Integrating a Hierarchy of Simulation Tools for Legged Robot Locomotion

Andrew Slatton*, Daniel Cohen*, Yang Ding*, Paul B. Umbanhowar*, Daniel I. Goldman*,
G. Clark Haynes†, Haldun Komsuoglu†, and Daniel E. Koditschek†

*School of Physics
Georgia Institute of Technology
837 State St NW, Atlanta, GA 30332

†Electrical and Systems Engineering
University of Pennsylvania
200 S. 33rd St, Philadelphia, PA 19104

Abstract— **We are interested in the development of a variety of legged robot platforms intended for operation in unstructured outdoor terrain. In such settings, the traditions of rational engineering design, driven by analytically informed and computationally assisted studies of robot-environment models, remain ineffective due to the complexity of both the robot designs and the terrain in which they must operate. Instead, empirical trial and error often drives the necessary incremental and iterative design process, hence the development of such robots remains expensive both in time and cost, and is often closely dependent upon the substrate properties of the locomotion terrain. This paper describes a series of concurrent but increasingly coordinated software development efforts that aim to diminish the gap between easily interfaced and physically sound computational models of a real robot's operation in a complex natural environment. We describe a robot simulation environment in which simple robot modifications can be easily prototyped along and "played" into phenomenological models of contact mechanics. We particularly focus on the daunting but practically compelling example of robot feet interacting granular media, such as gravel or sand, offering a brief report of our progress in deriving and importing physically accurate but computationally tractable phenomenological substrate models into the robot execution simulation environment. With a goal of integration for future robot prototyping simulations, we review the prospects for diminishing the gap between the integrated computational models and the needs of physical platform development.**

I. INTRODUCTION

We have been collaborating with others [1] on the development of a variety of legged robot platforms intended for operation in unstructured outdoor terrain such as vertical (building exteriors, trees, etc.) [2] and broken, unstable level ground [3]. Mobile robot design for such natural outdoor settings defies purely analytical approaches. Successful implementation [4], [5], [6], [7] requires repeated iteration of passive mechanical construction and actuated system components, ultimately driving the design cost both financially and in time. Rapid prototyping techniques such as SDM [5] and modular electromechanical designs [3], [8] enlarge the accessible design space and accelerate the iterated design-build-test cycle, and these novel materials may be more challenging to model in computationally tractable form than the previous generation of rigid members joined along fixed seams. Moreover, substrate properties, for which there may not even be clear first principles models, and contact mechanics play a central role in determining ultimate performance. A flexible and programmable simulation environment that captures both the robot physics as well as the substrate properties would be a great asset further speeding the design-test cycle and guiding the physical construction process. Although we have gained substantial benefit from running different types of simulations at different points in the design cycle, it remains impossible to draw any reliable predictions about the likely success of a given robot in the common, natural environments we target. This paper will explore the various components of the simulation tools we have developed, and review the prospects for diminishing the gap between the integrated computational models and the needs of physical platform development.

We introduce "SimLib"[1], an environment conducive to the rapid prototyping of kinematic chain simulations. This environment allows a robot behavior developer to quickly test modifications to robot morphology, as well as to the contact models to simulate interactions between feet and surfaces. With an API identical to that of a series of physical robots [3], [7], [9], [2], we provide a tool for robot programmers to design behaviors with real-time, interactive simulations, by reducing simulator accuracy in favor of speed, or to batch run computationally-intensive simulations of higher accuracy.

Here, the problem goes beyond questions of computational efficiency and speedup of simulation for rapid prototyping. Locomotion of a robot over complex ground requires novel models of the foot-surface interaction which are only now being developed and validated. We focus on common substrates like gravel or sand that are characterized by small grain interactions between myriad solid particles. Such granular media are challenging to a robot because the material can develop fluid-like and solid-like forces in response to penetration of

---

[1] SimLib is part of the RHexLib control software, originally developed by the DARPA RHex project [4] http://www.rhex.org and continued within the DARPA RiSE project [1] http://www.riserobot.org, and is available by request from the authors. Future plans exist for eventual public distribution.

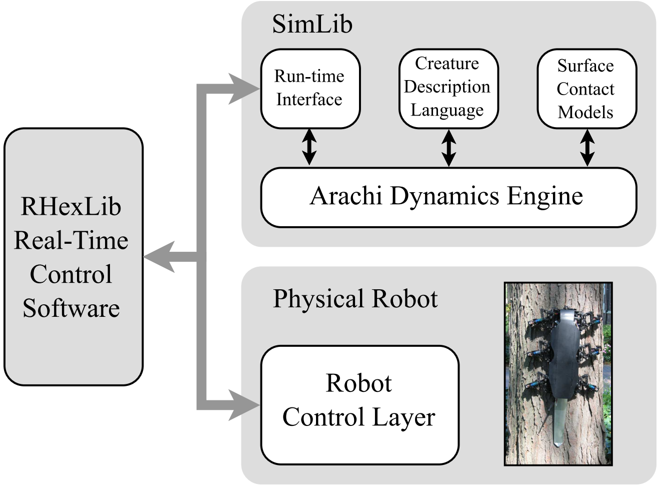

Fig. 1. Organization of SimLib within our robot control software. SimLib exposes an API identical to a physical robot. A run-time interface translates robot translates motor commands and sensors readings to and from the simulation. The CDL is used to input a simulation environment. Lastly, contact models describe specific surface interactions to use within the simulation.

a limb, and this transition depends on the packing of the material. They are challenging to model effectively as well.

We conclude the paper with a description of the phenomenological abstraction models - yet another intermediate computational layer in this hierarchy - designed to present the kinematic representation of the robot with a surface model accurate enough to give the designer insight into how the physical robot may fare in the physical environment. We discuss the integration path and report progress on our efforts to import these phenomenological models into the SimLib design environment.

II. SIMLIB: SIMULATING LEGGED ROBOTS

The top level of our simulation hierarchy, “SimLib” offers an environment conducive for rapid prototyping of kinematic chain simulations. Using a scripting language designed specifically to encode basic simulations, as well as allowing a simulation designer to specify contact models amongst surfaces in the simulation, SimLib is geared toward simulations of legged robots interacting with non-trivial surface types.

SimLib itself consists of the Creature Description Language (CDL), a library of contact models, and a run-time interface that exposes a simulation API identical to that of a physical robot. A diagram showing the various components, and how they coarsely interact is shown in Fig. 1.

SimLib interfaces with the Arachi Dynamics Engine[2], a C++ simulation library suited for multibody dynamics. While the components of SimLib are designed to be agnostic respecting any specific simulation implementation, the Arachi engine is well-suited for our simulations. Its ability to simulate at the exact time-of-contact between two surfaces allows relatively straightforward incorporation of customized robot-surface interaction models, as will be discussed below.

A. *Creature Description Language*

The CDL is an abstraction of the low-level SimLib simulation library, providing a simple language, programmed using the common tools of `lex` [10] and `yacc` [11] , that is designed specifically to describe multi-body linkages. It has support for an assortment of shapes, joints, and other common features of robot simulations. The language syntax facilitates building a tree representation of a simulation, in which objects are defined by their connections to previous elements. The connections between objects include definitions of linkages using rigid transforms or by defining joints. Using these language features, the CDL allows a simulation designer to describe complex robots using much less code than would be required if interfacing directly with a C++ library.

A CDL script typically involves performing coordinate transforms, declaring geometric objects within the environment, and connecting objects via degrees of freedom. A simple example is shown in Prog. 1.

Program 1 Simple CDL Example

```
Camera(4,0,0); // (X,Y,Z) position of camera
Lookat(0,0,0); // position camera is pointed at
Gravity(0,0,-9.8);
FreeBody {
  // A Sphere, with radius 1.0 m, mass 0.5 kg
  Sphere(1.0) : (mass = 0.5);
}
```

We define basic camera properties (which we shall skip in further examples) using the common camera location and focus point, as well as add in global gravity into the simulation. The use of `FreeBody{...}` surrounding the declaration of the sphere is an example of stacking operations. The sphere is *disconnected* from the workspace, since a “free body joint” is placed in between (three linear degrees of freedom plus a spherical joint). The sphere, with mass pulled by gravity, falls downward upon simulation.

A more detailed example, that of a two-link pendulum, demonstrates the use of joint types, object declarations, and using rotations and translations to stack coordinate frames. This is seen in the Prog. 2.

Upon each successive call in the above script, the simulation introduces a new frame of reference. After the first call to `RevoluteJoint` (and rotations to correctly align the joint), all enclosed calls are connected to the world through the revolute joint. A fixed transformation then translates to the center position where we declare a block of our pendulum, a new frame of reference, before adding another joint and block as a second portion of the pendulum. To run a simulation, the CDL interpreter—written in C++—reads a CDL script, performs the associated function calls within the

---

[2] Arachi Dynamics Engine, http://www.arachi.com, open-sourced as the TAO Dynamics Engine, http://simtk.org/home/tao_de

Program 2 Dual-Link Pendulum Example

```
Gravity(0,0,-9.8);
// place revolute joint with rotation aligned with X axis
Rotation(0,1,0,pi/2),
RevoluteJoint,
Rotation(0,1,0,-pi/2) {
  Translation(0,0.5,0) {
    Block(0.1,1.0,0.1) : (mass = 1);
    Translation(0,0.5,0) {
      // place second joint at end of first block
      Rotation(0,1,0,pi/2),
      RevoluteJoint,
      Rotation(0,1,0,-pi/2) {
        Translation(0,0.5,0) {
          Block(0.1,1.0,0.1) : (mass = 1);
        }
      }
    }
  }
}
```

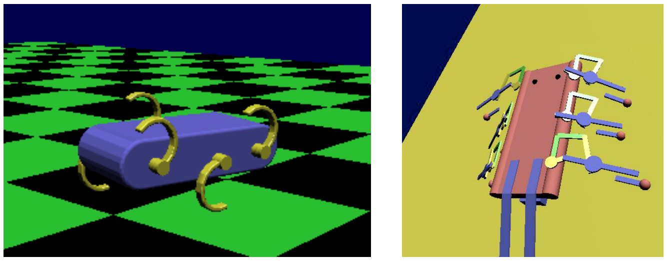

Fig. 2. The RHex (left) and RiSE (right) robots, as simulated in the Arachi Dynamics Engine, modeled using the Creature Description Language (CDL). Models include definitions of leg compliances, as well as the definition of four-bar mechanisms in the case of the RiSE robot [2].

simulation library, and displays output. Upon simulation of this two-link pendulum, the system exhibits typical chaotic behavior.

Built-in support has been developed for the named geometric specification of spheres, cylinders, blocks, and general convex shapes. Analogously named and supported joint types include the most common pairs: revolute, prismatic, and spherical. Furthermore, naming of joints allows one to apply active control of joint torque from elsewhere. Alternatively, providing values for stiffness and damping of a joint creates unactuated, compliant joints.

Using the Creature Description Language, utilizing preprocessors such as ‘cpp’ [12] and ‘m4’ [13] to enable the use of macros, we have built simulations of the RHex [4] and RiSE [2] robots, complete with definitions of compliant elements and complex linkages, in far less code writing than by using the original simulation interface. Furthermore, many typical programming errors are avoided through use of the simpler simulation language. All object creation and deletion is handled within the CDL, reducing memory leaks, and the CDL’s stack handles the OpenGL and Arachi coordinate frames.

The CDL layer also includes syntax supporting the description of named surface types, passed as parameters to defined objects, such as `Block(1,1,1) : (surface="block_surface")`. By then listing pairwise relationships amongst surfaces, we define specific contact models to utilize when the named surfaces interact. This approach has been used to create simulated feet to model adhesion during climbing and represents the most efficient interface between the abstract rigid body and molecular dynamics simulation levels in SimLib.

B. Run-time Operation

SimLib permits the use of simulation in lieu of a physical robot, providing programmatic interfaces identical to that of a robot. Adapted to the RHexLib robot control software, positions, velocities, and forces can be read from the simulation system, to which commanded torques are passed, with additional internal modeling of motor states.

We can vary our simulations between run-time versions in which an operator may interact with a simulated robot, and much slower than real-time versions for high-accuracy simulations of physical phenomena. These varying levels of accuracy serve different sets of user goals at different points of the design cycle. To be of greatest use for real applications, we desire the capability to move back and forth between these levels at will. A relevant analogy is provided by distinguishing CAD programs that provide basic renderings in real-time, but must be asked to produce the nicer-looking ray- traced versions which take much longer. Integrating models of interaction with the physical substrate remains a major hurdle.

III. MOLECULAR DYNAMICS

Locomotion of a robot over complex ground like sand cannot be understood without recourse to a model of the environment. Granular media are challenging to a robot because the material can develop fluid-like and solid-like forces in response to penetration of a limb, and this transition depends on the packing of the material. The bulk of the theory/models for forced granular media is either in the rapid granular flow regime [14] or the quasi-static deformation regime [15] (soil mechanics [16]), both unsuitable for models in the locomotion regime. There have been a few studies of impact and drag in a granular medium in the regime of interest; the bulk have been in the slow, quasi-static regime [17], [18]. Most studies of drag have been conducted on relatively simple objects which were not actuated or multi-DOF. Thus we need a modeling approach that will allow us to compare in experiment and simulation multi-DOF objects interacting with granular media

Our approach to design and prediction of the interaction of the robot with granular media is through Molecular Dynamics (MD) and phenomenological models. Our modeling approach will yield accurate but slow modeling through coupled multi-body/MD simulation and fast but less accurate coupled multi-body/phenomenological models. MD models are useful in understanding diverse regimes of granular

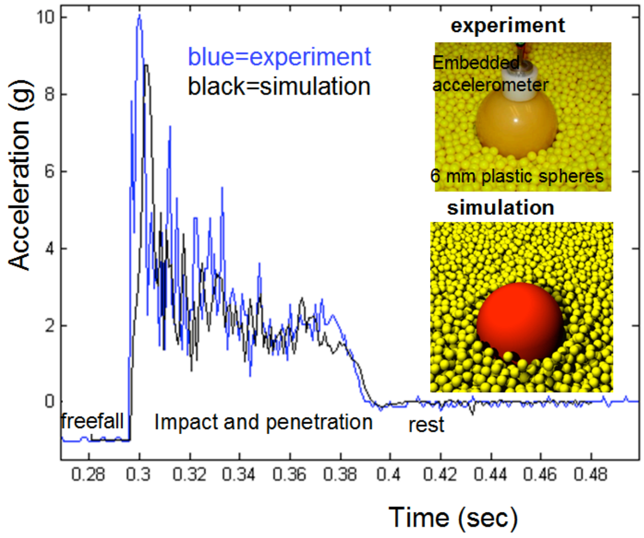

Fig. 3. Acceleration of a sphere during impact into a granular medium in experiment and MD simulation at 2 m/sec. Once grain interaction parameters are tuned, the simulation captures dynamics for a range in initial impact velocity.

behavior such as impact and penetration [19], pattern formation [20] and packing of spheres [21]. MD models directly model each grain-grain interaction to produce models of bulk flow. Typically grain-grain models include a normal elastic restoring force, a normal dissipation (mimicking energy loss during collision), which depends on collision velocity, and tangential dissipative frictional forces. It has been shown when the microscopic physics is captured, the bulk properties (even structures at grain level), agree remarkably [22].

A. *Experiments and numerical models of impact*

We have developed an experimentally validated MD simulation that accurately reproduces the forces in a complex penetration event: the force on a freely falling intruder as it impacts a granular medium (Fig. 3). We use models found in [19] and [23] to simulate up to 10^7 particles on a desktop PC. In Fig. 6, we show the acceleration of a 5 cm diameter as it impacts a collection of 6 mm plastic spheres. After adjustment of grain-grain collision parameters like elasticity, coefficient of restitution, and static and dynamic friction, we obtain excellent agreement in the acceleration time profile of the large sphere as it impacts and comes to rest within the granular medium. The simulation allows direct interrogation of the forces at the grain level, something that is currently impossible in experiment in three dimensions.

B. *Accurate numerical simulation of robot movement*

Experiments on a hexapod robot, SandBot [9], have revealed a striking sensitivity to packing of the granular media (Fig. 2). To understand this sensitivity requires understanding the penetration and drag forces on a limb. We have successfully integrated our 2D Molecular Dynamics (MD) code with the commercial software package Working Model 2D (Design Simulation Technologies) to couple grain interaction to actuated multi-body simulation to compute the dynamical forces on a model SandBot (Fig. 4) which can be controlled in an identical way to the real device; the multi-body solver integrates the equations of motion of the coupled links and the MD calculates the resultant force from the grain collisions. This allows us to do pilot studies of efficacy of different locomotor models of the robot as well as to develop qualitative understanding of force production; the MD allows us to visualize the stress field.

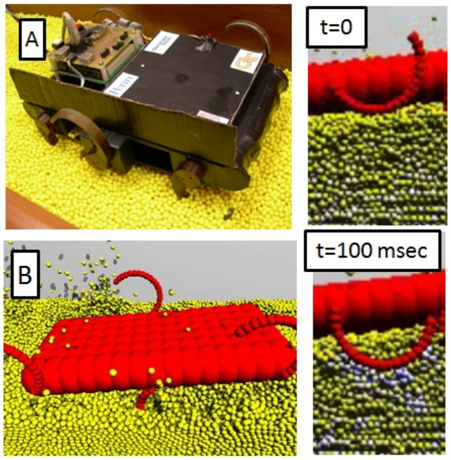

Fig. 4. Left panels: (A) Experimental model locomotor, Sandbot, whose limb kinematics can be precisely controlled. (B) Integrated multi-body / Molecular Dynamics simulation of Sandbot. Right Panels: time development of stress (lighter grains) during foot impact.

C. *Phenomenological models for foot penetration forces*

MD integrates slowly but accurately. We would like to have simpler few-parameter "phenomenological" models that can be integrated into robotic controllers.

It is a research question into the physics of granular media to discover a computationally well-characterized hierarchy of such models and, then, a research question in robotics to determine the right level in this hierarchy to expose to the locomotion simulation at what stage in the design process. The current models with just a few parameters describing granular penetration dynamics of simple objects [17], [24], [25] have been used successfully in analytical models that capture gross features of robot motion on sand for a range of packing fractions and limb rotation frequencies [9]. We propose to build on these analytical models to develop the more detailed layers of phenomenological simulation simulations that accurately predict performance features of legged robots (speed, power) over a larger region of parameter space, including more realistic kinematics and different foot, limb and body morphologies. However, the level of descriptive detail that must be incorporated in a simple model is not yet known. We expect that for simple shapes (spheres, disks) and simple kinematic trajectories (constant velocity penetration), that the current models will yield results that will be accurate with respect to the MD simulation. The goal will be to add complexity to the limb shape and trajectories and develop simple models that maintain accuracy in a restricted range of parameters (grain friction, particle diameters). We will be able to check failure of the models by performing MD simulation in a more limited range of parameters.

To this end, we will continue to study the detailed physics of interaction (including penetration and drag) of simple and complex shapes with a range of preparations of granular media. We will investigate how existing force models for vertical and horizontal drag that scale with depth and the square of depth respectively can be adapted to describe richer kinematics where foot angle, direction, and speed are all changing simultaneously. Since both foot deformation and changes in volume fraction, have been shown to dramatically affect locomotor performance, we will use high speed x-ray imaging and MD simulation to simultaneously observe deformation of the feet and forces on the feet and modify existing force laws through few parameters which describe compliance . Reducing overall mass is often an issue of importance; accordingly, we will study how the thrust forces are generated by feet of fixed area but differing internal geometry (e.g. a small plate vs a large plate with holes). We will investigate in experiment and MD simulation whether penetration force models need account for details of foot geometry (possible toes, compliant elements, shape, etc), or can parameterize these specifications using, for example, just the foot area. Earlier work showed that a strategy for effective motion is to maintain shear stress below yield stress while accelerating, but detailed models are not available to predict kinematics which optimize this. We will thus perform experiments on motion profiles that maximize and minimize applied stresses in combination with varying the penetration depth.

IV. TOWARDS FUTURE INTEGRATION EFFORTS

In this paper, we have reviewed the SimLib robot development environment, focusing on its user interface, Creature Description Language, as well as the "hooks" it provides to access in principle "arbitrarily" realistic surface contact models. For simulations involving robot interaction with granular media, realistic models such as MD indeed run accurately—but far too slowly, in practice, to integrate directly in this manner into the run-time compatible SimLib environment.

We have explored as well the challenges to developing appropriately abstract phenomenological models through the introduction of alternative intermediate computational layers in this hierarchy. These layers must be carefully designed to present the lumped parameter representation of the robot with a surface model accurate enough to give the designer insight into how the physical robot may fare in the physical environment — at the level of detail, accuracy and throughput appropriate to the particular design stage at which it is invoked. Since "internal models" abound in control theoretic thinking, it is important to note that such simpler, few-parameter models might also be integrated into the robot controller itself.

Our two groups are continuing to explore these issues and consider the best next steps toward the desired integration. From experimental measurements and MD simulation we have built a database of phenomenological models [25] that can be used to rapidly simulate motion in complex media. MD allows us to determine forces at the grain level and allows us to vary in simulation grain parameters like friction and elasticity needed to inform the phenomenological models that are relevant to natural substrates but are inconvenient to vary in experiment. We continue to explore the physics, computational science, and robotics science underlying an effective future integration path to bring these models into the SimLib environment, and will report new progress on our efforts in the near future.

ACKNOWLEDGMENTS

The authors would like to thank Uluc Saranli and Al Rizzi for work on the early development of SimLib, as well as Adam Kamor and Chen Li for work on Molecular Dynamics. SimLib development was performed under the RiSE project, supported by the Defense Advanced Research Projects Agency within the DSO Biodynotics Program under contract DARPA/SPAWAR N66001-03-C-8045 and N66001-05-C-8025. Haynes is supported by an IC Postdoctoral Fellowship. Umbanhowar and Goldman are supported by the Burroughs Welcome Fund.

REFERENCES

[1] K. Autumn, M. Buehler, M. Cutkosky, R. Fearing, R. J. Full, D. Goldman, R. Groff, W. Provancher, A. A. Rizzi, U. Saranli, A. Saunders, and D. E. Koditschek, "Robotics in scansorial environments," *Unmanned Ground Vehicle Technology VII*, vol. 5804, no. 1, pp. 291–302, 2005. [Online]. Available: http://link.aip.org/link/?PSI/5804/291/1

[2] M. J. Spenko, G. C. Haynes, J. A. Saunders, M. R. Cutkosky, A. A. Rizzi, R. J. Full, and D. E. Koditschek, "Biologically inspired climbing with a hexapedal robot," *Journal of Field Robotics*, vol. 25, no. 4-5, pp. 223–242, 2008.

[3] H. Komsuoglu, K. Sohn, and D. E. Koditschek, "A physical model for dynamical arthropod running on level ground," in *Proceedings of 11th International Symposium on Experimental Robotics*, 2008.

[4] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: A Simple and Highly Mobile Hexapod Robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001. [Online]. Available: http://ijr.sagepub.com/cgi/content/abstract/20/7/616

[5] J. G. Cham, S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky, "Fast and robust: hexapedal robots via shape deposition manufacturing," *International Journal of Robotics Research*, vol. 21, no. 10, pp. 869–883, 2002.

[6] K. Autumn, M. Buehler, M. Cutkosky, R. J. Fearing, R. Full, D. Goldman, R. Groff, W. Provancher, A. A. Rizzi, U. Saranli, and D. E. Saunders, A. Koditschek, "Robotics in scansorial environments," in *Proceedings of SPIE 2005*, 2005, pp. 291–302.

[7] J. E. Clark, D. I. Goldman, P.-C. Lin, G. Lynch, T. S. Chen, H. Komsuoglu, R. J. Full, and D. E. Koditschek, "Design of a bio-inspired dynamical vertical climbing robot," in *Robotics Science and Systems*, 2007.

[8] M. Yim, D. G. Duff, and K. D. Roufas, "Polybot: a modular reconfigurable robot," in *Proceedings of the IEEE Conference on Robotics and Automation*, 2000, pp. 514–520.

[9] C. Li, P. B. Umbanhowar, H. Komsuoglu, D. E. Koditschek, and D. I. Goldman, "Sensitive dependence of the motion of a legged robot on the compaction of granular media," in preparation.

[10] [Online]. Available: http://www.gnu.org/software/flex/

[11] [Online]. Available: http://www.gnu.org/software/bison/

[12] [Online]. Available: http://gcc.gnu.org/onlinedocs/cpp/

[13] [Online]. Available: http://www.gnu.org/software/m4/

[14] J. Jenkins and M. Richman, *Arch. Rat. Mech. Anal.*, vol. 87, p. 355, 1985.

[15] R. Nedderman, *Statics and kinematics of granular materials*. Cambridge University Press, 1992.

[16] K. Terzaghi, *Theoretical soil mechanics*. Wiley, 1943.

[17] R. Albert, M. A. Pfeifer, A.-L. Barabási, and P. Schiffer, "Slow drag in a granular medium," *Phys. Rev. Lett*, pp. 205–208, 1999.

[18] R. Soller and S. Koehler, *Physical Review E*, vol. 74, p. 2006, Drag and lift on rotating vanes in granular beds.

[19] M. P. Ciamarra *et al.*, "Dynamics of drag and force distributions for projectile impact in a granular medium," *Phys. Rev. Lett.*, vol. 92, no. 19, 2004.

[20] D. I. Goldman *et al.*, "Lattice dynamics and melting of a nonequilibrium pattern," *Phys. Rev. Lett.*, vol. 90, p. 104302, 2003.

[21] C. S. O'Hern, S. A. Langer, A. J. Liu, and S. R. Nagel, "Random packings of frictionless particles," *Physical Review Letters*, vol. 88, p. 075507, 2002.

[22] C. Bizon *et al.*, "Patterns in 3-dimensional vertically oscillated granular layers: Simulation and experiment," *Phys. Rev. Lett.*, vol. 80, 1998.

[23] D. C. Rapaport, *The art of molecular dynamics simulation*. Cambridge University Press, 2004.

[24] M. A. Ambroso, R. D. Kamien, and D. J. Durian, "Dynamics of shallow impact cratering," *Physical Review E*, vol. 72, p. 041305, 2005. [Online]. Available: doi:10.1103/PhysRevE.72.041305

[25] D. I. Goldman and P. Umbanhowar, "Scaling and dynamics of sphere and disk impact into granular media," *Physical Review E*, vol. 77, no. 2, 2008.