. Newton's second law of motion is then applied to all the particles to determine the resulting accelerations of each particle.
5. Numerical integration of the accelerations yield relative velocities and new positions for each particle. Commonly adopted integration schemes include, central difference (Mishra and Rajamani (1992), Itasca Consulting Group (1999)) and second-order predictor-corrector (Cleary (2001), Morrison and Cleary (2004)).

One of the basic and original assumptions incorporated into the DEM is that the time step is less than the critical time step (Δt_{cr}), which is the smallest duration for which a disturbance caused to the system by a contact event does not propagate beyond the contacting objects (Cundall and Strack (1979)). The simulation time step (Δt) used, is normally set to a fraction of the critical time step, Δt_{cr}. The value for Δt_{cr} is based on the characteristic natural frequency for a system determined as in equation 2.1 (Rajamani et al. (2000)).

$$\Delta t_{cr} = 2\sqrt{m/k_n} \tag{2.1}$$

The contact search, contact model, and contact model parameters form the three major components of a DEM simulation and are discussed in further detail in the following sub-sections.

2.2.1 Contact Search

Mishra (2003) observed that the contact search process for the simulation of N interacting particles involves an $N(N-1)/2$ -pair of contacts search problem. A DEM simulation involving $400,000$ particles will therefore check $79,999,800,000$ possible contacts in a single time step!

In order to reduce the time spent on searching, the entire working area is normally divided into square boxes (for 2-Dimensional simulations) or cubes (for 3-Dimensional simulations). This is refereed to as "Boxing" in DEM literature (Mishra (2003)). The size of the cube or box is set to the maximum particle size. Every box has an associated box list (also referred to as neighbor list). A box list for a particular box, is the list of all the particles whose circumscribing square (or box) share a common boundary with that particular box. For every box list only members of the box list are checked thereby reducing the search time. The box lists are generated for each box and updated after every time step. The presence of a particle to particle contact and contact overlap is calculated from the particles´ coordinates and sizes.

Several contact search algorithms are available for spherical particles. The more efficient DEM codes employ contact search algorithms of $O(N)$, (Morrison and Cleary (2004), Williams et al. (1996)). These requires N contact attempts for a problem involving N particles. Most DEM codes however use contact search algorithms of $O(N \log N)$, (Munjiza et al. (1993), DEM Solutions (2006)).

2.2.2 The Contact Model

In DEM a collision which is either particle-to-particle or particle-to -environment is represented by the contact model which dictates how the colliding partners react to the collision. There are various types of contact models available for use. These include the linear spring-and-dashpot (Cundall and Strack (1979), Morrison and Cleary (2004), Itasca Consulting Group (1999), and Datta and Rajamani (2002)); the modified linear viscous damping model (Sarracino et al. (2004)), the bi-linear (Sarracino et al. (2004)); the Hertz and Mindlin non-linear spring-and-dashpot (DEM Solutions (2006), Mishra and Murty (2001)); and the elastic perfectly plastic contact models (Mishra and Thornton (2002)). A review of the two most commonly used contact models namely the linear spring-and-dashpot and the Hertz and Mindlin models is given below.

2.2.2.1 The Linear Spring-and-Dashpot Model

The DEM analysis of tumbling mills to date has almost exclusively used the linear spring-and-dashpot model (Cundall and Strack (1979), Morrison and Cleary (2004), Datta and Rajamani (2002), McBride et al. (2004a)). This is most likely due to the fact that it is linear, has a long history and is well understood mathematically. It has been shown that the linear spring-and-dashpot model gives fairly accurate measurements of the charge motion and power draw of the mill (Cleary (1998), Datta et al. (1999), Venugopal and Rajamani (2001)).

In the linear spring-and-dashpot model, the interaction between the particles is assumed to be a linear spring and dashpot. The particle interactions are resolved into the normal and shear contact components as shown in figure 2.1.

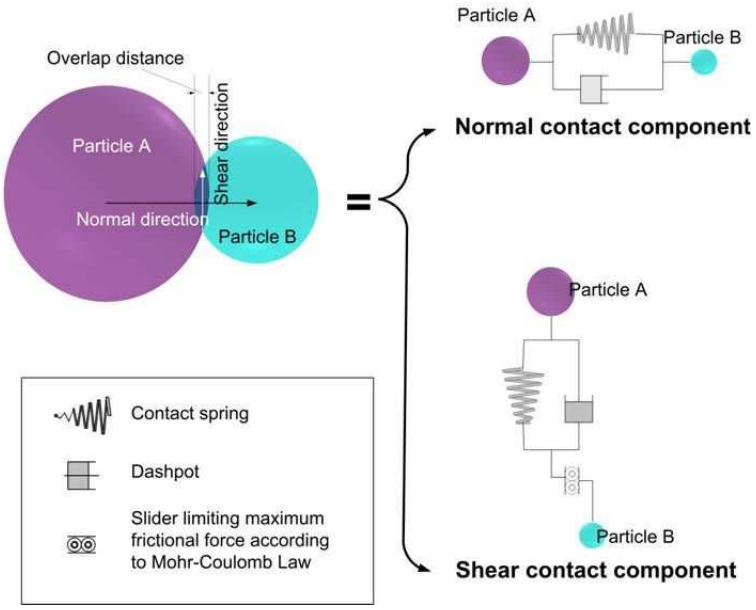

Figure 2.1: *Schematic of the linear spring-and-dashpot model, showing the use of numerical spring and dashpot assemblies to represent normal and shear components of the collision. After* McBride et al. (2004b).

The spring accounts for the repulsion of the particles as they collide and overlap. The dashpot acts to provide some damping and dissipate a specified portion of the particles' kinetic energy. The shear contact component is also represented by a spring, which represents the elastic tangential deformation of the contacting surfaces, in parallel with a dashpot, which again dissipates energy from tangential motion.

In the linear spring-and-dashpot contact model, the contact force in the normal direction, F_n, is calculated by

$$F_n = -b_n V_n + k_n U_n \tag{2.2}$$

Where:

- b_n is the normal damping constant
- U_n is the overlap of the contacting particles in the normal direction
- k_n is the normal contact stiffness

- V_n is the relative normal velocity of the particles

The contact shear force, F_s, is calculated incrementally from the contact shear force during the previous time step, F_s^0, as follows

$$F_s = F_s^0 + k_s \left(U_s - U_s^0\right) \tag{2.3}$$

Where:

- k_s is the shear contact stiffness
- U_s is the relative tangential displacement
- U_s^0 is the relative tangential displacement for the previous time step, Δt

The maximum shear force, $F_{s,max}$, is limited according to the mohr-coulomb law which means that the maximum shear force will not exceed the product of the coefficient of friction, μ, and the normal contact force, F_n (equation 2.4).

$$F_{s,max} = \mu F_n \tag{2.4}$$

The linear spring-and-dashpot model is represented mathematically by the governing differential equation (Sarracino et al. (2004)) in equation 2.5

$$\begin{aligned} m\ddot{U} &= -b\dot{U} - kU \quad for\ (U > 0) \\ m\ddot{U} &= 0 \quad for\ (U \leq 0) \end{aligned} \tag{2.5}$$

where:

- m is mass of the particles
- $\ddot{U}$ is acceleration
- $\dot{U}$ is velocity
- U is contact overlap

Although the linear spring-and-dashpot model is widely used, it has one major problem: it is unphysical as a model for particle collisions. It is however well suited for particles in fluids. The model assumes that viscous damping will be maximum as the particles are coming into contact and also as the particles are about to separate. This is not what is expected to happen - damping should be a minimum when the particles first come into contact and also as the particles rebound. It is due to the rather unphysical nature of the linear spring-and-dashpot model that it is hypothesized (Sarracino et al. (2004)) that although it may be adequate for charge motion and power draw predictions, it wouldn't generate accurate impact energy spectrum predictions.

2.2.2.2 The Hertz-Mindlin Model

The Hertz-Mindlin contact model provides an alternative, to the more common linear spring-and-dashpot model. The Hertz-Mindlin model is more detailed and realistic than the spring-and-dashpot model. Unlike the linear contact model, in the Hertz-Mindlin contact model the normal spring stiffness, k_n, varies according to the amount of overlap, U_n, between the contacting particles, in accordance with Hertzian contact theory developed by Hertz in 1882 (Johnson (1985)).

In the Hertz-Mindlin contact model, the normal interaction is represented by Hertz theory (Johnson (1985)) as shown in equation 2.6.

$$F_n = \frac{2}{3} P_{max} \pi a^2 \tag{2.6}$$

Where:

- P_{max} is the maximum pressure at the point of contact and
- a is the radius of the area covered by the contact and is given by

$$a = \left(\frac{3 P_{max} R^*}{4 E^*} \right)^{\frac{1}{3}} \tag{2.7}$$

In equation 2.7, R^* and E^* are the reduced radius of the contacting particles and Young's modulus, respectively. R^* and E^* are obtained from the following relations:

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2} \tag{2.8}$$

$$\frac{1}{E^*} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2} \tag{2.9}$$

Where:

- ν is Poisson's ratio
- The subscripts 1 and 2 represent the first and second contacting partners.

The Force-displacement relation in the normal direction is calculated from

$$F_n = -k_n U_n^{\frac{3}{2}} \tag{2.10}$$

Where:

- U_n is the contact overlap in the normal direction, and is given by

$$U_n = \frac{a^2}{R^*} \tag{2.11}$$

- and the normal contact stiffness k_n, is given by

$$k_n = \frac{E^* \sqrt{2R^*}}{3\left(1-\nu^2\right)} \tag{2.12}$$

The shear contact force in the nonlinear contact model is obtained in the same way as for the linear contact model, using the force-displacement relation in equation 2.3. However, the theory developed by Mindlin and Deresiewicz (1953), is widely applied to determine the value for the shear contact stiffness, k_s, as shown below.

$$k_s = \frac{E^* \sqrt{2R^*}}{(1+\nu)(2-\nu)} \sqrt{U_n} \tag{2.13}$$

The governing differential equation, for a system without damping, is therefore written as

$$m\,\ddot{U} = -k\,U^{\frac{3}{2}} \;\; for \;\; (U < 0) \tag{2.14}$$

When damping is incorporated, equation 2.14, generally takes the form

$$m\,\ddot{U} = -\zeta\,\dot{U} - k\,U^{\frac{3}{2}} \tag{2.15}$$

where:

- ζ is an arbitrary parameter

There are several variations of the non-linear spring-and-dashpot contact model in the literature. These variations differ only in the form of the arbitrary parameter, ζ, which however is always a function of the damping constant, b.

- The *DEM Solutions*' $EDEM^{TM}$, DEM software DEM Solutions (2006) employs ζ of the form

$$\zeta = 2\left(\frac{5}{3}\right)^{\frac{1}{2}} b\left(m^{*}2\,E^{*}\left(R^{*}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}} U^{\frac{1}{4}} \tag{2.16}$$

- Velusami (Mishra (2003)) determined ζ from experiments, as

$$\zeta = b\,U^{\frac{3}{2}} \tag{2.17}$$

- Tsuji et al. (1993) determined ζ heuristically, by incorporating a coefficient of restitution dependent constant, a, as

$$\zeta\,a = a\,(mk)^{\frac{1}{2}} U^{\frac{1}{4}} \tag{2.18}$$

2.2.3 Contact Model Parameters

The contact models discussed earlier utilise material properties and material interaction parameters, like the spring stiffness - or Young's modulus and poisson's ratio, damping constant - or coefficient of restitution and coefficient of friction. Chandramohan (2005) observed that, *"Applying measured material interaction properties in DEM rather than fitting or estimating them, in the contact models that govern the inter-particle contact - force laws, can contribute to the overall reliability and accuracy in predicting the motion of griding media in rotary mills".*

Ideally, it would be desirable to use the actual parameters for the particles' materials. However, in reality, it is not easy to obtain parameters based on

reliable experimental data. The current DEM measurements normally uses contact model parameters that are carefully selected (or fitted) based on a number of assumptions.

The following sections discuss the material properties and material interaction parameters that are needed in DEM simulations.

2.2.3.1 Damping Constant and Coefficient of Restitution

The damping constant and coefficient of restitution are important for the prediction of charge motion and energy distribution because they represent measures of the energy that is lost during a collision. The coefficient of restitution, ε, is defined as the ratio of the relative velocities of colliding bodies just before contact, to the relative velocities just after the collision as given in equation 2.19.

$$\varepsilon = \frac{V_f}{V_i} \tag{2.19}$$

With: V_f and V_i representing the final relative velocity and initial relative velocity respectively

The relationship between the damping constant, b, and the coefficient of restitution, ε, as shown by Corkum (1989), is given as

$$b = -\frac{2\,\ln(\varepsilon)\,\sqrt{m^*\,k}}{\sqrt{\ln^2(\varepsilon) + \pi^2}} \tag{2.20}$$

where

- m^* is the reduced mass of the two contacting particles with masses m_1 and m_2 and is given by:

$$\frac{1}{m^*} = \frac{1}{m_1} + \frac{1}{m_2} \tag{2.21}$$

The literature reveals that the coefficient of restitution can be experimentally determined for simple impact situations like a single ball hitting a wall. For example, Mishra and Rajamani (1992) used a video camera to record the rebound heights of steel balls falling on a metal plate, to represent ball-wall collisions. They then calculated ε from the initial and rebound heights, h_i and h_r as shown in equation 2.22.

$$\varepsilon = \sqrt{\frac{h_r}{h_i}} \tag{2.22}$$

The experiments carried out by Mishra and Rajamani (1992), for ball-ball collisions, involved pendulums where one ball was hung stationary while the other was swung from a pre-determined height to hit the stationary one. The rebound height of the stationary ball was recorded using a video camera. The coefficient of restitution, ε, for this setup was calculated from the initial height, h_i, and rebound height, h_r, as shown in equation 2.23.

$$\varepsilon = 2\sqrt{\frac{h_r}{h_i}} - 1 \tag{2.23}$$

The coefficient of restitution, however is a material interaction property and as such depends on the actual environment of the collision, Johnson (1985). It should therefore be ideally measured under the same conditions as in the mill, Mishra and Rajamani (1992). The situation where a ball in flight hits another stationary suspended ball is highly unlikely to occur in the mill. Chandramohan (2005) used the experimental setup in figure 2.2 to investigate the behaviour of two spheres colliding in free-fall conditions - to enable an accurate unconstrained measure of particle interaction properties over a full range of collision angles.

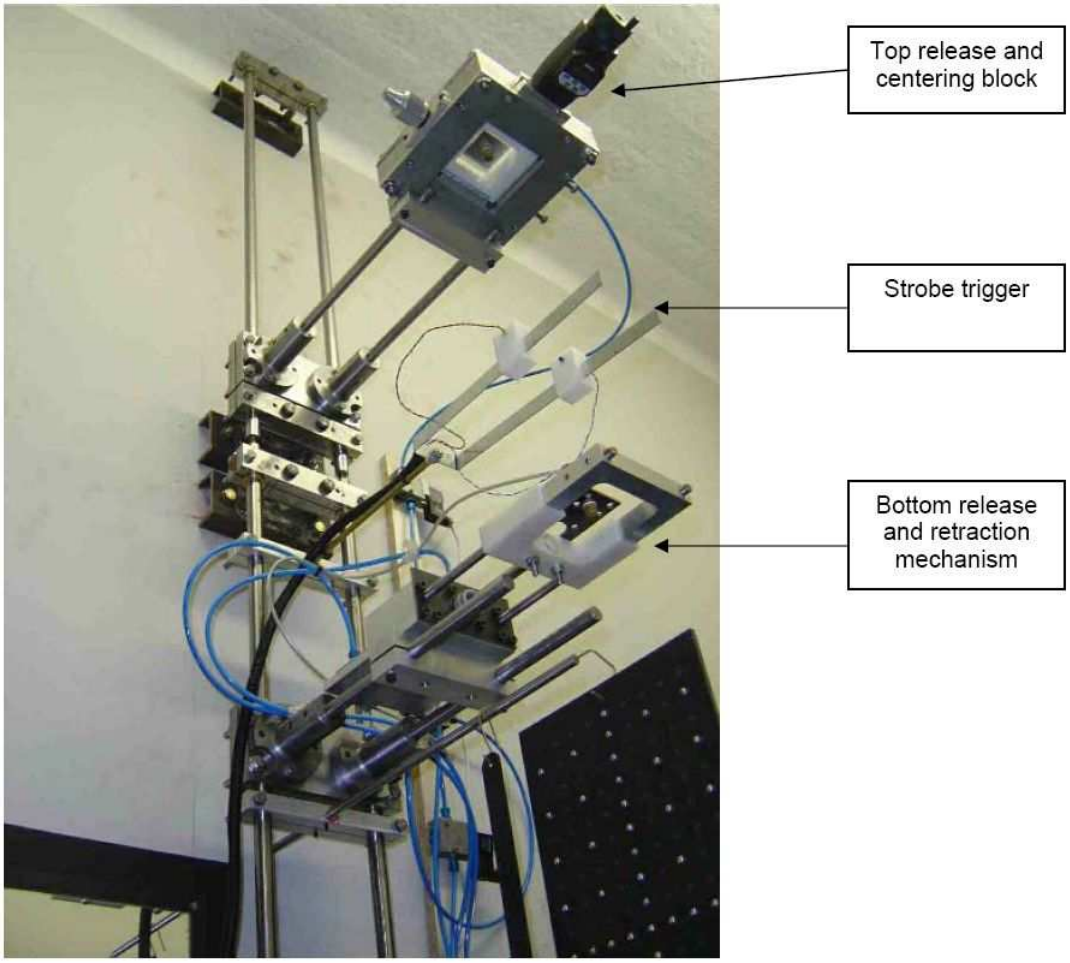

Figure 2.2: *Setup of experiment to measure coefficients of friction and restitution. After Chandramohan (2005).*

2.2.3.2 Contact Stiffness, Young's modulus and Poisson's ratio

The forces resulting from the overlap at the contact point is a function of the contact stiffness. The choice of the value of the contact stiffness is therefore key to the accuracy of DEM measurements. The contact stiffness can be measured by analysing the force-displacement relationship obtained from particle collision experiments. Mishra and Rajamani (1992), Mishra and Murty (2001) used an Ultra-Fast Load Cell (UFLC) to obtain the force-displacement relationships. In the UFLC experiments, a steel ball was dropped on to the flat end of a steel rod fitted with ultrafast strain gauges, which measured the strain waves generated by the impact. The

contact stiffnesses were calculated from the measured impact forces and displacements as shown in table 2.1, below.

Ball Mass (kg)	Fall Height (m)	Max. Force (N)	Max. Disp. (mm)	Stiffness (MN/m)
0.0926	0.3	8109	0.075	140
0.2523	0.3	11066	0.119	135
0.6468	0.3	24872	0.138	126

Table 2.1: *Result of UFLC experiments to determine material stiffness. After Mishra and Rajamani (1992).*

In the Hertz-Mindlin contact model, the stiffness, k, is expressed as a function of Young's modulus, E, and Poisson's ratio, ν as in equations, 2.12 and 2.13. E and ν are related via equation 2.24.

$$E = 2(1+\nu)G \tag{2.24}$$

Where:

- G is the elastic shear modulus

The normal practice when using linear contact models in DEM simulations is to choose the contact stiffness constant (or Young's modulus) in such a way that the fraction of the maximum contact overlap is a small fraction of the diameter of the colliding particles. In their work to investigate power draw in an experimental two-dimensional mill, van Nierop et al. (2001) reported that changes in the normal stiffness had little effect on the calcu-

lated power (table 2.2), provided that the corresponding time step was also changed accordingly.

Normal stiffness (Nm^{-1})	Timestep (μs)	Power (W)
$4x10^5$	10	26.42
$4x10^6$	10	26.85
$4x10^7$	3.5	26.02
$4x10^8$	1	27.47

Table 2.2: *Variation of power draw with normal stiffness (van Nierop et al. (2001))*

2.2.3.3 Coefficients of Friction

The coefficient of friction dictates the initialisation of slip between two particles experiencing tangential interaction according to equation 2.4. The coefficient of friction may be an important parameter in DEM simulations as suggested by van Nierop et al. (2001) who reported a variation of power draw with coefficient of friction. However, Mishra (2003) observed that "*the coefficient of friction is difficult to measure, and it may vary during grinding*".

2.3 Application Of DEM To Milling

Since its inception and successful adaptation to milling problems, DEM has been widely used to simulate mills. The ultimate aim is to use DEM as a design and diagnostic tool capable of predicting important milling parameters such as charge motion, power draw, liner wear, impact energy spectrum and particle breakage. It has to be noted here, that most research in the application of DEM to milling has been concentrated on charge motion and power draw predictions.

2.3.1 Charge Motion and Shape

In the tumbling mill, the material to be ground enters the mill through one end and leaves via the other end after being reduced in required size. The mill is filled with steel balls (grinding media), which break the material as the mill is rotated. The steel balls and material to be ground which may be wet or dry, is what is termed the mill charge. Charge motion is the behavior of the mill charge, as the mill operates. Charge motion is very important for the optimization of the milling process.

The mill's charge is characterised by its Centre of Circulation (CoC), equilibrium surface, shoulder, toe, en mass charge, angle of repose and Centre of Mass (CoM), Powell and McBride (2004). These regions and concepts were defined by Powell and McBride (2004), Powell and Nurick (1996) (also see figure 2.3) as follows:

- En mass charge - the region of the charge that is not in free-flight.
- Centre of Circulation (CoC) - The axis about which all the particles circulate.
- Equilibrium surface - the surface that separates the en masse ascending charge from the en mass descending charge.
- Head - the highest point that the charge reaches in the mill.
- Departure shoulder - the highest point at which the charge starts to leave the mill shell.
- Impact toe - the region where falling particles in free-flight lands on the mill shell.
- Bulk toe - the region on the shell where the descending cascading en mass charge turns as it starts moving in the mill rotation direction.

- Centre of Mass (CoM) - the point in the charge at which if all the mass were concentrated would exert a torque about the centre, equivalent the mill torque.
- Angle of repose - for a non-centrifuging charge. The angle formed between the line connecting the mill centre and the CoC and the vertical, from the mill centre.
- Rate of circulation - the number of revolutions about the CoC, that a particle makes per mill revolution.

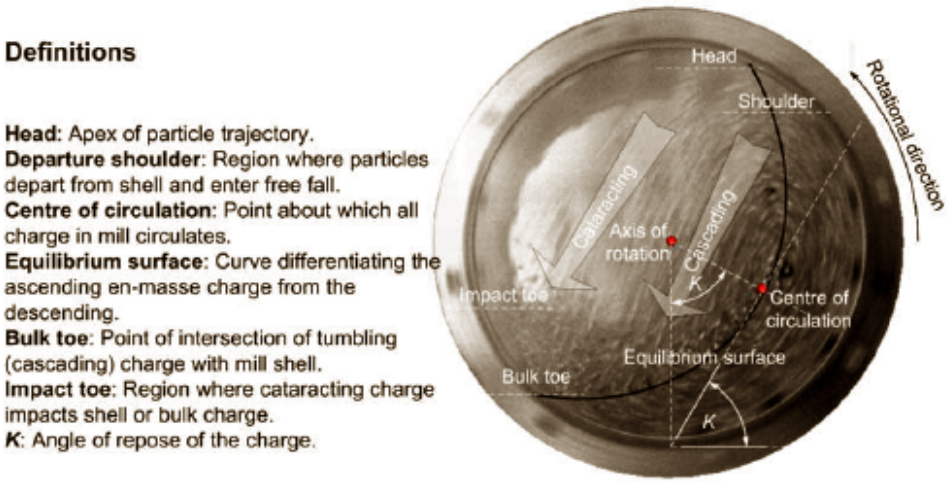

Figure 2.3: *Definitions of the regions of the charge. After Powell and McBride (2004).*

Powell and McBride (2004) also developed methods to not only analytically but also mathematically derive the regions defined above. These methods can be applied to either experimentally and DEM simulated particle trajectory information. The prediction of charge motion using DEM is relatively easy to perform since DEM simulates at particle level. Charge motion prediction involves tracking of the motion of each individual particle through the mill. Charge motion predictions using DEM have been widely done (Cleary (2001, 2004, 1998), Powell and McBride (2004), Mishra and Rajamani (1992), Misra and Cheung (1999), Rajamani et al. (2000), Venugopal and Rajamani (2001)) with fairly accurate results.

The charge shape is the instantaneous snap-shot of the charge in motion. Figure 2.4 from the work of Misra and Cheung (1999) shows the charge shape of a $2m$ diameter mill taken every $1/4$ cycle of a revolution. From these results (figure 2.4), Misra and Cheung (1999) observed that the charge shape appeared to repeat every half revolution of the mill. This was probably because the overall rate of circulation of the charge about the CoC was two revs per mill rev.

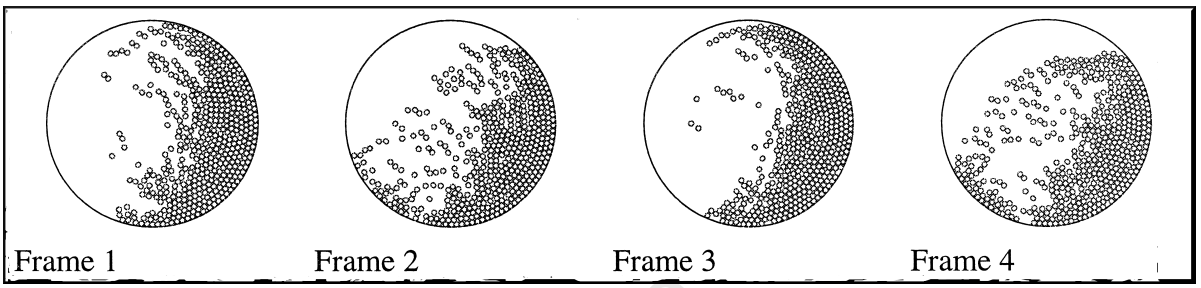

Figure 2.4: *Charge shape at every $\frac{1}{4}$ cycle of a revolution. After Misra and Cheung (1999).*

In order to validate the results of DEM prediction of charge motion, the DEM results are usually compared to experimental observations. The commonly used method is to build a laboratory mill with exactly the same dimensions as the one used in the DEM simulations. The experimental mill is then operated under the same conditions as the DEM mill and a high speed digital video camera used to capture the dynamic behaviour of the laboratory mill. "Snap shots" of the DEM charge prediction and the laboratory mill are then compared to check the accuracy of the DEM simulations. The results obtained from the DEM simulations are usually in good agreement with the expected experimental observations, as can be seen from an example in figure 2.5, from the work of Venugopal and Rajamani (2001).

A more detailed and improved DEM simulations validation method based on the technique of Powell (1993) was recently suggested by Govender et al. (2004). In their work, Govender, et. al. used an X-ray angiographic equipment to experimentally track particles in a small lab-scale mill. From the

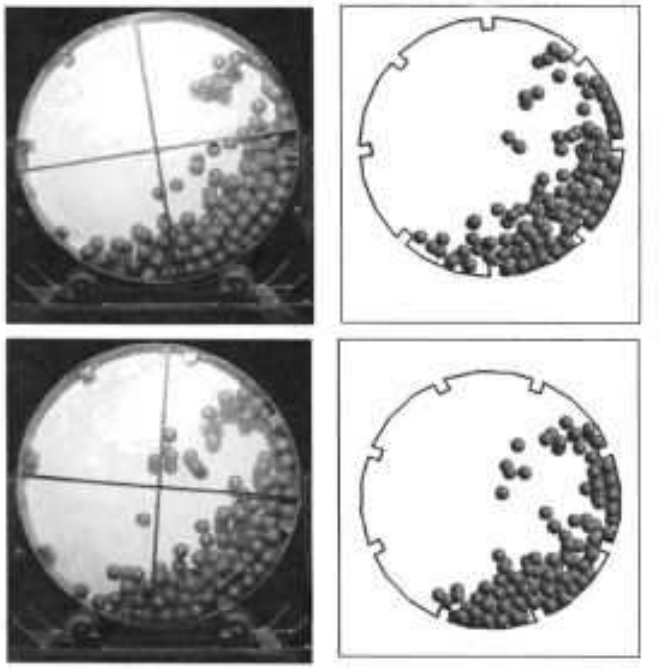

Figure 2.5: *Comparison of actual and simulated charge profiles for a 90-cm lab mill at 50% critical speed and a volume filling of 20% Venugopal and Rajamani (2001)*

X-ray images, Govender et al. (2004) re-constructed the 3D trajectory motion of the tracked particles, which was compared to DEM simulations. A binning technique was used. In this technique, the cross-section of the mill is divided into fine grids. Apart from the charge trajectory positions, other charge motion parameters, like the Center of Circulation (CoC) and equilibrium surface (as defined by Powell and Nurick (1996), were calculated and compared.

2.3.2 Power Draw

A substantial potion of a mineral processing plant's operating costs is expended in tumbling mills. The estimated energy consumption for grinding rock to fine product size is $10 - 30\,kWh/t$.

Traditionally, researchers (includes Morrell (1992), Moys (1993), Harris and Arbiter (1982), Bond (1961a,b)) have used empirically derived models to predict mill power. Recently Govender (2005) suggested a power model

based on the study of the trends of the CoC and CoM obtained from experimentally determined charge motion.

The motion of the charge in the mill is as a result of the power drawn from the electric motor. It follows that prediction of a mill's power draw using DEM is derived from the prediction of the charge motion. Energy is lost within the mill as the particles collide, which is represented by the numerical damping (in the dashpot) in the contact model. One of the common methods (Datta et al. (1999), Venugopal and Rajamani (2001)) of predicting power draw using DEM involves recording the energy lost at every collision. The collision energies are then summed and divided by the simulation time to get the power draw of the mill. Powell and McBride (2004) and Powell and Nurick (1996) proposed the use of the torque-arm method where the Centre of Mass (COM) was multiplied by the displacement from the centre of the mill rotation to obtain the total equivalent torque. In another method (Cleary (2001) and van Nierop et al. (2001)) the forces of all the particles in contact with the mill shell are used to calculate the torques. These torques are then integrated over all the contacts (particle contacts with the liner) and time steps to obtain the total equivalent torque which is then used to calculate the power draw as in equation 2.25.

$$P = 2\,\pi\,N\,T \tag{2.25}$$

where:

- P is the mill´s power draw
- N is the angular speed of the mill in $revs/s$
- T is the total equivalent torque

2.3.3 Impact Energy Spectrum

When particles (rocks and steel balls) collide inside the tumbling mill, a certain amount of energy is absorbed by the colliding particles and is responsible for particle breakage. In DEM simulations, the dissipated energy is modelled as the energy lost via the dashpots. The energy lost at a collision is calculated from the force-displacement relationship, equation 2.26.

$$E_{Lost} = F_b \dot{U} \, \Delta t \tag{2.26}$$

where:

- E_{Lost} is the energy lost at a collision
- F_b is the force experienced by the dashpot
- $\dot{U}$ is the relative velocity of the contacting partners
- Δt is the time step

Using $F_b = b\dot{U}$, from equation 2.2, for the linear spring-and-dashpot contact model, the energy lost during a contact is given by equation 2.27.

$$E_{Lost} = b\dot{U}^2 \Delta t \tag{2.27}$$

where:

- b is the damping constant

While for the non-linear Hertz-Mindlin model, $F_b = \zeta\dot{U}$, which results in energy lost being generally given by equation 2.28.

$$E_{Lost} = \zeta \dot{U}^2 \Delta t \tag{2.28}$$

where:

- ζ is an arbitrary parameter that is a function of b

The energy lost during a contact, E_{Lost}, is normally referred to as the collision energy or contact energy in DEM literature.

The distribution of contact energies in the tumbling mill is usually given by the impact energy spectrum. The impact energy spectrum can be depicted as a histogram plot of the number of contacts versus the average contact energy as shown in figure 2.6.

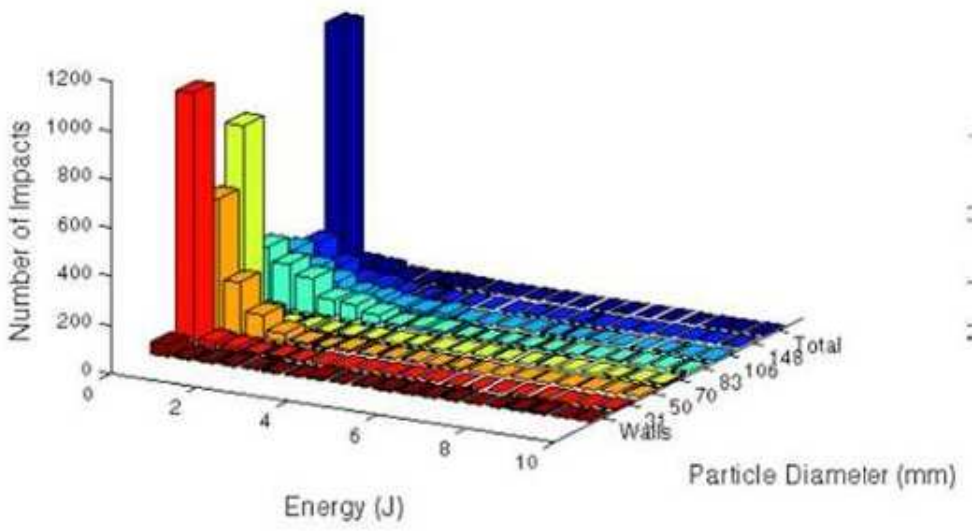

Figure 2.6: *Impact energy spectra for different size fractions. After Powell et al. (2003).*

In order to construct an energy spectra like the one shown in figure 2.6, the X-axis (or full range of collision energies) is divided into a number of equally spaced intervals (or bins). The number of collisions shown on the Y-axis is actually the number of all the collisions, where the energy lost is within a particular interval (or bin). The collision energy shown on the X-axis is usually the mid-points of the energy intervals (or bins).

There are many different ways of depicting energy distribution in tumbling mills presented in the literature. The type of energy spectra shown in figure 2.6, is the most commonly used and easy to interpret as it shows the frequency of collisions and the corresponding average energy. Other researchers, however opt to use line plots on a log-log (logarithm scaling on both the X- and Y- axes) scale (Cleary (2001), Morrison and Cleary (2004)). A different way of showing the distribution of energies in a tumbling mill is to divide the mill into equally spaced bins. A binning algorithm is then utilised to allocate contact energies according to the contact positions. This results in a spatial distribution of the contact energies as shown in figure 2.7. Energy distribution can also be shown with respect to time as shown in figure 2.8. Morrison and Cleary (2004) used this type of energy distribution plot to show that DEM predicts the expected rise in collision energy as particles pass through the toe of the charge.

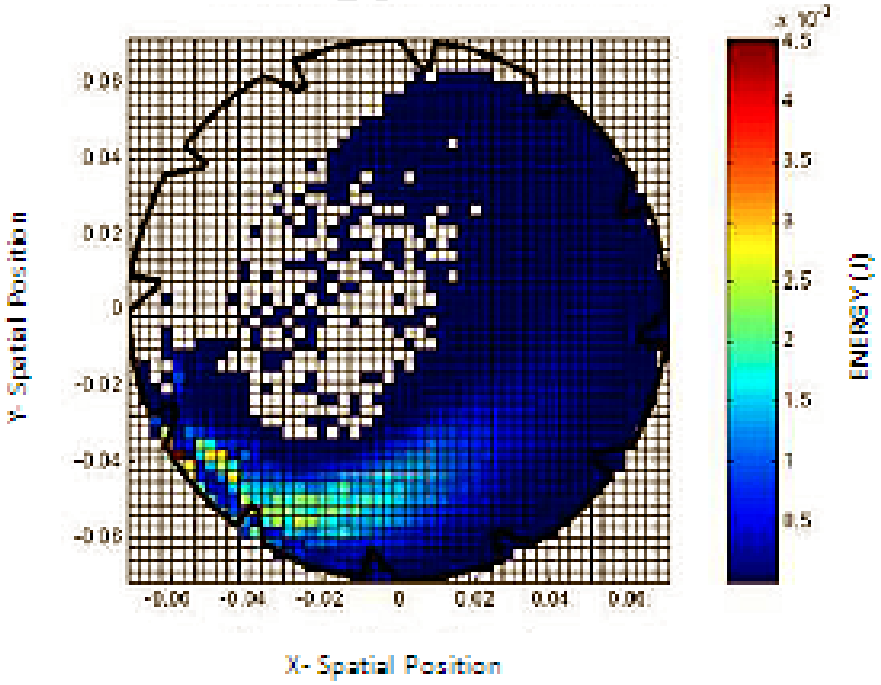

Figure 2.7: *Spatial distribution of contact energies. After Sarracino et al. (2004).*

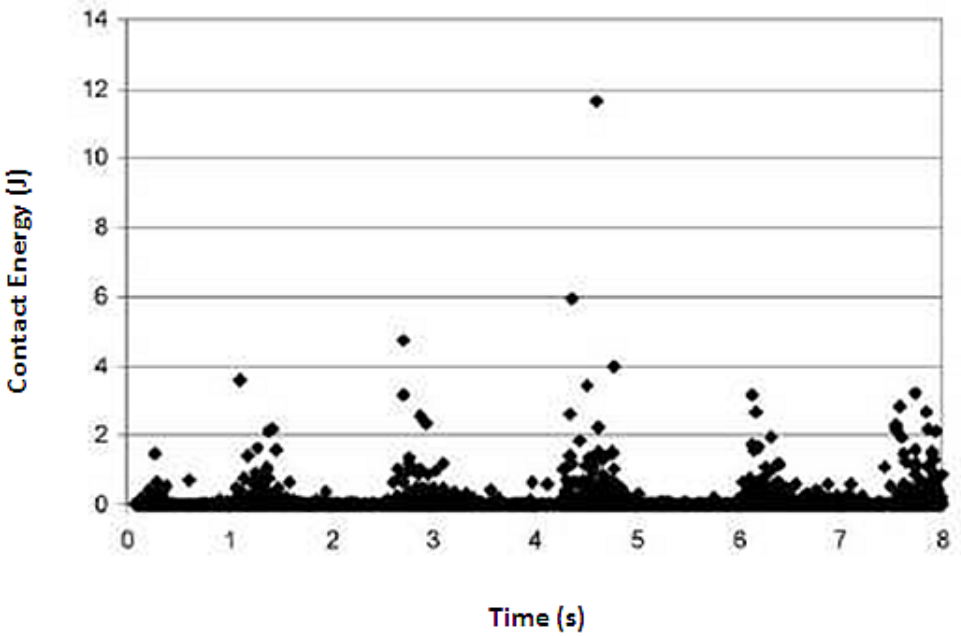

Figure 2.8: *Time distribution of contact energies. After Morrison and Cleary (2004).*

2.4 Overview Of Tumbling Mill Models

Comminution models are used for design and the optimisation of the performance of industrial comminution devices. Traditionally, laboratory-scale mills, operated under the same conditions as the industrial-size mill, are used to obtain data that is used in the model to scale-up to the larger mill.

The JKMRC mineral processing handbook (Napier-Munn et al. (1999)) reports that the earliest comminution models were developed by Rittinger (1867) and Kick (1883). These two models had major weakness, which included among other things, consideration of only particle surface (Rittinger) or particle volume (kick), and the use of cubes in developing the model theories. However it was not until 1952 when Bond (1952) introduced his *"third theory of comminution"* that they were questioned. In his model, Bond used an empirically determined factor (work index) to predict the product as shown in equation 2.29.

$$W = 10\,W_i \left(\frac{1}{\sqrt{P_{80}}} - \frac{1}{\sqrt{F_{80}}} \right) \tag{2.29}$$

where:

- W is the work input
- F_{80} and P_{80} are the sizes at which 80% of the feed and product passes, respectively
- W_i is the work index (work required to grind a feed of infinite size to the given P_{80})

The Bond theory is still used at present although it is restricted mainly to ball mills and a top feed size of below $25mm$. Criticism of the Bond theory include, the use of only one index correction factor, the description of the entire product size distribution using only one P_{80} size, Datta and Rajamani (2002).

Napier-Munn et al. (1999) reported that comminution models are generally categorized into two types, namely; black box models and fundamental models. In the black box models, the final product is predicted while the comminution device itself is treated merely as a "black box". Fundamental models on the other hand, attempt to characterize the underlying breakage mechanism at the fundamental levels to predict the product.

2.4.1 Black Box Models

There are two main variations of black box models used for tumbling mills. The population balance model (Epstein (1947)) dates back as far as 1947, while the perfect mixing model was formulated by Whiten (1976).

2.4.1.1 The Population Balance Model

The population balance model assumes a constant number of impacts per unit time for each size fraction. With the number of impacts per unit time, being kept constant, the model consequently assumes that the rate of breakage of particles of a certain size fraction, is solely a function of the mass of the size fraction in the mill.

Based on the material balance around a single size fraction in steady state (i.e. $Accumulation = 0$), equation 3.2,

$$Feed\,in + Breakage\,in = Product\,out + Breakage\,out \tag{2.30}$$

the general population balance equation is written as in equation 2.31.

$$p_i = f_i - k_i s_i + \sum_{j=1}^{i-1} b_{ij}\, k_j\, s_j \tag{2.31}$$

Where:

- the subscript i, represents the i^{th} size range
- f is the feed into the mill
- k is the rate of breakage constant
- s_i is the mass of the i^{th}size range
- b_{ij} is the breakage function (fraction of size range, j, that breaks into size range, i)

2.4.1.2 The Perfect Mixing Model

The perfect mixing model (Whiten (1976)) is very much like the population balance model. It is also based on the material balance equation shown

earlier in equation 3.2. The perfect mixing model is as shown in equation 2.32.

$$f_i + \sum_{j=1}^{i} a_{ij}\, r_j\, s_j = p_i + r_i s_i \tag{2.32}$$

Where:

- a_{ij} is the appearance function (breakage function)
- r_j is the rate constant

In the perfect mixing model, the particle is allowed to break into the original size fraction, i.e. some of the original particle can remain in the same size fraction after breakage. Apart from the range of the appearance function, the major distinction of the perfect mixing model from the population balance model is that it assumes a perfectly mixed charge. For each size fraction, the product rate is therefore related to the mill contents via equation 2.33.

$$p_i = d_i\, s_i \tag{2.33}$$

Where:

- d_i is the discharge rate of the product

In normal practice, the feed and the product are readily measured from the pilot-scale mill. The appearance function is normally known or assumed. This leaves the rate constant and the mill contents. The mill contents are particularly difficult to determine, since the mill is treated as a “black box”. From equation 2.33, the mill contents can be determined from equation 2.34.

$$s_i = \frac{p_i}{d_i} \tag{2.34}$$

Substituting equation 2.34, into equation 2.32, and rearranging yields equation 2.35.

$$\frac{r_i}{d_i} = \frac{1}{p_i\,(1 - a_{ii})} \left\{ f_i + \sum_{j=1}^{i-1} \left[\frac{a_{ij}\, r_j\, p_j}{d_j} \right] - p_i \right\} \tag{2.35}$$

The ratio r_i/d_i can therefore be calculated for each size fraction.

2.4.2 Fundamental Models

Fundamental models seek to characterise or model the underlying breakage mechanism at the fundamental (particle) level rather than just the product. Fundamental models enable the physics of the process and the actual mechanical environment of the comminution device to be incorporated in the breakage process. At the heart of the current fundamental models for tumbling mills, is the DEM. DEM simulations of the tumbling mill are used in conjunction with single particle breakage data (Morrison and Cleary (2004), Datta and Rajamani (2002)) or Computational Fluid Dynamics (CFD) and discrete grain breakage models in High Fidelity Simulations (HFS) (Herbst and Nordell (2001)) to model breakage. The work by Datta and Rajamani (2002) and Morrison and Cleary (2004) is reviewed in detail in section 2.5.

2.5 A Review of Selected Articles

In this section, selected articles are reviewed. The criteria used for selection was mainly relevance and relatedness of the work to this thesis. For each article, the methodologies used and the primary outcomes are first presented, followed by a discussion by the author of this thesis.

2.5.1 Using DEM to model ore breakage within a pilot scale SAG mill

Morrison and Cleary (2004)

In this article, Morrison and Cleary reported on work to utilise data derived from collision histories of particles in DEM simulations to predict breakage. DEM simulations of a 1.8m x 0.6m pilot SAG mill and a 0.5m slice of a full scale 36ft SAG mill were performed. The operating conditions for the two mills were as follows:

	Pilot Mill	**Full Size Mill**
Size	1.8m x 0.6m	11m x 1.8m
Ball Load	5%	10%
Volumetric Filling	39%	30%
Mill Speed (% critical)	76%	78%

Table 2.3: *Operating conditions of pilot and full size mill that was simulated by Morrison and Cleary (2004)*

The linear spring-and-dashpot contact model, described earlier in subsection 2.2.2.1, was used to dictate the behavior of colliding particles. For the representation of the rock in SAG mill charge, Morrison and Cleary used spheres of sizes ranging from a top size of 122mm to a cut-off bottom size of 6mm. It is necessary to 'cut-off' the very small particle sizes in the DEM model in order to keep the total number of particles to a reasonable number[1]. The DEM cut-off size was chosen such that $> 90\%$ of the actual charge is represented in the model.

As mentioned in section 2.4, results from tests on pilot-scale mills are often used to carry out scale-up designs of full-scale mills. In this work, Morrison and Cleary included a very interesting comparison between results from

[1]Due to computational limitations, DEM codes have a finite maximum number of particles

DEM simulations of the pilot mill and full-size mill. The impact energy spectra for the normal and shear collisions were shown for the two mills. Apart from the expected difference in the maximum collision frequencies and energies, the spectra for the full-size mill had a more pronounced peak than the one for the pilot mill (figure 2.9). They reported that based on the traditional power based mill scaling relationship[2], the scaling from the pilot mill to the full-size mill was expected to be "154 *times*" the power draw.

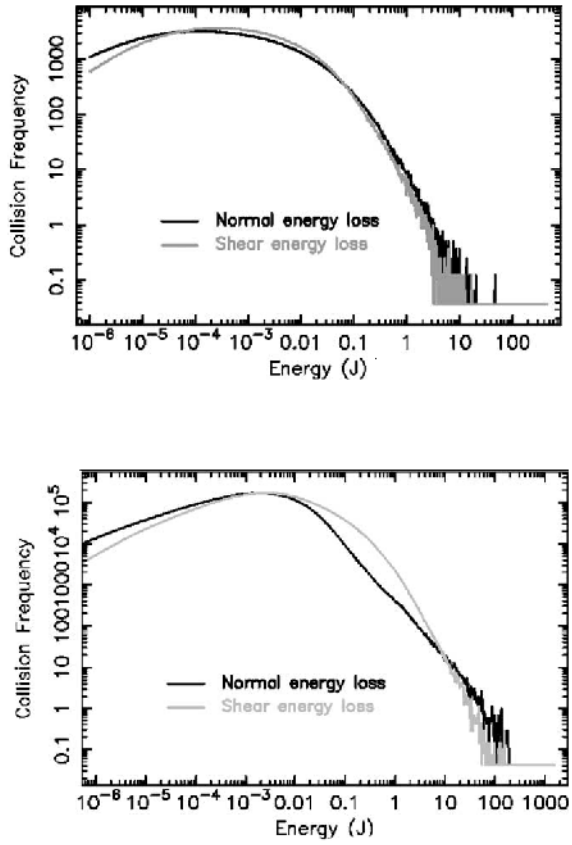

Figure 2.9: *Comparison of impact energy spectra of the pilot SAG mill (top) and full-size SAG mill (bottom). Morrison and Cleary (2004).*

The degree of breakage of a particle after a collision event is often measured using the t_{10}parameter which is the percentage of the original particle that passes one-tenth ($\frac{1}{10}$) of the original size and is given by:

[2]Scaling is "linear with mill length and diameter to the power of 2.5"

$$t_{10} = A\ \{1 - \exp(-b\, E_{cs})\} \tag{2.36}$$

where:

- A is the maximum degree of breakage
- b is a parameter of hardness
- E_{cs} is the specific input comminution energy

The energies needed to obtain degrees of breakage, t_{10}, equal to 1, 5 and 30 for an ore with a density of $2800\,kg/m^3$were calculated for a range of particle sizes as shown in table 2.4.

Nominal diameter (mm)	Mass (kg)	Energy (J) for 0.1 kWh/t	Energy (J) for $t_{10} = 1$	Energy (J) for $t_{10} = 5$	Energy (J) for $t_{10} = 30$
200	11.5	4140	363	1894	16,475
160	5.825	2097	184	960	8345
83	0.855	308	27	141	1225
35	0.0622	22	2	10	89
27	0.0303	11	1	5	43

Table 2.4: *Energy and breakage requirements for a range of particles. Morrison and Cleary (2004).*

Results obtained from the DEM simulations of the pilot SAG and full-size mills (figure 2.9) showed that most collision energies were much smaller than the input energies required to get even a small breakage degree of $t_{10} = 1$.

Furthermore individual particles were tracked to register their trajectory information and collision histories as shown in figure 2.10, for a 35mm and 27mm ore particle. Again these results showed that the particles experienced no or very few collisions that would result in any considerable breakage.

The predictions from DEM showed that the majority of collisions involved low energies. This led Morrison and Cleary to suggest that incremental

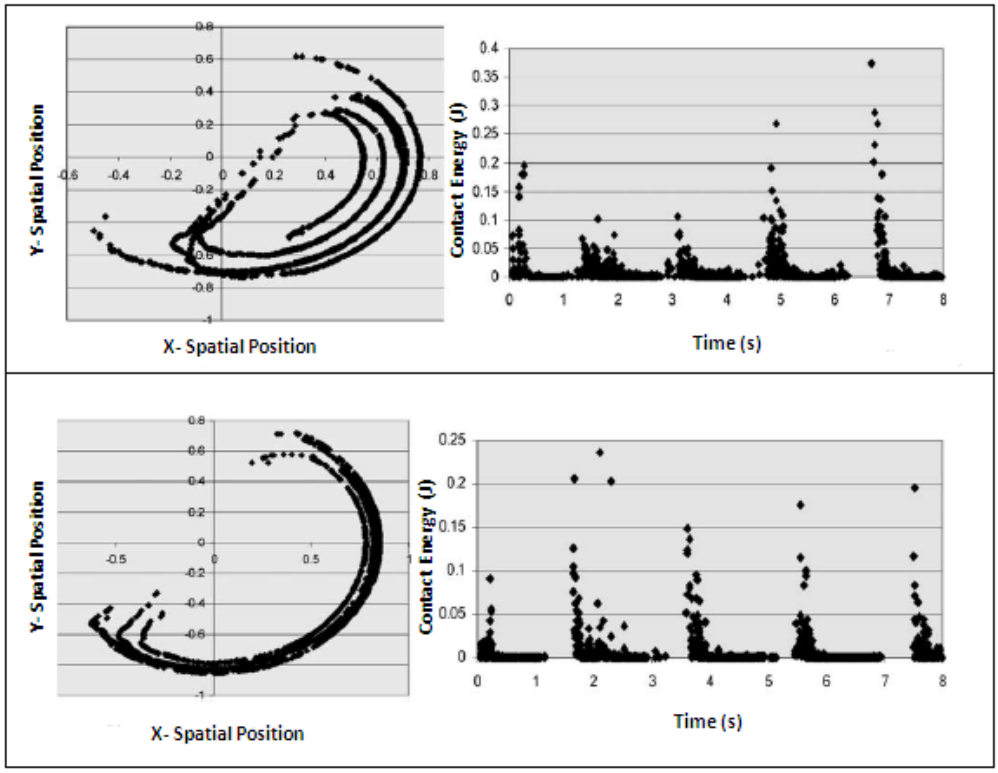

Figure 2.10: *DEM simulation trajectory and collision distribution for a 35mm (top) and (27mm) ore particle. Morrison and Cleary (2004).*

damage of particles, Briggs and Bearman (1995), contribute a lot to breakage in mills. They further reported that assuming that the collision energies experienced by particles were in any way cumulative, then the smaller particles (27mm and 35mm, for their case) would only last about half a mill revolution before being severely broken, while the larger particles (e.g. 83mm) would never undergo considerable damage after one pass through the mill charge toe. Based on the above Morrison and Cleary suggested that the likely dominant breakage mode for the larger particles would be abrasion.

In this work, it was reported that if the energy per unit mass relationship (specific energy) is valid (or assumed) for smaller ore particle sizes, then the energies required to attain severe breakage will be very small. Further comparison of energies required to achieve breakage and energy distribution results from DEM simulations showed that a very large number of collisions are available for breakage for smaller particles.

It was suggested that the vast number of very low energies as supported by DEM simulation results contribute to abrasion and in particular ¨rounding¨ of irregular (or angular) ore particles in the mill. The current standard JKMRC abrasion parameter, t_a, (which is equal to one-tenth ($\frac{1}{10}th$) of the t_{10} value obtained after tumbling 3kg of 45mm ore particles for 10 minutes in an abrasion mill) is not easy to map into DEM simulation results.

In their concluding remarks, Morrison and Cleary suggested that DEM was predicting considerably low collision energies due to the failure by the idealised perfect spherical ¨ore particles¨ to lock together. This would mean that instead of absorbing more energy during collisions, the hit particle would tend to slide and waste the energy in the form of kinetic energy. They also pointed out that although DEM simulations suggested that most of the collisions do not result in immediate damage but rather cumulative damage, further work needs to be carried out to determine the damage accumulation criteria and prediction of breakage degree.

Discussion

The findings in this work by Morrison and Cleary were significant and mainly showed that DEM simulation of the mill can be used to try and understand the breakage regimes in the mill.

This was the first work to report on a comparison between the energy distribution results obtained from DEM simulation of a pilot-scale SAG mill and a full-size SAG mill. A direct comparison between power draw predictions, for which DEM has proved to be very good at (Cleary (1998), Datta and Rajamani (2002), Datta et al. (1999)), was however not done.

It was shown that the majority of collisions in the mill occur at such low energies that they are not sufficient to cause breakage in after one collision. This led to a significant suggestion that breakage in the mill occurred mainly as a result of several repetitive collisions at low energies. The breakage modelling was based on a comparison of the DEM energy distribution

and the input energies required to achieve breakage. However DEM probably showed such relatively lower energies compared to the required breakage energy due the following:

1. The required breakage energies were calculated based on the JKMRC´s Drop Weight Tester (DWT) for which the input energy is calculated such that the particle is always crushed (Napier-Munn et al. (1999)).

2. The way the DEM collision energy is calculated, is not recorded in the article. Most DEM codes represent the collision energy as energy lost via the dashpots. If the DEM energies referred to in this work are dashpot energies, then DEM is likely to predict less than input energies. The DEM energy will match the input energy if say a cataracting particle falls on a liner shell and does not bounce. This would entail a zero coefficient of restitution between the liner and particle liners. There is always a finite coefficient of restitution for the interaction between the different materials in a DEM simulation.

2.5.2 Validation of a model for impact breakage incorporating particle size effect

Shi and Kojovic (2006)

In this work, Shi and Kojovic developed a breakage model similar to the JKMRC's t_{10} equation (2.36). The A and b parameters that describe the degree of breakage, t_{10} in equation 2.36 are obtained by fitting the equation to data obtained From the Drop Weight Tester. These A and b parameters are the same for all particles of the tested ore, regardless of the particle size. Shi and Kojovic questioned the reliability of the original model especially for SAG/AG mills where particle sizes are known to substantially vary. They therefore suggested a breakage model that incorporates the effect of particle size.

This work was based on earlier efforts by Vogel and Peukert (2004), who had developed a probability of breakage model that incorporated material breakage property, particle size, impact energy and a threshold energy below which breakage does not occur. The Vogel and Puekert model is shown in equation 2.37.

$$S = 1 - exp\left\{-f_{mat}.x.k\left(W_{m,kin} - W_{m,min}\right)\right\} \tag{2.37}$$

where:

- S is the probability of breakage
- $f_{mat}$is the breakage property of the material
- x is the particle size
- k is the number of impacts
- $W_{m,kin}$ is the mass-specific kinetic impact energy
- $W_{m,min}$ is the mass-specific threshold energy

Shi and Kojovic modified equation 2.37 to come up with the t_{10}equation that was a function of particle size, material property and cumulative impact energy. The modified t_{10}equation developed by Shi and Kojovic is shown in equation 2.38.

$$t_{10} = M\ \left\{1 - exp\ \left[-f_{mat}.x.k\left(E_{cs} - E_{min}\right)\right]\right\} \tag{2.38}$$

where:

- M is the maximum degree of breakage (t_{10})
- E_{cs} is the mass specific impact energy

- E_{min} is the minimum that can result in breakage

The modified t_{10} equation of Shi and Kojovic was similar in structure to the original JKMRC t_{10} equation: M was the same as A, the product $f_{mat}.x$ was equal to b, while $k\,(E_{cs} - E_{min})$ was equivalent to E_{cs}.

The similarities between the modified and original t_{10} equations however were only in the structure and the maximum degree of breakage (M and A). The modified equation incorporated the particles' size and material property, which factors were not in the original equation. The modified model also included the number of impacts and the threshold energy below which the particle does not break.

in order to validate their model, Shi and Kojovic used the historical data from Drop weight Tests that had been accumulated at the JKMRC. The modified equation was fitted to the Drop Weight Tester data to obtain curves similar to the JKMRC t_{10} versus Ecs curves. Comparison of the curves obtained by fitting the modified and standard equations to the same DWT data (figure 2.11) showed that the Shi-Kojovic model resulted in a better fit.

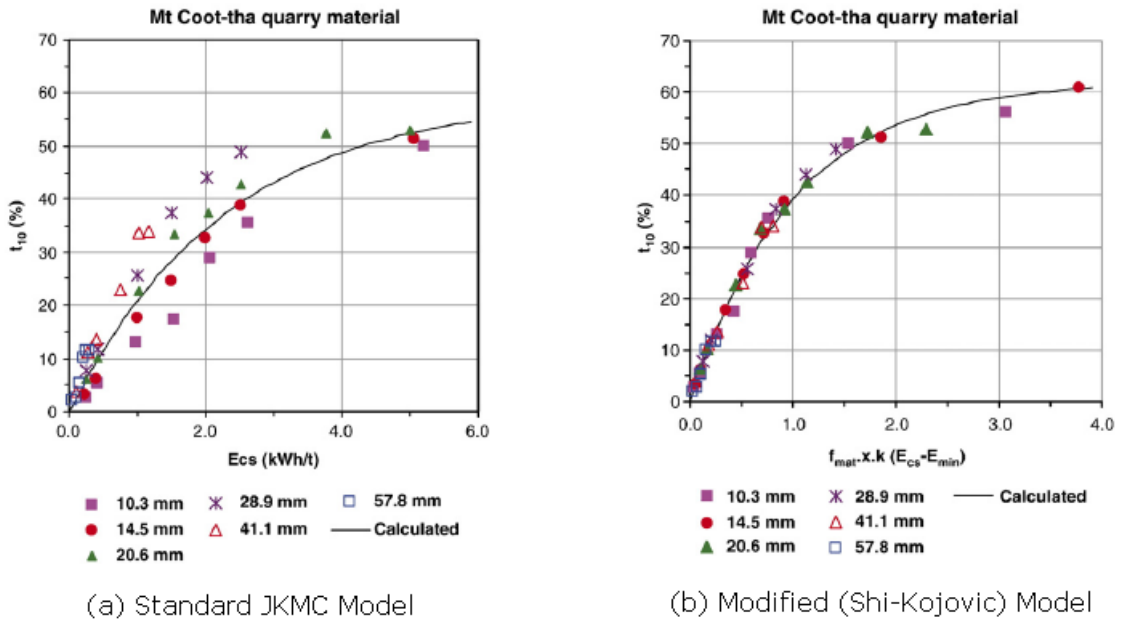

Figure 2.11: *Validation of the Shi-Kojovic modified t_{10} model: Comparison with the original model. After Shi and Kojovic (2006).*

Discussion

The significance of the Shi and Kojovic's work is the inclusion of the particle size effect in the breakage model. The Shi-Kojovic model also incorporates the particle's material property and the threshold impact energy for breakage. In this article they also showed that their model fitted the DWT data better than the standard JKMRC model.

The Shi-Kojovic model provides a possible mechanism for determining and incorporating into the JKMRC t_{10} model, the threshold energy (E_{min} in the Shi-Kojovic model) below which breakage will not occur regardless of the number of impacts. Other researchers had previously suggested the existence of E_{min} (Powell and McBride (2006), Morrison and Cleary (2004)) but did not provide a mechanism for calculating it. Vogel and Peukert (2004) had however earlier used single impact and labscale mill tests to experimentally define E_{min}.

However, there is need to carry out further validation of the Shi-Kojovic model as the DWT data based validation might not be the best for the following reasons:

- The current DWT experiment breaks the particle/s in one impact. The impact of the k parameter in Shi-Kojovic model will therefore not be tested using data from the DWT (k = 1). In order to test the Shi-Kojovic model's sensitivity to the number of impacts, it will be necessary to use a breakage experiment that does not always break the particle in one shot.
- Although the Shi-Kojovic model provides a way of calculating E_{min}, the minimum operating energy of the DWT in its present might be higher than the expected $E_{min}$for most ores. Shi and Kojovic did not show the fitted E_{min} for the validation in this article. A breakage experiment with operating energies lower than the DWT might be required.

- Figure 2.11, appears to suggest that the maximum degree of breakage obtained by the Shi-Kojovic model (61%) is higher than that obtained by the standard JKMRC model (55%).

2.5.3 What is required from DEM simulations to model breakage in mills?

Powell and McBride (2006)

Having questioned the usefulness of current outputs of DEM simulations, to modelling breakage, Powell and McBride proposed several ways in which DEM simulations could be better utilised in breakage modelling. The key issues covered in this work are given in the paragraphs below.

It was proposed that the following factors that apply to comminution device modelling, in general, be considered:

1. The DEM simulations should be used as a tool to give an accurate and detailed description of the mechanical environment in the mill and also data that is useful to modelling of the wear and breakage processes.
2. The mechanical environment variables experienced by the particles, i.e. force, strain, energy, etc, should be related (or linked) to the damage and breakage degree of the particle.
3. In order to achieve (2) above, there is a need to track the complete history of the individual particles in the simulation. All the relevant data like magnitude and angle of the impact force, contact event duration, rate of collisions, etc, need to be recorded.

In order to reduce the storage memory requirements and expedite the analysing and interpretation of the results, it was suggested that binning techniques (like in Govender (2005), Govender et al. (2004), McBride et al. (2004a),

Powell and McBride (2004)) should be used to store the vast amounts of data obtained from tracking individual particles. The use of binning techniques would also mean that experimental results could be readily related to DEM simulation results.

The need for further investigations into the form of the contact model (see section 2.2.2)(especially for DEM simulations whose results should inform breakage modelling) and measurement of material interaction properties (see section 2.2.3) was identified.

In order to relate DEM collision data to particle damage, the current general technique is to split the collision into the normal and shear components. The normal components of the collisions are all allocated to impact (or bulk) damage while the shear components are allocated to abrasion (or surface) damage. Powell and McBride argued that this allocation mechanism is not correct and proposed a different mechanism based on their earlier work (Powell et al. (2003)) in which, for a oblique impact event, abrasion damage only occurs after slip has initiated. A schematic to further illustrate these mechanisms is shown in figure 2.12.

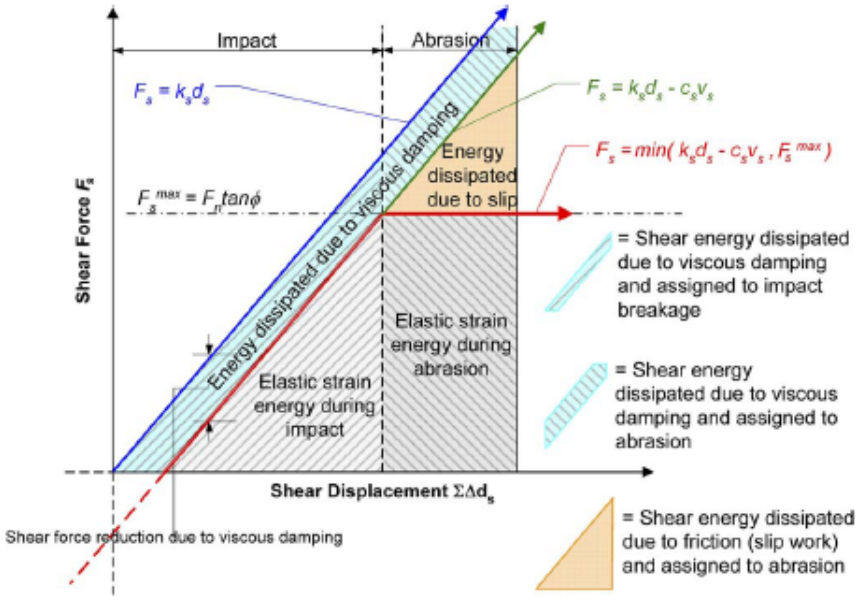

Figure 2.12: *Shear energy dissipation mechanisms Powell et al. (2003)*

The work by Morrison and Cleary (2004), as reviewed in 2.5.1, supported by results from Split Hopkinson Pressure Bar (SHPB) breakage experiments, Bbosa (2005, 2006), to postulate that low energy collisions could be contributing to particle breakage.

Three main breakage mechanisms were discussed namely; impact, abrasion and chipping and spalling. It was proposed that in order for DEM simulations to be useful in breakage modelling, all the three mechanisms should be represented and accounted for.

Impact breakage was further divided into three energy classes (from smallest to largest) as follows:

- Sub-critical - In this class energies are so small that they do not contribute to damage at all
- Sub-breakage - This is the class where the energies will result in breakage after several repeated collisions
- Breakage - Energies in this range result in breakage in one collision or hit. The energy ranges used by the JKMRC's DWT will all fall in this class.

For abrasion damage, Powell and McBride also echoed Morrison and Cleary's observations that more work to come up with more detailed tests to relate the abrasion breakage degree to energy and outputs from DEM simulations need to be carried out. It was reported that tumbling of large pebbles was being investigated as a way of simulating chipping and spalling of irregular shaped rocks, which was postulated as a major contributor to fines generations in the mill.

Discussion

Although DEM simulations are still far from simulating particle breakage in the mill, the proposals made by Powell and McBride in this paper should

prove useful in using DEM simulations as an input into particle breakage models. The DEM simulation calculation algorithm involves the discrete tracking of the individual particles. However the logging of extra information for each individual particle as suggested by Powell and McBride would surely introduce a considerable amount computational overheads in terms of simulation time. Most commercial DEM codes however already log, way too much contact information, most of which is not directly useful for breakage modelling. If, only the data pertinent to breakage modelling is tracked and logged these DEM codes might run faster even with the proposed new information to be recorded. There might therefore also be need to write and develop fast DEM codes that are tailor made for comminution and particle breakage.

The suggestions made in this work, form some of the design considerations of the comminution model that was developed in this thesis. The three impact energy classes were used to describe impact damage in the model. In this thesis work, DEM was used to describe the mechanical environment and predict the energy distribution. This information was then fed into an independent comminution model as suggested in this paper.

Some of the other suggestions made in this paper (like determination of the threshold energies for the three impact energy classes) are already subjects of active research by other workers in breakage testing and DEM at the University of Cape Town and JKMRC.

2.5.4 A direct approach of modelling batch grinding in ball mills using population balance principles and impact energy distribution

Datta and Rajamani (2002)

In this paper, Datta and Rajamani used a combination of impact energy spectra, obtained from DEM simulations, and population balance principles (see 2.4.1.1) to predict the product of batch grinding. This work was

one of the first to couple results from DEM simulations and breakage experiments in a grinding model. Han et al. (2002) also used results from a combined DEM/CFD simulation and Ghadiri's breakage and chipping models in a jet milling grinding model. The approach taken by Datta and Rajamani was effectively to use DEM to predict the breakage rate (or selection function), k in equation 2.31, in a modified population balance equation shown in equation 2.39.

$$\frac{d}{dt}M_i(t) = -\sum_{k=1}^{N}\lambda_k m_{i,k}\frac{M_i(t)}{H} + \sum_{k=1}^{N}\sum_{j=1}^{i-1}\lambda_k m_{j,k} b_{ij,k}\frac{M_i(t)}{H} \tag{2.39}$$

where:

- λ_k is the collision frequency at energy e_k
- $m_{i,k}$ is the mass of material passing the bottom screen, at energy, e_k, in size fraction, i
- $\frac{M_i(t)}{H}$ is the mass fraction, of the total charge mass, of particles in size fraction, i, at a time, t
- $b_{ij,k}$ is the mass fraction of the broken off mass, in size fraction, j, that reports to size fraction, i, at energy, e_k

This model was for a batch mill (no continuous feed and discharge) and assumed that breakage was only due to impact collision events. Another major assumption in the model was that every collision, regardless of energy, resulted in some mass being broken off. Other assumptions included were; no incremental damage, a perfectly mixed mill content and a constant breakage function and broken mass with respect to particle size distribution in the mill.

The three major inputs into the model were the impact energy spectra, broken mass and breakage function. The impact energy spectra was obtained

from DEM simulations while the broken mass and breakage functions were measured from drop-ball breakage experiments.

A two dimensional DEM scheme was utilised to determine the impact energy distribution of the collisions in the mill. The collision frequency, λ_k, that was obtained from the impact energy spectra, represented collisions from all the size fractions in the mill charge. The term $\frac{M_i(t)}{H}$ in equation 2.39 was introduced as a way of linearly distributing λ_k, according to the mass fraction in the mill charge, of each size fraction.

For the breakage experiments, a steel ball was dropped onto a four-layer particle bed contained in a paper cup on a steel anvil. The broken particles were then analysed to determine the broken mass and breakage functions. The variation of broken mass with impact energy (figure 2.13) showed similar trends as the variation of the JKMRC's t_{10} with specific comminution energy (figure 2.14).

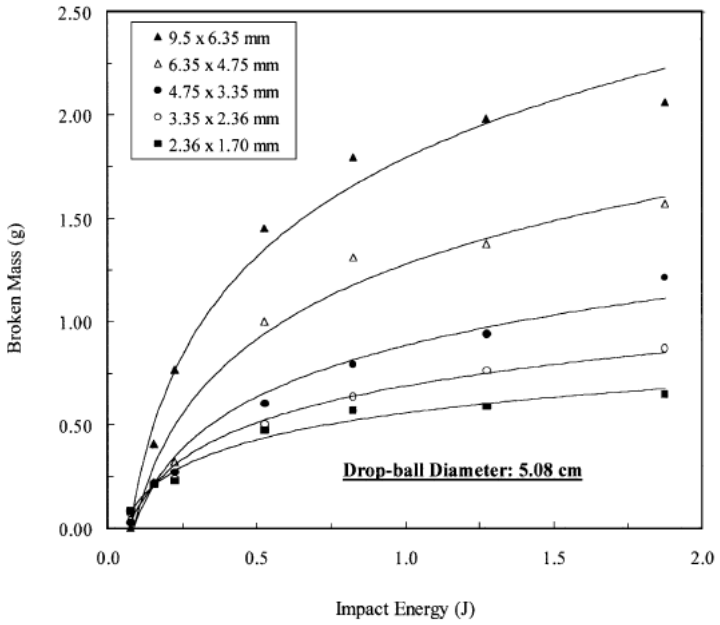

Figure 2.13: *Variation of broken mass with particle size and impact energy. Datta and Rajamani (2002).*

The breakage function results showed a finer product distribution, with increase in impact energy as shown in figure 2.15.

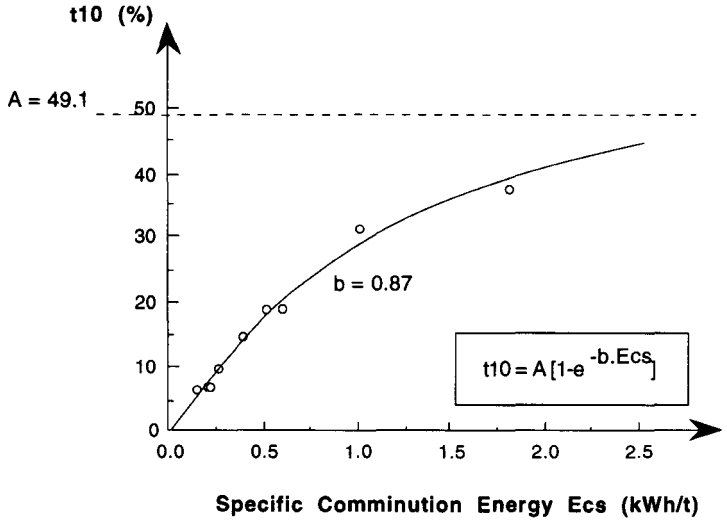

Figure 2.14: *t_{10} versus E_{cs} graph (Morrison and Cleary (2004), Napier-Munn et al. (1999))*

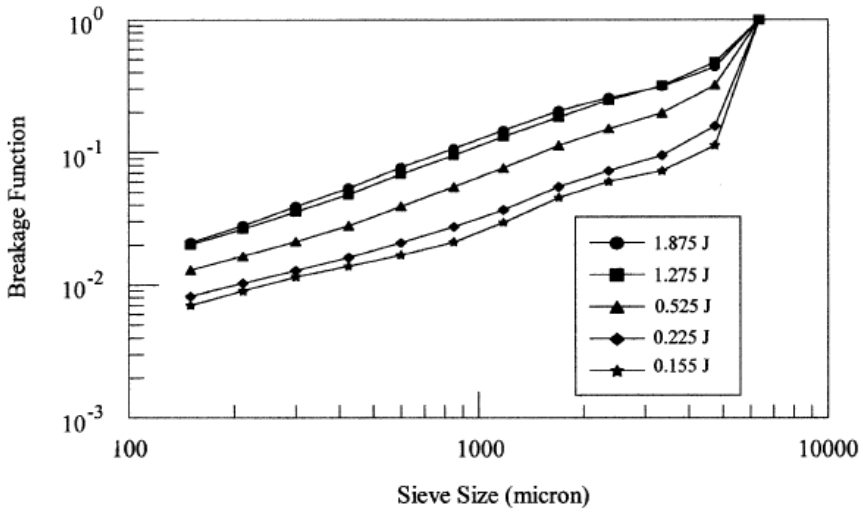

Figure 2.15: *Variation of breakage functions with sieve size and impact energy. Datta and Rajamani (2002).*

In their work Datta and Rajamani (2002) showed that the breakage function for different particle sizes was the same when plotted against the normalised size (with respect to the original particle size) as shown in figure 2.16.

In this paper, it was reported that, unlike the breakage functions, the broken mass was a function of drop ball size. To cater for this, a proportionality factor based on the mass fraction for each ball size was applied.

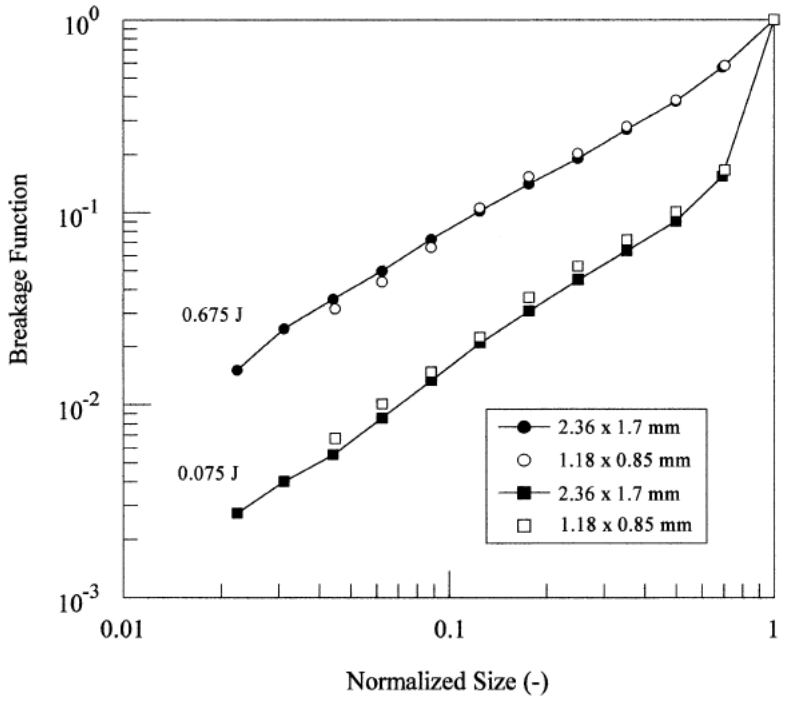

Figure 2.16: *Breakage functions versus normalised size for different particle sizes. Datta and Rajamani (2002).*

The predictions of the model presented in this paper appeared to be in good agreement with measured values (figure 2.17).

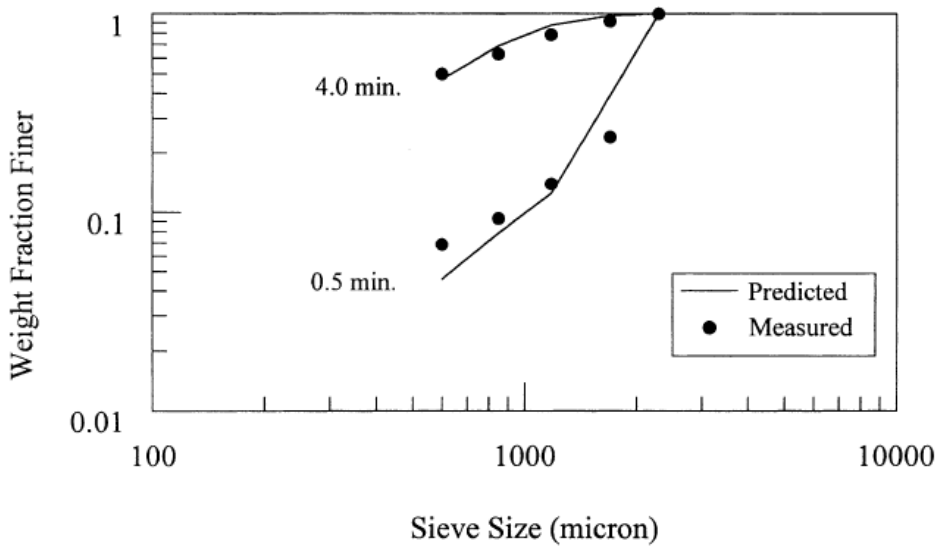

Figure 2.17: *Predicted and measured product size distributions. After Datta and Rajamani (2002).*

Discussion

From this work, Datta and Rajamani (2002), demonstrated that the product of a batch mill can be predicted using results from DEM simulations and breakage experiments. However a correction factor of 0.8 (which was attributed to a possible reduction of breakage with increase in fines) had to be applied to the breakage functions. The over-grinding prediction by the Datta and Rajamani's model which prompted them to introduce this correction factor could be as a result of the following:

1. The assumption that all the collisions, regardless of intensity, results in breakage could be erroneous. The results in figure 2.13, appears to show that there is a finite energy for each size at which the broken mass is zero. This also supports, the suggestion by Powell and McBride (2006) that there is an impact energy threshold, below which the collision does not contribute to any breakage. This assumption could lead to over-grinding predictions because collisions of intensity below the sub-critical should not break anything.

2. Incremental damage which is widely believed to be present in mills (Morrison and Cleary (2004), Powell and McBride (2006)) was not incorporated in the model.

3. The breakage experiments were done for a four-layer particle bed. This could lead to errors because the layer thickness will vary in actual mill operation.

4. A combined impact energy spectra was obtained for all collisions in the charge and distributed over the particle sizes according to the mass fraction. This could introduce errors due to the possible inclusion of collisions between grinding media which do not contribute to breakage. A possible way to avoid this is to predict different energy spectra according to size from the DEM simulation.

Datta and Rajamani also identified the assumption highlighted in the first point, above as a possible contributor to the over-grinding predictions. However they looked at it in terms of spatial distribution of the collisions and not collision energy intensities. This model assumed that breakage is a function of energy intensity only. Adding spatial distribution of the collisions might simply result in increased complexity, when the over-prediction could be alluded to the issues given above.

2.6 Summary of the Literature Review

DEM simulations form one of the main inputs into the tumbling mill model proposed in this thesis. The literature reviewed showed that many of the tools in the area of DEM simulations exist. However the proposed work of using impact energy spectra results from a DEM simulation to inform a tumbling mill model has not been fully conducted previously and is therefore a worthwhile area of investigation.

Chapter 3

THE PROPOSED TUMBLING MILL MODEL

A tumbling mill model that utilises results from DEM simulations and breakage experiments is proposed. The theory behind the proposed model is presented. This chapter discusses the mathematical structure of the model and how it links and uses the inputs from the DEM simulations and breakage experiments to predict the product.

3.1 Introduction

The design, optimisation and performance of comminution devices constitute an important aspect of mineral processing. Conventionally experimental data based on small laboratory-scale mills is scaled-up using black box models to predict the performance of industrial-scale mills. This is an entirely empirical procedure which does not incorporate the actual breakage regime in the mill. The scale-up procedure is believed to result in inaccuracies due to differences in charge motion behaviour between laboratory-scale mills and industrial-scale mills (Datta and Rajamani (2002)).

The DEM on the other hand, is able to simulate directly the motion of the charge for industrial-scale mills. The configuration parameters and operating conditions of the DEM simulations can be closely matched with the actual configurations and operating conditions, that control the throughput and product of the mills (Powell and McBride (2006)). In order to overcome the inherent shortcomings of the conventional tumbling mill models, a DEM-based model for tumbling mills is proposed. Since DEM does not predict breakage, data from breakage experiments is integrated into the tumbling mill model, in order to predict the comminution action.

The incorporation of DEM simulations in the model, enables an accurate description of the mechanical environment experienced by the mill charge. The model attempts to characterise the mechanisms underlying the comminution device and not simply the comminution process, like the conventional black box models.

The development of this tumbling mill model constitutes the first step towards the development of the Unified Comminution Model (UCM) under the AMIRA, P9 project (see Powell (2005, 2006)). The grand vision of the UCM is to have a unified framework which can be used to model all the different types of comminution devices. The UCM attempts to model the fundamental process of comminution, which can then be applied to any comminution device. The envisaged structure of the UCM is shown in figure 3.1, with the central block showing the model principle and the fringe blocks being the required inputs into the model structure.

3.2 The Mathematical Structure of the Model

A framework based on the perfect mixing principle is employed to deal with bulk breakage resulting from impact interaction events. Here impact interaction events are defined as collision events that result in bulk or gross

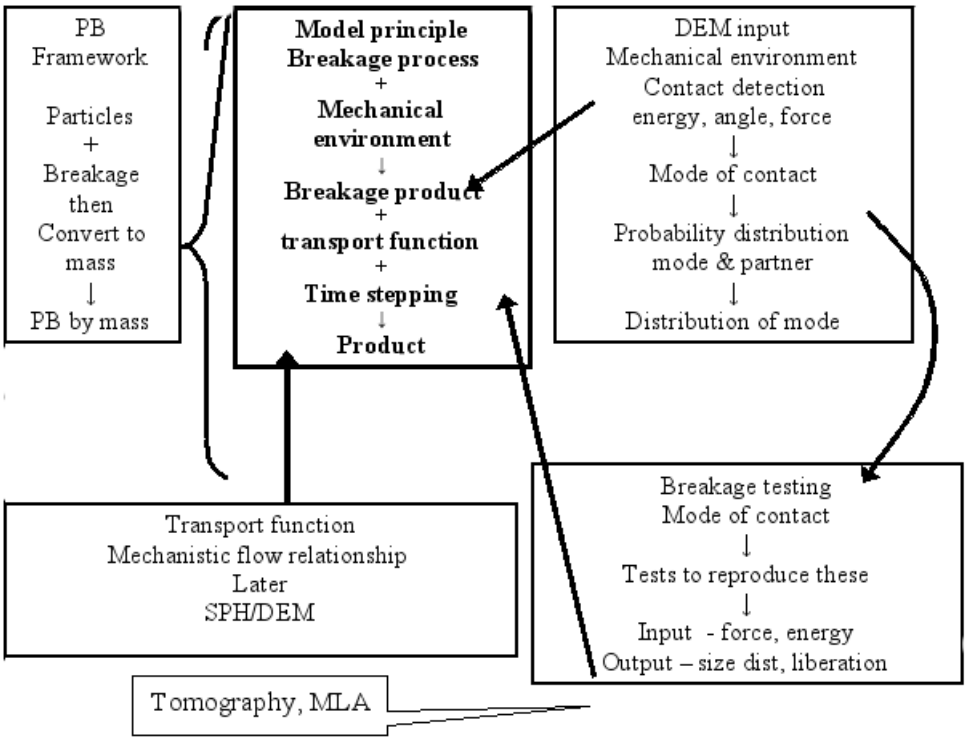

Figure 3.1: *Envisaged structure and components of the UCM. After* Powell (2006)

breakage of particles, yielding different fragments over a range of sizes. The mathematical structure for bulk breakage, for one particle class, $u * i*$, is as shown in equation 3.1.

$$\frac{d}{dt}M_{u*i*} = f_{u*i*} - \sum_{d=1}^{N} \lambda_{u*i*,d}\ m_{u*i*} + \sum_{j=1}^{L}\sum_{d=1}^{N} \lambda_{u*j,d}\ m_{u*j}\ A_{u*j,d,u*i*} - p_{u*i*} \qquad (3.1)$$

where:

- f is the feed into the tumbling mill
- λ is the breakage rate
- m is the average mass of the particles in a size fraction
- A is the appearance or breakage function
- p is the product discharged from the mill

The model structure is such that it can easily deal with a system processing a mixture of different particle classes, where a class includes size, density, ore type, mineralogy, etc. In equation 3.1, the following superscript and subscript convention has been adopted.

- u refers to the particle material class.
- j, i and k refer to particle size class, with $j > i > k$.
- An asterisk, $*$, placed after a subscript indicates a specific value of that subscript. For example, $u*$, refers to a specific material class, say density.
- For a particle to be fully described, one needs to know what material the particle is (the particle material class, $u*$) and the size of the particle (particle size class, $i*$). Such a combination of classes is herein referred to simply as the particle class, $u * i*$.
- d refers to an interaction energy level or bin
- The comma separated subscripts are used to show the involved particle classes and interaction energy level. For example, $u * i*, d$ indicate that particle class $u * i*$ is involved in a collision of energy level, d.

Close examination of equation 3.1, reveals that it is simply a material balance equation around one particle class:

$$Accumulation = Feed\ in - Breakage\ out + Breakage\ in - Product\ out \quad (3.2)$$

Equations 3.2 and 3.1 apply both to batch and continuous grinding. In case of continuous grinding, the $accumulation$ term becomes $zero$ under steady state conditions, while there in no *feed-in* or *product-out* for the batch grinding mill. In this model, for a certain particle class, $u * i*$, the

amount of mass that breaks out of the particle class is calculated by summing up all the breakages across all the energy level bins. Before going into further details of the mechanisms of the model, it is of preponderant importance that the incorporated assumptions are laid out:

1. Breakage is a function of size, interaction energy and material properties of the interacting particles. The breakage rate, λ, is affected by the particle class, collision energy and breakage properties.
2. The number of interactions per unit time influences the final breakage. Depending on the collision energy level, the particle might break after one hit, several hits or not at all.
3. The energy dissipated during a DEM collision is shared equally between the interacting particles.
4. No new particle size classes will be generated. That is, whenever a particle breaks, it will always break into the same pre-defined particle size classes.
5. No new energy classes will be produced. This means that the values of the collision energies will be pre-defined and remain constant.
6. No new appearance functions will evolve. This implies that the distribution of particle class $u * i*$, upon breakage with dashpot energy d, into the $u * j$ size classes will remain constant with time.
7. The contents of the mill charge are perfectly mixed.

The rationale of this proposed tumbling mill model is similar to that of Datta and Rajamani (2002)[1]. To determine the amount of material breaking out of a certain particle class, the number of breakages in a specified time period (breakage rate, λ) is multiplied by the average mass of the particles

[1]Reviewed in previous chapter (section 2.5.4)

in that particle class, m. This is done for all the collision energy bins and the masses summed up to get the total breakage out. For the material breaking into a certain particle class, the breakage rate of the particle class that is breaking in, is multiplied by it average mass and fraction of the original that is breaking into the particle class being considered (appearance function,A). This is summed up over all the particle classes breaking in for each energy bin and also over all the energy bins to get the total material breaking in.

The differences and improvements of this model over the Datta and Rajamani (2002) model are:

- The model can be adapted to handle both batch and continuous mills.
- The model was extended to include the fine particles that can not be simulated by most of the current DEM codes.
- The model was also extended to predict the product that results from surface breakage (abrasion, shear, etc).
- Impact energy spectra are obtained separately for each particle class and not obtained from a mass fraction linear distribution of the total impact energy spectrum.
- The perfect mixing principle was used where particles are allowed to break into their original size fraction.
- Not all collision events resulted in breakage. A threshold energy below which no breakage occurs was incorporated into the model.
- The principle of incremental breakage was also embraced in the model.
- Breakage experiments are performed using the Drop Weight Tester (DWT) or Split Horizontal Pressure Bar (SHPB).

3.3 The Mechanical Environment (DEM Simulation)

The Discrete Element Method (DEM) as described earlier in chapter 4, is used to profile the mechanical environment of the tumbling mill. In this context, the mechanical environment refers to the physical boundaries within which the charge is confined (liners, lifters, end plates, etc). With the ability to easily import CAD generated drawings into modern DEM packages, DEM is well suited to describe the mechanical environment. DEM can be used to simulate any tumbling mill whose mechanical environment can be successfully constructed using CAD. The incorporation of DEM, therefore means that the model can be used for any such CAD generation capable tumbling mills.

It is possible to even include the feed trunnion and discharge chamber, complete with grates and pulp lifters (like in Cleary (2004)) in a DEM simulation. In order to discharge the fines, it will be however necessary to use a DEM package with full fluid flow coupling capability. The DEM package used in this thesis work did not have such fluid flow coupling. The discharge grates and chamber were therefore not included in the DEM simulations in this thesis. The feed trunnion was also replaced by a solid face plate. The mill in a DEM simulation effectively operates in batch mode, because no fresh feed is added and no product discharged.

The key assumption used in the model is that the number of interactions experienced, per unit time by a particle affects its breakage. DEM is used to simulate the interactions of the ball and ore particles with each other and the surrounding environment. For each full collision event[2], the energy dissipated via the dashpot is recorded. The number of collisions is binned according the collision energy level, in an impact energy spectrum. The impact energy spectrum for each ore size class forms one of the key

[2]A full collision event is from the time the contacting partners first touch to the time they separate

inputs to this tumbling mill model. The procedures followed for constructing the energy spectra are detailed in the following chapter, which gives an example of a DEM simulation for incorporation into the proposed tumbling mill model.

3.4 Breakage Experiments

Apart from the DEM simulation of the tumbling mill, breakage experiments form an integral part of the proposed model. The breakage experiments inform the model on the actual number of collisions needed to break a particle. The appearance functions are also determined from breakage experiments.

There are several different breakage experiments available for use in comminution (DWT, UFLC, SHPB, Piston and Die, Rotary Breakage Tester (RBT), etc). The ideal breakage experiments for producing the required input into this model would be the SHPB or the DWT.

3.4.1 The Split Hopkinson Pressure Bar (SHPB) Experiment

Breakage in the SHPB experiments is achieved when a short $striker$ bar is propelled to hit the $incident$ bar which, together with the $transmission$ bar, sandwiches the sample rock particle as in the schematic below. The sample particle is housed in a protective glass enclosure which also doubles as the collection bucket for the broken pieces (figure 3.2).

The SHPB is able to accurately measure the energy that is absorbed by the rock specimen being tested. This is accomplished by comparing the transmitted strain wave that is propagated forward through the rock specimen and the reflected (or incident) strain wave as it is reflected from the incident bar.

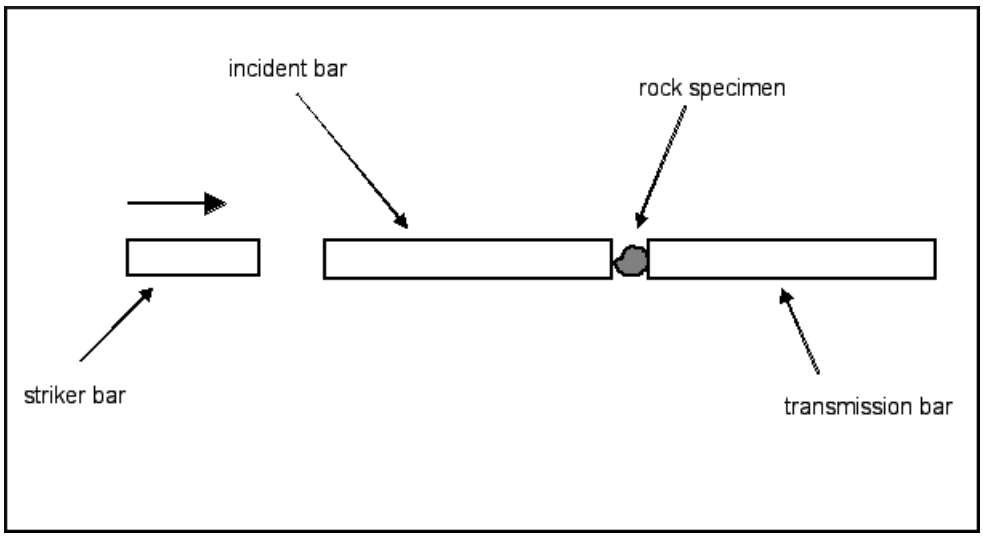

Figure 3.2: *Schematic of the Split Hopkinson Pressure Bar experiment*

The ability to measure energy absorbed when breakage occurs would make the SHPB the preferred choice of breakage experiment for use with the proposed mill model. This is because the input energy absorbed by the rock specimen is equivalent to the energy dissipated via the numerical dashpot in the DEM simulation. This would be the ideal breakage experiment to determine the actual number of impacts required to break a particle, for comparison with the DEM results to calculate the breakage rate.

Work to use the SHPB to produce data to be used in this tumbling mill model was ongoing under another Master's research project at the University of Cape Town (Bbosa (2007)) at the time of writing this thesis. Preliminary scoping results from the research have identified the need to modify the current SHPB rig to achieve low input energies required to measure incremental breakage. The other modification would be to enable it to cover a wider range of particle sizes.

3.4.2 The Drop Weight Tester (DWT) Experiment

In the JKMRC's DWT experiment, a pre-defined steel weight is dropped from a known pre-defined height on to a single particle. The whole experiment

(figure 3.3), is conducted in a perspex housing and the broken fragments are collected for sizing and screening.

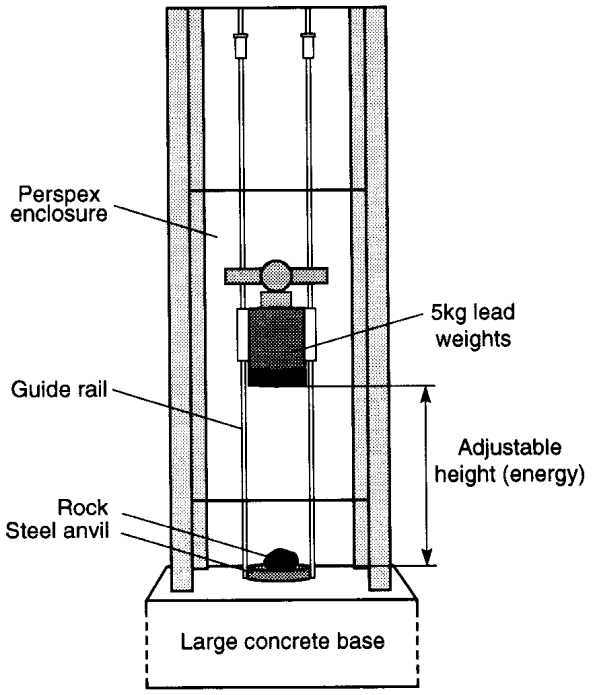

Figure 3.3: *Schematic of the JKMRC Drop Weight Tester. Taken from Napier-Munn et al. (1999).*

For the DWT, the quantifiable energy is the specific comminution energy (Ecs) which is usually specified in kWh/t. This Ecs energy is the gross input potential energy that is applied to break the particle/s. Unlike the SHPB, the DWT does not measure the energy that is absorbed by the rock under test. Using the current JKMRC DWT procedures, the rock particle under test is always broken. Although Whyte (2005), adapted the DWT to achieve incremental breakage, the current DWT can not be used to investigate incremental breakage. The current DWT also has a restriction on the range of applicable input energies and particle size.

Despite the above, as is demonstrated later in this thesis (Chapter 6), the information obtained from the DWT can be post-processed to be used as input to the proposed tumbling mill model.

3.5 Incremental Breakage and Energy Regimes

Apart from the use of DEM, one of the major improvements of this model over existing ones is that incremental breakage, which is believed to contribute to breakage in mills (Briggs and Bearman (1995), Morrison and Cleary (2004), Powell and McBride (2006), Whyte (2005)), is incorporated. Depending on the number of hits required to result in breakage, the collision energies are classified into three regimes as follows:

- *No breakage* - In this regime, the collision energy is below the minimum threshold energy, E_o, required to cause any damage that can contribute to breakage of the particle.
- *1-hit breakage* - In this category, breakage occurs after only one collision. The collision energy in the range is equal or higher than the critical energy required to break the colliding particles, E_{crit}.
- *Incremental breakage* - This is the category where the particles break after more than one collision. Breakage here occurs after a finite number of collisions (greater than one). This energy category falls between the *no breakage* range and *1-hit breakage* range and the energies are between E_0 and E_{crit}.

3.5.1 Calculating the Breakage Rate

In the model, the number of interactions in a certain energy bin obtained from the DEM simulation impact energy spectra is compared to the number of actual collisions at that energy, needed to cause breakage. This gives us the number of actual breakages that occur over a known duration, effectively yielding the breakage rate as in equation 3.3.

$$\lambda = \frac{\Lambda^{DEM}}{\Lambda^{Exp}\, t_{sim}} \tag{3.3}$$

where:

- λ is the breakage rate in breakages per second
- Λ^{DEM} is the number of collisions from the DEM impact energy spectrum
- Λ^{Exp} is the actual number of collisions to break
- t_{sim} is the duration over which the DEM collision energies in the impact energy spectra, were collected, seconds.

It should be noted that:

$\lambda = 0$, for the *no-breakage* energy regime, while

$\lambda = \frac{\Lambda^{DEM}}{t_{sim}}$, for the *1-hit breakage* energy regime ($\Lambda^{Exp} = 1$).

The breakage rate is calculated separately for each particle class. When a collision event occurs, the energy recorded by DEM is the total energy dissipated. In this model, the energy that is assumed to cause the breakage of a particle is the energy that is absorbed by the particle. The energy recorded by DEM is the total absorbed by both colliding partners. The apportioning of this energy between the respective colliding partners is not addressed or fully understood in the literature. In the absence of such information, in this thesis we assume that the DEM dashpot energy is shared equally between the interacting partners. The issue of collision energy apportioning needs to be investigated further.

3.6 Obtaining the Appearance Function

The appearance function, A, is a mathematical function that describes the progeny size distribution. In equation 3.1, the appearance function, $A_{u*j,d,u*i*}$, gives the distribution of the particles that report to size, $u * i*$, when a particle of size, $u * j$, is broken during a collision of energy level, d.

The appearance function is usually given as fractions of the original particle, which, for a particular particle size, $j*$, and energy bin, $d*$, should sum to 1, as shown in equation 3.4, below.

$$\sum_{i}^{b} A_{u*j*,d*,u*i} = 1 \tag{3.4}$$

In this work, size fractions covering all the sizes between the bottom and top screens are considered. It is therefore possible for a some of the original particle to report to its original size fraction.

Using any of the three breakage experiments above, the appearance function can be determined by screening and sizing the broken pieces after an experiment. This would however require the experiments to repeated for each collision energy bin in the model. The lower bound of the collision energy bins, E_0, as determined from the preliminary work of Whyte (2005) would be in the neighbourhood of $0.01\ kWh$. Such a lower threshold energy is too small to be attained, for the currently available breakage experiments.

In the absence of breakage data from the SHPB project and bearing in mind the afore-mentioned limitations of the DWT, the raw information obtained from DWT experiments can be post-processed to determine the appearance functions. In this thesis, the following procedure, which is based on the JKMRC's $SimMet$ model, is used.

3.6.1 Step 1 - Determine the t_{10} vs Ecs curves

The degree of breakage is quantified using the JKMRC's t_{10} terminology. The t_{10} value is defined as the percentage of the original particle that passes a screen size of one tenth of the original particle size. The degree of breakage, t_{10}, is related to the input energy (specific comminution energy, Ecs) by equation 3.5, below.

$$t_{10} = A\,[1 - e^{(-b\,Ecs)}] \tag{3.5}$$

where:

- A is the maximum degree of breakage that can be achieved.
- b is a measure of the energy level at which maximum breakage is achieved.

The parameters A and b are ore impact breakage parameters and are determined experimentally from the DWT. Figure 3.4 shows a typical variation of t_{10} with Ecs, obtained from DWT experiments.

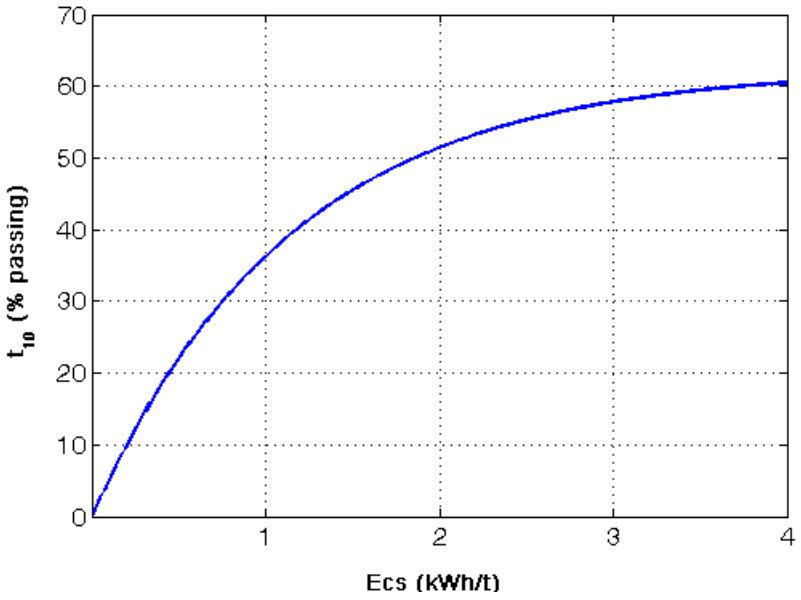

Figure 3.4: *$t_{10} - Ecs$ curve for the target ore with A and b values of 62.4 and 0.87, respectively*

The collision energy calculated by DEM is the energy that is dissipated during the collision. On the other hand, the energy used in the DWT experiments is the gross input energy, used to break a particle. The consequence of this major difference in the DEM and the DWT energies, is that the DWT energies are likely to be much higher than the DEM energies. In order

to use the JKMRC's $t_{10} - Ecs$ relationship, which is derived from DWT experiments, it was necessary to devise an appropriate scaling mechanism between DEM and DWT energies.

An equivalent DWT experiment was simulated in the DEM. In the current DEM, it is possible to construct the cylindrical steel drop weights. However this could only be done as what is termed as "geometry" or "walls" in DEM, which does not have a definite mass. In this work, the drop weight was represented by a steel ball, which was dropped on to an ore particle . The drop height was calculated as described in Napier-Munn et al. (1999), from the masses of the steel ball and ore particle and the desired input energy as in equation 3.6

$$h_i = \frac{m\, E_{is}}{0.0272\, M_d} \tag{3.6}$$

where:

- h_i is the drop height
- m is the mass of the ore particle
- E_{is} is the desired specific input energy
- M_d is the mass of the steel drop particle

To obtain the required scaling, the dashpot energy recorded from the DEM simulation was compared to the DWT input energies, for each ore particle size.

The DEM energies were scaled-up and binned as shown earlier. The scaled-up DEM collision energy bins, combined with the DWT determined A and b values, were used in equation 3.5 to calculate the t_{10} values for each size fraction.

3.6.2 Step 2 - Get breakage for standard t-values

In the JKMRC's t_n notation, the parameter t_2 denotes the percentage of the original particle passing one half of the original particle size. This is extended to all the standard t_n values, $t_2, t_4, t_{10}, t_{25}, t_{50}$, and t_{75}. Each t_n value can be plotted againstt_{10}, yielding the so-called $t_{10}-t_n$ family of curves. Work at the JKMRC by Narayanan and Whiten (1988), had shown that most ores can be represented using a the same "master" $t_{10} - t_n$ family of curves, described as spline functions passing through predetermined spline knots. A typical $t_{10} - t_n$ family of curves is shown in figure 3.5.

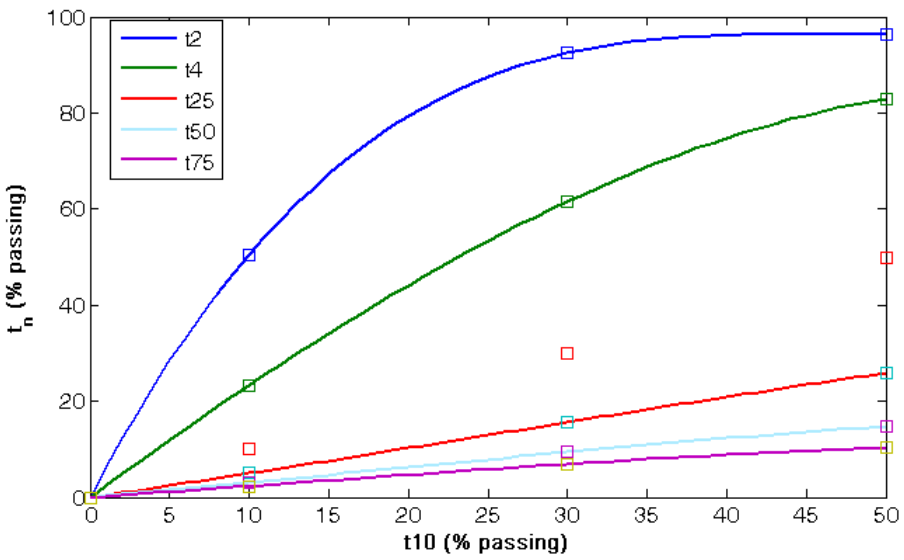

Figure 3.5: *Typical $t_{10} - t_n$ curves*

From the $t_{10} - t_n$ curves, the percentage passing the standard t_n sizes are calculated for each collision energy bin and each particle class. This is done by reading-off the t_{10} value for each energy from the $t_{10} - E_{cs}$ curve. This t_{10} is used to read-off the percentage passing each of the standard t_n sizes.

3.6.3 Step 3 - Extrapolate size distribution to standard sizes

In step 2, the percentage passing the standard t_n sizes for each particle size are obtained for each collision energy bin. The t_n sizes will be different

for each original particle size. The proposed tumbling mill model uses the standardised $\sqrt{2}-series$ particle sizes. Therefore the percentage passing the $\sqrt{2}-series$ particle sizes needs to be calculated for each size and energy bins. In this work, the Rosin-Rammler size distribution function (Napier-Munn et al. (1999)), shown in equation 3.7 was fitted to the data points obtained in step 2, to get the percentage passing the model's particle sizes (figure 3.6). The percentage passing the model sizes, are finally converted into fractions of the original particle's mass to obtain the appearance function.

$$W_r = 100\ e^{[-(\frac{x}{a})^b]} \tag{3.7}$$

where:

- W_r is the percentage weight retained
- x is the particle size
- a is the size at which the percentage weight retained, W_r is 36.8%
- b is the slope of the plot of $ln\,ln\,(100/W_r)\ vs\ ln\,x$

3.7 Handling Sub-DEM and Abrasion (Surface) Breakage

3.7.1 Sub-DEM breakage

The simulation time, memory, storage and post-processing time for a DEM simulation are proportional to the total number of particles in the simulated system. The number of particles in a full size mill will be in excess of $1\,million$. Running a DEM simulation of such a mill would pose serious computational constraints and limitations on today's normal desktop computers. In order to reduce the number of particles to manageable proportions, the range of particles in DEM simulations is normally cut-off at a

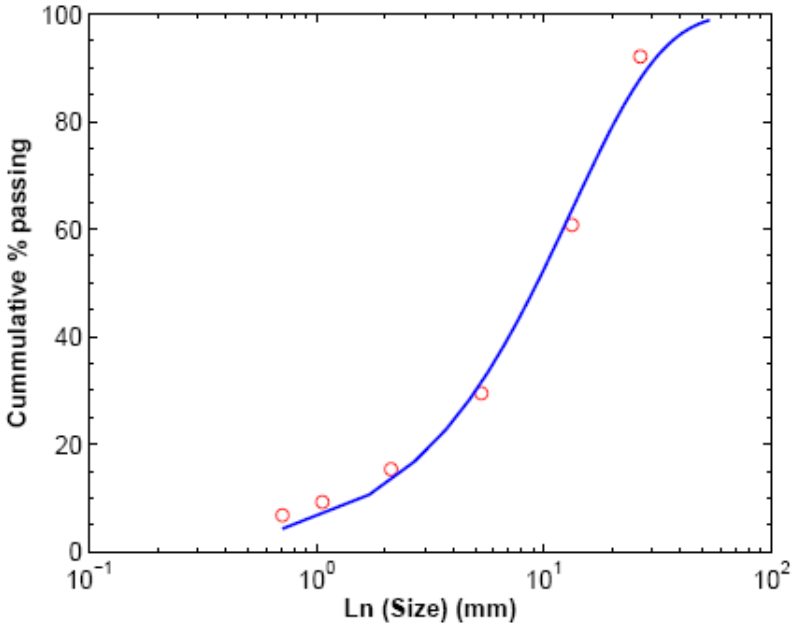

Figure 3.6: *A Rosin-Rammler distribution fitted to % passing t_n sizes data points*

certain minimum size. The very fine particles are normally left-out because due to their relatively very small volumes, their numbers in the actual mill are very large. The very fine particles of sizes below the DEM cut-off size are referred to in this thesis as the "*sub-DEM*" particles. The DEM cut-off size is selected by some researchers (Cleary (2004), Morrison and Cleary (2004)) in such a manner that at least 90% of the total mass of the particles in the actual mill are represented in the DEM simulation.

Because the DEM simulation does not normally include the very fine particles, the proposed tumbling mill model described thus far, will not cater for the very fine particles. Depending on the DEM cut-off size, the sub-DEM particles would sometimes be just larger than the mill product sizes. In such a case, the breakage of the sub-DEM particles would actually also contribute to the mill product. Omitting the sub-DEM particles would therefore result in some considerable errors in the model.

In this thesis, a way of dealing with the sub-DEM particles is suggested. This method is based on the Bond Work Index theory and though not as

elaborate as the above purely DEM based tumbling mill model, is presented in this thesis work for the sake of completeness. As highlighted in the Literature Review, the Bond method has a number of limitations (it falls short when it comes to predicting the behaviour of real closed circuits (Napier-Munn et al. (1999)), among others). In their widely referenced book, *Mineral Comminution Circuits - Their Operation and Optimisation*, Napier-Munn et al. (1999) point out that despite it's limitations, the Bond method is still commonly used today due to it's simplicity and works well for particle sizes less than $4.9mm$.

The review of literature in this thesis indicated that the DEM predicts the power draw of the tumbling mill, quite successfully. Preliminary work by the author showed that DEM's predictions of power draw were in fact always less than the actual experimentally determined power draw. The suggested method of including the sub-DEM breakage, is based on the premise that the DEM power draw prediction is lower than the measured power draw because the power drawn by interactions involving sub-DEM particles is not included. The power draw due to the sub-DEM particles' interactions can therefore be calculated by equation 3.8:

$$P_{sub-DEM} = P_{Measured} - P_{DEM} \tag{3.8}$$

The Bond Work Index equation, as described earlier in the Literature Review (equation 2.29) is as shown below:

$$W = 10\, W_i \left(\frac{1}{\sqrt{P_{80}}} - \frac{1}{\sqrt{F_{80}}} \right) \tag{3.9}$$

where:

- W is the work input

- F_{80} and P_{80} are the sizes at which 80% of the feed and product passes, respectively
- W_i is the work index (work required to grind a feed of infinite size to the given P_{80})

The power draw due to the sub-DEM particles' interactions as given by Bond (Napier-Munn et al. (1999)) can be calculated from equation

$$P_{sub-DEM} = T\,W \tag{3.10}$$

where:

- T is the throughput of new feed

Combining equation 3.9 and equation 3.10 results in the following equation:

$$\frac{P_{sub-DEM}}{T} = 10\,W_i\left(\frac{1}{\sqrt{P_{80}}} - \frac{1}{\sqrt{F_{80}}}\right) \tag{3.11}$$

The Bond work index $W_i$can be determined experimentally via the Bond ball mill test which utilises a batch ball mill like the one used at MINTEK (shown in figure 3.7).

The through put of the feed, T, and the size at which 80% of the feed passes, F_{80}, are usually known parameters. The only unknown parameter in equation 3.11, is the size at which 80% of the product passes, P_{80}. Equation 3.11 can therefore be used to determine the product resulting from the breakage of sub-DEM particles. This can then be easily integrated into the DEM-based tumbling mill model to determine the full product as will be demonstrated in the next chapter.

Figure 3.7: *Batch ball mill used to determine the Bond work index at MINTEK*

3.7.2 Abrasion (Surface) Breakage

The proposed tumbling model so far described above will only be able to handle breakages that result from bulk (or impact) interaction events. Here bulk (or impact) interaction events are taken as all those interactions that result in bulk or gross breakage of the particle, resulting in different fragments over a range of sizes. The scheme described so far, does not cater for breakages that result from surface interactions. A surface (or Abrasion) interaction event is taken as any interaction that results in breakage only at the surface of the particle, leaving the original particle largely intact. In order to apply equation 3.1 for surface breakage, breakage experiments that measure surface breakage will have to be used to obtain the appearance function. The breakage experiments described thus far (Split Hopkinson Pressure Bar and Drop Weight Tests) are specific to bulk breakage and do

not measure surface breakage.

One of the most commonly used surface breakage experiments is the JKMRC abrasion test. In the JKMRC abrasion test, a 3kg sample of $38mm-55mm$ diameter ore particles are tumbled in the standard abrasion test mill ($300mm$ diameter by $300mm$ long with $4\,x\,10mm$ lifters) for 10 minutes at 70% of the critical speed (Napier-Munn et al. (1999)). The mill products are then sized to determine the t_{10} parameter. The usual result from the JKMRC abrasion test is the t_a parameter which is a measure of the material's resistance to abrasion. The t_a parameter is defined as

$$t_a = \frac{t_{10}}{10} \tag{3.12}$$

Work at the University of Cape Town by Van Eck (2007) employed a procedure similar to that of Loveday and Naidoo (1997) to measure abrasion. Van Eck (2007) had to use two mills of different sizes in order to obtain two different input energies.

In order to use the abrasion test in the proposed model, one would therefore need to carry out the tests over the full range of input energies. The input energy for the abrasion test is determined by the size of the tumbling mill. This would entail repeating the test using different mill sizes according to the required number of input energies.

Due to the issues articulated above, the proposed tumbling mill model as presented in this thesis work did not include surface breakage.

3.8 Discharging the Product

For a mill operating at steady-state, the term, $\frac{d}{dt}M_{u*i*}$, in equation 3.1 is equal to 0. The product from the mill is discharged according to the discharge (or grate classification) function. For a perfectly mixed charge, the

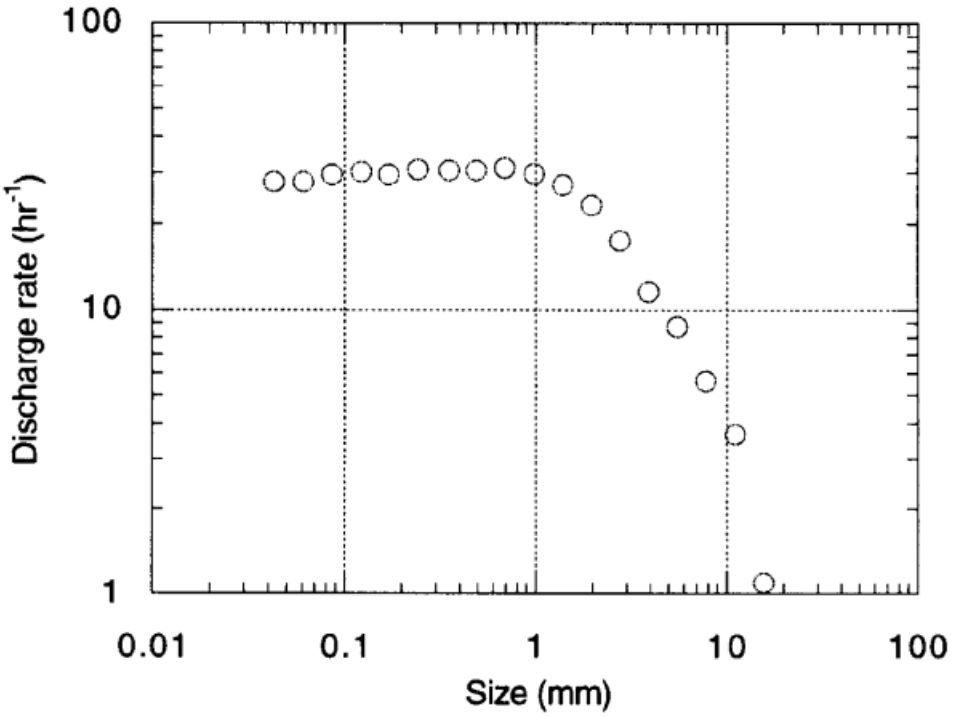

Figure 3.8: *Typical discharge rate vs size curve. For the Alcoa Pinjara SAG mill. After Napier-Munn et al. (1999)*

rate of discharge for particles of a particular size can be calculated from equation 2.33, as

$$d_i = \frac{p_i}{s_i} \tag{3.13}$$

Where:

- p_i is the product rate of particles of size, i
- s_i is the mill contents of particles of size, i

A typical discharge rate versus particle size curve and grate classification function are shown in figures 3.8 and 3.9.

In the proposed tumbling mill model, the discharge rate is pre-determined before running the model. While s_i, the mill contents are known, p_i, the product rate is unknown and has to be calculated from the feed rate and

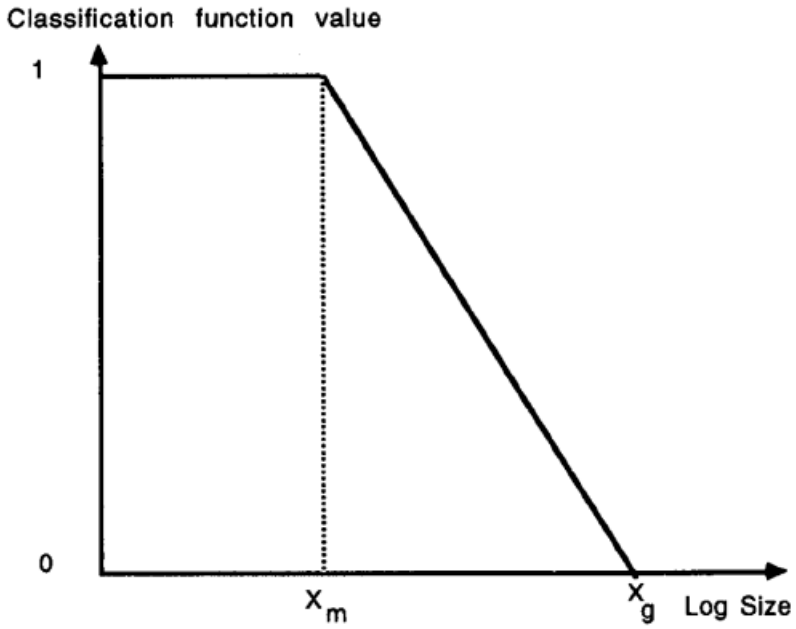

Figure 3.9: *Typical grate classification function. After Napier-Munn et al. (1999).*

product particle size distribution as follows (note that this equation assumes that the rate at which particles are fed into the mill is the same as for the product leaving the mill):

$$p_i = \frac{(\%passing\, product)_i}{100} * feed\, rate \tag{3.14}$$

Once the product rate is known, the discharge rate, d_i, is calculated from equation 3.13. To obtain the predicted product, the calculated (pre-determined) discharge rate, d_i, and simulated mill contents, $s_{i_{sim}}$, are used in a variation of equation 3.13 as follows:

$$p_i(predicted) = d_i * s_{i_{sim}} \tag{3.15}$$

Chapter 4

DEM SIMULATION OF A PILOT-SCALE SAG MILL

In this thesis work, the DEM constituted one of the key inputs into the proposed tumbling mill model. This chapter presents the methodology employed to carry out the DEM simulations. Results obtained for the DEM simulation of a pilot-scale SAG mill, which is used to demonstrate the proposed tumbling model, are also presented.

4.1 Introduction

DEM formed an integral part of this thesis, that seeks to demonstrate the use of DEM in a tumbling mill model. Careful consideration and effort, therefore had to be given to the development of the methodology for the setting up and running of the DEM simulations. A methodology for the extraction and analysis of the results from the DEM simulations such that they can directly be used as input in the proposed tumbling mill model was also developed. To demonstrate the methodology, a pilot-scale SAG mill, operated according to actual experimental tests, is used.

4.2 The DEM Software

The use of DEM for modelling particulate matter has proliferated so much since the seminal work by Cundall and Strack (1979), that there are quite a number of DEM specialised commercial software that are available for use. Examples of specialised DEM commercial software include Particle Flow Code (PFC) by Itasca Consulting, EDEM by DEM solutions, PASSAGE/DEM by Technalysis, ChuteMaven by Hustrulid Technologies and ELFEN by Rochfield Software. Further and more information on the above specialised DEM commercial software can be obtained from the following manuals and world wide web sites: Itasca Consulting Group (1999) (PFC), DEM Solutions (2006) (EDEM), $http://www.technalysis.us/demsoftware.aspx$ (PASSAGE/DEM), $http://www.chutemaven.com/$ (ChuteMaven) and $http://www.rockfield.co.uk/elfen.htm$ (ELFEN).

PFC3D and EDEM were used for the simulations in this thesis work. Most of the simulations were done using EDEM. The author found EDEM to be relatively easy to use with a user friendly and well structured Graphical User Interface (figure 4.1). Unlike PFC3D, EDEM however, at the time writing of this thesis, did not allow a user to choose or define a contact model. The other setback was that EDEM could not output collision energy distribution graphs. These had to be constructed from raw collision data as detailed in section 4.6.

4.3 Simulation Parameters

One of the major advantages of a DEM based tumbling mill model over the traditional black box models is that the scale-up stage (which is believed to introduce errors (Datta and Rajamani (2002)) is bypassed. The actual mill that needs to be modelled is simulated. The DEM simulation parameters, including the mill environment, operating conditions and material

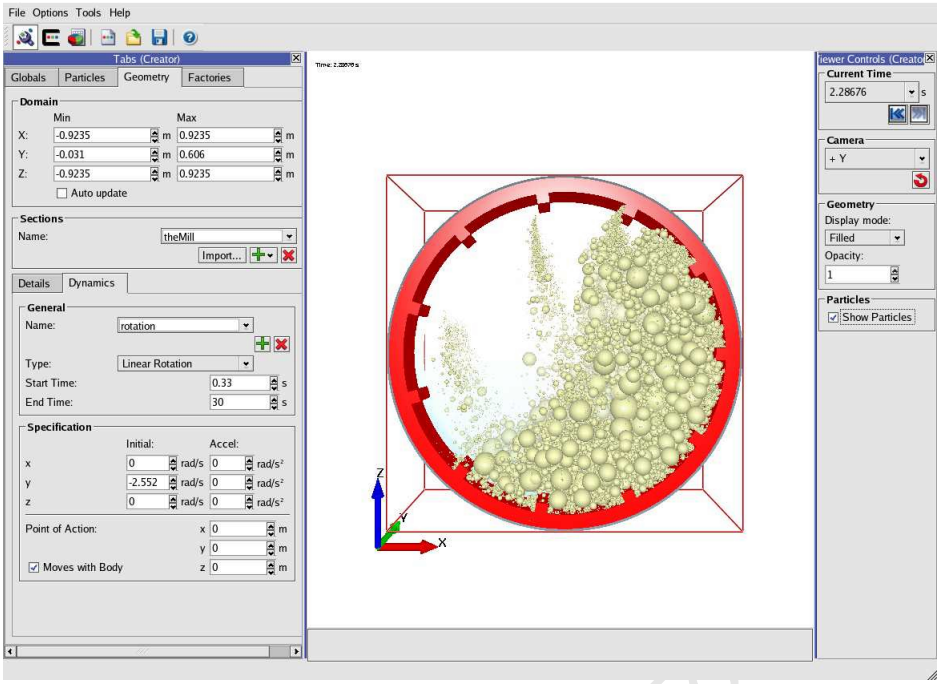

Figure 4.1: *EDEM v1.1 screen shot*

parameters, therefore need to be based on the actual mill. For this thesis work, the pilot-scale SAG mill (figure 4.2) at MINTEK, Johannesburg, South Africa was simulated.

Figure 4.2: *MINTEK pilot SAG mill*

4.3.1 Mill Geometry

Comminution in tumbling mills results from the transfer of kinetic energy from the rotation of the mill to the charge. This transfer of energy is mainly via the mill shell and lifters. It is therefore necessary to model the geometry of the mill shell and lifters as realistically as possible.

The MINTEK pilot SAG mill geometry parameters were as given in table .

Length	0.575m
Diameter	1.695m
Number of lifters	11
Lifter Geometry	Square
Lifter Width	45mm
Lifter Height	55mm

Table 4.1: *Geometry parameters for the MINTEK pilot SAG mill*

The native geometries included in EDEM are cylinder, box and polygon. Constructing a tumbling mill geometry complete with lifters, from these basic shapes easily becomes cumbersome and tedious. EDEM however also allows the user to import CAD geometries saved in either STEP (.stp) or IGES (.igs) format, thus providing for any geometry that can be successfully constructed using CAD.

The geometry for the MINTEK pilot SAG mill was constructed using the PRO/ENGINEER WildFire 2.0, 3D solid modelling CAD software package, as shown in figure 4.3. The CAD geometry was then saved in 'iges' format and imported into EDEM.

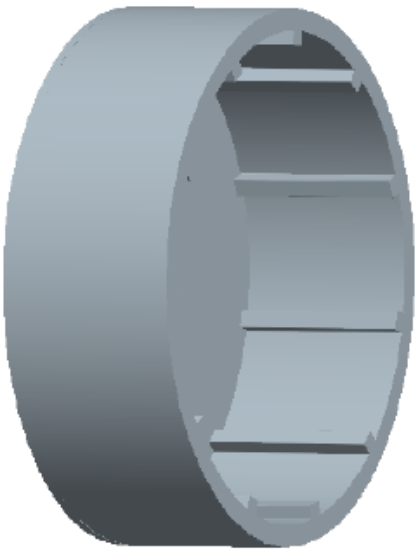

Figure 4.3: *MINTEK pilot SAG mill shell CAD geometry that was imported into the EDEM software*

4.3.2 Operating Conditions

The performance of a tumbling mill depends on the operating conditions, like speed, ball filling and charge filling. It is important that the DEM simulation's operating conditions are closely matched to the actual conditions used to operate the mill. The operating conditions used for the pilot mill simulation were based on actual tests carried out earlier by another student within the comminution group of the Centre for Minerals Research (CMR) at the University of Cape Town (Condori (2006)) . Table 4.2 shows the operating conditions used in the actual tests and DEM simulation:

4.3.3 Particle Size Distributions

The contents of the pilot SAG mill were sized using $\sqrt{2}$ series screens. This was also done outside the scope of this thesis work by another student. The particle size distributions of the mill contents (actual measurements and DEM simulation) are shown in figure 4.4.

Ball Load	5%
Total Volumetric % Filling	35%
Mill Speed (% Critical)	75%
Power Draw (kW)	11.93

Table 4.2: *MINTEK pilot SAG mill operating conditions*

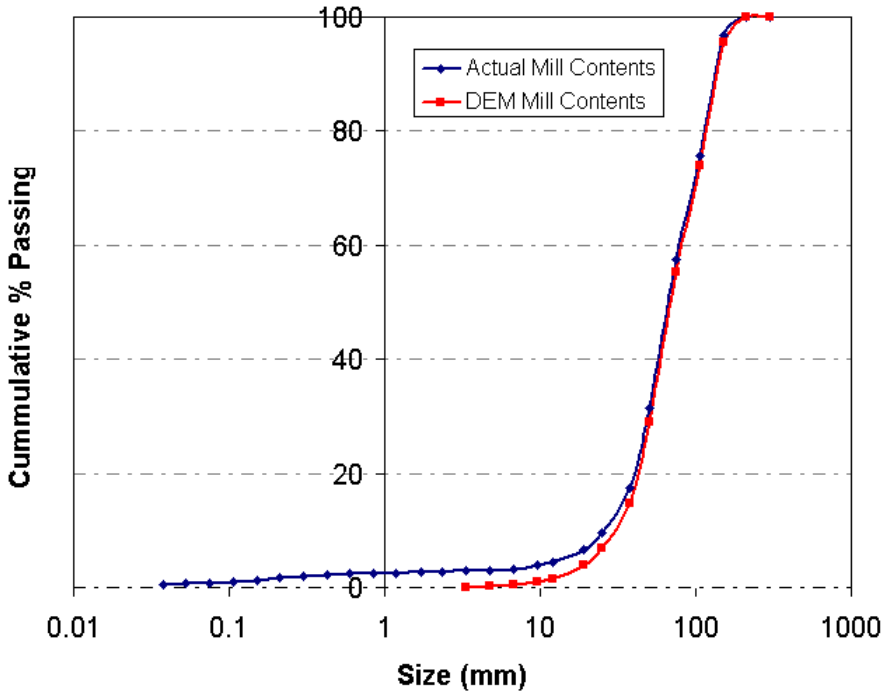

Figure 4.4: *Actual and DEM particle size distributions of the mill contents*

To calculate the number particles for each size fraction in the DEM simulation, the actual masses measured when sizing the mill contents (see Appendix A.1) were used as in equation 4.1.

$$N_i = \frac{M_i}{24\,\pi D_i^3 \rho_i^3} \tag{4.1}$$

where:

- N_i is the number of DEM particles in size fraction, i
- M_i is the measured total mass of the particles in size fraction, i
- D_i is the geometric mean size of the particles in size fraction, i
- ρ is the density of the particle

The calculated number of DEM particles from equation 4.1 was not always a whole number. This was due to the assumption that all particles were perfect spheres, and the use of one geometric mean size. The calculated number of DEM particles were rounded up to whole numbers.

The DEM particle size distributions shown in figure 4.4 are for recalculated masses based on the number of DEM particles, after rounding up, as shown in table 4.3.

Mean size (mm)	Actual mass (kg)	Mass per particle (kg)	Number of particles	Adjusted DEM No. of particles	Adjusted DEM mass (kg)	DEM Cum. % Passing
250	-	21.926	-	-	-	100.00
180	18.00	8.184	2.2	3	24.55	95.47
125	116.00	2.741	42.3	43	117.85	73.73
90	100.00	1.023	97.8	98	100.25	55.24
63	143.00	0.351	407.6	408	143.16	28.83
45	77.00	0.128	602.2	603	77.11	14.61
31.5	43.00	0.044	980.4	981	43.03	6.67
22.4	16.00	0.016	1014.5	1015	16.01	3.72
16	13.00	0.00575	2261.8	2262	13.00	1.32
11.2	2.81	0.00197	1427.1	1428	2.82	0.80
8	2.97	0.00072	4129.5	4130	2.97	0.25
5.6	1.36	0.00025	5517.1	5518	1.36	0.00
TOTAL	533.14		16,482.41	16,489	542.09	

Table 4.3: *Calculated particle size distribution (PSD) for DEM simulation*

It can be noted that the minimum size in the DEM particle size distribution ($4.75mm$) is much larger than the minimum size in the actual mill contents size distribution ($38\mu m$) in appendix A.1. It was necessary to 'cut off' the

DEM particle size distribution due to the computational limitations. The simulation time, memory, storage and post-processing time for a DEM simulation are proportional to the total number of particles. Due to the volume reduction, the number of particles drastically increases as the minimum DEM particle size decreases. For the example simulation, the total number of particles changed from $30,225$ at a DEM cut-off size of $3.35mm$, to $54,276$ at a the next smaller DEM cut-off size of $2.36mm$. The DEM cut-off size criteria used by some DEM researchers is that the cut-off size should be chosen so that at least 90% of the actual charge is represented in the DEM simulation (Morrison and Cleary (2004), Cleary (2004)). For the example simulation, the chosen DEM cut-off size of $4.75mm$ resulted in 98.51% of the actual charge being represented in the simulation, thereby satisfying the $> 90\%$ criteria.

4.3.4 Material Properties and Interaction Parameters

As discussed in the *Review of Literature* chapter, material properties used in a DEM simulation are normally measured and fitted. No experiments to measure the material properties were carried out in this thesis work. The material properties and interaction parameters used in the example simulation were obtained from the work of McBride and Powell (2006), McBride (2006) who simulated a slice of a full SAG mill treating similar ore to the one used in the MINTEK mill. Table 4.4 shows the respective values used.

4.4 Setting Up The Simulation

The following steps were followed when setting up the DEM simulation. The details were EDEM specific although the general steps can be applied to other DEM codes:

Material Properties			Interaction Parameters			
	Steel	Ore		Steel-Steel	Ore-Ore	Steel-Ore
Poisson's Ratio	0.29	0.5	Restitution Coeff.	0.5	0.3	0.5
Shear Modulus (pa)	7.5E+08	4.50E+07	Static Friction Coeff.	0.74	0.8	0.5
Density (kg/m3)	7800	2500	Rolling Friction Coeff.	0.002	0.005	0.003

Table 4.4: *Material properties and interaction parameters*

4.4.1 Step 1 - Global simulation parameters

The first step (figure 4.5) is to define the simulation parameters. The materials used in the simulation are defined by inputting their material properties (which for EDEM are the poisson's ratio and bulk modulus of elasticity) and the interaction parameters (EDEM uses the coefficients of restitution, static friction and rolling friction). Gravity is turned on by default and set to $9.81m/s^2$. The user can however switch off the gravity or set a different value and direction. For the simulation, gravity settings were left at default.

4.4.2 Step 2 - Definition of particles

Particles in EDEM were by default defined as spheres, because the implemented contact search algorithm was for spherical surfaces only. The user had a choice between creating the particles directly, importing from CAD geometry or using a template. For this thesis work, the particles were created directly. Each particle size fraction in the charge was represented by a single geometric mean sized particle. For the example simulation, four (4) steel ball sizes and eleven (11) ore sizes were defined, according to actual measured data (Appendix A.1 and A.3). A screen-shot of the particle definition stage is shown in figure 4.6.

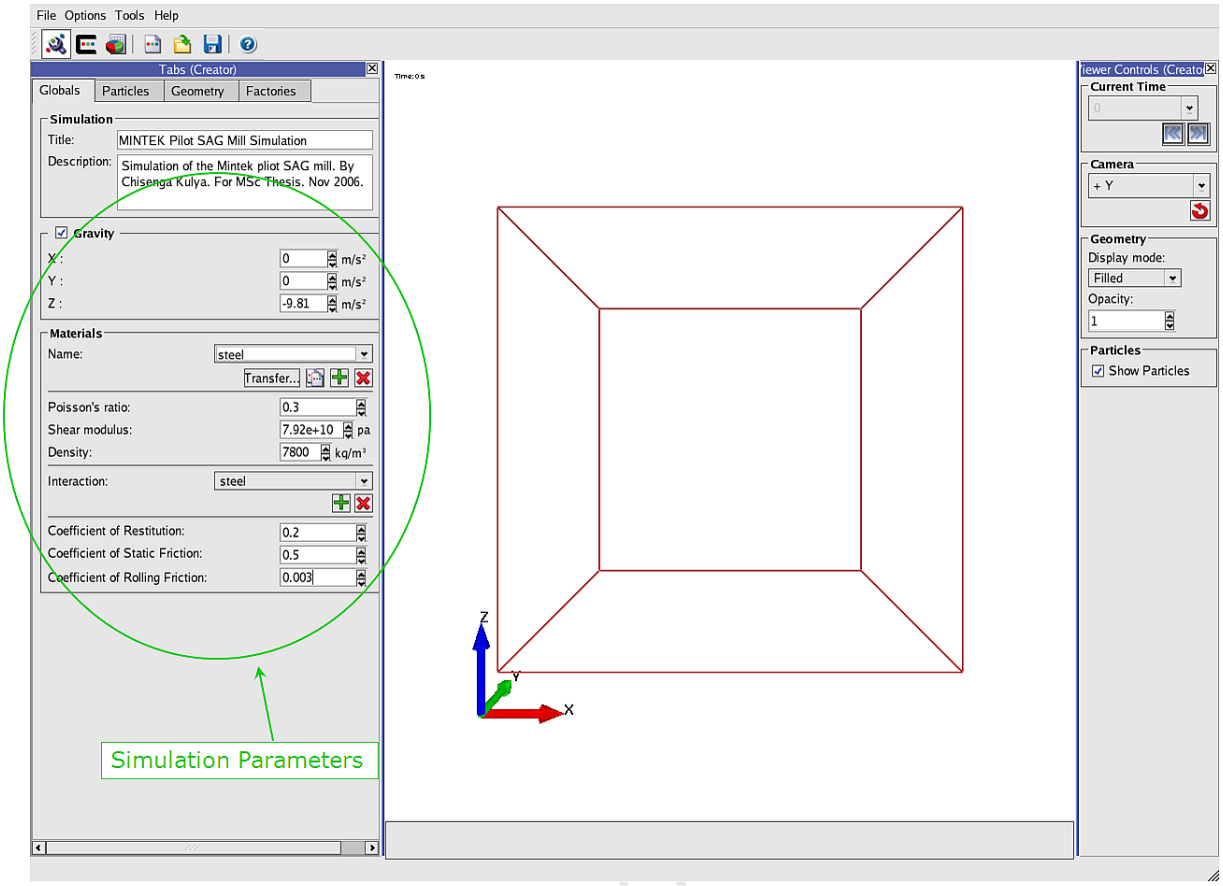

Figure 4.5: *Setting the global simulation parameters*

4.4.3 Step 3 - Geometry

Simulation geometry was imported from a CAD geometry as a volume. The geometry material was set to steel. The dynamics of the geometry (rotational speed for the mill) were also set at this stage. The actual measured speed for the MINTEK pilot mill was recorded in % critical speed while EDEM requires the speed to be set in *rads/s*. The measured rotational speed of the mill geometry was calculated by converting the measured % critical speed to *rads/s* as follows:

$$S = \frac{2\,\pi S_{cr} S_{\%cr}}{60} \tag{4.2}$$

where:

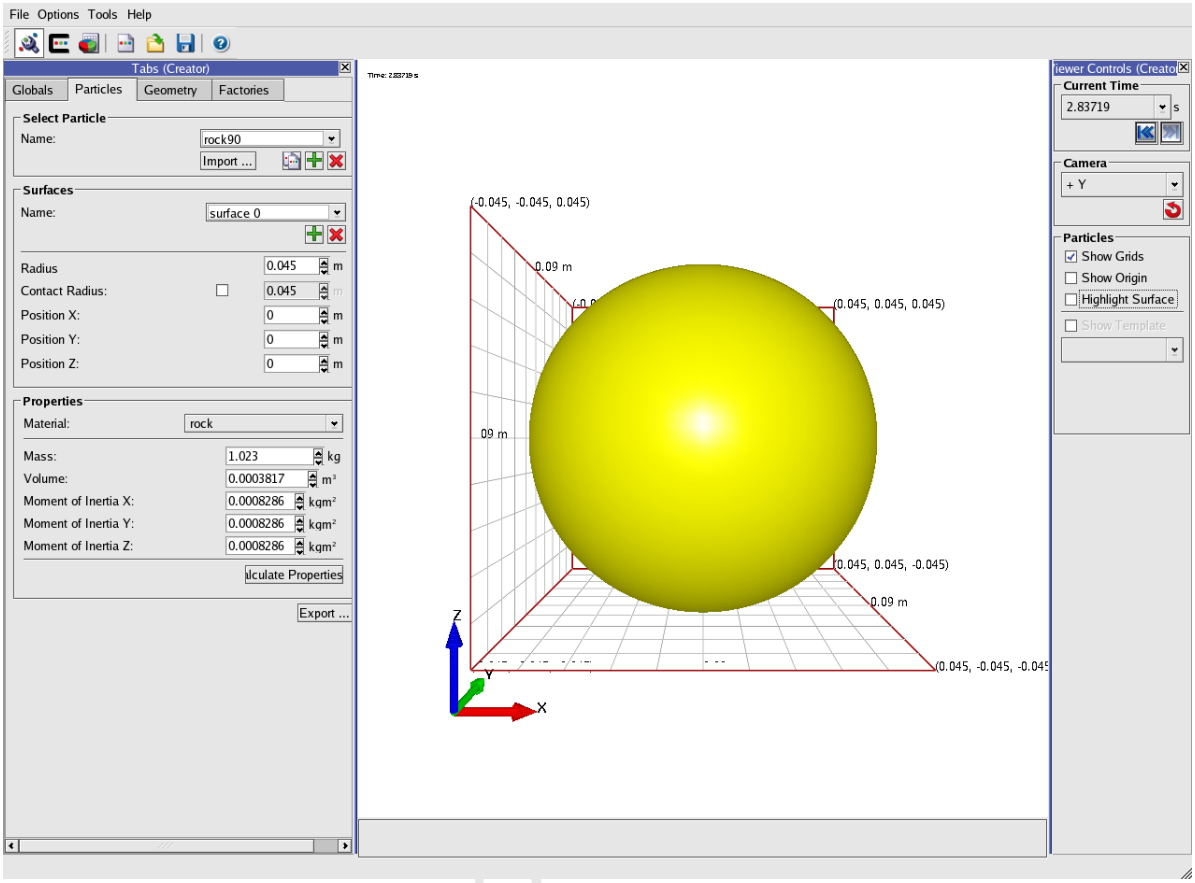

Figure 4.6: *Definition of particles*

- S is the mill speed in $rads/s$
- $S_{\%cr}$ is the actual mill operating speed (percent of critical mill speed)
- S_{cr} is the critical mill speed, in $rads/s$, which is given by

$$S_{cr} = \frac{42.3}{\sqrt{D_m}} \tag{4.3}$$

with D_m = inner diameter of the mill

The geometry speed and start time for the rotation were set as shown in figure 4.7.

4.4.4 Step 4 - Particle factories

The last simulation setup step was to define the particle factories. The important settings that were used in the particle factories' setup included the

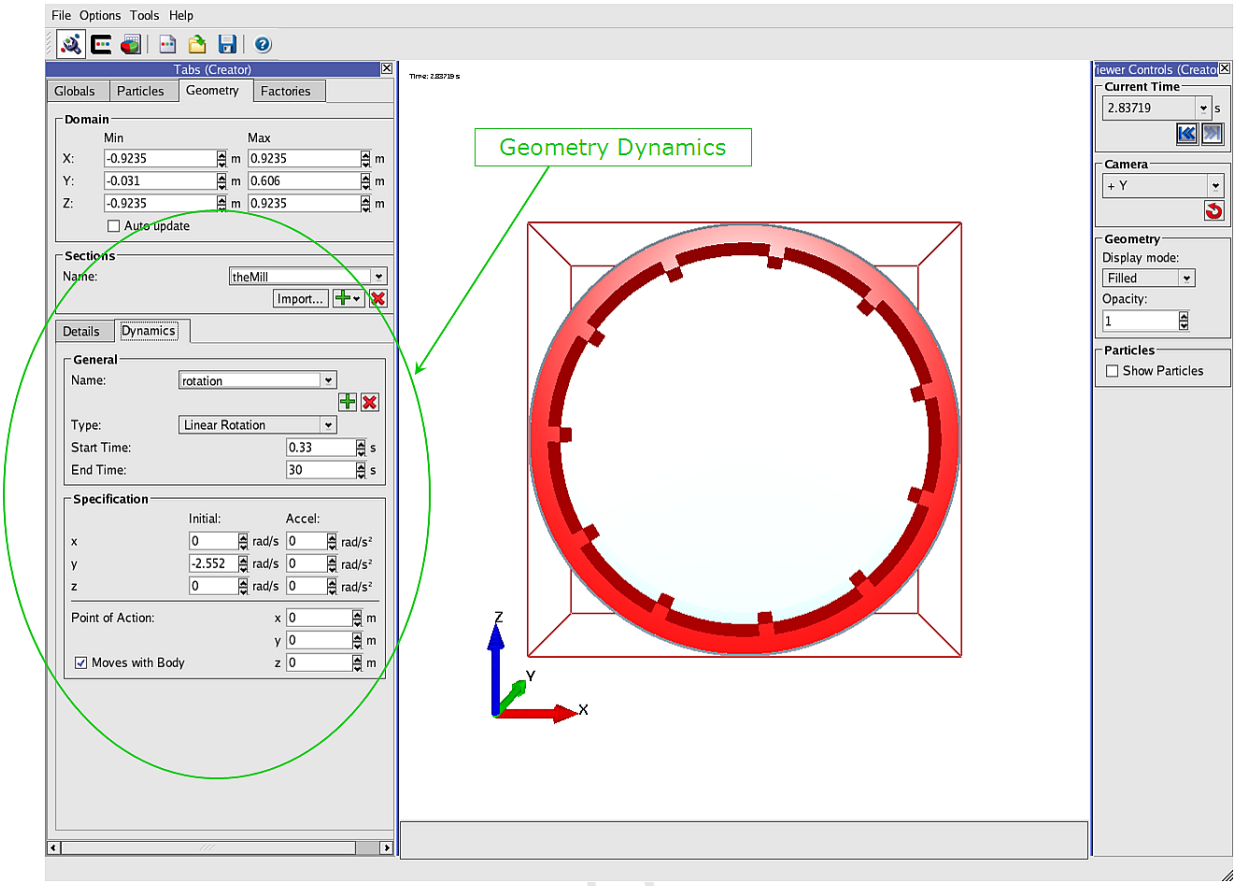

Figure 4.7: *Geometry dynamics setup*

number of particles created, creation rate, spatial position and material. A factory was created for each particle class in the mill charge. EDEM allows the user to choose between static particle creation (particles are all created at the start of the simulation) and dynamic particle creation (particles are created during the course of the simulation).

In this thesis work, both static and dynamic particle creation were explored. It was observed that static creation resulted in quicker simulation times than dynamic creation. This was because for static creation, the particles were all created at *simulation time* = 0 and the creation of particles didn't contribute to the total simulation time. On the other hand dynamic creation had a finite particle creation period which depended on the particle creation rate (maximum creation rate = $1e04$ particles/s).

The challenge with the static creation that was encountered, however was that in order to fit all the particles in the mill at one time it would be necessary to use a packing lattice (body-centred cubic, face-centred cubic or cubic) and depending on the chosen lattice, carry out detailed calculations to determine the starting co-ordinates for each particle size fraction that would ensure that all the particles fit in the mill. The other disadvantage of static creation was that it resulted in particles of each particle size fraction being all initially created in the same zone. This would in turn impact on the simulation time required before the charge was considered to be mixed.

For dynamic creation, the particles were created at random locations within the mill. The charge that was created by dynamic creation was therefore "already somewhat mixed" and therefore would result in less time to reach steady state. Packing complications could be further avoided in dynamic creation by choosing the creation rates and start times in such a manner that the larger particles were created earlier than the finer particles.

Based on the above analysis, the author decided to use dynamic particle creation, in order to promote mixing of the particles in the charge and avoid packing complications. The particles were created with a packing fraction of 0.4. Figure 4.8 shows a screen-shot of the EDEM interface for inputting particle creation parameters.

4.5 Running The Simulation

In order to ensure stability, the DEM simulation time step should be set to a fraction of the critical (or Raleigh) time step. EDEM automatically calculates and displays the Raleigh time step and allows the user to set the fraction of the Raleigh time step. For the example simulation, the Raleigh time step was $0.000203s$ and the author set the simulation time step to 4% ($8.11e06s$). The default EDEM contact model (Hertz-Mindlin (with no slip))

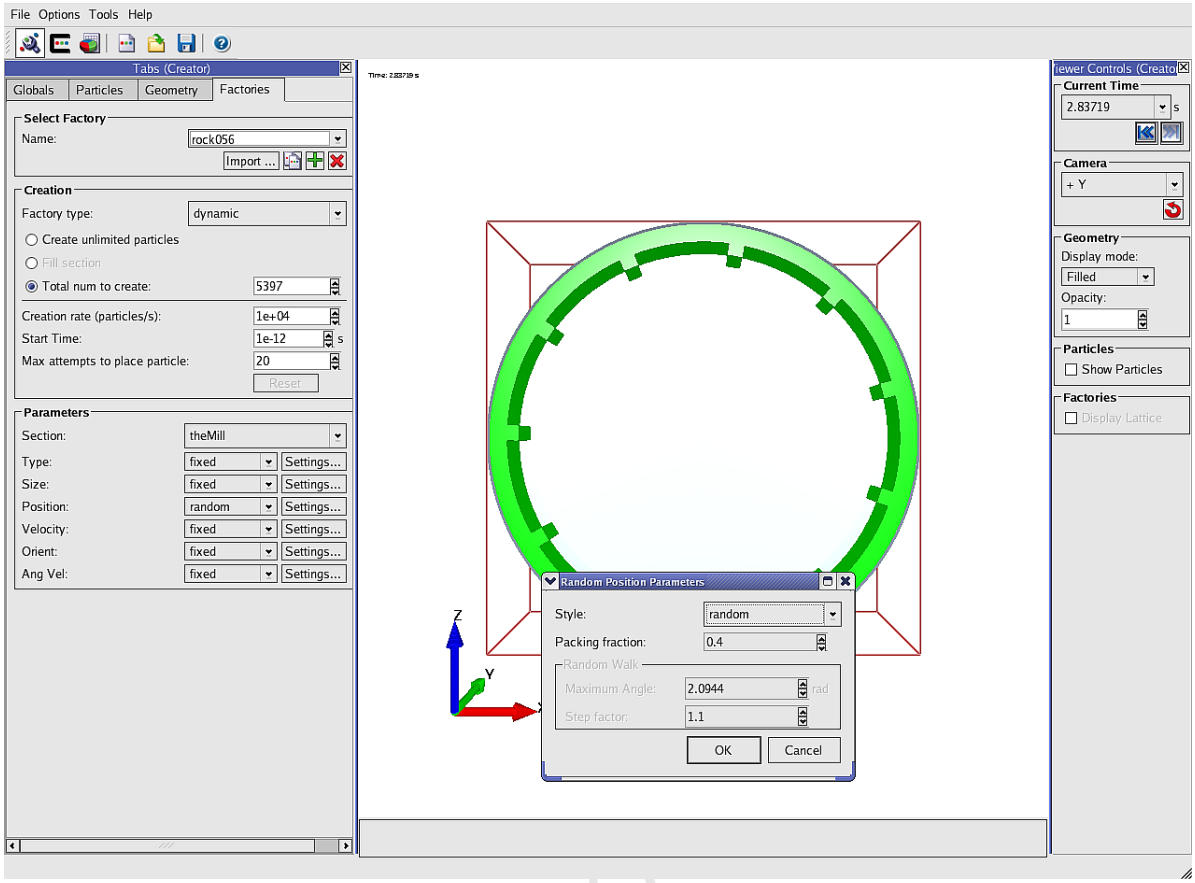

Figure 4.8: *Defining particle creation parameters*

was used.

4.5.1 Determining Steady-state

Prior to extracting the data needed for input into the tumbling mill model, the mill was rotated until the system reached steady-state. The criteria for determining when the mill has reached steady-state were not documented in the DEM literature reviewed by the author. In this thesis the following two criteria were used to determine the mill's steady-state:

4.5.1.1 Charge motion and shape

As highlighted in the previous chapter (section 2.3.1), the nature of the DEM enables fairly good prediction of the charge dynamics. The review of literature also revealed that the motion of the charge in the tumbling mill

appears to follow a cyclic pattern (Misra and Cheung (1999), Morrison and Cleary (2004)).

In developing this steady-state criteria, the observations of Misra and Cheung (1999), Morrison and Cleary (2004) were taken advantage of. The premise was that if the motion of the charge was cyclic, then snap-shots of the charge at a fixed point in the cycle could be compared to determine if steady state was reached. In this work, the Snap-shots of the charge shape were taken as the simulation progressed at intervals of every revolution and consecutive pictures were compared. It was observed that consecutive charge shape pictures became more and more similar as the simulation progressed. In comparing the charge pictures, the shape of the shoulder, toe area and the cataracting bands were looked at. By the fourth revolution, consecutive charge shape pictures didn't show significant differences (figure 4.9). It was then assumed that steady-state was reached.

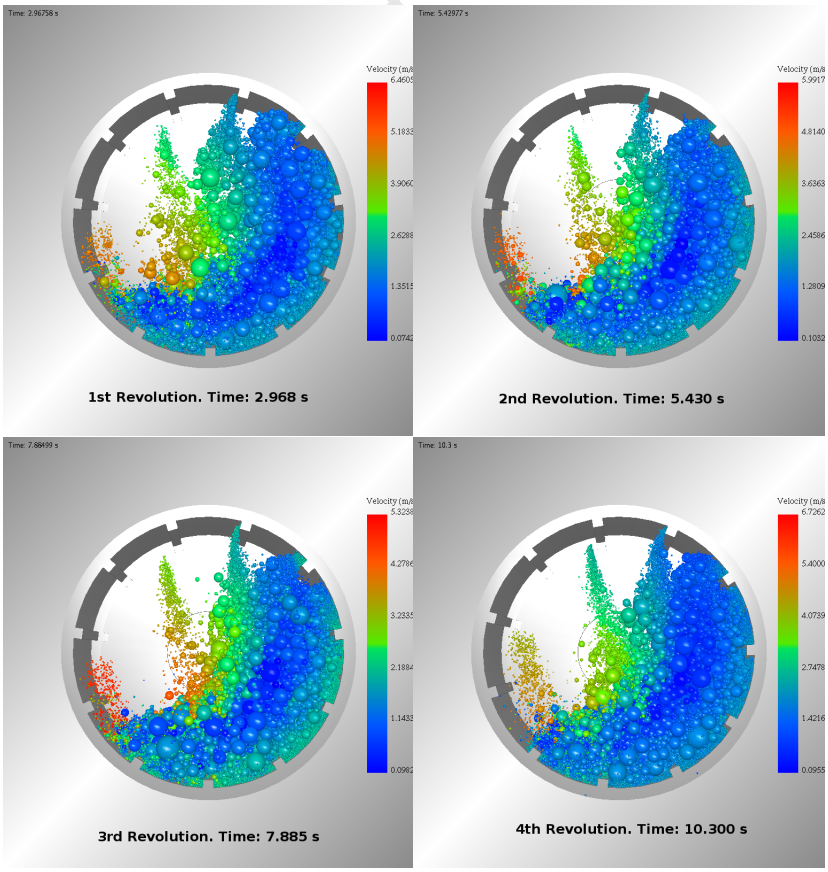

Figure 4.9: *Determining steady-state from charge shape*

4.5.1.2 Kinetic energy

The average kinetic energy of all the particles in the mill charge was also used to determine steady-state. The particle's average kinetic energy was plotted against simulation time for each full revolution as shown in figure 4.10. The first revolution showed some significant spikes and dips as the particles were created and started to tumble. In the second and third revolutions, the trend started to show a cyclic pattern. The mean value for all the instantaneous particle's average kinetic energy was also plotted for each full revolution, and increased initially, from the first to second revolution and then reduced, from the second to the third revolution. The third and fourth revolutions showed significantly similar trends in the kinetic energy vs time graphs. The mean value of the instantaneous kinetic energies was also almost the same. Based on the above observations, it was assumed that steady-state was reached after four revolutions.

For the pilot-scale SAG mill that was simulated in the thesis, it was determined that steady-state was reached after 4 revolutions. However the number of revolutions required to reach steady-state will depend on a combination all the DEM simulation parameters (mill geometry, operating conditions, particle size distributions, and material properties).

4.6 Simulation Results

Having determined that steady-state had been attained at the end of the fourth revolution, the simulation was ran for another (fifth) full revolution. The results reported in this section were obtained after the fourth revolution during which steady-state appeared to have been achieved (based on the set criterion). In this work, it was necessary to capture the collision data for the full collision period. This however resulted in increased simulation times.

4.6.1 Charge Motion and Shape

The snapshots of the charge in figure 4.9, show that the shape and profile of the charge at steady-state is clearly defined. The charge profile shows a 'horizontal' surface coming from the toe and a 'vertical' surface from the shoulder.

Figure 4.11 shows the velocity vectors of the charge as it tumbles, at an instantaneous time. In the figure the direction of the vector indicated the direction in which the particles were moving while the length of vectors rep-

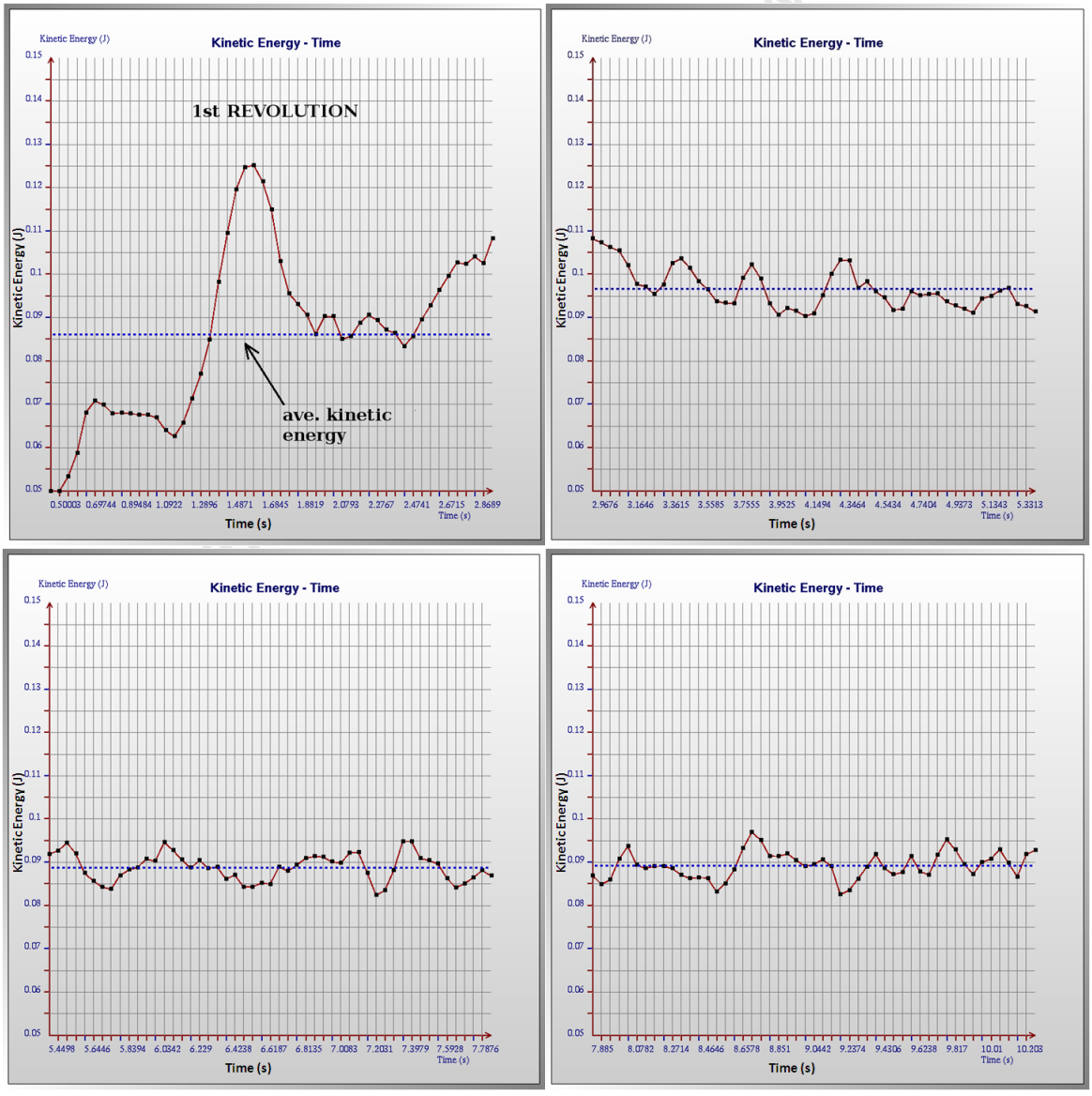

Figure 4.10: *Determining steady-state from average kinetic energy*

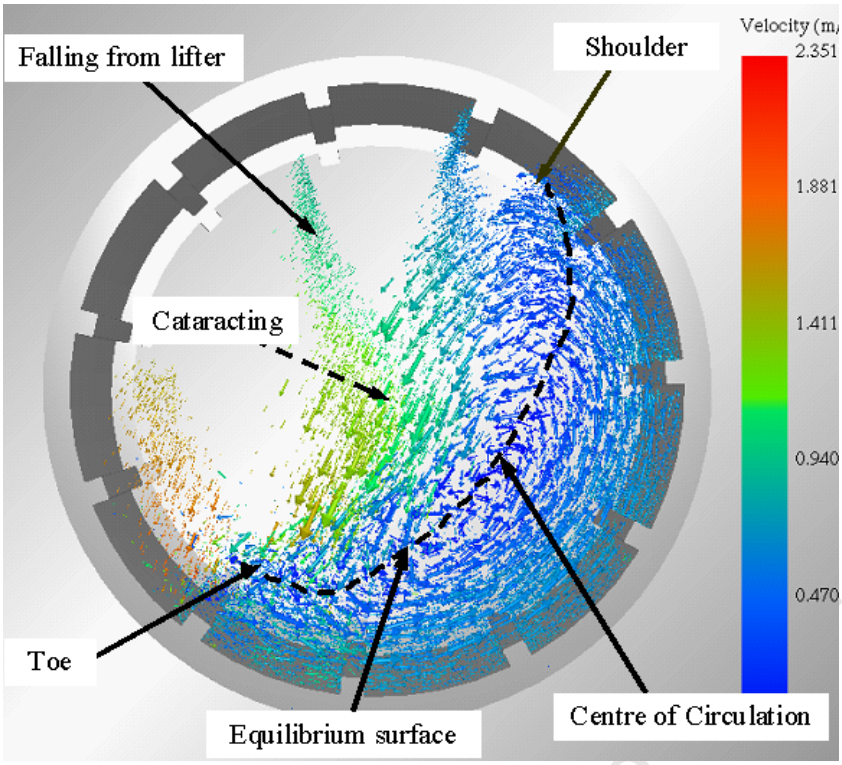

Figure 4.11: *Velocity vectors showing charge motion*

resented the mass of the particles. The colour of the vectors denoted the magnitude of the velocity of the particles according to the colour bar (from blue to red). Some of the regions of the charge (as defined by Powell and Nurick (1996), Powell and McBride (2004)) can be seen from the charge motion results in figure 4.11. The slowest moving particles (indicated by blue vectors) are around the Centre of Circulation (CoC) of the mill charge and the equilibrium surface, which separates the en masse ascending charge from the descending charge. The layer of particles in contact with the mill liners and lifters are relatively faster than the inner layers. As the particles scooped by the lifters reach the shoulder, they experience free-fall and cataract to the toe region. Some of the particles, particularly the finer particles packed on the lifter face, however continue after the departure shoulder and only fall off the lifter as its face becomes vertical. This creates distinctive bands of fine materials as they fall from progressive lifters. Most of the finer particles falling off the lifters land outside the bulk toe region, on the mill shell in an area termed the impact toe.

4.6.2 Power Draw

As highlighted in Chapter 2 (Review of Literature), the power drawn by the mill can be calculated from the charge motion predictions. In this thesis, the method of the energy expended during interactions was used to calculate the power drawn by the mill.

This method is based on the assumption that the power drawn by the mill is expended during the particle-particle and particle-geometry interactions (Datta et al. (1999)). The energy dissipated at all the collisions that occurred during the full revolution were summed up over all the collisions and time steps to obtain the total energy expended. The power draw was then obtained by dividing the duration of the revolution into the total energy as in equation 4.4, below.

$$P = \frac{\sum_{t}\sum_{k} E_{dash}}{T} \tag{4.4}$$

where:

- E_{dash} is the total energy (shear and normal) dissipated via the dashpot during a collision
- T is the total time to complete one revolution
- The subscripts, k and t represent collisions and time steps, respectively

The dashpot energy outputs obtained from the DEM software used in this work, EDEM, were in the comma-separated values (CSV) format. A script was written using the *Mathworks'* MATLAB technical computing language to read the csv files, sum up all the energies over the time steps and collisions and divide the total energy by the simulation time to obtain the power drawn. The MATLAB scripts and routines used to calculate the power draw

are given in Appendix B.1. The power draw of the pilot-scale SAG mill calculated using this method was $11.43kW$. The measured power drawn by the MINTEK pilot-scale SAG mill was $11.93kW$.

4.6.3 Energy Distribution

EDEM does not directly calculate the energy distribution of the collisions in the simulated mill, although it is able to output all the raw data needed to construct an impact energy spectrum. In order to predict the impact energy spectra of the pilot-scale SAG mill it was necessary for the author to develop procedures to post-process the raw data from EDEM using the MATLAB technical computing language.

The results of the impact energy spectra prediction were critical to the proposed tumbling mill model. It was therefore cardinal that the format and information contained in the predicted impact energy spectra were aligned to requirements of the proposed tumbling mill model.

One of the requirements of the proposed tumbling model was that separate impact energy spectra needed to be obtained for each rock size fraction in the charge. The literature reviewed by the author (Chapter 2) showed that this had not been done before, apart from Cleary (2001, 2004), who predicted separate impact energy spectra for rock-rock, ball-ball and ball-rock interactions. The requirements of the proposed model however, dictated that the impact energy spectra resulting from all the interactions involving the different rock size fractions be predicted separately. The procedures adopted in this thesis to predict these impact energy spectra is outlined in the following paragraphs.

Using EDEM's analyst tools, a text file for all the collision data, was written in csv format. This file contained the collision energies (normal, tangential and total) for the collisions experienced during the fifth revolution. The output file also contained the IDs for each of the contacting partners.

The first step was to determine the number and sizes of the histogram bins required to bin the collision frequencies. The maximum collision energy was obtained from the EDEM output file using MATLAB. The upper bound for the energy bins was set such that it includes the maximum collision energy ($\sim 18J$). For convenience reasons, the size of one energy bin was set to $0.01J$. The selection of such small energy bin ensured that even the smaller rock particle classes which are expected to experience mostly low energy collisions, had wide breakage ranges. Having set the energy bins, the collision energies were binned to get the impact energy spectra for all the collisions in the simulation for one revolution, like the one shown in figure 4.12.

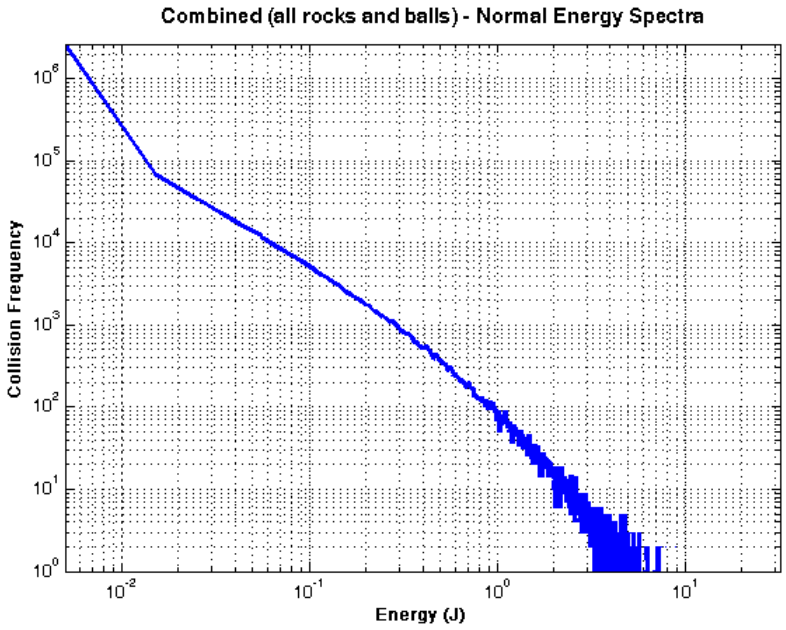

Figure 4.12: *Normal impact energy spectra for all collisions*

Figure 4.13 shows the distribution of the collision energies with respect to the cumulative number of collisions. This graph was obtained by sorting all the normal collision energies in ascending order and plotting against the number of collisions. This distribution shows that most of the collisions are at energies between $10^{-6}J$ and $1J$. There are very few collisions at the lower and higher energy bands or levels.

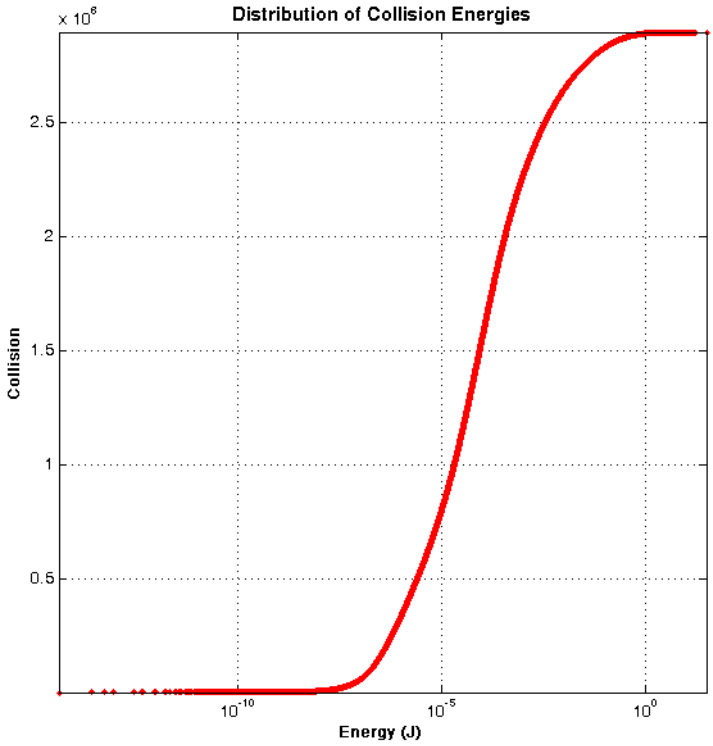

Figure 4.13: *Distribution of collisions with respect to collision energies*

The collision energy recorded in a DEM simulation is the energy lost via the numerical dashpot and represents the total energy lost by the colliding partners. The proposed tumbling mill model, however required that the DEM dashpot energy be split between the colliding partners. This again presented a problem that had never been tackled before in the literature reviewed in this thesis. Preliminary investigations in the possible mechanisms of apportioning the dashpot energy to the colliding partners revealed that several physical and material parameters like mass, velocity, stiffness, etc could be used to provide information on the most plausible split. To avoid the highlighted complication, for the work in this thesis, it was assumed that the collision energy is split equally between the colliding partners.

The EDEM output file described earlier contained data for all the collisions, involving all the various rock classes and steel balls. It was therefore necessary to find a way of distributing the collision energies according to the

different rock size fractions. In order to do this, an EDEM output file that contained the IDs for each rock fraction, was written. The collision energies obtained from the collision data EDEM output file were distributed according to the contacting rock classes. Impact energy spectra and collision energy distributions were then predicted as earlier described. A flowchart showing the steps taken to implement this scheme is shown in figure 4.14 and the MATLAB scripts are given in appendix B.1.

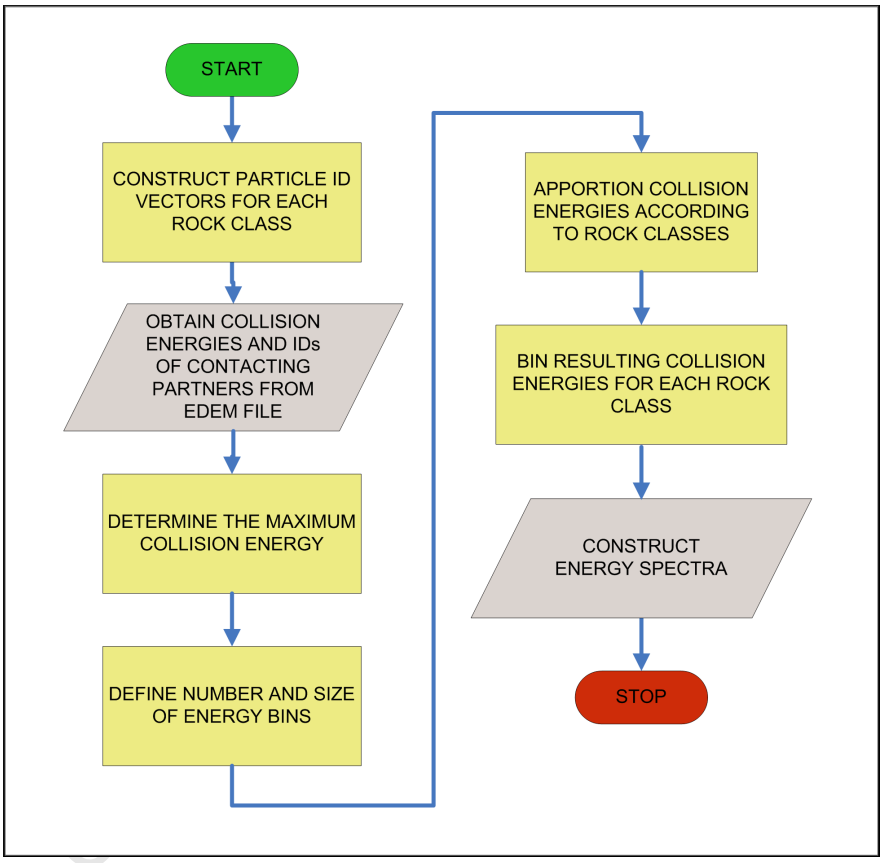

Figure 4.14: *Flowchart for construction of impact energy spectra*

The collision energy distributions and normal impact energy spectra for the 11 rock size fractions in the example simulation are given in figures 4.15, 4.16, 4.17 and 4.18.

The collision energy distributions presented above showed that the average collision energy increased with particle size. The cumulative number of collisions increased as the particle size decreased. This is expected because there were more smaller particles than the larger particles in the mill contents. The results also showed that most of the collisions occurred at

energies of less than $1J$. The smaller particles ($31.5mm$ and below) in fact did not experience any collisions at more than $1J$.

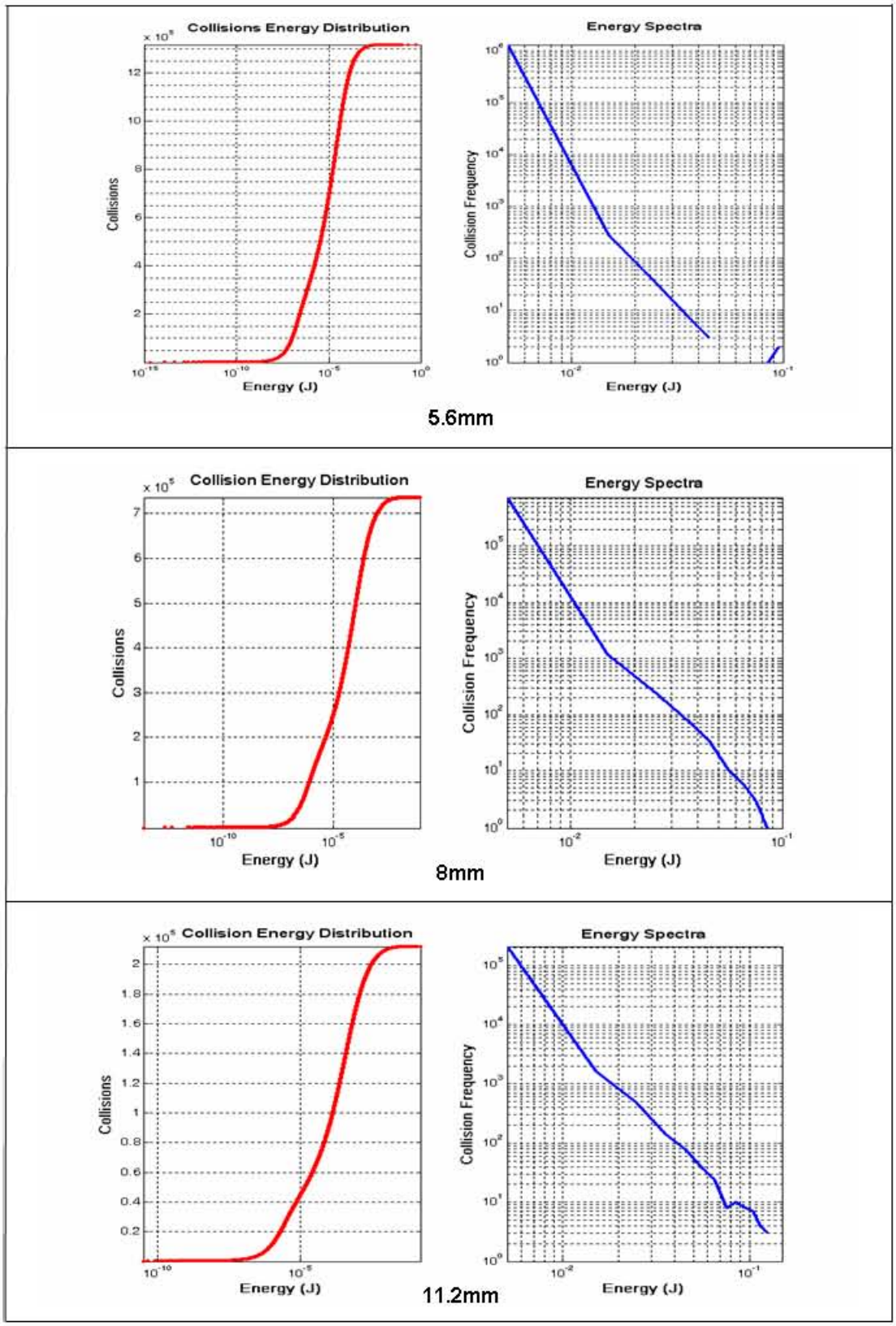

Figure 4.15: *Collision energy distributions and normal impact energy spectra for various rock classes (5.6mm - 63mm diameter)*

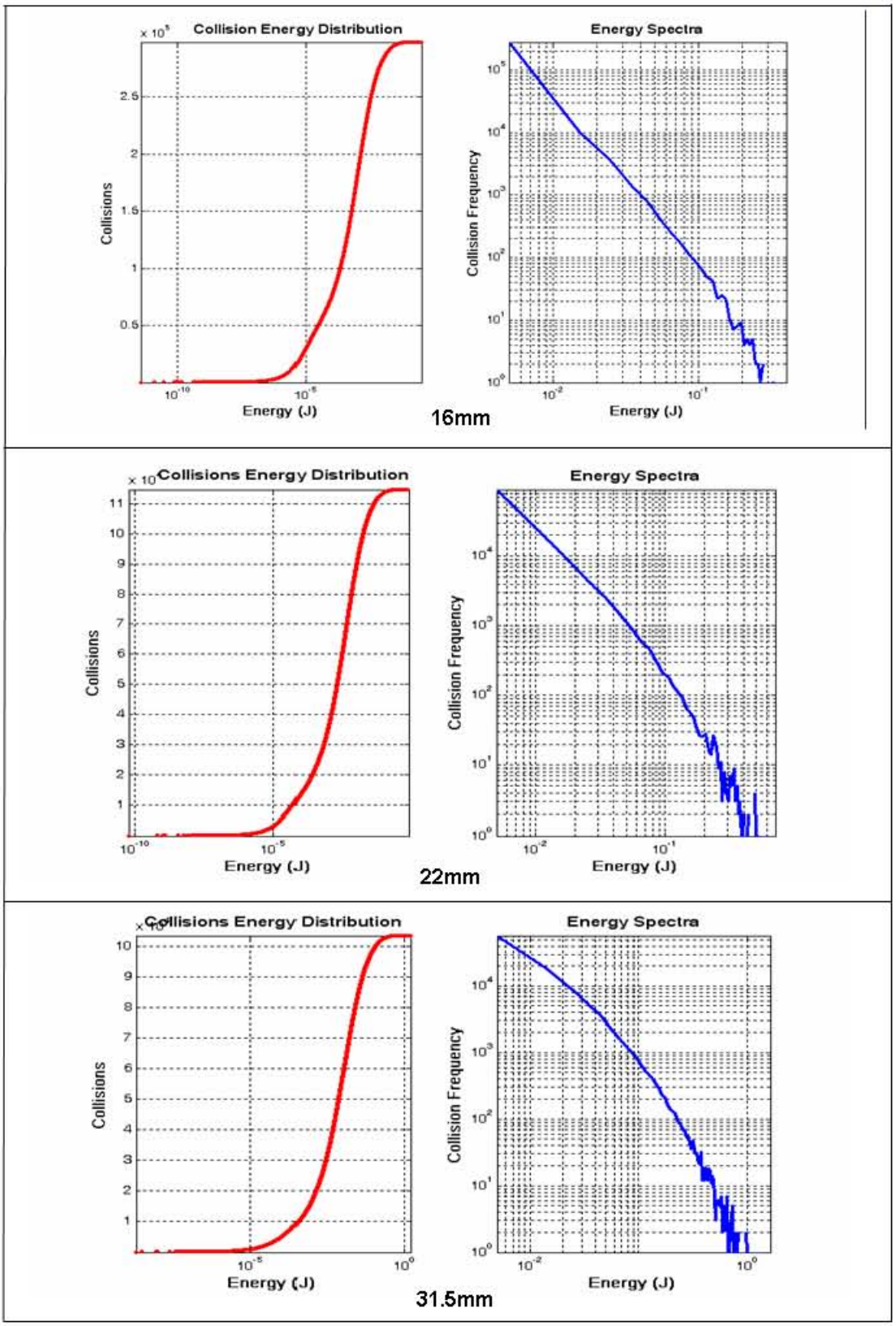

Figure 4.16: *Collision energy distributions and normal impact energy spectra for various rock classes (90mm, 125mm and 180mm diameter)*

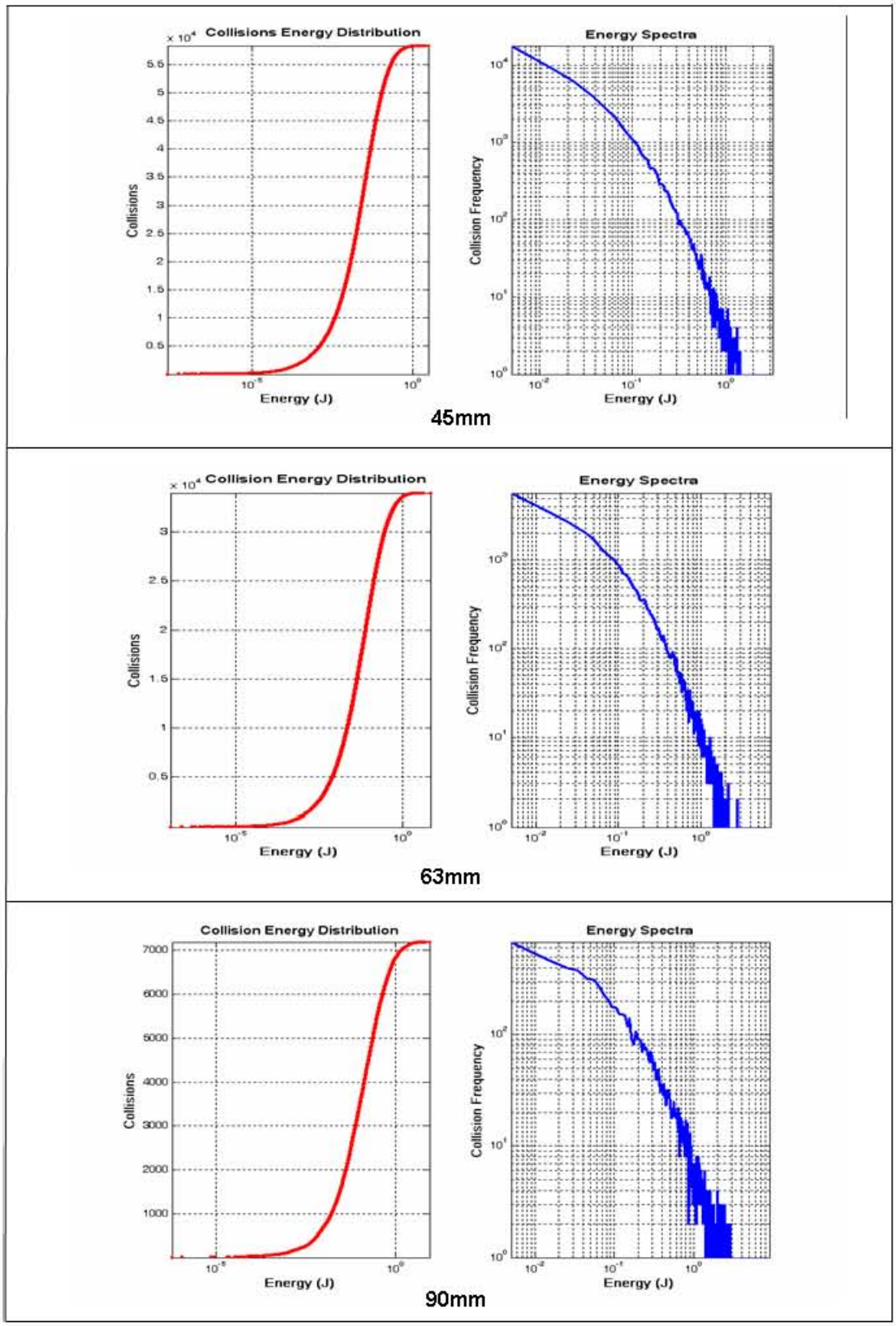

Figure 4.17: *Collision energy distributions and normal impact energy spectra for various rock classes (90mm, 125mm and 180mm diameter)*

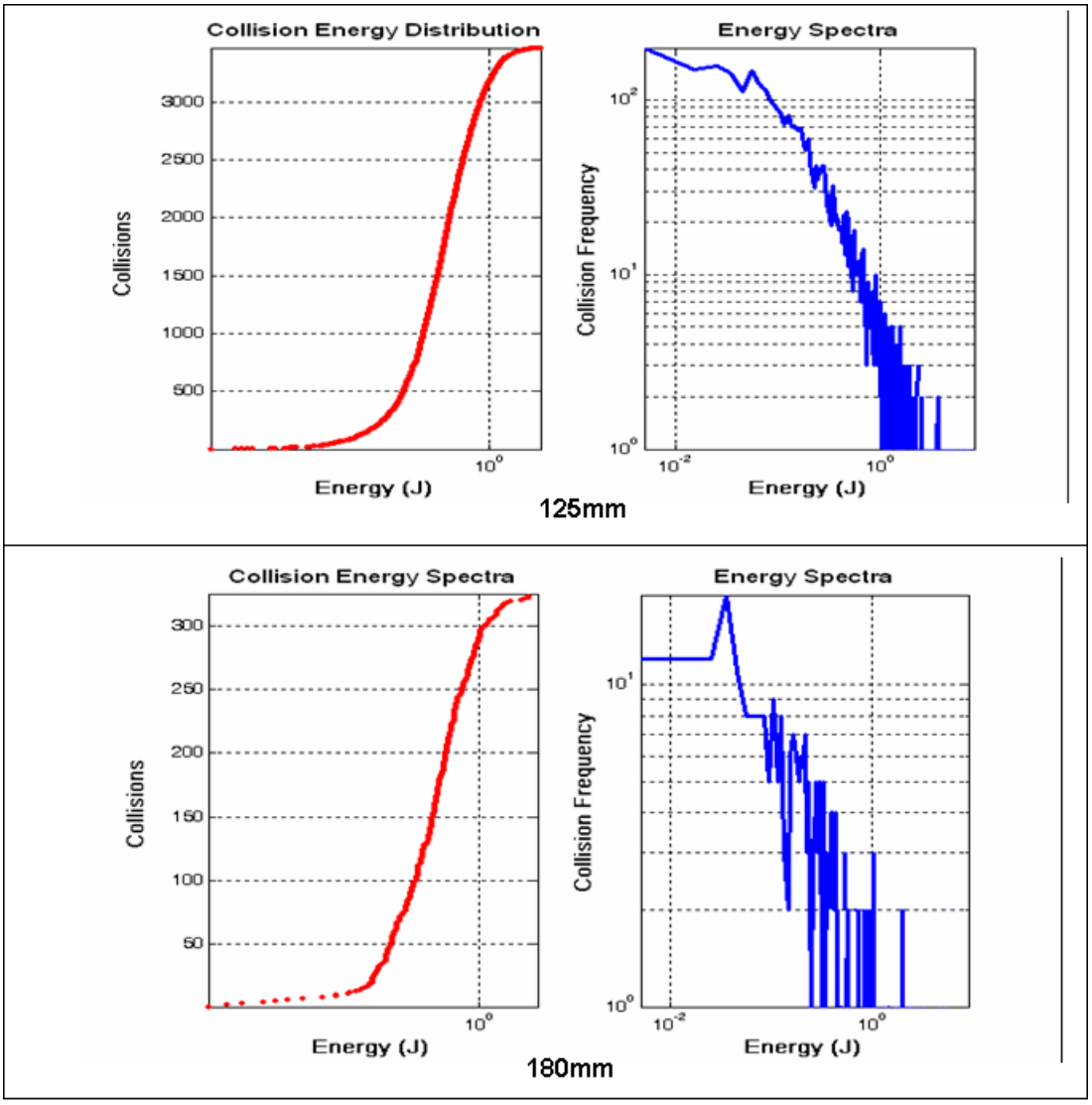

Figure 4.18: *Collision energy distributions and normal impact energy spectra for various rock classes (90mm, 125mm and 180mm diameter)*

Chapter 5

BREAKAGE EXPERIMENTS

This Chapter, presents a description of method that was employed to post-process results from Drop Weight Tester experiments for use the proposed tumbling mill model. Breakage experiments were performed on the ore trialled using the MINTEK pilot plant outside the frame of this thesis. The results from these breakage experiments were used in this thesis to obtain appearance functions for the ore.

5.1 Introduction

Breakage experiments form part of the main inputs to the proposed tumbling mill model. The breakage experiments are used to inform the model on the form of the appearance functions for the each ore particle size class in the mill charge.

As highlighted in Chapter 3 (section 3.4), the present form of breakage experiments that were available to the author (Drop Weight Tester (DWT) and Split-Hopkinson Pressure Bar (SHPB)) will require some modifications in order to be used in the proposed tumbling mill model. These modifications were not be completed in time to be incorporated in this thesis to prove the

concept of the UCM. Future stages of the UCM will incorporate the breakage data from the SHPB. Despite the lack of suitable data from the SHPB the mechanism described below to post-process the raw data from the DWT for use in the UCM program was developed.

5.2 The DWT Results

In this work, the data from the DWT experiments performed on the ore trialled using the MINTEK pilot plant was used. Table 5.1 shows the results obtained from the DWT experiment performed on the ore trialled using the MINTEK pilot. The table shows the $\%passing$ different sieve sizes for each particle class at three ECS energy levels. This is the data that is used to determine the average t_{10} values for the particle size classes.

sieve sizes *sample size* *Number of Particles* *ECS energy levels*

	63 x 53			45 x 37.5			31.5 x 26.5			22.4 x 19			16 x 13.2		
	10	10	10	15	15	15	30	30	30	30	30	30	30	30	30
	0.395	0.248	0.099	1.001	0.250	0.100	2.503	0.999	0.251	2.500	0.997	0.249	2.502	1.001	0.249
53.0	0.00	0.00	13.92												
37.5	4.56	6.20	41.33	0.00	0.00	7.77									
26.5	13.29	27.40	17.84	0.00	19.53	40.46	0.00	0.00	4.02						
19.0	25.11	21.73	9.55	2.16	23.15	20.95	0.00	3.72	20.61	0.00	0.00	16.12			
13.2	15.77	13.16	5.62	16.03	22.24	13.70	0.00	1.63	21.39	0.00	0.70	20.58	0.00	0.00	14.38
9.50	10.52	9.58	3.92	14.79	10.05	5.53	1.09	8.07	15.99	0.79	7.44	17.26	0.00	0.58	28.78
6.70	8.44	5.86	2.15	16.20	5.92	3.96	9.44	19.27	12.66	3.75	15.54	17.22	0.69	8.60	25.04
4.75	4.01	3.24	1.22	8.23	4.17	1.97	11.73	12.23	6.45	7.91	16.14	7.83	4.52	11.84	9.69
3.35	3.94	2.93	1.16	7.72	3.76	1.12	12.16	11.65	4.38	15.10	14.42	5.41	12.05	17.98	6.89
2.36	2.99	2.16	0.69	6.94	2.44	1.05	12.33	9.42	3.71	14.37	10.57	4.53	15.81	15.93	3.83
1.70	1.96	1.39	0.48	4.51	1.63	0.69	8.29	6.07	2.20	9.56	7.10	2.40	11.87	9.07	2.70
1.18	1.78	1.21	0.40	4.27	1.45	0.61	7.60	5.77	1.78	9.19	5.93	2.03	10.67	8.62	2.10
0.850	1.28	0.90	0.30	3.07	1.00	0.42	5.61	3.82	1.21	6.27	3.96	1.39	7.68	5.32	1.33
0.600	1.03	0.85	0.24	2.51	0.80	0.33	4.34	3.15	0.92	5.00	3.21	1.02	5.78	4.11	1.06
0.425	0.89	0.60	0.21	2.30	0.67	0.27	3.94	2.61	0.76	4.26	2.66	0.83	5.97	3.33	0.84
0.300	4.42	2.78	0.98	2.04	0.58	0.23	3.57	2.31	0.67	3.75	2.27	0.70	4.26	2.97	0.69
0.212				9.24	2.60	0.95	3.68	2.12	0.66	3.76	2.20	0.60	3.94	2.63	0.63
0.150							16.21	8.18	2.58	3.06	1.75	0.57	4.03	2.11	0.54
0.106										13.21	6.11	1.51	2.98	1.98	0.54
0.075													9.75	4.93	0.95
Total Wt	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Percentage Retained

Table 5.1: *Raw data obtained from the DWT experiments of target Gold ore*

5.3 Determining the Appearance Function from DWT Raw Data

The description of how the appearance functions are obtained is given in Chapter 3. The appearance functions for use in the UCM program could not be directly determined from the breakage experiments. In the UCM program the appearance functions were required to be determined for each collision energy level and particle size class. The steps below demonstrate the scheme that was developed to calculate the appearance functions from the raw results of the DWT experiments. The raw data obtained from the DWT experiments on the Target Gold Ore (table 5.1) were used. The MATLAB scripts and functions that were used for determining the appearance function are given in appendix B.2.

5.3.1 Step 1 - Determine the t_{10} vs Ecs curves

Before determining the t_{10} vs Ecs curves for each ore particle size, there was need to calculate the scaling factor for converting DEM collision energies to DWT specific comminution energies (Ecs). The author's theory about comparison between DEM and input energies is explained in sub-section 3.6 of chapter 3.

In order to determine the $DEM - DWT$ scaling, DEM simulations of the Drop Weight Test experiments were performed for a range of input energies. The DEM simulation of the DWT experiment was performed by dropping a steel sphere onto an ore particle. A schematic of the setup for DEM simulation of the DWT experiment is shown in figure 5.1. The weight and height of the steel sphere were calculated according to equation 5.1 and standard DWT experiment practice.

$$h_i = \frac{m\, E_{is}}{0.0272\, M_d} \tag{5.1}$$

where:

- h_i is the drop height
- m is the mass of the ore particle
- E_{is} is the desired specific input energy
- M_d is the mass of the steel drop particle

These DEM simulations were performed at the University of Queensland's Julius Kruttschitt Mineral Research Centre (JKMRC) and the author acknowleges the help of Dr Manoj Kanal who carried out the simulations (Powell (2008)). The DEM collision energy was recorded and compared to the input energy.

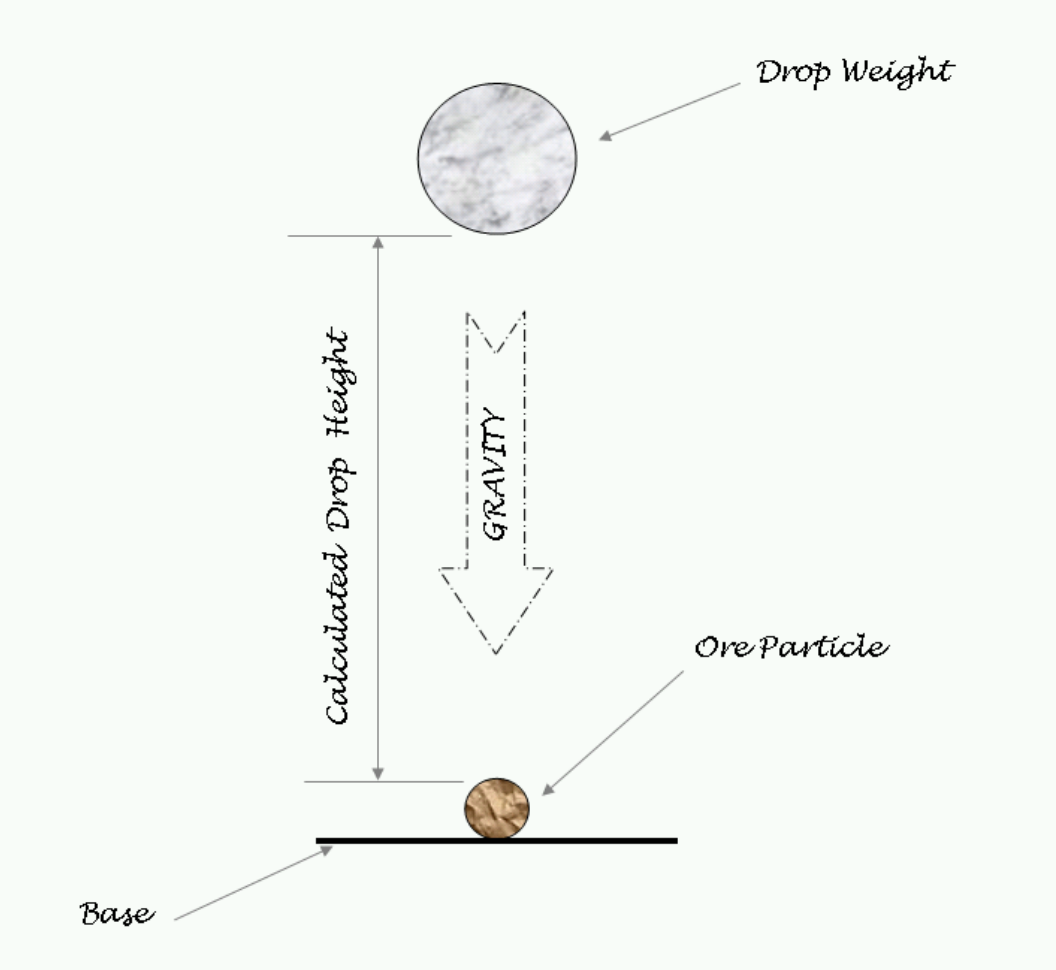

Figure 5.1: *Schematic of the DEM simulation of the DWT experiment*

Table 5.2 shows the comparison between the DEM dashpot energies and the input energies. The DEM-DWT ratio was plotted against the DEM dashpot energy as shown in figure 5.2. In order to obtain the general trend of the variation of DEM/DWT ratio to dashpot energy, the exponential function in equation 5.2 was fitted to the measured DEM/DWT data points. The collision energies that were used in the UCM calculations in this thesis work were obtained dividing the DEM/DWT ratios into the DEM dashpot energies.

$$DEM/DWT = 0.085 + 0.4 * exp\,(-0.4 * DWT) \tag{5.2}$$

Where:

- DEM/DWT is the DEM-DWT ratio
- DWT is the DEM dashpot energy

Input Energy	Dashpot Energy	DWT/ DEM Ratio	Measured DEM/DWT Ratio	Fitted DEM/DWT Ratio	Error^2	Weighted Error
8.30	2.148506	3.9	0.2589	0.2544	2.0161E-05	
21.12	3.682504	5.7	0.1744	0.1767	5.3229E-06	
86.30	8.95077	9.6	0.1037	0.0961	5.7306E-05	
219.97	17.6594	12.5	0.0803	0.0853	2.5617E-05	2.7102E-05

Table 5.2: *Comparison between input and DEM dashpot energies*

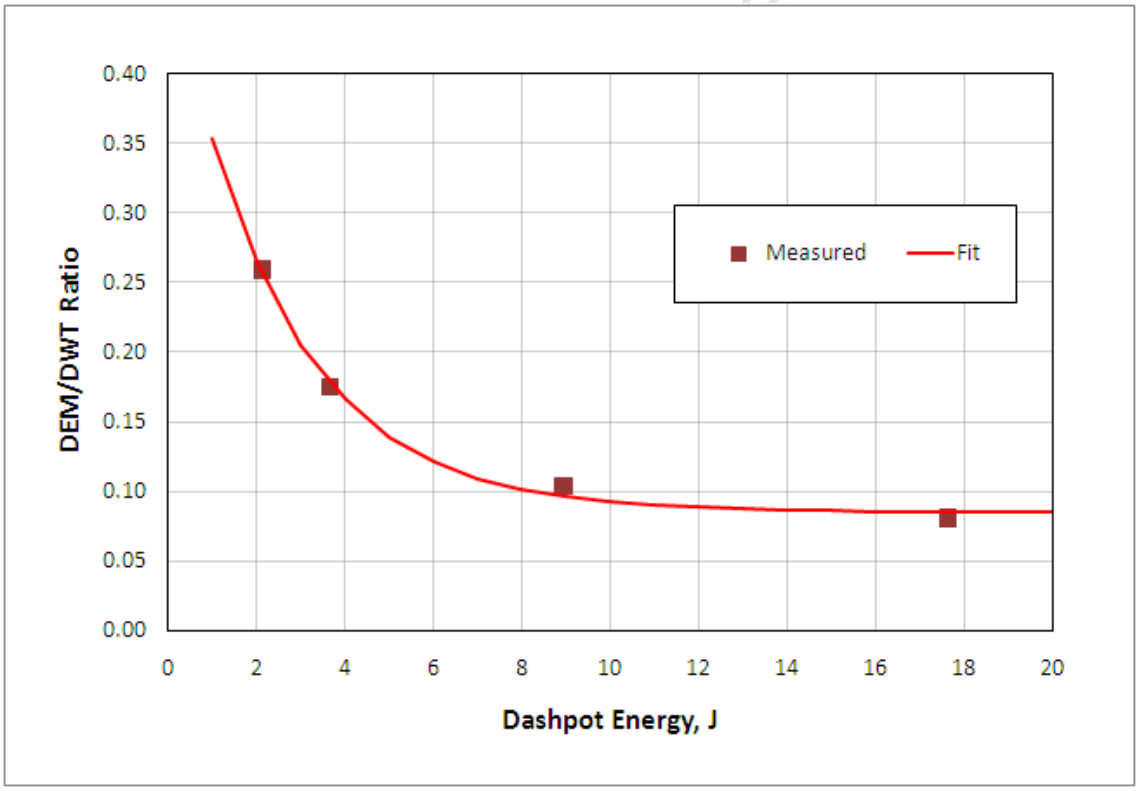

Figure 5.2: *Fitted variation of DEM/DWT ratio with DEM dashpot energy*

Having obtained the required DEM-DWT scaling factors, the next step was to get the t_{10} versus E_{cs} curves for each particle size. In this thesis the standard JKMRC breakage degree model, Napier-Munn et al. (1999), Morrison

and Cleary (2004), was used to get the t_{10} versus E_{cs} curves. The t_{10} equation (5.3 below) was fitted to the raw DWT data (table 5.1) to obtain the t_{10} versus E_{cs} curve shown in figure 5.3.

$$t_{10} = A\,\{1 - \exp(-b\,E_{cs})\} \tag{5.3}$$

where:

- A is the maximum degree of breakage
- b is a parameter of hardness
- E_{cs} is the specific input comminution energy

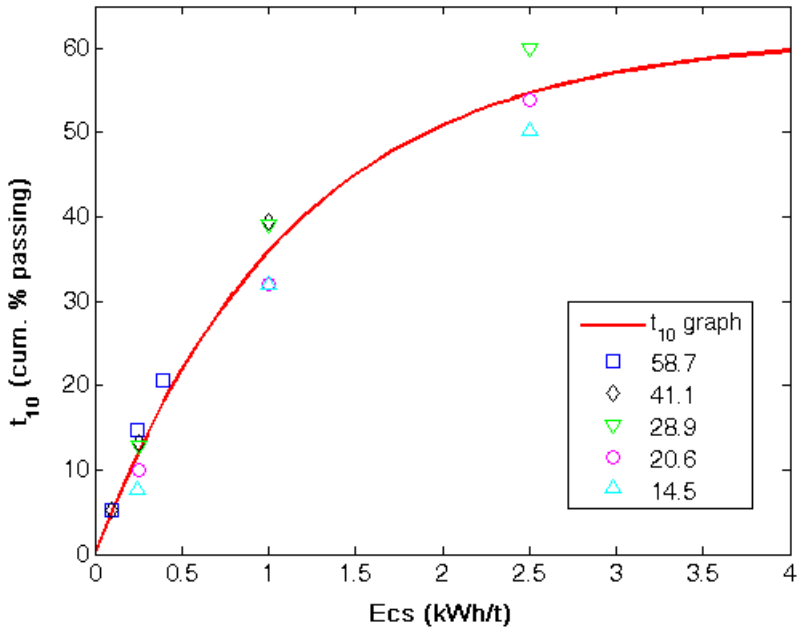

Figure 5.3: *Standard JKMRC t_{10} model fitted to Target ore DWT data*

From the t_{10} curve fit, the maximum degree of breakage, A, was 61.6027 while the value of the parameter, b, was 0.8730. The quality of the fit, R^2, obtained for the fit was 0.9716 (the quality is better as the value approaches 1). The t_{10} versus E_{cs} curves were then obtained for each particle size by using

the corresponding particle size when calculating the specific comminution energy Ecs. An example t_{10} versus E_{cs} curve for the $8mm$ particle size is shown in figure 5.4.

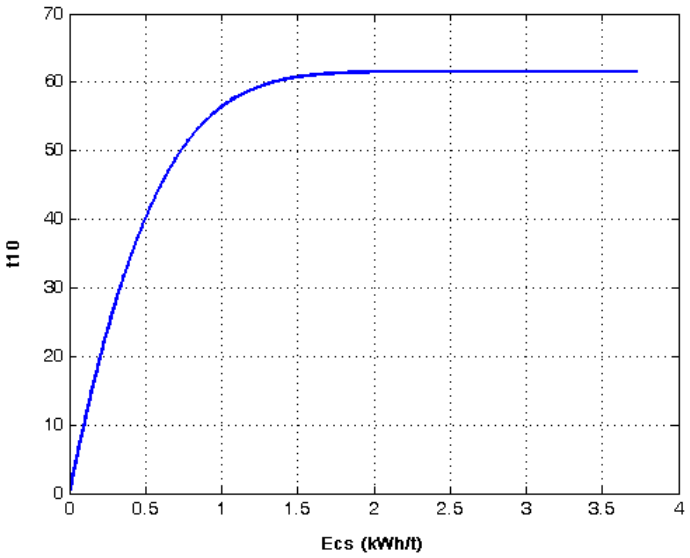

Figure 5.4: *An example t_{10} versus E_{cs} curve (for an $8mm$ particle)*

5.3.2 Step 2 - Get breakage for standard t-values

The next step after obtaining the t_{10} versus Ecs was to obtain the appearance functions at the standard JKMRC t_n sizes ($\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{10}$, $\frac{1}{25}$, $\frac{1}{50}$ and $\frac{1}{75}$ of the original particle size) as outlined in sub-section 3.6.2 of chapter 3. In this thesis, $t_{10} - t_n$ family of curves obtained from the data of Banini (2000) (shown in figure 5.5) were used.

Table 5.3 shows an example of the appearance functions at $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{10}$, $\frac{1}{25}$, $\frac{1}{50}$ and $\frac{1}{75}$ of the original particle size, obtained for an $8mm$ particle at a DEM collision energy of $1.5J$.

5.3.3 Step 3 - Calculate appearance functions for system particle sizes

The final step in the calculation of the appearance functions was to calculate the $\%$ *retained* values for all the particle sizes in the system. The Rosin-

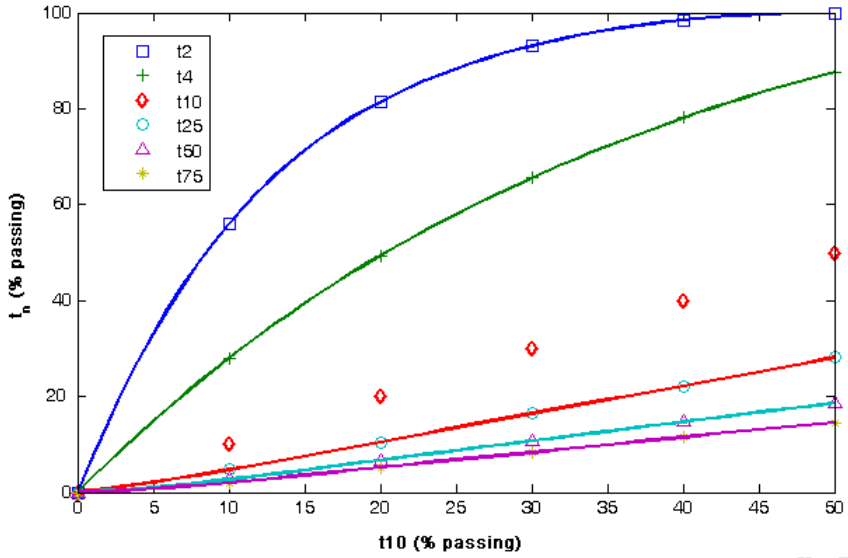

Figure 5.5: $t_{10} - t_n$ *family of curves, from the data of Banini (2000)*

Fraction of original size	Particle size (mm)	Appearance function
1/2	4	0.0067
1/4	2	0.0458
1/10	0.8	0.3542
1/25	0.32	0.2541
1/50	0.16	0.1196
1/75	0.1067	0.2196
Total Apperance function		1.0000

Table 5.3: Appearance functions at standard sizes for an $8mm$ particle

Rammler distribution function was utilised as detailed in sub-section 3.6.3 under chapter 3.

The MATLAB function for fitting the Rosin-Rammler distribution function utilised the Least-Squares-Fit method. This method proved to give very good fits although the author had to ensure that good guesses for the coefficients were used. An example of the fitted Rosin-Rammler curve for an $8mm$ particle, at a DEM collision energy of $1.5J$ is given in figure 5.6.

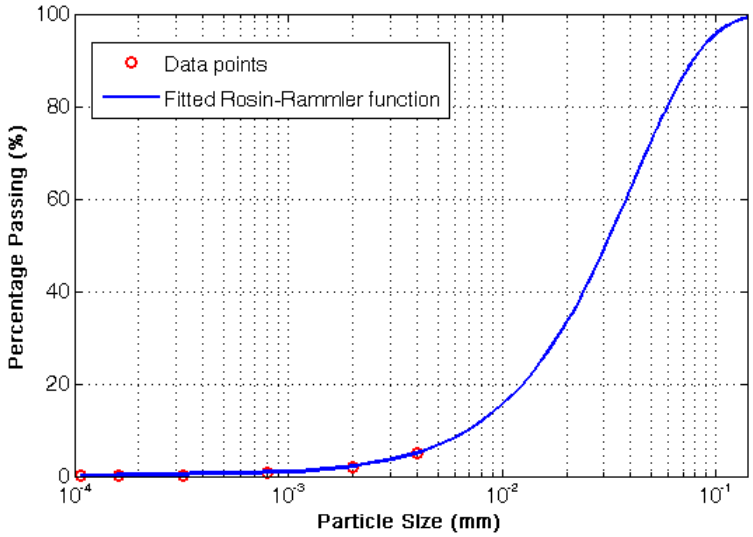

Figure 5.6: *A Rosin-Rammler function fitted to data points for an* $8mm$ *particle, at a DEM collision energy of* $1.5J$

After fitting the Rosin-Rammler function the *%-passing* values for each of the particle sizes equal or smaller than the original particle size were calculated. These were then converted into *%-retained* values and fractions of the original particle sizes which gave the appearance functions. Example appearance functions for a DEM collision energy of $0.1J$ are given in table 5.4.

The appearance functions form part of the main input into the UCM program. The MATLAB script to calculate the appearance functions was run prior to running the main UCM program. An *11x12x900* matrix containing appearance functions for each particle size at each collision energy level was saved in a *MAT* file.

	Fraction of original particle for each size											
Original Particle Size (mm)	180mm	125mm	90mm	63mm	45mm	31.5mm	22.4mm	16mm	11.2mm	8mm	5.6mm	-5.6mm
180	0.9999	5.47E-05	3.17E-05	2.23E-05	1.35E-05	9.22E-06	5.6E-06	3.4E-06	2.7E-06	1.5E-06	1.0E-06	1.8E-06
125	-	0.9996	1.50E-04	1.05E-04	6.39E-05	4.36E-05	2.67E-05	1.58E-05	1.30E-05	7.08E-06	4.83E-06	8.40E-06
90	-	-	0.9988	4.29E-04	2.60E-04	1.77E-04	1.09E-04	6.45E-05	5.28E-05	2.88E-05	1.97E-05	3.42E-05
63	-	-	-	0.9966	0.0012	8.14E-04	4.99E-04	2.96E-04	2.42E-04	1.32E-04	9.03E-05	1.57E-04
45	-	-	-	-	0.9906	0.0034	0.0021	0.0012	0.001	5.57E-04	3.80E-04	6.61E-04
31.5	-	-	-	-	-	0.973	0.0094	0.0056	0.0046	0.0025	0.0018	0.0031
22.4	-	-	-	-	-	-	0.9266	0.0231	0.0193	0.0107	0.0073	0.013
16	-	-	-	-	-	-	-	0.8262	0.0638	0.0369	0.0261	0.047
11.2	-	-	-	-	-	-	-	-	0.543	0.1287	0.1049	0.2234
8	-	-	-	-	-	-	-	-	-	0.1936	0.1565	0.6499
5.6	-	-	-	-	-	-	-	-	-	-	0.021	0.979

Table 5.4: *Example appearance functions for an impact energy of* $0.1J$

Chapter 6

PREDICTING THE MILL PRODUCT

This Chapter demonstrates how the proposed tumbling mill model discussed in Chapter 3 can be used to predict the mill product size distribution. The data from the DEM simulation presented in Chapter 4 and breakage experiments performed on the ore trialled using the MINTEK pilot plant given in Chapter 5 *was used in this demonstration. A step-by-step description of the computer program used to execute the model is also presented.*

6.1 Introduction

Having formulated the structure of the proposed tumbling mill model, outlined in Chapter 3, a computer program for the proposed model was developed. The MATLAB technical computing language was chosen as the programming language for the model program due to its user-friendliness, ease of handling formulas and flexibility. Once a program is written in MATLAB it can easily be compiled into a stand- alone executable, thereby enabling it to be run on any computer.

The work in this thesis on the development of this proposed tumbling mill model was envisioned as one of the first steps towards the development of the Unified Comminution Model (UCM) under the AMIRA sponsored, P9 project. The computer program developed in this thesis is therefore the first step towards the development of the UCM proposed by Powell (2005). The tumbling mill model and program presented in this thesis will be developed further in the P9 and other associated project to achieve the unified comminution model which will have capabilities to predict the performances of all comminution devices (Powell (2005)). The name, "UCM", is however used to refer to the tumbling mill model proposed in this thesis, for the sake of convenience and as a reminder that the overall objective of the AMIRA project under which this work was performed is to develop a model that can apply to all comminution devices.

Due to the absence of appropriate experiments that can measure breakage resulting from surface interactions, the developed model assumes that breakage occurs only due to normal or bulk interactions. The following sections describe the UCM program and at the same time gives a demonstration of using the program to determine the mill product for the MINTEK pilot SAG mill[1].

6.2 Overview of the UCM Program

The general flowchart of the UCM program, which applies to one ore particle class, is as shown in figure 6.1.

The program receives its main inputs from the DEM simulation of the mill and breakage experiments of the ore tested at MINTEK. Other inputs into the program include the measurements of the contents of the mill and the feed.

[1] See Chapter 4

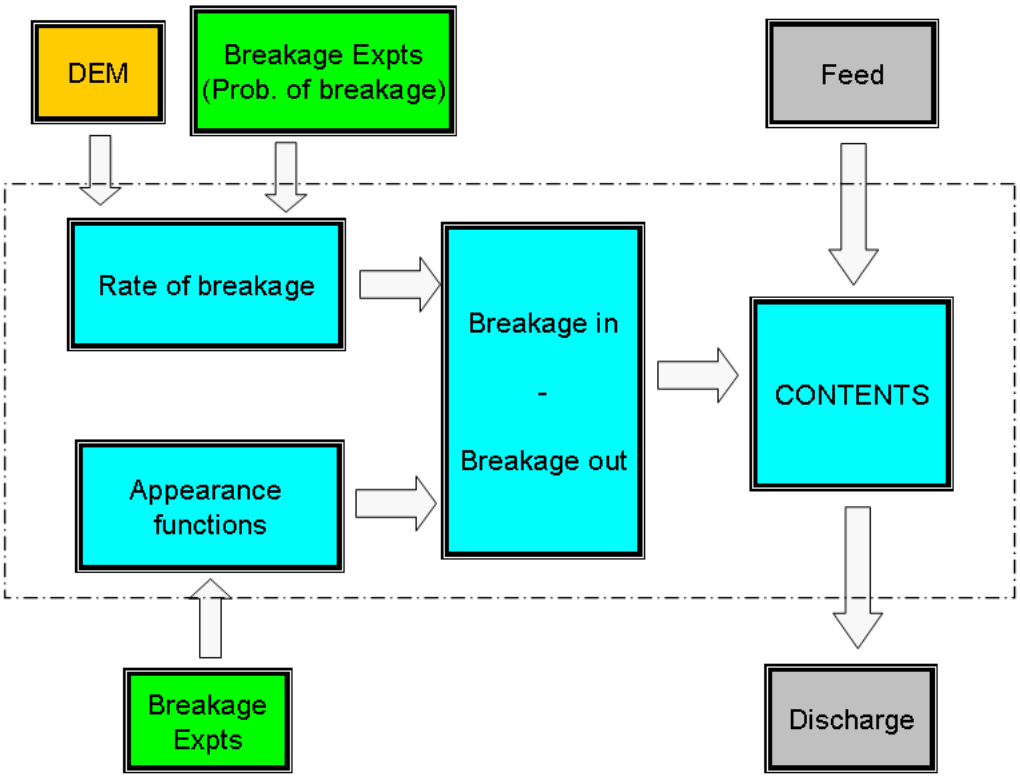

Figure 6.1: *General flow chart of the UCM program*

In figure 6.1, the area enclosed in dotted lines denotes the tumbling mill model process. Results from the DEM simulation (in particular the impact energy spectra) are combined with the probability of breakage derived from breakage experiments to obtain the rate of breakage. Information from breakage experiments is also processed to get the appearance function. The breakage-in and breakage-out is calculated from the rate of breakage and the appearance function according to equation 3.1. The feed is added and the product is discharged via a discharge mechanism.

6.3 Inputs

6.3.1 DEM Simulation

The DEM simulation of the mill to be modelled must be performed before the model is run because outputs from the DEM simulation are part of the required UCM model inputs. The DEM simulation of the MINTEK pilot SAG mill described in Chapter 4 was used in the UCM simulation discussed in this chapter. The Impact Energy Spectra obtained from the DEM simulation (section 4.6.3) is input into the UCM program. The Impact Energy Spectra for the MINTEK pilot SAG mill are given in figures 4.15, 4.16, 4.17 and 4.18.

Post-processing of the DEM raw particles' interaction data to obtain the Impact Energy Spectra is performed using MATLAB as described earlier (code is given in Appendix B.1) and the results saved as binary data in a MATLAB "*mat*" file. One important thing to note here is that the size and number of the collision energy intervals to be used in the UCM program, needs to be known before running the MATLAB program that outputs the Impact Energy Spectra (900 intervals of $0.01J$, for the example simulation).

The Impact Energy Spectra data in the "*mat*" file was stored as one matrix measuring 11 x 900. The matrix contained the number of interactions at each collision energy level (columns) for each of the rock particles in the simulation (rows). This is actually the raw data used to construct the Impact Energy distributions shown in figures 4.15 and 4.16. Table 6.1 shows the first 10 columns (collision energy levels) of the Impact Energy Spectra matrix obtained from the example simulation.

12	12	12	19	11	8	8	8	8	5
198	148	155	143	111	146	125	112	100	89
679	464	396	378	320	314	265	226	204	177
5760	3478	2679	2212	1900	1558	1291	1162	1051	921
17751	8558	5881	4228	3222	2549	2146	1673	1340	1169
56944	17135	8624	5209	3527	2373	1734	1327	1023	861
90777	11986	4594	2440	1466	899	597	459	314	210
280498	10573	3462	1413	760	402	252	174	117	87
209553	1678	459	149	81	40	24	8	10	8
734836	1165	255	83	35	11	6	3	1	0
1316330	284	35	8	3	0	2	0	1	2

Table 6.1: *Impact Energy Spectra data matrix for the example simulation, showing the first 10 columns*

6.3.2 Breakage Experiments

Results from breakage experiments are used to inform the model on the form of the appearance functions for the each ore particle size class in the mill charge. Chapter 5 described the method that was used in this work to obtain appearance functions from the Drop Weight Tester experiments performed on the ore trialled using the MINTEK pilot plant.

6.3.3 Other Inputs

Apart from the above inputs from the DEM simulation and breakage experiments, the UCM program requires information about the feed coming into the mill (feed size distribution and its rate) and the particle size distribution of the contents. The particle size distributions for feed and contents for the example simulation are presented in Appendices A.2 and A.1 respectively.

The UCM program also required other minor inputs which are shown in figure 6.2. The *"number of DEM revs"* referred to the number of mill revolutions that were performed when the collisions data was being captured.

The *"Sim. Time Step (s)"* was the number of duration for the UCM program time step.

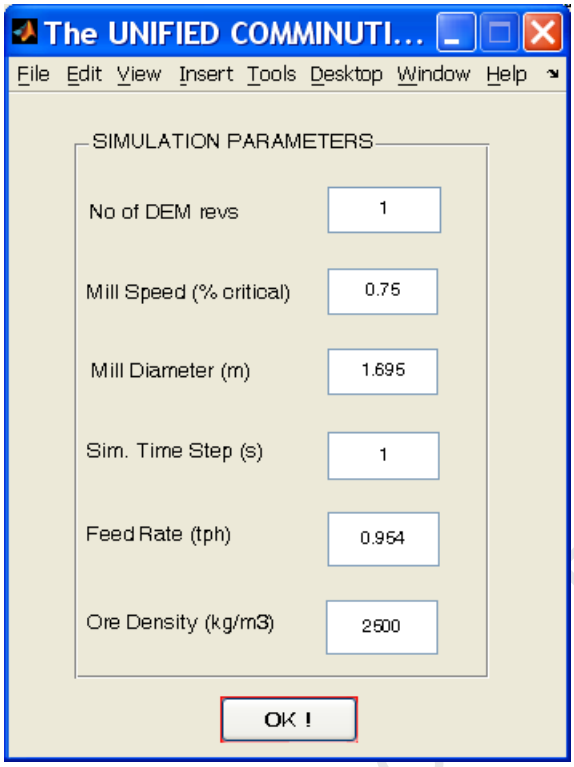

Figure 6.2: *The UCM program input screen*

6.4 The UCM Process

Before running the MATLAB script for the actual UCM program, the user was required to have all the inputs ready. The MATLAB scripts for determining the energy spectra and appearance functions, therefore had to be run prior to running the main program. The scripts output the results in the form of *MAT* files containing energy spectra and appearance functions, respectively which are loaded into the main program. The MATLAB script and functions for the main UCM program are presented in B.3.

The execution process of the program is shown in figure 6.3. With all the required inputs in-place, the next step is to apply the proposed tumbling mill model equation (3.1) to determine the breakage of particles in the DEM size range (the three processes enclosed in dotted lines). After that, the

breakage from the sub-DEM particles is determined. The "DEM" and sub-DEM breakages are combined and a discharge function is then applied to take out the product. The feed is finally added to the mill contents to obtain the contents after one complete time-step and before another iteration of the UCM is applied.

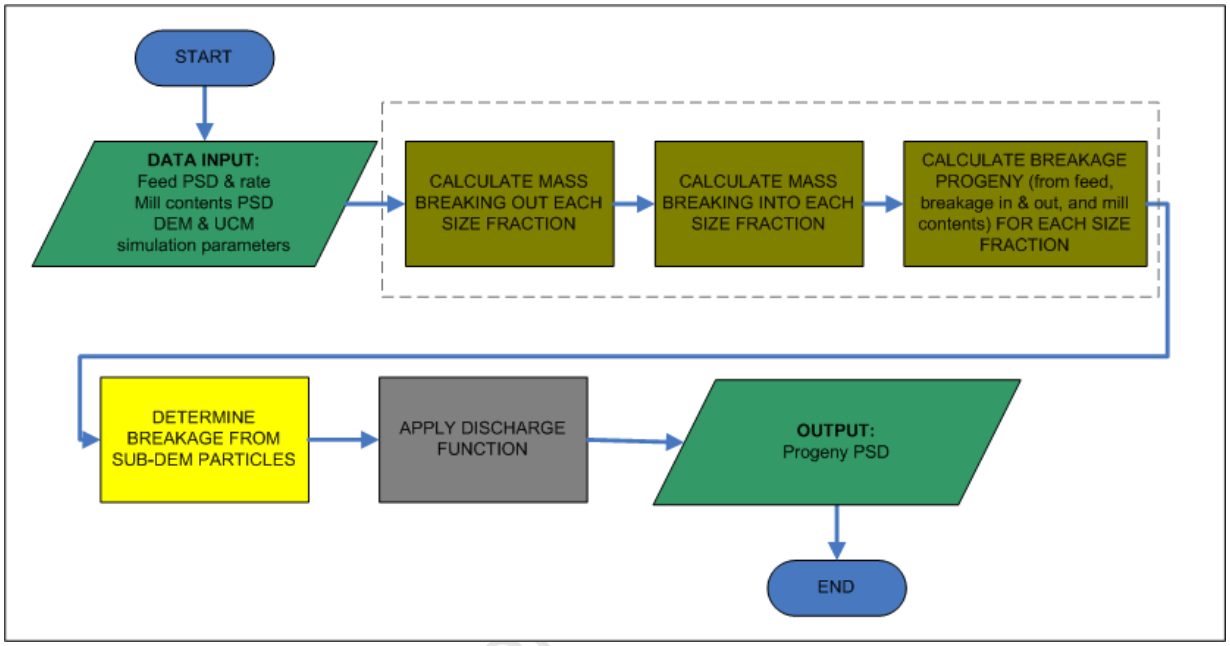

Figure 6.3: *Flowchart of the UCM process*

6.4.1 Breakage In and Out

The calculation of the amounts of mass that is broken-in and broken-out of a particular particle size class forms one of the cardinal components of the proposed tumbling mill model. Figure 6.4 shows the flowchart for the determination of the breakage-in and breakage-out.

In calculating the amount of breakage, three energy classes that could lead to breakage were recognised. The energy classes discussed earlier in sub-section 3.5 of chapter 3, namely *no breakage*, *incremental breakage* and *1-hit breakage* were used.

As shown in the flowchart in figure 6.4, for each particle size and bin energy the process of determining breakage was as follows:

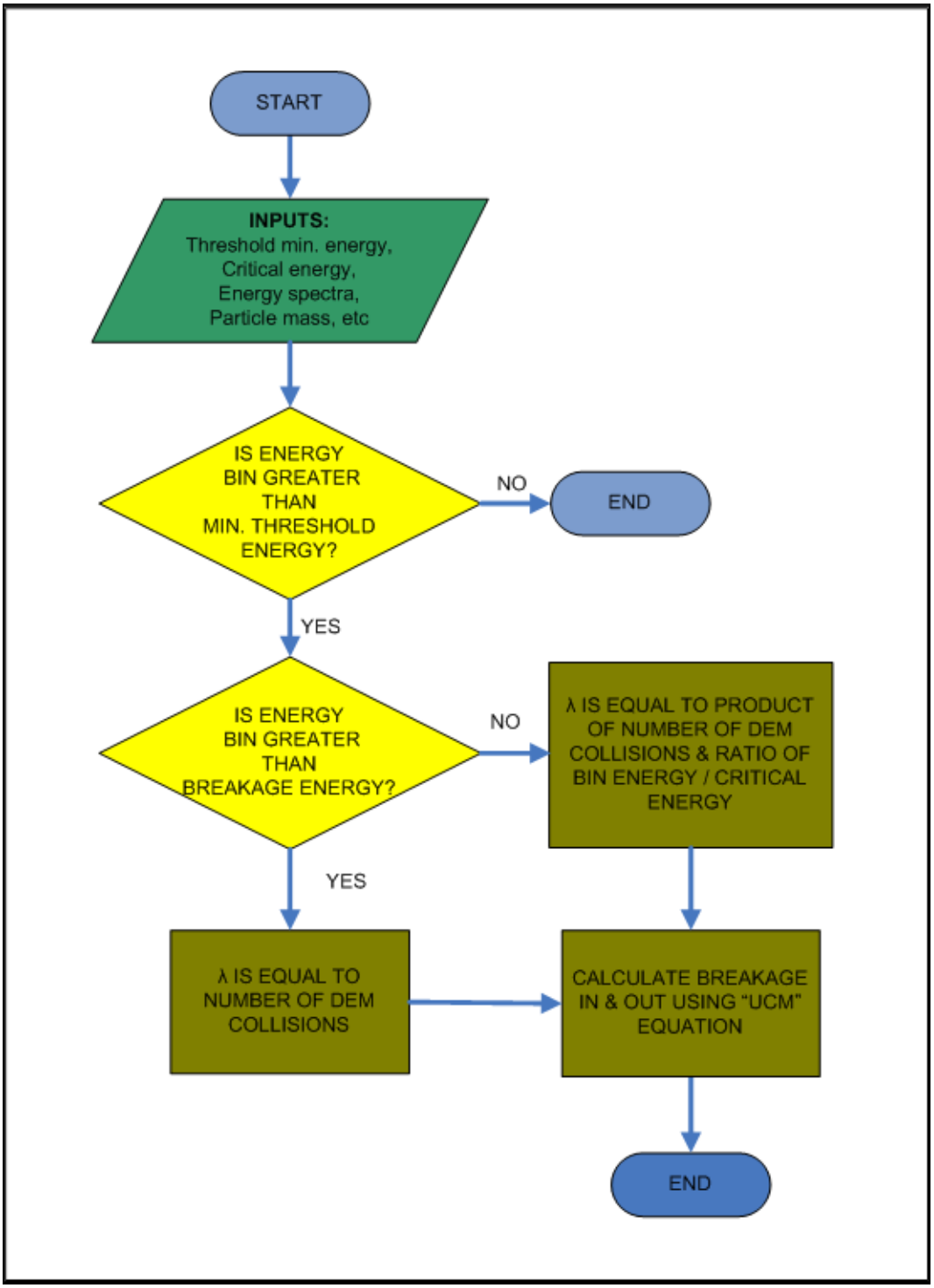

Figure 6.4: *Flowchart for the determination of breakage-in and breakage-out*

6.4.1.1 *No breakage*

The bin energy was compared with the threshold breakage energy. In this thesis work, the threshold breakage energy was set to *2.8e-5kWh/t*. If the

bin energy was smaller than the threshold breakage energy, it was assumed that the energy was in the *no breakage* range and breakage did not occur.

6.4.1.2 *Incremental breakage*

If the bin energy was greater than the threshold energy, then it was further compared to the *1-hit breakage* energy. None of the literature reviewed by the author indicated actual measured or calculated values for the *1-hit breakage* energy. In this thesis work, the *1-hit breakage* energy was assumed to be equal to $0.01kWh/t$. If the bin energy was less than the *no breakage* energy, then it was assumed that the energy was in the *incremental breakage* range and resulted in breakage after more than one collision.

The impact energy spectra from DEM gives the number of collisions that occurred at a certain bin energy. In order to determine the number of breakages resulting from a certain number of DEM collisions, (λ in the flowchart and "UCM" model equation), it was assumed that likelihood of breakage was proportional to the ratio of the collision energy to the *1-hit breakage* energy given in equation 6.1.

$$\lambda = \frac{bin\ energy}{1 - hit\ breakage\ energy} \tag{6.1}$$

Having calculated for λ, the breakage in and out was determined by applying the "UCM" model equation (6.2).

$$\frac{d}{dt}M_{u*i*} = f_{u*i*} - \sum_{d=1}^{N} \lambda_{u*i*,d}\ m_{u*i*} + \sum_{j=1}^{L}\sum_{d=1}^{N} \lambda_{u*j,d}\ m_{u*j}\ A_{u*j,d,u*i*} - p_{u*i*} \tag{6.2}$$

where:

- f is the feed into the tumbling mill

- λ is the breakage rate
- m is the average mass of the particles in a size fraction
- A is the appearance or breakage function
- p is the product discharged from the mill

6.4.1.3 *1-hit breakage*

If the bin energy was greater than the *1-hit breakage* energy, it was assumed that breakage always occurred after one collision (meaning that λ =number of collisions). Again, the breakage in and out was calculated by applying the "UCM" model equation (6.2).

6.4.2 Mass Balance Check

In the proposed "UCM" tumbling mill model, the amount of mass that breaks in and out of a certain particle class was calculated before adding the feed and removing the product. In order to maintain the material balance in the system, it was therefore necessary that the total mass that is broken out is equal to the total mass broken in.

The following scheme was devised to check the mass balance for the system before the feed was added and the product removed. During the UCM main program run, the masses breaking in and out of each particle size class were calculated and stored in vectors.

Table 6.2 shows the masses that were broken in and out if each particle size class. These were summed up to obtain the total breakage-in and breakage-out. The *breakage-out* included the mass that remained in the original particle size class. For example, the broken material from the $212mm$ particle size class remained in the same particle size class. The *Net Broken Mass* was obtained by subtracting the *breakage-out* from the *breakage-in*.

A negative *Net Broken Mass* therefore meant that the net effect of the comminution action was that particles in the respective particle size class had broken into smaller size particles. A positive *Net Broken Mass* on the other hand meant that net effect of the comminution action was that the respective particle size class had gained from larger size particles that had broken into the size class.

The results showed an exact match between the *breakage-in* and *breakage-out* total mass. A further check was performed by calculating the net broken mass for each particle size class. The sum of the net broken mass was $-3E-15kg$, indicating that there was effectively no net gain/loss of material in the system.

6.4.3 Comminution Action Check

The fourth and fifth columns of table 6.2 were used to do a preliminary check of the results of the UCM model. These results were for the comminution action for the particle size classes within the DEM range. The tumbling mill action is generally expected to result in the breakage of the larger particle into smaller particles. The larger particle size classes are therefore expected to lose material (negative net broken mass) while the smaller particle classes are expected to gain material (positive net broken mass). The results in table 6.2 showed that the larger particle size classes ($25mm$ - $212mm$) had a negative net broken mass while the smaller particle size classes exhibited net gains in material. These results showed that the comminution action in proposed tumbling mill model was as expected.

The percentage gain in material is expected to be inversely proportional to the size of the particle size because all the bigger particles will contribute some breakage into the smaller particle size classes. The highest percentage gain in material were therefore expected to be in the $4.75mm$ and $-4.75mm$ particle size classes. This was however not the case. This

Particle Size (mm)	Breakage Out (kg)	Breakage In (kg)	Original Mass (kg)	Net Broken Mass (kg)	% Broken Mass (kg)	Simulated Charge After 1sec
250	0	0	0	0	0	0
212	0	0	0	0	0	0
150	2.8574	2.8517	18	-0.0057	-0.032	17.9943
106	27.127	27.0287	116	-0.0983	-0.085	115.9017
75	42.3756	42.1093	100	-0.2663	-0.266	99.7337
50	98.2012	97.3419	143	-0.8593	-0.601	142.1407
38	70.5078	69.989	77	-0.5188	-0.674	76.4812
25	40.8383	40.6822	43	-0.1561	-0.363	42.8439
19	12.2493	12.5751	16	0.3258	2.036	16.3258
12	5.939	6.2178	13	0.2788	2.145	13.2788
9.5	0.5812	1.0458	2.814	0.4646	16.513	3.27812
6.7	0.2428	0.5088	2.967	0.266	8.966	3.23288
4.75	0.0315	0.2327	1.360	0.2012	14.798	1.5608
-4.75	0	0.3681	16.86	0.3681	2.183	17.2281
TOTAL	**300.9511**	**300.9511**	**550**	**-3E-15**		**550**

Table 6.2: *Breakage in and out mass balance check*

was probably because the model, as currently applied, does not cater for surface breakage which is expected to contribute to the production of fine particles.

6.4.4 Obtaining The Simulated Charge

Having obtained and checked the results of the comminution action, the next step was to obtain the simulated charge after one time-step (*1 second*). To achieve this, the net broken-mass was added to the original charge mass for each particle size class. The results of the simulated charge after 1sec are shown in the last column of table 6.2.

6.4.5 Sub-DEM Breakage

In this thesis, the method of handling and including breakage due to particles smaller than the minimum DEM particle was suggested in sub-section 3.7.1 of Chapter 3. This method which is a direct use of the Bond Work Index equation (3.9).

The suggested method, is based on the assumption that the DEM will predict a power draw which is lower than the measured (actual) power draw because the power drawn by interactions involving sub-DEM particles is not included.

For the MINTEK pilot-scale SAG mill that was used in this thesis work, the measured actual power drawn by the mill was $11.93\,kW$. The power draw that was predicted by the DEM method in sub-section 4.6.2 of Chapter 4 was $11.43kW$. The power draw of the sub-DEM particles was calculated from equation 3.8 as $0.5kW$.

A MATLAB subroutine was developed to estimate the breakage of the sub-DEM size particles in the feed (Appendix B.3). Figure 6.5 shows a flow chart for calculation of the simulated feed of the sub-DEM size particles.

The steps for implementing the suggested Bond-based method of determining sub-DEM breakage were as follows:

1. Accept the required inputs. The required inputs were: the Bond Work Index and feed rate which were obtained from actual measurements of the example MINTEK pilot-scale SAG mill; the assumed power draw of the sub-DEM particles calculated from the measured and DEM power draws as shown above; and Particle Size Distributions of the feed.
2. The particle size at which 80% of the feed (f_{80} parameter) was then determined from the size distribution of the feed. In doing this, only the sub-DEM particles of the feed was considered.

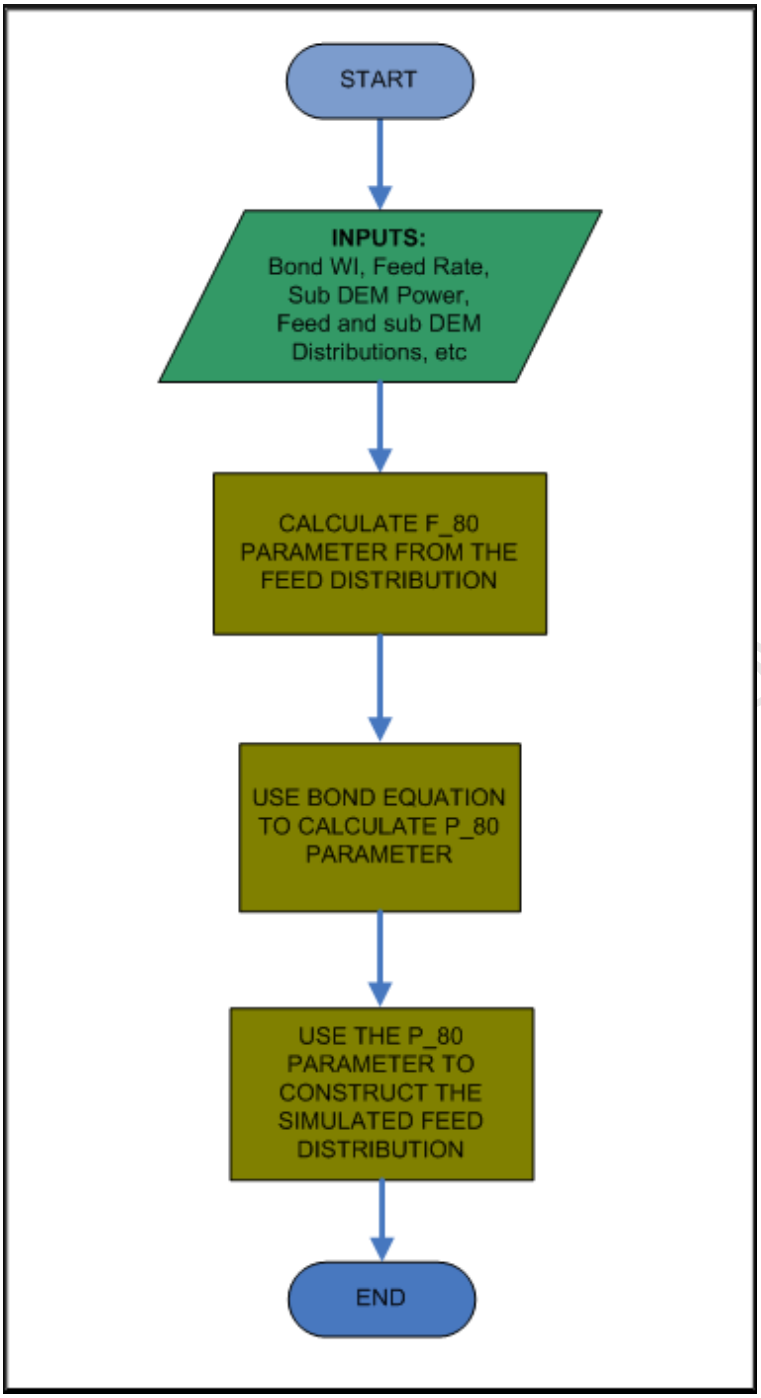

Figure 6.5: *Flow chart for calculating the simulated feed of the sub-DEM size particles*

3. Having obtained the f_{80} parameter, the next was to apply the Bond equation to calculate the particle size at which 80% of the product was passing (the p_{80} parameter).

4. Finally the p_{80} parameter was used to obtain the particle size distribution of the simulated feed.

The simulated charge after one time-step of the UCM was obtained by adding the simulated feed to the original charge for all the sub-DEM particle size class. Table 6.3 shows the results of the UCM simulation of the sub-DEM size particles.

Particle Size (mm)	Feed (kg/s)	Simulated Feed After 1sec (kg)	Original Charge (kg)	Simulated Charge After 1sec (kg)
3.35	0.0070	0.0064	1.1510	1.1574
2.36	0.0048	0.0048	0.7502	0.7550
1.7	0.0041	0.0042	0.4867	0.4909
1.18	0.0042	0.0042	0.4788	0.4830
0.85	0.0039	0.004	0.4649	0.4689
0.6	0.0038	0.0037	0.6878	0.6915
0.425	0.0042	0.0042	1.0147	1.0189
0.3	0.0040	0.004	1.2468	1.2508
0.212	0.0034	0.0036	1.6421	1.6457
0.15	0.0027	0.0026	1.7462	1.7488
0.106	0.0026	0.003	1.4448	1.4478
0.075	0.0008	0.0005	1.1210	1.1215
0.053	0.0048	0.0044	0.9211	0.9255
0.038	0.0053	0.0054	0.8386	0.8440

Table 6.3: *Results of the UCM simulation of the sub-DEM size particles*

6.4.6 Combining the DEM and sub-DEM Parts of the Model

In order to obtain the overall evolution of the mill contents, the results from the DEM-based tumbling mill model were combined with the Bond-based sub-DEM model. This was achieved by combining the masses of the

simulated charge after one time-step for all the particles from both systems (DEM and sub-DEM). Table 6.4 and Figure 6.6 shows a comparison of the overall original and the simulated mill contents particle masses and particle size distributions, respectively.

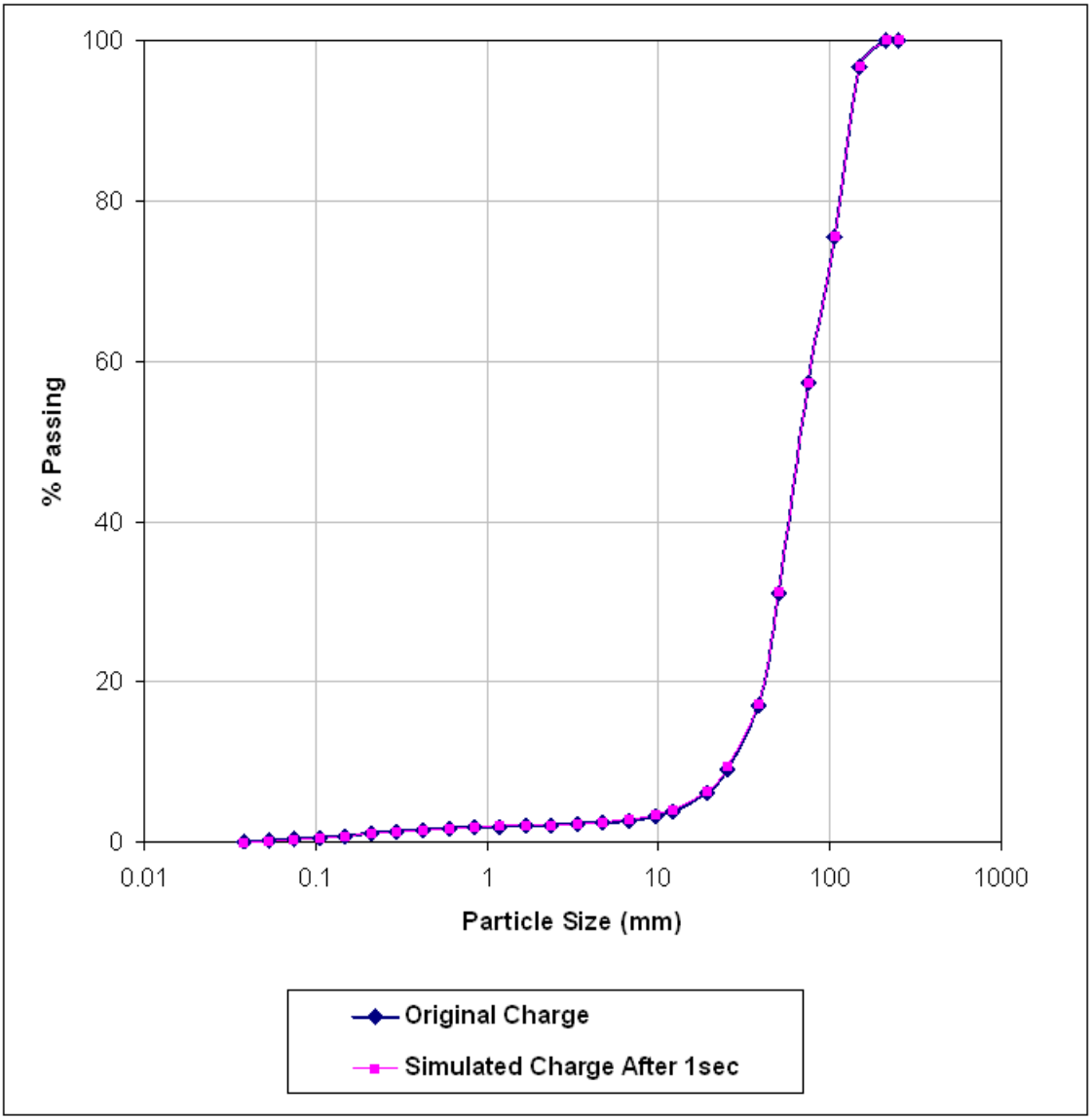

Figure 6.6: *Comparison of the overall original and evolved contents particle size distributions*

6.5 Predicting the Product

For a mill operating at steady-state and containing a perfectly mixed charge, the product rate is related to the mill contents as per equation 3.13.

In this thesis work, the product discharge rate was calculated from the measured product rate, mill contents and feed rate as shown in Section 3.8 of Chapter 3. Equation 6.3 (which is a combination of equations 3.13 and 3.14) was used to calculate the discharge rate for each particle size, i.

$$d_i = \frac{(product\,\%retained)_i}{100} * \frac{feed\,rate}{s_i} \tag{6.3}$$

Where:

- d_i is the discharge rate
- s_i is the mill contents

In this work, the actual measured feed rate (954kg/hr or 0.265kg/s) and product rate for the MINTEK pilot mill was used. The results that were obtained from the discharge rate calculations are shown in table 6.5 The particle size was plotted against discharge rate to obtain the curve shown in figure 6.7 which was similar to the typical curve (shown in figure 3.8).

The predicted product from the proposed tumbling mill model was finally calculated from equation 3.13 as follows:

$$P_{i_{UCM}} = s_{i_{UCM}} * d_i \tag{6.4}$$

Where:

- $P_{i_{UCM}}$ is the predicted product of particle size class, i
- $s_{i_{UCM}}$ is the simulated mill contents after 1sec for particle size class, i
- d_i is the discharge rate, calculated from measured values, for particle size class, i

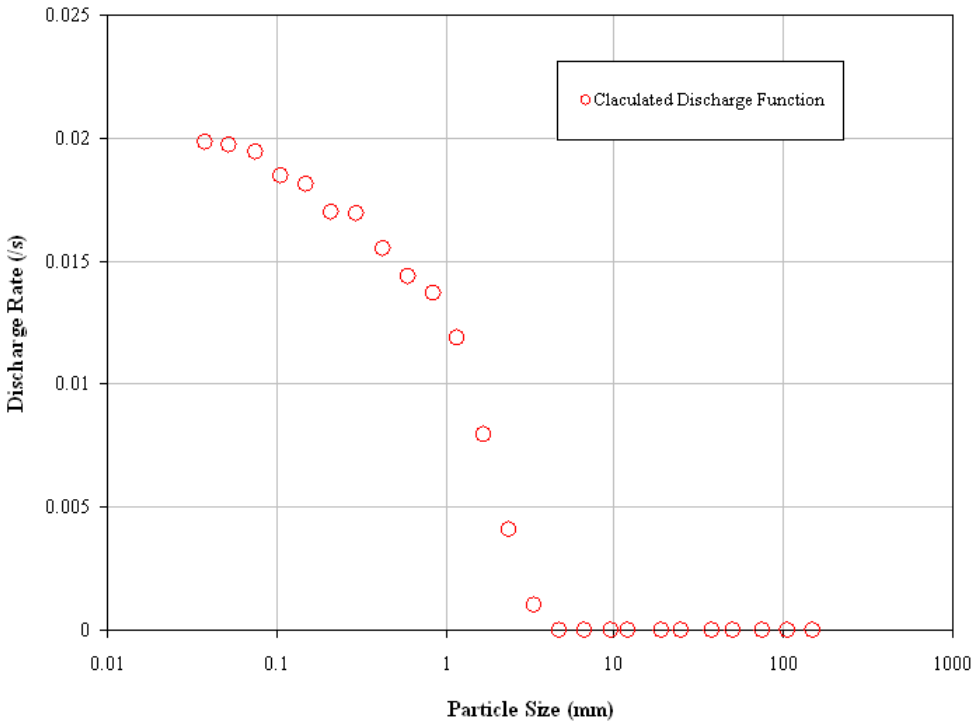

Figure 6.7: *Calculated and fitted discharge function (plot of particle size versus discharge rate*

To obtain the mill contents after 1 time-step, the predicted product was subtracted (discharged) from the simulated charge and the feed added as per equation 3.1. Table 6.6 presents the final results of the proposed tumbling mill model. Figure 6.8 show a comparison of the original charge and the simulated charge after 1 second. A comparison of the expected (calculated) product and the predicted product is shown in figures 6.9 and 6.10.

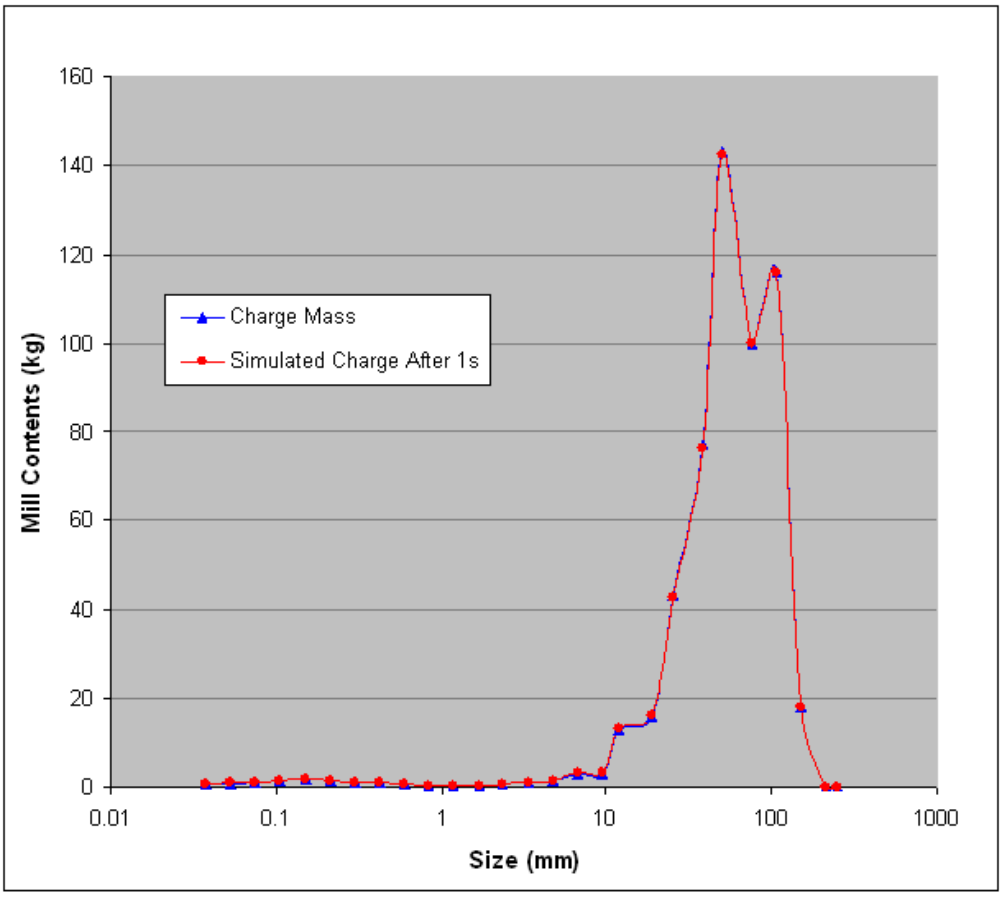

Figure 6.8: *Comparison of the original charge and the simulated charge after 1 sec*

6.6 Discussion Of The Results

The results of the proposed DEM-based tumbling mill model presented in table 6.6 show that the model is able to predict the comminution action. The results represent the comminution action after a time-step of 1 second. Comparison of the mill charge versus particle size, before and after the UCM simulation shows a very fine difference because of the small time-step of *1 second.* The predicted product after 1 second of the UCM simulation showed a good comparison with the calculated (or expected) product.

The following is a discussion of the limitations and constraints of the model:

1. The model in its present form does not predict breakage resulting from surface interactions. This will result in under prediction of breakage

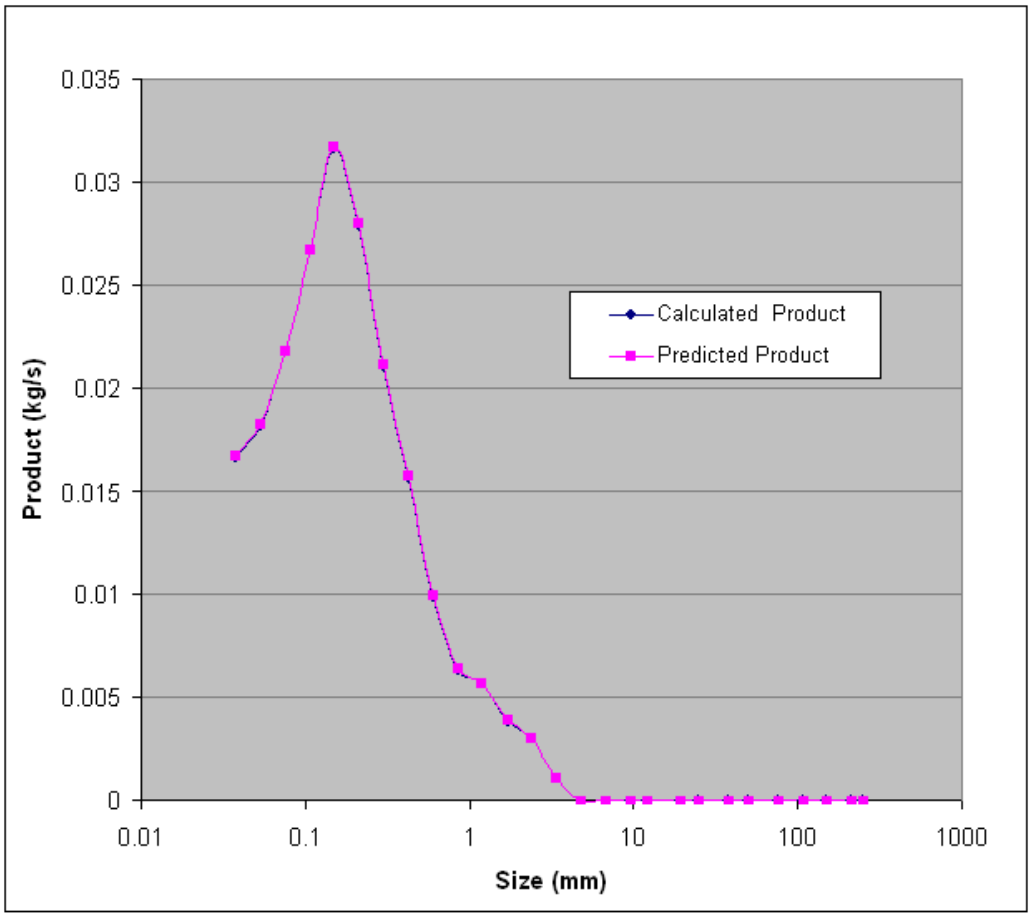

Figure 6.9: *Comparison of the expected (calculated) product and the predicted product*

into the fine particle size range.

2. The assumption that the DEM collision energy is shared equally between the colliding partners could also result into some errors.

3. The DEM has a finite minimum particle size that it can simulate and as such, the DEM-based model had to be extended to include the sub-DEM breakage. The Bond Theory- based method used in this thesis work could suffer from the following short-comings:

- The amount of mass that were breaking into sub-DEM particle sizes from the DEM particle sizes were not taken into consideration.
- The Bond-based method considers the feed coming into the mill while the DEM-based model considers the contents in the mill.

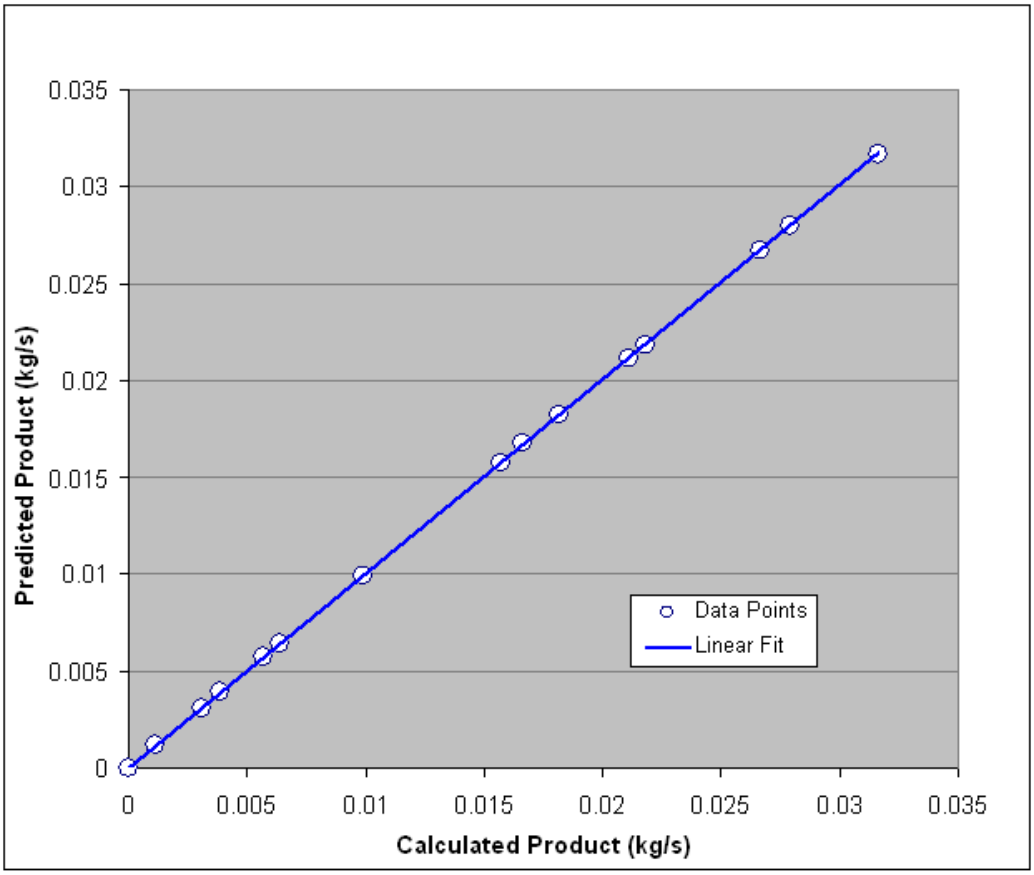

Figure 6.10: *Comparison of the expected (calculated) product and the predicted product*

Particle Size (mm)	Original Charge	Simulated Charge After 1sec	Net Broken Mass (kg)
250	0	0	0
212	0	0	0
150	18	17.994	-0.006
106	116	115.902	-0.098
75	100	99.734	-0.266
50	143	142.141	-0.859
38	77	76.481	-0.519
25	43	42.844	-0.156
19	16	16.326	0.326
12	13	13.279	0.279
9.5	2.814	3.278	0.465
6.7	2.967	3.233	0.266
4.75	1.360	1.561	0.201
3.35	1.151	1.157	0.006
2.36	0.750	0.755	0.005
1.7	0.487	0.491	0.004
1.18	0.479	0.483	0.004
0.85	0.465	0.469	0.004
0.6	0.688	0.692	0.004
0.425	1.015	1.019	0.004
0.3	1.247	1.251	0.004
0.212	1.642	1.646	0.004
0.15	1.746	1.749	0.003
0.106	1.445	1.448	0.003
0.075	1.121	1.122	0.000
0.053	0.921	0.926	0.004
0.038	0.839	0.844	0.005
-0.038	2.865	-	-
TOTAL	550		

Table 6.4: *Comparison of the overall original and evolved contents particle masses*

Size (mm)	Charge (kg)	Measured Product (% Retained)	Measured Product Rate (kg/s)	Discharge Rate (/s)
250	0	0	0	0
212	0	0	0	0
150	18	0	0	0
106	116	0	0	0
75	100	0	0	0
50	143	0	0	0
38	77	0	0	0
25	43	0	0	0
19	16	0	0	0
12	13	0	0	0
9.5	2.8135	0	0	0
6.7	2.9669	0	0	0
4.75	1.3596	0	0	0
3.35	1.1510	0.4367	0.00116	0.00101
2.36	0.7502	1.1533	0.00306	0.00407
1.7	0.4867	1.4633	0.00388	0.00797
1.18	0.4788	2.1500	0.00570	0.01190
0.85	0.4649	2.4067	0.00638	0.01372
0.6	0.6878	3.7367	0.00990	0.01440
0.425	1.0147	5.9333	0.01572	0.01550
0.3	1.2468	7.9600	0.02109	0.01692
0.212	1.6421	10.5433	0.02794	0.01701
0.15	1.7462	11.9400	0.03164	0.01812
0.106	1.4448	10.0667	0.02668	0.01846
0.075	1.1210	8.2200	0.02178	0.01943
0.053	0.9211	6.8500	0.01815	0.01971
0.038	0.8386	6.2833	0.01665	0.01986
-0.038	2.8651	20.8567	0.05527	0.01929
TOTAL	550	100	0.265	

Table 6.5: *Calculation of discharge rate for each particle size*

Particle Size (mm)	Charge Mass (kg)	Simulated charge After 1s (kg)	Calculated Discharge Rate (/s)	Predicted Product Rate (kg/s)	Calculated Product Rate (kg/s)	Contents After Discharge (kg)	Feed (kg/s)	Mill Contents After 1s (kg)
250	0	0	0	0	0	0	0	0
212	0	0	0	0	0	0	0.0109	0.0109
150	18	17.9943	0	0	0	17.9943	0.0119	18.0062
106	116	115.9017	0	0	0	115.9017	0.0171	115.9188
75	100	99.7337	0	0	0	99.7337	0.0174	99.7511
50	143	142.1407	0	0	0	142.1407	0.0193	142.1600
38	77	76.4812	0	0	0	76.4812	0.0180	76.4992
25	43	42.8439	0	0	0	42.8439	0.0257	42.8696
19	16	16.3258	0	0	0	16.3258	0.0201	16.3459
12	13	13.2788	0	0	0	13.2788	0.0260	13.3048
9.5	2.814	3.2781	0	0	0	3.27812	0.0117	3.2898
6.7	2.967	3.2329	0	0	0	3.23288	0.0127	3.2455
4.75	1.360	1.5608	0	0	0	1.5608	0.0104	1.5712
3.35	1.151	1.1574	0.00101	0.0012	0.0012	1.1563	0.0082	1.1563
2.36	0.750	0.7550	0.00407	0.0031	0.0031	0.7520	0.0070	0.7520
1.7	0.487	0.4909	0.00797	0.0039	0.0039	0.4870	0.0048	0.4870
1.18	0.479	0.4830	0.01190	0.0057	0.0057	0.4773	0.0041	0.4773
0.85	0.465	0.4689	0.01372	0.0064	0.0064	0.4624	0.0042	0.4624
0.6	0.688	0.6915	0.01440	0.0100	0.0099	0.6816	0.0039	0.6816
0.425	1.015	1.0189	0.01550	0.0158	0.0157	1.0031	0.0038	1.0031
0.3	1.247	1.2508	0.01692	0.0212	0.0211	1.2296	0.0042	1.2296
0.212	1.642	1.6457	0.01701	0.0280	0.0279	1.6177	0.0040	1.6177
0.15	1.746	1.7488	0.01812	0.0317	0.0316	1.7172	0.0034	1.7172
0.106	1.445	1.4478	0.01846	0.0267	0.0267	1.4211	0.0027	1.4211
0.075	1.121	1.1215	0.01943	0.0218	0.0218	1.0997	0.0026	1.0997
0.053	0.921	0.9255	0.01971	0.0182	0.0182	0.9073	0.0008	0.9073
0.038	0.839	0.8440	0.01986	0.0168	0.0167	0.8272	0.0048	0.8272
-0.038	2.865	-	-	-	-	-	-	-
Totals	550.000	546.8218	0.19807	0.2104	0.2097	546.6113	0.2597	546.8125

Table 6.6: *Final results of the proposed tumbling mill model (UCM)*

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, conclusions from the thesis work are drawn. The author also puts forth some recomendations for further work that could be carried out on the thesis topic.

7.1 Conclusions

In this thesis a tumbling mill model that utilises inputs from DEM simulations and breakage experiments was developed. A demonstration of how the developed model can be used was performed using the MINTEK mill pilot scale work conducted earlier. In order to achieve this, methodologies were developed for extracting and analysing the data from DEM simulations, in such a way that it is useful to the prediction of the collision energy distribution in the tumbling mill. A methodology for using and analysing data from breakage experiments to inform the DEM based tumbling mill model was also developed.

The following hypotheses were tested in this thesis work:

1. Outputs from a DEM simulation of the tumbling mill can be used to inform tumbling mill models.
2. The distribution of the collision energies in the tumbling mill obtained from DEM simulations can be combined with data from breakage experiments in a population balance framework, that can be used to predict the breakage of the coarser particles in the tumbling mill.
3. DEM's prediction of the collision energy distribution and Bond work index laboratory tests can be used to predict the breakage of fine particles in the tumbling mill.

7.2 Recommendations

From the thesis work, the following recommendations were drawn for further work on the thesis topic:

7.2.1 Extension Of The Model To Include Surface Breakage

The developed tumbling mill model assumed that all breakage in the mill resulted from bulk (or impact) particle interaction events. In order to extend the model to include breakage from surface interaction events, further work should be done on developing breakage experiments that can measure surface breakage over a wide range of input energies.

The present surface breakage experiments fall short of the requirements of the developed tumbling mill model.

7.2.2 "Model Specific" Breakage Experiments

In this thesis work a methodology for utilising the results from the JKMRC Drop Weight Test breakage experiments was developed and used in the

proposed model. There is however a need to carry out further work to develop a breakage experiment that will be specific to the requirements of the developed model.

As pointed out in the thesis, the Split Hopkinson Pressure Bar (SHPB) breakage experiment is more suited to the model's requirement as it measures the energy that dissipated during an impact. The SHPB experiment in it's present form however can not cover the whole range of inputs energies required by the model.

7.2.3 Sharing of Dissipated Energy Among Colliding Partners

The developed tumbling mill model is based on the premise that breakage for a particular particle size class, is a function of the DEM collision energy distribution. This therefore requires the energy absorbed by each colliding partner be quantified. The DEM however, measures the energy that is dissipated during a collision and not what is absorbed by each colliding partner.

In this thesis work, it was assumed that the collision energy is shared equally between the colliding partners. Further work needs to be carried out to develop a mechanism for apportioning the DEM collision energy to the respective colliding partners.

7.2.4 Further Iterations To Obtain The Actual Mill Product (Discharge)

This thesis work served to develop and demonstrate the use of, the presented DEM-based tumbling mill model. The energy and collision distribution changes as the size distribution varies, so in order to obtain the final mill product that is discharged, one would need to run further iterations of the model by inputting the evolved contents into subsequent fresh DEM simulations.

7.2.5 Sub-DEM Breakage

The DEM has a finite minimum particle size that it can simulate. The model is therefore not able to predict breakage for particles smaller than the DEM minimum particle size. In this thesis, a method to incorporate the sub-DEM breakage using the "Bond Theory" was suggested. This method had some short-comings (see sub-section 6.6).

Further work needs to be carried out to develop a methodology for incorporating the sub-DEM breakage into the model.

Bibliography

Banini, G. A., 2000. An integrated description of rock breakage in comminution machines. Ph.D. thesis, Department of Mining, Metallurgical and Materials Engineering, University of Queensland.

Bbosa, L., 2005. Utilising a shpb to test ore breakage. Final Year Thesis.

Bbosa, L., 2006. An investigation of impact breakage of rocks using the split hopkinson pressure bar. SAIMM 106 (4), 291–296.

Bbosa, L., 2007. Measurement of impact breakage properties of ore particles using a series of devices, MSc thesis submitted to the University of Cape Town, 2007.

Bond, F. C., 1952. The third theory of comminution. Transactions of the AIME 193, 484–494.

Bond, F. C., 1961a. Crushing and grinding calculations: Part one. British Chemical Engineering 6, 378–385.

Bond, F. C., 1961b. Crushing and grinding calculations: part two. British Chemical Engineering 6, 543–634.

Briggs, C. A., Bearman, R. A., 1995. The assessment of rock breakage and damage in crushing machinery. In: Proceedings Expo'95. AusIMM, Brisbane, Australia, pp. 167–172.

Chandramohan, R., 2005. Measurement of particle interaction properties for the incorporation into discrete element methods. Master's thesis, University of Cape Town.

Cleary, P. W., 1998. Predicting charge motion, power draw, segregation and wear in ball mills using discrete elemnt methods. Minerals Engineering 11 (11), 1061–1080.

Cleary, P. W., 2001. Recent advances in dem modelling of tumbling mills. Minerals Engineering 14 (10), 1295–1319.

Cleary, P. W., 2004. Large scale industrial dem modelling. Engineering Computations 21 (2/3/4), 169–204.

Condori, P., July 2006. Private Communication, University of Cape Town, South Africa.

Corkum, B. T., 1989. The discrete element method in geotechnical engineering. Master's thesis, University of Toronto, Canada.

Cundall, P. A., Strack, O. D. L., 1979. A discrete numerical model for granular assemblies. Goetechnique 29 (1), 47–65.

Datta, A., Mishra, B. K., Rajamani, R. K., 1999. Analysis of power draw in ball mills by the discrete element method. Canadian Metallurgical Quarterly 38, 133–140.

Datta, A., Rajamani, R. K., 2002. A direct approach of modelling batch grinding in ball mills using the population balance principles and impact energy distribution. International Journal of Mineral Processing 64, 181–200.

DEM Solutions, 2006. Discrete element modelling. EDEM software, v1.1, online Manual, www.DEM-SOLUTIONS.com.

Epstein, B., December 1947. The material description of certain breakage mechanisms leading to the logathmic-normal distribution. Journal of the Franklin Institute 244 (6), 471–477.

Flavel, M. D., Rimmer, H. W., 1981. Particle breakage study in an impact breakage environment. In: SME-AIME Annual Meeting. SME Publications, Littleton, Ohio, USA.

Ghabousi, J., Barbosa, R., 1990. Three dimensional discrete element method for granular materials. International Journal for Numerical and Analytical Methods in Geomechanics 14 - 7, 451–472.

Govender, I., 2005. X-ray motion analysis of charge particles in a laboratory mill. Phd thesis, University of Cape Town.

Govender, I., McBride, A. T., Powell, M. S., 2004. Improved experimental tracking techniques for validating discrete element method simulations of tumbling mills. Experimental Mechanics 44 (6), 593–607.

Han, T., Kalman, H., Levy, A., October 2002. Dem simulation of particle comminution in jet milling. Particulate Science and Technology 20 (4), 325 – 340.

Harris, C. C., Arbiter, N., 1982. Grinding mill scale-up problems. Mining Engineering 34 (1), 43–46.

Herbst, J. A., Nordell, L., 2001. Optimisation of the design of sag mill internals using high fidelity simulation. pp. 150–164.

Itasca Consulting Group, 1999. Particle Flow Code in 3 Dimensions: Theory and Background. Itasca Consulting Group, Inc, Minneapolis, Minnesota, USA.

Johnson, K. L., 1985. Contact Mechanics. Cambridge University Press, Cambridge CB2 1RP.

Kick, F., 1883. The law of proportional resistance and its application to sand and explosions. Dinglers J 247, 1–5.

Loveday, B., Naidoo, D., 1997. Rock abrasion in autogeneous milling. Minerals Engineering 10 (3), 603–612.

McBride, A. T., July 2006. Private communication, University of Cape Town, South Africa.

McBride, A. T., Govender, I., Powell, M. S., Cloete, T. J., 2004a. Contributions to the experimental validation of the discrete element method apploed to tumling mills. Engineeing Computations 21 (2/3/4), 119–136.

McBride, A. T., Makepe, T., Scheele, F., 2004b. Discrete element method analysis of a bearing capacity experiment. In: Zingoni (Ed.), Progress in Structural Engineering, Mechanics and Computation. Taylor & Francis Group, London, pp. 1003–1017.

McBride, A. T., Powell, M. S., 2006. A structured approach to modelliing sag mill liner wear - numerical modelling of liner evolution. SAG 2006, University of British Columbia.

Mindlin, R. D., Deresiewicz, H., 1953. Elastic spheres in contact under varying oblique forces. Journal of Applied Mechanics 20, 327–344.

Mishra, B. K., 2003. A review of computer simulation of tumbling mills by the discrete element method: part i - contact mechanics. International Journal of Mineral Processing 71, 73–93.

Mishra, B. K., Murty, C. V. R., 2001. On the determination of contact parameters for realistic dem simulations of ball mills. Powder Technology 115, 290–297.

Mishra, B. K., Rajamani, R. K., November 1992. The discrete element method for the simulation of ball mills. Applied Mathematical Modelling 16 (11), 598–604.

Mishra, B. K., Thornton, C., 2002. An improved contact model for ball mill simulation by the discrete element method. Advanced Powder Technology 13 (1), 25–41.

Misra, A., Cheung, J., 1999. Particle motion and energy distribution in tumbling ball mills. Powder Technology 105, 222–227.

Morrell, S., 1992. Prediction of grinding mill power. Transactions of the Institute of Mining and Metallurgy 101, C25–C32.

Morrison, R. D., Cleary, P. W., 2004. Using dem to model ore breakage within a pilot scale sag mill. Minerals Engineering 17, 1117–1124.

Moys, M. H., 1993. A model of mill power as affected by mill speed, load volume and liner design. Journal of SAIMM 93, 135–141.

Munjiza, A., Bicniic, N., Owen, D. R. J., 1993. Bsd contact detection algorithm for discrete elements in 2d. In: Proceedings of the 2nd International Conference on the dicrete element method. IESL publication edition, MA, USA, pp. 39–52.

Napier-Munn, T. J., Morrell, S., Morrison, R. D., Kojovic, T., 1999. Mineral Comminution Circuits. Their operation and optimisation. Julius Kruttschnitt Mineral Research Centre (JKMRC).

Narayanan, S. S., Whiten, W. J., 1988. Determination of comminution characteristics from single particle breakage tests and its application to ball mill scale-up. Transactions of the Institutions of Mining & Metallurgy. 97, C115–C124.

Powell, M. S., 1993. A study of charge motion in rotary mills, with particular reference to the grinding action. Ph.D. thesis, University of Cape Town.

Powell, M. S., June 2005. Unified comminution model, the concept. P9N Research Report Third.

Powell, M. S., September 2006. The unified comminution model - a conceptually new model. IMPC, Instabul, Turkey, pp. 1783–1788.

Powell, M. S., January 2008. Private communication, JKMRC, Australia.

Powell, M. S., McBride, A. T., 2004. A three-dimensional analysis of media motion and grinding regions in mills. Minerals Engineering 17 (11-17), 1099–1109.

Powell, M. S., McBride, A. T., 2006. What is required from dem simulations to model breakage in mills? Minerals Engineering 19, 1013–1021.

Powell, M. S., McBride, A. T., Govender, I., 2003. Application of dem outputs to refining applied sag mill models. In: Lorenzen, L., Bradshaw, D. (Eds.), Proceedings: XXII International Mineral Processing Congress. Document Transformation Technologies, Cape Town, South Africa, pp. 307–316.

Powell, M. S., Nurick, G. N., 1996. A study of charge motion in rotary mills, part 1 - extension of the theory. Minerals Engineering 9(2), 259–268.

Rajamani, R. K., Mishra, B. K., Venugopal, R., Datta, A., 2000. Discrete element analysis of tumbling mills. Powder Technology 109, 105–112.

Rittinger, R. P., 1867. Textbook of mineral dressing. Ernst and Korn, Berlin.

Sarracino, R. S., McBride, A. T., Powell, M. S., October 2004. Using particle flow code to investigate energy dissipation in a rotary grinding mill. In: Shimizu, Y., Hart, R., Cundall, P. (Eds.), Numerical Modeling in Micromechanics Via Particle Methods: Proceedings of the 2nd International Pfc Symposium, Kyoto, Japan. Balkema, Lisse, Netherlands, pp. 111–118.

Shi, F., Kojovic, T., 2006. Investigation of a model for impact breakage incorporating particle size effect. International Journal of Mineral Processing.

Ting, J. M., 1992. A robust algorithm for ellipse based discrete element modelling of granular materials. Computers and Geotechniques 13, 175–186.

Tsuji, Y., Kawaguchi, T., Tanaka, T., 1993. Discrete particle simulation of two-dimensional fluidized bed. Powder Technology 77 (1), 79–87.

Van Eck, M., 2007. Investigating the products from models of particle breakage testing. Master's thesis, University of Cape Town.

van Nierop, M. A., Glover, G., Hinde, A. L., Moys, M. H., 2001. A discrete element investigation of the charge motion and power draw of an experi-

mental two-dimensional mill. International Journal of Mineral Processing 61, 77–92.

Venugopal, R., Rajamani, R. K., 2001. 3d simulation of charge motion in tumbling mills by the discrete element method. Powder Technology 115, 157–166.

Vogel, L., Peukert, W., 2004. Determination of material properties relevant to grinding by practicable labscale milling tests. International Journal of Mineral Processing 74S, S329–S338.

Wang, L., Park, J., Fu, Y., 2005. Representation of real particles for dem simulation using x-ray tomography. Construction and Building Materials 21, 338–346.

Whiten, W. J., 1976. Ball mill simulation using small calculators. In: Proceedings AUSIMM.

Whyte, R., 2005. Measuring incremental damage in rock breakage by impact. BE (Honours) Thesis, University of Queensland, Australia.

Williams, J. R., O'Connor, R., Rege, N., 1996. Discrete element analysis and granular vortex formation. Electronic Journal of Geotechnical Engineering 1.

Appendix A

Measured MINTEK Pilot SAG Mill Data

In this thesis, the pilot-scale SAG mill at Mintek was used as the main example DEM simulation. The operating conditions and particle size distribution data used in the simulation used actually measured at site. The actual measured data is given in the following appendices.

A.1 Mill Contents (Ore) Particle Size Distributions (PSD)

	ACTUAL DISTRIBUTION				DEM SIMULATION DISTRIBUTION		
Screen Size (mm)	Mass on screen (kg)	Cum. % Passing	Mass per particle (kg)	Number of particles	Adjusted DEM No. of particles	Adjusted DEM mass (kg)	DEM Cum. % Passing
300	-	100	-	-	-	-	100.00
212	-	100.00	22.50708	-	-	-	100.00
150	18.00	96.73	7.95745	2.3	3	23.87	95.60
106	116.00	75.64	2.81338	41.2	42	118.16	73.84
75	100.00	57.45	0.99468	100.5	101	100.46	55.34
50	143.00	31.45	0.32224	443.8	444	143.07	28.98
38	77.00	17.45	0.11622	662.6	663	77.05	14.79
25	43.00	9.64	0.04109	1046.5	1047	43.02	6.87
19	16.00	6.73	0.01453	1101.4	1102	16.01	3.92
12	13.00	4.36	0.00483	2691.0	2691	13.00	1.53
9.5	2.81	3.85	0.00171	1647.2	1648	2.81	1.01
6.7	2.97	3.31	0.00071	4163.6	4164	2.97	0.46
4.75	1.36	3.07	0.00025	5396.7	5397	1.36	0.21
3.35	1.15	2.86	8.91E-05	12922.6	12923	1.15	0.00
2.36	0.75	2.72	3.12E-05	24050.9			
1.7	0.49	2.63	1.13E-05	43162.3			
1.18	0.48	2.54	3.99E-06	120094.8			
0.85	0.46	2.46	1.41E-06	329804.1			
0.6	0.69	2.33	5.11E-07	1345856.5			
0.425	1.01	2.15	1.81E-07	5615682.3			
0.3	1.25	1.92	6.39E-08	19516327.6			
0.212	1.64	1.62	2.25E-08	72958386.7			
0.15	1.75	1.31	7.96E-09	219447094.8			
0.106	1.44	1.04	2.81E-09	513545147.8			
0.075	1.12	0.84	9.95E-10	1127033952.7			
0.053	0.92	0.67	3.52E-10	2619250866.7			
0.038	0.84	0.52	1.27E-10	6611675444.8			
pan	2.87	0.00					
Total Mass	550					542.94	

A.2 Feed and Discharge Particle Size Distributions (PSD)

	FEED		PRODUCT	
Screen Size (mm)	Mass on screen (kg)	Cum. % Passing	prod, kg/s	product, cum % passing
	0	100	0	
250	73.0	98.5	0	
180	233.0	93.9	0	
126	277.5	88.3	0	
90	330.0	81.7	0	
63	344.5	74.8	0	
45	360.5	67.6	0	
31.5	369.0	60.3	0	
22.4	433.5	51.6	0	
16.0	439.0	51.6	0	
11.2	346.0	35.9	0	
8.0	288.5	30.1	0	
5.6	206.0	26.0	0	100
4.0	181.5	22.4	0.00116	99.6
2.8	136.5	19.6	0.00306	98.4
2.0	117.0	17.3	0.00388	96.9
1.4	79.5	15.7	0.00570	94.8
1.0	76.5	14.2	0.00638	92.4
.710	80.5	12.6	0.00990	88.7
.500	69.5	11.2	0.01572	82.7
.355	75.0	9.7	0.02109	74.8
.250	79.0	8.1	0.02794	64.2
.180	67.0	6.8	0.03164	52.3
.125	60.0	5.6	0.02668	42.2
.090	46.5	4.6	0.02178	34.0
.063	37.0	3.9	0.01815	27.1
.045	30.5	3.3	0.01665	20.9
.032	163.5	.0	0.05527	0.0

A.3 Mill Contents (Balls) Particle Size Distribution (PSD)

Screen Size (mm)	Mass on screen (kg)	Cum. % retained	Mass per particle (kg)	Number of particles
100	347.8	57.79	4.0841	85
80	165.0	27.41	2.0910	79
60	61.3	10.18	0.8822	69
40	27.8	4.62	0.2614	106
Total Mass	601.85			

Appendix B

MATLAB Scripts and Functions

The MATLAB scripts and functions used in this thesis, are presented in this appendix. These are arranged in the order in which they appear in the thesis.

B.1 Energy_Spectra and Power_Draw

```
% Filename: rockParticleIDvectors.m
% Date: 03 September 2006
% rockParticleIDvectors function extracts the particle IDs from data files
% for each rock particle class.
%
% Output: 'mat' file containing vectors of particle IDs for each rock
% particle class

clear all; clc;
rock180_IDs = extractParticleIDs('rock180_IDs.csv');
rock125_IDs = extractParticleIDs('rock125_IDs.csv');
rock90_IDs = extractParticleIDs('rock90_IDs.csv');
rock63_IDs = extractParticleIDs('rock63_IDs.csv');
rock45_IDs = extractParticleIDs('rock45_IDs.csv');
rock31_IDs = extractParticleIDs('rock31_IDs.csv');
rock22_IDs = extractParticleIDs('rock22_IDs.csv');
rock16_IDs = extractParticleIDs('rock16_IDs.csv');
rock11_IDs = extractParticleIDs('rock11_IDs.csv');
rock08_IDs = extractParticleIDs('rock08_IDs.csv');
```

```
rock056_IDs = extractParticleIDs('rock056_IDs.csv');

save rockParticleIDvectors



% Filename: extractCollisionData.m
% Date: 03 September 2006
% extractParticleIDs function extracts the particle IDs from a comma-
% separated values (csv) file written by the EDEM analyst.
%
% Inputs: particleDataFile. This is the file name of the EDEM csv file
%
% Output: particleIDS. Retains the particle IDs for the particle class

function [particleIDs] = extractParticleIDs(particleDataFile);

fid = fopen(particleDataFile,'r'); %open EDEM data file

ignoreLines = true;
particleIDs = [];
readDataFlag = 0;

while 1
  line=fgetl(fid); %read every line in the data file
  if ~isstr(line), break, end %stop reading if a non-string encountered
  if ( (size(line,2)>=4) & (line(1:4) == 'TIME'))
  ignoreLines = false; %ignore all header lines before 'TIME'
  end
  if ((ignoreLines == false) & (size(line,2)>=1) )
  lineSize = size(line,2);
  if ( (lineSize >=5) & line(1:5) == 'TIME:')
  theTime = str2double(line(7:end));
            continue;
elseif ( (lineSize >=17)& line(1:17) == 'Q1 : Particle ID:')
line = line(18:end);
            particleIDs = [particleIDs; str2num(line)'];
            readDataFlag = 1; %the lines after this will purely contain data. set flag to true
            line = line(1:end); %the next lines will be from 1:end
            continue;
end
    end
    if readDataFlag
        particleIDs = [particleIDs; str2num(line)'];
    end
```

```
end

fclose all;



% Filename: extractCollisionEnergy.m
% Date: 03 September 2006
% extractCollisionEnergy extracts the collision data (energy) from a comma-
% separated values (csv) file written by the EDEM analyst. It also plots
% the histogram and line impact energy spectra for each rock particle class
%
% Inputs: particleDataFile. This is the file name of the EDEM csv file
%        energyBinRange. This the vector containing the energy bins
% Output: lambda_DEM_d. 'mat' file containing a matrix of binned collision
%                       frequencies for each rock particle class
%         energyBinRange. vector containg energy bins
%         powerDraw. Predicted power draw of the mill


clear all; close all; clc;
load rockParticleIDvectors;
energyBinRange = 0.005:0.01:9;
pack;
particleDataFile = 'collision_data.csv';
fid = fopen(particleDataFile,'r');

counter=0;
ignoreLines = true;
normEnergy180 = [];
normEnergy125 = [];
normEnergy90 = [];
normEnergy63 = [];
normEnergy45 = [];
normEnergy31 = [];
normEnergy22 = [];
normEnergy16 = [];
normEnergy11 = [];
normEnergy08 = [];
normEnergy056 = [];
normEnergy = [];
tanEnergy = [];
totEnergy = []
colIDs1 = [];
colIDs2 = [];
```

```
ii = 0; check = 0;
while 1
  counter=counter+1;
  line=fgetl(fid);
  if (rem(counter,10000) == 0)
  disp(['Line number ', num2str(counter)]);
  end
  if ~isstr(line), break, end
  if ( (size(line,2)>=4) & (line(1:4) == 'TIME'))
  ignoreLines = false;
    end

    % Extract collision energies and collision IDs for contacting
    % partners.
  if ((ignoreLines == false) & (size(line,2)>=1) )
  lineSize = size(line,2);
  if ( (lineSize >=5) & line(1:5) == 'TIME:')
  theTime = str2double(line(7:end));
display([particleDataFile, 'The time: ', num2str(theTime), ' Total Time 17.5']);
elseif ( (lineSize >=34)& line(1:34) == 'Q3 : Collision Normal Energy Loss:')
line = line(35:end);
            normEnergy = [normEnergy; str2num(line)'];
        elseif ( (lineSize >=28)& line(1:28) == 'Q1 : Collision ID Element 1:')
line = line(29:end);
            colIDs1 = [colIDs1; str2num(line)'];
        elseif ( (lineSize >=28)& line(1:28) == 'Q2 : Collision ID Element 2:')
line = line(29:end);
            colIDs2 = [colIDs2; str2num(line)'];
        elseif ( (lineSize >=38)& line(1:38) == 'Q4 : Collision Tangential Energy Loss:')
line = line(39:end);
            tanEnergy = [tanEnergy; str2num(line)'];
        elseif ( (lineSize >=33)& line(1:33) == 'Q5 : Collision Total Energy Loss:')
line = line(34:end);
            totEnergy = [totEnergy; str2num(line)'];
        end
end
end

fclose all;

% Sort the normal collision energies into respective rock particle classes
for i=1:size(normEnergy,1)
    display(['Counting Down: ',num2str(size(normEnergy,1)-i)]);
```

```
check=any(find(colIDs1(i,1)==rock056_IDs));
if check; normEnergy056 = [normEnergy056; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock056_IDs));
if check; normEnergy056 = [normEnergy056; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock08_IDs));
if check; normEnergy08 = [normEnergy08; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock08_IDs));
if check; normEnergy08 = [normEnergy08; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock11_IDs));
if check; normEnergy11 = [normEnergy11; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock11_IDs));
if check; normEnergy11 = [normEnergy11; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock16_IDs));
if check; normEnergy16 = [normEnergy16; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock16_IDs));
if check; normEnergy16 = [normEnergy16; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock22_IDs));
if check; normEnergy22 = [normEnergy22; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock22_IDs));
if check; normEnergy22 = [normEnergy22; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock31_IDs));
if check; normEnergy31 = [normEnergy31; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock31_IDs));
if check; normEnergy31 = [normEnergy31; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock45_IDs));
if check; normEnergy45 = [normEnergy45; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock45_IDs));
if check; normEnergy45 = [normEnergy45; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock63_IDs));
if check; normEnergy63 = [normEnergy63; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock63_IDs));
if check; normEnergy63 = [normEnergy63; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock90_IDs));
if check; normEnergy90 = [normEnergy90; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock90_IDs));
if check; normEnergy90 = [normEnergy90; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock125_IDs));
if check; normEnergy125 = [normEnergy125; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock125_IDs));
if check; normEnergy125 = [normEnergy125; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs1(i,1)==rock180_IDs));
if check; normEnergy180 = [normEnergy180; normEnergy(i,1)/2]; continue; end;
check=any(find(colIDs2(i,1)==rock180_IDs));
if check; normEnergy180 = [normEnergy180; normEnergy(i,1)/2]; continue; end;
```

```
end

% Obtain collision frequencies
lambda_DEM_d(1,:) = hist(normEnergy180,energyBinRange);
lambda_DEM_d(2,:) = hist(normEnergy125,energyBinRange);
lambda_DEM_d(3,:) = hist(normEnergy90,energyBinRange);
lambda_DEM_d(4,:) = hist(normEnergy63,energyBinRange);
lambda_DEM_d(5,:) = hist(normEnergy45,energyBinRange);
lambda_DEM_d(6,:) = hist(normEnergy31,energyBinRange);
lambda_DEM_d(7,:) = hist(normEnergy22,energyBinRange);
lambda_DEM_d(8,:) = hist(normEnergy16,energyBinRange);
lambda_DEM_d(9,:) = hist(normEnergy11,energyBinRange);
lambda_DEM_d(10,:) = hist(normEnergy08,energyBinRange);
lambda_DEM_d(11,:) = hist(normEnergy056,energyBinRange);

% Plot normal collision energy spectra for each particle class
plotEnergySpectra(normEnergy180,lambda_DEM_d(1,:),energyBinRange)
plotEnergySpectra(normEnergy125,lambda_DEM_d(2,:),energyBinRange)
plotEnergySpectra(normEnergy90,lambda_DEM_d(3,:),energyBinRange)
plotEnergySpectra(normEnergy63,lambda_DEM_d(4,:),energyBinRange)
plotEnergySpectra(normEnergy45,lambda_DEM_d(5,:),energyBinRange)
plotEnergySpectra(normEnergy31,lambda_DEM_d(6,:),energyBinRange)
plotEnergySpectra(normEnergy22,lambda_DEM_d(7,:),energyBinRange)
plotEnergySpectra(normEnergy16,lambda_DEM_d(8,:),energyBinRange)
plotEnergySpectra(normEnergy11,lambda_DEM_d(9,:),energyBinRange)
plotEnergySpectra(normEnergy08,lambda_DEM_d(10,:),energyBinRange)
plotEnergySpectra(normEnergy056,lambda_DEM_d(11,:),energyBinRange)

% Claculate power draw
sumTotEnergy = sum(totEnergy);
powerDraw = sumTotEnergy / 2.462;

save DEMSpectra lambda_DEM_d energyBinRange powerDraw
save energyVectors normEnergy tanEnergy totEnergy normEnergy180 normEnergy125 ...
    normEnergy90 normEnergy63 normEnergy45 normEnergy31 normEnergy22 normEnergy16 ...
    normEnergy11 normEnergy08 normEnergy056



% Filename: plotEnergySpectra.m
% Date: 03 September 2006
% plotEnergySpectra plots the collision energy distribution and spectra
%
% Inputs: energyVector. This is the vector containing collision energies
%         lambda. Contains the binned collision frequencies
```

```
%         energyBinRange. This the vector containing the energy bins

function plotEnergySpectra(energyVector,lambda,energyBinRange)

% Plot energy distributions
figure('name','Energy distribution') %,'NumberTitle','off'
subplot(1,2,1); semilogx(sort(energyVector,1)',1:size(energyVector,1),'r.');
axis tight
title('collisions vs energy')
xlabel('Energy (J)')
ylabel('Sorted collision number')
grid on

% Plot energy spectra
subplot(1,2,2); loglog(energyBinRange,lambda,'b','LineWidth',2.5);
axis tight
title('Energy Spectra')
% xlim([0.05 maxE])
xlabel('Energy (J)')
ylabel('Collision Frequency')
grid on
grid minor
```

B.2 Appearance_Function

```
% File name: Appearance.m
% Date: 04 October 2006
% Purpose: This script calculates the appearance functions for a given ore.
% It requires the raw data from DWT experiments and outputs the appearance
% functions (variable called, AppearanceFunc) in a mat file named
% AppearanceFunctions.mat

clc; clear all;
fig=0;
EnergyBins = [0,0.005:0.01:9];
meanSize = [0.18 0.125 0.09 0.063 0.045 0.0315 0.0224 0.0165 0.0112 0.008 0.0056];
density = 2500;

% obtain the coeffs from fit of Shi-Kojovic model to raw DWT data
[coeffs r_squared] = fitt10();
A = coeffs(1); % degre of breakage
b = coeffs(2); % material parameter f_mat

% get the t10 values corresponding to each energy bin
%DEM_DWT_scaling = [41 37 35 32 28 26 22 20 15 15 13]; %[137 46 17 36 72 122 75 23 2.5];
% These scaling values were obtained from DEM simulations of the DWT experiments
x = [58.694 58.694 58.694 41.079 41.079 41.079 28.892 28.892 28.892 20.630 20.630 ...
    20.630 14.533 14.533 14.533];
for i = 1:size(meanSize,2)
    % convert energies to kWh/t; (Energy bin / mean mass) / 3.6e3;
    meanMass = ((4/3)*pi.*(meanSize(i).^3./8).*density);
    EnergyBins_kWh_t(i,:) = (EnergyBins ./ meanMass)./ 3.6e3;
    % scale EnergyBins_kWh_t up to DWT values (ScaledEnergyBins_kWh_t)
    ScaledEnergyBins_kWh_t(i,:) = EnergyBins_kWh_t(i,:) ./ (0.085 + 0.4.*exp(-0.4 ...
        .*EnergyBins_kWh_t(i,:)));% .* DEM_DWT_scaling(i);
    for d = 1:size(EnergyBins,2)
        t10(i,d) = A .*(1 - exp(-(b).*ScaledEnergyBins_kWh_t(i,d)));
    end
end

% %hardwired values for t10 and corresponding % passing values for each t_n;
% from Banini PhD thesis
hardwired_t10_values = [0 10 20 30 40 50];
hardwired_t2 = [0 55.9 81.4 93.1 98.5 100];
hardwired_t4 = [0 27.9 49.3 65.6 78.1 87.7];
hardwired_t25 = [0 4.7 10.5 16.4 22.2 28.1];
```

```
hardwired_t50 = [0 2.7 6.7 10.7 14.7 18.6];
hardwired_t75 = [0 2 5.2 8.3 11.5 14.6];

% get the functional forms by splining each y
f_2 = spline(hardwired_t10_values,hardwired_t2);
f_4 = spline(hardwired_t10_values,hardwired_t4);
f_10 = spline(hardwired_t10_values,hardwired_t10_values);
f_25 = spline(hardwired_t10_values,hardwired_t25);
f_50 = spline(hardwired_t10_values,hardwired_t50);
f_75 = spline(hardwired_t10_values,hardwired_t75);

% determine the % pass for standard size fractions from the t_n vs t10 curves;
for i = 1:size(meanSize,2)
    for d = 1:size(EnergyBins,2)
        percentPass(i,d,1) = fnval(f_2, t10(i,d));
        percentPass(i,d,2) = fnval(f_4, t10(i,d));
        percentPass(i,d,3) = fnval(f_10, t10(i,d));
        percentPass(i,d,4) = fnval(f_25, t10(i,d));
        percentPass(i,d,5) = fnval(f_50, t10(i,d));
        percentPass(i,d,6) = fnval(f_75, t10(i,d));
    end
end


% solve for a & b parameters; determine the % passing at the regular sizes
% calculate the appearance functions
for i = 1:size(meanSize,2)
    size_tn(i,:) = meanSize(i) ./ [2 4 10 25 50 75];
    for d = 1:size(EnergyBins,2)
        for k = 1:size(percentPass,3)
            passing_t(k) = percentPass(i,d,k);
        end
        [a_and_b] = RosinRamlerParamFitting([26 1],size_tn(i,:),passing_t);
        a(i,d) = a_and_b(1);
        b(i,d) = a_and_b(2);
        % get the apperance functions
        AppearedEarlier=0;
        for aa = i:size(meanSize,2)
            AppearanceFunc(i,d,aa) = 100*exp(-(meanSize(aa)/a_and_b(1))^a_and_b(2)) ...
                - AppearedEarlier;
            AppearedEarlier = sum(AppearanceFunc(i,d,:));
        end
        AppearanceFunc(i,d,aa+1) = 100-sum(AppearanceFunc(i,d,:));
    end
```

```
end
AppearanceFunc = AppearanceFunc/100; % as fractions of 1.

% save a mat file containing Appearance functions
save AppearanceFunctions2 AppearanceFunc ScaledEnergyBins_kWh_t meanSize;

disp(' ')
disp('AM DONE, THANKS!');



% Function name: fitSKt10.m
% Date: January 2007
% Purpose: This function fits the modified Shi-Kojovic breakage model to
% DWT raw data. It requires the raw data from DWT experiments and outputs the
% coefficients and r-squared value of the fit.

function [coeffs_F r_squared] = fitt10();
%Data from target ore DWT
sizes = [58.694 58.694 58.694 41.079 41.079 41.079 28.892 28.892 28.892 ...
    20.630 20.630 20.630 14.533 14.533 14.533];
filename = 'targetDWTdata.xls';
[t10data Ecs] = t10DWT(filename,sizes);
t10data1 = t10data(1:3);
t10data2 = t10data(4:6);
t10data3 = t10data(7:9);
t10data4 = t10data(10:12);
t10data5 = t10data(13:15);
EcsData1 = Ecs(1:3);
EcsData2 = Ecs(4:6);
EcsData3 = Ecs(7:9);
EcsData4 = Ecs(10:12);
EcsData5 = Ecs(13:15);
guessedCoeffs = [52 0.42]; %Guessed coefficients: M, f_mat and x.W_k_min
[coeffs_F r_squared] = t10LSfit(guessedCoeffs,Ecs,t10data); %
disp(coeffs_F)
disp(r_squared)

%Plot data points and fit
figure(2);
xRange = [0:0.1:4];
t10 = coeffs_F(1) .* (1 - exp(-coeffs_F(2).*xRange));
plot(xRange,t10,'r','LineWidth',2);
xlabel('Ecs (kWh/t)');
ylabel('t_1_0 (cum. % passing)')
```

```
hold on;
plot(EcsData1,t10data1,'bs',EcsData2,t10data2,'kd',EcsData3,t10data3,'gv', ...
    EcsData4,t10data4,'mo',EcsData5,t10data5,'c^');
legend('t_1_0 graph','58.7','41.1','28.9','20.6','14.5','Location','SouthEast')
axis tight;
hold off;

function [t10data Ecs] = t10DWT(filename,sizes);
Ecs = xlsread(filename, 'B3:P3');
appearSizes = xlsread(filename, 'A4:A23')';
appears = xlsread(filename, 'B4:P23');
for i = 1:size(Ecs,2)
    t10size = sizes(i)/10;
    passingAppears = retained2passing(appears(:,i));
    t10data(i) = spline(appearSizes,passingAppears,t10size);
end




function [coeffs r2 resnorm residual exitflag] ...
    = t10LSfit(guessedCoeffs,EcsData,t10Data);

[coeffs resnorm residual exitflag] = lsqcurvefit(@f, guessedCoeffs, EcsData, t10Data);
SSE = sum(residual.^2); %sum of the squares of the errors
SST = sum((t10Data - mean(t10Data)).^2); %sum of the squares of the deviation of the yData
r2 = 1 - (SSE/SST);

function t10=f(guessedCoeffs,EcsData); % The function to fit
A = guessedCoeffs(1);
b = guessedCoeffs(2);
t10 = guessedCoeffs(1) .*(1 - exp(-guessedCoeffs(2).*EcsData));



% retained2passing: function to convert % retained PSD to cum. % passing PSD
function passing = retained2passing(retained,start)

if (nargin == 1)
    pass_now = 100;
    for ii = 1:size(retained,1)
        passing(ii,1) = pass_now - retained(ii,1);
        pass_now = passing(ii,1);
    end
```

```
elseif (nargin == 2)
    pass_now = start;
    for ii = 1:size(retained,1)
        if ii == 1
            passing(ii,1) = start;
        else
            passing(ii,1) = pass_now - retained(ii-1,1);
            pass_now = passing(ii,1);
        end
    end
end
```

```
function [coeffs r2] = RosinRamlerParamFitting(guess,xdata,ydata);
% [coeffs] = RosinRamlerParamFitting(guess,xdata,ydata)
% This function uses the least squares curve fit MATLAB function to
% calculate the 'a' and 'b' parameters for the Rosin-Rammler curve fit
% on particle size distribution data, which are returned as coeffs:
%   guess is a 1x2 vector containg guessed values for a & b
%   xdata is the vector containing the available sizes
%   ydata is the vector containing the cummulative passing the
%       corresponding sizes in xdata
% [coeffs r2] = RosinRamlerParamFitting(guess,xdata,ydata):
%   r2 is the r-squared "quality of fit" parameter

% do a least squares fit on xdata and ydata
[coeffs resnorm residual exitflag] = lsqcurvefit(@f, guess, xdata, ydata);
% SSE = sum(residual.^2); %sum of the squares of the errors
SST = sum((ydata - mean(ydata)).^2); %sum of the squares of the deviation of the yData
r2 = 1 - (resnorm/SST);

function y=f(guess,xdata); % The function to fit
y = 100 - 100.*exp(-(xdata./guess(1)).^guess(2));
```

B.3 Main_UCM_Program

```
% File name: UCMMainProg.m
% Date: 04 October 2006
% Purpose: This is the main script for running the UCM program.
% Before running the script it is required have the DEM spectra and
% appearance functions MAT files and the file containing PSD's for the
% feed and mill contents
% ------------------------------------------------------------------------

clc; clear all; fig = 0; %close all;

% define global variables
global noMaterials noSizes noEnergies millDiameter percentSpeed ...
    DEMsimTime result1 origChargeDistr theSizeRange feedDistr...
    fig productDistr disFuncn newFeedDistr timeStep feedRate product;

% get required inputs from user
[density revs speed diameter timeStep feedRate] = UCMstartGUI;

disp('Loading DEM simulation and Experimental data ..... ')
disp(' ')
% load energy spectra. It is arranged in ascending order (lowest energy first)
load DEMSpectra;

% load the appearance functions
load AppearanceFunctions2.mat;
brkgFunction_d = AppearanceFunc;

% read in the file containing PSD
feedDistr = xlsread('UCM Excel Inputs.xls','N8:N20'); %
chargeMass = xlsread('MSP UCM Inputs & Calcs.xls','mill product','B5:B16');
origChargeDistr = xlsread('MSP UCM Inputs & Calcs.xls','mill product','J5:J16');
DEMsizes = xlsread('MSP UCM Inputs & Calcs.xls','mill product','A5:A16');
origsubDEMChargeDistr = xlsread('UCM Excel Inputs.xls','D26:D39');
recSubDEMchargeDistr = xlsread('UCM Excel Inputs.xls','E26:E39');
subDEMfeedDistr = xlsread('UCM Excel Inputs.xls','O21:O34');
subDEMsizes = xlsread('UCM Excel Inputs.xls','H21:H34');
subDEMmass = xlsread('UCM Excel Inputs.xls','M21:M34');
percSubDEM = xlsread('UCM Excel Inputs.xls','N20:N20');
% dischRate = calcDischFuncn(timeStep,feedRate);

% calculate the mean masses of the ore particles
```

```
for i=1:size(meanSize,2)
    meanMassLoss_d(i,1) = ((4/3)*pi*(meanSize(i)/2)^3)*density;
end

% initialise global variables & calculate DEM run time
noSizes = size(lambda_DEM_d,1);
noEnergies = size(lambda_DEM_d,2);
millDiameter = diameter;
percentSpeed = speed;
DEMsimTime = revs*((sqrt(millDiameter)*60)/(42.3*percentSpeed));
% theSizeRange = [212; 150; 106; 75; 53; 38; 25; 19; 12; 9.5; 6.7; 4.75];

disp('Calculating product PSD & discharge function ..... ')
disp(' ')
[lostMass_d gainedMass_d productDistr] ...
    = getMassRate2(lambda_DEM_d,meanMassLoss_d,brkgFunction_d,ScaledEnergyBins_kWh_t,meanSize);
% [lostMass_s gainedMass_s productDistr] ...
%     = getMassRate(lambda_DEM_s,meanMassLoss_s,brkgFunction_s,ScaledEnergyBins_kWh_t,meanSize,W_m_min);
grandTotalMassChange = sum(gainedMass_d,2);

disp(' ')
disp('BATCH MODE MASS BALANCE CHECK: ')
disp(' ')
disp('Mass Gain    Mass Loss')
netBreakIn = gainedMass_d' - lostMass_d';
disp([gainedMass_d' lostMass_d' netBreakIn])
disp([sum(gainedMass_d) sum(lostMass_d)])

newChargeMass = chargeMass + netBreakIn; %(1:end-1)

% sub-DEM breakage
[bondWI feedRate subDEMPower] = subDEMdata;
[newSubDEMmass newSubDEMfeedDistr] = subDEMshift(bondWI,feedRate,subDEMPower,subDEMmass, ...
                            subDEMfeedDistr,subDEMsizes,percSubDEM,timeStep);

% Combine DEM and sub-DEM results
sizes = cat(1,DEMsizes,subDEMsizes);

origMasses = cat(1,chargeMass,subDEMmass);
origRetained = (origMasses ./ sum(origMasses,1)) .* 100;
origDistr = retained2passing(origRetained,100);

newMasses = cat(1,newChargeMass,newSubDEMmass);
newRetained = (newMasses ./ sum(newMasses,1)) .* 100;
```

```
newDistr = retained2passing(newRetained,100);

plotDistrResults(sizes,newDistr,origDistr);
% %plot the sub DEM results
% plotSubDEMResults(subDEMsizes,newSubDEMfeedDistr,subDEMfeedDistr);

% newMass = [product(2:end-1,1); newSubDEMmass];
% newPctRetained = 100.*(newMass./sum(newMass,1));
% newPctPass = [100; retained2passing(newPctRetained)];
%
% origSieveSizes = [theSizeRange; subDEMsizes];
% subDEMPSD = xlsread('initialPSD.xls','Sheet2','D15:D29');
% origPSD = xlsread('MINTEK 29 April msp2','Sheet1','H5:H29');
% % origPSD = [origChargeDistr(1:end-1,1); subDEMPSD];
% newSieveSizes = [theSizeRange; productSizes];
% % newPSD = [productDistr(1:end-1,1); subDEMPSD];
% % result1 = num2str([[theSizeRange;-9.5] chargeMass product product-chargeMass]);
% % UCMresults; %comparison of mass of initial charge and final product
% %plot PSD of the feed, intial charge and final products
% newPSD = newPctPass;
% plotUCMResults(origSieveSizes,origPSD,newSieveSizes,newPSD,timeStep,dischRate);
%



function varargout = UCMstartGUI(varargin)
% UCMSTARTGUI M-file for UCMstartGUI.fig
%      UCMSTARTGUI, by itself, creates a new UCMSTARTGUI or raises the existing
%      singleton*.
%
%      H = UCMSTARTGUI returns the handle to a new UCMSTARTGUI or the handle to
%      the existing singleton*.
%
%      UCMSTARTGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in UCMSTARTGUI.M with the given input arguments.
%
%      UCMSTARTGUI('Property','Value',...) creates a new UCMSTARTGUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before UCMstartGUI_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to UCMstartGUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
```

```
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help UCMstartGUI

% Last Modified by GUIDE v2.5 21-Oct-2007 21:53:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @UCMstartGUI_OpeningFcn, ...
                   'gui_OutputFcn',  @UCMstartGUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before UCMstartGUI is made visible.
function UCMstartGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to UCMstartGUI (see VARARGIN)

% Choose default command line output for UCMstartGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes UCMstartGUI wait for user response (see UIRESUME)
uiwait(handles.theUCM_fig);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = UCMstartGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.Density;
varargout{2} = handles.revs;
varargout{3} = handles.speed;
varargout{4} = handles.diameter;
varargout{5} = handles.TimeStep;
varargout{6} = handles.FeedRate;
delete(handles.theUCM_fig);


function mill_speed_Callback(hObject, eventdata, handles)
% hObject    handle to mill_speed (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of mill_speed as text
%        str2double(get(hObject,'String')) returns contents of mill_speed as a double


% --- Executes during object creation, after setting all properties.
function mill_speed_CreateFcn(hObject, eventdata, handles)
% hObject    handle to mill_speed (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in OK.
```

```
function OK_Callback(hObject, eventdata, handles)
global result1
% hObject    handle to OK (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
millSpeed = str2double(get(handles.mill_speed, 'String'));
if isnan(millSpeed)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
no_revs = str2double(get(handles.noRevs, 'string'));
if isnan(no_revs)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
millDiameter = str2double(get(handles.mill_diameter, 'string'));
if isnan(millDiameter)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
theTimeStep = str2double(get(handles.timeStep, 'string'));
if isnan(theTimeStep)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
theFeedRate = str2double(get(handles.feedRate, 'string'));
if isnan(theFeedRate)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
theOreDensity = str2double(get(handles.oreDensity, 'string'));
if isnan(theOreDensity)
    errordlg('You must enter a numeric value','Bad Input','modal')
end

handles.speed = millSpeed;
handles.revs = no_revs;
handles.diameter = millDiameter;
handles.TimeStep = theTimeStep;
handles.FeedRate = theFeedRate;
handles.Density = theOreDensity;

guidata(GCBO, handles);
uiresume;


function noRevs_Callback(hObject, eventdata, handles)
```

```
% --- Executes during object creation, after setting all properties.
function noRevs_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function mill_diameter_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function mill_diameter_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function timeStep_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function timeStep_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function feedRate_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function feedRate_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function OreDensity_Callback(hObject, eventdata, handles)
% hObject    handle to OreDensity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of OreDensity as text
%        str2double(get(hObject,'String')) returns contents of OreDensity as a double


% --- Executes during object creation, after setting all properties.
function OreDensity_CreateFcn(hObject, eventdata, handles)
% hObject    handle to OreDensity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function oreDensity_Callback(hObject, eventdata, handles)
% hObject    handle to oreDensity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of oreDensity as text
%        str2double(get(hObject,'String')) returns contents of oreDensity as a double


% --- Executes during object creation, after setting all properties.
function oreDensity_CreateFcn(hObject, eventdata, handles)
% hObject    handle to oreDensity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% File name: getMassRate.m
% Date: October 2006
% Purpose: This function calculates "mass" breakage rate
% Inputs: Energy spectra, ore particle masses and sizes, appearance functions,
%         collision energy bins and threshold breakage energies
% Outputs: broken mass, gained mass and PSD
% ------------------------------------------------------------------------

function [lostMass gainedMass productPass] = getMassRate2(lambda_DEM,meanMassLoss,...
    brkgFunction,Ecs_DEM_DWT,meanSize);

global noMaterials noSizes noEnergies millDiameter percentSpeed DEMsimTime timeStep ...
    feedDistr feedRate origChargeDistr product;

%E0 and Ecrit
E_0 = 0.10022744/3600;
Ecrit = 0.01; %J/kg

% sub-DEM size will not break; set all relevant values to zero!
theEnd = noSizes+1;
brkgFunction(theEnd,:,:) = 0;
lostMass(theEnd) = 0;
breakageRate(theEnd,:) = 0;

% calculate the mass lost per particle class (i)
    for i = 1:noSizes
        lostMass(i) = 0; % reset lost mass counter
%         Ecrit_J_kg = Ecrit / meanSize(i);
        for d = 1:noEnergies
            if Ecs_DEM_DWT(i,d) >= E_0;
                if Ecs_DEM_DWT(i,d) >= Ecrit
                    breakageRate(i,d) = ((lambda_DEM(i,d) * timeStep) / DEMsimTime) ...
                        * meanMassLoss(i);
                    lostMass(i) = lostMass(i) + breakageRate(i,d);
                else
```

```
                    breakageRate(i,d) = (((lambda_DEM(i,d) * timeStep)/ DEMsimTime) ...
                        * meanMassLoss(i)) * (Ecs_DEM_DWT(i,d) / Ecrit);
                    lostMass(i) = lostMass(i) + breakageRate(i,d);
                end
            else
                lostMass(i) = 0;
                breakageRate(i,d) = 0;
            end
        end
    end
%     lostMass = [0,lostMass];

% calculate the mass gain per particle class (i)
    for i = 1:noSizes+1; % include sub-DEM fines here
        gainedMass(i) = 0;% initialise gained mass
        for j = 1:i
            for d = 1:noEnergies
                gainedMass(i) = gainedMass(i) + breakageRate(j,d)*brkgFunction(j,d,i);
            end
        end
    end
%     gainedMass = [0,gainedMass];

% add the feed to obtain the product!
chargeRetained = passing2retained(origChargeDistr);
chargeRetained(end+1,1) = 100 - sum(chargeRetained,1);;
feedRetained = passing2retained(feedDistr);
feedRetained(end+1,1) = 100 - sum(feedRetained,1); % to cater for sub-DEM feed
totalFeed = feedRate*1000/3600 * timeStep; % total feed in kg (for UCM time step)
    for i = 1:noSizes+1 % include sub-DEM sizes
        chargeMass(i,1) = ((chargeRetained(i,1)/100)*550);
        product(i,1) = chargeMass(i,1) + ((feedRetained(i,1)/100) * totalFeed) ...
            + gainedMass(i) - lostMass(i);
    end

% calculate the product PSD
productRetained = (product ./ sum(product)) .* 100;
productPass = retained2passing(productRetained);



% File name: passing2retained.m
% Date: October 2006
% Purpose: This function converts % passing PSD to % retained
% Inputs: % passing values
```

```
% Outputs: % retained values
% ------------------------------------------------------------------------

function retained = passing2retained(passing,start)

if (nargin == 1)
    pass_now = 100;
    for ii = 1:size(passing,1)
        retained(ii,1) = pass_now - passing(ii,1);
        pass_now = passing(ii,1);
    end
elseif (nargin == 2)
    pass_now = passing(start,1);
    for ii = start+1:size(passing,1)
        retained(ii-start,1) = pass_now - passing(ii,1);
        pass_now = passing(ii,1);
    end
end



% File name: retained2passing.m
% Date: October 2006
% Purpose: This function converts % retained PSD to % passing
% Inputs: % retained values
% Outputs: % passing values
% ------------------------------------------------------------------------

function passing = retained2passing(retained,start)

if (nargin == 1)
    pass_now = 100;
    for ii = 1:size(retained,1)
        passing(ii,1) = pass_now - retained(ii,1);
        pass_now = passing(ii,1);
    end
elseif (nargin == 2)
    pass_now = start;
    for ii = 1:size(retained,1)
        if ii == 1
            passing(ii,1) = start;
        else
            passing(ii,1) = pass_now - retained(ii-1,1);
            pass_now = passing(ii,1);
        end
```

```
    end
end



% File name: calcDischFuncn.m
% Date: October 2006
% Purpose: This function calculates the predicted discharge function
% It requires the feed rate and the UCM program time step and outputs
% the discharge rate
% ------------------------------------------------------------------------

% function to calculate the predicted discharge function
function [dischRate] = calcDischFuncn(timeStep,feedRate);

feedRate = feedRate*1000/(3600*timeStep); %convert feed rate to /sec
dischMassRate = xlsread('MINTEK 29 April msp','discharge','C7:C32')./100.*feedRate; %kg/s
millContents = xlsread('MINTEK 29 April msp','discharge','B7:B32'); %in mass, kg
dischRate = dischMassRate ./ millContents; %discharge rate (/s)




function [newSubDEMmass product] = subDEMshift(bondWI,feedRate,subDEMPower,subDEMmass, ...
                            subDEMfeedDistr,subDEMsizes,percSubDEM,timeStep)

%the power expended in subE0 energies is calculated in file subE0work.m
%load subE0energy.mat which is saved by subE0work.m
% load subE0energy;
% thePower = totalSubE0e/(2*2.462);
% percSubDEM = 32.81; %NEEDS TO AUTODIAL THIS!!!! of the contents that is sub DEM
thruputSubDEM = percSubDEM/100*feedRate; %assumed to be based on actual feed (tph)
%get f80 from subDEM contents
f80 = spline(subDEMfeedDistr,subDEMsizes,80); %in mm


%calculate p80 (mm)
p80 = ((10*sqrt(f80)*thruputSubDEM*bondWI)/((subDEMPower*sqrt(f80)+10*bondWI*thruputSubDEM)))^2; %
newSubDEMsizes = (subDEMsizes.*p80)./f80; % new product sizes at the original % passing values
%obtain the new function. use cubic splines
f80 = p80;
newFunction = spline(newSubDEMsizes,subDEMfeedDistr);
product = fnval(newFunction,subDEMsizes);

subDEMretained = passing2retained(product);
```

```
newMass = (subDEMretained./100).*sum(subDEMmass,1);
massDiff = (newMass - subDEMmass) .* timeStep;
newSubDEMmass = subDEMmass + massDiff;



function plotUCMResults(origSieveSizes,origPSD,newSieveSizes,newPSD,timeStep,dischRate); ;
% plotUCMresults plot the particle size distribution of the feed, initial
% charge and the product of the UCM simulation

close all
%plot PSD's
figure('name','UCM RESULTS - PSDs','NumberTitle','off')
plotSize = origSieveSizes(1):-0.01:origSieveSizes(end);
% plotFeedDistr = spline(theSizeRange,feedDistr(1:end-1,1),plotSize);
plotOrigPSD = spline(origSieveSizes,[origPSD;0],plotSize);
plotNewPSD = spline(origSieveSizes,[newPSD;0],plotSize);
% newPlotSize = [plotSize,subDEMsizes(2:end)'];
% newPlotChargeDistr = [plotChargeDistr,subDEMcontents(2:end)'];
% newPlotProductDistr = [plotProductDistr,newContents(2:end)'];
semilogx(plotSize,plotOrigPSD,'r',plotSize,plotNewPSD,'--b','LineWidth',2.6)%newPlotSize,plotFeedDistr,'

hold on;
% semilogx(theSizeRange,feedDistr(1:end-1,1),'ko',theSizeRange,productDistr(1:end-1,1),'bo', ...
%     theSizeRange,chargeDistr(1:end-1,1),'ro')
legend('Initial Contents','Evolved Contents','Feed','Location','SouthEast')
xlabel('Size (mm)','FontWeight','demi','FontSize',11)
ylabel('Cum. Passing (%)','FontWeight','demi','FontSize',11)
title('Particle Size Distibutions','FontWeight','bold','FontSize',12)
grid on;  hold on;
axis tight
% %plot the discharge function
% figure('name','UCM RESULTS - Dis. Funcn','NumberTitle','off')
% fullSizes = xlsread('MINTEK 29 April msp','discharge','A7:A32');
% semilogx(fullSizes,dischRate,'ro','LineWidth',2.6)
% hold on
% grid on;
% xlabel('size (mm)')
% ylabel('Discharge Rate')
% axis tight
% %fit the standard discahrge rate
% %region 1
% coeffss = fitLine(1,fullSizes(end-2:end),dischRate(end-2:end));
% yfit1 = coeffss(1).*[1;1;1];
```

```
% semilogx(fullSizes(end-2:end),yfit1,'g','LineWidth',3)
% hold on
% %region 2
% xdata2 = fullSizes(end-11:end-2);
% ydata2 = dischRate(end-11:end-2);
% coeffs2 = fitLine(2,xdata2,ydata2);
% yfit2 = coeffs2(2).*xdata2.^2 + coeffs2(1).*xdata2 + coeffs2(3);
% plot(xdata2,yfit2,'g','LineWidth',3)



function [newSubDEMmass product] = subDEMshift(bondWI,feedRate,subDEMPower,subDEMmass, ...
                            subDEMfeedDistr,subDEMsizes,percSubDEM,timeStep)

%the power expended in subE0 energies is calculated in file subE0work.m
%load subE0energy.mat which is saved by subE0work.m
% load subE0energy;
% thePower = totalSubE0e/(2*2.462);
% percSubDEM = 32.81; %NEEDS TO AUTODIAL THIS!!!! of the contents that is sub DEM
thruputSubDEM = percSubDEM/100*feedRate; %assumed to be based on actual feed (tph)
%get f80 from subDEM contents
f80 = spline(subDEMfeedDistr,subDEMsizes,80); %in mm


%calculate p80 (mm)
p80 = ((10*sqrt(f80)*thruputSubDEM*bondWI)/((subDEMPower*sqrt(f80)+10*bondWI*thruputSubDEM)))^2; %
newSubDEMsizes = (subDEMsizes.*p80)./f80; % new product sizes at the original % passing values
%obtain the new function. use cubic splines
f80 = p80;
newFunction = spline(newSubDEMsizes,subDEMfeedDistr);
product = fnval(newFunction,subDEMsizes);

subDEMretained = passing2retained(product);
newMass = (subDEMretained./100).*sum(subDEMmass,1);
massDiff = (newMass - subDEMmass) .* timeStep;
newSubDEMmass = subDEMmass + massDiff;
```