10th ESA Workshop on
Advanced Space Technologies for
Robotics and Automation

'ASTRA 2008'

11-13 November 2008 at ESTEC,
Noordwijk, the Netherlands

A CONTACT DYNAMICS SIMULATION FRAMEWORK FOR ROBOTICS

Yves Gonthier and **Christian Lange**

Canadian Space Agency, Space Technologies
6767 route de l'Aéroport, Saint-Hubert, Quebec, Canada J3Y 8Y9
{Yves.Gonthier|Christian.Lange}@space.gc.ca

1. Introduction

Contact dynamics simulations are of great importance for the planning, real-time support and analysis of most missions involving the operation of the robotic components of the Mobile Servicing System, Canada's contribution to the International Space Station. In particular, the CSA is concerned with producing accurate and reliable simulations of tele-operated manipulators to insert payloads into specially designed worksites, and ensuring that the operations are performed smoothly and without causing damage. To allow for an efficient and easy to use end-to-end work flow --- including modelling, parameter identification, simulation, and analysis --- CSA's Space Technologies has developed a contact dynamics modelling framework for its in-house multibody dynamics modelling and simulation toolbox MuT.

The contact dynamics simulation framework has already been successfully applied to a wide spectrum of cases, ranging from academic examples (e.g., Tippe-Top) to flight-hardware, i.e., Orbital Replacement Units (ORU) like Arm Computer Units and Battery Boxes. Future work will include investigation to facilitate the analytical and experimental parameter identification process as well as the exploitation of parallel processing to speed up the simulation time.

2. Contact Model Framework

The object-oriented framework written in C++ facilitates the creation of contact models based on compliance. The framework provides a standardized structure to implement the components of a contact model, such that they can be easily combined to create specialized and complex models. It includes a standard C-code interface function for the Simulink simulation environment from The MathWorks, Inc. Furthermore, it features utilities to automatically compute all model input quantities and map the model force outputs to the respective body frames allowing the user to focus on the specific details of the contact model.

The components of the contact model are: the geometric-pair component and the force model component(s). Adding the contact model components to a standard contact model container class (CMC) creates the contact models. Each component is designed as a stand-alone software object. The geometric-pair component holds all of the information regarding the geometries of the simulated objects, and includes standard methods to query this information. It also provides methods to apply the forces and torques computed by the force model components to the objects. The force model components fall into two categories: the normal contact models and the friction models.

Figure 1 illustrates how the standard contact model container class interacts with a numerical simulation environment. The kinematics information (position, orientation and velocities) of the colliding objects is passed on to the CMC (shown in grey). This kinematics information corresponds to the motion of a body-fixed frame attached to each body. Internally, the CMC calls the standard methods of the geometric component to calculate the location of the contact site, the direction of the constraint force (contact normal), the compliance metrics (e.g. penetration depth) and the relative velocities of the bodies at the contact site location.

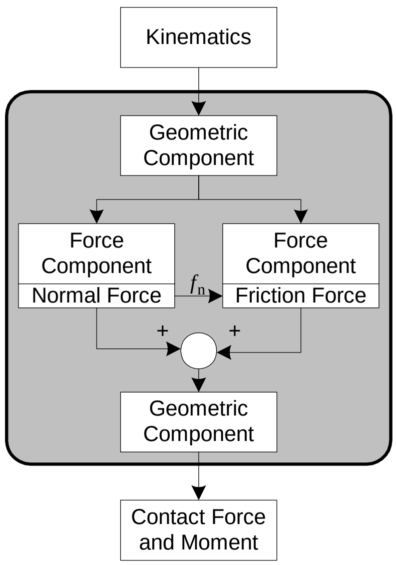

Figure 1. The standard contact model container class.

The compliance metrics and the relative normal velocity information are passed on to the normal force component. This latter component is not aware of the contact site location or even of the direction of the constraint force: it only computes the magnitude f_n of the constraint force. This constraint force is obtained using some model for the force needed to keep to two bodies apart, and may include a damping model to dissipate energy during impact [2]. The magnitude of the normal contact force and the relative tangential body velocities are fed to the tangential friction component, which computes a resulting load-dependant friction force [3]. Again, this force component is also not aware of the location of the contact site; it must determine the resulting friction force using exclusively the provided information. Finally, the geometric component, which *is* aware of the location of the contact site, applies the resulting total contact force and moments to the body-fixed frame of each body.

2.1 Contact model components

All contact model components are defined with a standard interface and, hence, they can be combined in different ways to create a variety of contact models. The user is not restricted to changing the contact model settings or parameters; the contact model itself can be changed. Each contact model includes one geometric component and one or two force model components. Any force component can be used in conjunction with any geometric component, thus allowing a large variety of contact models to be created. The process of combining the contact model components is straightforward and can be accomplished by writing a few lines of C pre-processor instructions.

Each contact model component includes a set of component-specific and user-selected constant parameters, e.g., a sphere radius, a spring stiffness constant. Standardized methods are provided to set and read these values. In addition the force model components can have the following attributes:

- States,
- Internal parameters.

The internal parameters are used to store contact model specific information, e.g., the velocity at the time of impact. This basic set of attributes is sufficient to allow the implementation of a many different contact model components.

2.2 Geometric pair component

The geometric component holds all of the information regarding the geometries of the simulated objects and includes standard methods to query this information. The geometric information is always specific to the geometry of a *pair* of object, e.g. a sphere and a cylinder. However, the geometric component itself can be composed of a combination of many sub-geometric pairs. In this case, the CMC must loop through the list of geometric pairs that are part of the geometric component to obtain the corresponding inputs to the force components. The resulting forces and moments are then combined such that the overall force is applied at the respective frames of each body. This latter feature of the contact model framework allows the modeling of contacts between objects with complex geometries, such as the battery box shown in Figure 2.

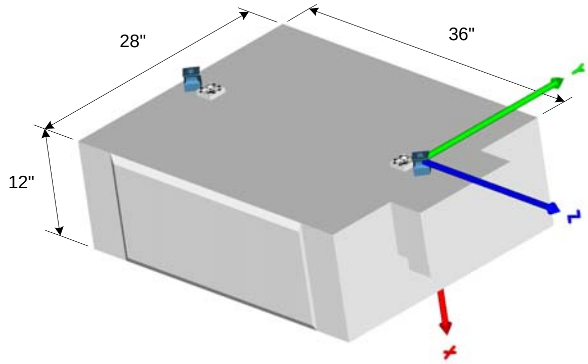

Figure 2. A battery box ORU.

Defining the standards methods providing the geometric information related to a specific geometric pair is a tricky problem. Although the algorithms for the methods can be quite simple in the case of simple geometric pairs, e.g. a sphere and an infinite plane, they can become tediously complex and unmanageable for more complicated cases. For example, a geometric pair algorithm describing the relationship between a sphere and a cube must include sub-cases for the sphere-face, the sphere-edge and the sphere-corner interfaces. Furthermore, the implementation must ensure that the transitions between the sub-cases are smooth and continuous, and in particular, that the algorithms has no degenerate cases. For example, a geometric pair algorithm defining the contact point location between two edges may select a point on the line that is the common perpendicular to the lines defining theses edges. When these lines become parallel, the result of the algorithm is undefined. On the other hand, these geometry-specific algorithms result in a very efficient code, and hence are suitable for real-time simulation.

To deal with a more general class of geometries an alternative approach is needed. The next section introduces a new contact modelling approach based on volume metrics. The key feature of this new approach is that the standard methods of the geometric pair components are defined in terms of volumetric properties, which can always be computed for *any* geometric pair, and hence, the model can be used as a general-purpose tool for modelling contacts.

3. Volumetric Contact Model

Traditionally, contact force interactions are predicted using one of the three following methods. On one hand, the colliding bodies can be assumed to be infinitely stiff. The associated rigid body model must be used in conjunction with an impact hypotheses, such as Newton's, Poisson's, and Stronge's [4,5]. Hereby, the equations of motion are derived by balancing the system's momenta before and after the impact, i.e., without explicitly considering contact compliances. This allows for great simplifications since changes in velocity become instantaneous, and can be calculated without integrating accelerations over the contact period. However, [6] shows that the contact stiffness and the duration of the impact phase greatly influences the stability of robotic systems under discrete-time control. Hence, rigid body models provide an inadequate reference for validating the performance of robotic systems in contact.

In contrast, penalty methods describe the rate-dependent normal and tangential compliance relations over time, and are usually based on Hertz theory. These second methods rely on elastic half-space theory to find contact stiffness properties of the bodies. Here, the stiffness is directly related to the interfering geometries and to the material properties of the bodies. A nonlinear spring model is then used to calculate the magnitude of the contact forces throughout the contact phase. However, the solution of the Boussniesq integral can only be found in the cases where the geometries are simple, and the area of the contact patch is assumed to be small in comparison to the size of the bodies.

The third option is to rely on continuum models, which model the mechanical interaction of the bodies in detail as a function of the material properties. These models are often implemented using Finite Element Methods (FEM) [7], and their application is restricted to non real-time simulation because of the large computational overhead associated with FEM. The body geometries are transformed into a mesh, and the FEM computes the deformation of each element of the mesh. Obviously, the accuracy of the prediction is directly related to the meshing pattern used and to the resulting number of elements.

The model proposed here uses another method that could be considered ``in-between'' the latter two options: the *volumetric* approach. The volumetric contact model is derived from first principles assuming a simplified mechanical behaviour of the materials: the Winkler elastic foundation model [8,9]. In [9], the geometries are assumed to be composed of polygons, and the global body-to-body contact force is obtained by numerically summing the force contribution from each contacting polygon. The local polygon force is readily found as a function of the polygon area and the inter-penetration depth.

In [10], the contact force expression is derived by defining an expression for the contact pressure as a function of the body geometries. The latter is integrated over the contact area, and the overall body-to-body force is obtained. It was shown that this contact force could be expressed in terms of the volumetric properties of the *volume of interference* between the two bodies, defined as the volume spanned by the intersection of the two un-deformed geometries of the colliding bodies. The properties of interest are: the volume of the volume of interference, the position of its centroid, and the inertia tensor about the centroid.

The volumetric approach is in fact a continuum model by nature, i.e., the effect of each infinitesimal element of the contact surface is taken into account in the overall contact force prediction. However, in its implementation, the volumetric contact model becomes a penalty model: the model corresponds to a nonlinear spring whose force is proportional to the volume of the volume of interference. Since the volumetric contact model is based on volumetric properties and because these can always be calculated, the model does not have any degenerate condition, i.e. it *always* works.

The volume metrics are obtained either analytically or numerically. In the former case, because the mathematical framework is simple, closed-form solutions can be found for a larger range of geometries then with the elastic half-space theory. For the latter case, the volume metrics are found by decomposing the volume of interference into small cubes called *voxels*, and extracting the volume metrics from these simple primitives. It is worthwhile to note that the discretization is not performed *a-priori* as with FEM, but is done on-line, with a dynamically selected resolution. Furthermore, the processed geometries are not restricted to have polygonal descriptions, and can be described using an exact mathematical representation [11], e.g, where a sphere is perfectly round.

The solid modelling technology from Parallel Geometry Inc.[1] (LLG) is based on a pure mathematical representation of geometry and was used for the geometrical modelling of the geometric pairs. The algorithms used to extract the volume metrics use a recursive octree decomposition of the volume of interference, and, hence, are highly suitable for parallel processing by nature. Consequently, the computer implementation of the simulation can benefit from the huge computational power available in the latest high performance computing platforms, which are based on parallel multi-core architectures.

[1] www.llgeometry.com

4. Simulation Results

The proposed framework can be used to create many different contact models. It was design to be compatible with any numeric simulation environment. A Simulink block implementing the volumetric contact model discussed in the previous section was prepared using the proposed framework. This section presents a few contact dynamics simulation examples.

4.1. A ball falling inside a cylinder

A typical Simulink diagram implementing a contact dynamics simulation is presented in Figure 3. The block on the top computes the dynamics of a free-floating object falling under the action of gravity. The block in the bottom is the contact model. The simulation computes the motion of a ball falling on the inside surface of a cylinder; see Figure 4. Notice that the curved surfaces are perfectly smooth, without flat segments; this is a feature of the LLG solid modelling technology that uses an exact mathematical description of geometry — curved surfaces are represented as is, without resorting to polygons.

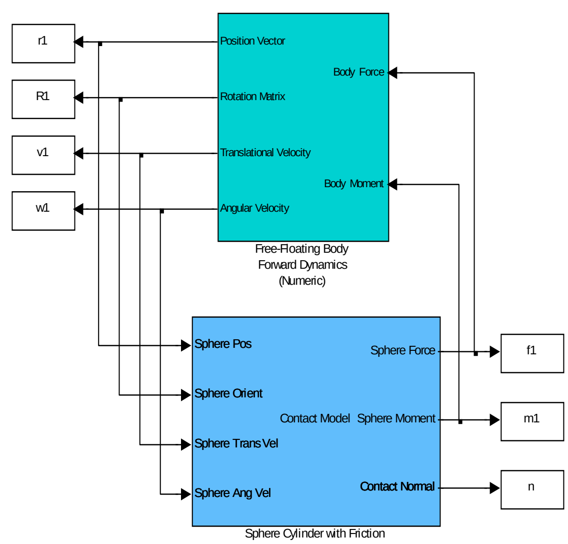

Figure 3. A Simulink diagram of a contact dynamics simulation.

The contact model takes as input the position, orientation and velocity of the ball and computes the resulting force and moment applied to it. The motion of the center of the ball is presented in Figure 5. It projection ($z = 0$) on the x-y plane is shown below. The contact model also features rolling resistance and spinning friction torques, such that the ball eventually comes to rest.

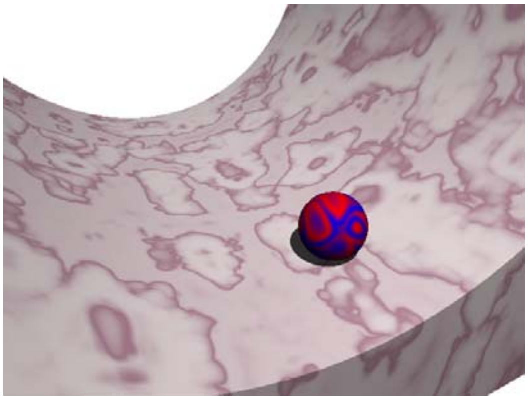

Figure 4. LLG rendering of the ball rolling on the inside surface of the cylinder.

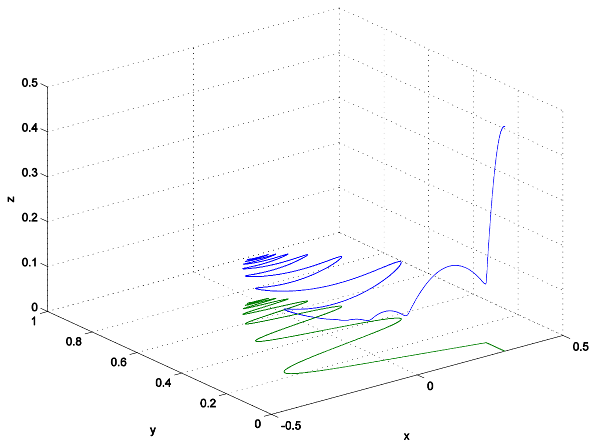

Figure 5. The ball center point motion.

4.2. A tippe-top simulation

The tippe-top simulation from [12] was implemented using the same blocks as in the previous section, but this time using the geometric model of a tippe-top; see Figure 6. The tippe-top is a special kind of top that, when spun properly, flips over while it is spinning. Figure 6 shows the top in its statically stable configuration, i.e. "un-flipped".

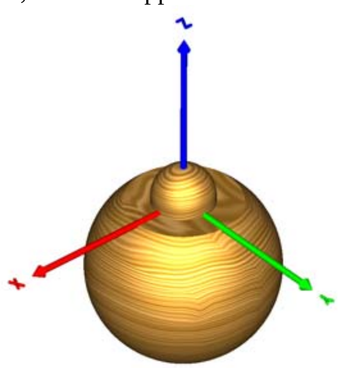

Figure 6. A Tippe-top model.

The center of mass is located below the center of the larger sphere. The flipping phenomenon is the results of a complex interaction between inertial and the contact forces. In particular, the frictional forces play an important role in the flipping motion. Here, because the top is spinning rapidly about its axis of symmetry, the tangential friction force becomes negligible. This effect was first observed by P. Contensou in 1963.

The tilt angle of the top as a function of time is given in Figure 7. The top initially has a small angle with respect to the *z*-axis and first starts tilting back up. It becomes straight up (0 rad inclination) at 0.15 s and then starts titling down. At 1.7 s the small sphere touches the ground and the top flips upside down. It settles into a stable spinning motion but gradually looses it angular velocity. At around 4 s, the top no longer has enough angular velocity to keep its center of mass in a statically unstable configuration, so it falls down and eventually stops moving.

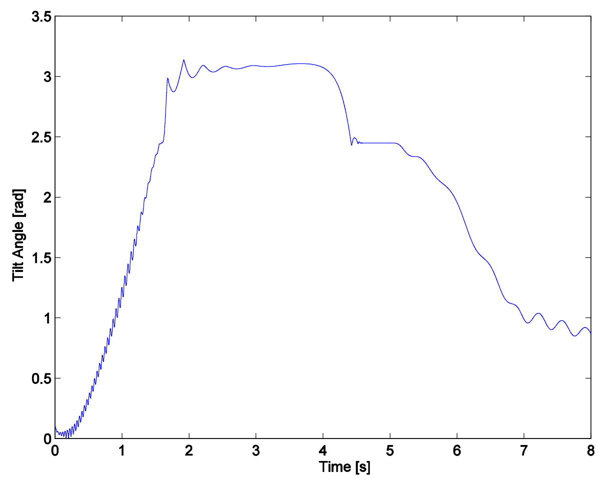

Figure 7. The tippe-top flip motion.

4.3. A battery box falling into its worksite

A simulation of the battery box presented in Figure 2 falling into its worksite (see Figure 8) was then implemented. This simulation scenario involves objects with complex geometries and makes use of the multi-geometric pair feature of the geometric component of the framework; see Section2.2. As the battery box falls, it simultaneously touches the worksite at many locations and the contact model correctly processes these multiple contact sites. Images depicting the motion of the battery box as it falls into its worksite are shown in Figure 9. The image sequence is given with time increments of 0.2 s.

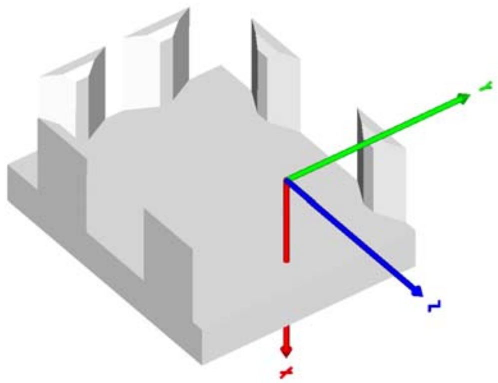

Figure 8. The worksite of the battery box.

5. Conclusion

This paper introduced an object-oriented framework written in C++ that facilitates the creation of contact models based on compliance. The framework provides a standardized structure to implement the components of a contact model, such that they can be easily combined to create specialized and complex models. Contact model are created by adding contact model components into a standard contact model container class. This class also includes facilities to interface with numerical simulation environments, such that the development effort only needs to be focused on the model components. The contact model can have one geometric component and one or two force model components.

The geometric components can be composed of multiple sub-geometric pairs, such that the contact model can handle complex objects. The geometric component methods can be easily written for objects with simple geometries, but are tedious to implement for more complicated cases. A volumetric contact model is then presented. This new contact model can deal in a general manner with any geometry because it uses information about the volume of interference between the objects, and this information is always readily obtainable in all cases.

A Simulink block implementing the volumetric contact model was prepared using the proposed framework. Contact dynamics simulation results were presented for a ball falling into a the inside surface of a cylinder, a tippe-top flipping upside down under the combined effects of the inertial forces and the contact forces, and finally a battery box falling into its worksite. This latter simulation demonstrated the capability of the contact model to handle objects with complex geometries.

6. References

[1] J.-C. Piedbœuf, F. Aghili, M. Doyon, Y. Gonthier, E. Martin, and W.-H. Zhu, “Emulation of space robot through hardware-in-the-loop simulation”, *In Proc. Of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: iSAIRAS*, Canadian Space Agency, St-Hubert, Quebec, Canada, June18–22, 2001.

[2] K. H. Hunt and F. R. E. Crossley, “Coefficient of Restitution Interpreted as Damping in Vibroimpact”, *Transactions of the ASME Journal of Applied Mechanics*, 1975, Vol 42 , pp. 440-445, Series E.

[3] Y. Gonthier, J. McPhee, C. Lange and J.-C. Piedbœuf, “A Regularized Contact Model with Asymmetric Damping and Dwell-Time Dependant Friction”, *Multibody System Dynamics*, Kluwer, 2004, Vol. 11, No. 3, pp. 209-233,

[4] W. J. Stronge, *Impact Mechanics*, Cambridge University Press, 2000.

[5] C. Glocker, *Set-Valued Force Laws — Dynamics of Non-smooth Systems*, Springer, Berlin, 2001.

[6] J. Kövecses, L.L. Kovács and G. Stéphán, “Dynamics modelling and stability of robotic systems with discrete-time force control”, *Archive of Applied Mechanics (Ingenieur Archive)*, 2007, Vol. 77, No. 7, pp. 293–299.

[7] M. A. Puso and T.A. Laursen, “A mortar segment-to-segment frictional contact method for large deformations”, *Computer Methods in Applied Mechanics and Engineering*, 2004, Vol. 193, pp.4891–4913.

[8] K.L. Johnson, *Contact Mechanics*, Cambridge University Press, London, 1985.

[9] G. Hippmann. “An algorithm for compliant contact between complexly shaped bodies”, *Multibody System Dynamics*, 2004, Vol. 12, No. 4, pp.:345–362.

[10] Y. Gonthier, *Contact Dynamics Modelling for Robotic Task Simulation*, PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2007

[11] J.-F. Rotgé and J. Farret, “Universal solid 3D format for high performance urban simulation”, *Proc. of the IEEE Urban Remote Sensing Joint Event*, Paris, France, April 11–13, 2007.

[12] R.I. Leine and C. Glocker, “A set-valued force law for spatial Coulomb-Contensou friction”, *European Journal of Mechanics*, 2003, Vol. 22, No. 2, pp. 193–216.

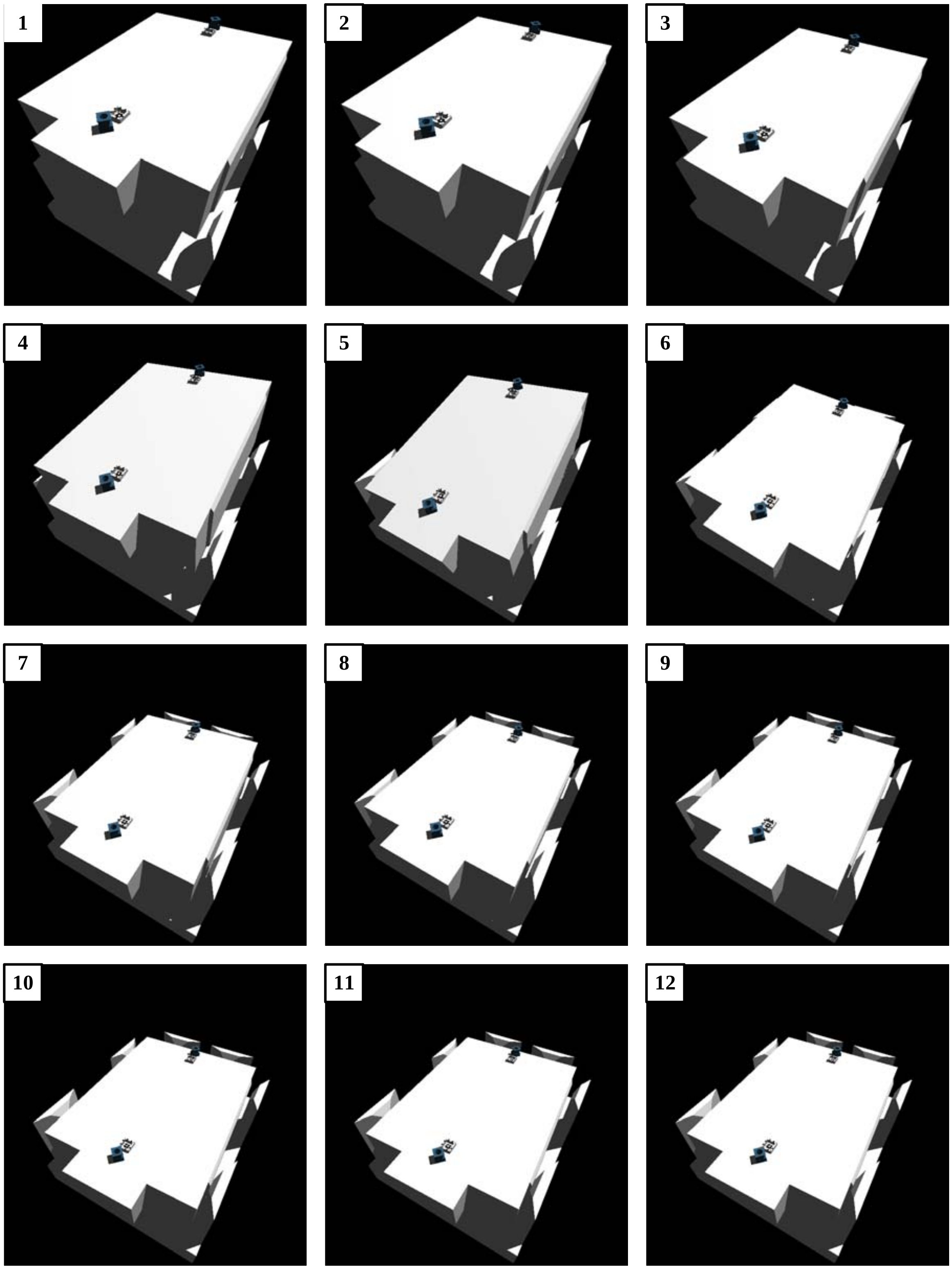

Figure 9. The battery box falling into its worksite.