

ENHANCED SURGICAL SIMULATION FOR THE DAVINCI ROBOT SYSTEM

Project Proposal (CONFIDENTIAL - bounded by NDA with Intuitive Surgical Inc.)

Goal

This project aims to create a simulation interface for the daVinci S System [4] used for robotic surgery. We would be interfacing the simulation sandbox written by Intuitive Surgical Inc. (ISI) [4] with the master console of the daVinci S system, present in the *Swirnow Mock Operating Room* at Johns Hopkins University (JHU). By doing so, we would be providing an open simulation playground for researchers to design their own tasks and applications in this controlled environment.

Team Members

Anand Malpani, Third Year Graduate Student, Dept. of Computer Science

Mentors

Anton Deguet, Sr. Software Engineer, Laboratory of Computational Robotics and Sensing, JHU

Russ Taylor, Professor, Director - CISST ERC, Dept. of Computer Science, JHU

Simon DiMaio, Senior Manager, Research and Advanced Systems Development, ISI

Ashwinram Suresh, Systems Engineer, ISI

Background and Motivation

Intuitive Surgical Inc. has been developing robotic surgery products since 1995. Currently, ISI has three generations of daVinci robotic systems viz. *STANDARD*, *S* (system in Hackerman 134 - Mock OR, JHU), and *SI*. The daVinci robot is a tele-robotic system (Fig. 1) with a master console from where the surgeon can see a stereoscopic view of the slave (patient) side with the use of a stereo endoscope mounted on one of the robotic slave arms. The surgeon uses Master Tele Manipulators (MTM's) to control the configuration of the Patient Side Manipulators (PSM's) which have seven degrees of freedom. While the system has its advantages for making minimally invasive surgery simpler and more intuitive, it has shortcomings for training surgeons to use it. For example, the system large and requires a spacious room for setting up a training center; like most surgical training equipment phantoms get wasted; a daVinci robot is a scarce resource in medical institutions with less time allotted for surgical training.

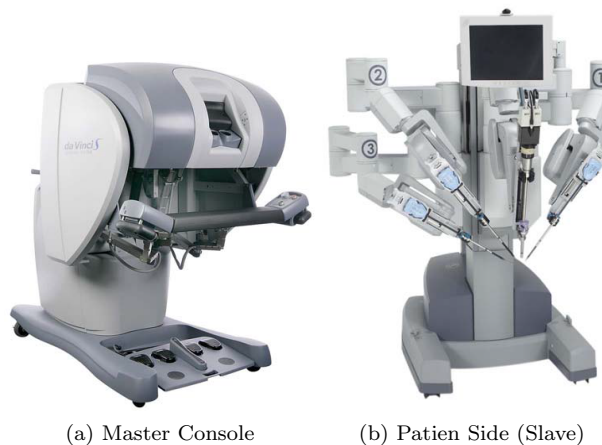


Figure 1: daVinci S Surgical System Components [6]

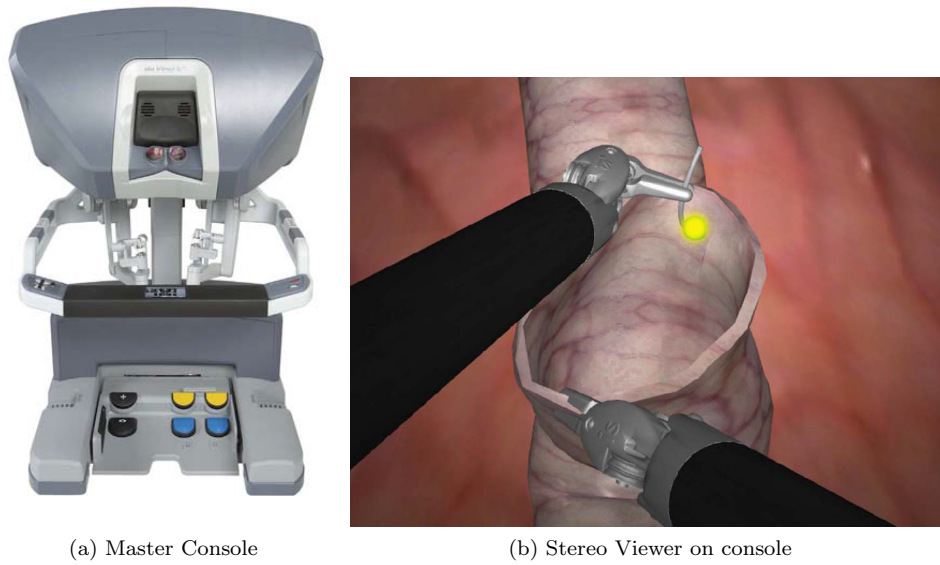


Figure 2: daVinci Skills Simulator [7]

With this and other factors in view, ISI has recently launched a *Skills Simulator* system (Fig. 2) that gets rid of the patient side robot by simulating the PSM's on the master console stereo viewer (Fig. 2(a)). This serves the purpose of reducing training time spent on the actual robot and provides the add-on features of repeatability and structured task environments, which are essential for standardizing training and assessing the performance of a trainee. Such a platform for simulating robotic surgical tasks could have wider applications in the fields of skill evaluation, surgical planning, task modeling, and so on.

However, at present, the simulation environment of the daVinci Skills Simulator is handled and outsourced to Mimic Technologies Inc. [8]. The graphics component appears as a black-box which takes the MTM and PSM state information and renders the virtual PSM's and other graphical components of the environment like the anatomy, manipulation objects, etc. accordingly. Thus, a developer does not get access to the knowledge of where the graphical components are rendered. And, the above mentioned applications of learning task models, developing new tasks are hindered. This is the motivation of our project which aims to overcome this obstacle.

Significance

We are working towards developing an open simulation interface for the daVinci system, in general, and extending it to the daVinci S console available at JHU. Once in place, this framework would:

- Enable researchers to design their own environments using the graphics, physics models developed by ISI
- Enable developers to add new object models to the existing library of models for new applications
- Prove to be a great source for groundtruth data (as we would have knowledge about the rendering)
- Be the testing ground for learning approaches which model a task performance before applying them in the real world
- Allow simulating patient specific anatomical data to plan a robotic surgery and learn before hand about any difficulties that could be arising while using the planned approach

Technical Approach

The projects aims to create a framework for slave side simulation of the daVinci system using open source dependencies (except for the ISI BB API [1], which is used to communicate back and forth with the robot). ISI has already created a “*Sandbox*” containing some object models and a couple of training tasks using H3D [2], an open source scene graph API. Currently, the sandbox communicates directly to the Skills Simulator console using the ISI SIM API [?] and can be used only on that particular version of the console. However, we would like to interface this sandbox using the CISST libraries and the SAW framework [3] as it would provide a component based application.

ISI API’s There are three different versions of the ISI API *viz.* ISI API, ISI BB API and ISI SIM API. They are compared below. We would like to be able to use the sandbox with the daVinci S console and, as we can see that we lack the necessary functionality in the ISI BB API of being able to set-up the slave side configuration. This can still be overcome by using inverse kinematics and computing the active joint angles. However, the ISI BB API was written to interface with a real slave (which has its own “Core” that maintains a state machine on the slave side). Thus, in order to be able to use the ISI BB API with the virtual slaves created in the simulation sandbox, we need to implement a virtual slave “Core” for state maintenance and communications with the master “Core”. This is done through an existing “Core” implemented by ISI on the Skills Simulator which is restricted to the Si console only.

Type	Read	Write	Setup	CISST-SAW component	daVinci version
ISI API	✓	✗	✓	✓	STANDARD, S, Si
ISI BB API	✓	✓	✗	✓ (lacks features)	S
ISI SIM API	✓	✓	✓	✗	Si

CISST-SAW The CISST libraries have been developed by JHU for serving computer assisted interventions. The SAW (Surgical Assistant Workstation) aims at enhancing a surgeon’s capabilities through implemented components that interface with many of the computer integrated surgical devices. A key aspect of the CISST-SAW framework is the component-interface model it is built on. Thus, as long as a component can “provide” some interfaces and be given some “required” interfaces from another component, the components can communicate with each other. An existing component `sawIntuitiveDaVinci` already interfaces to the daVinci systems to receive information from the master and slave sides. We would be extending this framework to support some new functions absent in the ISI BB API (but, provided by the ISI SIM API). Additionally, we would be creating a virtual slave “Core” component to enact the functionalities served by the actual “Core” on the daVinci system or Skills Simulator like state maintenance, etc.

H3DAPI Currently, the graphics and physics modeling of objects, tools, and the environment is performed using the H3D framework which is built using the OpenGL library (for graphics rendering) and HAPI library (for haptics library). It has a physics toolkit, **H3DPhysics** which implements some of the common physics engines like Open Dynamics Engine (ODE), Bullet Physics, and PhysX. The existing sandbox uses the ODE implementation (supports rigid body physics only). We would be using PhysX implementation to support a future goal of including anatomical models as well. ISI had added custom nodes to the H3DPhysics framework for interacting with the ISI SIM API. We would be mimicking the same for SAW component of the ISI BB API and the virtual slave “Core”.

Now that all the components and libraries related to the framework have been introduced, we would like to present a system block diagram for our “final” sandbox (Fig. 3). As seen, the PC running the simulation sandbox would be communicating back and forth with the master console via the CISST-SAW wrapper of the ISI BB API via a network connection. Additionally, the sandbox would be rendering stereo simulation (based on the H3D library) on the stereo viewer of the console via DVI cables.

Once, we have the proposed simulation sandbox for the daVinci S console, the aim would be to start using existing models and/or developing new models for the application of idea port placement, drop-in ultrasound probe simulation, etc. We would also work towards improving the sandbox by adding more features which are mentioned under the deliverables section.

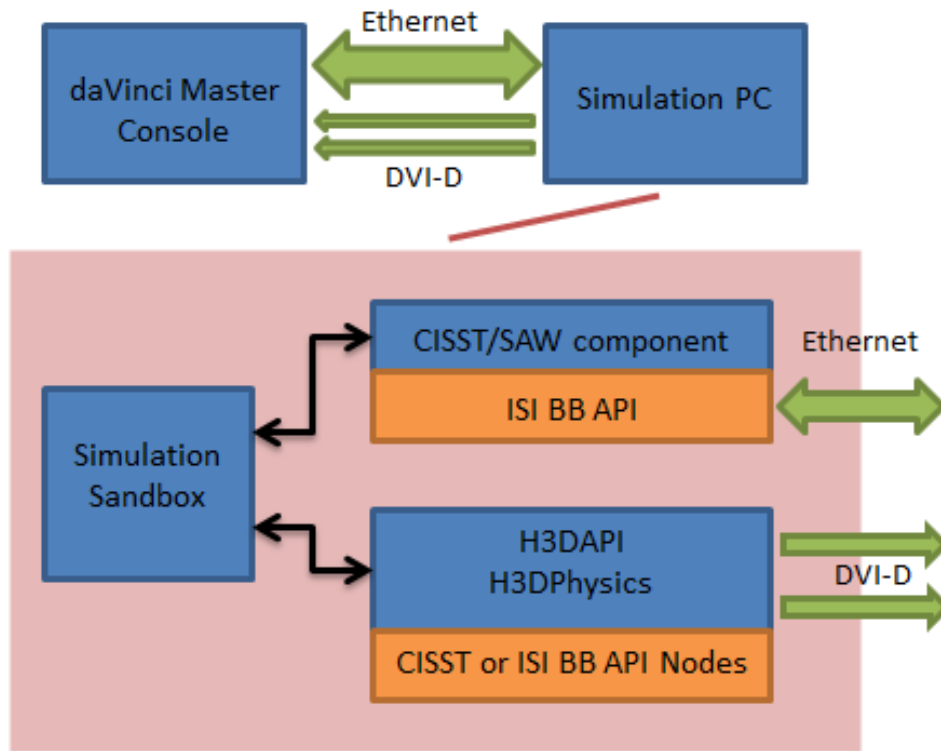


Figure 3: System block diagram of the proposed simulation sandbox

Deliverables

Minimum

- Extend and develop the existing CISST-SAW component for interfacing with the ISI BB API to include additional functions related to setup of the virtual slaves for simulation.
- Create a simple virtual slaves “Core” for emulating the actual “Core” of the slave side of the robot.
- Create a new CISST component for interfacing to the simulation library (H3D) (if needed).
- Demonstrate the sandbox using a basic example with basic features.

Expected

- Extend the virtual slave “Core” to incorporate other features of the daVinci console e.g. swapping instruments, camera control, etc.
- Mimic the existing sandbox sample tasks from ISI to interface to the CISST components (modifying according to available features in the sandbox)
- Demonstrate the complex tasks using the extended sandbox application on the daVinci S console.

Maximum

- Develop a new application involving new models using the extended simulation sandbox, perhaps demonstrating the implications of this framework.
- Extend the sandbox for including features needed in the new applications developed.

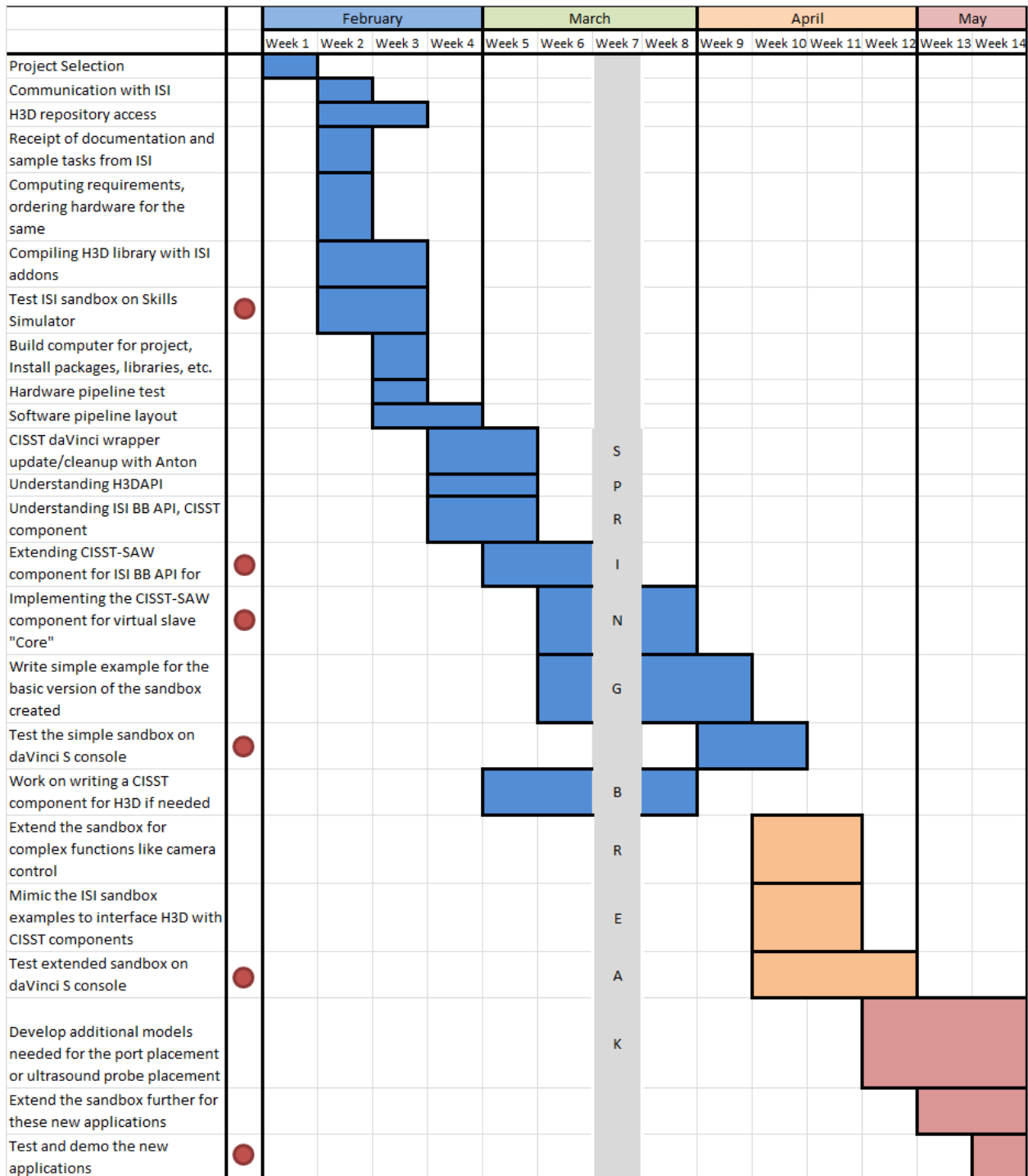
Dependencies

Dependency	Resource	Alternative	Impact
daVinci Skills Simulator access	Other project usage	None	Not much, project can be carried forward without this test
Development Drive for Simulator [DONE]	Anton	None	Not much, project can be carried forward without this test
daVinci S System access	Others using it for their projects	Make sure to reserve on calendar	Slow down the project
Computer for simulation [DONE]	Info from ISI, Approval from Prof. Taylor	None	Project can't happen without this!
Test hardware pipeline	New GPU	other GPU	project heavily depends on this!
H3D library svn access [ON GOING]	Firewall block, Network Security JHU	Use a particular SVN revision checkout from outside Hopkins and don't bother about updates	Shouldn't be much unless some bug gets fixed on the library
CISST ISI BB API wrapper update	Anton	Do myself	Crucial for communicating with the robot
ISI BB API intro	Anton	Do myself or ask others experienced with ISI BB API	Not much (slow down)
ISI BB API access [DONE]	Anton	None	Project can't happen without this!
Sample Tasks from ISI [DONE]	Simon, Ashwin	Refer to examples from H3D documentation	Project will be slowed down!
Documentation and troubleshoot on models from ISI [ON GOING]	Ashwin	Do myself	Slow down the project

Management Plan

- Weekly meetings with Anton are set for Wednesdays at 2pm.
- Working with Alison to schedule the weekly meeting at a time when Prof. Taylor can join too.
- Fortnightly meetings with ISI already set for the JHU-ISI group on Thursdays at noon.
- Standing meeting time tentatively Thursdays at noon with Anton, Simon, Ashwin.

Timeline, Milestones



Acknowledgements

This project is funded through the Swirnow Family Foundation and Intuitive Surgical Inc. This would have not been possible without the gift from Richard A. Swirnow (JHU alumnus) for building the Mock Operating Room. The daVinci S Surgical System and the daVinci Skills Simulator are on loan from Intuitive Surgical Inc.

References

- [1] Intuitive Surgical Inc., ISI API User Guide.
- [2] H3D.org, Open Source Haptics Library, <http://h3dapi.org/>
- [3] ERC CISST, <https://trac.lcsr.jhu.edu/cisst/>
- [4] Intuitive Surgical Inc., <http://www.intuitivesurgical.com/>
- [5] http://www.intuitivesurgical.com/products/davinci_surgical_system/
- [6] Intuitive Surgical Inc., daVinci S System User Manual
- [7] Intuitive Surgical Inc., daVinci Skills Simulator Manual
- [8] Mimic Simulation, <http://www.mimicsimulation.com/>