Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Shahriar Ahmadi Ghoohaki

Entitled
High-Fidelity Modeling of a Backhoe Digging Operation using an Explicit Multibody Dynamics Finite Element Code with Integrated Discrete Element Method.

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Tamer Wasfy
Chair

Hazim El-Mounayri

Likun Zhu

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Tamer Wasfy

Approved by: Jie Chen 05/10/2013
Head of the Graduate Program Date

HIGH-FIDELITY MODELING OF A BACKHOE DIGGING OPERATION USING AN EXPLICIT MULTIBODY DYNAMICS FINITE ELEMENT CODE WITH INTEGRATED DISCRETE ELEMENT METHOD

A Thesis

Submitted to the Faculty

of

Purdue University

by

Shahriar Ahmadi Ghoohaki

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

August 2013

Purdue University

Indianapolis, Indiana

ACKNOWLEDGMENTS

This thesis could not have been achievable without the assistance and support of several individuals who in one way or another contributed their valuable help in the preparation and completion of this study.

First of all, I would like to express my special thanks to Dr. Tamer Wasfy for providing this opportunity for me to learn and gain this much experience throughout my study. Without his technical advice and financial support, this work could not be done.

Also, I want to thank my advising committee, Dr. Hazim El-Mounayri and Dr. Likun Zhu for their time and direction during the completion of this thesis.

I also appreciate Mechanical Engineering Department faculty for their instruction and advice through my undergraduate and graduate courses.

I would also like to extend my gratefulness to Valerie Lim Diemer for her kindness in formatting this thesis and assisting me during my graduate studies.

I would like thank my dear parents and sister for supporting and motivating me through all the stages of my life. I am very proud to have such a nice and warm family.

Last but not least, I want to say thank you to my lovely wife who supports me patiently and provides me a quiet atmosphere for studying. I am very lucky to have her close to myself all the time.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Ahmadi Ghoohaki, Shahriar. M.S.M.E., Purdue University, August 2013. High-Fidelity Modeling of a Backhoe Digging Operation using an Explicit Multibody Dynamics Finite Element Code with Integrated Discrete Element Method. Major Professor: Tamer Wasfy.

In this thesis, a high-fidelity multibody dynamics model of a backhoe for simulating the digging operation is developed using the DIS (Dynamic Interactions Simulator) multibody dynamics software. Sand is used as a sample digging material to illustrate the model. The backhoe components (such as frame, manipulators links, track segments, wheels and sprockets) are modeled as rigid bodies. The geometry of the major moving components of the backhoe is created using the Pro/E solid modeling software. The components of the backhoe are imported to DIS and connected using joints (revolute, cylindrical and prismatic joints). Rotary and linear actuators along with PD (Proportional-Derivative) controllers are used to move and steer the backhoe and to move the backhoes manipulator in the desired trajectory. Sand is modeled using cubic shaped particles that can come into contact with each other, the backhoes bucket and ground. A cubical sand particle contact surface is modeled using eight spheres that are rigidly glued to each other to form a cubical shaped particle, The backhoe and ground surfaces are modeled as polygonal surfaces. A penalty technique is used to impose both joint and normal contact constraints (including track-wheels, track-terrain, bucket-particles and particles-particles contact). An asperity-based friction model is used to model joint and contact friction. A Cartesian Eulerian grid contact search algorithm is used to allow fast contact detection between particles. A recursive bounding box contact search algorithm is used to allow fast contact detection for polygonal contact surfaces and is used to detect contact between: track and ground; track and wheels; bucket and particles; and ground and

particles. The governing equations of motion are solved along with joint/constraint equations using a time-accurate explicit solution procedure.

The sand model is validated using a conical hopper sand flow experiment in which the sand flow rate during discharge and the angle of repose of the resulting sand pile are experimentally measured. The results of the conical hopper simulation are compared with previously published experimental results. Parameter studies are performed using the sand model to study the effects of the particle size and the orifices diameter of the hopper on the sand pile angle of repose and sand flow rate.

The sand model is integrated with the backhoe model to simulate a typical digging operation. The model is used to predict the manipulators actuator forces needed to dig through a pile of sand. Integrating the sand model and backhoe model can help improving the performance of construction equipment by predicting, for various vehicle design alternatives: the actuator and joint forces, and the vehicle stability during digging.

1. INTRODUCTION

1.1 Motivation

Earth moving equipment is used in different industries such as construction, agriculture, and mining. Earth moving equipment manufacturers are looking for ways to improve the performance of heavy duty earth moving equipment in order to achieve faster excavation operations, and at the same time increase workers safety. High-fidelity multibody dynamic analysis of earth moving equipment can help achieve these goals by quickly predicting the performance of alternative designs.

Multibody dynamic simulation tools can be used to predict the positions, orientations and internal forces/torques of the vehicles components during various operating scenarios. A typical earth moving vehicle model such as a backhoe can be represented as a complex multibody system which consists of:

- Rigid and/or flexible components.
- The components are connected using various types of joints such as revolute, cylindrical and prismatic joints.
- Contact between: the vehicles tires/track and ground, the track and the wheels, and the bucket and ground.
- Linear and rotary actuators.
- Control systems to control the speed and steering of the vehicle and to control the manipulator.

1.2 Literature Review

Multibody dynamics software has been used to model earth moving equipment, including: predicting the actuator forces/torques needed to generate a prescribed motion [1], predicting the actuator forces during digging [2] to [4], and studying the vehicles motion stability on inclined terrains [5]. Generally, PD (Proportional-Derivative) controllers are used in order to keep the bucket in track with the desired trajectory [4].

In the aforementioned papers the terrain was modeled as a rigid surface. However, terrains that earth moving vehicles operate on are typically compliant granular/semi-solid terrains such as sand, mud, gravel, snow, etc. In order to predict the digging forces, vehicle mobility, and vehicle stability on those types of terrains, a more accurate terrain model is needed. Computational simulation methods that have been used for modeling granular materials include Monte Carlo, cellular automata, and granular dynamic simulations. However, these methods have not been accurate enough to compute the dynamic internal forces and displacement between discrete particles. Accurate terrain models can be achieved using the discrete element method (DEM) by modeling the terrain using particles. The DEM is the most accurate and promising method to model granular materials because both particle geometry and contact forces can be included in the model. The method was first proposed in [6] , [7], for modeling granular materials and fluids. In the DEM inter-particle forces include: normal contact forces which prevent the particles from penetrating each other, tangential contact forces and distance dependent forces [7] to [13]. Normal contact forces include spring (normal deflection dependent) and damping (normal velocity dependent) forces. Tangential contact forces include friction and viscous (tangential velocity dependent) forces. Distance dependent forces include gravity, electrostatic and magnetic forces.

Normal and tangential contact forces are present between two particles, only if the two particles are in contact at one or more contact points. The accuracy of the DEM simulations depends on the force model and particle geometric model.

Most DEM papers represented the particle geometry using circles in 2D and spheres in 3D [7] to [13]. A penalty spring technique is used to prevent a particle from penetrating into neighboring particles, where the normal contact force between two particles is equal to the normal penetration distance between the two particles multiplied by the penalty spring stiffness. Viscous damping forces in the normal and the tangential directions are added to predict the particles rotational and translational tangential motions. The maximum penetration magnitude and the required coefficient of restitution are defined by the penalty spring stiffness and damping coefficient in the normal direction [7] , [9]. In [13] the nonlinear damping constant that depends on both displacement and velocity was derived based on Hertzian theory and the coefficient of restitution.

When circular/spherical particles are used, simulation of granular materials pile formation and predicting the angle of repose of the pile have been challenging due to the problem of stabilizing a conical pile of circular/spherical particles after pouring and preventing them from rolling under gravity. In order to solve this problem, some assumptions were used to simplify the models; such as ignoring the rotation of particles and tangential forces. Those assumptions caused inaccurate results, especially in predicting the angle of repose and particle flow rate through an orifice [8]. In [8] to [10] a method to stabilize a pyramid of sand is presented by incorporating rolling friction into the model. Experimental methods as well as a numerical method (DEM) were used to validate the model. The effects of the rolling and sliding friction coefficient and particle size on the angle of repose of the sand pile were studied [8].

Rolling friction is an approximate method to account for particle shape using spherical particles. Accurate accounting of the particle shape is critical for accurate simulation of particle piles and particle flow [14] to [16]. Using the actual particle geometrical shape in the model allows a pile to achieve a stable angle of repose

without adding non-physical rolling friction. In addition, it allows the particles to flow correctly without adding additional sliding friction. In [15] it was shown that 2D rectangular and oval particles flow as much as 30% slower than circular particles. In [16] polyhedral particles were used to model a particle box shear test. An incremental model for the tangential Coulomb friction force was used. Normal and tangential damping were added to eliminate undesirable vibrations during the simulations. The tangential damping force was capped to be less than the tangential friction force to ensure nearly accurate particle sliding behavior.

Another key component of the DEM is searching and detection of inter-particle contacts. In [17] a space decomposition contact search algorithm was developed in which the space where the particles can move is decomposed into a Cartesian Eulerian grid of cubic boxes. The list of boxes that each particle intersects are found, then the list of particles that each box intersects are found. The neighboring particles that can come into contact with a particle are the particles that intersect the boxes that the particle intersects. Thus, only the neighboring particles are searched for a contact with a particle. This makes the contact search algorithm an $O(N)$ algorithm instead of $O(N^2)$ for an algorithm where each particle is checked against all other particles. In [16] a second level search was added, to improve the speed of the algorithm, in which the bounding sphere for a particle is checked with the bounding spheres of the neighboring particles. If the bounding spheres do not intersect then the particles are not in contact.

In [18] a two-dimensional DEM particle method was used along with a bucket model to predict the digging forces. The DEM simulations were compared to experiments of digging corn grains. Results of the free-surface shape and bucket force did not agree well with the experimental results most likely due to the fact that the particles were modeled using spheres rather than the actual particle shape.

1.3 Objective

In this thesis, a three-dimensional DEM model is used along with a full backhoe multibody dynamics model to predict the actuators forces needed to dig through a granular material pile. The DEM model, that is used to model the granular material, supports arbitrarily shaped 3D particles, where each particle is modeled as a rigid body. The particle shape can be represented using: a superquadric surface, a polygonal surface or an assembly of spheres rigidly fixed to the rigid body. The vehicle components are also modeled as rigid bodies. The rigid bodies rotational equations of motion are written in a body-fixed frame with the total rigid body rotation matrix updated each time step using incremental rotations [25]. An asperity-based Coulomb friction model is used to model tangential friction forces [26] between the particles and each other and between the particles and the vehicle components. The friction model along with the accurate representation of the particle shape allows the formation of a stable (static) particle pile and accurate behavior of particles during sliding. A penalty contact technique, with a penalty spring and an asymmetric damper, is used to model normal contact forces. Tangential damping (viscous) forces can also be included in the model. The Eulerian grid space decomposition along with a bounding sphere contact check used in [18] are used in the present model to speed inter-particle contact detection. Contact between rigid bodies (including the particles and the vehicle components) and polygonal contact surface(s) (such as the surfaces of the vehicle components) is detected using a binary tree contact search algorithm which allows fast contact search. A penalty formulation is also used for modeling joint constraints including spherical, revolute, cylindrical and prismatic joints [12]. The equations of motion are integrated using a time accurate explicit predictor-corrector solution procedure.

The main contributions of this thesis are:

- Creating a sand model using the DEM modeling capability of a high-end commercial multibody dynamics code and validating the model using previously

published experimental results. In particular, the model was used to predict the flow rate of sand from an orifice as a function of the orifice diameter and the sand particle size. The model was also used to predict the stable angle of repose of a sand pile.

- Creating a high-fidelity integrated multibody dynamics model of a backhoe and DEM model for granular materials, using the multibody dynamics code. The integrated model is used to simulate a typical backhoe digging operation. The validated sand model is used as a sample granular material to illustrate the model. The integrated model is used to predict time-histories of: position and orientation of the backhoes components and the sand particles; and backhoes actuator forces needed for executing the digging operation.

1.4 Tools Used

In this research, two main CAD/CAE software tools were used to create the backhoe and sand models: DIS (Dynamic Interactions Simulator) [29] and Pro-Engineer [34].

The DIS [29] finite element/multibody dynamics code was used to create the multibody dynamics models of the backhoe and the DEM sand model. DIS is a commercial flexible multibody dynamics code that can be used to simulate mechanical systems which include multiple rigid and flexible components that can come into contact with each other and that are connected using various types of joints. The code also allows adding linear and rotary actuators and the control systems controller used along with those actuators. The DIS code uses a computationally efficient explicit predictor-corrector time integration method to solve the equations of motion. DIS has been used to predict the dynamic response for many multibody dynamic applications including automotive, aerospace and industrial applications. Further details about the formulation and solution procedure used in the DIS code are provided in Chapter 2.

Pro-Engineer was used to create the geometry for the various backhoe components. Pro-Engineer is one of the most popular CAD packages to create the geometry of various components and complicated assemblies.

1.5 Thesis Organization

This thesis is organized as follows. In Chapter 2 the multibody dynamics formulation and solution procedure used in the DIS code are presented. In Section 2.1, the translational and rotational semi-discrete equations of motion are presented. In Section 2.2, the contact model including the normal force model and tangential asperity friction force model; contact search algorithm; and particle shape representations are presented. In Section 2.3, the penalty algorithm for imposing joint constraints is presented. Linear and rotary actuators models are outlined in Section 2.4. In Section 2.5, the explicit solution procedure is presented. The sand model and simulation results of the sand flow from a conical hopper and the sand pile angle of repose along with comparison to previously published experiments are presented in Chapter 3.The backhoe multibody dynamics model and the sand model, along with the simulation results of the digging operation are presented in Chapter 4. Finally, concluding remarks are given in Chapter 5.

2. MULTIBODY DYNAMICS FORMULATION

In this chapter, the equations of motion along with the time-integration rule used in the DIS [25, 29] code are outlined. In the subsequent equations the following conventions are used:

- The indicial notation is used.
- The Einstein summation convention is used for repeated subscript indices unless otherwise noted.
- Upper case subscript indices denote node numbers.
- Lower case subscript indices denote vector component number.
- The superscript denotes time.
- A superposed dot denotes a time derivative.

2.1 Equations of Motion

The vehicles components and the particles are modeled as rigid bodies. A rigid body is modeled as a finite element node located at the rigid bodys center of mass. The algorithm for writing and integrating the equations of motion for three-dimensional rigid bodies using an explicit finite element code was presented in [25]. Each node (rigid body) has 3 translational degrees-of-freedom defined with respect to the global inertial reference frame and a rotation matrix defined also with respect to the global inertial frame. The use of the total body rotation matrix to measure rigid body rotation avoids singularity problems associated with 3 and 4 parameter rotation measures such as Euler angles and Euler parameters [26, 27].

The translational equations of motion for the nodes are written with respect to the global inertial reference frame and are obtained by assembling the individual node equations. The equations can be written as Equation (2.1):

$$M_K \ddot{x}^t_{ki} = F^t_{s_{Ki}} + F^t_{a_{Ki}} \tag{2.1}$$

where t is the running time, K is the global node number (no summation over K; $K = 1 \rightarrow N$ where N is the total number of nodes), i is the coordinate number ($i = 1, 2, 3$), M_K is the lumped mass of node K, x is the vector of nodal Cartesian coordinates with respect to the global inertial reference frame, and $\ddot{x}$ is the vector of nodal accelerations with respect to the global inertial reference frame, F_s is the vector of internal structural forces, and F_a is the vector of externally applied forces, which include surface forces and body forces. For each rigid body (node), a body-fixed material frame is defined. This origin of the body frame is located at the bodys center of mass. The mass of the body is concentrated at the center of mass and the inertia of the body is given by the inertia tensor I_{ij} defined with respect to the body frame. The orientation of the body-frame is given by $R^{t_0}_k$ which is the rotation matrix relative to the global inertial frame at time t_0. The rotational equations of motions are written for each node with respect to its body-fixed material frames as Equation (2.2):

$$I_{Kij} \ddot{\theta}^t_{Kj} = T^t_{s_{ki}} + T^t_{a_{ki}} - (\dot{\theta}^t_{Ki} \times (I_{Kij} \dot{\theta}^t_{Kj}))_{Ki} \tag{2.2}$$

Where I_k is the inertia tensor of rigid body K, $\ddot{\theta}_{kj}$ and $\dot{\theta_{kj}}$ are the angular acceleration and velocity vectors components for rigid body K relative to its material frame in direction $j (j = 1, 2, 3)$,$T_{s_{Ki}}$ are the components of the vector of internal torque at node K in direction i, and $T_{a_{Ki}}$ are the components of the vector of applied torque. The summation convention is used only for the lower case indices i and j. Since, the rigid body rotational equations of motion are written in a body (material) frame, the inertia tensor I_K is constant.

The trapezoidal rule is used as the time integration formula for solving Equation (2.1) for the global nodal positions x:

$$\dot{x}^t_{Kj} = \dot{x}^{t-\Delta t}_{Kj} + 0.5\Delta t(\ddot{x}^t_{Kj} + \ddot{x}^{t-\Delta t}_{Kj}) \tag{2.3}$$

$$x^t_{Kj} = x^{t-\Delta t}_{Kj} + 0.5\Delta t(\dot{x}^t_{Kj} + \dot{x}^{t-\Delta t}_{Kj}) \tag{2.4}$$

where Δt is the time step. The trapezoidal rule is also used as the time integration formula for the nodal rotation increments:

$$\dot{\theta}^t_{Kj} = \dot{\theta}^{t-\Delta t}_{Kj} + 0.5\Delta t(\ddot{\theta}^t_{Kj} + \ddot{\theta}^{t-\Delta t}_{Kj}) \tag{2.5}$$

$$\Delta\theta^t_{Kj} = 0.5\Delta t(\dot{\theta}^t_{Kj} + \dot{\theta}^{t-\Delta t}_{Kj}) \tag{2.6}$$

where $\Delta\theta_{Kj}$ are the incremental rotation angles around the three body axes for body K. Thus, the rotational equations of motion are integrated to yield the incremental rotations angles. The rotation matrix of body $K(R_K)$ is updated using the rotation matrix corresponding to the incremental rotations angles:

$$R^t_K = R^{t-\Delta t}_K R(\Delta\theta^t_{Ki}) \tag{2.7}$$

Where $R(\Delta\theta^t_{Ki})$ is the rotation matrix corresponding to the incremental rotation angles from Equation (2.6). The explicit solution procedure used for solving Equations (2.1)-(2.7) along with constraint equations are presented in Section 2.5. The constraint equations are generally algebraic equations, which describe the position or velocity of some of the nodes. They include:

- Contact/impact constraints (Section 2.2):

$$f(\{x\}) \geq 0 \tag{2.8}$$

- Joint constraints (Section 2.3):

$$f(\{x\}) = 0 \tag{2.9}$$

- Prescribed motion constraints:

$$f(\{x\}, t) = 0 \tag{2.10}$$

2.2 Contact Model

The penalty technique is used to impose the normal contact constraints between contact points on a rigid body and a surface on another rigid body [25], [27]. The first step is to find the position and velocity of a contact point on a rigid body. The global position x_{Gp} and velocity $\dot{x}_{Gp}$ of a contact point are given by:

$$x_{Gp_i} = X_{B_i} + R_{B_{ij}} x_{Lp_j} \tag{2.11}$$

$$\dot{x}_{Gp_i} = \dot{X}_{B_i} + R_{B_{ij}} (W_{BF} \times x_{Lp})_j \tag{2.12}$$

Where X_B and $\dot{X}_B$ are the global position and velocity vectors of the rigid bodys frame, R_B is the rotation matrix of the rigid body relative to the global reference frame, W_B is the rigid bodys angular velocity vector relative to its local frame, and x_{LP} is the position of the contact point relative to the rigid bodys frame. The contact

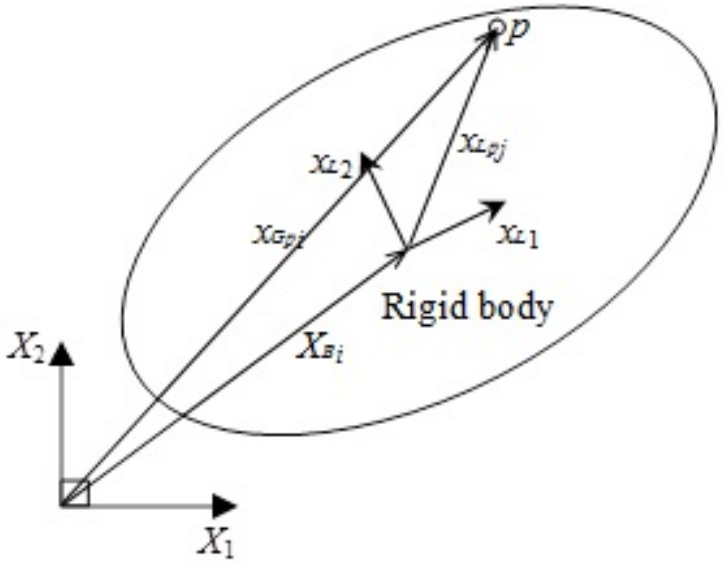

Fig. 2.1. Location of a point on a rigid body with respect to the local body frame (X_{LP}) and the global reference frame (X_{GP}). [30]

force Fc at each contact point is the sum of the normal contact force (Fn) and tangential friction force (Ft):

$$F_{c_i} = F_{t_i} + F_{n_i} \tag{2.13}$$

Fc is transferred as a force and a torque to the center of the rigid body (the node) using:

$$F_i = F_{c_i} \tag{2.14}$$

$$T_i = (x_{LP_i} \times R_{BF_{ji}} F_{c_i}) \tag{2.15}$$

$$x_{Lp_j} = R_{BF_{ji}}(x_{Gp_i} - X_{BF_i}) \tag{2.16}$$

Where F_i is the contact force at the CG of the rigid body (center of the body frame), T_i is the contact moment on the rigid body, $x_{L_{cp}}$ is the position of the contact point relative to the rigid bodys frame and $x_{G_{cp}}$ is the position of the contact point relative to the global reference frame. Similarly, the negative of this force is transferred to the center of the contacting rigid body as a force and moment.

2.2.1 Penalty Normal Contact Model

The penalty technique is used for insuring that particles or rigid bodies do not interpenetrate. A normal reaction force (Fn) is generated when a node penetrates in a contact body whose magnitude is proportional to the penetration distance. The force is given by:

$$|fn| = AK_p d + A \begin{cases} C_p \dot{d} & \dot{d} \geq 0 \\ S_p C_p \dot{d} & \dot{d} < 0 \end{cases} \tag{2.17}$$

$$\dot{d} = v_{rel_i} \tag{2.18}$$

$$v_{n_i} = \dot{d}_{n_i} \tag{2.19}$$

$$V_{t_i} = v_{rel_i} - v_{n_i} \tag{2.20}$$

Where A is the area associated with the contact point, K_p and C_p are the penalty stiffness and damping coefficients per unit area, respectively; d is the closest distance between the node and the contact surface in Figure 2.2; $\dot{d}$ is the signed time rate of change of d; S_p is a separation damping factor between 0 and 1 which determines the amount of sticking between the contact node and the contact surface at the node (leaving the body); v_{rel_i} is the relative velocity vector between the contact point and

the contact surface; $\overrightarrow{n}$ is the normal to the surface, v_{n_i} is the velocity vector in the direction of $\overrightarrow{n}$, and $\overrightarrow{v}_{t_i}$ is the tangential velocity vector.

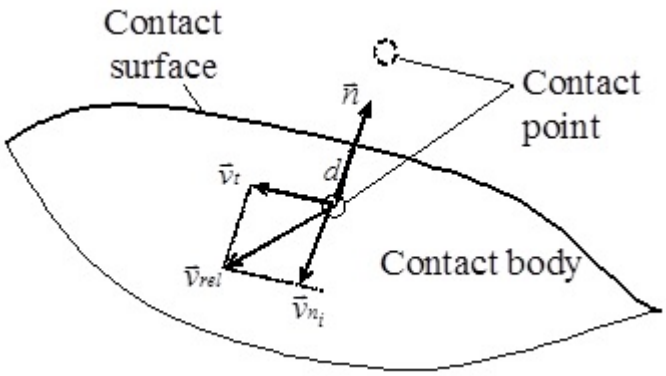

Fig. 2.2. Contact surface and contact node. [30]

2.2.2 Asperity Friction Model

An asperity-spring friction model is used to model contact and joint friction [26]. In this model friction is modeled using a piece-wise linear velocity-dependent approximate Coulomb friction element in parallel with a variable anchor point spring. The model approximates asperity friction where friction forces between two rough surfaces in contact arise due to the interaction of the surface asperities Figure 2.3. The tangential friction contact force vector transmitted to the contact body at the contact point. (F_{t_i}) is given by:

$$F_{t_i} = t_i |F_t| \tag{2.21}$$

$$t_i = v_{t_i} / |\overrightarrow{v_t}| \tag{2.22}$$

Where t_i is a unit vector along the tangential contact direction. The asperity friction model is used along with the normal force to calculate the tangential friction force ($|F_t|$) [26]. When two surfaces are in static (stick) contact, the surface asperities act like tangential springs. When a tangential force is applied, the springs elastically deform and pull the surfaces to their original position. If the tangential force is

large enough, the surface asperities yield (i.e. the springs break) allowing sliding to occur between the two surfaces. The breakaway force is proportional to the normal contact pressure. In addition, when the two surfaces are sliding past each other, the asperities provide resistance to the motion that is a function of the sliding velocity and acceleration, and the normal contact pressure. Figure 2.4 shows a schematic diagram of the asperity friction model. It is composed of a simple piece-wise linear velocity-dependent approximate Coulomb friction element in parallel with a variable anchor point spring. Note that in order to connect two points on two bodies using an asperity spring, the model must keep track of which rigid bodies are in contact and of the local position vectors of the asperity spring anchor points on the two contacting bodies. Also, note that the two rigid bodies can be in contact at more than one point, therefore the model must keep track of the corresponding contact points on the two contact bodies.

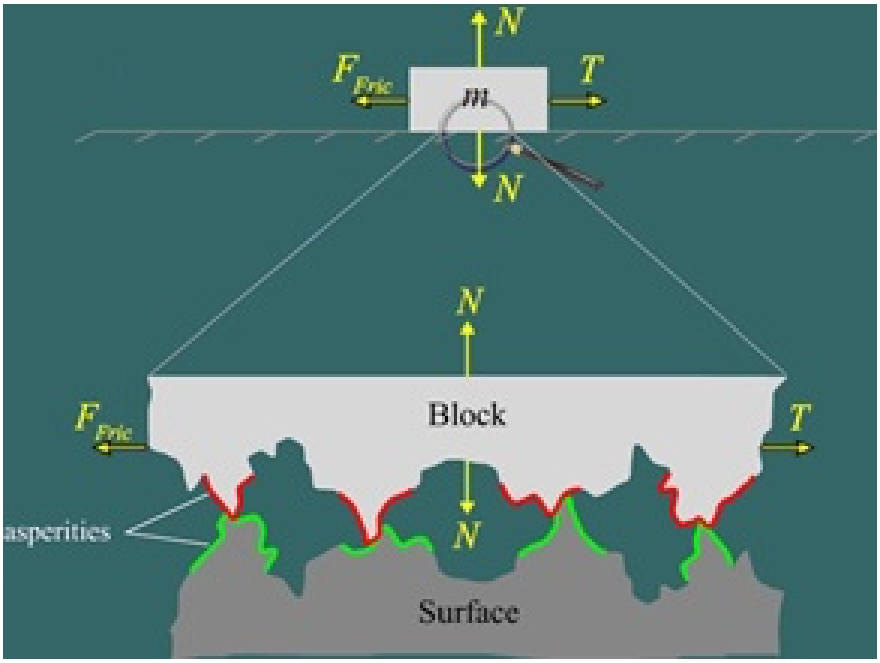

Fig. 2.3. Asperity-based physical interpretation of friction. [30]

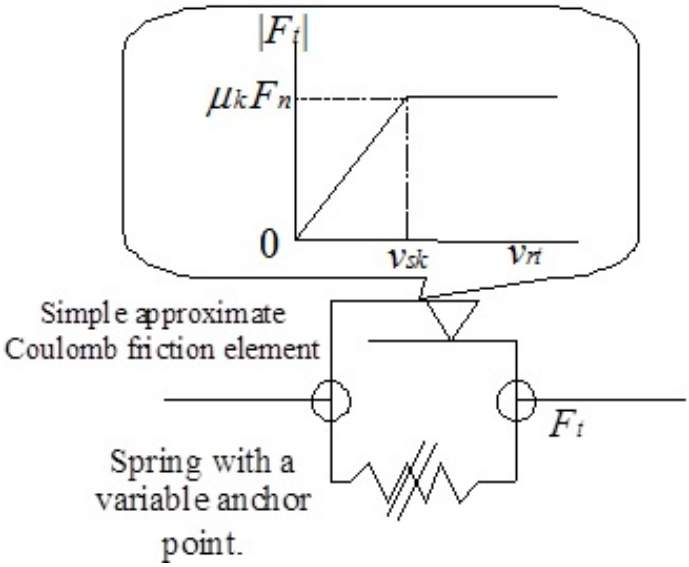

Fig. 2.4. shows the Asperity spring friction model. F_t is the tangential friction force, F_n is the normal force, μ_k is the kinetic friction coefficient, and v_{rt} is the relative tangential velocity between the two points in contact. [30]

2.2.3 Inter-Particle Contact Search

Another key component of the DEM is search and detection of inter-particle contacts. In [17] a space decomposition contact search algorithm was developed. This algorithm is implemented to search for particle contact. The steps of the algorithm are as follows:

1- The space where the particles can move is decomposed into a Cartesian Eulerian volume grid of equally sized boxes (Figure 2.5).

2- We loop over all the particles. For each particle, the grid boxes that intersect the particle are found. The minimum and maximum vertical and horizontal grid numbers of a particle can be easily found knowing the position of the center of the particle and the bounding box of the particle. In addition, each grid box has a list of particles that it intersects. Thus, in the same loop for each grid box that intersects the particle we add the particle to the grid box.

3- For each particle a list of neighboring particles that can come into contact with a particle is generated. This list consists of the particles that intersect the boxes that the particle intersects.

4- For each particle, the distance between the center of the particle and the center of a neighboring particle is calculated. If that distance is larger than the sum of bounding sphere radii of the two particles, then the particles are not in contact. Otherwise the particles may be in contact and a detailed contact point detection is carried out to find the contact point (if any).

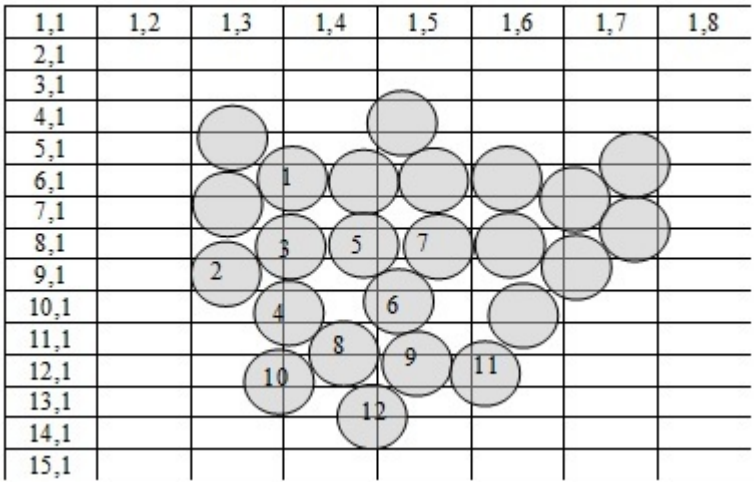

Fig. 2.5. Cartesian grid domain decomposition. [32]

5- The algorithm keeps track of the boxes that contain particles. Only those boxes are initialized to zero particles each time step. This is important since the number of Cartesian boxes is typically much larger than the number of particles. So the algorithm must not keep initializing boxes that have zero particles each time step. For example in Figure 2.5:

- Particle 6 intersects vertical grids 9 to 11 and horizontal grids 4 to 5.
- Grid (9,4) intersects particles: 3, 4, 5, 6.
- Grid (9,5) intersects particles: 5, 6. 7

- Grid (10,4) intersects particles: 4, 6
- Grid (10, 5) intersects particles: 6
- Grid (11, 4) intersects particles: 4, 6, 8, 10
- Grid (11,5) intersects particles: 6, 8, 9, 11
- Thus, particle 6 neighbors are: particles: 3, 4, 5, 7, 8, 9, 10, 11.

The main algorithm loop is over the number of particles N. Thus, algorithm is $O(N)$ instead of $O(N^2)$ for an algorithm where each particle is checked against all other particles. The additional bounding sphere search quickly eliminates most of the neighboring particles thus the particles that the most computationally expensive contact point detection is carried over are the neighboring particles that have a very high likelihood of being in contact with the particle.

2.2.4 Contact Point Search

a. Superquadric and Spherical Surfaces Contact Search Contact detection is performed between contact points on a contact surface of a rigid body, called the master contact surface, and a surface on another rigid body, called the slave contact surface. The slave contact surface can be: a superquadric surface (Figure 2.7), a collection of bounding spheres (Figure 2.8) or a polygonal surface (Figure 2.9). A superquadric surface centered at point $(0, 0, 0)$ is defined by the equation:

$$(\frac{x_1}{r_1})^{p_1} + (\frac{x_2}{r_2})^{p_2} + (\frac{x_3}{r_3})^{p_3} = 1 \tag{2.23}$$

Where x_i is the coordinate of a point on the surface, r_i is the radius in direction i, andP_i is the exponent in direction i. If the master contact point coordinates xc_i in the slave contact body frame satisfies the following condition:

$$(\frac{x_1}{r_1})^{p_1} + (\frac{x_2}{r_2})^{p_2} + (\frac{x_3}{r_3})^{p_3} \leq 1 \tag{2.24}$$

Then the master contact point is inside the contact slave body. Similarly, if the slave contact surface is a collection of spheres (Figure 2.7) then Equation (2.24) can be used to detect contact if the contact point coordinates xc_i are calculated in the local frame of each sphere on the slave contact body. In the case of a sphere, $p_1 = p_2 = p_3 = 2$, and $r_1 = r_2 = r_3 = r$. In that case Equation (2.24) becomes:

$$(x_{c_1})^2 + (x_{c_2})^2 + (x_{c_3})^2 \leq r^2 \tag{2.25}$$

Where r is the radius of the sphere. b. Polygonal Surface Contact Search If

Fig. 2.6. Particle of cubical shape modeled using superquadric with N = 3 (left) and N = 8 (right). [32], [33]

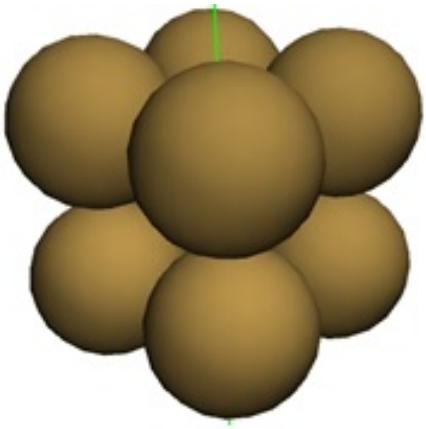

Fig. 2.7. Particle of cubical shape modeled using 8 spheres. [32], [33]

the slave contact surface is a polygonal surface (see Figures 2.8, 2.9), then contact

between a contact point of the master surface and the polygons of the slave surface is detected using a binary tree contact search algorithm. [32, 33] At the initialization of the algorithm the following steps are performed:

- Each slave polygonal contact surface is divided into 2 blocks of polygons. The bounding box for each block of polygons is found. Then each of those blocks of polygons is divided into 2 blocks and again the bounding boxes for those blocks are found. This recursive division continues until there is only one polygon in a box. itemFor each master contact surface the contact points are divided into 2 blocks. The bounding sphere for each block of points is found. Then, each of those blocks of points is divided into 2 blocks and again the bounding spheres for those blocks are found. This recursive division continues until there is only one point (with a bounding sphere of radius 0).

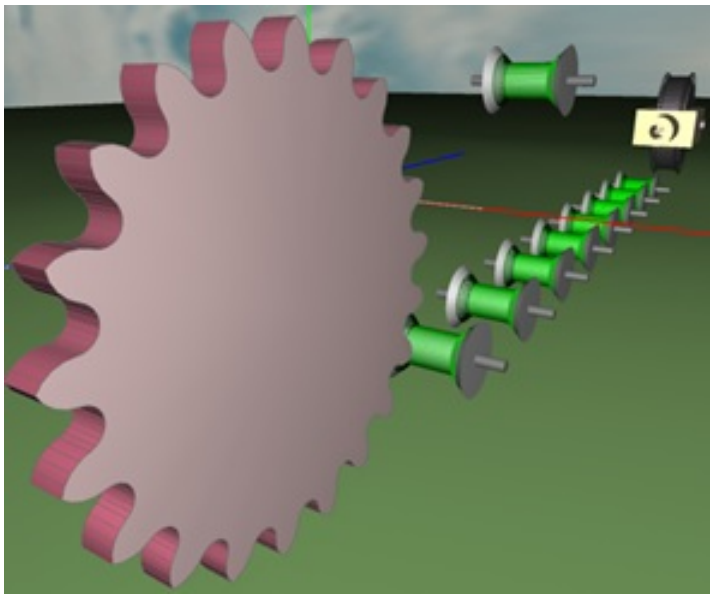

Fig. 2.8. Polygonal contact surfaces for the vehicle sprocket and wheels. [32, 33]

During the solution the following steps are performed. For each master contact sphere, the radius of the contact sphere is added to the size of the bounding box, then we check if the center point of the sphere is inside a bounding box. If the center of the

Fig. 2.9. Particle of cubical shape modeled using a polygonal surface. [32, 33]

contact sphere is not inside any bounding box, then all the points inside that sphere are not in contact with the surface. If the center of the contact sphere is inside a bounding box then the two sub-bounding boxes are checked to determine if the point is inside either one. If it is, then the sub-contact spheres are checked. If a contact point is found to be inside the lowest level bounding box, then a more computationally intensive contact algorithm between a point and a polygon is used to determine the depth of contact and the local position of the contact point on the polygon.

2.3 Joint Constraints

A joint connects two rigid bodies. A joint imposes motion constraints between points on two rigid bodies. Each rigid body can have a number of connection points. A connection point is a point on the body where joints can be located. A connection point does not add additional DOFs to the system. The position and velocity of a connection point are given by Equations (2.11) and (2.12), respectively, where x_{LP} is the position of the connection point relative to the bodys frame and x_{GP} and $\dot{x}_{GP}$ are the position and velocity of the connection point relative to the global reference frame.

2.3.1 Spherical Joints

A joint is defined by specifying the relation between connection points. A spherical joint between two connection points is defined as [30]:

$$x^t_{c1_i} = x^t_{c2_i} \tag{2.26}$$

Where $x^t_{c1_i}$ is the position of the first point on the first body and $x^t_{c2_i}$ is the position of the second point on the second body, both with respect to the global reference frame. A spherical joint leaves 3 relative rotational DOFs (degrees-of-freedom) between the two rigid bodies free and constrains 3 relative translational DOFs. This constraint is imposed using the penalty technique as:

$$F_c = K_p|v| + c_p v_i \dot{v}_i \tag{2.27}$$

$$d_i = x^t_{c1i} - x^t_{c2_i} \tag{2.28}$$

$$\dot{d}_i = \dot{x}^t_{c1i} - \dot{x}^t_{c2_i} \tag{2.29}$$

$$d = \sqrt{d_1^2 + d_2^2 + d_2^2} \tag{2.30}$$

$$F_{C_i} = F_c d_i / d \tag{2.31}$$

Where F_{c_i} is the penalty reaction force on the connection point, k_p is the penalty spring stiffness, cp is the penalty damping, d_i is the relative displacement vector between points $c1$ and $c2$, and $\dot{d}_i$ is the relative velocity vector between points $c1$ and $c2$. The constraint force is applied on the two connection points in opposite directions.

The constraint force is transferred to the node at the center of the body as a force and a moment using Equations (2.14) and (2.15).The force acting on rigid body 2 is equal and opposite to the force acting on rigid body 1. The constraint forces given by Equation (2.14) are assembled into the global structural forces F_s in Equation (2.1). The constraint torques given by Equation (2.15) are assembled into the global structural torques T_s in Equation (2.2).

2.3.2 Revolute Joints

A revolute joint constrains three translational DOFs and two rotational DOFs between two rigid bodies, thus leaving only one rotational DOF free between the bodies. Revolute joints can be modeled by placing two spherical joints along a line. Therefore, the same mathematical methods which are used for the spherical joints can be used to define revolute joints.

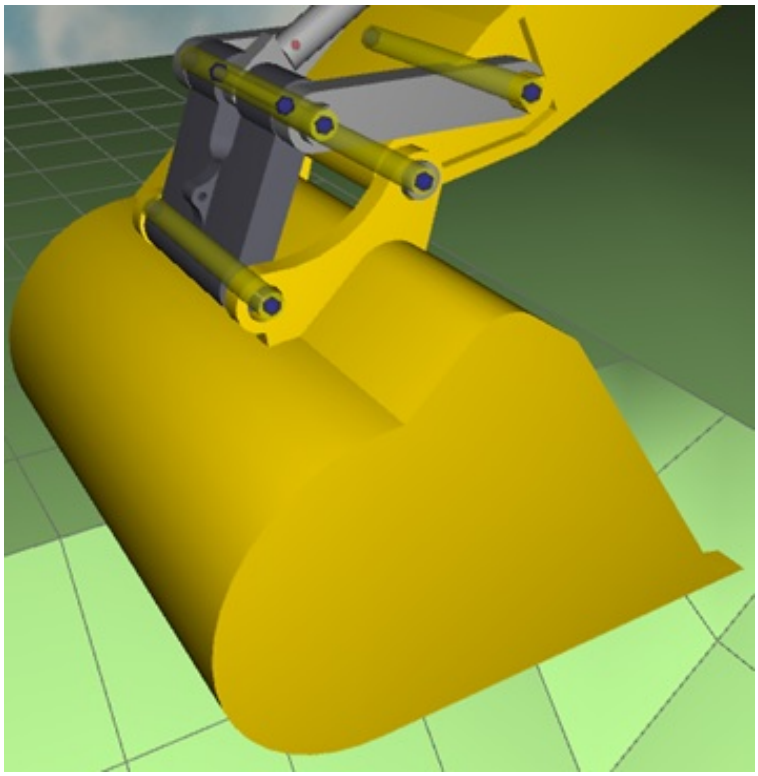

Fig. 2.10. Backhoe bucket model, revolute joints shown as yellow cylinders with each revolute joint modeled using two spherical joints shown as a blue spheres.

2.3.3 Cylindrical Point Joints

A Cylindrical point joints constrains two translational DOFs between two rigid bodies and leaves one translation DOF and three rotational DOFs free. A cylindrical

joint can be modeled by constraining a point on a rigid body to move along a line on another rigid body Figure 2.11 [30]. Similar to the other joint models the penalty technique is used to define this constraint between the two rigid bodies. The equation

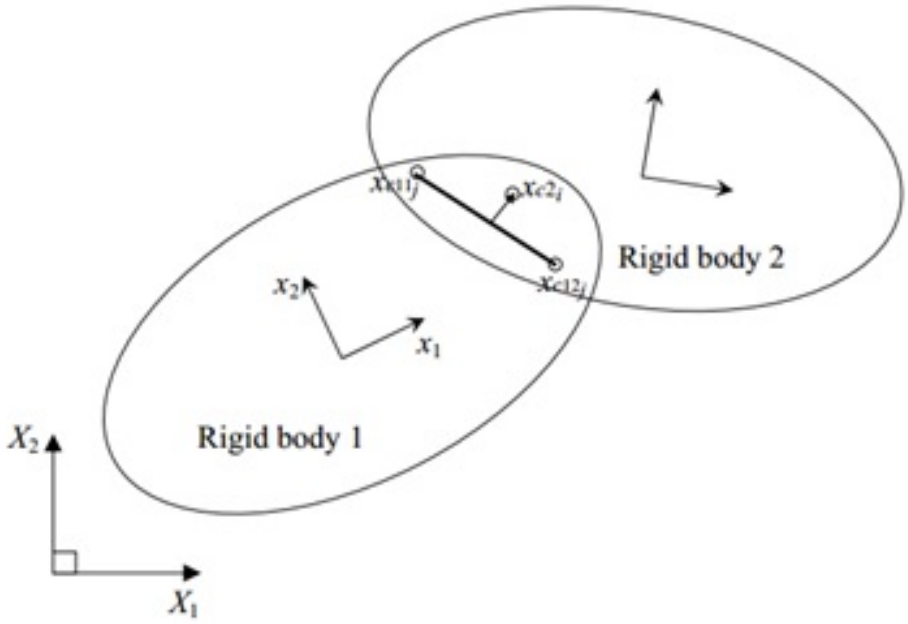

Fig. 2.11. Cylindrical point joint connecting one point on rigid body 1 to a line on rigid body 2. [30]

of a line on the first rigid body is given by:

$$X^t_{c1_i} = (1-s)X^t_{c11_i} + sX^t_{c12_i} \tag{2.32}$$

where s is a parameter from 0 to 1.

A cylindrical point joint connects a point on that line to a fixed point on the second rigid body:

$$X^t_{c1_i} = X^t_{c2_i} \tag{2.33}$$

$$(1-s)X^t_{c11_i} + sX^t_{c12_i} = X^t_{c2_i} \tag{2.34}$$

$$((1-s)X^t_{c11_i} + sX^t_{c12_i}) - X^t_{c2_i} = 0 \tag{2.35}$$

This constraint is imposed using the penalty technique as:

$$F_p = k_p d + c_p d_i \dot{d}_i \tag{2.36}$$

$$d_i = ((1-s)X^t_{c11_i} + sX^t_{c12_i}) - X^t_{c2_i} \tag{2.37}$$

$$\dot{d}_i = ((1-s)\dot{X}^t_{c11_i} + s\dot{X}^t_{c12_i}) - \dot{X}^t_{c2_i} \tag{2.38}$$

$$d = \sqrt{d_1^2 + d_2^2 + d_3^2} \tag{2.39}$$

$$F_{P_i} = \frac{F_p d_i}{d} \tag{2.40}$$

$$d_i\dot{d}_i = d_1\dot{d}_1 + d_2\dot{d}_2 + d_3\dot{d}_3 \tag{2.41}$$

where:

F_p: penalty force magnitude.

k_p: penalty spring stiffness.

c_p: penalty damping.

d_i: relative displacement vector between points $c1$ and $c2$.

$\dot{d}_i$: relative velocity vector between points $c1$ and $c2$.

$\dot{X}^t_{c1_i}$: global velocity vector for point $c1$.

$\dot{X}^t_{c2_i}$: global velocity vector for point $c2$.

F_{P_i}: penalty reaction force on connection point $c1$.

$-F_{P_i}$: penalty reaction force on connection point $c2$.

s is chosen such that:

$$d_i v_i = 0 \text{ , (vector } \overrightarrow{d} \text{ is normal to vector } \overrightarrow{v}) \tag{2.42}$$

where $v_i = X^t_{c12_i} - X^t_{c11_i}$ is the vector along the cylindrical joint line.

$(((1-s)X^t_{c11_i} + sX^t_{c12_i}) - X^t_{c2_i})v_i = 0$

$((1-s)X^t_{c11_i}v_1 + sX^t_{c12_i}v_1) - X^t_{c2_i}v_1 + ((1-s)X^t_{c11_i}v_2 + sX^t_{c12_i}v_2) - X^t_{c2_i}v_2 + ((1-s)X^t_{c11_i}v_3 + sX^t_{c12_i}v_3) - X^t_{c2_i}v_3 = 0$

$s(X^t_{c12_1}v_1 + X^t_{c12_2}v_2 + X^t_{c12_3}v_3 - X^t_{c11_1}v_1 - X^t_{c11_2}v_2 - X^t_{c11_3}v_3) + X^t_{c11_1}v_1 + X^t_{c11_2}v_2 + X^t_{c11_3}v_3 - X^t_{c2_1}v_1 - X^t_{c2_2}v_2 - X^t_{c2_3}v_3 = 0$

$$s = -\frac{X^t_{c11_1}v_1 + X^t_{c11_2}v_2 + X^t_{c11_3}v_3 - X^t_{c2_1}v_1 - X^t_{c2_2}v_2 - X^t_{c2_3}v_3}{X^t_{c12_1}v_1 + X^t_{c12_2}v_2 + X^t_{c12_3}v_3 - X^t_{c11_1}v_1 - X^t_{c11_2}v_2 - X^t_{c11_3}v_3} \tag{2.43}$$

if $s < 0$ then $s = 0$ and if $s < 1$ then $s = 1$

2.3.4 Cylindrical Joints

A Cylindrical joints constrains two translational and two rotational DOFs between two rigid bodies and leaves one translational DOF and one rotational DOFs free Figures 2.12 and 2.13. A cylindrical joint between two rigid bodies can be modeled using a cylindrical point joint by placing two points on the second body rather than one point [30]. Thus, a cylindrical joint connects a line on the first rigid body to two points (or a line) on the second rigid body. The two points are constrained to move on the line.

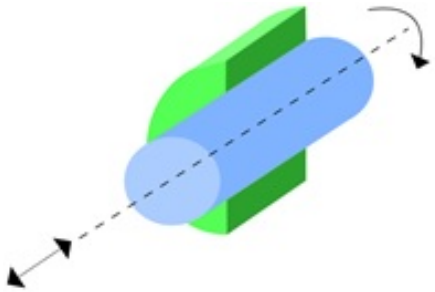

Fig. 2.12. Cylindrical joint.

2.3.5 Prismatic Joints

A prismatic joint constrains two translational and three rotational DOFs between two rigid bodies and leaves only one translational DOF free Figures 2.14 and 2.15. Prismatic joints can be modeled by placing two cylindrical joints in parallel [30].

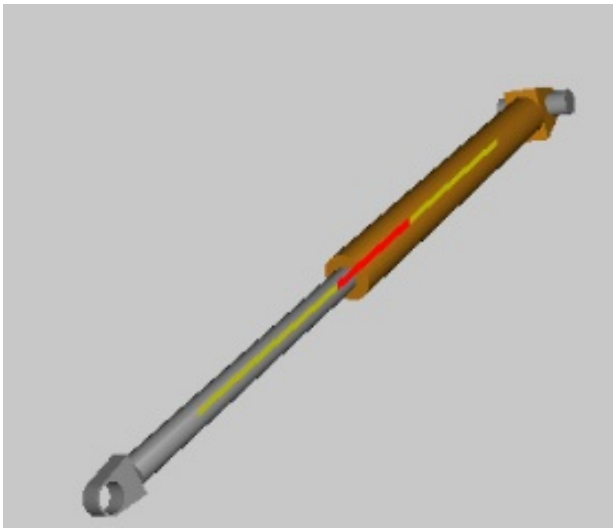

Fig. 2.13. Backhoe hydraulic cylinder model with a cylindrical joint model shown as a yellow/red line.

Fig. 2.14. Prismatic joint in 2D.

2.4 Actuators

2.4.1 Linear Actuator

A linear actuator connects two points on two rigid bodies (Figure 2.16). Using a PD controller, the force F is generated by the actuator is given by:

$$F = K(l - l_{des}) + c(\frac{l_i \dot{l}_i}{l} - \dot{l}_{des}) \tag{2.44}$$

$$l_i = x^t_{c1_i} - x^t_{c2_i} \tag{2.45}$$

$$\dot{l}_i = \dot{x}^t_{c1_i} - \dot{x}^t_{c2_i} \tag{2.46}$$

$$l = \sqrt{l_1^2 + l_2^2 + l_3^2} \tag{2.47}$$

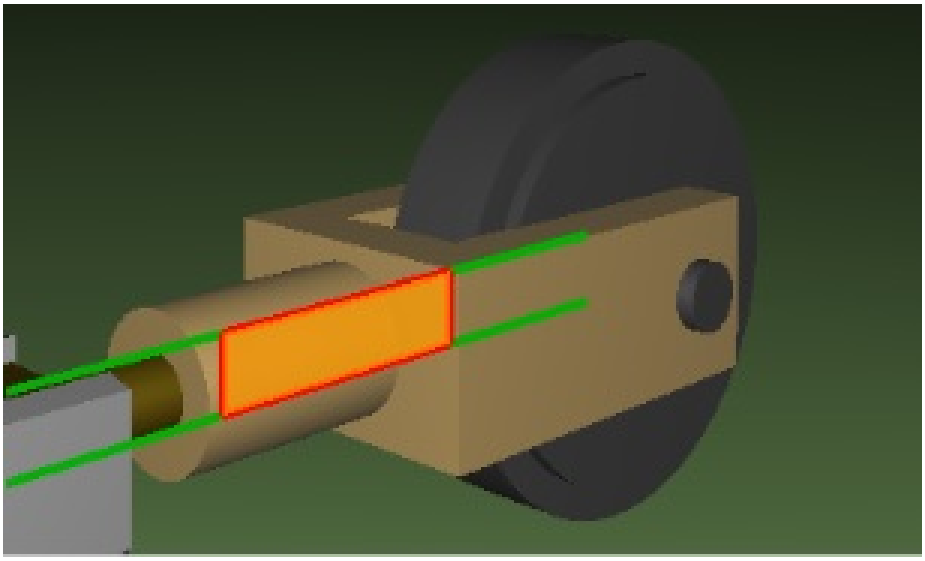

Fig. 2.15. Backhoe road wheel model, prismatic joint shown as an orange box with parallel green lines.

$$F_i = Fl_i/l \tag{2.48}$$

Where $x^t_{c1_i}$ is the position of the first point on the first body and $x^t_{c2_i}$ is the position of the second point on the second body, both with respect to the global reference frame; K is the controller proportional gain, c is the controller derivative gain, l_{des} is the desired length of the actuator,l is the current length of the actuator, and is F the actuator force vector. [30], [32] The actuator force is transferred to the rigid bodies center as force and moment using Equations (2.14) and (2.15).

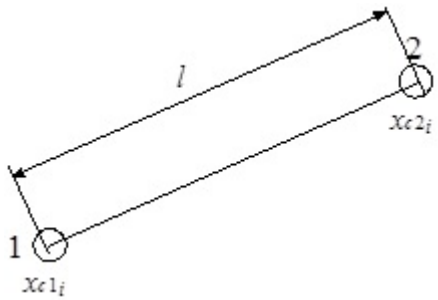

Fig. 2.16. Linear actuator connecting two points. [30], [32]

2.4.2 Rotational Actuator

A rotational actuator connects three points on two rigid bodies (Figure 2.17). Two of the points are on one rigid body and the remaining point is on the second rigid body. Using a PD controller the torque T generated by the actuator is given by:

$$T = k(\theta - \theta_{des}) + c(\dot{\theta} - \dot{\theta_{des}}) \tag{2.49}$$

Where θ is the current angle of the actuator, θdes is the desired angle, k is the proportional gain and c is the derivative gain. The rotary spring forces are transferred to the rigid bodies center as force and moment using Equations (2.14),(2.15).

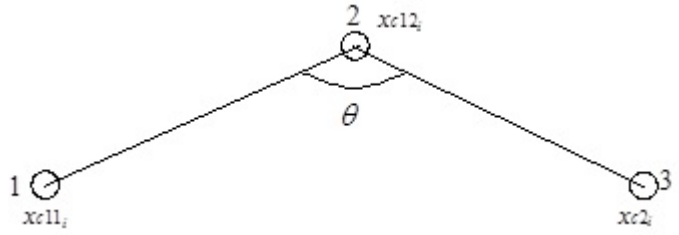

Fig. 2.17. Rotational actuator connecting three points. [30], [32]

2.5 Explicit Solution Procedure

The solution fields for modeling multibody systems are defined at the model nodes. Note that a rigid body is modeled as one finite element node. These solutions fields are:

- Translational positions.
- Translational velocities.
- Translational accelerations.
- Rotation matrices.
- Rotational velocities.

- Rotational accelerations.

The explicit time integration solution procedure predicts the time evolution of the above response quantities. An advantage of explicit solution procedures is that they are embarrassingly parallel. The procedure described below achieves near linear speed-up with the number of processors on shared memory parallel computers. The procedure is implemented in the DIS [30] (Dynamic Interactions Simulator) commercial software code and is outlined below:

1- Prepare the run:

a. Set the initial conditions for the solution fields identified above.

b. Create a list of all the finite elements that includes the master contact surfaces.

c. Create a list of elements that will run on each processor. This is done using an algorithm which tries to make the computational cost on each processor equal.

d. Create a list of all the prescribed motion constraints.

e. Calculate the solid masses for each finite element node by looping through the list of finite elements. Note that the masses are fixed in time.

f. Loop over all the elements and find the minimum time step for the explicit solution procedure.

2- Loop over the solution time and increment the time by Δt each step while doing the following:

a. Set the nodal values at the last time step to be equal to the current nodal values for all solution fields.

b. Do 2 iterations (a predictor iteration and a corrector iteration) of the following:

i. Initialize the nodal forces and moments to zero.

ii. Perform the particle Contact search algorithm (Section 2.2.3)

iii. Calculate the nodal forces and moments by looping

through all the elements while calculating and assembling the element nodal forces.This is the most computational intensive step. This step is done in parallel by running each list of elements identified in step 1.c on one processor.

iv. Find the nodal values at the current time step using the semi-discrete equations of motion and the trapezoidal time integration rule (Equations (2.1)to(2.7)).

v. Execute the prescribed motion constraints which set the nodal value(s) to prescribed values.

vi. Go to the beginning of step 2. [32]

3. DISCRETE ELEMENT METHOD (DEM) MODELING AND SIMULATION RESULTS

3.1 Sand Model

Several test simulations were performed in order to find the best sand model. For the first model, each sand particle was modeled as a sphere with appropriate normal contact penalty force model and a tangential friction force model (see Section 2.2). The main advantage of using spherical particles is their low computational cost. However, spherical particles can easily roll and thus resulting in a very small angle of repose for a particle pile. This was fixed in previous studies by adding a rolling coefficient of friction. However, this coefficient of friction is not physical and thus is hard to tune and may cause inaccurate estimation of the pile angle of repose and/or the particle flow rate through an orifice.

In [15] it was found that two-dimensional rectangular and oval particles flow as much as 30% slower than circular particles. Since sand particles have a shape that is closer to cubical then to spherical as shown in Figure 3.1, it was decided to use cubical-shaped particles as a better approximation for modeling sand particles.

In order to model cubical sand particles Figure 2.9, each particle was modeled using 8 spheres (see Figure 2.7). An appropriate contact force model is used for each sphere inside the cubical sand particles to predict translational and rotational movement of each cubical particle with respect to the global inertia frame.

Using cubical sand particles provides the interlocking mechanism which reduces rolling of particles on each other and decreases the sands flow rate through the aperture of the conical hopper. However, using cubical particles increases the computational cost compared to spherical particles.

Fig. 3.1. Close up ($1 \times 1cm$) of sand from the Gobi Desert Mongolia [31]

3.2 Sand Hopper Model and Simulation Results

Predicting the mass flow rate of granular materials from storage containers through orifices is important in many industries (such as construction, agricultural, food and pharmaceutical industries) and thus was studied in many papers [19] to [23]. Three regimes of flow are detected during the discharge through an orifice: a continuous flow, an intermittent flow, and a complete blockage flow. Beverloo et al. [19] developed an empirical equation that gives the particle mass flow rate from a hopper through an orifice given the orifice diameter and the particle size (for particles larger than 500 μm) [19]. Using the Beverloo law in a continuous flow regime, the mass flow rate W is predicted to scale as $D^{5/2}$, where D represents the outlet diameter [19, 22].

Predicting the angle of repose of a particle pile is also an important factor in many industries. Several factors like the internal friction angle, sliding and rolling friction

factors, shape, size, and density of particles as well as humidity of the environment, methods used to pour the particles, and the height of the pouring material affect the angle of repose [7, 8, 24].

A DEM/multibody dynamic model of sand particles flow from a conical hopper was used to predict the flow rate of sand particles during the discharge process and the angle of repose (friction angle) of the stable sand pile as a function of the sand particles size and the orifice diameter of the hopper. In order to validate the sand model, the results of the numerical model are compared to experimental results previously published in Beverloo et al. [19].

Figure 3.2 shows the initial conditions of the sand particles and the conical hopper. Also, the parameters for the simulations of the conical hopper sand flow experiment are presented in Table 3.1. The conical hopper and the ground are modeled as polygonal surfaces. The conical hopper top diameter radius is $200mm$ and height is $200mm$ (Figure 3.2). The orifice diameter (bottom diameter of the conical hopper) is set to: $1cm$, $1.5cm$, $2cm$ and $2.5cm$ in order to study the effect of changing the orifice diameter on the particle flow rate.

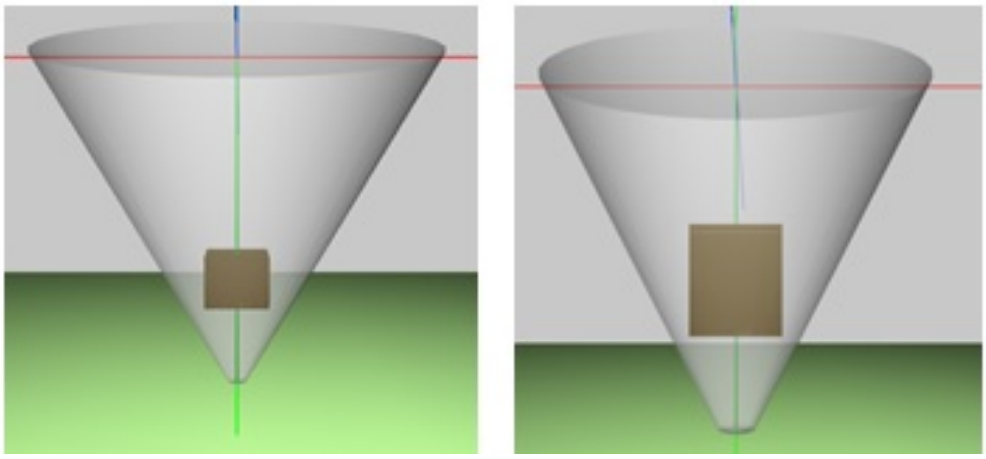

Fig. 3.2. Conical hopper and initial conditions of the sand particles for particle size $s = 1mm$ (left) and $s = 2mm$ (right). [33]

Table 3.1
List of the initial conditions of the conical hopper sand flow experiment simulations.

	$s = 1mm$	$s = 2mm$
Mass of each particle (g)	0.0017	0.0136
Number of the particles	25200	12348
Eulerian search grid size (cm)	$16 \times 13 \times 16$	$16 \times 18 \times 16$
Resolution	$160 \times 130 \times 160$	$80 \times 90 \times 80$
Total simulation time (sec)	5	15
Computer run time ($hours/sec$ of simulation time)	1.9 (Six 4.0 GHz Intel i7-3930K processor cores)	0.94 (Six 4.0 GHz Intel i7-3930K processor cores)
Simulation time step (sec)	$1.1e-005$	$1.1e-005$
Normal contact stiffness (N/m)	1500	15000
Normal contact damping ($N.sec/m$)	0.00025	0.002
Separation damping factor	0.05	0.05
Coefficient of friction	0.5	0.5

The particle shape chosen is shown in Figure 2.7. Each particle consists of 8 spheres, arranged as shown in the figure, in order to model the cubical shape of a sand particle. Note that the particles are drawn as cubes in the figures in this section, but the actual particle contact geometry is shown in Figure 2.7. Two particle sizes are used in the model $s = 1mm$ and $s = 2mm$, where s is the side length of the cube. A coefficient of friction of 0.5 is used between: sand particles, the hopper walls and the sand particles, and the ground and the sand particles.

The sand true density is taken to be $\rho_t = 2.62g/cm^3$. In order to calculate the actual mass of a cubical sand particle, the volume of each particle was multiplied by the sand true density. Since each cubical particle contains 8 spheres, the volume of each sphere is calculated and multiplied by 8 to find the total volume of the spheres inside the cube. In order to compute the total volume of the cube, volume A which is approximately equal to the total void volume around the spheres (volume that no particle can occupy due to the geometry of the contact surface) was added to the spheres total volume. s is the side length of the cube, $r = s/4$ is the radius of each

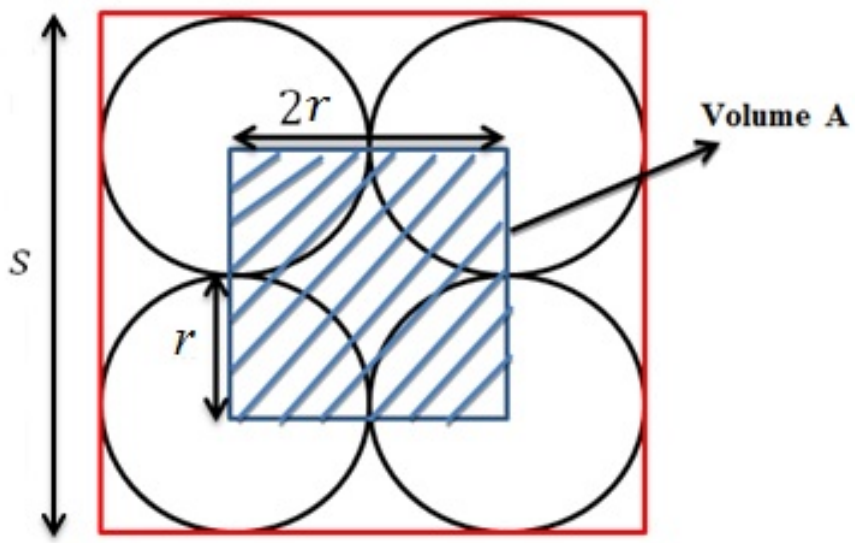

Fig. 3.3. Two dimensional front view of a cubical sand particle.

sphere inside the cube. The total volume of the sand particle is given by:

$$\text{Total Volume} = 8 \times ((4/3) \times \Pi \times r^3) + (2r)^3 \tag{3.1}$$

The mass of a sand particle is given by:

$$\text{Mass of each particle} = \text{Total Volume} \times \rho_t \tag{3.2}$$

Therefore, for $s = 1mm$, $r = 1/4mm$:

$$\text{Total Volume} = 8 \times ((4/3) \times \Pi \times 0.025^3) + (0.05)^3 = 0.000648599cm^3 \tag{3.3}$$

$$\text{Mass of each particle} = 0.000648599 \times 2.62(gr/cm^3) = 0.0017gr \tag{3.4}$$

And for $s = 2mm$, $r = 1/2mm$:

$$\text{Total Volume} = 8 \times ((4/3) \times \Pi \times 0.05^3) + (0.1)^3 = 0.000518879cm^3 \tag{3.5}$$

$$\text{Mass of each particle} = 0.000518879 \times 2.62(gr/cm^3) = 0.0136gr \tag{3.6}$$

For the $1mm$ particle model a rectangular array of $28 \times 30 \times 30 = 25200$ particles is used (Figure 3.2). For the $2mm$ particle model a rectangular array of $28 \times 21 \times 21 = 12348$ particles is used. The initial configuration of the particles is shown in Figure 3.2. For the $1mm$ particle model an Eulerian search grid of size $16 \times 13 \times 16cm$ and resolution $160 \times 130 \times 160$ is used (Figure 3.4). The side length of a grid cell was chosen to be equal to the diameter of the particle. Similarly, for the $2mm$ particle model an Eulerian search grid of size $16 \times 18 \times 16cm$ and resolution $80 \times 90 \times 80$ is used. The simulation starts by letting the rectangular array of particles fall in the conical hopper while the orifice opening is closed in the first 1 second. This allows the sand particles to fill the conical hopper and come to rest (Figure 3.5). At this point the bulk density of the sand is measured by dividing the total volume of the particles inside the conical hopper by the total mass of the particles. The average bulk density for the $1mm$ particles was 1.362 g/cm^3 and the average bulk density for the $2mm$ particles was 1.386 g/cm^3. Figures 3.6 and 3.7 show the bulk density as a function of the orifice diameter for the $1mm$ and $2mm$ particles, respectively. The

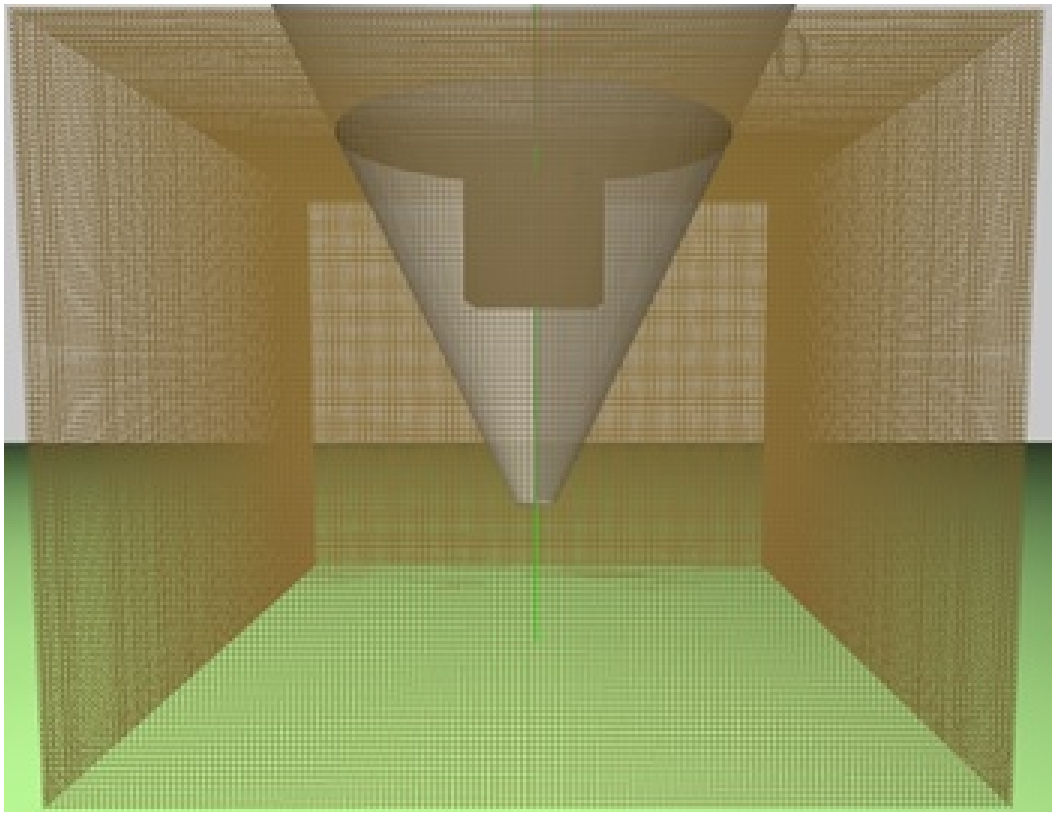

Fig. 3.4. Cartesian Eulerian search grid for particle size $s = 1mm$ (Eulerian grid box size $16 \times 13 \times 16cm$ and resolution $160 \times 130 \times 160$). [33]

reason for the small variations of the bulk density is mainly due to measuring error since it is not possible to measure precisely the volume of the sand particles in the conical hopper. Also, the particles interlock slightly differently in each simulation. The theoretical bulk density for sand is approximately 1.5 g/cm^3 [19]. The main reason for the difference between the numerical model and the experimental bulk densities is most likely due to the fact that equal sized particles are used in the numerical model. If particles of varying sizes are used then small particles will fill more voids between larger particles thus leading to a higher bulk density.

After 1 sec from the start of the simulation, the hopper orifice is opened and the sand is allowed to flow from the hopper and fall on the ground forming a sand pile (Figure 3.8). A sensor is placed around the hopper to count the total number

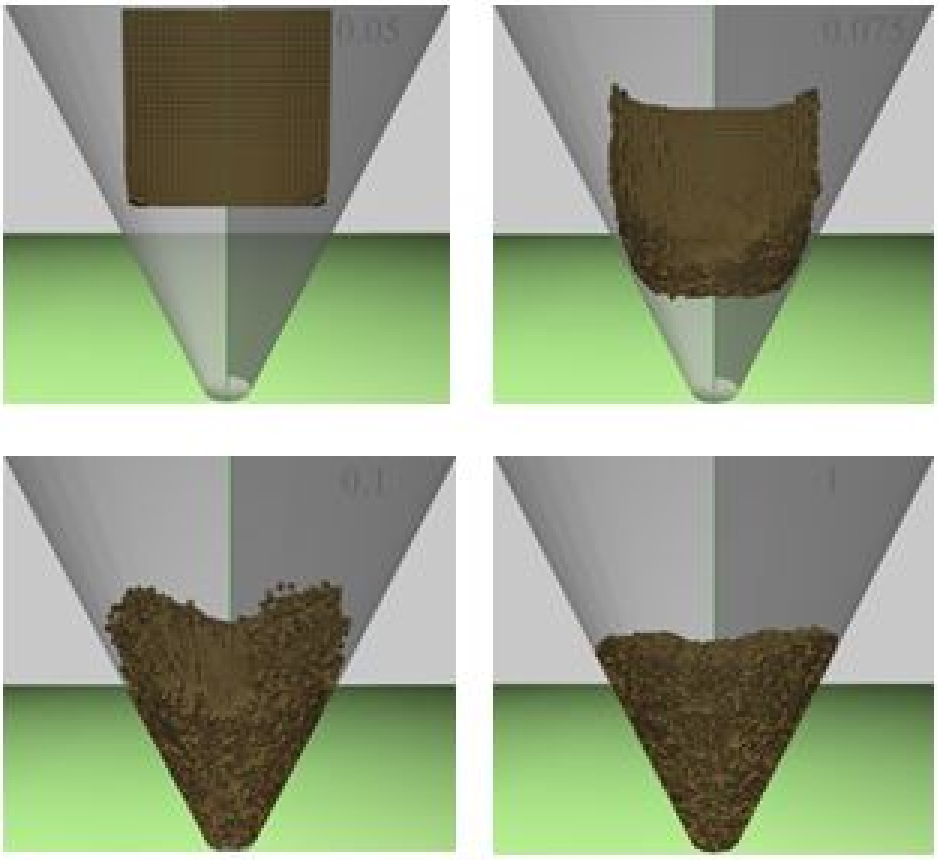

Fig. 3.5. Snapshots of the conical hopper for particle size $s = 1mm$ from time 0.05 to time 1 sec. [33]

of particles inside the hopper. The flow rate is calculated by plotting the number of particles inside the hopper as a function of time. Figures 3.9 and 3.10 show the number of particles inside the hopper as function of time for various hopper orifice diameters, and $1mm$ and $2mm$ particle sizes, respectively. For the $1mm$ particles for orifice diameters 1, 1.5, 2, and 2.5 cm, the flow rate is nearly constant for most of the discharge process, except at the start and end. For the $2mm$ particles the flow rate is nearly constant for orifice diameters 1.5, 2, and 2.5 cm (except for the start and end of the discharge process) and is intermittent for the $1cm$ orifice diameter (Figure 3.10) [33].

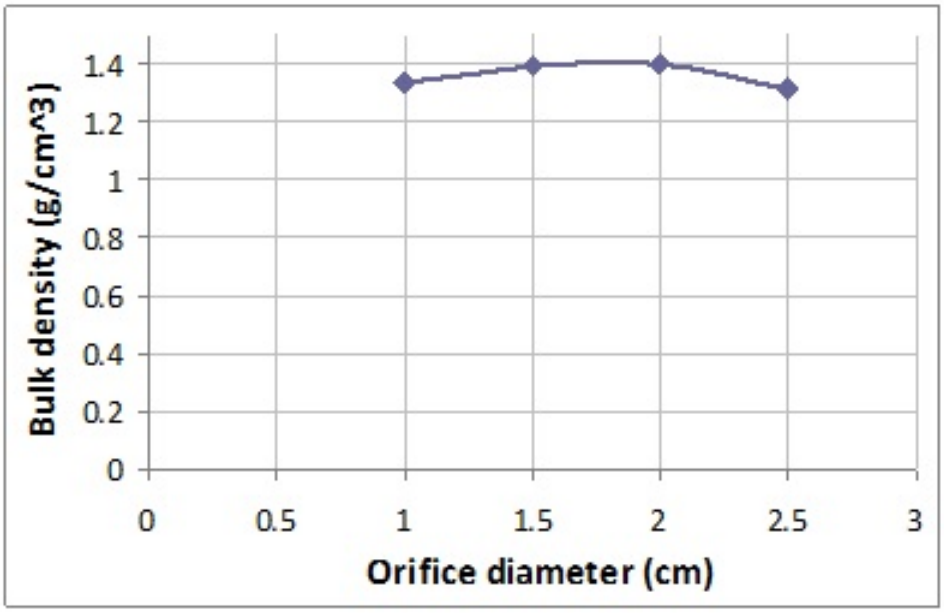

Fig. 3.6. Bulk density as function of the hopper orifice diameter for particle size $s = 1mm$. [33]

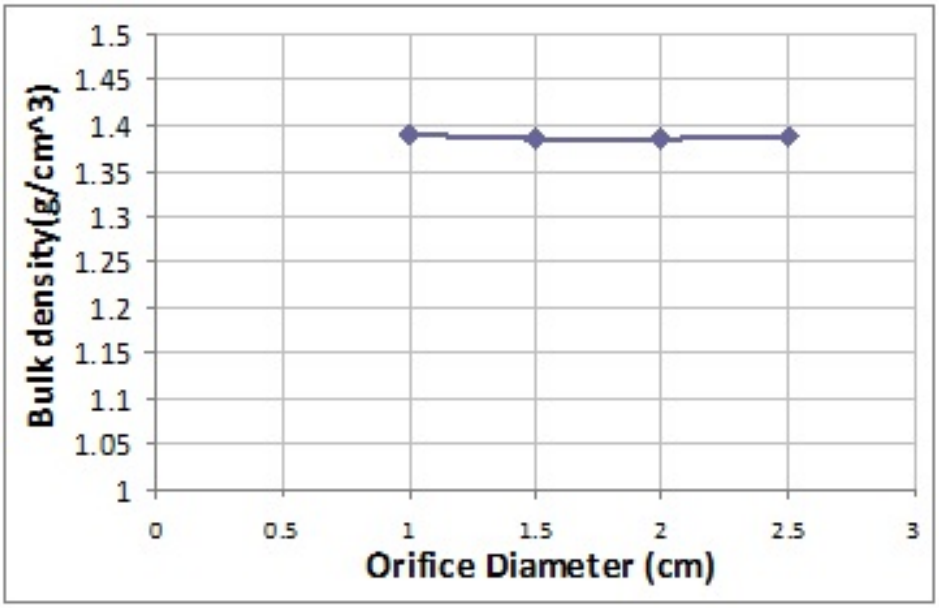

Fig. 3.7. Bulk density for particle size $s = 2mm$. [33]

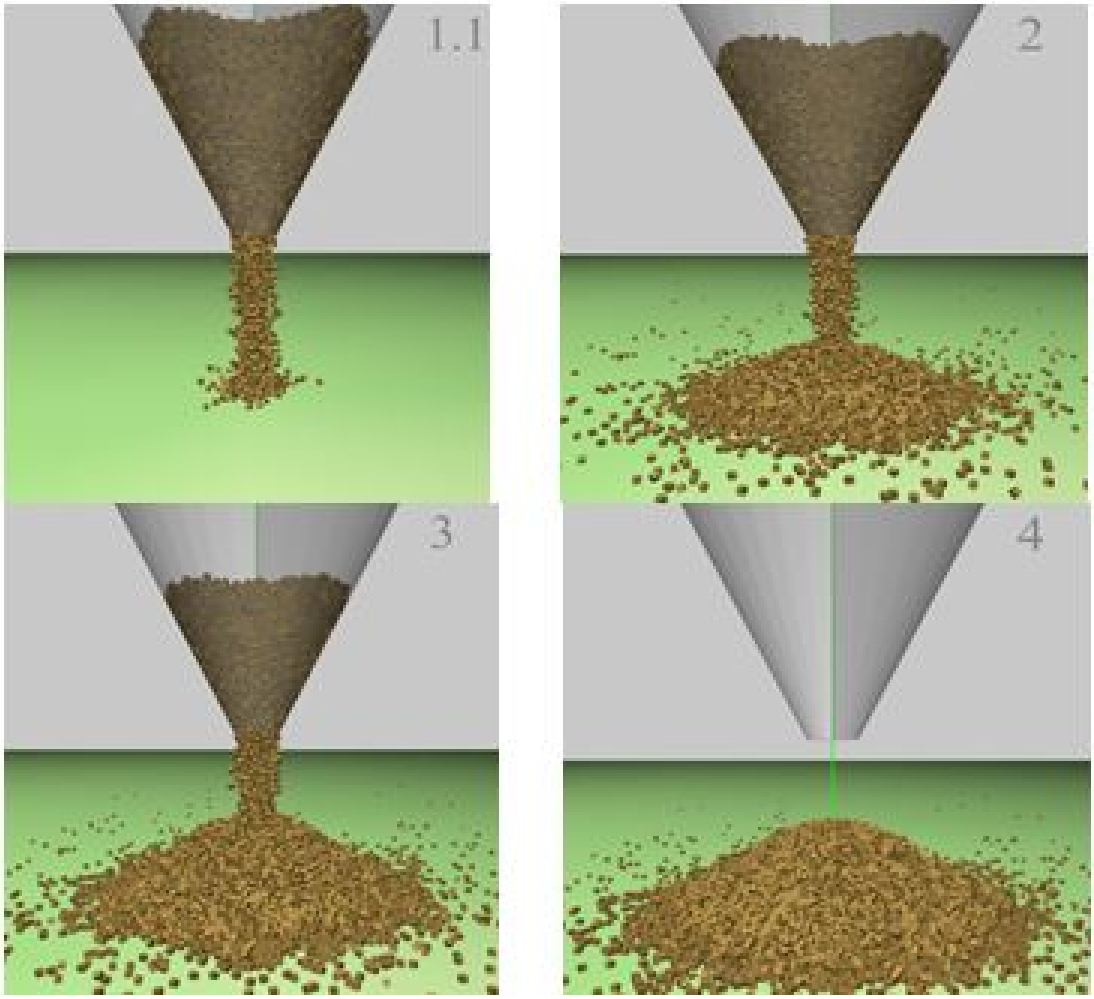

Fig. 3.8. Snapshots of the conical hopper for particle size $s = 1mm$ from time 1.1 to time 4 sec. [33]

In Berverloo et al. [19], an empirical equation (based on physical experiments), called the Beverloo equation, was proposed for calculating the flow rate of granular materials. The Beverloo equation is:

$$W = C\rho_B\sqrt{g}(D_0 - kd)^{2.5} \tag{3.7}$$

Where W is the flow rate through the hoppers orifice in g/min, C is a constant that is approximately equal to 38.8 for sand, ρ_B is the material bulk density in g/mm^3 and g is the acceleration of gravity in mm/s^2, D_0 is the orifice diameter in cm, d is

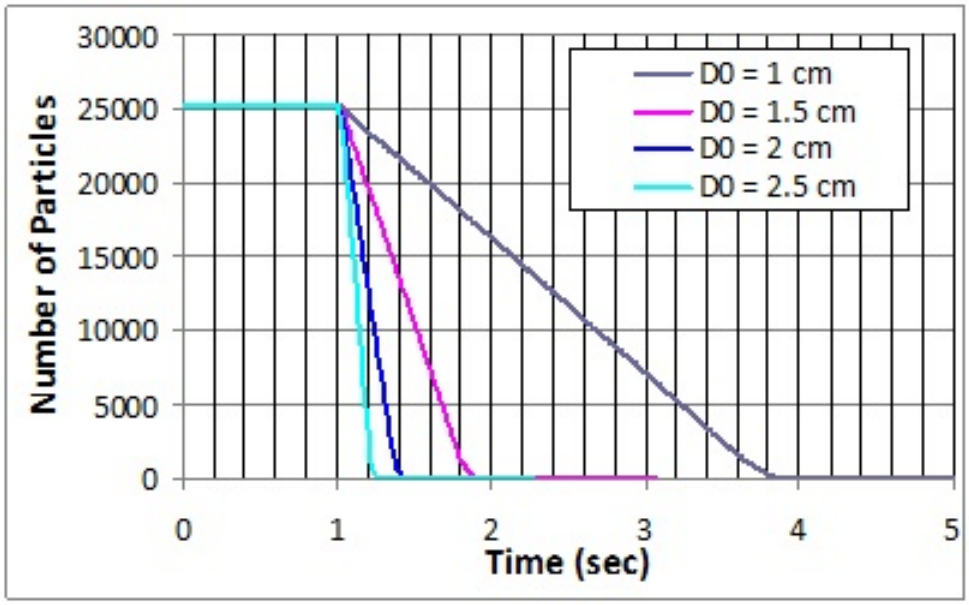

Fig. 3.9. Number of particles remaining in the hopper as a function of time as for various orifice diameters and particle size $s = 1mm$. [33]

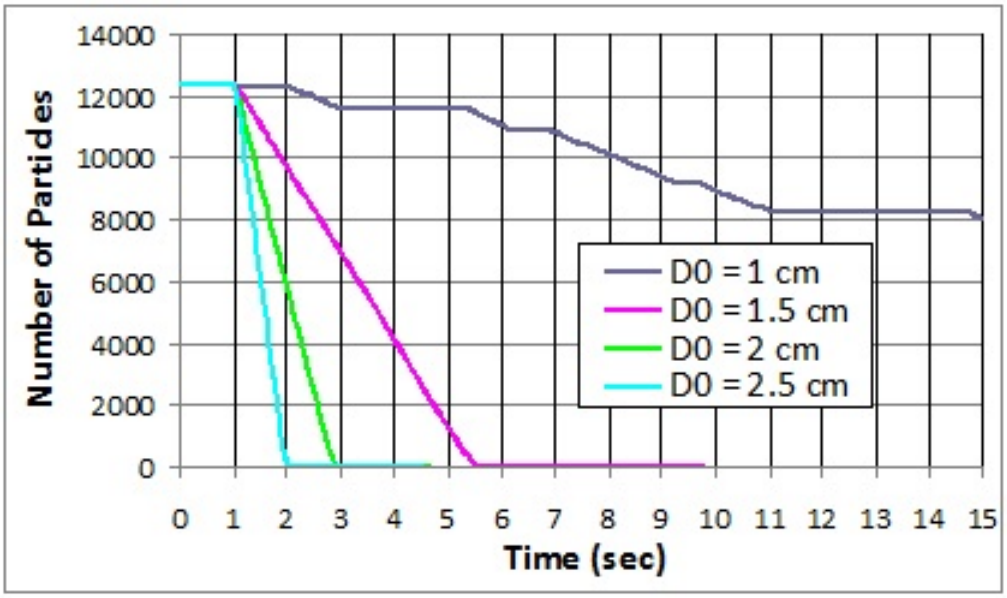

Fig. 3.10. Number of particles remaining in the hopper as a function of time as for various orifice diameters and particle size $s = 2mm$. [33]

the particle diameter in cm, k is a constant which depends on the shape and the size of the particles (k= 2.9 for sand) [19] to [22]. Based on the particle geometry given in Figure 2.7, we can calculate an average cubical size (d) of the particles by using the volume of the particle given by Equation (3.1):

$$d = V^{1/3} \tag{3.8}$$

For $s = 1$ mm particles, using the volume calculated in Equation (3.3):

$$d = 0.000648599^{1/3} = 0.865mm.$$

For $s = 2$ mm particles, using the volume calculated in Equation (3.5):

$$d = 0.000518879^{1/3} = 1.73mm.$$

In Figures 3.11 and 3.12 the sand flow rate as a function of the orifice diameter predicted using the numerical model is compared to the experimental flow rate given by Equation (3.7) for sand particle sizes $s = 1mm$ and $s = 2mm$, respectively. For particle size $s = 1mm$, the maximum difference between the model and the experimental results is less than 5% For particle size $s = 2mm$, the numerical model predicts on average 10% smaller flow rate than Equation (3.7).

After all the particles have fallen to the ground and formed a sand pile (see Figure 3.8), the angle of repose of the sand pile can be calculated. The angle of repose is calculated by placing a virtual cone around the sand pile (Figure 3.13). The cones radius R is set approximately at the start of the sand piles slope and the height h of the cone is set so that the average slope of the sand pile is equal to the slope of the cone. The angle of repose α is calculated using:

$$\alpha = tan^{-1}(\frac{h}{R}) \tag{3.9}$$

Figures 3.14 and 3.15 show the sand pile angle of repose as a function of the orifice diameter for particle sizes $s = 1mm$ and $s = 2mm$, respectively. If the sand pile is formed by a slow discharge rate ($D_0 = 1cm$), then the angle of repose is between 33 to

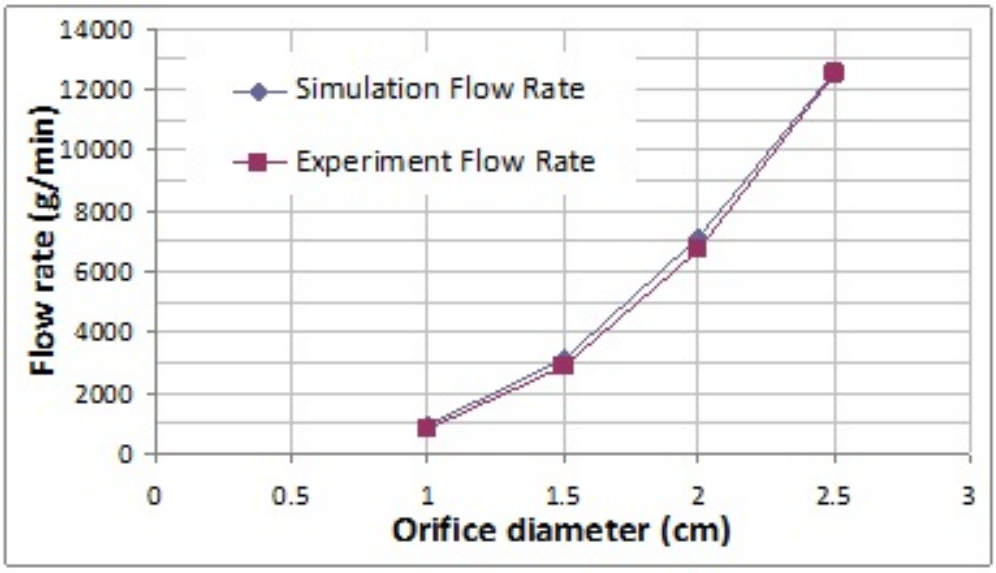

Fig. 3.11. Comparison between experiment and simulation flow rates for particle size $s = 1mm$. [33]

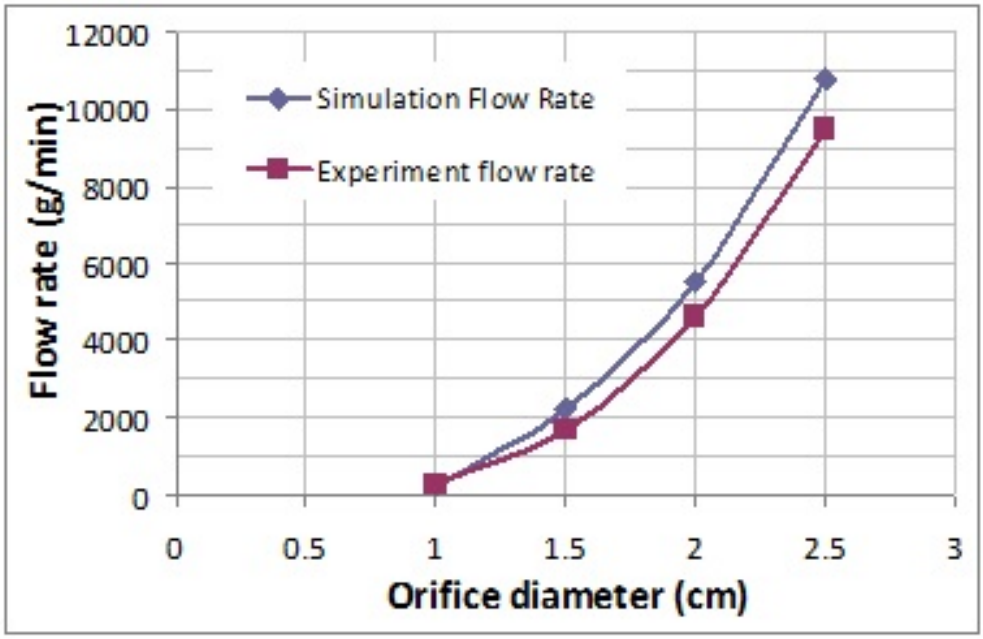

Fig. 3.12. Comparison between experiment and simulation flow rates for particle size $s = 2mm$. [33]

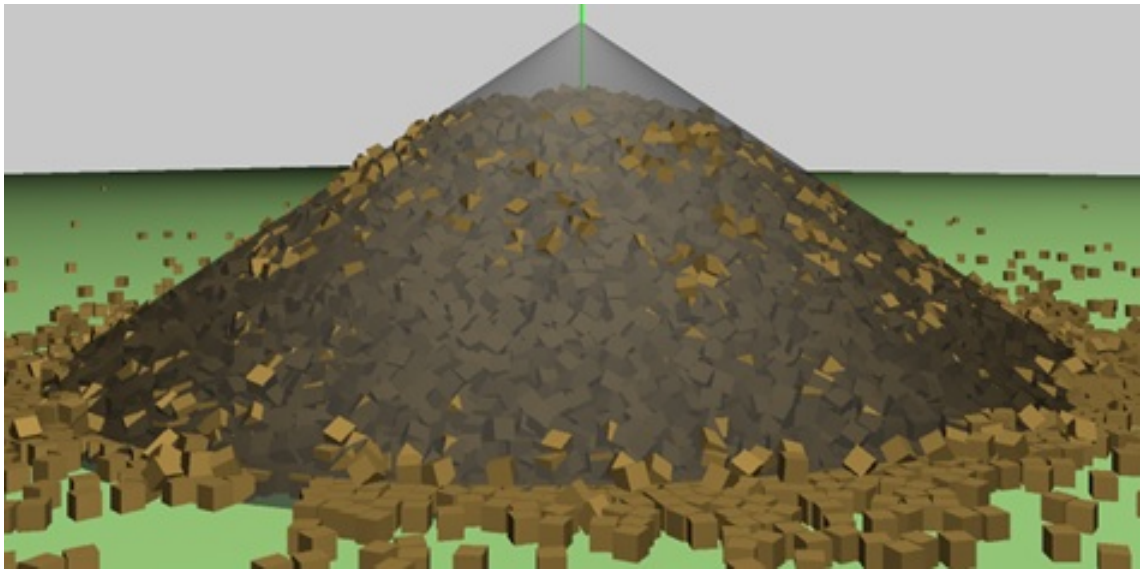

Fig. 3.13. Virtual cone to measure the angle of repose for particle size $s = 1mm$ and orifice diameter $D_0 = 1cm$ [33]

35 degrees. For medium discharge rates the angle of repose is about 26 to 30 degrees. For very fast discharge rates, the angle of repose is between 20 and 25 degrees. This was also observed in previously published experimental results [24] to [26].

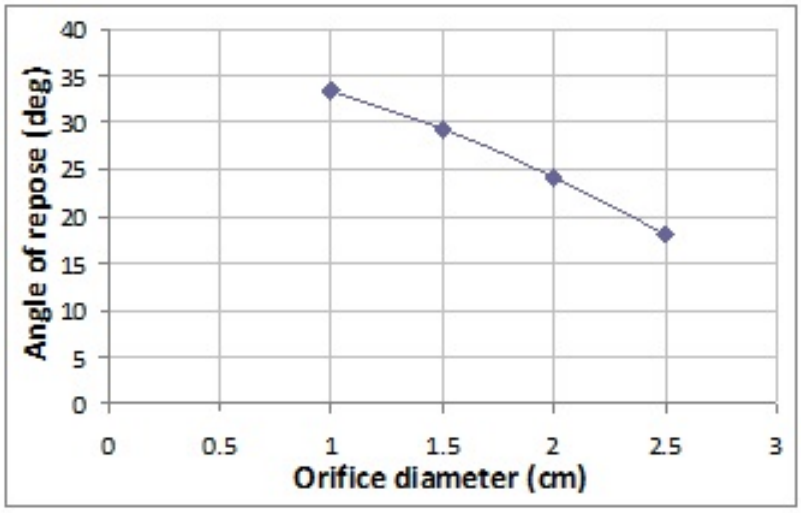

Fig. 3.14. Angle of repose for particle size $s = 1mm$ as a function of the orifice diameter [33]

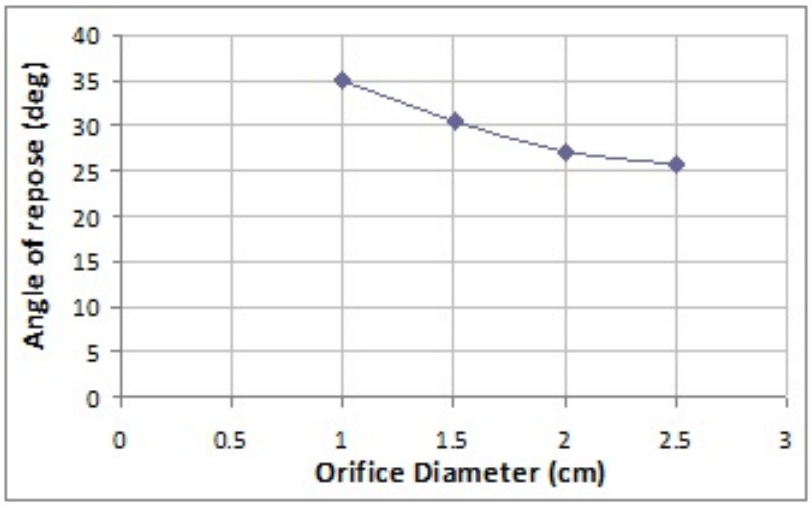

Fig. 3.15. Angle of repose for particle size $s = 2mm$ as a function of the orifice diameter [33]

4. BACKHOE MODEL AND SIMULATION RESULTS

There are five main steps for creating the multibody dynamics model of the backhoe:

1. Creating a CAD model of the system.
2. Connecting the components of the backhoe using various joints.
3. Placing linear and rotary actuators on the various components.
4. Adding control laws and specifying the desired actuator motions.
5. Running the simulation.

The DIS [29] explicit-time integration finite element code was used to create the multibody dynamics model of the backhoe and generate the simulation results of a backhoe digging through a sand pile. Using Pro Engineer, the geometry of major moving components was created and modeled as rigid bodies. Those include: undercarriage, road wheels, idler wheels, drive sprocket, track segments, cabin, boom, stick, bucket, hydraulic cylinders, and hydraulic pistons. Figures 4.1 through 4.9 show the three dimensional CAD models of the backhoes components along with their major dimensions. Also, the momentum inertia and mass of those are presented in Table 4.1. The parameters of the backhoe simulation are presented in Table 4.2. Figures 4.10 shows the full multibody dynamics model of the backhoe.

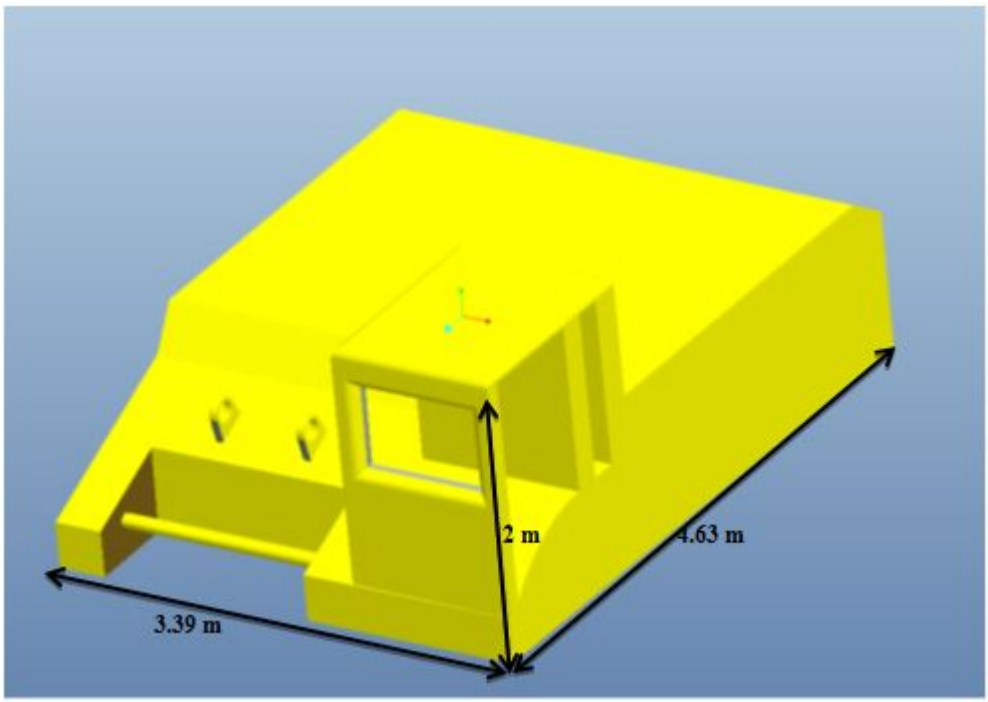

Fig. 4.1. Cabin CAD model

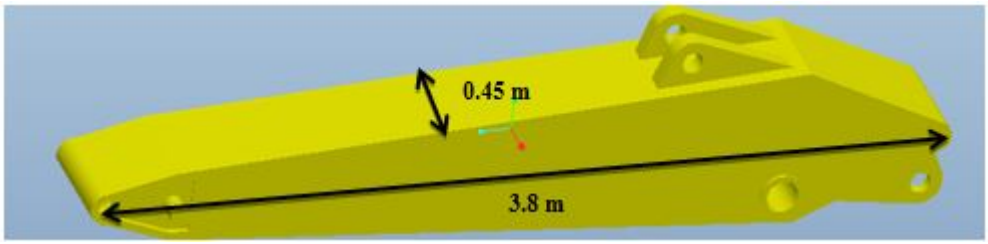

Fig. 4.2. Stick CAD model

Figure 4.11 shows the backhoe track model with connected track segments. Those unit segments are connected using revolute joints to form the track. Figure 4.12 shows the track system components. Each track system consists of 100 segments, a driving sprocket, a front wheel, 8 road wheels and 2 idler wheels. Each drive sprocket includes a rotary actuator and a controller to control the steering and speed of the backhoe. The idler wheels and road wheels are each modeled as a rigid pulley. The road wheels distribute the weight of the vehicle along the bottom part of the track, while the

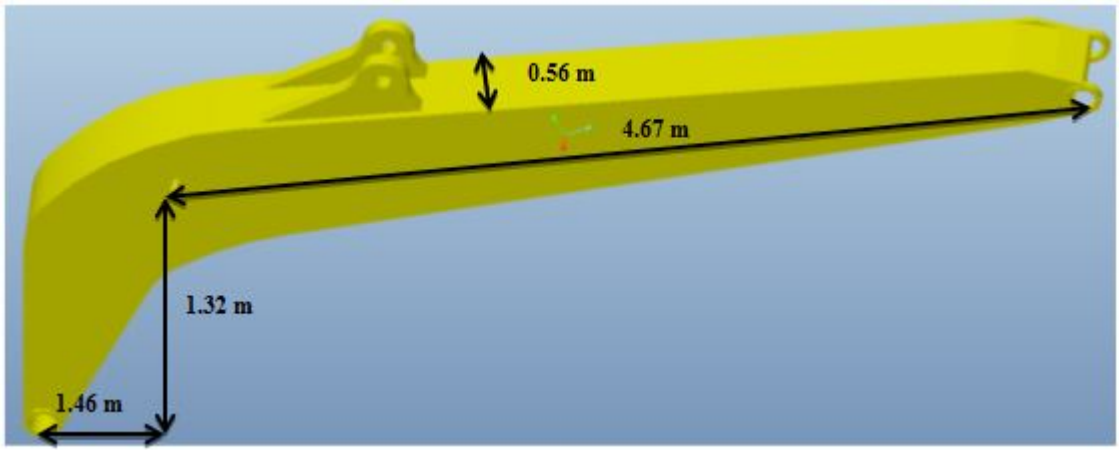

Fig. 4.3. Boom CAD model

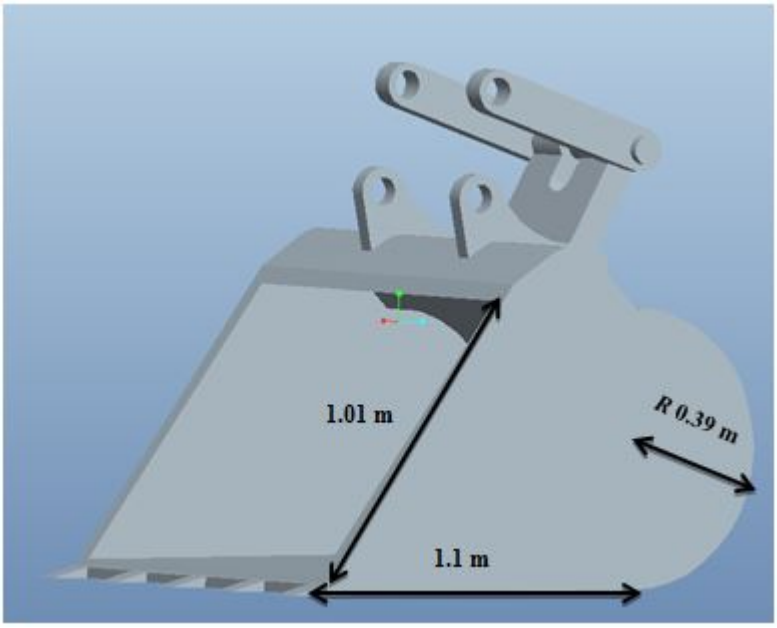

Fig. 4.4. Bucket CAD model

idler wheels keep the top part of the track from sagging. The cabin is connected to the undercarriage by a revolute joint and a rotary actuator is used to steer the cabin to the desired angle with respect to the undercarriage. The total mass of the undercarriage including the tracks is 11,759.3 Kg. The total mass of the cabin and manipulator is 9,780 Kg.

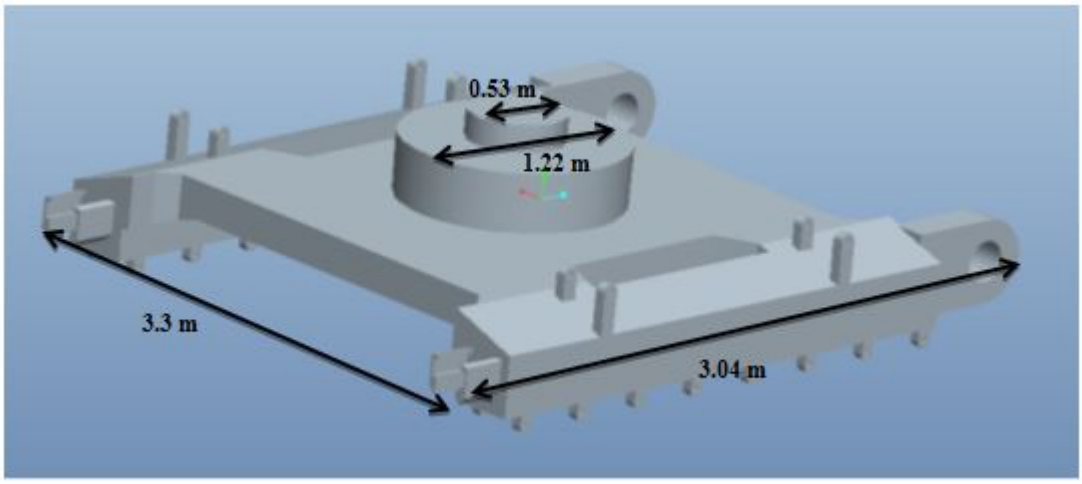

Fig. 4.5. Undercarriage CAD model

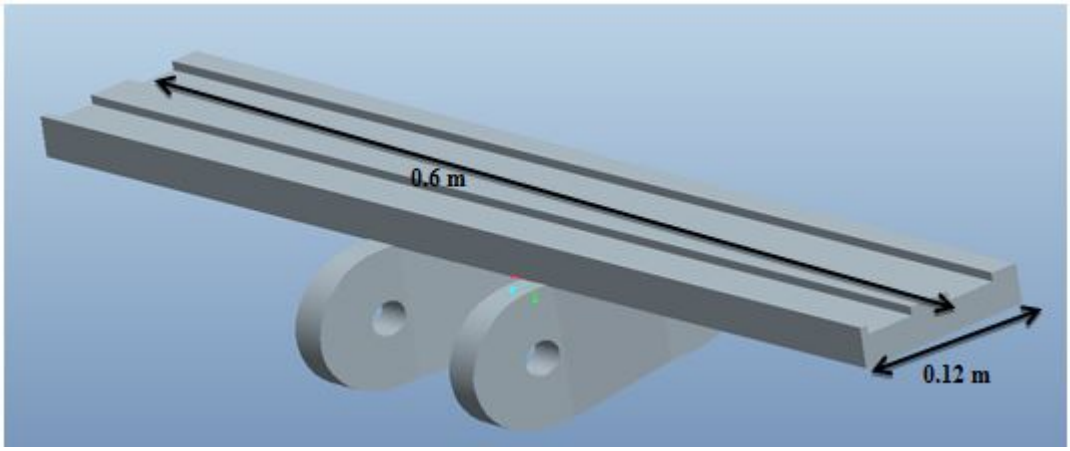

Fig. 4.6. Unit track segment CAD model

Contact surfaces are defined between the various components. Those include: sprocket - track segments; wheels - track segments; sprocket track segments; track segments ground; bucket - soil particles; and soil particles - soil particles. Figure 4.10 shows the joints (14 revolute joints and 4 cylindrical joints) and the 4 hydraulic actuators (left and right boom actuators, stick actuator and bucket actuator) for the backhoe manipulator. Each of these hydraulic actuators consists of a piston and a cylinder connected using a cylindrical joint. Bucket links 1 and 2 are used to connect

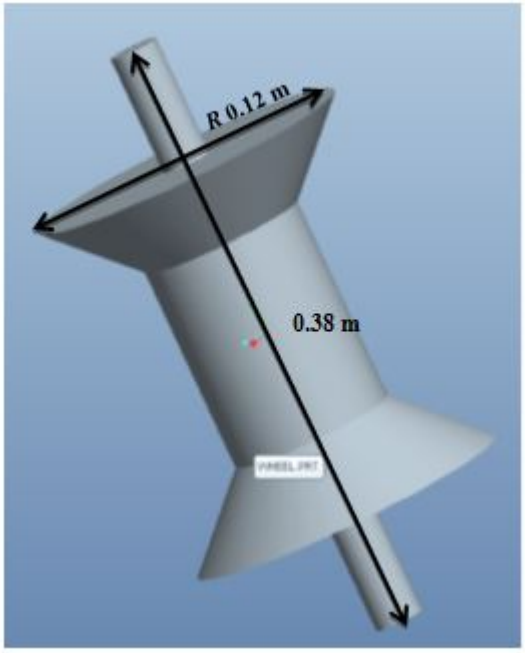

Fig. 4.7. Idler wheels CAD model

Table 4.1
List of the components Weight and Moment of Inertia.

name of the Components	Mass(kg)	Intertia(I_{xx}, I_{yy}, I_{zz})
Undercarriage	6000	4979,9463,6188
Cabin	8000	4979,9463,6188
Boom	600	2000,180,2000
Bucket	150	25,25,25
Stick	400	1400,1400,100
Sprocket	287.82	12.051,12.051,16.263
Idler Wheel	23.4	0.19422,0.19422,0.0585
Road Wheel	287.82	12.051,12.051,16.263

the bucket to the stick. In order to keep the bucket in the desired trajectory and

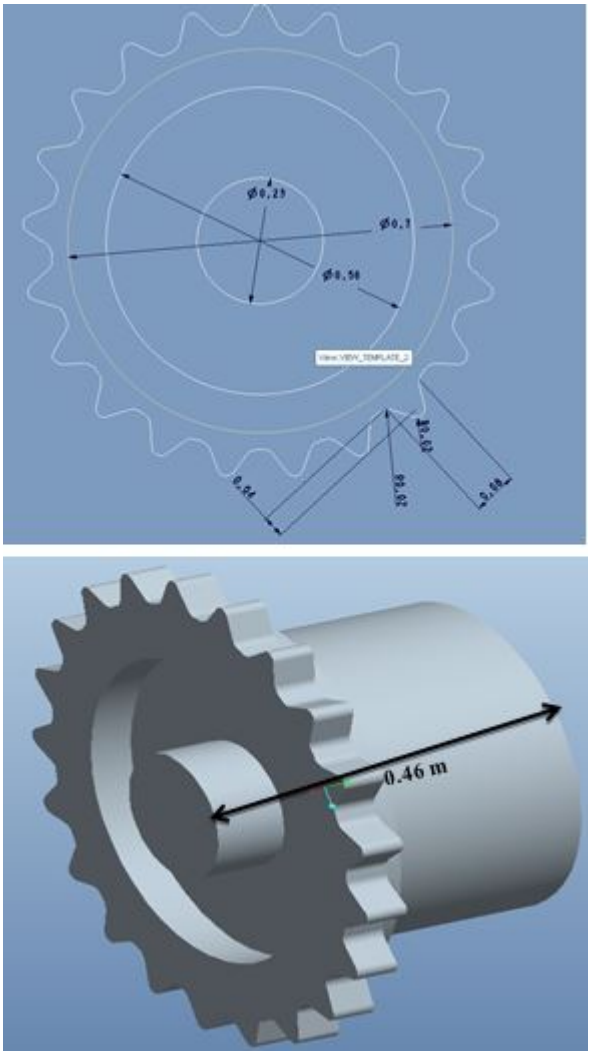

Fig. 4.8. Sprocket CAD model

control the force exerted by the actuators, PD controllers are used for all the actuators in the model with proportional and derivative gain factors listed in Table 4.3.

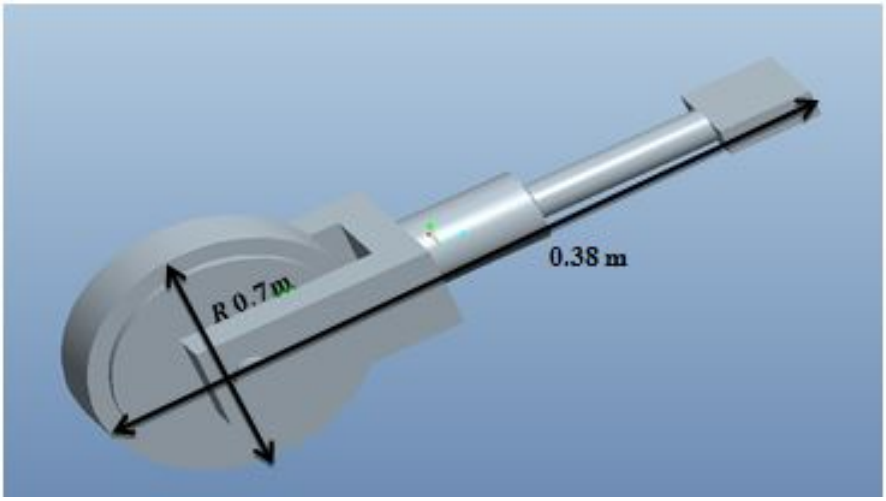

Fig. 4.9. Road wheel CAD model

Table 4.2
List of the initial conditions of the backhoe model simulation.

Number of rigid bodies in the backhoe model	241
Number of sand particles	27000
Eulerian particle box resolution	$90 \times 45 \times 90$
Eulerian particle boxsize(m)	$10 \times 5 \times 10$
Total simulation time (sec)	16
Simulation time step (sec)	$1.1e - 0.005$
Computer time (hours/sec)	2.05 (Six 4.0 GHz Intel i7-3930K cores)
Normal contact stiffness (N/m)	$5e + 0.006$
Normal contact damping ($N.sec/m$)	200

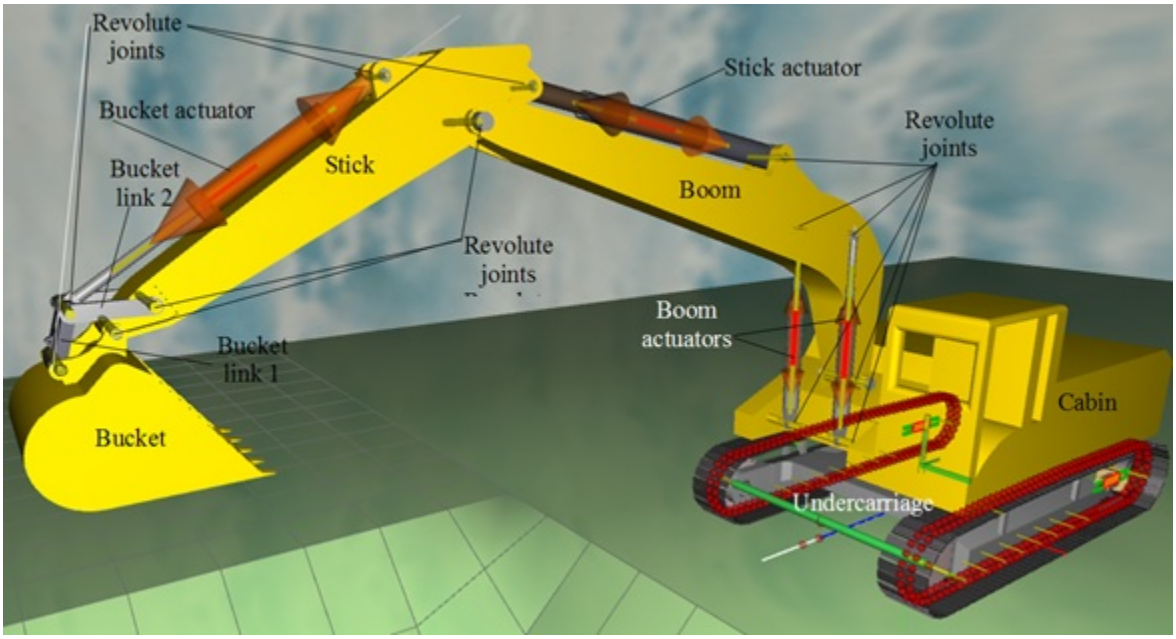

Fig. 4.10. Multibody dynamics mode of the backhoe [32].

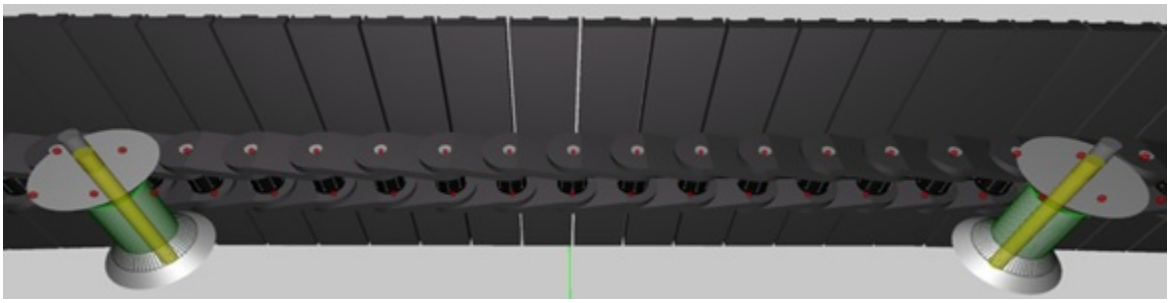

Fig. 4.11. Connected track segments [32].

Figure 4.13 shows the initial conditions of the backhoe and sand particles. The ground and bucket surface are modeled as polygonal surfaces. The particle shape chosen is shown in Figure 2.7. Each particle consists of 8 spheres, arranged as shown in the figure, in order to model the cubical shape of a sand particle. Note that the particles are drawn as cubes in the figures in this section, but the actual particle

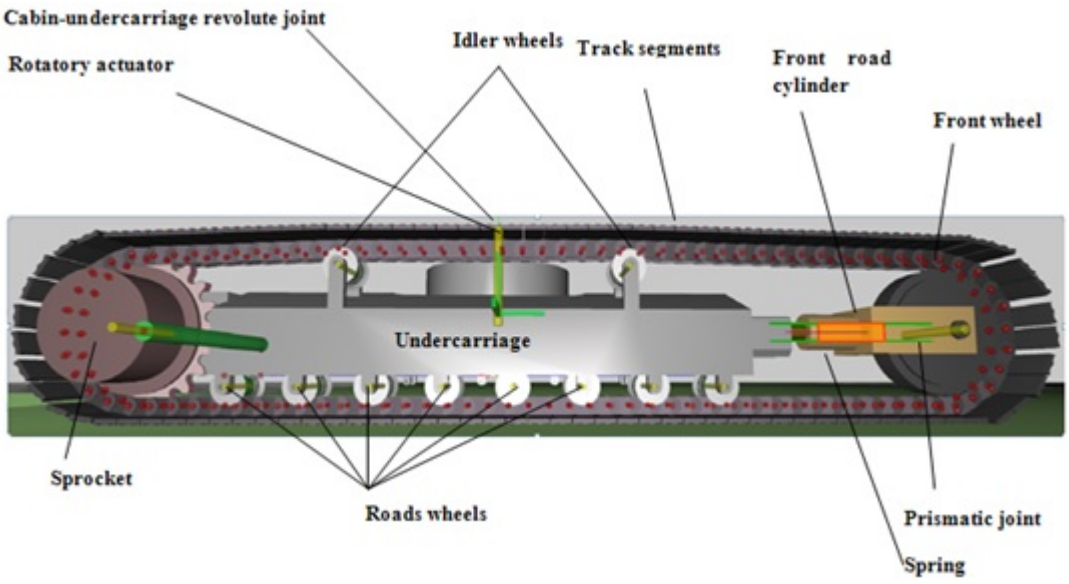

Fig. 4.12. Backhoe track model [32].

Table 4.3
Proportional and derivative gains for the various actuators [32].

	Proportional gain	Derivative gain
Cabin-Undercarriage rotary actuators	$1e^8$	$8e^4$
Left & right boom linear actuators	$1e^7$	$3e^5$
Stick linear actuator	$2e^7$	$1e^6$
Bucket linear actuator	$2e^6$	$2e^5$

contact geometry is shown in Figure 2.7. The particle size is $s = 0.1m$, where s is the side length of the cube. The sands true density is taken to be $2620Kg/m^3$. Based on the particle volume and sand density, the mass of each sand particle is $1.7Kg$. A coefficient of friction of 0.5 is used between: sand particles; the bucket and the sand particles, and the ground and the sand particles. A rectangular array of $30 \times 30 \times 30 = 27000$ particles is used [32]. The initial configuration of the particles is shown in Figure 4.13. An Eulerian search grid of size $10 \times 5 \times 10m^3$ and resolution

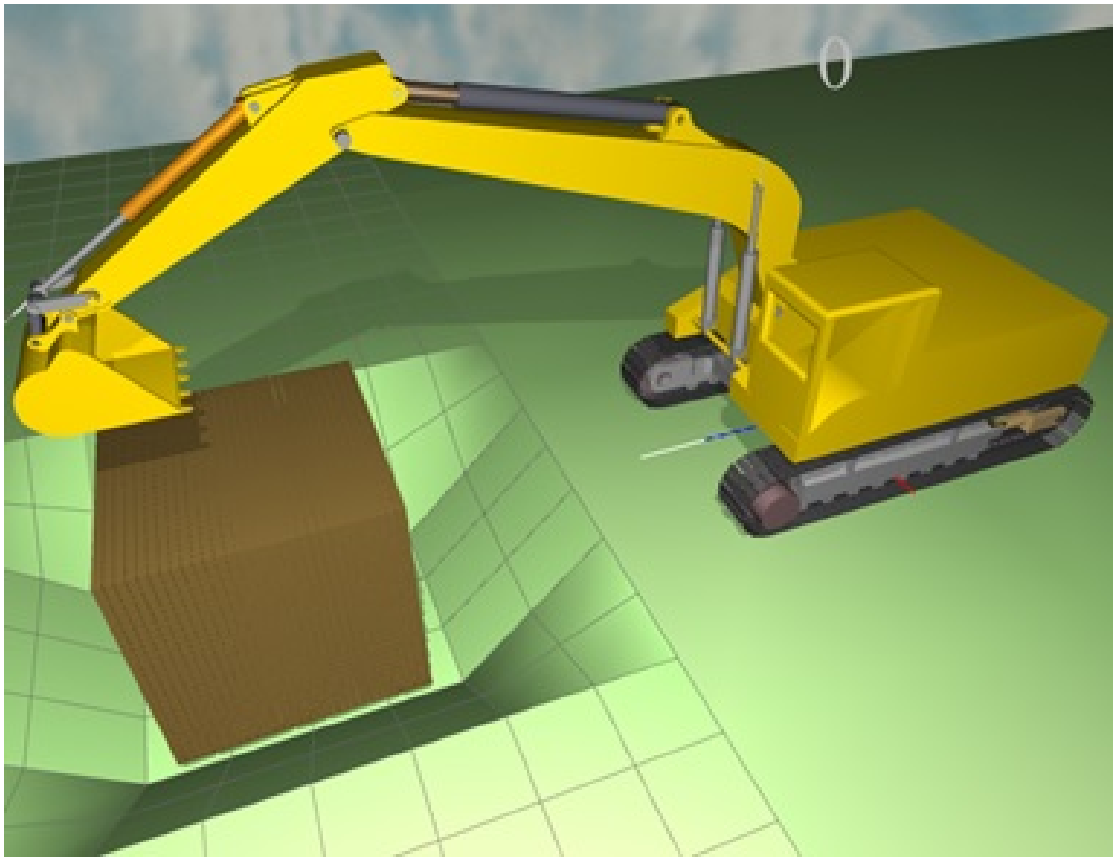

Fig. 4.13. Initial conditions of the backhoe and sand particles at time [32].

$90 \times 45 \times 90$ is used (Figure 4.14). The side length of a grid cell was chosen to be almost equal to the diameter of the particle. The simulation starts by letting the rectangular array of particles fall on the hole in the ground while holding the backhoe components stationary for 3 sec. This allows the sand particles to fill the hole and come to rest at about time 4 sec (Figure 4.15).

In order to keep the backhoe manipulator in the desired trajectory, desired actuator lengths of the boom, stick, and bucket are defined as a function of the simulation time. Figures 4.16 through 4.18 show the desired and actual actuator length of the backhoe manipulators components.

At time 3 sec the (left and right) boom actuator desired length is ramped from 1.5 m to 1.17 m in 3 sec, then kept constant for 1.5 sec, then ramped to 1.4 m in 3 sec,

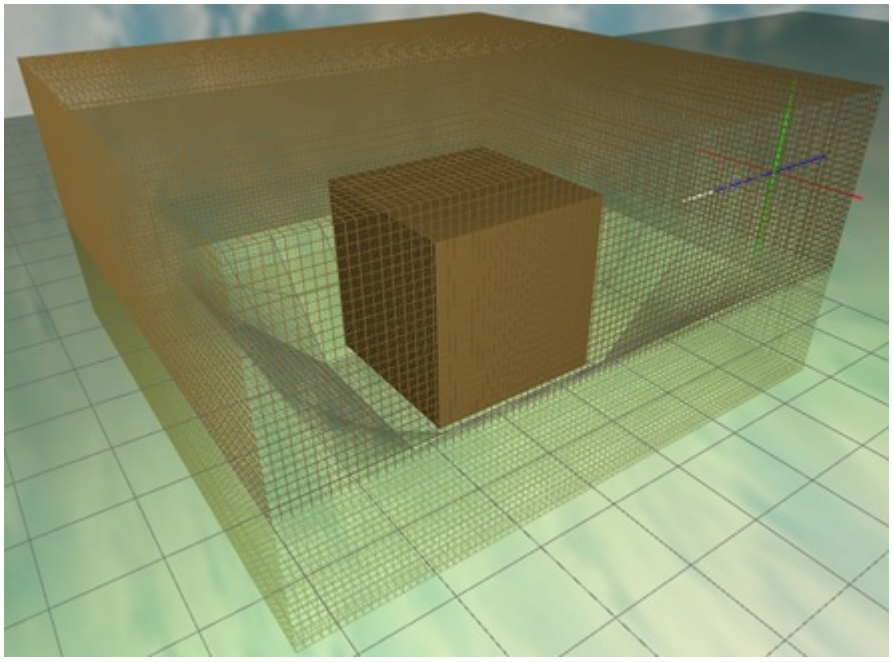

Fig. 4.14. Cartesian Eulerian search grid (Eulerian grid box size $10 \times 5 \times 10m^3$ and resolution $90 \times 45 \times 90$) [32].

then kept constant for the remainder of the simulation (Figure 4.16). The difference between the actual length of the boom (left and right) actuator in the simulation with sand particles and the desired length from time 4 to time 11 sec is due the resistance of the digging material (sand particles). For the simulation with no sand particles the actual length of the boom actuator in the simulation and the desired length are nearly coincident. Figure 4.17 shows the desired and actual stick actuator length with particles and without particles. The stick actuator desired length is kept fixed at 2 m. Figure 4.18 shows the desired and actual bucket actuator length with particles and without particles. At time 3 sec the bucket actuator desired length is ramped from 2 m to 2.37 m in 1 sec, then kept constant for 3 sec, then ramped to 1.95 m in 1.5 sec. Similar to the boom actuators, the difference between the actual length of the stick and bucket actuators in the simulation with sand particles and their desired length is due to the resistance of the sand particles during the digging operation. Figure 4.19 shows snapshots of the motion of the backhoe from time 5 to time 10 sec.

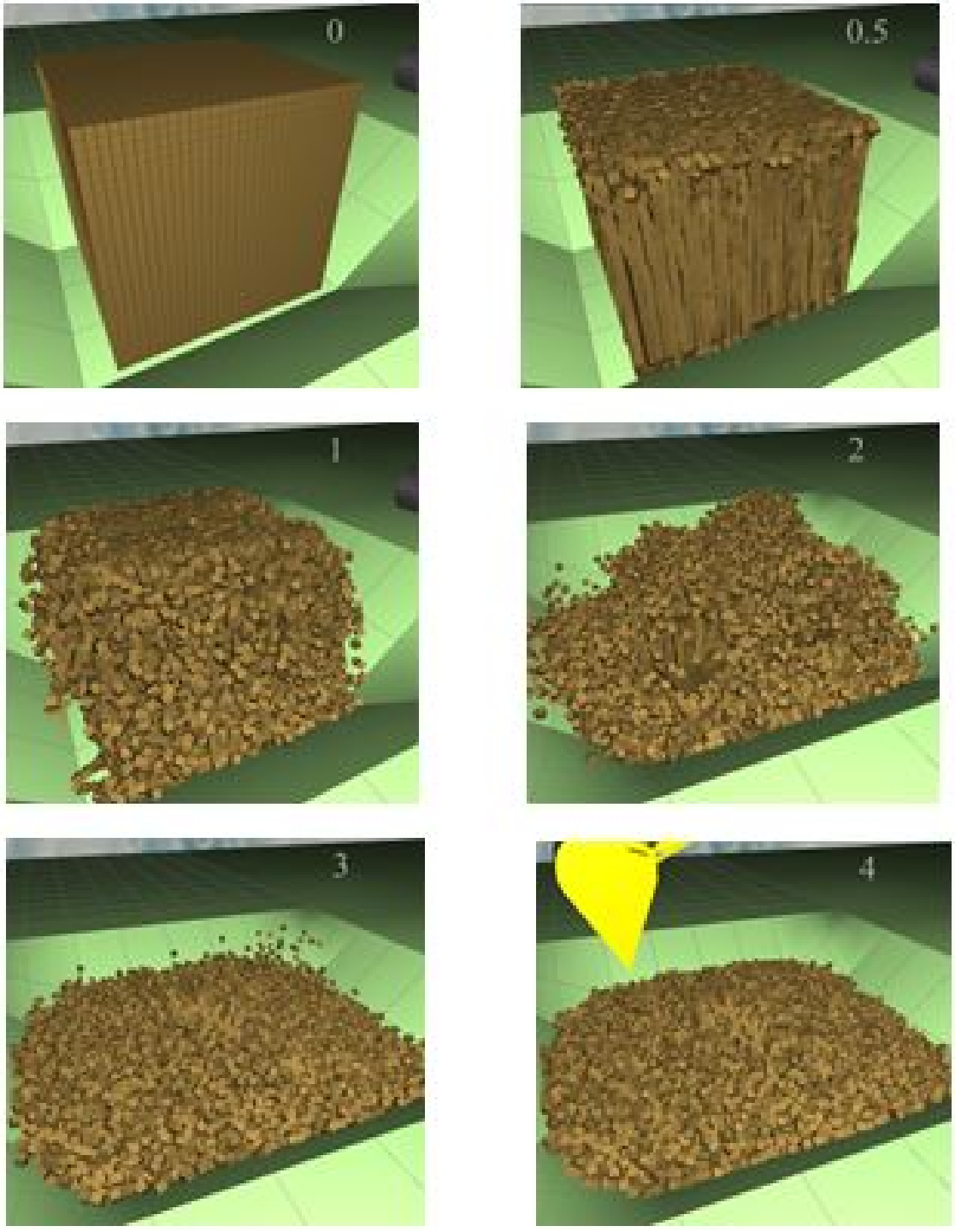

Fig. 4.15. Snapshots of the motion of the sand from time 0 to time $4s$ [32].

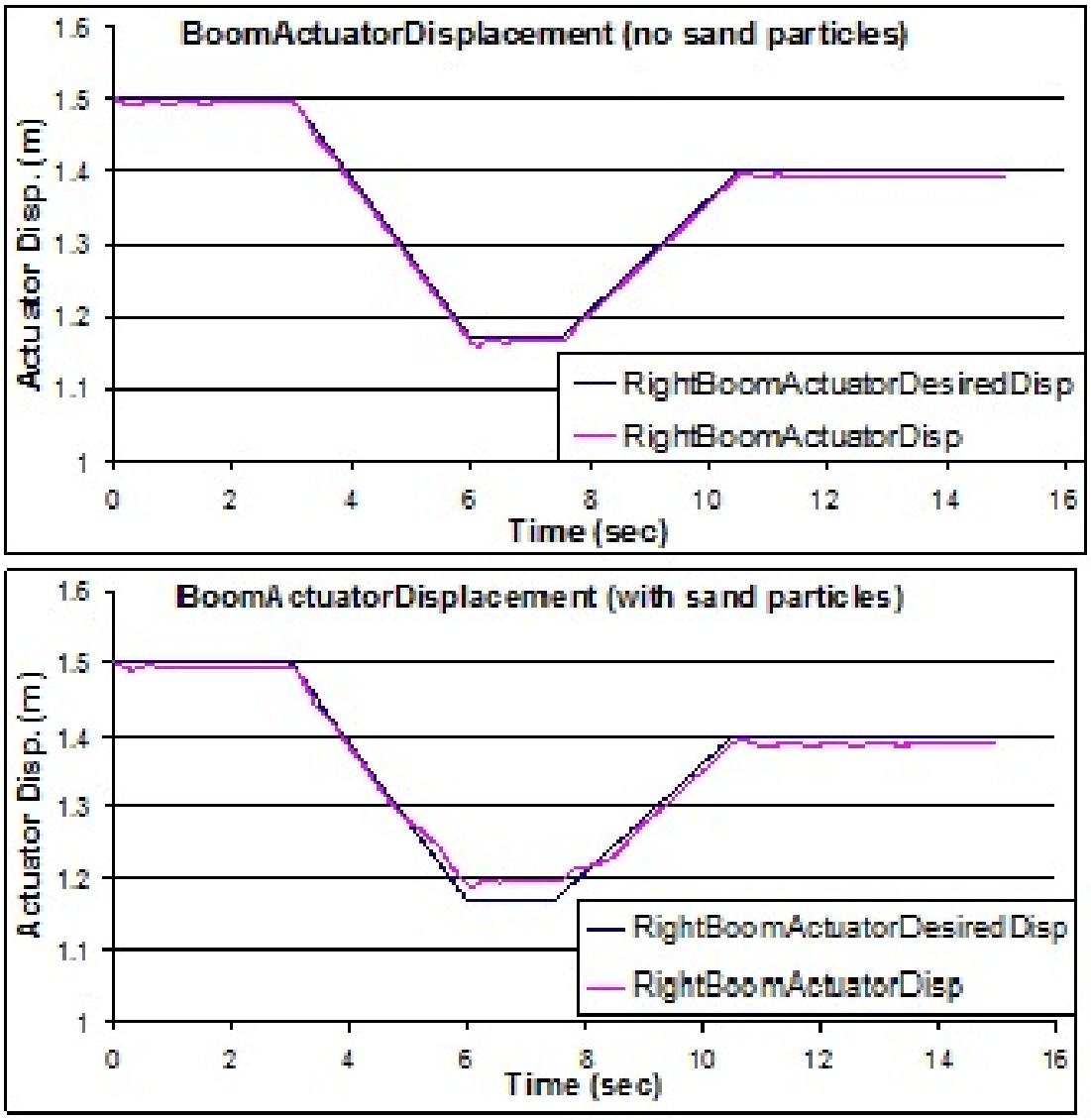

Fig. 4.16. Snapshots of the motion of the backhoe and sand pile from time 5 to 10 s [32].

Figures 4.20 through 4.24 show the backhoe manipulator actuators forces (digging forces) during the simulation. Calculating the actuators forces is needed in the design and selection of the hydraulic actuators. The hydraulic actuators must be able to exert the maximum required digging forces which are caused by the resistance of the sand

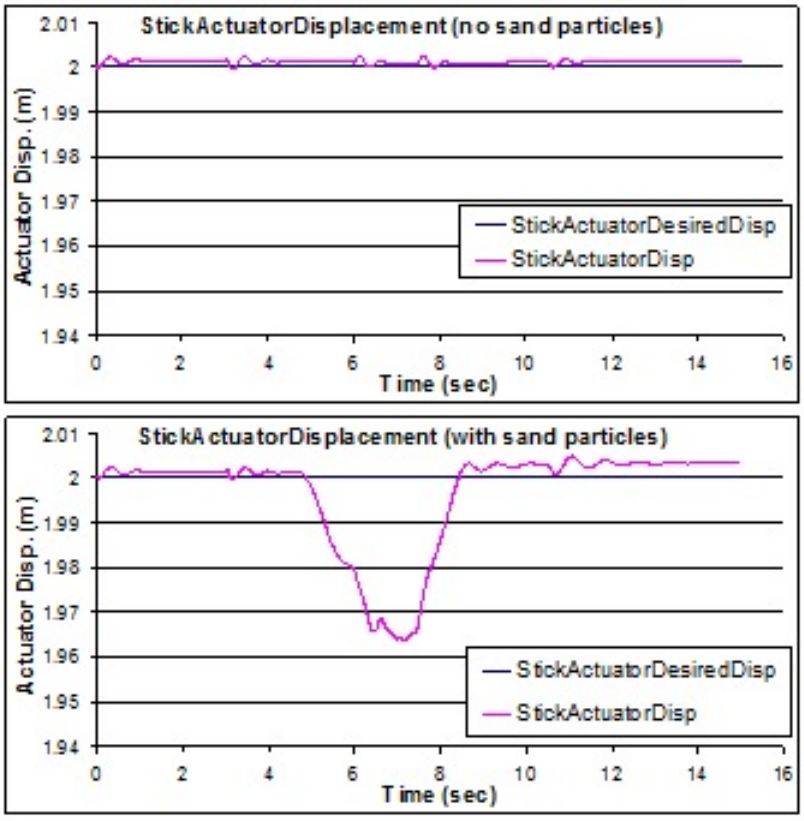

Fig. 4.17. Boom (left and right) actuators desired and actual displacements with no sand particles (top) and with sand particles (bottom) [32].

particles. Also, the magnitude of actuators forces vary depending on the trajectory of the bucket and motion speed during the digging operation.

Figure 4.20 shows the left and right boom actuator forces with particles and without particles. The value of the actuator forces are the same for simulations with sand particles and with no sand particles until the bucket hits the sand pile at about time 5 second. The oscillations from 0 to 2 sec occur because the PD controller is trying to stabilize the manipulator under the gravity load. At time 2 second the boom actuators exert a constant force of around -60,000 N to cancel the effect of the gravity on the backhoe manipulator. The oscillations from time 3 to 5 sec occur because of the start of the motion of the manipulator at time 3 sec. At time 5 second the values

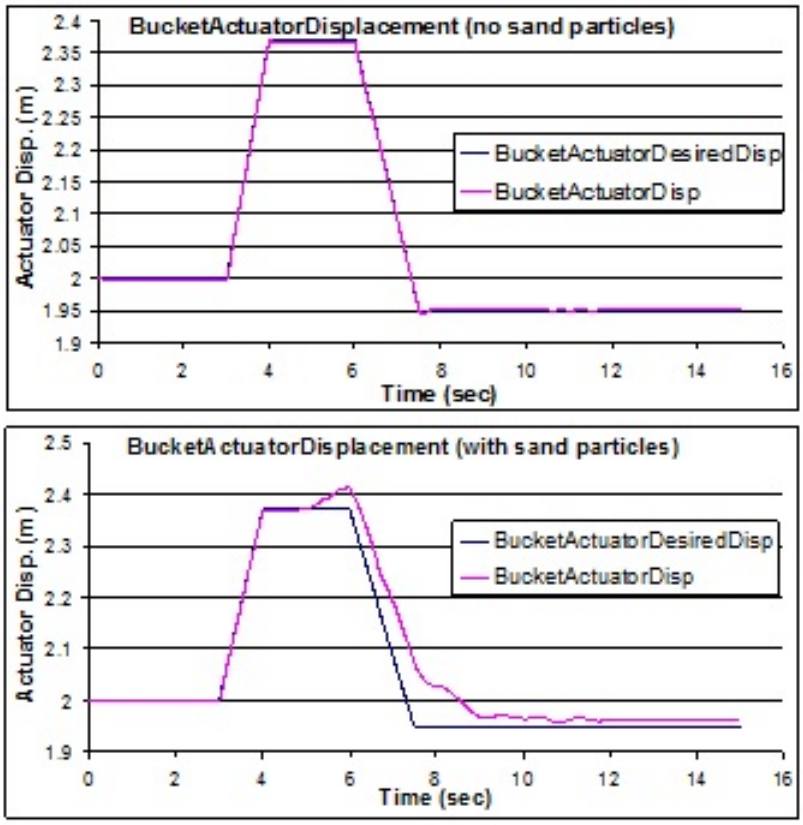

Fig. 4.18. Stick actuators desired and actual displacements with no sand particles (top) and with sand particles (bottom) [32].

of the boom actuator forces increase due to the resistance of the sand during digging until they reach the a maximum of about 250,000 N at about time 7 sec. At about time 8.5 sec, the digging operation ends (i.e. the bucket is no longer in contact with the sand pile) and the boom force drops until it reaches about -100,000 N at time 10 sec. The oscillations between 10.5 and 12.5 sec occur due to stopping the backhoe manipulator. The difference between the boom actuator forces with particles and without particles is the boom actuators digging force shown in Figure 4.21.

Similar to the boom actuators, at time 5 sec, for the simulation with sand particles, when the bucket hits the sand pile, the stick actuator force increases until it reaches a maximum of around -700,000 N at time 7 sec. Figure 4.22 shows the stick actuators force with particles and without particles. The difference between the two graphs

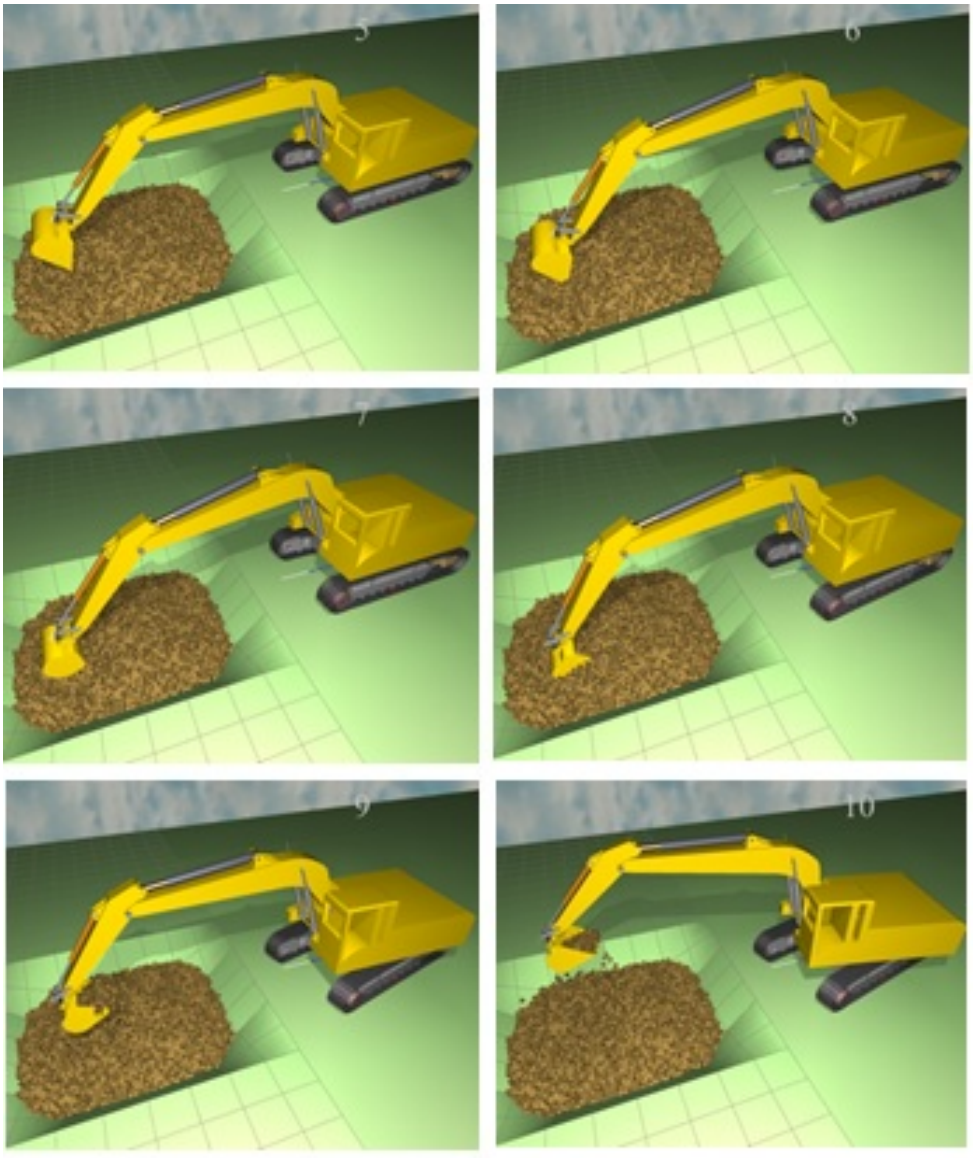

Fig. 4.19. Bucket actuators desired and actual displacements with no sand particles (top) and with sand particles (bottom) [32].

in Figure 4.22 is the stick actuators digging force shown in Figure 4.23. Figure 4.24 shows the bucket actuators force with particles and without particles. For the simulation with sand particles, the bucket actuators force is near zero for the first 5 seconds until the bucket hits the sand pile at time 5 sec. Then, the bucket actuator force increases until it reaches a maximum of 250,000 N at about time 7.5 sec. The

difference between the two graphs in Figure 4.24 is the bucket actuators digging force shown in Figure 4.25.

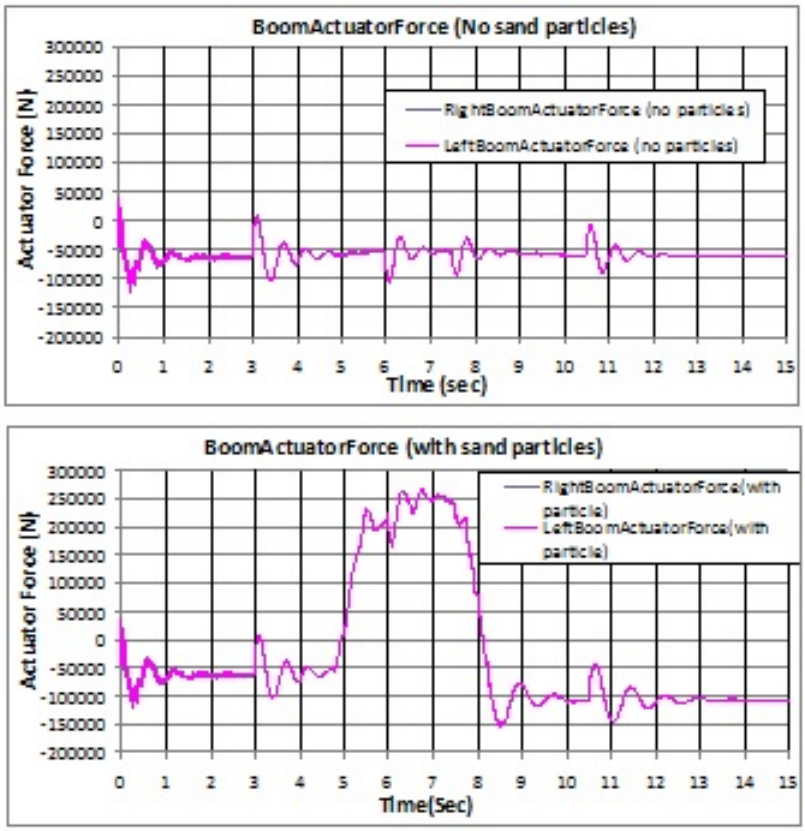

Fig. 4.20. Boom (left and right) actuators force with no sand particles (top) and with sand particles (bottom) [32].

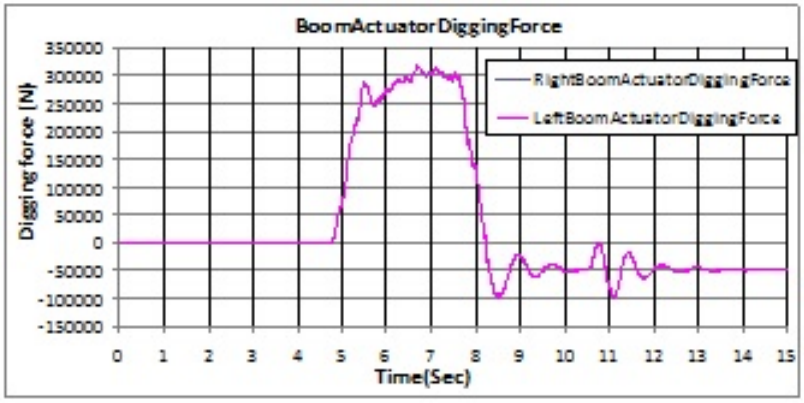

Fig. 4.21. Boom (left and right) actuators digging force (difference between actuators forces with sand with without sand) [32].

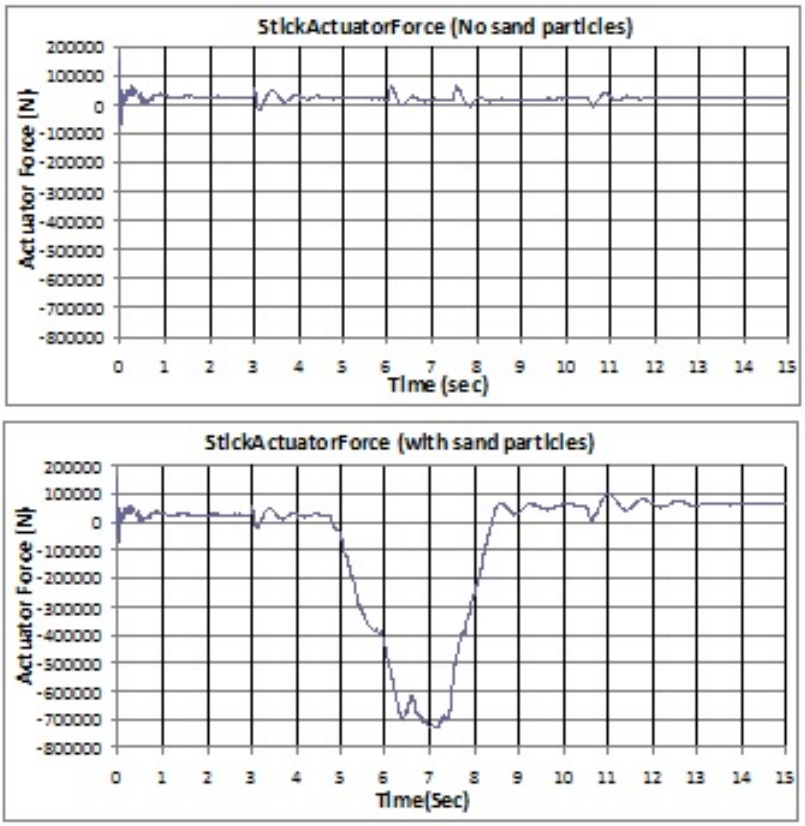

Fig. 4.22. Stick actuators force with no sand particles (top) and with sand particles (bottom) [32].

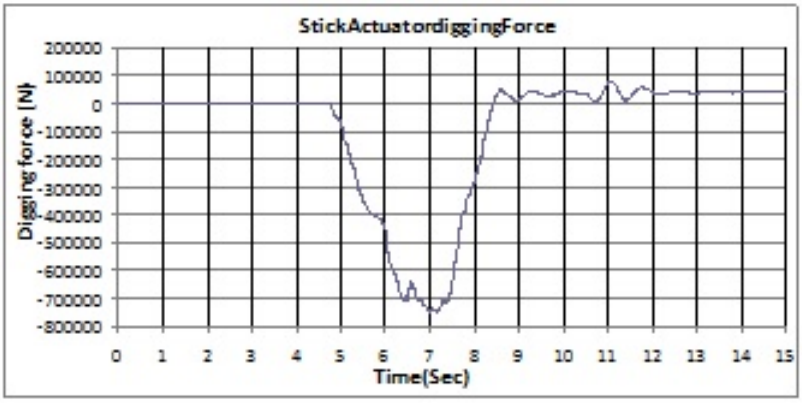

Fig. 4.23. Stick actuators digging force (difference between actuators force with sand with without sand) [32].

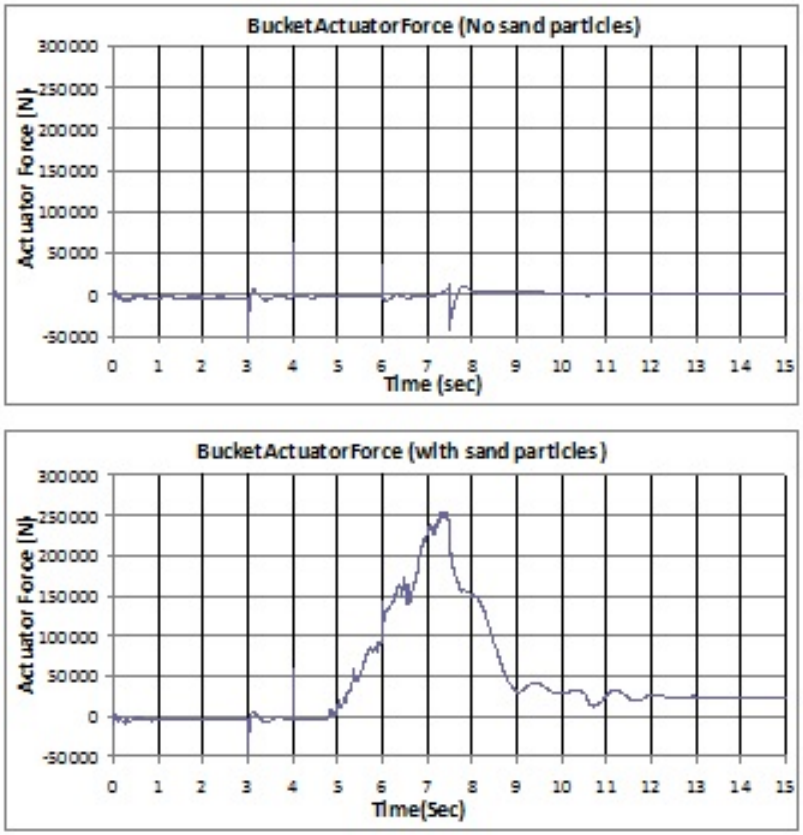

Fig. 4.24. Bucket actuators force with no sand particles (top) and with sand particles (bottom) [32].

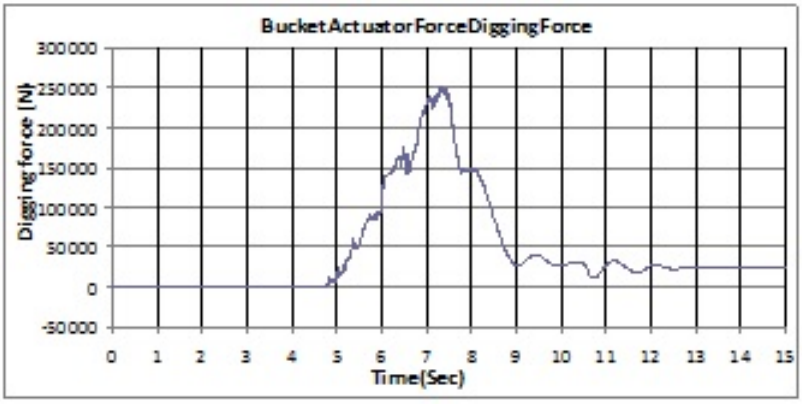

Fig. 4.25. Bucket actuators digging force (difference between actuator force with sand with without sand) [32].

5. CONCLUDING REMARKS AND FUTURE WORK

An explicit time integration multibody dynamics code was used to create a three-dimensional multibody dynamics model of a backhoe and a sand pile, in order to simulate a backhoe digging scenario. The model has the following major characteristics:

- A high fidelity multibody dynamics model is used for the backhoe that includes detailed models of the track and manipulator.
- Particles geometric shape can be modeled using three types of shape representations: superquadric surface, assembly of spheres, and polygonal surface.
- A Cartesian Eulerian grid contact search algorithm is used to allow fast contact detection between particles.
- Normal contact constraints are modeled using the penalty technique with a penalty spring and an asymmetric damper.
- Friction is modeled using an asperity-based approximate Coulomb friction model.
- The rigid bodies rotational equations of motion are written in a body-fixed frame with the total rigid body rotation matrix updated each time step using incremental rotations.
- The governing equations of motion are solved along with the contact constraint equations using a time-accurate explicit solution procedure.

The sand model was used to predict the sand angle of repose and flow rate during discharge from a conical hopper, using cubical sand particles with a coefficient of friction of 0.5. Parametric studies were performed in order to study the effects of the

particle size and the orifices diameter of the hopper on the angle of repose and sand flow rate. The results of the simulation were found to be in very good agreement with previously published experimental results.

Integrating the backhoe model and sand model provides the capability to predict the manipulator actuator forces needed to dig through a pile of sand. The various joint forces can also be obtained from the backhoe model [32], [33].

There is more work to be done in the future to improve the accuracy of the current model for simulating the backhoe digging operation and the sand flow hopper experiment:

- Increasing the number of the particles up to10 million. This goal can be achieved by using massive parallel processing (including GPUs).
- Modeling sand particles with various sizes and shapes (pentagonal and hexagonal) and analyzing the impacts of these shapes and sizes on the sand angle of repose and flow rate.
- Analyzing the stability of the backhoe with different: controller parameters, system configurations, motion speeds, digging depth, and digging material weight.

LIST OF REFERENCES

LIST OF REFERENCES

[1] P. K. Vaha and M.J. Skibniewski,"Dynamic model of excavator, in *Journal of Aerospace Engineering*, Vol. 6, No. 2, 1993

[2] R. Mitrev, R. Gruychev, and P. Pobegailo,"CAD/CAE Investigation of a large mining excavator in *Mechanical Design*, Vol 3, No. 1, pp. 17-22, 2011

[3] S. Frimpong,Y. Hu, and H. Inyang,"Dynamic modeling of hydraulic shovel excavators for geomaterials, in *International Journal of Geomechanics*, Vol. 8, No. 1, pp. 1532-3641, 2008.

[4] A.J. Koivo, M. Thoma,E. Kocaoglan, and J. Cetto,"Modeling and control of excavator dynamic during digging operation, in *Journal of Aerospace Engineering*, Vol. 9, No. 1, 1996.

[5] D. Sun and L. Xinhui,"Analysis of static stability of articulated hydraulic excavator with two degree of freedom, in *Advanced Materials Research*, Vol. 430-432, pp. 1663-1666,2012.

[6] P. A. Cundall,"A computer model for simulating progressive large-scale movements in block rock mechanics, in *Proc. Symp. Int. Soc. Rock Mech. Nancy*, Vol. 2, 1971.

[7] P.A. Cundall and O. D. L. Strack,"A discrete numerical model for granular assemblies, in *Geotechnique*, Vol. 29, No. 1, pp. 47-65, 1979.

[8] Y. C. Zhou, B. H. Xu,A. B. Yu, and P. Zulli,"An experimental and numerical study of the angle of repose of coarse spheres, in *Power Technology*, Vol 125, pp. 45-54, 2002.

[9] P. W. Cleary and M. L. Sawley,"There-dimensional modeling of industrial granular flows, in *CSIRO Mathematical and information sciences*, 1999.

[10] Y. C. Zhou, B. D. Wright, R. Y. Yang, B. H. Xu, and A. B. Yu,"Rolling friction in the dynamic simulation of sandpile formation, in *Physica A*, Vol 269, pp. 536-553, 1999.

[11] D. Zhang, and W. J. Whiten"The calculation of contact forces between particles using spring and damping models, in *Power Technology*, Vol. 88, pp. 59-64, 1996.

[12] T. Elperin, and E. Golshtein"Comparison of different models for tangential forces using the particle dynamic method, in *Physica A*, Vol. 242, pp. 332-340, 1997.

[13] G. Hu, Z. Hu, B. Jian, L. Liu, and H. Wan,"On the determination of the damping coefficient of non- linear spring- dashpot system to model Hertz contact for simulation by Discrete element method, in *Journal of computer*, Vol. 6, No. 5, 2011.

[14] C. Hogue,"Shape representation and contact detection for discrete element simulations of arbitrary geometries, in *Engineering Computation*. Vol 15, No. 3, pp. 374-390, 1998.

[15] P. W. Cleary and L. M. Sawley," DEM modeling of industrial granular flows:3D case studies and the effect of particle shape on hopper discharge in *Applied Mathematical Modelling*, Vol. 26, No. 2, pp. 89-111, 2002.

[16] D. Zhoa, E. Nezami, M. A. Y. Hashash, and J. Ghaboussi,"There dimensional discrete element simulation for granular materials in *Engineering Computer International Journal for Computer-Aided Engineering and Softwear*. Vol 23, No. 7, pp. 749-770, 2006.

[17] P. A. Cundall," Formulation of a three-dimensional distinct element model Part I: a scheme to detect and represent contacts in a system composed of many polyhedral blocks, in *International Journal of Rock Mechanics, Mining Science and Geomechanics*, Vol. 25, No. 3, pp. 107-116, 1988.

[18] C. J. Coetzee, A. H. Basson, and P. A. Vermeer,"Discrete and continuum modeling of excavator bucket filling, in *Journal of Terramechanics*, Vol. 44, pp. 177-186, 2007.

[19] W. A. Beverloo, H. A. Leniger, and Van de Velde,"The flow of granular solids through orifices, in *Chemical Engineering Science*, vol 15, pp, 260-269, 1960.

[20] M. A. Agurirre, J. G. Grande, and A. Calvo,"Granular flow through an aperture : Pressure and flow rate are independent," in *Franco-Argentina International Associated Laboratory in the Physics and Mechanics of Fluid*, 2011.

[21] E. Freyssingeas, M. J. Dalbe, and J. C. Geminard,"Flowers in flour: Avalanches in cohesive granular matter, in *Physics Review*, 83, 051307, 2011.

[22] C. Perge, M. Aguirre, A. Gago, and L. Pugnaloni,"Evaluation of pressure profiles during the discharge of a silo," in emphPhysics Review, 85,021303, 2012.

[23] P. Kumar, E. Beck, and S. Das,"Preliminary investigation on the deposition of fine powder through miniature hopper-nozzles applied to multi-material solid freeform fabrication," in *Department of mechanical engineering The university of Michigan, Ann Arbor*.

[24] H. Maleki, F. Ebrahimi, E. Oskoee,"The angle of repose spherical grains in granular hele show cells a molecular dynamic study, in *Mat. Stat-Mech*, 2007

[25] T. M. Wasfy,"Modeling spatial rigid multibody systems using an explicit-time integration finite element solver and a penalty formulation, in *ASME Paper No. DETC2004-57352, 28th Biennial Mechanisms and Robotics Conference, DETC, Salt Lake, Utah*, 2004.

[26] T. Wasfy,"Asperity spring friction model with application to belt-drives, in *ASME Paper No DETC2003/VIB-48343*,2003.

[27] M. J. Leamy and T. M. Wasfy,"Transient and steady-state dynamic finite element modeling of belt-drives, ASME Journal of Dynamics Systems," in *Measurement, and Control*, Vol. 124, No. 4, pp. 575-581, 2002.

[28] T. M. Wasfy,"Finite element modeling of the dynamic response of tracked vehicles, in *ASME Paper No DETC2009-86563*, 2009.

[29] Advanced Science and Automation Corp.,"DIS :Dynamic Interactions Simulator," 2013. [Online]. Available: http://www.ascience.com/ScProducts.htm. Last accessed March 2013.

[30] T. Wasfy,"Introduction to Hybrid-Electric Vehicle ,in *IUPUI Course ME 5970*, Chapter 3.

[31] Sand, [Online]. Available:Wikipedia.org, http://en.wikipedia.org/wiki/Sand. Last modified March 2013.

[32] S. G. Ahmadi, T. M. Wasfy, H. M. Wasfy, and J. M. Peters,"High-Fidelity modeling of a backhoe digging operation using an explicit multibody dynamics code with integrating discrete particle modeling capability, in *Proceedings of ASME International Design Engineering Technical Conference*, 2013.

[33] T. M. Wasfy, S. G.Ahmadi, H. M. Wasfy, and J. M. Peters,"Multibody Dynamic Modeling of Sand Flow from a Hopper and Sand Pile Angle of Repose, in *Proceedings of ASME International Design Engineering Technical Conference*, 2013.

[34] PTC Product and Service Advantage,"Creo Elements/Pro 5.0," 2013. [Online]. Available: http://www.ptc.com/products/creo-elements-pro/wildfire5/. Last accessed March 2013.