A Toolkit for Haptic Force Feedback in a Telerobotic Ultrasound System

Reza Fotouhi (reza.fotouhi@usask.ca)
University of Saskatchewan https://orcid.org/0000-0002-3972-1468
Atieh Najafi Semnani
University of Saskatchewan
QianWei Zhang
University of Saskatchewan
Scott J. Adams
University of Saskatchewan College of Medicine
Haron Obaid
University of Saskatchewan College of Medicine

Research note

Keywords: Haptic Simulator, Gilbert–Johnson–Keerthi algorithm, Ultrasound, Tele-sonography

Posted Date: June 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-646007/v1

Abstract

Objective

To develop a collision engine (haptic force feedback simulator) compatible with a 5-degrees-of-freedom (DOF) haptic wand. This has broad applications such as for telerobotic ultrasound systems. Integrating force feedback into systems is critical to optimize remote scanning. A collision engine compatible with a 5-DOF haptic wand was developed based on the Gilbert–Johnson–Keerthi algorithm. The collision engine calculated force during collision between the wand and a virtual object based on code developed using MATLAB. A proportional force was subsequently returned to a user via the haptic wand, thereby simulating the collision force for the user. Three experiments were conducted to assess the accuracy of the collision engine on curved and flat surfaces.

Results

The average errors in calculation of distances between the wand and virtual object were 2.1 cm, 3.4 cm, and 4.2 cm for the model of the human hand, cylinder, and cuboid, respectively. The collision engine accurately simulated forces on a flat surface, though was less accurate on curved surfaces. Future work will incorporate haptic force feedback into a telerobotic ultrasound system. The haptic force simulator presented here may also be used in the development of ultrasound simulators for training and education.

Introduction

Ultrasound imaging is a common imaging modality for diagnosis of a wide variety of pathology. However, patients in remote and rural communities often have limited access to ultrasound imaging due to the lack of specialized sonographers in these communities. These challenges could be tackled with introducing a telerobotic ultrasound system that would allow specialists to remotely operate and control an ultrasound probe and remotely perform ultrasound scanning [1],[2],[3].

Most telerobotic ultrasound systems currently lack the ability to provide the remote operator with feedback of applied force on the patient during the ultrasound examination. The concept of haptic force feedback simulators is to replicate, in real-time, the physical interaction between the ultrasound probe and human body to give the operator the opportunity to adjust the forces applied during telerobotic scanning [4], [5].

To support integration of haptic force feedback during interaction with a remote object (such as a patient when performing a telerobotic ultrasound exam) or a virtual object (such as a simulated patient in an ultrasound simulator), integration of new haptics software is required. Many haptic packages which are currently available are proprietary rather than open source. Open source packages, such as V-COLLIDE and SOLID, operate on Linux and are not compatible with haptic devices, such as the 5-DOF haptic wand developed by Quanser which is used in this study [25].

To address this limitation, the objective of this study was to develop a collision engine (haptic force feedback simulator) compatible with a 5-DOF haptic wand and a Windows operating system. When the 5-DOF haptic wand collides with a virtual object, a proportional force is returned to the user via the haptic wand, thereby providing force feedback to the user. In this study we assess the performance of this haptic simulator on flat and curved surfaces of three different virtual objects.

Literature Review

Telerobotic Ultrasound Systems

A commercially available telerobotic ultrasound system is MELODY, developed by AdEchoTech [6],[7],[8]. The MELODY system currently does not allow the sonographer to control the amount of force applied by the ultrasound probe (head) and could benefit from haptic technology. *Haptics* is defined as “touch interactions that occur for the purpose of perception and manipulation of objects” [9],[10],[11]. Haptic rendering is a functionality that adds certain haptic properties to an object to give it a realistic feel [10].

Haptic Rendering- Collision Detection

Collision detection algorithms detect collisions between objects and an avatar in a virtual environment [4]. In order to detect a collision, the position of the end-effector (avatar) in the virtual environment must be determined. If the avatar is in free space and not colliding or touching a virtual object, then the calculated contact forces on the interface will be zero. However, if the avatar is touching a virtual object, forces are felt by the user touching the end-effector.

Haptic Rendering- Collision Response

Collision response, or force-response, algorithms compute interaction force between avatars and virtual objects. This force replicates contact between real objects in the real world as closely as possible. It calculates appropriate amount of force to be passed onto the haptic interface device. This force, F, is calculated using the following equation:

$$F = k\,x + C\,v \quad (1)$$

Here k is stiffness of object, C is viscous damping, and x is penetration depth between the object and avatar (e.g. haptic wand), and v is the linear velocity of the avatar.

Collision Detection Engine

In order to develop a virtual reality haptic simulator, a software package (collision detection engine) and a haptic interface are required. The purpose of a collision detection engine is to detect collisions between two objects, calculate the distance between colliding objects, calculate the collision force, and generate an equal force via the haptic device. Commonly used algorithms include the Gilbert–Johnson–Keerthi (GJK) algorithm, bounding volume algorithm, and virtual spring algorithm, among others [9], [12], [13], [14], [15].

In this toolkit, the GJK algorithm is utilized. The GJK algorithm is a method to detect collisions between two convex objects. The algorithm is based on calculation of the Minkowski difference between two convex objects and determining whether or not the origin included is different. If the origin is included, the two objects collide with each other and have some points in common [16].

Haptic Interface

Several algorithms have been used for training or simulation using haptic interface [17], [12], [18], [24], [19].

Evaluation of Haptic Interfaces

Haptic interfaces are devices that enable manual interaction with virtual environments. They are employed for tasks that are usually performed using hands in the real world [9], [18], [20]. The haptic device used for this research is a 5-DOF haptic wand. The interface has five degrees of freedom, consisting of three degrees of translation and two degrees of rotation (roll and yaw).

Current Limitations

Haptic packages that are utilized in majority of research works include Phantom Omni and Open Haptics [17]; however, neither are open source. V-COLLIDE and SOLID are two open source software packages commonly implemented as collision engines [11], both on Linux.

Materials And Methods

The collision engine which is developed in this study is based on collision detection using the GJK algorithm [21]. To calculate the approximate amount of force during the collision of the lower tip of a haptic wand and a virtual object, we developed MATLAB code which calculates the force based on equation (1) [11]. An overview of the methodology is shown in (Figure A4) and described in further detail as follows:

1) 3D objects

Three 3D objects were imported into SolidWorks and saved in STL (stereolithography) format. To compare results between flat and curved surfaces of objects imported into the collision engine, STL files of the following objects were imported into MATLAB and Simulink (see Figure A5):

a. A human hand reconstructed from a CT scan [22];
b. A cylinder with a 0.025 m radius and a 0.1 m height; and
c. A cuboid measuring $0.1 \times 0.1 \times 0.2$ m^3.

2) Conversion from STL to X3D format

Using Blender (v. 2.49), we converted the STL files to X3D format, which is the required format for Quanser 3D Viewer. The converted human hand model in X3D format is shown in Figure A6. Considering that visualization and force calculation are two parallel and separate processes (Figure A4), a transformation matrix between MATLAB and Blender coordinate systems was used to calculate applied forces from object positions.

3) Import into 3D Viewer

The 3D Viewer was used for visualization of an avatar of the wand's tip and the three objects imported into the virtual environment. Objects represented in X3D format were imported into 3D Viewer. The wand and an object in X3D format in the virtual environment are shown in Figure A7.

4) Convert STL files to object vertices

Using a variation of MATLAB code [23], we converted STL files to object vertices (see Figure 1a, Figure 2a, and Figure 3a). Vertices of an object were used as inputs for the GJK algorithm. These vertices form an $n \times 3$ matrix, n- number of vertices, and 3 coordinates, which define the object in the X, Y, and Z planes. The cuboid, cylinder, and human hand were represented by 36, 456, and 3660 vertices, respectively.

5) GJK algorithm in Simulink and MATLAB

Vertices of the imported objects produced in the previous step, as well as stiffness and damping ratios of each object and real time position vectors of the haptic interface, were used as inputs for the main MATLAB [23]. For all objects, stiffness in the X, Y and Z directions was manually set to (160, 160, 160) N•s/m, and damping factors were set to (0.8, 0.8, 0.8) N•s/rad, respectively.

The position of the haptic interface is a 1 × 3 vector. "1" is the indicator a single point (haptic interface), and "3" is the X, Y, and Z coordinates of this point. Real time position vector is output of a specific QUARC block called "Haptic 5 DOF Wand Cartesian Plane" (Figure A5).

The center of mass of each object was set to (0, 0, -0.075). We simplified the wand to be comprised of a single point with one vertex and three degrees of freedom. Initial position of the wand was set at (0, 0, 0).

6) Simulating and generating haptic force feedback

A "CollisionFlag" was deployed so that when a collision was detected, the flag went to "1"; otherwise, it stayed at "0". If "CollisionFlag" was "1", the code calculated the force based on equation (1). The 5-DOF haptic wand generated force feedback when the wand collided with an object in the virtual environment.

Three virtual experiments were conducted to assess the accuracy of the collision engine: dorsal aspect of the human hand, curved surface of a cylinder, and flat surface of a cuboid. In these experiments, the wand collided with the surface of the virtual object over a 10-second period. Distance between the wand and the virtual object was recorded, and contact forces between the virtual objects and the wand were measured by the collision engine based on equation (1) in X, Y and Z directions.

Results And Discussion

Experiment with a Model of a Human Hand

Figure 1b shows the distance between the wand and the dorsal surface of the hand in the first virtual experiment in which the wand was collided with a model of a hand over 10-second. At collision, the calculated distance between the wand and hand is zero.

The simulated contact force between the hand and wand are presented in Fig. 1c. The average of these forces during contact were 0.998 N, 0.996 N and 0.345 N in *X*, *Y* and *Z* directions, respectively. The average of the forces for all 11 contact periods is given in Table A1.

Experiment with a Cylinder

Figure 2b shows distance between the wand and curved surface of a cylinder in the second experiment where the wand was collided with the cylinder over a 10-second period. In some proximities the engine miscalculated distances, which caused a noisy collision response. These miscalculated distances occurred at 6 instances: 3.8, 4.0, 4.2, 6.5 6.9, and 9.0 s, with an average error of 3.44E-2 m. The simulated contact force between the cylinder and wand are presented in Fig. 2c. The average of these forces during contact were 0.3561 N, 1.711 N and 0.6248 N in *X*, *Y* and *Z* directions, respectively as shown.

Experiment with a Cuboid

Figure 3b shows distance between the wand and surface of a cuboid in the third experiment where the wand was collided with the surface of a cuboid. In this experiment, the engine returned a false value for the calculated distance at instance of 2.4 s, which led to a noisy collision response, with an average distance error of 4.176E-2 m. The simulated force during contact between the cuboid and the wand in *X*, *Y* and *Z* directions is shown in Fig. 3c. The average contact forces in *X*, *Y* and *Z* directions were 1.62 N, 0.477 N, and 0.217 N, respectively, as shown. By comparing the results of the two experiments, as shown in Table A2, it can be observed that the collision engine can detect the collision between the wand and a flat surface more efficiently than with a curved surface.

To verify the generated force based on Eq. (1) from Simulink, we manually calculated these forces for two random instances at 8.098 s and 8.436 s in the cuboid experiment. The distance between the cuboid and wand at 8.098 s was 6.645E-2 m, with an expected force value of zero at that time. Figure 3c shows that the related force at 8.098 s is equal to zero in *X*, *Y*, and *Z* directions. Figure 3b shows that the distance at instant 8.436 s is 9.964E-4 m, which means the wand was very close to the cuboid. Table A3 shows the wand velocity, distance vector from the wand toward the center of mass of the cuboid, and force value for *X*, *Y*, and *Z* directions at instant 8.436 s. The calculated force at this instant in the *X*, *Y* and *Z* directions based on Eq. (1) is the same as the extracted value of the force from Fig. 3c.

Conclusion

In this paper, a collision engine compatible with a 5-DOF haptic wand to simulate haptic force feedback is presented. Through virtual experiments with a cuboid, a cylinder, and a human hand models, we have shown that our approach can work on several surfaces. The collision engine was able to accurately simulate haptic force feedback on a flat surface, though was less accurate on curved surfaces. One of the limitations of the GJK algorithm [15] which was utilized in this research is that it can only detect collisions between convex objects. The haptic force feedback simulator presented here will have applicability in development of ultrasound simulators for training and education, and in the telerobotic ultrasound system.

Limitations

The proposed algorithm utilized in the collision engine is limited since it can only detect collisions between convex objects. The convex decomposition method [17] can be incorporated into the engine to remedy this.

Abbreviations

3D
three Dimensional
DOF
degrees-of-freedom
GJK
Gilbert–Johnson–Keerthi
MATLAB
matrix laboratory, a commercial software
MELODY
a remote robotic ultrasound solution from AdEchoTech
QUARC
real-time control software from Quanser
Quanser
A manufacturing company for engineering lab equipment
SOLID
an acronym for the first five object-oriented design principles
STL
Standard Template Library
V-COLLIDE
a collision detection library for large environments (of polygonal objects)
X3D
a royalty-free ISO/IEC standard for declaratively representing 3D computer graphics.

Declarations

- **Ethics approval and consent to participate:**

Not applicable

- **Consent to publish**

The co-authors give consent to publish this manuscript to the BMC Research Notes

- **Availability of data and materials:**

Data are available publicly at HF2MS in https://sourceforge.net/projects/hf2ms/files/

- **Competing interests**

The authors declare that they have no competing interests.

- **Funding**

This research was supported through funding from the Natural Sciences and Engineering Research Council of Canada, and from the College of Medicine, University of Saskatchewan.

- **Authors' Contributions**

Reza Fotouhi, and Haron Obaid co-supervised Atia Najafı Semnani, who was a Master student.

QianWei Zhang and Scott J. Adams are other team members who worked on this project.

Atia Najafı wrote the first draft of the paper with guidance from supervisors; She modified a Simulink model from Quanser to enable a 5DOF haptic wand to generalize haptic force feedback response for imported objects. Reza Fotouhi did extensive editing and redrafted final version of the paper. Other co-authors also helped in editing and development of the manuscript.

- **Acknowledgements**

This Project is collaborative research between Robotics lab in Mechanical Engineering Department, and Department of Medical Imaging at the University of Saskatchewan. People involved in this research are co-authors here. We also acknowledge help from Mr. Doug Bitner, Mechanical Engineering in setting up and troubleshooting, the haptic Wand.

References

1. Fotouhi R, Oraji R, Mondragon C, Berryman B. Development of a remote ultrasound imaging system. ASME International Design Engineering Technical Conferences and Computers and Information in

Engineering Conference; Boston, MA; 2015:V003T14A008.

2. Boman K, Olofsson M, Forsberg J, Boström S-A. Remote-controlled robotic arm for real-time echocardiography: the diagnostic future for patients in rural areas? Telemed J E Health. 2009;15(2):142–7. doi:10.1089/tmj.2008.0079.
3. Avgousti S, Christoforou EG, Panayides AS, et al. Medical telerobotic systems: current status and future trends. Biomed Eng Online. 2016;15(1):96. doi:10.1186/s12938-016-0217-7.
4. Salisbury K, Conti F, Barbagli F. Haptic rendering: introductory concepts. IEEE Comput Graphics Appl. 2004;24(2):24–32.
5. Blum T, Rieger A, Navab N, Friess H, Martignoni M. A review of computer-based simulators for ultrasound training. Simul Healthc. 2013;8(2):98–108.
6. AdEchoTech P. 2016. , Accessed December 4, 2020.
7. Adams SJ, Burbridge BE, Badea A, Kanigan N, Bustamante L, Babyn P, et al. A crossover comparison of standard and telerobotic approaches to prenatal sonography. J Ultrasound Med. 2018;37(11):2603–12.
8. Adams SJ, Burbridge BE, Badea A, et al. Initial experience using a telerobotic ultrasound system for adult abdominal sonography. Can Assoc Radiol J. 2017;68(3):308–14. doi:10.1016/j.carj.2016.08.002.
9. Kraut J, Hochman JB, Unger B. Temporal bone surgical simulation employing a multicore architecture. 26th IEEE Canadian Conf on Electrical and Computer Eng; Regina, SK; 2013:1–6.
10. Singapogu R, Sander ST, Burg TC, Cobb WS. Comparative study of haptic and visual feedback for kinesthetic training tasks. Stud Health Technol Inform. 2008;132:469–71.
11. Laycock SD, Day AM. A survey of haptic rendering techniques. Comput Graph Forum. 2007;26(1):50–65.
12. Ericson C. Chapter 9 - Convexity-based methods. In: Ericson C, editor. Real-Time Collision Detection. San Francisco: Morgan Kaufmann; 2005. pp. 383–412.
13. Liu L, Wang Z, Xia S. A volumetric bounding volume hierarchy for collision detection. In: 2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics; Beijing, China; 2007:485–488.
14. Bryan J, Stredney D, Wiet G, Sessanna D. Virtual temporal bone dissection: a case study. In: Proceedings of the Conference on Visualization '01. San Diego, CA; 2001:497–500.
15. Barbieri E. Improving the GJK algorithm for faster and more reliable distance queries between convex objects. ACM Trans Graph. 2017;36(4):151a.
16. Ericson C. Chapter 3 - A Math and Geometry Primer. In: Ericson C, editor. Real-Time Collision Detection. San Francisco: Morgan Kaufmann; 2005. pp. 23–73.
17. Noborio H, Sasaki D, Kawamoto Y, Tatsumi T, Sohmura T. Mixed reality software for dental simulation system. 2008 IEEE International Workshop on Haptic Audio Visual Environments and Games; Ottawa, ON; 2008:19–24.

18. 3D Systems. Scanners and Haptics. 2020. Accessed July 14, 2020. https://www.3dsystems.com/scanners-haptics.
19. Kim L, Sukhatme GS, Desbrun M. A haptic-rendering technique based on hybrid surface representation. IEEE Comput Graph Appl. 2004;24(2):66–75.
20. Unger B, Sepehri N, Rampersad V, Pisa J, Gousseau M, Hochman JB. Elements of virtual temporal bone surgery: manipulandum format may be more important to surgeons than haptic device force capabilities. Laryngoscope Investig Otolaryngol. 2017;2(6):358–62.
21. Talasaz A, Trejos AL, Perreault S, Bassan H, Patel RV. A dual-arm 7-degrees-of-freedom haptics-enabled teleoperation test bed for minimally invasive surgery. J Med Devices. 2014;8(4):041004.
22. Hand V1 3D Model. free3d.com/3d-model/hand-v1–675788.html, Accessed Jul 10, 2020.
23. Lebel P. GJK algorithm distance of closest points in 3D. 2018. MathWorks File Exchange. Accessed July 10, 2020. https://www.mathworks.com/matlabcentral/fileexchange/62429- gjkalgorithm-distance-of-closest-points-in-3d.
24. Virtual Reality Dental Training System. Accessed July 14. 2020. http://www.novint.com/VRDTS.htm.
25. Quanser H. 2020. Accessed July 14, 2020. quanser-update.azurewebsites.net/quarc/documentation/quarc_using_devices_haptics.html.

Figures

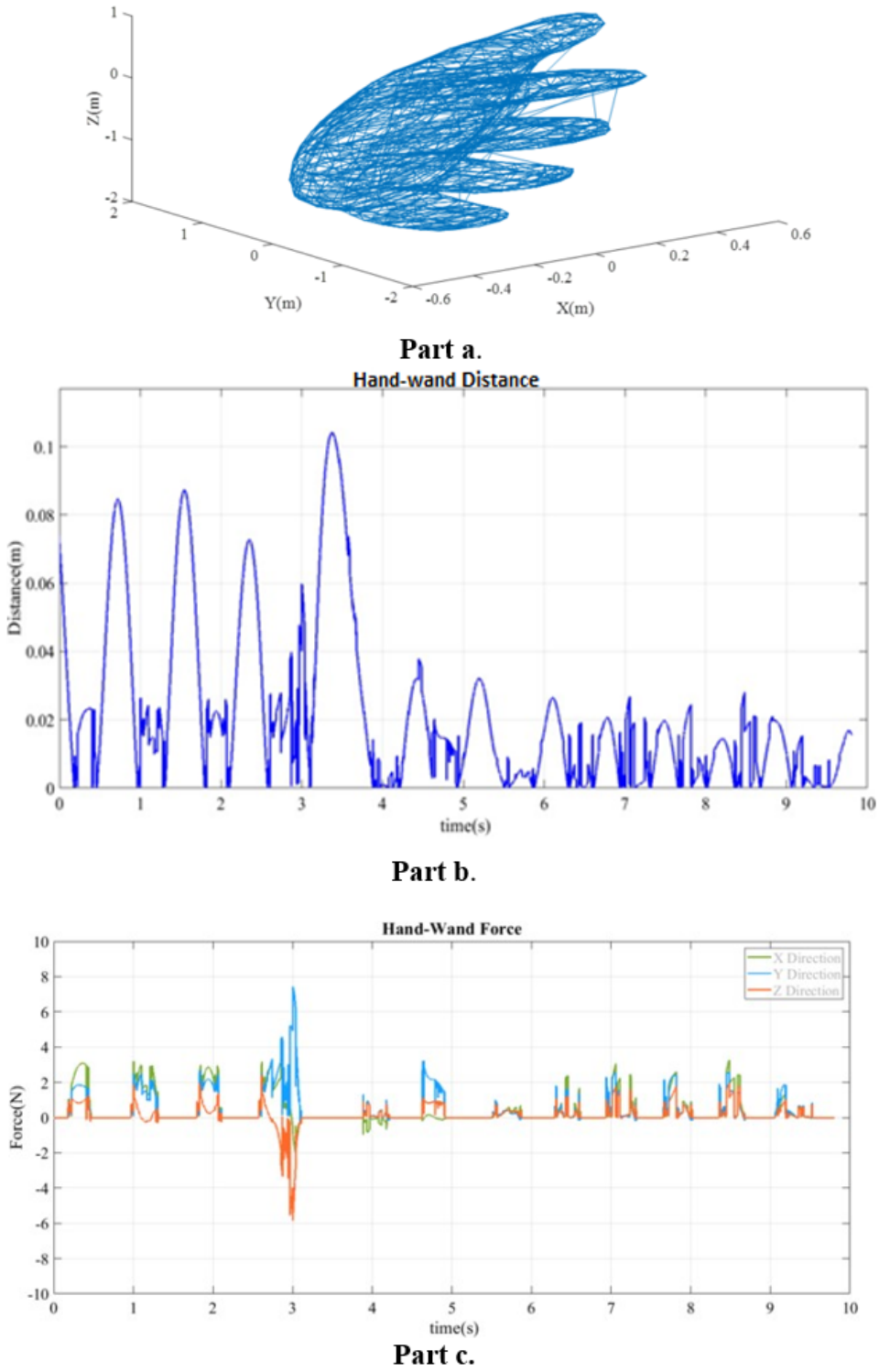
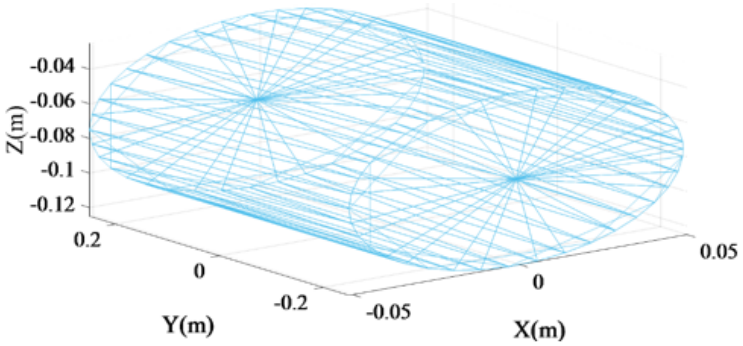
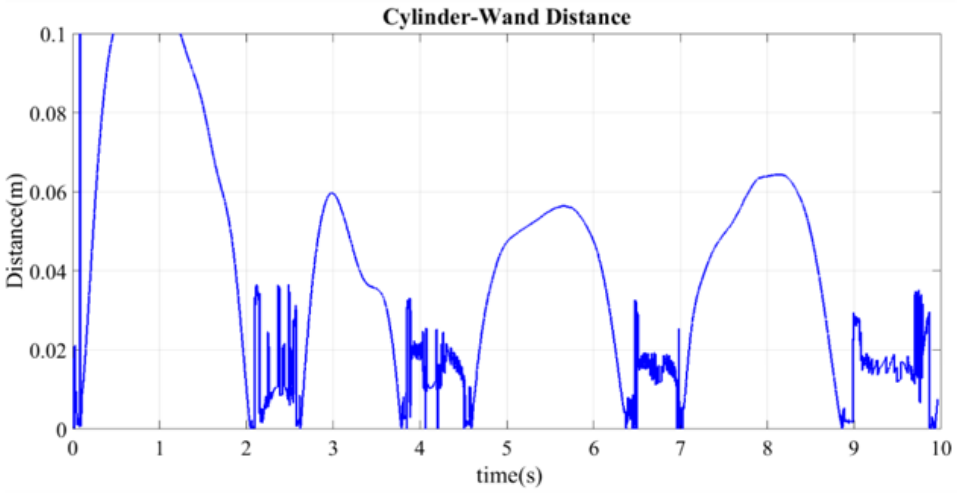


Figure 1

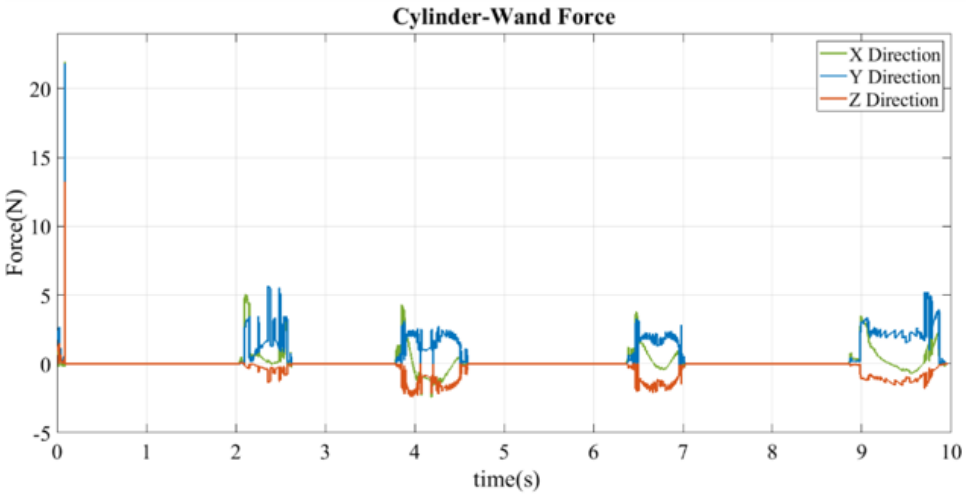
a. Plot of human hand model vertices; b. Distance sequence for human hand model and wand collision; c. Force sequence for human hand model and wand collision.



Part a.



Part b.



Part c.

Figure 2

a. Plot of cylinder model vertices; b. Distance sequence for cylinder model and wand collision; c. Force sequence for cylinder model and wand collision.

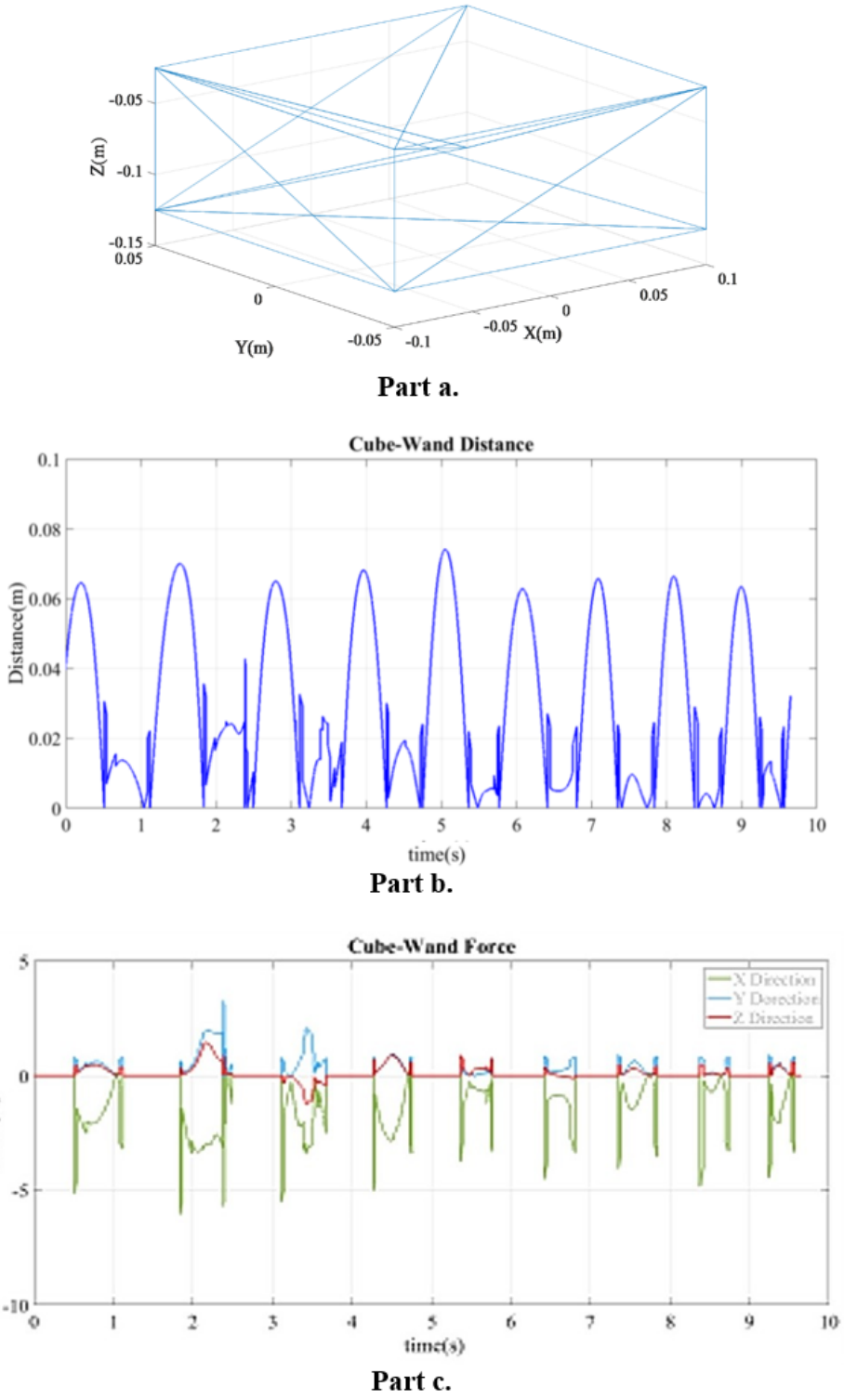


Figure 3

a. Plot of cube model vertices; b. Distance sequence for cube model and wand collision; c. Force sequence for cube model and wand collision.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- AtiarevisedmanuscriptBMCMar2021v3appendix.pdf