

Digital Elevation Map Reconstruction for Port-based Dynamic Simulation of Contacts on Irregular Surfaces

Martin Wassink*, Raffaella Carloni*, Pantelis Poulakis** and Stefano Stramigioli*

Abstract—This paper presents a method to utilize a port-based multibody contact model for simulating dynamic interaction between irregular surfaces. The existing compliant contact model requires an analytic parametrization of the surfaces involved in the interaction, the definition of a Gauss frame in each of the contact points and initialization of the candidate contact points. The authors intend to apply this contact model for cases in which the surfaces of the 3D interacting bodies can not be described (easily) by a geometrically defined surface. Such surfaces are often represented by 3D point meshes. This implies that the surfaces of the bodies have to be reconstructed from 3D point meshes where initialization of the candidate contact points is not arbitrary.

This paper proposes the reconstruction of such surfaces by means of polynomial interpolation. With the aim of having a computationally efficient simulator, the surface interpolation is restricted to small patches around the candidate contact points. To select these candidates, the proposed simulation approach is completed by using a fast (existing) collision detection algorithm. Simulation results are validating the effectiveness of the proposed modeling and simulation approach.

I. INTRODUCTION

Modeling contact is a core issue encountered in the simulation of multibody dynamics. Significant work has been carried out by the scientific community for the formulation of models that capture the compliance and the shape of the contact in several simulation scenarios and especially for leg-terrain or wheel-soil interaction in walking or vehicle motion respectively, as well as in haptics and robotic manipulation and grasping.

In both regular and generalized contact kinematic approaches appearing in the literature, two aspects of major importance for the applicability and the consistency of the contact model have been identified. Firstly, the type of surfaces that can be handled. The way to represent an arbitrary surface in a computer aided design (CAD) tool is via 3D polygonal meshes or 3D point meshes, for which it is almost impossible to have the surfaces fully analytically defined. Thus contact algorithms needing a complete analytical representation of a surface, e.g. [5], [6], [9], [1] among others, are restricted to handling trivial and easily parameterizable surfaces, like spheres, cones, cylinders or

ellipsoids. Secondly, the approach taken on the application of contact dynamics. Following the discussions in [9], [1], the authors believe that the most physically consistent way to apply dynamics on a simulated contact is to track the evolution of the contact points on a geometrically defined surface. In this approach the contact points are derived based on the kinematics and the relative motion of the bodies. In fact, the use of only collision detection algorithms on polygonal surfaces may induce discontinuities in the proper calculation of the contact and, moreover, it requires the a priori graphical definition of certain parameters, e.g. penetration depth.

This paper intends to start from the theory in [9], [1] for the modeling of the kinematics and the dynamics of compliant contacts between relatively convex objects described by geometrically defined surfaces and to extend it to handle geometrically undefined surfaces, described by digital elevation maps (DEMs). Motivation of this choice is that DEMs are widely used as a graphic primitive for the description of really wide scenes, like in the case of ground surface topography and terrains in locomotion and of non-regular object surfaces in grasping. The use of points as a graphic primitive reduces the object to the essentials so that the representation is greatly optimized [4]. However, it provides a mere discretization of geometry without an explicit storage of topology information. Therefore, geometry processing and feature detection methods encourage the use of 3D point meshes that meet the right balance between descriptive power and computational costs.

Following this discussion, the authors aim to extend the compliant contact model presented in [1] and to implement it in simulation so that it is able to deal with 3D scenes of great dimension described through DEMs. Since smooth and geometrically defined surfaces are needed for the aforementioned contact model, and at the same time, the advantages of having an essential description of irregular surfaces is highly desirable, this paper proposes to reconstruct smooth and geometrically defined portions of the surface, named patches [7], [3], [13], [11], [12]. Patches are created by interpolating only a subset of the points of the DEM in the vicinity of the contact points. In this way, the patches preserve the properties of the real scene or of some specific features. They are built with low computational cost while the evolution of contact is dynamically tracked over patches by the port-based contact model. Furthermore, the usage of patches allows to simultaneously instantiate multiple contact point pairs between two objects on different parts of their surfaces.

This work has been carried out as part of the FALCON project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Embedded Systems Institute (BSIK03021) program.

*{m.wassink,r.carloni,s.stramigioli}@utwente.nl, IMPACT Institute, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands.

**pantelis.poulakis@esa.int, European Space Research and Technology Centre, European Space Agency, P.O. Box 299, 2200AG, Noordwijk, The Netherlands.

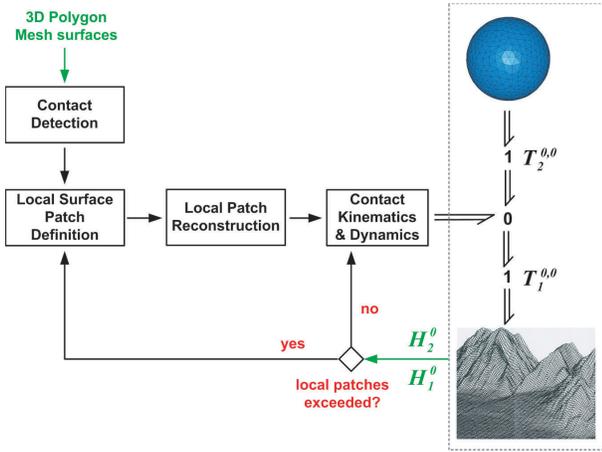


Fig. 1. Schematic of the proposed algorithm for the case of two objects. The 3D mesh representation of the objects is fed to the collision detection block. After the contact points are detected, patch areas are defined around them. Then, locally, on each of the defined patch areas, analytic smooth surfaces are reconstructed. As the contact evolves, the patches are created and destroyed following the motion.

This paper is organized as follows. Section II discusses the implementation of the contact model together with the proposed DEM reconstruction method in a general simulation scheme. Thereafter, Section III presents a simple but efficient patch interpolation method. Section IV describes the validation of the proposed DEM reconstruction algorithm. The resulting proof of concept is discussed in Section V, which includes a reference to the attached video. Conclusions and future work directions finalize this paper.

II. SIMULATION SCHEME

This section presents an overview of the implementation of the proposed simulation scheme. Fig. 1 highlights its general structure.

A. Contact Kinematics and Dynamics: a port-based model

The *Contact Kinematics and Dynamics* block of Fig. 1 represents the port-based contact model developed in [1], which is briefly summarized here. The contact model kinematics do not only consider the velocities of the two coincident points on the two bodies that have zero distance, but also look at the kinematics of the points on the two surfaces which have the shortest Euclidean distance. In this way, candidate contact points are tracked before they actually get into contact. The contact model requires the definition of a local C^2 continuous parametrization $f(u, v)$ of the surfaces that are going to get in contact, together with its derivative mapping $f_* = \begin{pmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \end{pmatrix}$. If the two bodies are relatively convex, then the velocities of the contact points are uniquely determined as function of the derivative mapping g_* of the Gauss map g of the surfaces. Note that $g = \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}$ and $g_* = \begin{pmatrix} \frac{\partial g}{\partial u} & \frac{\partial g}{\partial v} \end{pmatrix}$.

As depicted in the right part of Fig. 1, via the 0-junction, the *Contact Kinematics and Dynamics* block obtains the relative twist expressed in the world reference frame for each pair of contact points. Inside the *Contact Kinematics and*

Dynamics block, a coordinate transformation is performed to express the twist in the Gauss frame of one of the objects. The contact dynamics are calculated from this twist. Hence, the interaction forces are injected directly between the contact points. For more details on the contact model itself, the reader is referred to [1], [8], [9], [10].

B. Local Surface Patch Definition and Reconstruction

The *Contact Kinematics and Dynamics* block requires smooth, well defined surfaces to track the evolution of the contact consistently. However, reconstructing large surfaces is computationally inefficient. This problem is tackled by introducing a *Local Surface Patch Definition* module, which is responsible for restricting the surface reconstruction to a small area (i.e. a patch) of the DEM around the detected contact point. As the contact evolves, new patch areas are defined in order to follow the motion. The parameters of the patch area definition algorithm are tuned to the needs of the application.

Once the patch area has been defined, the *Local Patch Reconstruction* block is in charge of interpolating each instantiated patch in run-time, deriving a well defined, smooth (C^2 continuous), repeatable, local parametrization $f(u, v)$ of the surface.

These two blocks are the focus of the paper. In fact, the main contribution of this work is the determination of a smooth and fast interpolation method for surface reconstruction for the application of dynamic port-based contact modeling. The contact model can be used as long as the combination of the patch surface and the object surface results in relatively convex contacts locally on the patch [1]. Hence, the patch itself is not necessarily absolute convex.

C. Contact Detection

For the detection and initialization of the candidate contact points, a computational geometry algorithm is incorporated, specifically designed for collision detection in computer graphics applications. The original DEM description of the surfaces as 3D meshes is utilized to let the algorithm find the candidate contact points, enhancing the generic applicability of the algorithm and providing maximum computational efficiency. Once the patches are defined and reconstructed, it is possible to compute the exact contact points on the smooth surfaces.

In this work, the SWIFT++ computational geometry library [2] was chosen for the first tests. SWIFT++ efficiently implements proximity queries between polyhedral objects using convex surface decomposition. The SWIFT++ algorithms have been encapsulated in the *Contact Detection* block. The 3D meshes are used to load a 3D triangulated surface description of the objects during the simulation preprocessing phase. During run-time, the algorithm selects candidate contact points, based on selected thresholds. Their coordinates are propagated to the *Local Surface Patch Definition* module.

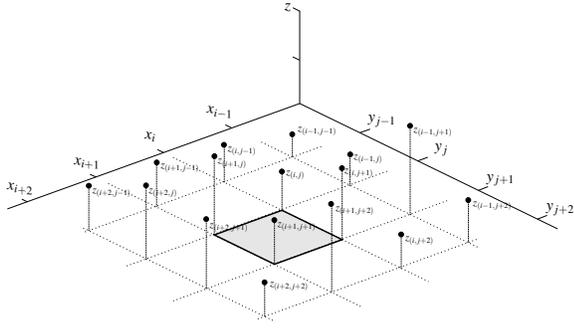


Fig. 2. Digital Elevation Map (DEM). On top of a uniform xy -grid with grid size Δ , the DEM gives an altitude value z_{ij} for each discrete x_i, y_j -pair.

III. PATCH INTERPOLATION METHOD

This section describes a method for the construction of a patch which realizes the interpolating surface on a portion of a DEM. This scenario is depicted in Fig. 2.

The DEM is assumed to be based on a uniform $n \times k$ -grid of (x_i, y_j) pairs with $i \in [0, 1, \dots, n]$, $j \in [0, 1, \dots, k]$ with grid size Δ and to have an unique altitude per grid point. Note that the grid size determines the maximum curvature that is taken into account for the interacting 3D bodies.

The goal is to find a smooth interpolating function describing the surface altitude (z) as a function of x and y , i.e. $z = F(x, y)$. To do so, the surface above the whole grid is split into patches $f_{ij}(x, y)$ such that $F(x, y) = \bigcup f_{ij}(x, y)$, $\forall i, j$, where $f_{ij}(x, y)$ describes the surface for the particular patch as function of $x \in [x_i, x_{i+1}]$ and $y \in [y_j, y_{j+1}]$.

To make sure that $F(x, y)$ is smooth over the whole surface, the patches need to be constructed such that their interconnection is C^2 continuous. For reasons of readability, the remainder will show the construction of C^1 continuous patches. It will be shown how to easily extend to C^2 continuity.

A. Polynomial Interpolated Line Segment

First, a line interpolation algorithm is introduced as the basis for the rest of the construction of $f_{ij}(x, y)$. A third order polynomial is used to create a line segment $f_j(x)$ through the points $z(i, j)$ and $z(i+1, j)$, with

$$f_j(x) = a_j + b_j x + c_j x^2 + d_j x^3 \quad \forall x \in [x_i, x_{i+1}]. \quad (1)$$

The parameter vector $\bar{a}_j = (a_j, b_j, c_j, d_j)^T$ is found by setting four constraints and solving the system. Two constraints are of course the two points to interpolate through, i.e. $(x_i, z(i, j))$ and $(x_{i+1}, z(i+1, j))$. The other two constraints are the derivatives in these points, which are chosen to be equal to the average linear slopes around the points $(x_i$ and $x_{i+1})$, such that

$$\frac{\partial f_j}{\partial x}(x_i) = \frac{1}{2\Delta}(z(i+1, j) - z(i-1, j))$$

$$\frac{\partial f_j}{\partial x}(x_{i+1}) = \frac{1}{2\Delta}(z(i+2, j) - z(i, j)).$$

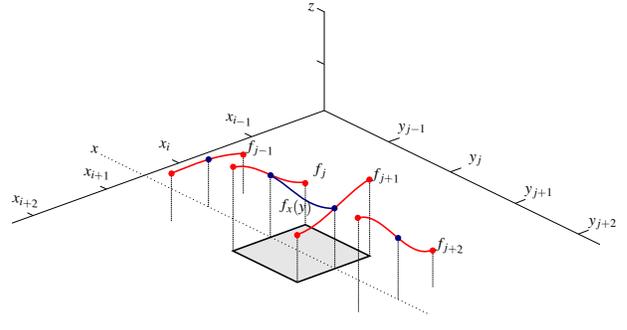


Fig. 3. Patch construction: four constructed line segments $f_j(x) \forall x \in [x_i, x_{i+1}]$ are used to construct a fifth parametrized line segment $f_x(y)$ at some position $x \in [x_i, x_{i+1}]$, for which the parameter vector is actually a function of x , s.t. $f_x(y) = f_{ij}(x, y)$, which represents the patch surface.

These explicit constraints assure smooth interconnection of neighboring line segments. The parameter vector \bar{a}_j is found to be

$$\bar{a}_j = X^{-1} \cdot \bar{z}_j, \quad (2)$$

where

$$X = \begin{pmatrix} 1 & x_i & x_i^2 & x_i^3 \\ 1 & x_{i+1} & x_{i+1}^2 & x_{i+1}^3 \\ 0 & 1 & 2x_i & 3x_i^2 \\ 0 & 1 & 2x_{i+1} & 3x_{i+1}^2 \end{pmatrix}$$

$$\bar{z}_j = \begin{pmatrix} z(i, j) \\ z(i+1, j) \\ \frac{\partial f_j}{\partial x}(x_i) \\ \frac{\partial f_j}{\partial x}(x_{i+1}) \end{pmatrix} = \begin{pmatrix} z(i, j) \\ z(i+1, j) \\ \frac{1}{2\Delta}(z(i+1, j) - z(i-1, j)) \\ \frac{1}{2\Delta}(z(i+2, j) - z(i, j)) \end{pmatrix}.$$

Hence, to create a line segment between two DEM points, two extra surrounding DEM points are needed.

For the construction of a full patch, Eq. 2 is used to construct four line segments, i.e. $f_{j-1}(x)$, $f_j(x)$, $f_{j+1}(x)$ and $f_{j+2}(x)$ with $x \in [x_i, x_{i+1}]$ (see Fig. 3). This gives four parameter vectors: $\bar{a}_{j-1}, \bar{a}_j, \bar{a}_{j+1}, \bar{a}_{j+2}$.

B. Polynomial Patch

For the complete patch, another line segment is constructed at some position $x \in [x_i, x_{i+1}]$ as function of $y \in [y_j, y_{j+1}]$, i.e. $f_x(y)$ as depicted in Fig. 3. The four constraints are found to be

$$\begin{aligned} f_x(y_j) &= f_j(x) = \bar{x} \cdot \bar{a}_j \\ f_x(y_{j+1}) &= f_{j+1}(x) = \bar{x} \cdot \bar{a}_{j+1} \\ \frac{\partial f_x}{\partial y}(y_j) &= \frac{1}{2\Delta}(f_{j+1}(x) - f_{j-1}(x)) \\ &= \frac{1}{2\Delta} \cdot \bar{x}(\bar{a}_{j+1} - \bar{a}_{j-1}) \\ \frac{\partial f_x}{\partial y}(y_{j+1}) &= \frac{1}{2\Delta}(f_{j+2}(x) - f_j(x)) \\ &= \frac{1}{2\Delta} \cdot \bar{x}(\bar{a}_{j+2} - \bar{a}_j), \end{aligned}$$

where \bar{x} is a row vector based on some chosen position x , i.e. $\bar{x} = (1 \quad x \quad x^2 \quad x^3)$.

Following Eq. 1, $f_x(y)$ is written as

$$\begin{aligned} f_x(y) &= \alpha(x) + \beta(x)y + \gamma(x)y^2 + \delta(x)y^3 \\ &= \bar{y} \cdot \bar{\alpha}(x), \end{aligned} \quad (3)$$

where $\bar{y} = (1 \ y \ y^2 \ y^3)$ and the parameter vector $\bar{\alpha}(x) = (\alpha(x), \beta(x), \gamma(x), \delta(x))^T$. Hence, $\bar{\alpha}$ follows from

$$\begin{pmatrix} f_x(y_j) \\ f_x(y_{j+1}) \\ \frac{\partial f_x}{\partial y}(y_j) \\ \frac{\partial f_x}{\partial y}(y_{j+1}) \end{pmatrix} = Y \cdot \bar{\alpha}(x), \quad (4)$$

with

$$Y = \begin{pmatrix} 1 & y_j & y_j^2 & y_j^3 \\ 1 & y_{j+1} & y_{j+1}^2 & y_{j+1}^3 \\ 0 & 1 & 2y_j & 3y_j^2 \\ 0 & 1 & 2y_{j+1} & 3y_{j+1}^2 \end{pmatrix}.$$

Substituting the constraints gives the following equality

$$\begin{pmatrix} \bar{a}_j^T \\ \bar{a}_{j+1}^T \\ \frac{1}{2\Delta}(\bar{a}_{j+1} - \bar{a}_{j-1})^T \\ \frac{1}{2\Delta}(\bar{a}_{j+2} - \bar{a}_j)^T \end{pmatrix} \cdot \bar{x}^T = Y \cdot \bar{\alpha}(x)$$

and replacing the parameter vector \bar{a} by Eq. 2 results in

$$\begin{pmatrix} \alpha(x) \\ \beta(x) \\ \gamma(x) \\ \delta(x) \end{pmatrix} = Y^{-1} \cdot Z \cdot (X^{-1})^T \cdot \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \end{pmatrix}, \quad (5)$$

where

$$Z = \begin{pmatrix} \bar{z}_j^T \\ \bar{z}_{j+1}^T \\ \frac{1}{2\Delta}(\bar{z}_{j+1} - \bar{z}_{j-1})^T \\ \frac{1}{2\Delta}(\bar{z}_{j+2} - \bar{z}_j)^T \end{pmatrix}.$$

With $C_{ij} := Y^{-1} \cdot Z \cdot (X^{-1})^T$, it follows that

$$\bar{\alpha}(x) = C_{ij} \cdot \bar{x}^T,$$

such that Eq. 3 becomes

$$f_x(y) = \bar{y} \cdot \bar{\alpha}(x) = \bar{y} \cdot C_{ij} \cdot \bar{x}^T \quad \forall y \in [y_j, y_{j+1}].$$

Note that instead of choosing some $x \in [x_1, x_{i+1}]$, x can be a running variable on the domain of the patch. It follows that the full patch function f_{ij} is given by

$$f_{ij}(x, y) = \bar{y} \cdot C_{ij} \cdot \bar{x}^T \quad \forall x \times y \in [x_i, x_{i+1}] \times [y_j, y_{j+1}], \quad (6)$$

where the matrix C_{ij} is a constant, fully determined by the ij -part of the DEM that is under consideration for the patch f_{ij} . Fig. 4 shows some patches on an arbitrary DEM.

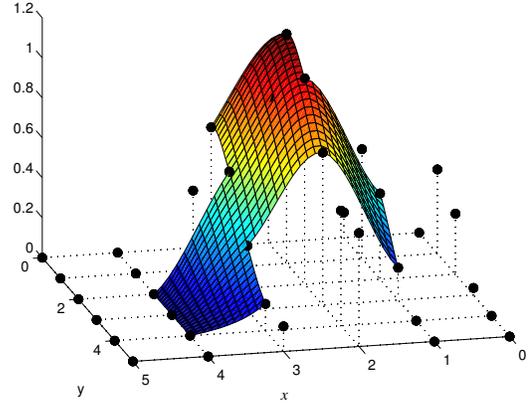


Fig. 4. Constructed example patches f_{ij} on an arbitrary DEM. Patches $f_{12}, f_{22}, f_{21}, f_{32}$ and f_{33} are shown.

C. Extend to C^2 Continuous Patches

The proposed method produces C^1 continuous patches by construction. By following the same procedure for a 5^{th} order polynomial and introducing two extra constraints, one can easily find C^2 continuous patches that can be applied for the port-based contact model described in [1]. The two extra constraints must represent the second order derivatives in the end-points of the DEM. A good choice would be to constrain these to

$$\begin{aligned} \frac{\partial^2 f_j}{\partial x^2}(x_i) &= \frac{1}{2\Delta}(z_{(i+1,j)} - z_{(i-1,j)}) - \frac{1}{\Delta}(z_{(i,j)} - z_{(i-1,j)}) \\ &= \frac{1}{2\Delta}(z_{(i+1,j)} - 2z_{(i,j)} + z_{(i-1,j)}), \end{aligned}$$

which represents an approximation of the curvature through the DEM point x_i .

D. Gauss Contact Frames

The parametrized analytic surface $f(u, v)$ for the contact model is found to be

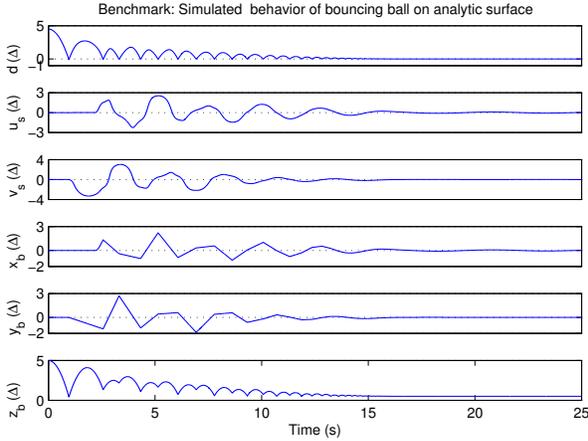
$$f(u, v) = \begin{pmatrix} u \\ v \\ f_{ij}(u, v) \end{pmatrix} \quad \forall u \times v \in [x_i, x_{i+1}] \times [y_j, y_{j+1}].$$

Also the derivative mapping f_* , from which the Gauss map g follows, can be easily computed as analytic expression:

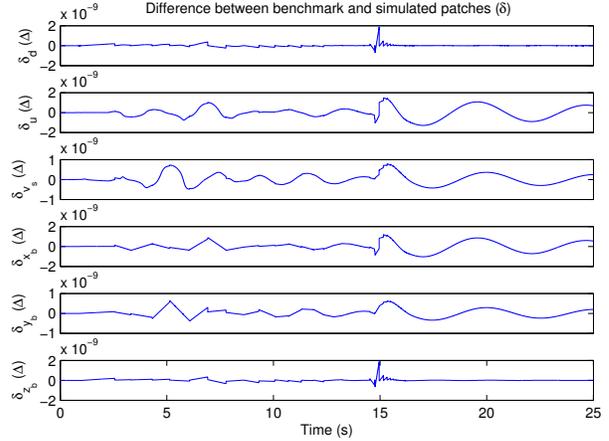
$$f_*(u, v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \frac{\partial f}{\partial u}(u, v) & \frac{\partial f}{\partial v}(u, v) \end{pmatrix},$$

where

$$\begin{aligned} \frac{\partial f}{\partial u}(u, v) &= (1 \ v \ v^2 \ v^3) \cdot C_{ij} \cdot \begin{pmatrix} 0 \\ 1 \\ 2u \\ 3u^2 \end{pmatrix} \\ \frac{\partial f}{\partial v}(u, v) &= (0 \ 1 \ 2v \ 3v^2) \cdot C_{ij} \cdot \begin{pmatrix} 1 \\ u \\ u^2 \\ u^3 \end{pmatrix}. \end{aligned}$$



(a) Simulation result of the benchmark simulation: a spherical ball bouncing on an analytically defined surface ($f(u, v) = (u, v, 0.2u^2 + 0.2v^2)^T$). Implemented as described in [1], using a compliant contact model with fairly stiff contact dynamics. The plotted variables verify the expected bouncing behavior.



(b) Difference between benchmark simulation and simulation with contact patches as given in Eq. 6. The small numerical errors validate the use of the proposed contact patches in combination with the port-based contact modeling approach in [1] when simulating non-analytic surfaces such as DEMs.

Fig. 5. Comparison results of validation tests based on a ball that bounces on a surface. The benchmark simulation uses an analytic function to describe the surface, while the patches are tested on a DEM which is a sampled version ($\Delta = 1$) of the same analytic function. The plots present the characteristic variables: d describes the minimal distance between sphere and surface, u_s, v_s are the coordinates of the candidate collision point on the surface and $(x, y, z)_b$ represent the 3D world coordinates of the ball position.

IV. VALIDATION METHOD

The proposed simulation scheme has been implemented in the simulation package 20-sim [14], as discussed in Section II.

An ad hoc test scenario was implemented to realize a proof of the concept presented in this paper. The test compares the simulation results of two different model implementations of the same test scenario. The test scenario represents a spherical ball that compliantly bounces in a bowl as depicted in Fig. 6(a). The bowl surface map is given by $f(u, v) = (u, v, 0.2u^2 + 0.2v^2)^T$. To have the ball bounce around the whole bowl, the ball starts with some initial rotation and gets disturbed by a force impulse after 2.2 seconds.

The benchmark simulation implements the model by using a geometrical description for both the sphere and the bowl surface. This is an exact implementation of the contact model as described in [1].

To validate the results of the proposed simulation method in this paper, the same test scenario is implemented by using the same geometrically defined sphere, while the bowl surface is now described by a DEM. This DEM is created by sampling the known surface map of the bowl with large grid size ($\Delta = 1$). During dynamic simulation, candidate collision points are selected through the SWIFT++ library and patches are constructed. The visco-elastic bond-graph contact model accepts the patches by receiving the candidate contact point and the geometrical equations of the map of the patch (f, f_*, g and g_*).

The same visco-elastic contact dynamics are implemented in both simulations. Hence, both simulations have the same contact kinematics and dynamics as in [1]. However, the benchmark simulation can do global tracking of the candidate contact points, while the validation simulation continuously

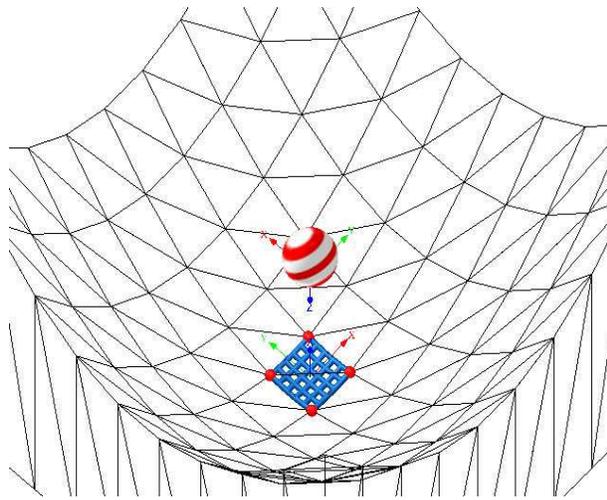
reads the DEM and renders local patches on which only local contact point tracking is possible. For the contact model, it does not matter whether the geometrical surface maps are only valid locally or globally, as long as the *Local Surface Patch Definition module* initializes the contact model correctly each time a new patch is instantiated.

V. RESULTS

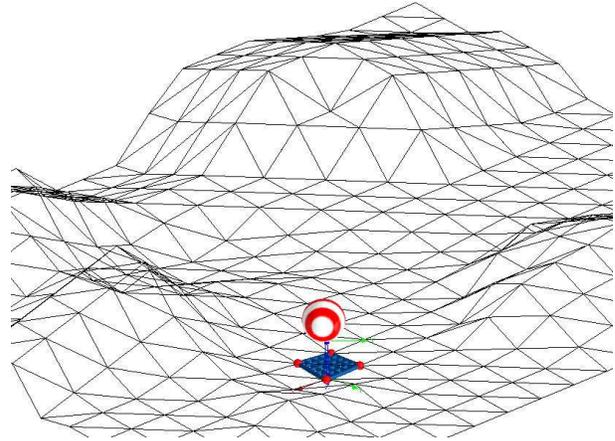
Fig. 5 shows the simulation results of the validation test. Some variables of the benchmark simulation are plotted in Fig. 5(a). From this plot it was verified that indeed the expected bouncing ball behavior was implemented, which can also be seen in the attached video. The difference between the benchmark simulation and the validation simulation, which uses the proposed method, is shown in Fig. 5(b). This comparison plot shows only minor numerical differences.

From this comparison plot (together with more simulation results), it is concluded that the proposed simulation model with patches for surface reconstruction of DEMs, gives useful and consistent results. Moreover, the computational costs (even if the dimension of the DEM is high) are considered low thanks to the simplicity of the method. To give an indication; in this validation test (using 20-Sim package on an ordinary modern pc), the benchmark simulation of 25 seconds simulation time uses 1.4 seconds to do 102,000 model calculations. The validation simulation adds 10 μs per model calculation for defining the patch area, reading the DEM and constructing the patch, while the SWIFT++ library adds another 100 μs per model calculation to select candidate contact points.

A simple DEM generated from an analytic map was used for validation reasons. However, in general, the simulation model has shown promising results in simulating contacts



(a) Validation scenario: a sphere bouncing on a digitized geometrically defined surface, represented by a DEM.



(b) Irregular surface simulation: a sphere bouncing on an irregular surface, represented by a DEM.

Fig. 6. Snapshots of the simulation scenarios. Although visualized by a wire frame, the DEM only consists of discrete altitudes stored in a separate file. The small grid beneath the sphere shows the currently instantiated patch.

with any surface generated from a DEM. An example is presented in the attached video, showing the simulation of a bouncing ball on a fully irregular DEM surface. Fig. 6(b) gives an impression of this simulation scenario.

It may be clear that each sub-model block, represented by *Contact Kinematics & Dynamics* in Fig. 1 together with its bond-graph structure, can track only one contact between two objects. However, multiple contacts can be modeled easily by repeating the bond-graph structure between the two objects, such that each structure generates its own set of contact forces.

VI. CONCLUSIONS

Starting from previous work on contact modeling [1], this work has extended the study of 3D multibody interaction simulation to interaction of objects with non-geometrically defined surfaces. A port-based simulation scheme has been proposed and implemented through bond-graph theory which cooperates with an external collision detection library.

Furthermore, an interpolation method for 3D point meshes, a DEM, was introduced that locally reconstructs the DEM surface by instantiating small patches around the candidate contact points. The usage of patches instead of whole surface reconstruction reduces computational costs and easily allows for multiple contact pair instantiation.

The combination of the port-based contact model, the collision detection library and the proposed surface reconstruction method proved to be useful for physically consistent modeling of contact dynamics in a computationally efficient way. In the video attached to this work, simulation results provide proof of concepts of the proposed reconstruction method in combination with the existing contact model.

VII. FUTURE WORK

Thanks to its modularity, the simulation scheme developed in this work can be extended with new features. Future work

will be in two directions. One direction involves further exploration of new concepts for contact models for non-convex 3D bodies and different concepts for surface models (e.g. soil, building floors, etc) for humanoid locomotion research and wheel-terrain research in space. The other direction focuses on investigating interpolation methods for the reconstruction of 3D surfaces of (closed) objects generated with e.g. CAD software. These methods will be used for studying and modeling robotic grasping.

REFERENCES

- [1] V. Duindam and S. Stramigioli, *Modeling the kinematics and dynamics of compliant contact*, IEEE Int. Conf. Robotics and Automation, 2003.
- [2] S. Ehmann and M. Lin, *Accurate and fast proximity queries between polyhedra using convex surface decomposition*, Jour. Eurographics Computer Graphics Forum, vol. 20, n. 3, 2001.
- [3] G. Farin, *Discrete Coons patches*, Elsevier Jour. Computer Aided Geometric Design, vol. 16, n. 7, pp. 691-700, 1999.
- [4] M. Gross, *Are points the better graphics primitives?*, Jour. Computer Graphics Forum, vol. 20, n. 3, 2001.
- [5] D.J. Montana, *The kinematics of contact and grasp*, Int. Jour. of Robotics Research, vol. 7, n. 3, pp. 17-32, 1988.
- [6] D.J. Montana, *The kinematics of contact with compliance*, IEEE Int. Conf. Robotics and Automation, 1989.
- [7] G. Sohl and A. Jain, *Wheel-Terrain contact modeling in the ROAMS planetary rover simulation*, Int. Conf. ASME-IDET Computers and Information in Engineering, 2005.
- [8] S. Stramigioli and V. Duindam, *Variable spatial springs for robot control*, IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2001.
- [9] S. Stramigioli and V. Duindam, *A novel lumped spatial model of tire contact*, IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2002.
- [10] M. Visser, S. Stramigioli and C. Heemskerk, *Screw bondgraph contact dynamics*, IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2002.
- [11] P. Volino, N. M. Thalmann, *The SPHERIGON: a simple polygon patch for smoothing quickly your polygonal meshes*, Int. Conf. IEEE Computer Animation, 1998.
- [12] A. Yvart, S. Hahmann and G. Bonneau, *Smooth Adaptive Fitting of 3D models using hierarchical triangular splines*, IEEE Int. Conf. Shape Modeling and Applications, 2005.
- [13] Zoltan J. Cendes and Steven H. Wong, *C¹ Quadratic Interpolation Over Arbitrary Point Sets*, IEEE Computer Graphics and Applications, 1987.
- [14] Controllab Products B.V., *20-sim*, <http://www.20sim.com>, 2009.