Surgical Robotics Environment for NVIDIA Isaac Sim

EN.601.456.01.SP24 Computer Integrated Surgery II

#### Team #8 Background Reading

Tae Wan Kim (tkim104@jhu.edu)

March 15th, 2024

# **Table of Contents**

[Team #8 Background Reading 0](#_95jtzw3bgnlj)

[**Table of Contents 1**](#_ozu41kx0c6q3)

[**1. Project Overview 2**](#_x77cwb6c0mzu)

[**2. Paper Selection 2**](#_6v8oi9ew58rl)

[**3. Background 2**](#_rvhkvbjfn1v3)

[**4. Summary 3**](#_4xb4u7ff316g)

[**5. Methods 3**](#_gxmx8u10l5kj)

[**6. Key Results and Significance 7**](#_8yze1onza6up)

[**7. Assessment and Relevance 7**](#_3fz0ef851w0s)

[**8. Conclusions 8**](#_jngi04h31uua)

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **1. Project Overview**

The primary goal of the project is to find out the potential and limitations of NVIDIA Isaac Sim for surgical robotics environment applications and compare its performance against the Asynchronous Multibody Framework (AMBF) developed by Dr. Munawar. To that end, we hope the simulator is able to do a number of tasks required from a surgical robotics simulator to eventually recreate the 2022-2023 Surgical Robotics Challenge done in AMBF, perform a wide range of dynamics calculations, while providing the added advantage of highly realistic video and images of a simulated environment.

## **2. Paper Selection**

*Citation:* **A. Munawar**, J. Y. Wu, G. S. Fischer, R. H. Taylor and P. Kazanzides, "Open Simulation Environment for Learning and Practice of Robot-Assisted Surgical Suturing," in *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3843-3850, April 2022, doi: 10.1109/LRA.2022.3146900.

"Open Simulation Environment for Learning and Practice of Robot-Assisted Surgical Suturing," is the seminal paper on the development of AMBF and its use within the surgical robotics environment research. While the Surgical Robotics Environment for NVIDIA Isaac Sim focuses on the features and capabilities that NVIDIA's software provides, many aspects of this project focuses on the comparison between what has been done in AMBF and what could be done in Isaac Sim. To get a fair comparison between both platforms would require extensive knowledge on both simulators, thus while this project heavily works with Isaac Sim, we require additional work and information that can be provided by papers on AMBF.

Overall, out of all the readings listed in our readings list, this paper provides the foundational comparison we need to successively complete the project and relates to what we hope to achieve in the minimum, expected, and maximum deliverables.

## **3. Background**

Current robotics simulators have developed into several specialities where each simulator performs certain tasks better than others depending on the application. Some may highlight its ability to run several instances of the simulator concurrently, showcasing its efficiency in representing hundreds if not thousands of robots in individual or shared environments. Other programs may focus on the realism of its physics engine, outputting thousands of data points to describe an object's position, velocity, and / or acceleration for machine learning purposes. Within the realm of surgical robotics, one of the more relevant features can involve fluid dynamics or soft-body environments reacting to different kinematics or linear forces. This can further specialize depending on the type of surgery or surgical task one would like to emulate in a virtual environment, or possibly automate as well. Through all of this, the choice of simulator becomes important for what task the user would like to simulate or conduct research on.

Shifting our focus to surgical robotics environments, some important features we would desire in a simulation include the dynamics mentioned before, real time control with the virtual robot, real time feedback from the environment, and the streaming of high fidelity information or "labels" for artificial intelligence development. When developing surgical environments, we want a realistic emulation of the real world to accurately explore how certain procedures could affect a patient, or how precisely we can automate certain repetitive tasks. Other possibilities include using the simulator as a training tool for new surgeons to practice with and develop confidence in for more efficient and supported learning. Considering the straightforward collection of ground truth data in segmentation and depth maps, simulators can provide an excellent framework for deep learning and machine learning models to train from as well.

## **4. Summary**

The paper introduces us towards the trend of automation entering multiple fields of work, including that of the medical world. Ultimately, robotics development has progressed to the point where the broader research space seeks to automate certain surgical tasks, such as suturing or other repetitive motions. Yet, while interest in these topics have increased overtime, the paper argues that there has not been a unified collective standard to test and research surgical environment systems for reproducible results. A multitude of simulators and several frameworks exist and have been in use in emulating medicine, but certain programs at the time did not provide all the relevant features involved in surgical robotics.

The paper presents a surgical environment built on top of the Asynchronous Multibody Framework (AMBF) simulation system to fill this gap in research at the time, demonstrating its use with the da Vinci Research Kit (dVRK); the primary software to access models and control of the da Vinci robot. Going over the additional features added to AMBF, the paper also highlights the key capabilities needed in a proper surgical robotics environment and ways AMBF is able to address those features.

The structure of the paper itself first provides an Introduction section to allow the readers to understand the context and significance of the paper and work involved, while the following sections highlight the AMBF system developed. Related Works covers the existing software already present at the time and reasons why those applications fail to address certain aspects of surgical robotics while the Methods give an overview of the AMBF framework itself. At the end of the paper, it summarizes its test results when used by a surgeon and fellow, and discusses possible future directions for their project.

## **5. Methods**

The paper breaks down their methods into 5 main sections going over what was developed and created in their AMBF setup: *System Architecture* provides and overview of their AMBF Description Format and how their files are organized, *Realistic Interaction Between Environment Objects* details the physics involved in their simulation, *Simulation of Surgical Suture* explains the surgical challenge at hand, *Controlling the Robot* illustrates how to manipulate the robot, and *Rendering Scene Data* describes the rendering pipeline on displays.

*System Architecture*

| Munawar et al.: OPEN SIMULATION ENVIRONMENT FOR LEARNING AND PRACTICE OF ROBOT-ASSISTED SURGICAL SUTURING |
| --- |

In summary, the AMBF system is built using 3 types of ADF files: one to describe the world containing the cameras and lights, one to describe any input devices to control a robot, and one to describe models within a virtual space. In addition to AMBF, they included the use of the Blender AMBF Addon and the Collaborative Robotics Toolkit Specifications (CRTK) to facilitate further ease of access with their simulation.

The Blendor AMBF Addon allows easier creation of assets for AMBF in the specific ADF file type, while the CRTK acts as the standardized way of interfacing with their system. While the ADF file type created in Blender allows the user to control several key aspects of an object, such as collision, meshes, and physics attributes, at the time the paper was written, the addon was only configured with ADF, thus limiting the file type other models could be designed in and ported for use in AMBF.

*Realistic Interaction Between Environment Objects*

One important aspect of simulations is the way objects react to each other efficiently, thus the collision algorithms involved were developed to support a group of primitive shapes or convex hulls in AMBF. The paper demonstrates this ability in the surgical challenges scene formulation with intuitive collision boxes on the models of a needle, the phantom, and the suturing entrance of the phantom as well.

| Munawar et al.: OPEN SIMULATION ENVIRONMENT FOR LEARNING AND PRACTICE OF ROBOT-ASSISTED SURGICAL SUTURING | Munawar et al.: OPEN SIMULATION ENVIRONMENT FOR LEARNING AND PRACTICE OF ROBOT-ASSISTED SURGICAL SUTURING |
| --- | --- |

*Simulation of Surgical Suture*

Another element of the simulation the paper explains is the setup used for the Surgical Suture challenge. AMBF utilizes the Bullet Physics Library, a popular open-source physics engine known for its ability to simulate collision detection, soft-body, and rigid-body physics, using its default velocity-based solver known as Sequential Impulse (SI) to simulate the robots. Its solved using the following impulse calculations and update parameters:

| Munawar et al.: OPEN SIMULATION ENVIRONMENT FOR LEARNING AND PRACTICE OF ROBOT-ASSISTED SURGICAL SUTURING |
| --- |

* F\_e: External force,
* X\_i: Current Positions
* X\_i+1: Next Positions
* V\_i = Current Velocities
* V\_i+1: Next Velocities
* M: Combined Mass Matrix
* M\_e: Effective Mass
* J: Jacobian Constraint
* b: Bias Velocity

The thread used in the challenge was simulated with 6 degrees of freedom between two rigid bodies. Other additional elements include trial by error tuning of parameters to represent the thread, and a ghost object placed in the phantom to determine whether or not a needle or thread entered the phantom, then applying viscous friction force against the movement of the needle to represent moving inside tissue.

*Controlling the Robot*

The robot inside the challenge is modeled after the dVRK Patient Side Manipulators (PSMs) and moves based on the inverse kinematics analytical solution. Because the dVRKs PSMs were designed with 6 degrees of freedom and a remote center of motion, the authors were able to speed up the control of the robot with the analytical solution to the inverse kinematics problem.

Two additional features the creators added to the framework includes a teleoperation package with compatibility with the dVRK Master Tool Manipulators (MTMs), Geomagic Touch, and Razer Hydras. As well, they have included friction based grasping, solved using SI. However, the authors developed a system such that the tool / needle held by the robot's manipulators prevent any slippage during the simulation. While this may be sufficient for their applications, for more accurate and flexible control in a surgical robotics environment would like to have more optimal control of the friction coefficients involved in the SI calculations.

| Munawar et al.: OPEN SIMULATION ENVIRONMENT FOR LEARNING AND PRACTICE OF ROBOT-ASSISTED SURGICAL SUTURING |
| --- |

*Rendering Scene Data*

Finally, the authors of the paper created their rendering pipeline natively in Open-GL which allows them to use simple Gouraud Shading in their scenes. However, they did update the AMBF so custom ADF files can be developed to use Bling-Phong or Physically Based Rendering with Image Based Lighting which are considered as more realistic representation of objects in rendering. Thus, the features for visual scene data are present for machine learning. Often machine learning would require depth maps (how far a pixel is in 3D space) or segmentation data (what group or object a pixel represents) to use as ground truths and learn from.

| *Woo, M., Neider, J., Davis, T., & Shreiner, D. (1999). OpenGL programming guide: the official guide to learning OpenGL, version 1.2. Addison-Wesley Longman Publishing Co., Inc.* | *https://rtarun9.github.io/blogs/physically\_based\_rendering/* |
| --- | --- |

## **6. Key Results and Significance**

One main result from the paper involved its ability to perform real-time updates in the simulator in time with reality. In achieving the real-time performance, the optimizations proved fruitful in creating a system that allows user training on surgical tasks. The other was receiving qualitative feedback from a trained surgeon and fellow when using the AMBF framework on the surgical challenge. Their feedback is summarized as follows:

* Suture behavior, teleoperation, and scene looks convincing
* Insertion dynamics and feedback need to be improved
* Needle grasp felt more rigid than the actual PSMs
* Simulated PSMs moved differently than actual PSMs

Overall, the authors succeeded in creating a simulator specific to surgical robotics settings and proved its use for real-time training and qualitatively argued for its accuracy in simulating a common surgical task.

## **7. Assessment and Relevance**

The authors of the paper address many of the flaws within AMBF they hope to amend with further plans in developing the system to handle features such as softbody dynamics, penetration collision, and visual rendering of the thread involved with the task. Overall, their system is a remarkable simulator that was able to conduct the challenge at hand and provide a decently realistic scene for robotics surgical applications.

However, there are some elements of the project that may need to be worked on further that have not been already mentioned. There was no discussion on how to or if there were plans to implement fluid dynamics in the simulator, thus surgical tasks that involved how fluids, like blood or cerebrospinal fluid, would interact in their environment. There was also no discussion over how the performance of a simulator may depend on the hardware it is running on. While the AMBF system can run in real-time for their setup (AMD Ryzen 3600 CPU, 32 GB of RAM, and an Nvidia 1080 GPU running Ubuntu 20.04.) it would be remiss not to go over how other hardware specs may affect the performance of the simulator and what tradeoffs are possible between software optimization, program accuracy, and hardware power. Certain physics calculation accuracy or visual rendering may perform better in other simulators, such as NVIDIA's Isaac Sim, than in AMBF.

Thus, AMBF provides our project a benchmark to compare performance and feature availability for Isaac Sim's ability to generate an accurate and possible real-time surgical robotics environment. Several of our deliverables are in direct comparison with AMBF and in fact involve recreating the surgical challenge described in the paper in Isaac Sim. Features such as asset creation, realistic rendering, and kinematic control of a robot, are items every robotics simulator must address, and are metrics to compare one simulator against another.

## **8. Conclusions**

The paper is a foundational piece to describe the goals we hope to accomplish with Isaac Sim. It describes the surgical challenge we wish to recreate in NVIDIA's software and provides several examples of what needs to be considered when creating such an environment for surgical robotics. The paper went over items such as asset creation and storage, the software and calculations involved in controlling a robot and representing a relevant scene. The description of the surgical challenge made in the paper is highly relevant to the tasks involved with the Isaac Sim project with its comparison to AMBF and its ability to create an environment for surgical robotics.