

**Technical Tutorial**  
for  
da Vinci Research Kit Starters

Prepared by Haoying(Jack) Zhou

*Worcester Polytechnic Institute  
Department of Robotics Engineering  
Automation and Interventional Medicine (AIM) Robotics Research Laboratory  
Gregory Fischer, Ph.D.  
Haoying Zhou*

June 30, 2023

# Contents

<b>Revision History</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Objective . . . . .	2
1.2 System Configuration . . . . .	2
1.3 Hardware and Terminology . . . . .	2
1.4 Document Conventions . . . . .	3
1.5 Intended Audience and Reading Suggestions . . . . .	3
1.6 Product Scope . . . . .	4
1.7 References . . . . .	4
<b>2 Hardware Setup</b>	<b>5</b>
2.1 Wire Connection . . . . .	5
2.1.1 Daisy Chain (Power cable) . . . . .	5
2.1.2 Firewire . . . . .	6
2.1.3 Foot Pedal . . . . .	6
2.1.4 Endoscope Focus Controller . . . . .	7
2.1.5 Ethernet UDP . . . . .	8
2.1.6 HRSV Video Pipeline . . . . .	8
2.1.7 Endoscope . . . . .	10
2.1.8 Other Hardware . . . . .	11
2.2 Robot Initialize and Power Check . . . . .	11
2.2.1 Emergency Stop Check . . . . .	11
2.2.2 Board ID Check . . . . .	12
2.2.3 Robot Initialization . . . . .	12
2.3 Calibration . . . . .	12
2.3.1 SUJ Current Calibration . . . . .	13
2.3.2 PSM Current Calibration . . . . .	13
2.3.3 ECM Current Calibration . . . . .	13
2.3.4 MTM Current Calibration . . . . .	13

---

2.4	Tool Deployment . . . . .	14
<b>3</b>	<b>Software and Applications</b>	<b>15</b>
3.1	dVRK Configuration Folder Path . . . . .	15
3.1.1	ROS 1 . . . . .	15
3.1.2	ROS 2 . . . . .	16
3.1.3	Understand JSON Configuration File . . . . .	16
3.1.4	Switch between ROS 1 and ROS2 . . . . .	16
3.2	Activate dVRK . . . . .	16
3.2.1	ROS 1 . . . . .	16
3.2.2	ROS 2 . . . . .	18
3.3	HRSV Activation . . . . .	18
<b>4</b>	<b>dVRK Baseline at WPI</b>	<b>19</b>
4.1	Hardware Check . . . . .	19
4.2	Software Run . . . . .	19
4.2.1	Connection Check . . . . .	19
4.2.2	ROS 1 . . . . .	19
4.2.3	ROS 2 . . . . .	20
<b>5</b>	<b>How to Run Teleoperation</b>	<b>21</b>
5.1	Move SUJ Manually . . . . .	21
5.1.1	PSM1 and PSM2 . . . . .	21
5.1.2	ECM . . . . .	22
5.1.3	PSM3 . . . . .	22
5.2	Move Robot Arms Manually . . . . .	23
5.3	Foot Pedal . . . . .	24
5.3.1	Endoscope Focus Modification . . . . .	24
5.3.2	Teleoperate PSM . . . . .	24
5.3.3	Teleoperate ECM . . . . .	25
<b>6</b>	<b>Common Q &amp; A</b>	<b>26</b>
6.1	Cannot Detect the Board . . . . .	26
6.1.1	All Boards not Detected . . . . .	26
6.1.2	Some Boards not Detected . . . . .	26
6.2	Teleoperation Stuck . . . . .	26
6.2.1	Trick 1 . . . . .	27
6.2.2	Trick 2 . . . . .	27
6.2.3	Trick 3 . . . . .	27

6.2.4 Trick 4 . . . . .	27
6.3 MTM Misalignment . . . . .	27

# Revision History

Revision	Date	Author(s)	Description
0.1	2023.05.18	Haoying Zhou	Create the initial document
0.2	2023.06.05	Haoying Zhou	Update Baseline for WPI operation
0.3	2023.06.18	Haoying Zhou	Update new general sections
1.0	2023.06.30	Haoying Zhou	First Release, link general instructions to GitHub

# Chapter 1

## Introduction

### 1.1 Objective

In this documentation, an instruction of operating first generation da Vinci surgical system with JHU CISST library is given. If you are interested in more details or need some additional information, please refer to the following link: <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/FirstSteps>.

### 1.2 System Configuration

This document is for a fully functional first generation clinical da Vinci surgical system (Also known as da Vinci Classic or dVRK Classic).

For WPI, we have one fully functional da Vinci surgical system in the operating room of PracticePoint. In addition, we have another dVRK setup at 85 Prescott Street second floor. They should share a similar setup process expect that the 85P one does not have a SUJ and thus it requires some additional registration or calibration.

For dVRK, the version 2.1.0 is utilized for this documentation.

### 1.3 Hardware and Terminology

- PSM: Patient Side Manipulator
- MTM: Master Tool Manipulator (L for Left and R for Right)
- ECM: Endoscopic Camera Manipulator
- HRSV: High Resolution Stereo Viewer
- SUJ: Setup Joints controller
- dVSS: da Vinci Surgical System
- dVRK: da Vinci Research Kit

For all the arms, the names and corresponding serial numbers are listed in Table 1.1:

Name	Serial Number	board ID
MTML	33100	0, 1
MTMR	36370	2, 3
ECM	25975	4, 5
PSM 1 (yellow)	23959	6, 7
PSM 2 (green)	29996	8, 9
PSM 3 (red)	32653	10 (A) , 11 (B)
SUJ	-	12 (C)

Table 1.1: dVSS arm configuration

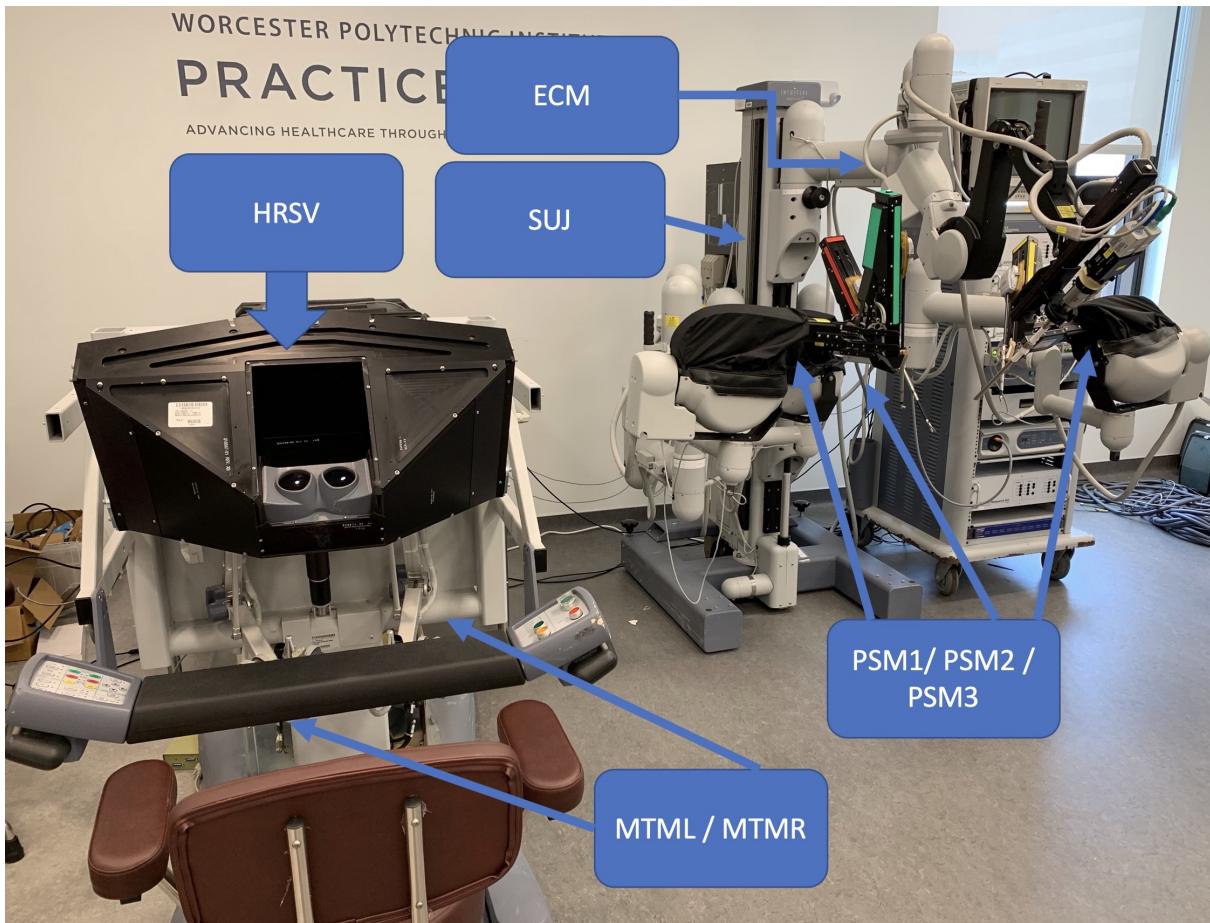


Figure 1.1: da Vinci surgical system

## 1.4 Document Conventions

If you have any further questions, please feel free to check the GitHub discussion board at <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/discussions>.

## 1.5 Intended Audience and Reading Suggestions

This is a tutorial for the starters of dVRK. Also, please check the Content list at the beginning for an overview.

## 1.6 Product Scope

This specification covers simple hardware and software setup. For advanced techniques, please check the websites given in the following section.

## 1.7 References

The whole specification is based on dVRK[1] and CTRK[2] papers. And their corresponding GitHub repositories are:

- da Vinci Research Kit (dVRK) ROS  
Link: <https://github.com/jhu-dvrk/dvrk-ros>
- Collaborative Robotics Toolkit (CTRK)  
Link: <https://github.com/collaborative-robotics/documentation/wiki>
- dVRK Wiki  
Link: <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/FirstSteps>

# Chapter 2

## Hardware Setup

This section is to introduce how to set up the dVRK in hardware level. This section would be helpful if you are a starter and the cables are unplugged.

### 2.1 Wire Connection

#### 2.1.1 Daisy Chain (Power cable)

This subsection is to set up the power system and integrate with the E-STOP. All the control boxes currently utilized have two connectors: one 4-pin connector and one 5-pin connect. A detailed wiring figure is shown in Figure 2.1:

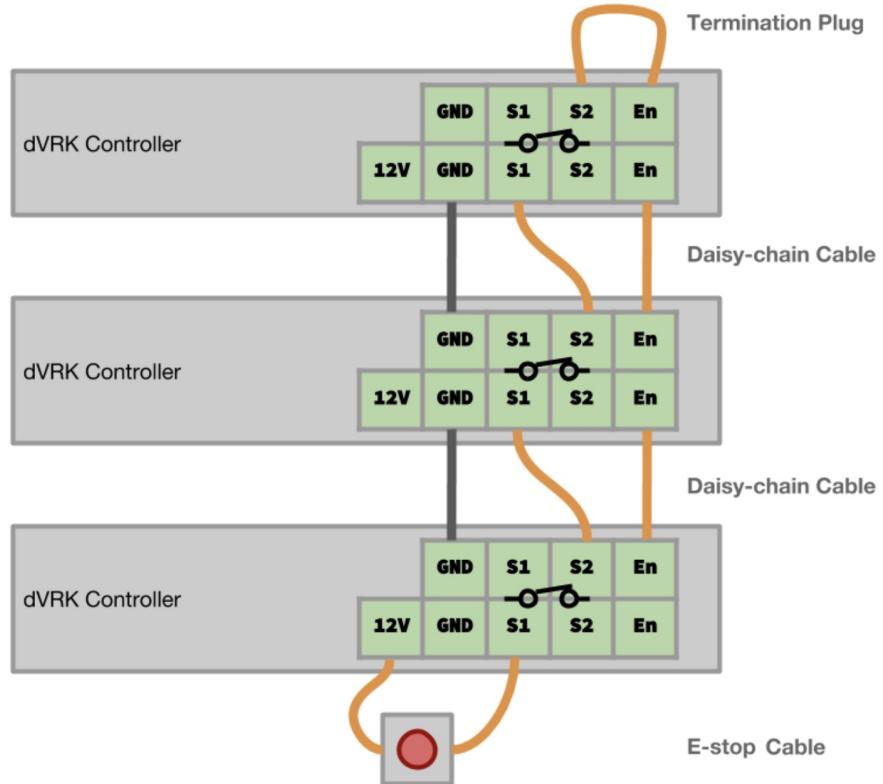


Figure 2.1: dVRK daisy chain

### 2.1.2 Firewire

All the firewire connections should follow the listed rules:

- input-output-input sequences shown in Figure 2.2
- the overall connection between the work station and the control boxes should be through either the total input (for our case, MTML control box) or total output (for our case, SUJ control box)
- the board ID should be continuous ascending or descending (AKA, 0 → 12 or 12 → 0)
- To follow the control box board ID sequence, our control box connection sequence is MTML → MTMR → ECM → PSM1 → PSM2 → PSM3 → SUJ

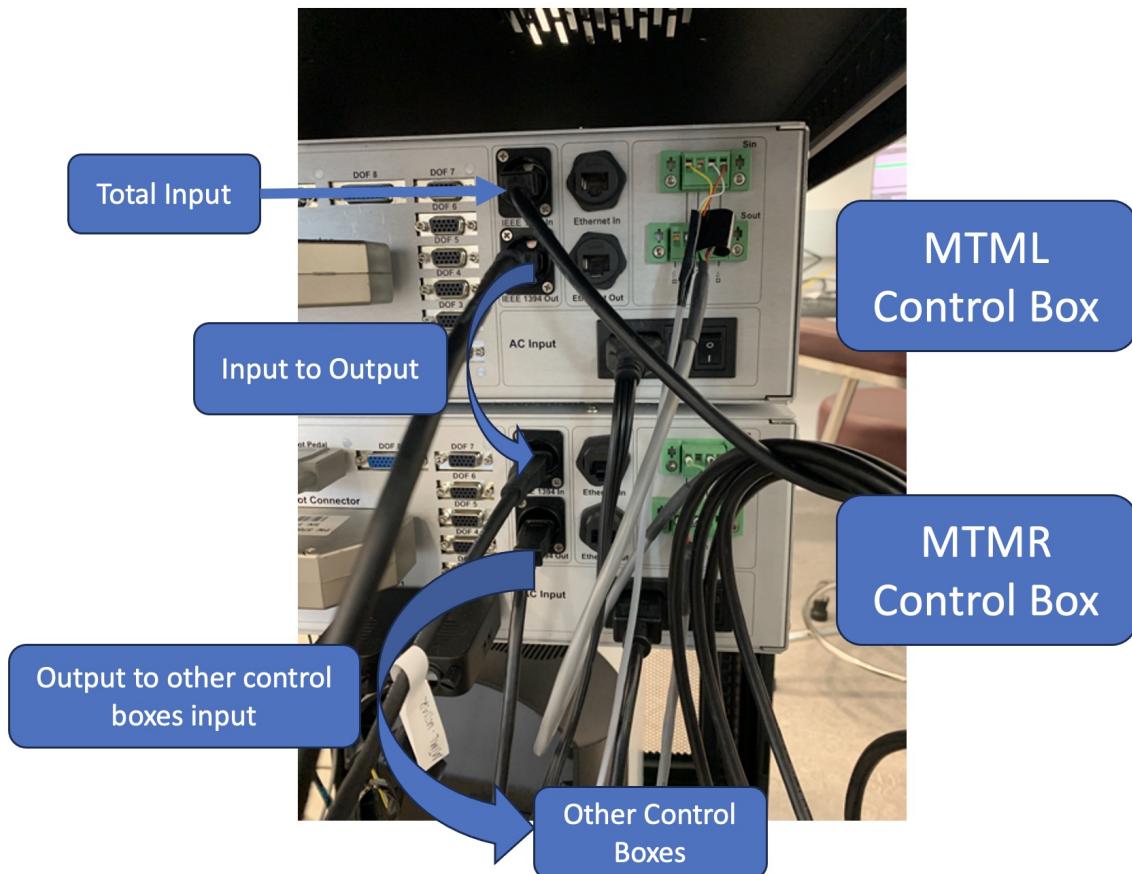


Figure 2.2: Firewire Connection Example

Also, base on the fact that most computers does not have firewire port any more, you may need to buy a firewire card<sup>1</sup>. For software level setup, please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Hardware#2-firewire> for more details. Generally, the software setup only needs to implement once for each computer.

### 2.1.3 Foot Pedal

The foot pedal is connected to MTMR control box as shown in Figure 2.3.

<sup>1</sup>Currently Installed Firewire Card Purchase link: <https://www.amazon.com/gp/product/B002S53IG8/>

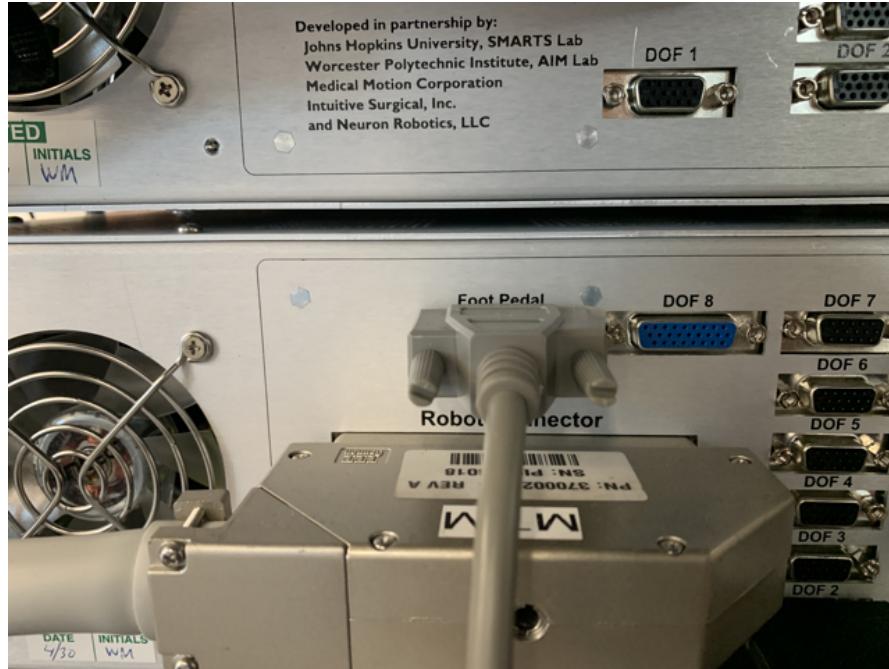


Figure 2.3: Foot Pedal Connection

If you need to make your own foot pedal, please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/FootPedals> for details.

#### 2.1.4 Endoscope Focus Controller

The endoscope focus controller is connected to MTMR control box as shown in Figure 2.4. Please make sure the cable color is also the same as shown in Figure 2.4.

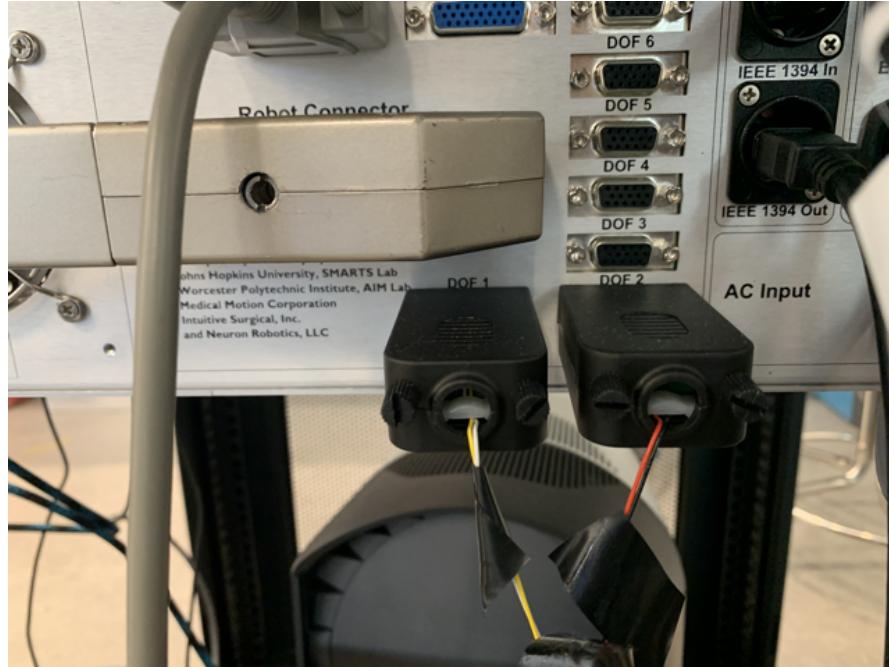


Figure 2.4: Endoscope Focus Controller Connection

The connection cable between dVRK control boxes and endoscope focus control box is a customized cable. For the details of cable construction and wiring instruction, please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Endoscope-Focus-Controller>.

### 2.1.5 Ethernet UDP

Even though we currently use firewire connection at WPI, Ethernet cable is also available for connections as shown in Figure 2.5. We highly recommend to use firewire connections among control boxes for both methods.

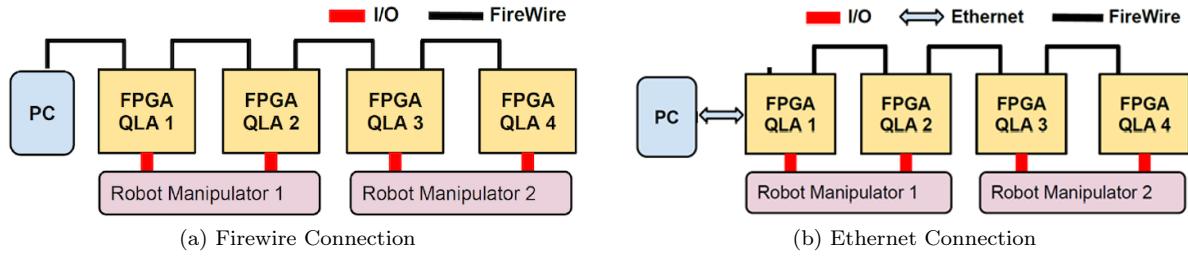


Figure 2.5: Connections Between PC and Control Boxes

If you are using Ethernet connection, please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/ControllerConnection#ethernet-udp> for network setup details. In addition, it may include some configuration JSON file modifications or run code with the option -pudp.

### 2.1.6 HRSV Video Pipeline

Please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Video-Pipeline> for an introduction on how video pipeline works.

The principle behind the HRSV Video Pipeline Setup is to regard the two internal monitors embedded in the HRSV as two external monitors for connection and put the corresponding endoscope images into the monitors.

For the internal monitors, we purchased some industrial monitors<sup>2</sup>. Those monitors fit the cavity at HRSV fairly well as shown in Figure 2.6:



Figure 2.6: HRSV monitor replacement

<sup>2</sup>Monitor Purchase link: <https://www.beetronics.com/15-inch-monitor-flush-4-3>

There are two common solutions for video streaming:

### SDI-to-USB3 Video Frame Grabber

This method is currently leveraged at WPI. We purchased two epiphan SDI-to-USB3 frame grabbers<sup>3</sup> shown in Figure 2.7. Those grabbers can convert the endoscope inputs to webcam inputs and thus it can be read by some ROS packages easily, such as ROS cv-camera package. Each camera require a separate frame grabber. Since each endoscope has two cameras, therefore at least two frame grabbers are required. If you would like to have a fixed connection via USB, it may require some basic knowledge on udev with Linux. Then, you may construct a customized udev rule for connection so that it can automatically detect the USB inputs with self-defined names.



Figure 2.7: Frame Grabber at WPI

### DeckLink Duo Frame Grabber

This solution is given by a subtopic under dVRK Wiki and currently utilized at Johns Hopkins University. Please check [https://github.com/jhu-dvrk/dvrk-ros/blob/devel/dvrk\\_robot/video.md](https://github.com/jhu-dvrk/dvrk-ros/blob/devel/dvrk_robot/video.md) for details.

<sup>3</sup>SDI-to-USB3 Frame Grabber Purchase link: <https://www.epiphan.com/products/avio-sdi/>

### 2.1.7 Endoscope

The endoscope needs to be connected to camera control units, focus control box and light source as shown in Figure 2.8.

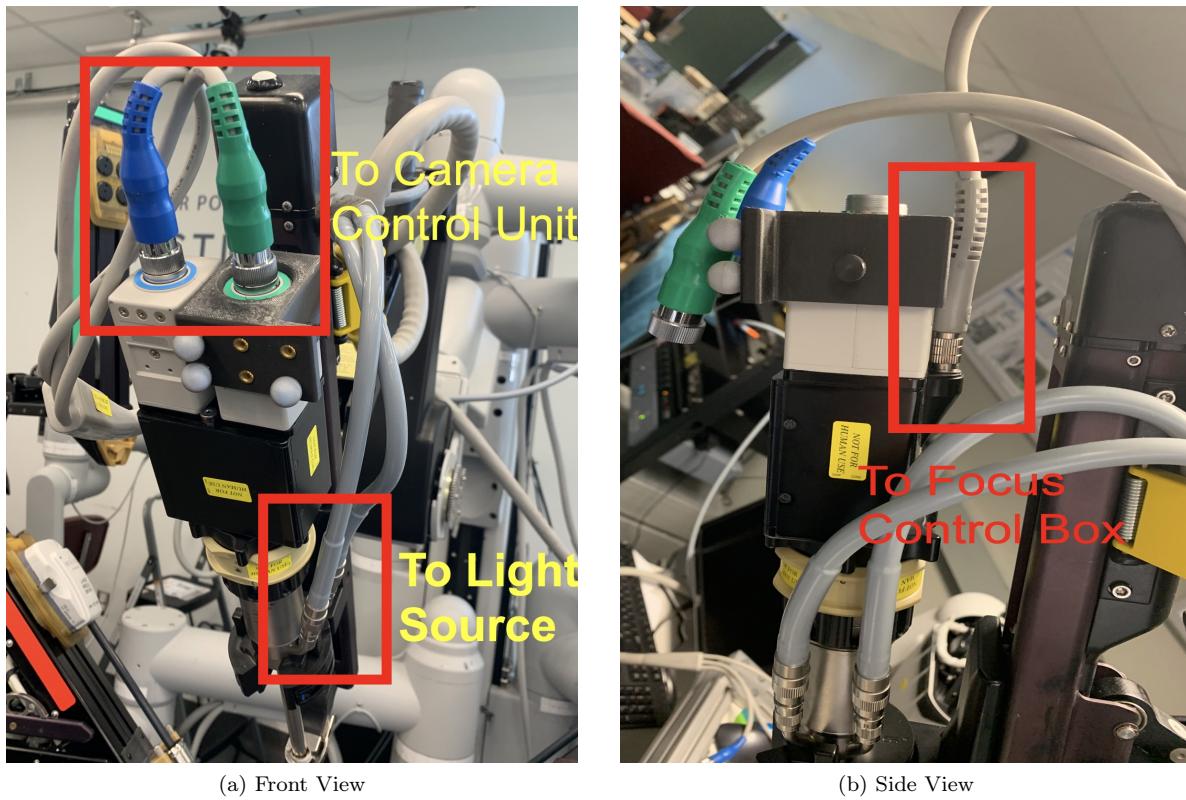


Figure 2.8: Endoscope

Camera control units, focus control box and light source are all located at the control tower as show in Figure 2.9.

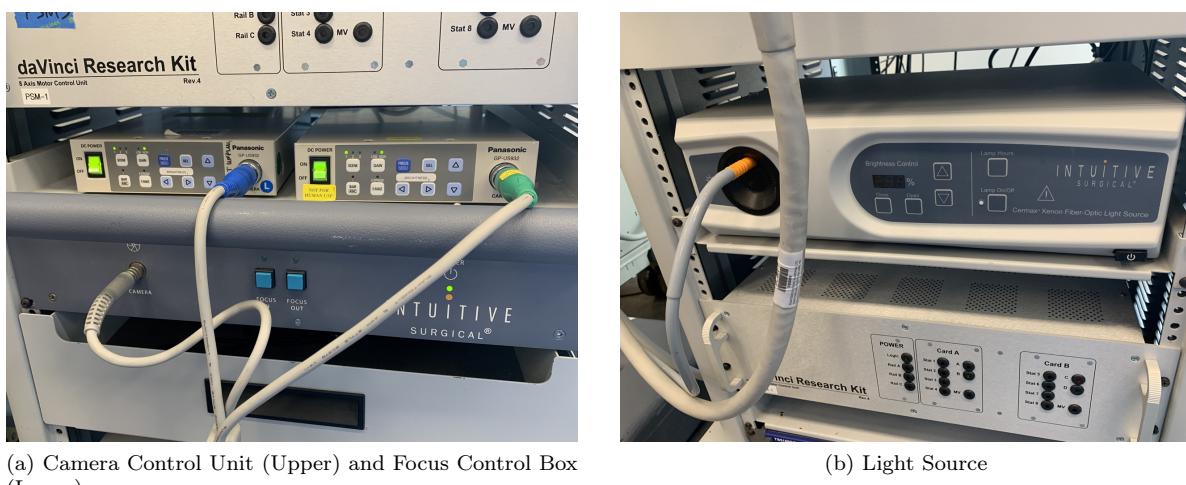


Figure 2.9: Endoscope Related Control Boxes on the Control Tower

### 2.1.8 Other Hardware

This section is for some hardware which has not been deployed at WPI dVRK setup.

#### Head Sensor

Head Sensor is a IR detector at HRSV outer shell as shown in Figure 2.10 . It can trigger the coag(mono) button as long as an optical obstacle was detected by the sensor. Please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/HeadSensor> for details.



Figure 2.10: HRSV Head Sensor

## 2.2 Robot Initialize and Power Check

Assume all the cables are connected correctly and all control boxes have been powered on.

### 2.2.1 Emergency Stop Check

Check the E-STOP and make sure it is released:



Figure 2.11: E-stop pushed (left) and released (right)

When finishing using, please make sure to push the ESTOP down before leaving for safety consideration.

## 2.2.2 Board ID Check

Open a new terminal, run the following command:

```
$ qladisp
```

If you manage to get the following figure and find 13 board for total, you may move forward to next step:

```
practicepoint@practicepoint:~$ qladisp
Usage: qladisp <board-num> [<board-num>] [-pP] [-b<r|w>] [-v] [-t]
      where P = port number (default 0)
            can also specify -pfw[:P], -peth:P or -pudp[:xx.xx.xx.xx]
      -br enables broadcast read/write
      -bw enables broadcast write
      -v displays full velocity feedback
      -t displays time information

Trying to detect boards on port:
ParseOptions: no option provided, using default fw:0
FirewirePort::Init: number of ports = 1
  Port 0: /dev/fw8, 14 nodes
FirewirePort::Init: successfully initialized port 0
Using libraw1394 version 2.1.2
FirewirePort::Init: successfully disabled cycle start packet
FirewirePort::InitNodes: base node id = ffc0
BasePort::ScanNodes: building node map for 13 nodes:
  Node 0, BoardId = 12, Firmware Version = 7
  Node 1, BoardId = 11, Firmware Version = 7
  Node 2, BoardId = 10, Firmware Version = 7
  Node 3, BoardId = 9, Firmware Version = 7
  Node 4, BoardId = 8, Firmware Version = 7
  Node 5, BoardId = 7, Firmware Version = 7
  Node 6, BoardId = 6, Firmware Version = 7
  Node 7, BoardId = 5, Firmware Version = 7
  Node 8, BoardId = 4, Firmware Version = 7
  Node 9, BoardId = 3, Firmware Version = 7
  Node 10, BoardId = 2, Firmware Version = 7
  Node 11, BoardId = 1, Firmware Version = 7
  Node 12, BoardId = 0, Firmware Version = 7
BasePort::ScanNodes: found 13 boards
```

Figure 2.12: expected terminal output for qladisp at WPI

`qladisp` command can actually accomplish some additional tasks other than initialization confirmation. Please check <https://github.com/jhu-cisst/mechatronics-software/wiki/Example-Programs> for details.

## 2.2.3 Robot Initialization

If you have released the ESTOP (at the begining or recovering from emergency stop status), we highly recommend you to run the following command in a new terminal:

```
$ qlacommand -c close-relays
```

This command is to enable the power of the control boxes. At the same time, you should hear the sound of activated fans. If not, please go back to check the cable connections, power and E-stop. This manual activation of power helps if the SUJ doesn't power on properly sometimes.

## 2.3 Calibration

Generally speaking, **you only do the calibration at the first time using the robot or when necessary (e.g. some arm misalignment problems [encoder reading and potentionmeter reading are inconsistent]).** You may skip this step if you have done the calibration in the past. We only introduce how to do the simplest current calibrations. For further calibrations, please refer to dVRK Wiki: <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Calibration>.

Firstly, you need to direct to the dVRK configuration folder (Please check section 3.1 for more information).

Before calibration, you should ensure that your relays are closed. If not, open a new terminal and run following command:

```
$ qlacommand -c close-relays
```

At the same time, you should hear the sound of activated fans. If not, please go back to check the cable connections, power and E-STOP.

Alternatively, you can also generate new xml for the given .cal from Intuitive Surgical as instructed in <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/XMLConfig>.

### 2.3.1 SUJ Current Calibration

Go back to the terminal which is directed to the dVRK configuration folder and run:

```
$ sawRobotI01394CurrentCalibration -c sawRobotI01394-SUJ.xml
```

Follow the instruction in the code and replace the `sawRobotI01394-SUJ.xml` with the calibrated xml file.

For advanced SUJ calibration (usually only need to do once), please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/SUJ>.

### 2.3.2 PSM Current Calibration

**Firstly, make sure your detachable surgical tools are not installed on the PSM when calibration**

Then, go back to the terminal which is directed to the dVRK configuration folder and run:

```
$ sawRobotI01394CurrentCalibration -c sawRobotI01394-PSM2-29996.xml
```

Follow the instruction in the code and replace the `sawRobotI01394-PSM2-29996.xml` with the calibrated xml file.

For the other PSMs, they share the identical procedures. You only need to modify the PSM number and corresponding series number.

### 2.3.3 ECM Current Calibration

Open a new terminal, direct to the dVRK configuration folder and run:

```
$ sawRobotI01394CurrentCalibration -c sawRobotI01394-ECM-25975.xml
```

Follow the instruction in the code and replace the `sawRobotI01394-ECM-25975.xml` with the calibrated xml file. You only need to modify your ECM serial number if you are not using WPI clinical da Vinci.

For ECM advanced calibration(usually only need to do once), please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/ECM>.

### 2.3.4 MTM Current Calibration

Open a new terminal, direct to the dVRK configuration folder and run:

```
$ sawRobotI01394CurrentCalibration -c sawRobotI01394-MTML-33100.xml
```

Follow the instruction in the code and replace the `sawRobotIO1394-MTML-33100.xml` with the calibrated xml file.

For the other MTMs, they share the identical procedures. You only need to modify the MTML/R and corresponding series number.

For MTM advanced calibration, please check <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Calibration#gripper-on-mtms>. For WPI clinical da Vinci, you don't have to do this calibration since we have mechanically lubricate and adjust some detachable surgical tools to make them grasp objects tightly.

## 2.4 Tool Deployment

Tool deployment includes two parts: tool detection and tool installation.

For tool detection, there are three models:

- Automatic: let the tool pedal auto-detect the tool type from the chip embedded in the tool
- Manual: select tool type manually
- Fixed: pre-selected fixed tool type for the arm

For tool installation, the major concern is the alignment of the tool and the tool pedal as shown in Figure 2.13.



Figure 2.13: Alignment of the rotating pans between the tool pedal and the tool

For our cases at WPI, we are currently using Manual model for tool detection. Therefore, please select your preferred tool and install it on the PSM before you activate da Vinci system. You may put the tool pedal around the home position shown in Figure 2.13 for an easier installation. If you select to use Automatic model, you may power on the system firstly and home the robot, then you can attach the tool to the arm and the tool information will be automatically detected. Nevertheless, if you have a customized tool, Manual model is required.

## Chapter 3

# Software and Applications

### 3.1 dVRK Configuration Folder Path

Before starting working on this section, you need to accomplish the construction of ROS work space. The work space name shown in the following sections are the names currently used in WPI local environment. You are free to modify the name based on your personal preference.

#### 3.1.1 ROS 1

For ROS 1, an instruction of building ROS work space is shown in the following link <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/CatkinBuild>.

#### Master Branch

For sawIntuitiveResearchKit repository in master branch, you can run following commands in a new terminal to be directed to the dVRK configuration folder path:

```
cd ~/catkin_ws_dvrk/src/cisst-saw/sawIntuitiveResearchKit/share
```

In this folder, you can see all shared configurations from different universities.

For instance, at WPI, you would need to head to `wpi_test` folder for more specific configuration JSON files.

You may refer to JHU configuration JSON files to create your own JSON files.

#### Develop Branch

For sawIntuitiveResearchKit repository in master branch, you can run following commands in a new terminal to be directed to the dVRK configuration folder path:

```
cd ~/catkin_ws_dvrk-devel/src/cisst-saw/sawIntuitiveResearchKit/share
```

In this folder, you can see all shared configurations from different universities.

For instance, at WPI, you would need to head to `wpi_test` folder for more specific configuration JSON files.

You may refer to JHU configuration JSON files to create your own JSON files.

### 3.1.2 ROS 2

For ROS 2, an instruction of building ROS work space is shown in the following link <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/BuildROS2>.

For the configuration folder, you may need to work on your own to create your own configuration JSON files, which requires some knowledge of the software architecture. You may also refer to JHU ROS2 configuration JSON files in [https://github.com/dvrk-config/dvrk\\_config\\_jhu](https://github.com/dvrk-config/dvrk_config_jhu).

For instance, at WPI, you can run following commands in a new terminal to be directed to the dVRK configuration folder path:

```
cd ~/ros2_ws/install/dvrk_config_wpi
```

### 3.1.3 Understand JSON Configuration File

You can check the existed configuration files as reference. Moreover, you can check this link <https://dvrk.lcsr.jhu.edu/documentation/schemas/v2.1/dvrk-console.html> to learn about the meanings of the parameters.

### 3.1.4 Switch between ROS 1 and ROS2

Generally speaking, source both ROS1 and ROS2 at bashrc is not recommended at Ubuntu 20.04. Therefore, as a solution, you can source one of the two ROS version at one time as shown in Figure 3.1.

```
137 # ROS 1
138 source /opt/ros/noetic/setup.bash
139 source /home/practicepoint/daVinci_simulation/src/ambf/build-devel/setup.bash
140 source /home/practicepoint/catkin_ws_dvrk-devel-devel/setup.bash
141 source /home/practicepoint/daVinci_simulation/devel/setup.bash
142
143 # ip=$(hostname -I | awk '{print $NF}')
144 # export ROS_IP=$ip
145 # export ROS_MASTER_URI=http://$ip:11311
146
147 # # export ROS_MASTER_URI=http://130.215.192.166:11311
148 # # export ROS_IP=130.215.192.166
149
150 # # ROS 2
151 # source /opt/ros/galactic/setup.bash
152 # export RTI_LICENSE_FILE=/home/practicepoint/rti_license/rti_license.dat
153 # source /opt/rti_connex_dds-6.1.2/resource/scripts/rtisetenv_x64Linux4gcc7.3.0.bash
154 # export CONNEXTDDS_DIR=${NDDSHOME}
155 # source /home/practicepoint/ross2_connex_dds/install/setup.bash
156 # source /home/practicepoint/ross2_ws/install/setup.bash
```

Figure 3.1: sections in the bashrc file for ROS1 and ROS2 source

- ROS 1 : Comment line 151, 155 and 156; Uncomment line 138-141
- ROS 2: Uncomment line 151, 155 and 156; Comment line 138-141

## 3.2 Activate dVRK

### 3.2.1 ROS 1

Open a new terminal, run following command to run ROS:

```
$ roscore
```

Create a new terminal which is directed to the dVRK configuration folder as shown in section 3.1.1 and run:

```
$ rosrun dvrk_robot dvrk_console_json -j console-[code name].json -p [timestep]
```

where

$$timestep = \frac{1}{desired\ frequency}$$

Here is the table introducing different code names:

Code Name	Purpose
ECM, MTML/R, PSM 1/2/3	Activate corresponding arms
ECM-MTML-MTMR-Teleop	Teleoperate ECM with two MTMs
MTML-MTMR	Activated two MTMs usually use when teleoperation in AMBF simulation [3]
SUJ	Activate SUJ to manually move PSM 1&2 and ECM with respect to SUJ
SUJ-PSM3	Activate SUJ and PSM3 so all PSMs can move
SUJ-ECM-PSM	Activate SUJ, ECM and PSM3 so that all arms can move
SUJ-MTMR-PSM1-Teleop	Teleoperate PSM1 with MTMR
SUJ-ECM-MTMR-PSM1-MTML-PSM3-Teleop	Teleoperate two arms with two MTMs utilized in the suture user study
SUJ-ECM-MTMR-PSM1-MTML-PSM2-PSM3-Teleop	Full system activated with teleoperation of all three PSMs enabled

Table 3.1: code name table

After running the above code, you should enter a GUI looks like:

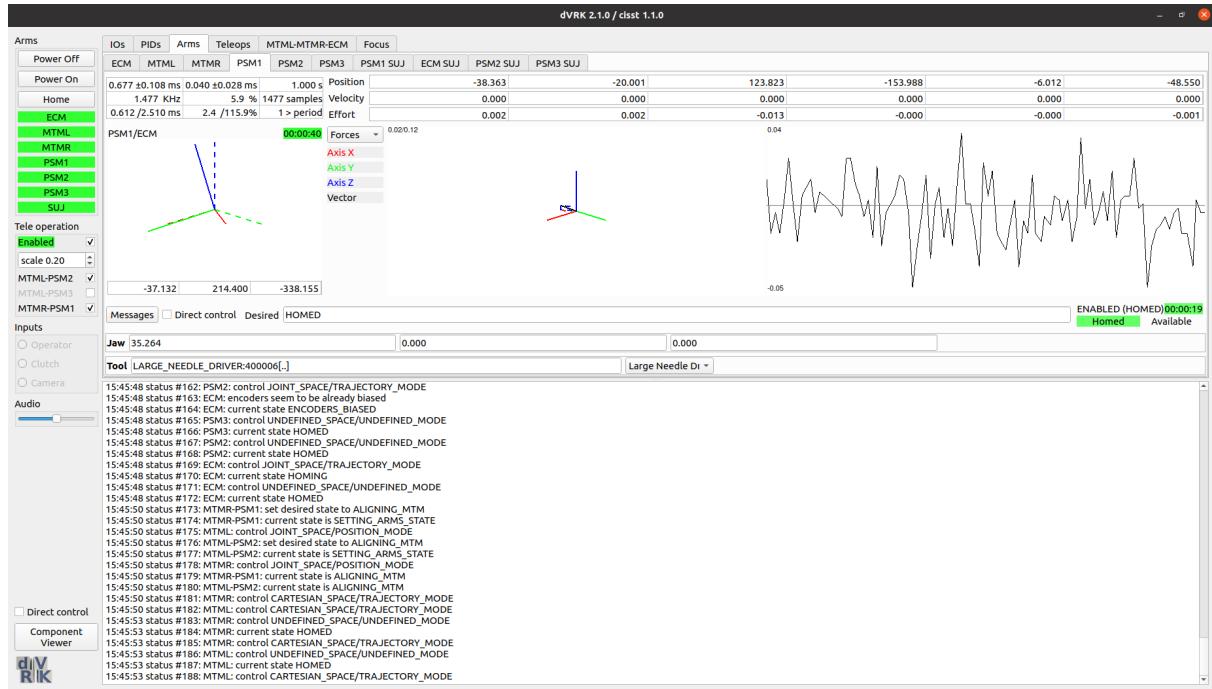


Figure 3.2: expected GUI

Click power on and then click home. If all the blocks turn out to be green, then you are good to move forward.

### 3.2.2 ROS 2

For dVRK in ROS 2, most of the procedures are identical to the procedures for ROS 1. The major differences are:

- different dVRK configuration folders as shown in section 3.1.1
- different commands for running the ROS consoles and no `roscore`

For ROS 2, please run the following commands when you are in the dVRK configuration folder:

```
$ ros2 run dvrk_robot dvrk_console_json -j console-[code name].json -p [timestep]
```

The expected GUI is the same as the one shown in Figure 3.2.

### 3.3 HRSV Activation

If you are using DeckLink Duo frame grabber, then please check the instruction in [https://github.com/jhu-dvrk/dvrk-ros/blob-devel/dvrk\\_robot/video.md](https://github.com/jhu-dvrk/dvrk-ros/blob-devel/dvrk_robot/video.md). This method is using `gscam` and `gstreamer` to obtain the image and publish to ROS image topics.

If you are using SDI-to-USB3 frame grabber, you can also use ROS `cv_camera` package to obtain the images and publish to ROS image topics. You may find the ROS launch file following the baseline instruction at WPI local environment. Alternatively, you can create your own launch file referring to the launch file shown in Figure 3.3:

```

1   <launch>
2       <node pkg="cv_camera" type="cv_camera_node" name="cv_camera_left_0" output="screen">
3           <param name="device_id" value="0"/>
4       </node>
5
6       <node pkg="rqt_image_view" type="rqt_image_view" name="image_view_0" args="/cv_camera_left_0/image_raw"/>
7
8
9       <node pkg="cv_camera" type="cv_camera_node" name="cv_camera_right_2" output="screen">
10      <param name="device_id" value="2" />
11      </node>
12
13      <node pkg="rqt_image_view" type="rqt_image_view" name="image_view_2" args="/cv_camera_right_2/image_raw"/>
14
15
16  </launch>
```

Figure 3.3: ROS launch file example

As shown in line 3 and 10, the left camera is `/dev/video0` and the right camera is `/dev/video2`. Nevertheless, we have not forced a udev rule to identify the USB cables for cameras at WPI. Therefore, if you reconnect the USB cables, you may have a chance to reverse the video ID. To confirm the camera IDs are correct, you may need to run the ROS launch firstly and push the `bar` code button on the camera control unit of the left camera. If the shown camera left figure turns to be a barcode, then you are good to go. Otherwise, you can switch the codes at line 3 and 10.

Eventually, make sure that you drag the image consoles to the HRSV extended monitors.

# Chapter 4

## dVRK Baseline at WPI

### 4.1 Hardware Check

Please follow the instruction in chapter 2. Check all the hardware are connected properly and powered on. Release the ESTOP before running the full system of dVRK.

In addition, you may need to check the camera ID if you are worried whether your flipped the camera. Generally, the left camera is `/dev/video0` and the right camera is `/dev/video2`. Nevertheless, we have not forced a udev rule to identify the USB cables for cameras at WPI. Therefore, if you reconnect the USB cables, you may have a chance to reverse the video ID. To confirm the camera IDs are correct, you can visualize the images via VLC using capturing device firstly and push the bar code button on the camera control unit of the left camera. If the shown camera left figure turns to be a barcode, then you are good to go. Otherwise, you can switch the codes at line 3 and 10.

### 4.2 Software Run

#### 4.2.1 Connection Check

Firstly, make sure all control boxes are connected to the computer. You may run following command in a terminal to check:

```
$ qladisp
```

If you can see 13 boards are connected as shown in Figure 2.12, then it means that all connections work fine. Then, you can run following command in the terminal:

```
$ qlacommand -c close-relays
```

This command is to enable the power of the control boxes. At the same time, you should hear the sound of activated fans. If not, please go back to check the cable connections, power and E-stop. This manual activation of power helps if the SUJ does not power on properly sometimes.

#### 4.2.2 ROS 1

Firstly, you need to activate ROS master via running following command in the terminal:

```
$ roscore
```

Then, you need to enable the endoscope image system with ROS 1. At the local environment, please run following command in a new terminal:

```
$ cd ~/cristina_ws/src/irb/launch
$ roslaunch camera.launch
```

Alternatively, if the camera launch file cannot be found or you would like to make a customized camera launch file. Then, please go ahead to take Figure 3.3 as a reference or template.

Eventually, you can head to the dVRK configuration folder as shown in chapter 3 section 3.1.1, then you can run the following command in the terminal to enable the full system teleoperation with 1000Hz refreshing rate:

```
$ rosrun dvrk_robot dvrk_console_json -j console-SUJ-ECM-MTMR-PSM1-MTML-PSM2-PSM3-Teleop.json
-p 0.001
```

For suturing user study, please go with following command (also 1000Hz):

```
$ rosrun dvrk_robot dvrk_console_json -j console-SUJ-ECM-MTMR-PSM1-MTML-PSM3-Teleop.json
-p 0.001
```

If you only run with AMBF simulation, please go with following command:

```
$ rosrun dvrk_robot dvrk_console_json -j console-MTML-MTMR.json -p 0.001
```

### 4.2.3 ROS 2

The endoscope image system with ROS 2 is currently not available yet. Therefore, you can only go with VLC to visualize the endoscope images.

Then, you can go to the dVRK configuration folder for ROS 2 as shown in chapter 3 section 3.1.2 and run following command in the terminal to enable the full system teleoperation with 1000 Hz refreshing rate:

```
$ ros2 run dvrk_robot dvrk_console_json -j console-SUJ-ECM-MTMR-PSM1-MTML-PSM2-PSM3-Teleop.json
-p 0.001
```

For two arm teleoperation for suturing, please go with following command:

```
$ ros2 run dvrk_robot dvrk_console_json -j console-SUJ-ECM-MTMR-PSM1-MTML-PSM3-Teleop.json
-p 0.001
```

# Chapter 5

# How to Run Teleoperation

## 5.1 Move SUJ Manually

There are four arms attached to the SUJ: ECM, PSM1, PSM2 and PSM3. Except for PSM3, all the other arms shared the same moving technique.

### 5.1.1 PSM1 and PSM2

As long as you have activated the SUJ, you can move the base of PSM 1 and 2 freely automatically. For example, if you are under dVRK configuration folder and run following commands:

```
$ rosrun dvrk_robot dvrk_console_json -j console-SUJ-***.json
```

Then push the button located at the SUJ PSM handle shown in Figure 5.1, you are free to move the PSM 1 and 2 around.



Figure 5.1: SUJ PSM handle

### 5.1.2 ECM

For moving the base of ECM, you have to enable the power of ECM firstly. For example, if you are under dVRK configuration folder and run following commands:

```
$ rosrun dvrk_robot dvrk_console_json -j console-SUJ-ECM-***.json
```

Then push the button located at the SUJ ECM handle shown in Figure 5.2, you are free to move the base of ECM around.



Figure 5.2: SUJ ECM handle

### 5.1.3 PSM3

For moving the base of PSM3, you have to enable the power of PSM 3 firstly. For example, if you are under dVRK configuration folder and run following commands:

```
$ rosrun dvrk_robot dvrk_console_json -j console-SUJ-***-PSM3-***.json
```

Then push the button located at the SUJ PSM3 handle shown in Figure 5.3, you are free to move the base of PSM3 around.

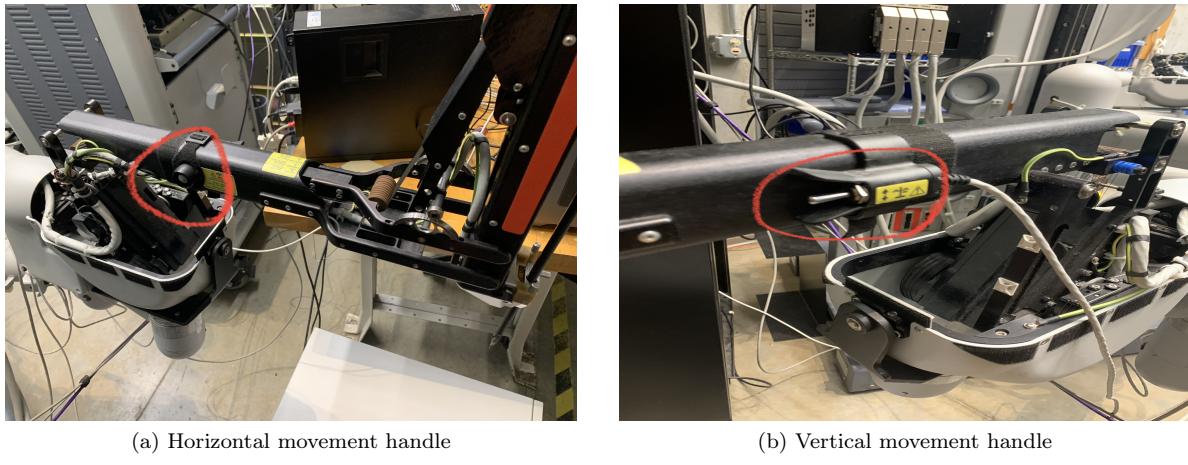


Figure 5.3: SUJ PSM3 handle

## 5.2 Move Robot Arms Manually

In section 5.1, we introduce how to move the base of robot arms via SUJ. In this section, we would like to introduce how to move the robot arms themselves. The pre-requirement of moving the arms is to enable the power of corresponding arm. Then, you can find a white button at the top of the arms as shown in Figure 5.4. When you press this button, you can move the arms freely.

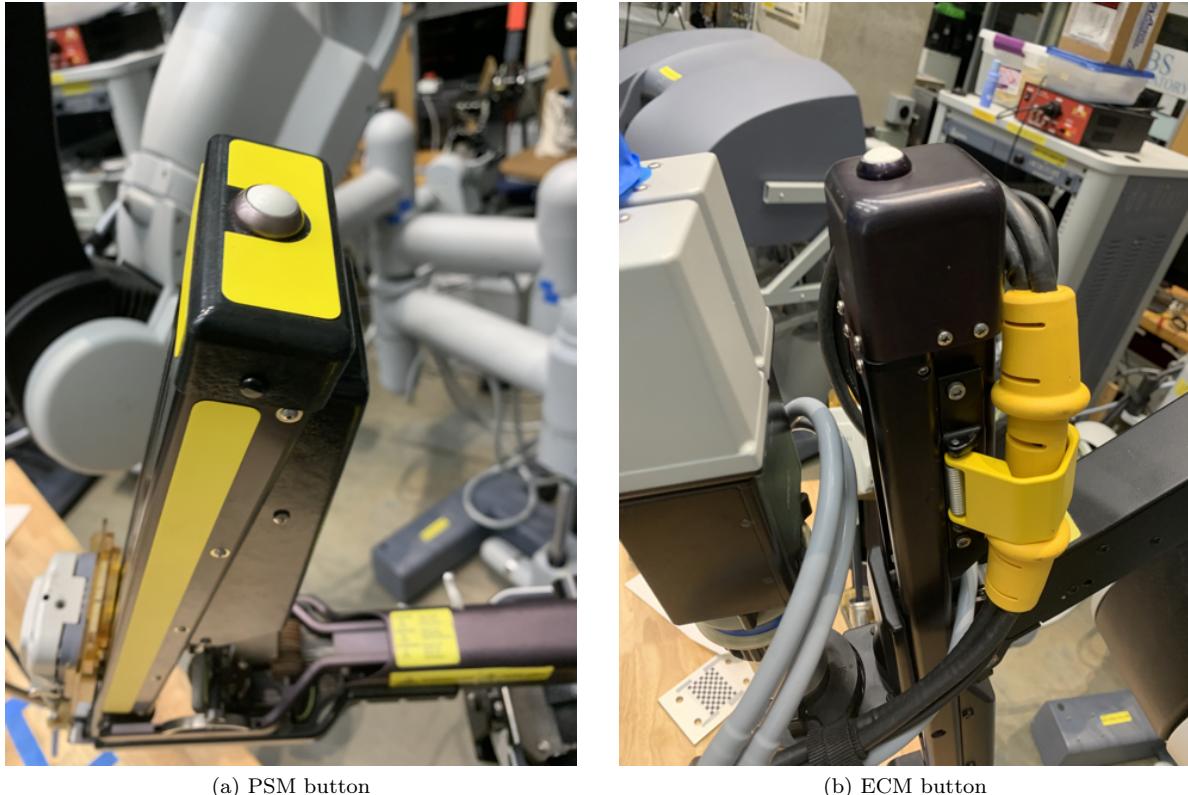


Figure 5.4: Buttons on different robot arms

### 5.3 Foot Pedal

You may set up your own foot pedal following the instruction in the given link: <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/FootPedals>. Alternatively, you may obtain a foot pedal shown in Figure 5.5 with the hardware of dVRK.



(a) WPI foot pedal



(b) JHU foot pedal

Figure 5.5: Foot pedal come with dVRK

Assume that you have ran the full system of dVRK with teleoperation enabled.

#### 5.3.1 Endoscope Focus Modification

If you would like to modify the focus of the endoscope, you can use the button in the middle of the foot pedal. It is a straightforward mapping from focus control box to the foot pedal button. The pre-requirement for this operation is the cable connection discussed in section 2.1.4 in chapter 2.

#### 5.3.2 Teleoperate PSM

Each MTM is corresponded to one PSM. The match between MTMs and PSMs is shown as following:

- MTMR - PSM1
- MTML - PSM2 or PSM3

The teleoperation is an one-to-one mapping. It maps your movements at MTM to PSM tool tip with a given scale (default to be 0.2).

The operation of the foot pedal for PSM teleoperation is shown as following:

- hold MONO/COAG pedal: enable the teleoperation
- hold MONO/COAG pedal and CLUTCH pedal: Move the MTM freely without moving PSM, leverage to extend your work space.
- press CLUTCH pedal without holding MONO/COAG pedal: switch the control pairs for PSM2 and PSM3 if enabling all three PSMs.

It may take some time to find the inverse kinematic of MTM if you move too fast or move around a singularity. At those times, the teleoperation is disabled for a while. If this happens, please hold MONO/COAG pedal and wait for MTM to self-adjust to the desired pose. If waiting doesn't fix the problem, you can also restart the console to fix it.

### 5.3.3 Teleoperate ECM

Two MTMs are corresponded to one ECM. Imagine you are operating a wheel with two hands, the teleoperation follows this manner.

The operation of the foot pedal for ECM teleoperation is shown as following:

- hold MONO/COAG pedal and CAMERA pedal: enable the teleoperation
- hold MONO/COAG pedal and CLUTCH pedal: Move the MTM freely without moving ECM, leverage to extend your work space or reset MTM poses.

Please avoid any sudden movements. If your movements are too swift, the ECM would be hard to follow and trigger a shutdown because of the inconsistency between the encoder and the potentiometer.

---

# Chapter 6

## Common Q & A

### 6.1 Cannot Detect the Board

#### 6.1.1 All Boards not Detected

Firstly, please ensure that all control boxes connection and hardware setup are valid following the instruction in chapter 2.

Then, unplug the firewire from the desktop. Wait for 5 seconds. Replug the firewire back to the desktop. Wait for 5 seconds. Then check the board detection with running following command in a new terminal:

```
qladisp
```

If this operation cannot solve the problem, please check the discussion in <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/discussions> or create a new issue in this discussion board.

#### 6.1.2 Some Boards not Detected

If you can detect some of the boards, please firstly check the board ID and refer to Table 1.1 to find out the first control box which is not detected. Then, unplug the firewire from the control box. Wait for 5 seconds. Replug the firewire back to the control box. Wait for 5 seconds. Then check the board detection with running following command in a new terminal:

```
qladisp
```

If this operation cannot solve your issue, you may open the control box to check whether the internal cables and FPGA/QLA board work or not. If one of them is broken, you may replace it if you have some spare components.

If all described operations still cannot solve your problem, please check the discussion in <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/discussions> or create a new issue in this discussion board.

### 6.2 Teleoperation Stuck

Teleoperation stuck means that you can still move MTMs but the PSMs are not following under teleoperation mode.

Firstly, you can check the GUI console to ensure everything is still working. If not, simply click power on, then click home and check Enabled under Tele operation box can save the problem. Alternatively, you can also close the dVRK robot console and restart in the terminal. I will list some tricks for recovering from the stuck status.

### 6.2.1 Trick 1

Hold MONO/COAG pedal and wait for MTM to stabilize. Then you can attempt to recover the teleoperation for PSMs.

### 6.2.2 Trick 2

Release all pedals and buttons on the foot pedal. Wait for 5 seconds. Then, hold MONO/COAG button and attempt to open and close the MTM grippers. It may allow you recover your teleoperation.

### 6.2.3 Trick 3

If you can visually check that the PSM is in a singularity or in a strange pose. You can manually move your PSM back to a regular pose via the instruction in section 5.2 in chapter 5. This may help you recover your teleoperation.

### 6.2.4 Trick 4

Close and restart the dVRK robot GUI console. Reboot should always be helpful.

## 6.3 MTM Misalignment

Sometimes, the MTM grippers can be misaligned when you home the MTMs as shown in Figure 6.1.



(a) Misaligned grippers



(b) Aligned grippers

Figure 6.1: MTM Misalignment

Sometimes, a current calibration as instructed in section 2.3.4 in chapter 2 can solve the problem.

Nevertheless, if a simple current calibration cannot solve the problem. We can use some advance techniques to solve this issue as shown in <https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Calibration#calibrating-scales>.

Taking MTML as an example, run following command in a new terminal under the path of dVRK configuration folder:

```
rosrun dvrk_robot dvrk_console_json -j console-MTML.json -i ../ros-io-MTML.json -C
```

It will enable a dVRK GUI console and click **home** button.

Then, open a new terminal under the same folder path and run:

```
rosrun dvrk_robot dvrk_calibrate_potentiometers.py -t scale -a MTML -c sawRobotI01394-MTML-33100.xml
```

Make sure you have enough free space for MTM to run this code. Then, just follow the instructions in the terminal.

After the scale calibration, your problem supposes to be solved.

# Bibliography

- [1] Peter Kazanzides et al. “An open-source research kit for the da Vinci® Surgical System”. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 6434–6439.
- [2] Yun-Hsuan Su et al. “Collaborative Robotics Toolkit (CRTK): Open software framework for surgical robotics research”. In: *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2020, pp. 48–55.
- [3] Adnan Munawar et al. “Open Simulation Environment for Learning and Practice of Robot-Assisted Surgical Suturing”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3843–3850.