Automated Control of External Ventricular Drain for Neuro-ICU

A Major Qualifying Project (MQP) Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements
for the Degree of Bachelor of Science in

Robotics Engineering
Computer Science
Electrical and Computer Engineering
Biomedical Engineering
Mechanical Engineering

By:

Matthew Duncan
Yujie Guo
Haotian Liu
Haoran Zhang

Project Advisors:

Christopher Nycz
Haicong (Kai) Zhang
Ziming Zhang

Sponsored By:

WPI Alumni: Mr. David Fitte

Date: April 2024

Abstract

Treating traumatic brain injury, subarachnoid hemorrhages, and other neurological conditions is a process that requires constant intervention from nurses and allows for little rest for patients. One symptom seen often in these cases is increased intracranial pressure (ICP). External ventricular drains (EVDs) are devices used to divert excess cerebrospinal fluid (CSF) from a patient's skull in order to manage a patient's ICP. These devices allow nurses to control the drainage rate of a patient's CSF to maintain a constant ICP. The ICP is set by physically leveling the EVD with a reference point (Kocher's point) on the patient's head. However, existing EVDs need frequent re-leveling due to patient movements, leading to constant intervention by hospital staff and disruption for the patient. Our project automates the movement of the EVD to improve the accuracy of ICP measurements and control. Our system provides real-time alerts to caregivers when changes occur and demonstrates accurate head motion tracking of targeted patients. This ensures automated alignment regardless of patient movement, which minimizes staff intervention and enhances patient comfort.

Acknowledgements

Our team would like to thank Mr. David Fitte, this project would not have been possible without his generous donation. We would also like to thank Professor Christopher Nycz for providing our team with constant feedback and support throughout the year.

Contents

List of Tables

List of Figures

1 Introduction

Patients with acute neurological conditions that affect cerebrospinal fluid (CSF) dynamics, such as hydrocephalus, intracranial hypertension, and intraventricular hemorrhage [1] often require the insertion of an external ventricular drain (EVD). An EVD is a device used to drain cerebrospinal fluid (CSF) from the ventricles of a patient's brain to relieve pressure or to remove blood and debris following hemorrhage. Insertion of an EVD is a common and important lifesaving procedure encountered frequently in the neurological intensive care unit [2]. Once these devices are inserted, management becomes a critical component of care. The management of an EVD requires constant care to prevent infections, ensure accurate measurements of intracranial pressure (ICP), and manage CSF drainage effectively. Existing EVDs operate based on hydrostatic forces, which means they need to be aligned with a particular point on a patient's skull in order to drain CSF at the appropriate rate. This means that every time a patient sits up or turns in their bed, the drainage rate is affected and an alarm will be triggered. To keep the drainage as consistent as possible, patients with an EVD remain on bed rest while the EVD remains in place and are asked to move as little as possible. This system significantly constrains the patient's mobilization [3] and puts a lot of strain on the nurses. Motivated by this, our project aims to allow patients to reposition in their beds and regain some mobility by creating an automated adjustment system. The project seeks to provide a platform for automating an EVD system that can be used for future testing and consider how this advancement could be used in a clinical setting.

Our system (Fig. 1) is structured around four principal objectives to enhance the management of EVDs through automation technology. The first objective is to design a prototype that is informed and inspired by the existing solutions in this field and integrates a linear actuator to reposition the EVD. The second objective involves designing a vision-based head motion tracking method to effectively measure the displacement of a patient's head. The third objective is the implementation of a feedback control algorithm to tell our linear actuator system how to move. The final objective involves conducting comprehensive validation experiments to evaluate the functionality of each component within the system. In the following subsections, each objective will be introduced in detail.

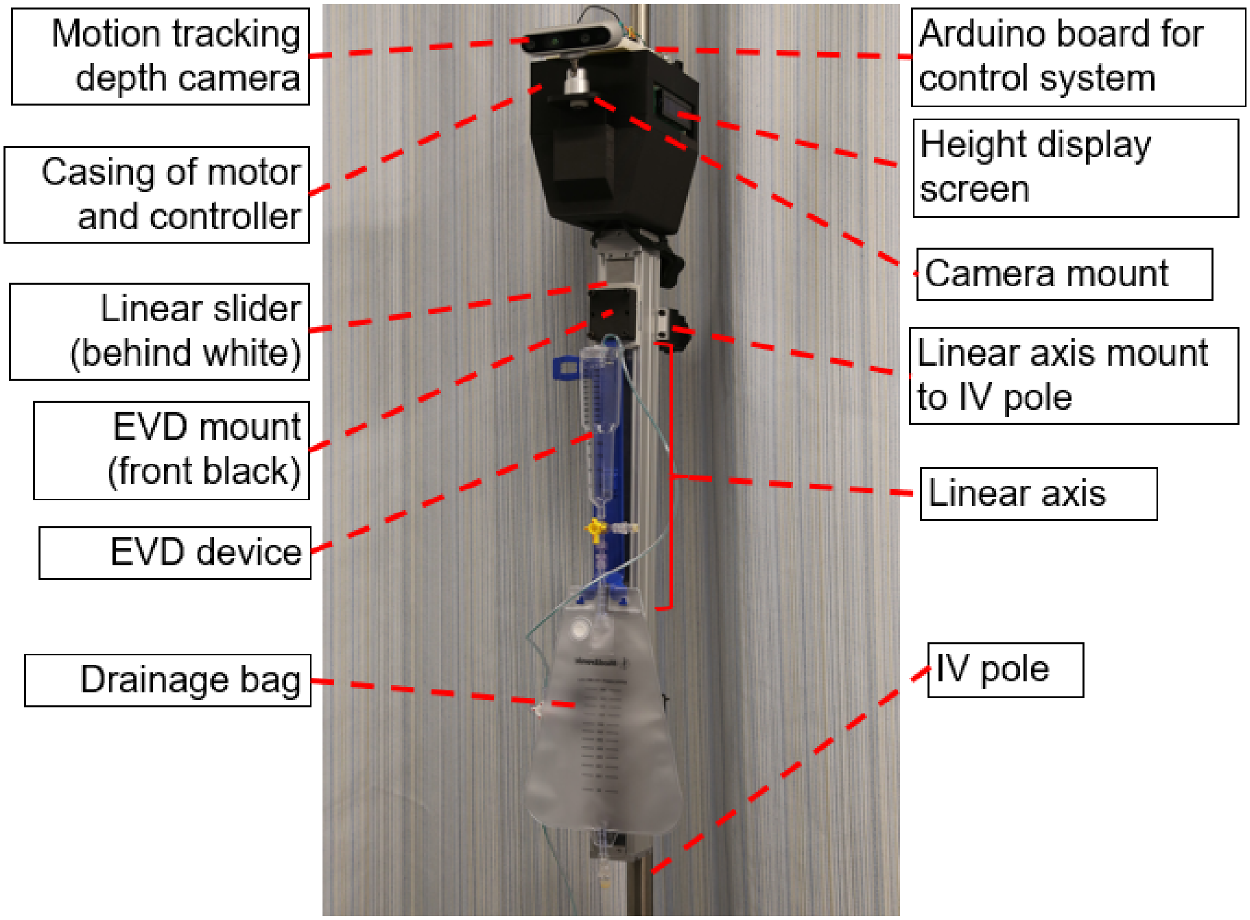

Motion tracking depth camera
Arduino board for control system
Casing of motor and controller
Height display screen
Camera mount
Linear slider (behind white)
Linear axis mount to IV pole
EVD mount (front black)
Linear axis
EVD device
Drainage bag
IV pole

Figure 1: System overview

1.1 Objective 1: Integration of Medical ICU Application Standards in HMI Design

To ensure safety and maintain quality management, the system works following the standards IEC 60601 and ISO 13485. Utilizing ICU-specific protocols, the system presents crucial Intracranial Pressure (ICP) and Cerebrospinal Fluid (CSF) management data. For tailoring Functional Requirements for ICU the system: 1) Enhancing alarm management systems to align with IEC 60601-1-8, focusing on minimizing alarm fatigue. 2) Implementing strong data security measures to ensure patient privacy by HIPAA and GDPR regulations. 3) Building user-friendly functions to enhance accessibility in high-pressure ICU environments.

1.2 Objective 2: Vision-based Head Motion Tracking

The system will employ a machine learning algorithm within a computer vision framework to monitor a specific point of interest on either the right or left side of the patient's head. This algorithm is designed to identify the start and end positions of the patient's head movement. Initially, it determines the vertical movement in terms of pixels and then converts this pixel-based measurement into real-world distances, represented in centimeters.

1.3 Objective 3: Feedback Control of Linear System for Following Movement

Integrate a feedback control algorithm into the linear system. This algorithm will process the distance measurement from head motion tracking and translate it into specific actions for the motor. It guarantees that the EVD is adjusted to the ideal height in response to head movements.

1.4 Objective 4: Experiments and Validations

Our group's motion-tracking algorithm was validated by comparing its tracking results with the output of the VICON motion capture system. This validation involved running multiple trials with the camera in different locations to allow our team to optimize our algorithm. In addition to the validation of the vision-based systems, our team performed IMU and EMG sensor tests, which helped us evolve to the current method.

2 Background

2.1 Components in an EVD

A functional EVD system should have the following components: A catheter, which is a flexible tube inserted into the ventricles of the brain. A tubing and stopcock, which connect the catheter to a graduated drip chamber. The graduated drip chamber, which is a cylindrical glass body with a measuring scale where the CSF will accumulate. The marked notch on the collection chamber allows nurses to calculate drainage rates. This rate varies frequently between different patients, and the physician will advise a reference drainage rate for nurses to monitor. To set the target pressure, the EVD system is equipped with a laser emitter. The laser is manually adjusted to the level of the lateral ventricle of the patient by aligning it with the Tragus of the ear, and the collection chamber can be moved up and down to set the desired pressure. Lastly, the system should also have a drainage bag to collect the cerebral fluid in the collection chamber after the drainage rate is obtained.

2.2 EVD Setup

The setup of EVD devices happens in the operation room, with physicians inserting the drainage catheter into the patient's ventricle (an explanation in Fig 2). The patient is typically sedated, and the skull is sterilized around the point where the drain will be inserted. The most frequently used area is called Kocher's Point. Kocher's Point is located 1 to 2 cm anterior to the coronal suture in the mid pupillary line, or 11 cm posterior from the glabella and 3 to 4 cm lateral from midline ipsilateral medial canthus and a line extending coronally from the ipsilateral tragus [4]. Before the drain is inserted, nurses will prime the drainage tube with saline. "Priming" the system refers to the process of inserting saline into the drainage tube to remove air and bubbles in the system. Once the EVD is ready and the catheter is successfully installed, physicians will connect the catheter to the EVD system. To set up the target pressure in the ventricles, the stopcock is aligned with the patient's ear. Physicians will move the collection chamber along the scale to increase or decrease the drainage rate of the EVD. The stopcock between the collection chamber and the drainage bag will be closed, and the stopcock between the catheter to the collection chamber will be opened to allow fluid to flow through. Nurses in the neuro-ICU will monitor the patient's status after the operation, including checking on the flow rate, substituting the drainage bag if full, and adjusting the pressure settings according to the physician's note.

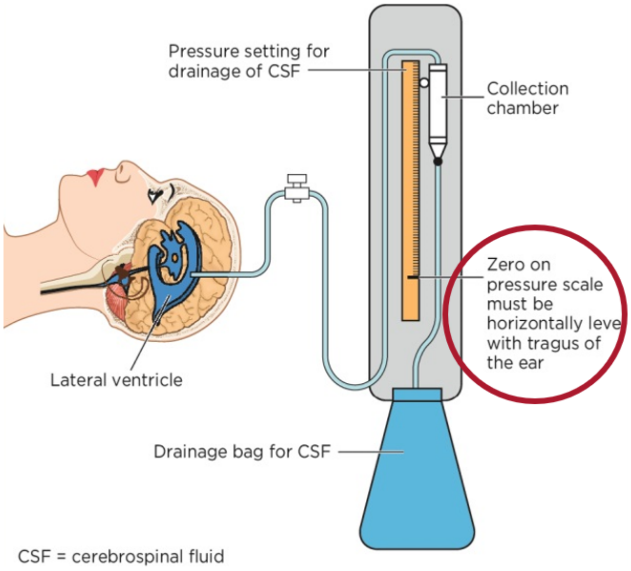

Pressure setting for drainage of CSF
Collection chamber
Zero on pressure scale must be horizontally leve with tragus of the ear
Lateral ventricle
Drainage bag for CSF
CSF = cerebrospinal fluid

Figure 2: A demonstration of EVD setup

2.3 EVD Management

Once an EVD is setup, it needs to be monitored frequently to ensure the CSF is draining at an appropriate rate. There are a number of things to consider during EVD Management. The first of these is infection prevention. Ventriculitis and meningitis are significant concerns with EVDs, so it's important that the devices are sterilized before use and remain clean while connected to the patient. [5]. The use of antibiotic-impregnated catheters has been shown to reduce the incidence of infections. Another major concern with EVD management is proper CSF drainage. Since the drainage rate of these devices is determined by their alignment with the patient, every time the patient moves, they need to be adjusted. This results in nurses constantly needing to adjust the level of the EVD in order to maintain an accurate drainage rate. The amount of CSF drained is carefully regulated based on clinical and radiological findings, aiming to maintain optimal ICP and ensure cerebral perfusion [6]. Overdrainage or underdrainage can lead to complications such as subdural hematomas or persistent elevated ICP, respectively [7]. Frequent neurological assessments are performed to monitor the patient's status and the effectiveness of the drainage [8]. Changes in neurological status may necessitate adjustments in the EVD settings or further intervention [9].

2.4 Previous Work of Automation EVD System

The University of Michigan Weil Institute for Critical Care Research and Innovation developed the Dynamically Enhanced Ventricular Drain (DEVD in Fig 3), which significantly advances the management of TBI by modernizing the CSF and ICP management techniques [10]. Traditional EVD systems rely on manual setup and constant monitoring by healthcare professionals to maintain appropriate CSF drainage and ICP levels. In contrast, the DEVD automates these processes, enhancing the precision and reliability of treatments. The DEVD system is equipped with high-precision pressure transducers and flow sensors that continuously monitor the dynamics of CSF and ICP. These sensors adjust automatically to changes in the patient's physiological state, thus ensuring accurate and responsive ICP regulation. This automation reduces the need for frequent manual adjustments typically required by traditional EVDs, thereby minimizing human error and increasing care efficiency.

While traditional EVDs require nurses or physicians to manually adjust the drainage rate by altering the position of the drip chamber, the DEVD's automated features allow for wireless real-time monitoring without direct human intervention. This not only reduces the workload on clinical staff but also enhances patient safety by providing consistent and timely responses to fluctuating pressure levels. However, the DEVD only performs its functions in initiating and monitoring the ICP in real-time, and medical staff are still spending a lot of time adjusting the drainage chamber by hand when patients change their positions. It doesn't provide an ideal solution for reducing the workload of clinical staff and relieving the discomfort of patients. To this end, we understand the ideal solution should be an automated leveling system to realign the EVD when patients move their heads during monitoring sessions. This motivation leads us to delve into this project.

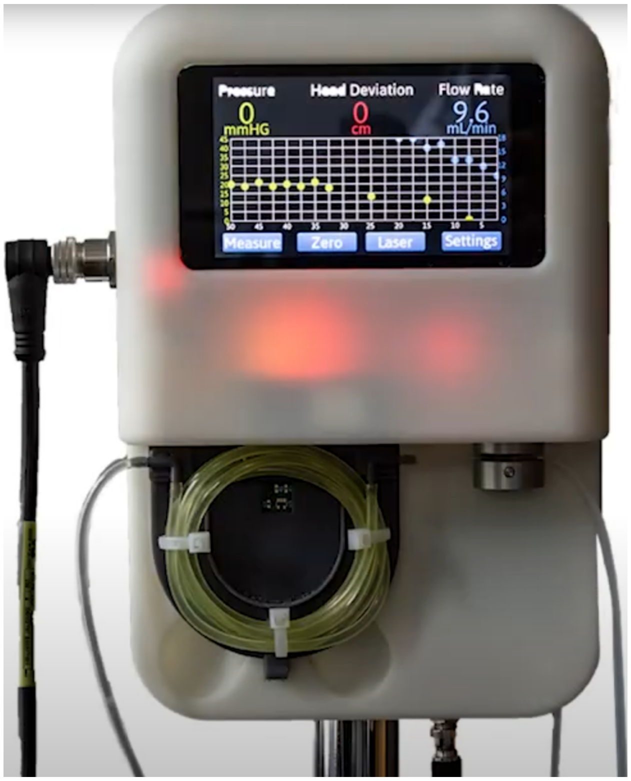


Figure 3: The DEVD from Umich

Another approach to solving this problem is the LiquoGuard system (LiquoGuard 7 CSF Management System, Moeller Medical, Fulda, Germany), seen below in Fig 4. This system removes the need for drip chamber alignment entirely by controlling the drainage rate with a pump. The CSF pressure is measured directly with two pressure sensors, and the CSF drainage rate is set based on these pressure readings. This allows for the system to simultaneously drain CSF and measure pressure. It also has an optional sensor that can be attached to the tip of the catheter to measure the ICP directly.

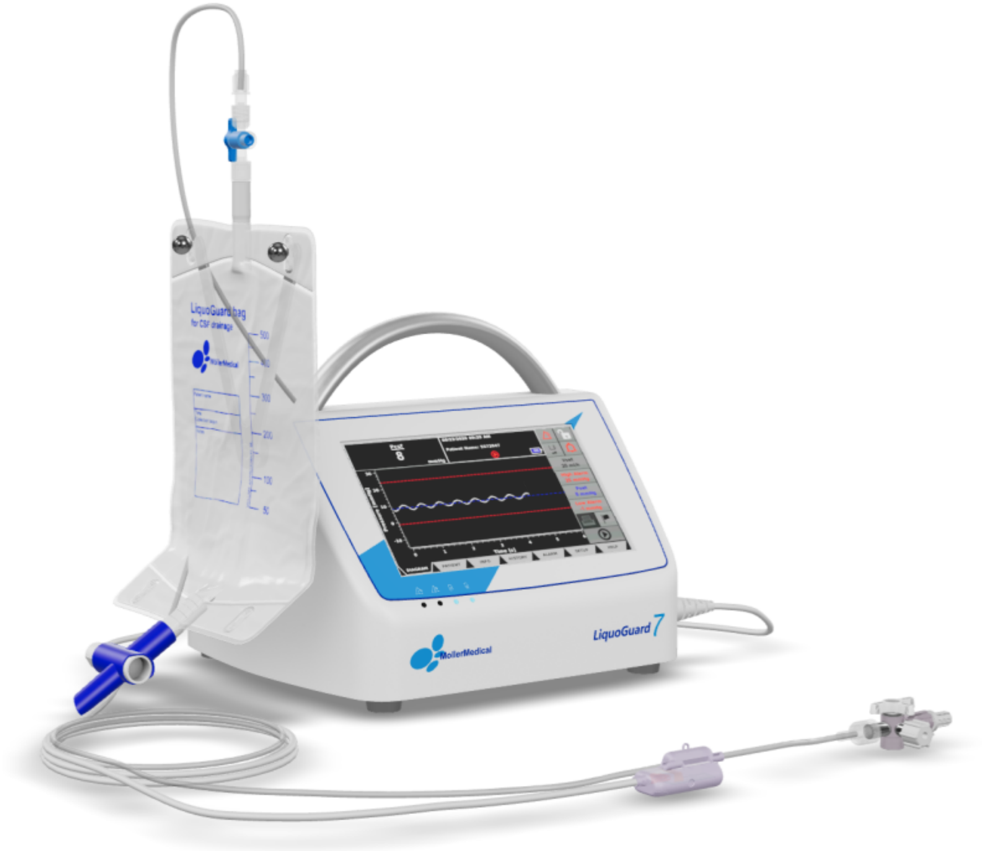


Figure 4: LiquoGuard 7 CSF Management System

While this system does reduce nurse intervention by automating the drainage process, controlling the drainage rate with a pump and sensors is inherently more dangerous than controlling it hydrostatically. If an existing EVDs' drainage rate needs to be changed, simply moving the drip chamber up or down allows the nurse to make that adjustment. If the pump system were to fail in the LiquoGuard, there is no system in place to immediately change the drainage rate. This concern led our team to design a system that operates similarly to existing EVDs and can be manually adjusted when necessary while avoiding the potential risk that a pump-based system would have.

2.5 Nurse Interview

Our team conducted an interview with a nurse with more than 10 years of experience at UMASS Memorial Hospital to get a better understanding of the problems with the current devices and workflow. This interview provided us with useful feedback on the current EVD devices they are using, as well as with the standard EVD monitoring protocol that nurses follow.

During the procedure, nurses are responsible for connecting the EVD to the patient line that was inserted by the neurosurgeon. The surgeon tells the nurses the desired intracranial pressure, drainage flow rate, and how long the drain will be connected to the patient. The nurses follow these instructions to set up the EVD and monitor the drainage process. While the patient is connected to the EVD, the standard protocol is to check on the patients on an hourly basis, unless there is an alarm being triggered. If the alarm is triggered due to a patient's condition, the nurses will readjust the height of the device if it is misaligned, or contact the surgeon for advice.

Some major complaints about the existing EVDs are that the plastic drip chamber and body are easy to break, the tubing is thin and the valve can get blocked by blood clots, the alarm trigger rate is incredibly high if the patient changes posture, and the device needs to travel with the patient frequently but it's difficult to maintain alignment with the patient during travel.

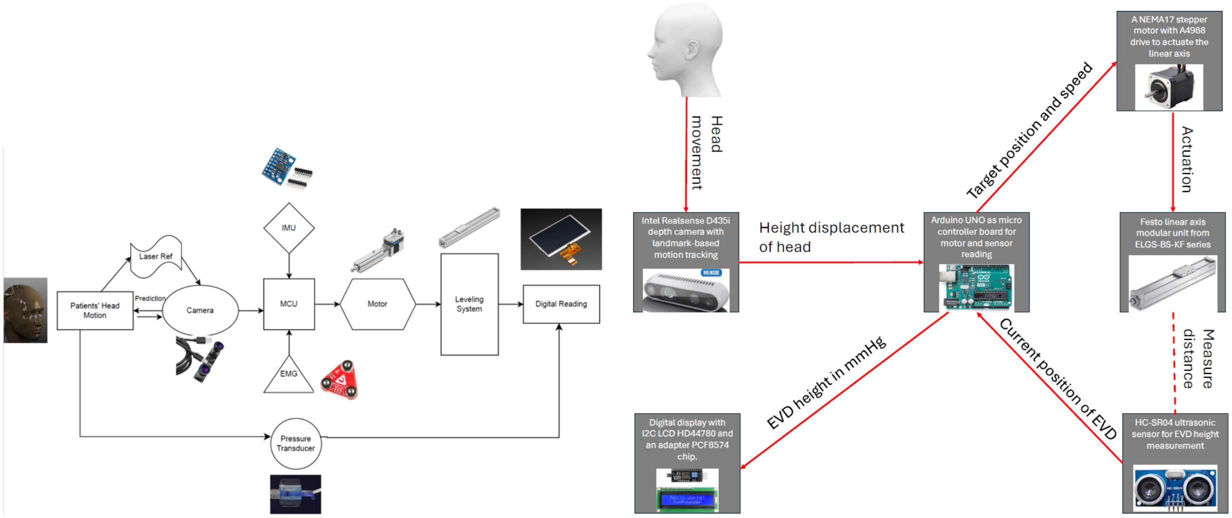
Finally, the team asked the nurses for their opinions on how the EVD equipment could be improved. They recommended a device that removes the easy to crack plastic body, but functions similarly to existing devices. Ideally the new device wouldn't require a lot of training to understand, as it would be time consuming and expensive to train a team of nurses on how to operate a completely new device.

3 Design Process

With the feedback from our interview in mind, our team decided to design a device that automates the movement of an existing device by mounting it on an actuator. This removes the flimsy plastic body the nurse expressed concern with, and functions similarly enough to an existing EVD that it would be easier to adopt than the devices created by the University of Michigan or Moeller. The figure below (Fig 5) outlines the overview of the current system compared with our initial one. Our initial design (Fig 5a), had a camera, laser, pressure transducer, EMG and IMU sensors, control unit, and actuator and proved to be more complex then necessary. Our final design (Fig 5b) is much simpler with just a camera, Arduino, actuator, and ultrasonic sensor. This section will discuss the different components of our design and how our team made the decisions we did.

3.1 Head Motion Tracking System

Developing a head motion tracking system is crucial for this project for several reasons. Firstly, the primary cause of unstable ICP is the patient's head motion. Therefore, understanding the dynamics


Laser Ref
Patients' Head Motion
Prediction
Camera
IMU
MCU
EMG
Motor
Leveling System
Digital Reading
Pressure Transducer
Head movement
Intel Realsense D435i depth camera with landmark-based motion tracking
Height displacement of head
Arduino UNO as micro controller board for motor and sensor reading
Target position and speed
A NEMA17 stepper motor with A4988 drive to actuate the linear axis
Actuation
Festo linear axis modular unit from ELGS-BS-KF series
Measure distance
HC-SR04 ultrasonic sensor for EVD height measurement
Current position of EVD
EVD height in mmHg
Digital display with I2C LCD HD44780 and an adapter PCF8574 chip.

(a) (b)

Figure 5: Initial design diagram (a), and current design diagram (b).

of a patient's head during neuro-ICU monitoring will be helpful in creating an automated EVD regulation device. Secondly, tracking the head's start and end positions allows for the direct measurement of its vertical displacement in the world frame. Initially, we employed a multi-modal sensor system combining electromyography (EMG) and inertial measurement units (IMU) for tracking, which yielded unsatisfactory results. Subsequently, we switched to using camera-based tracking, which produced more favorable outcomes.

3.1.1 Multi-modal Head Motion Tracking

The primary objective of the Multimodal Head Motion Tracking System is to employ a synergistic approach using two distinct sensors: an Electromyogram (EMG) and an Inertial Measurement Unit (IMU). This integration aims to achieve precise recording and prediction of head movements and ascertain the head's current position with high accuracy. The design is to use EMG and IMU integrated sensors to achieve real-time monitoring, when EMG receives muscle contraction signals, the program will ask for IMU input to calculate the integral of linear acceleration to obtain the displacement of patient's head.

We investigated the functionalities of EMG (Electromyograph) sensors and the possibility of integrating an EMG sensor into the automated EVD system. EMG sensors are devices used to record and measure the electrical activity produced by skeletal muscles. When the muscle contracts, the EMG sensor will capture the voltage potential. The design concept the team pursued is to situate one EMG sensor module on the back portion of the patient's neck. Given that head movements predominantly engage the neck muscles, the EMG sensor is ideally situated to capture muscular activity signals corresponding to these movements. Further investigations on the plausibility of integrating EMG sensors were carried on by the team, and the final conclusion on EMG sensors will be introduced in later sections.

In conjunction, the IMU sensor is placed at a level equivalent to the tragus, a region adjacent to the ear. This allows the IMU sensor to effectively track the brain ventricles' movements in conjunction with the patient's head motions. The sensor outputs six key parameters: three axes of acceleration (X, Y, and Z) and three axes of rotational data (gyroscope on X, Y, and Z). The gyroscope data is determining the direction of the movement, while the triaxial acceleration data are to computing the displacement in each respective direction. This displacement data is crucial for guiding the adjustment of the External Ventricular Drain (EVD) height through a linear actuator, ensuring precise and responsive alteration of linear actuator.

To effectively integrate these sensor outputs, our algorithmic framework utilizes the EMG sensor data as an initial trigger. Upon detecting a change in voltage potential indicative of muscular activity, the algorithm promptly requests the latest IMU data. The program will draw the acceleration information from the IMU input, and integrate the acceleration for Z-axis to acquire verticle displacement caused by patient movement. Additionally, the IMU sensor's acceleration data is continuously monitored. The real-time monitoring ensures that the system remains accurately responsive to any movements not detected by the EMG sensor, such as external adjustments to the patient's bed or other bodily movements that may result in a change in head elevation without engaging the neck muscles.

By the integration of EMG and IMU data, our system aims to deliver a robust and reliable solution for head motion tracking, essential for precise medical interventions and patient care.

EMG Sensor Validation

In this experiment, our team investigated the quality of EMG feedback signal using widely used, commercial EMG sensors, specifically, the Delsys Trigno Avanti Wireless EMG Sensor System with EMGworks Acquisition and Analysis Software. The experiment was conducted at a Simulated Intensive Care Unit (ICU) Environment, 50 Prescott St. Also, a iPhone camera is used to record the movement of patients, so that when the research team start analyzing and studying collected data, they would have a video reference on when and how did the patient moves in time domain.

The experiment procedure is presented as follows: 1. Initialize and configure the EMGworks Acquisition and Analysis Software on a dedicated computer system. 2. Establish a connection with one module of the Delsys Trigno Avanti Sensor using the wireless interface. 3. Carefully attach the Trigno Avanti Sensor to the participant's posterior cervical region, targeting the splenius muscle group for EMG signal acquisition. 4. The experiment will be conducted under the supervision of a qualified instructor. The participant was guided through a series of predetermined movements: lateral flexion of the head to the left and right, extension and flexion (looking up and down), axial rotation of the torso to the left and right, and positional changes from an upright posture to supine (laying flat). 5. Throughout this process, the participant's movements will be simultaneously recorded using high-resolution video equipment to visually correlate with the EMG data. 6. Post-acquisition, EMG data will be synchronized and analyzed in conjunction with the video footage to establish a comprehensive understanding of muscle activation patterns corresponding to each specific movement.

EMG Sensor Validation Result

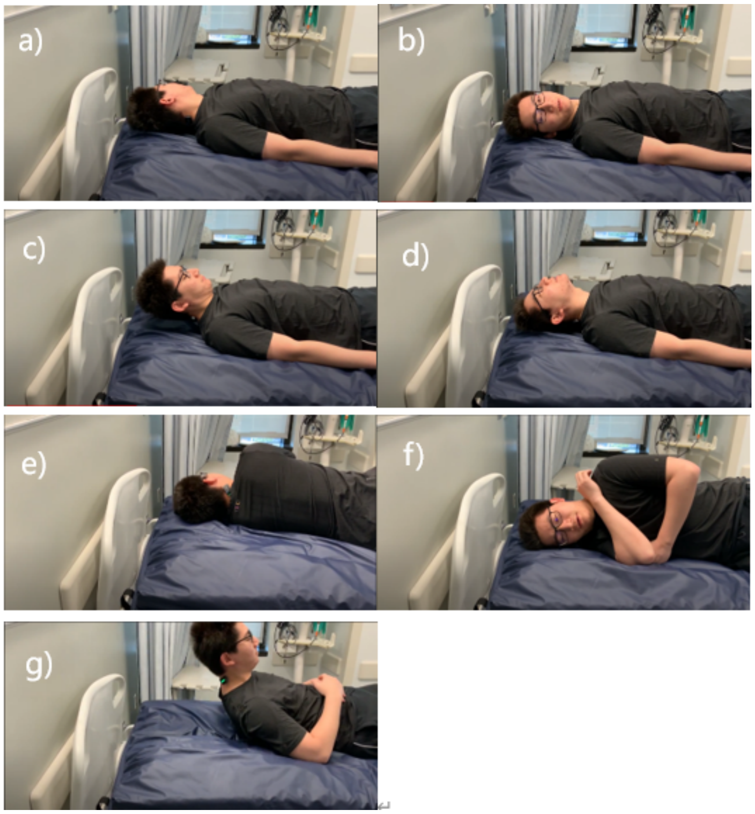


Figure 6: Patient Posture that is Tested in Experiment

a) Turn head to left b) Turn head to right c) Turn head down d) Turn head up e) Turn body to left f) Turn body to right g) Get up. The sequence of testing between postures follows alphabetical order, from posture a) to g).

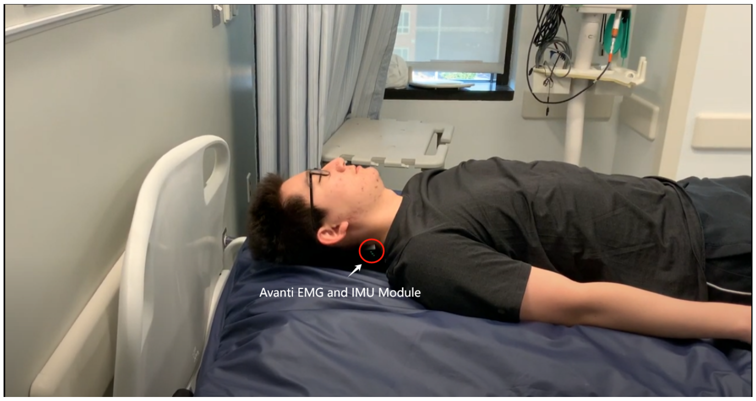


Figure 7: Experimental Set Up and EMG Senosr Placement

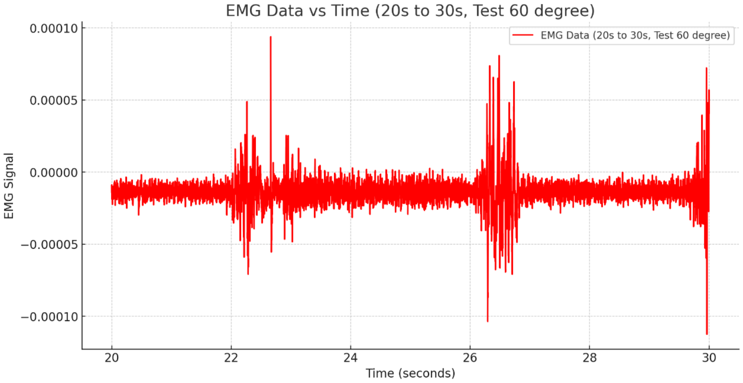


Figure 8: An Example of Solid EMG Data

This segment of captured EMG data demonstrates the desired EMG data, with clear and steady baseline, and obvious voltage change in magnitude when the patient moves.

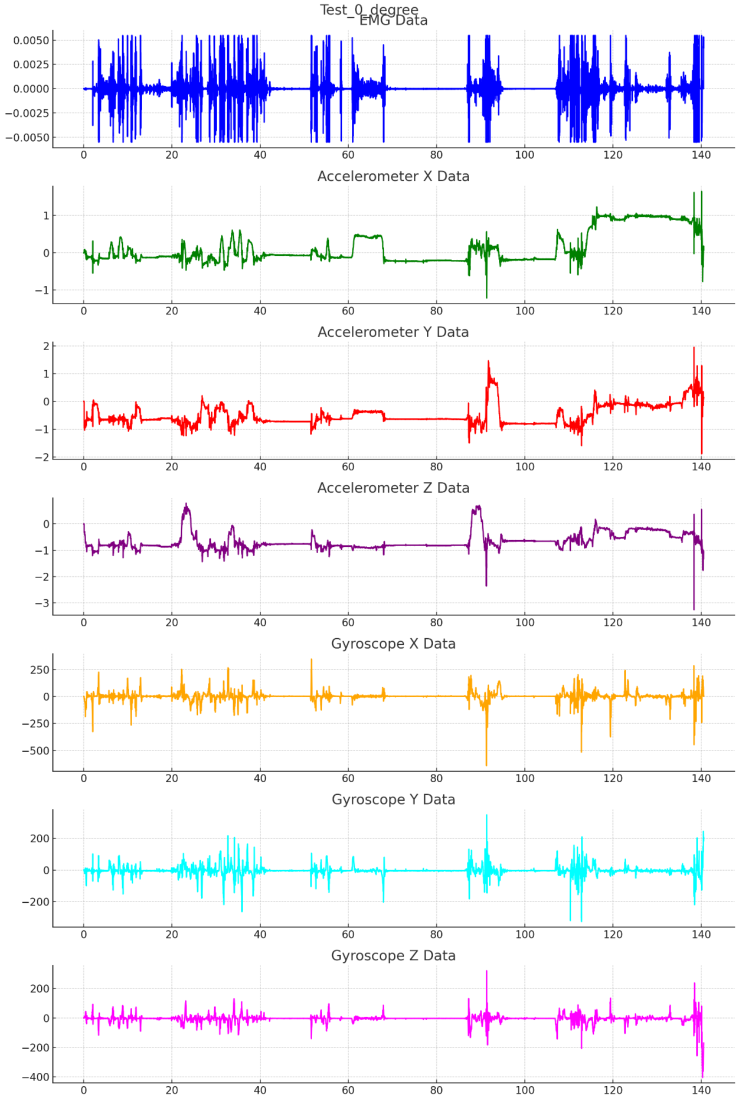


Figure 9: EMG Sensor Results Match Up with IMU FeedBack

The captured EMG data is able to align with the IMU acceleration sensor, with each acceleration corresponding to muscle movement. However, EMG data fails to convey other critical information such as movement direction and magnitude.

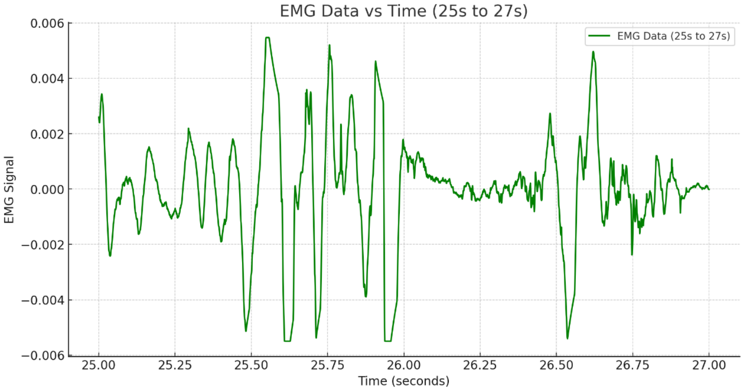


Figure 10: EMG Motion Artifact

The collected EMG data above within 2 second interval has a frequency around 1 Hz.But a typical EMG data has a frequency around 20-500 Hz. Therefore this segment is recognized as human artifact due to the EMG sensor sticker lost contact with the skin. The situation is prominent in our data capture.

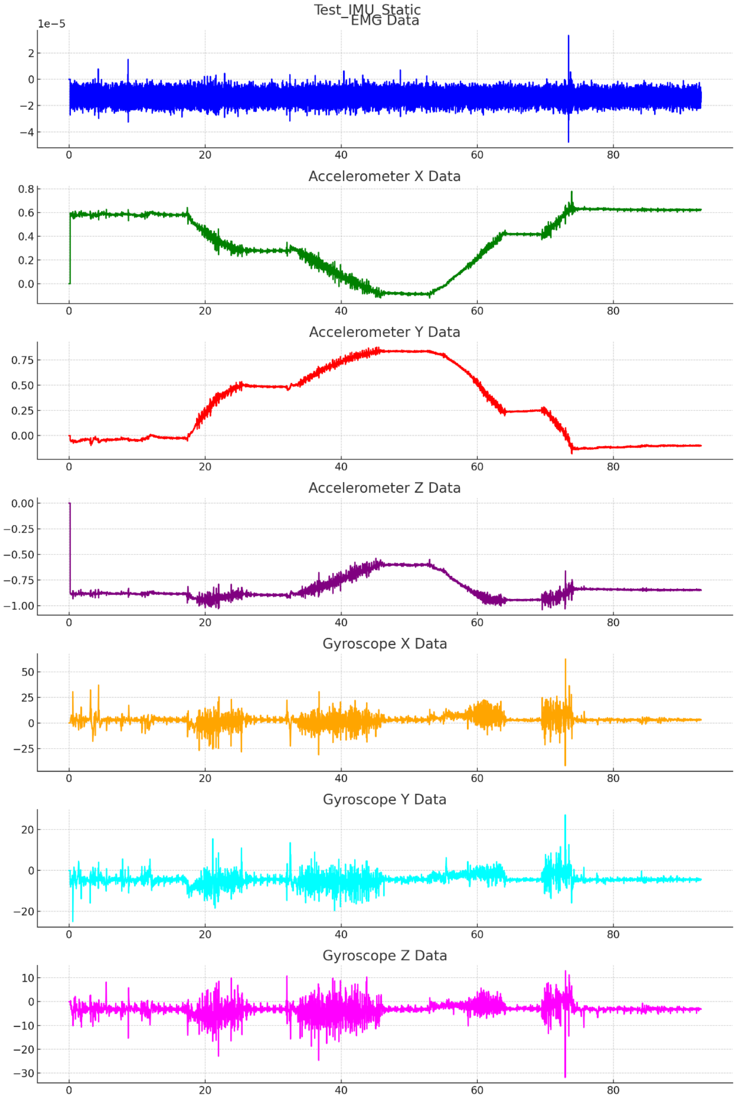


Figure 11: When Lifting the ICU Bed Electronically without Patient Movement, EMG Sensor Fails to Convey Potential Height Change

As part of our test, the team lifted the ”asleeped” patient with the bed slowly turning up, simulating situations where the nurses or patient family adjust the bed height but without patient movement. Since the patient does not move, the height change cannot be reflected on the EMG data.

Discussion On Implementing EMG Sensor

Our investigation into the EMG sensor reveals it’s capability in capturing neck muscle contraction when the patient movement involves the target muscle (Figure 5) and the contact between sensor and skin is solid. Nonetheless, it notably struggled in registering the tilt motion of the upper body as the target muscle does not involve in the movement, as shown on figure 7. Moreover, the sensor tends to lose contact with the skin from time to time during the experiment and introduced many noise in our capture (Figure 6). The sensor's limited capability in detecting upper body movement is attributed to the fact that the target muscle does not participate in such posture. Furthermore, since the team used only one EMG sensor on one side of the neck, it can only capture head-turning motion in one direction. That is, if the sensor is placed on the right, it only detects the right turn, and vice versa. Lastly, the sensor attachment is another challenge for sensor application; the sensor is situated in a region that involves a lot of twists and rotations. The adhesive on the electrodes may face failure for long time application. During the experiment, the sensor started to loose contact after 3 trials. The long-term usage of sensors is not positive for our research.Furthermore, in our future integration with the EMG and IMU unit, we will not be able to use the Delsys wireless system. Instead, the team will use the MyoWare 2.0 wired EMG sensor chip. The EMG sensor application requires an additional wire connection, resulting in a bulkier device. Our study's limitations include a confined testing environment, with the simulated ICU room is not a full representation in Hospital, there might be more complex scenarios that the experiment fails to include. Also, the sample size is limited to only 18 trials on 1 person, which might have affected the comprehensiveness of our results. Considering the pros and cons to EMG sensor, the team decided to remove the EMG sensor from the subsystem list, due to the complexity in system communication and bulkiness overthrown the value of inputs brought by EMG sensor.

3.1.1.1 IMU Sensor Validation

Similarly as we test out the EMG sensor, our team will validate the plausibility in using IMU unit for head motion prediction. In this validation, the team still used the Delsys Trigno Avanti Wireless EMG and IMU Sensor System with EMGworks Acquisition and Analysis Software. The experiment was conducted at a Simulated Intensive Care Unit (ICU) Environment, 50 Prescott St. Also, a iPhone camera is also used to record the movement of patients, so that when the research team start analyzing and studying collected data, they would have a video reference on when and how did the patient moves in time domain.

The experiment procedure is presented as follows: 1. Initialize and configure the EMGworks Acquisition and Analysis Software on a dedicated computer system. 2. Establish a connection with one module of the Delsys Trigno Avanti Sensor using the wireless interface. 3. Carefully attach the Trigno Avanti Sensor to the participant's posterior cervical region, targeting the splenius muscle group for IMU signal acquisition. 4. The experiment will be conducted under the supervision of a qualified instructor. The participant was guided through a series of predetermined movements: lateral flexion of the head to the left and right, extension and flexion (looking up and down), axial rotation of the torso to the left and right, and positional changes from an upright posture to supine (laying flat). 5. Throughout this process, the participant's movements will be simultaneously recorded using high-resolution video equipment to visually correlate with the IMU data. 6. Post-acquisition, IMU data will be synchronized and analyzed in conjunction with the video footage to establish a comprehensive understanding of muscle activation patterns corresponding to each specific movement.

3.1.1.2 IMU Sensor Testing Result

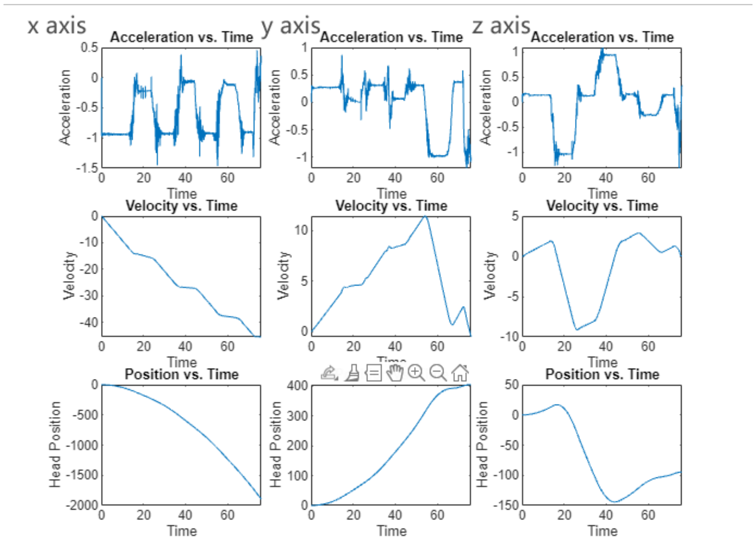


Figure 12: IMU Acceleration Data and Integral Calculation, for turning body movement at 0-degree ICU bed

Raw IMU acceleration data reflected when and how the body turned, by analyzing the acceleration data, we could obtain the direction in which the head moves. The data is extracted from original Delsys software and taken integral in MATLAB. From the graph, even when the patient is not moving, the sensor feedbacks non-zero acceleration. This results in an inaccuracy in double integral of the position acquisition. This could be solved by implementing an zeroing equation to the collected data, but it is challenging to remain zero in real-time monitoring and would make integration a problem.

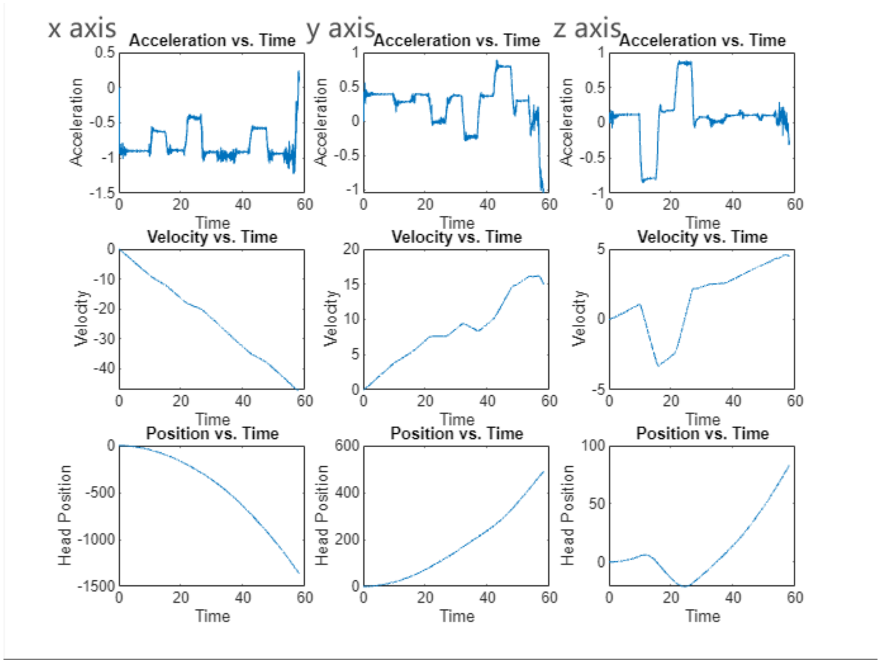


Figure 13: IMU Acceleration Data and Integral for Turning Head Only Movement.

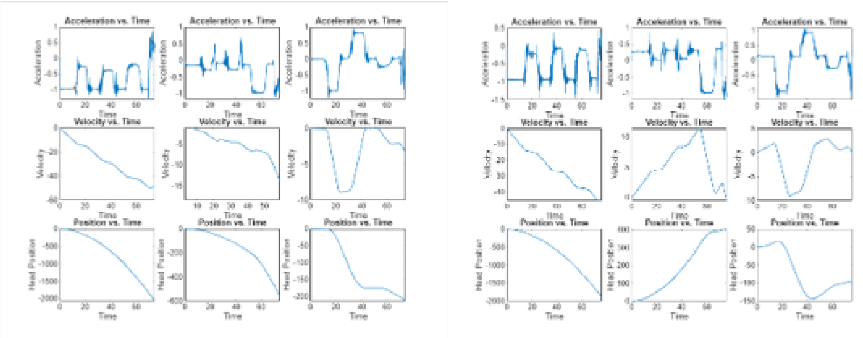


Figure 14: Comparison Between IMU Output With Same Motion

Both left and right graph is the collected IMU data and their integral calculation on the same set of body motion. However, the motion on left graph was performed at a 0 degree bed angle, while the right graph was collected on a 30 degree bed angle. Both IMU accelerations yielded similar patterns, however, the IMU data failed to convey the difference in height due to bed angle. In real-life application, the system cannot distinguish distinguish the pre-set bed angle unless manually introduced it into the system.

Discussion on Implementing IMU Sensor

In our experiment, we explored the functionality of an IMU sensor. According to our results, the IMU sensor provides ample data on the object's acceleration on x, y and z axis. The IMU acceleration sensor is responsive and accurate in capturing sudden movement and possesses high resistance to artificial noises.

However, it does have major drawbacks that force us to remove the sensor from our part. First of all, our implementation requires taking the integral of IMU acceleration data to obtain the actual displacement of the patient's head in the vertical direction. Nevertheless, even using the Delsys system, a highly precise industrial IMU sensor, we encounter minor IMU sensor drift, and the error caused by this drift is then amplified when performing double integral in MATLAB. Secondly, in our multi-model setup, the IMU sensor serves as the major input. Thus, it is vital to ensure the IMU sensor would give reliable data at all times. Nevertheless, the IMU sensor can not distinguish between different initial bed angle setups, as shown in figure 11. Subsequently, our team discovered the other utility such as the depth camera, which can perform a better job than integrating EMG+IMU units. The team decided to explore a vision-based head-tracking approach for easier implementation and a more reliable system.

3.1.2 Vision-based Head Motion Tracking

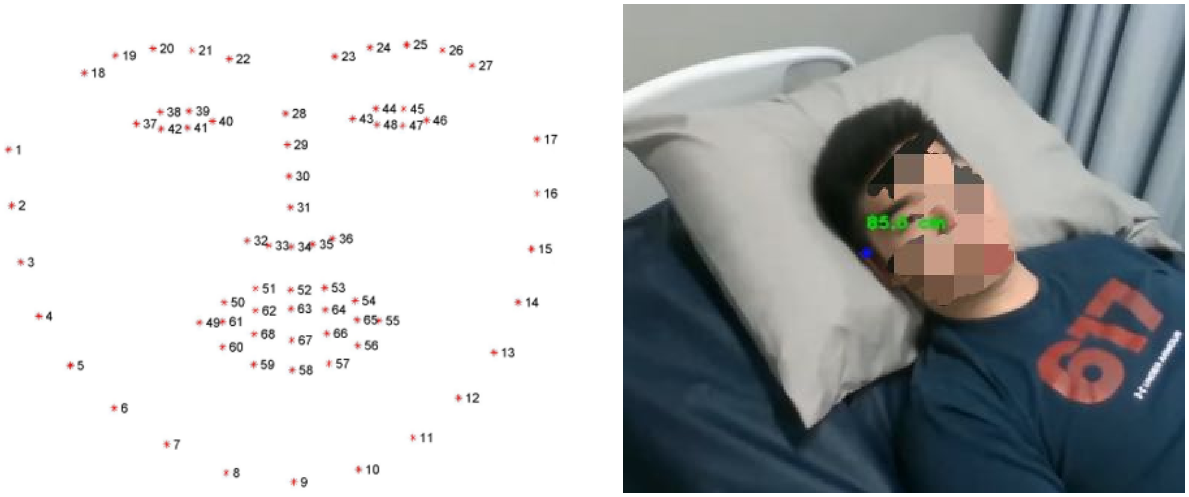
This system aims to track the head movement by detecting facial landmarks and allows sensing of initial and end positions autonomously. The system will use image input from the camera and use the '**OpenCV**' library to process. Algorithm 1 introduces a robust approach to tracking head movement using facial landmarks with **dlib** (a modern C++ toolkit containing machine learning algorithms) [11] and '**shape predictor 68 face landmarks model**' (in Fig 15 (a)) [12]. It includes a fallback mechanism to continue tracking even when specific landmarks (such as the ear) are not visible due to the head turning away from the camera. The main loop '**while True**': loop continuously reads frames from the camera. For each frame, the algorithm will convert the input image to grayscale (for more efficient processing) and use the '**detector**' function to find faces. Once faces are detected, the '**OpenCV**' window is activated, and the facial landmark '**predictor**' function can find facial structures on the face (the details of these two functions will be introduced in the next paragraph). Within the loop, for each detected face: 1) The script attempts to detect facial landmarks. Specifically, it tries to locate an ear by selecting a landmark (e.g., landmark 0 for the right ear). 2) If successful, it visualizes this by drawing a circle on the ear's location (Fig 15 (b)). 3) If the specific landmark is not visible (due to the head's orientation), it catches the failure and instead marks the general area of the face using the face's bounding box. After setting the initial and end positions, it calculates the vertical movements by subtracting the initial coordinates from the end coordinates. It releases the camera and closes any '**OpenCV**' windows. Then, it prints and returns the calculated displacement.

The '**detector**' function uses a machine learning model based on HOG (Histogram of Oriented Gradients) and combines it with a linear classifier, an image pyramid, and a sliding window for face detection [13]. The first step is to convert the face image into an HOG representation. The HOG descriptor algorithm

Algorithm 1 Vision-based Head Motion Tracking Algorithm

```
Function TrackHeadMovement(model_path)
  Initialize face_detector, landmark_predictor with model_path
  Open video_capture
  Define CalculateDistance($p_1, p_2$)
  While True do
    Capture frame $\rightarrow$ detector
    If $p_1$ not move
      initial_position $\leftarrow$ current_position
    ElseIf $p_2$ not move
      end_position $\leftarrow$ current_position
      break
  Release video_capture
  Destroy all_windows
  If initial_position $\neq$ NULL and end_position $\neq$ NULL then
    distance_moved $\leftarrow$ CalculateDistance(initial_position, end_position)
    return distance_moved
  Else
    return NULL
```

counts occurrences of gradient orientation in localized portions of an image [14]. The second step is the Sliding Window [15] operation: After generating the HOG feature vectors for the image, a sliding window is moved across the image (left to right, top to bottom) to detect objects [16]. At each position of the window, a classifier is used to determine whether a face is present in that section of the image. The third step is the Image Pyramid operation: it consists of images at different scales (scaled of the original image) [17]. By using this method, the Sliding Window can detect faces of varying sizes [18]. The last step is to apply a Machine Learning Classifier: Dlib detector uses a linear SVM (Support Vector Machine) as the classifier [11]. It has been trained on a labeled dataset of faces to distinguish between face and non-face images. In the next, '**predictor**' function can find facial structures on the face. Dlib has a pre-trained facial landmark detector, which is used to estimate the location of $68(x, y)$ coordinates that map to facial structures on the face [19]. At first, it uses the CLM (Constrained Local Model) [20] and an Ensemble of Regression Trees [21]: Dlib implements a variant of the CLM algorithm, which uses an ensemble of regression trees to predict the shape or pose of an object. The second step is initialization. The algorithm starts with an initial guess of where the facial landmarks are located in the image. The third step is Feature Extraction: Features are extracted from regions around the initial guess of the landmarks. To make an accurate estimation, in the fourth step, we have Regression Trees. It extracted features and fed them into an ensemble of regression trees that predict how to adjust the positions of the landmarks toward the actual facial features in the image. To optimize the output, the fifth step is an Iterative Refinement. This process was iterated several times to refine the landmark positions to increasingly accurate estimates. The last step is output. The model outputs '68 Landmarks', 68 specific points that correspond to facial features such as the jawline, mouth, eyes, nose,



(a)

(b)

Figure 15: 68 facial landmark model (a), and real human face detectioni (b).

and eyebrows. In general, this method is a lightweight model, which is the fastest way of real-time face detection and motion sensing that can be used on CPU-based devices [22].

After pixel-wised measurement, to faithfully reflect the real vertical displacement of the patient's head, using the 2D webcam and approximation calculation is not an optimal solution. We then switch to a depth camera, Intel RealSense D435i, to capture the depth information to obtain accurate feedback. We implemented depth measurement from the target head tracking point to the camera. Our camera utilizes stereo vision to perceive depth. The camera contains two lenses (similar to human eyes) that capture the same scene from slightly different angles. By comparing these two images, the camera's program package (**pyrealsense2**) can calculate the depth of each point in the scene based on the disparity (the difference in the position of a feature seen in the two images).

To this end, we can calculate the vertical displacement as follows. Given by $dFOV$ (Diagonal Field of View) in degrees, AR (Aspect Ratio) is $Width/Height$, then we can have $VFOV$ (Vertical Field of View).

$$VFOV = 2 \times \arctan\left(\frac{\tan\left(\frac{dFOV}{2}\right)}{\sqrt{AR^2+1}}\right) \tag{1}$$

After we have $VFOV$ and given by Distance to Object: the distance from the camera to the object in meters or centimeters, Pixel Movement: the number of pixels the object has moved vertically on the screen, and Resolution Height: the vertical resolution of the camera in pixels. Then we can have D: the real vertical displacement calculated by

$$D = 2 \times \left(\tan\left(\frac{VFOV}{2}\right) \times \text{Distance to Object}\right) \times \frac{\text{Pixel Movement}}{\text{Resolution Height}} \tag{2}$$

To validate our head motion tracking and displacement calculation, we conducted comprehensive experiments in the validation section.

3.2 Feedback Control of Linear Systems

Our group used a linear actuator to level the height of the EVD. We chose the Festo ELGS-BS-KF-45-500-10P-ST-M-H1-PLK-AA linear axis unit (Fig 16). This actuator is a ball screw drive actuator with a stroke length of 500mm. Ball screw drive actuators are better for highly accurate positioning than belt-driven actuators [23], and the EVDs our group had in the lab had a maximum moving range of 420mm. This actuator is also capable of supporting a vertical payload of 5 kg with 4.7 Nm torque and our EVD filled with water weighs (3 kg +- 0.2 kg). It also includes its own control system called IO-Link that our team believed would make controlling the actuator easier.

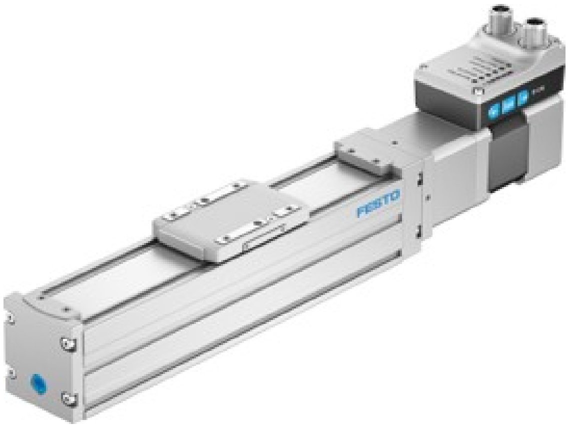
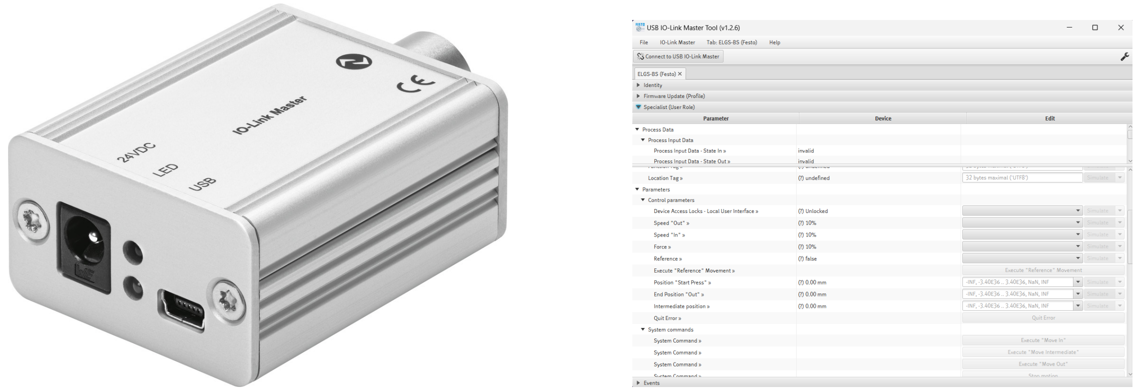


Figure 16: The Festo linear axis unit

3.2.1 Controlling the Linear Axis with the IO-Link Master Module

In order to control the actuator our group got, we also got an IO-Link standard communication module: Festo USB IO-Link Master (Fig. 17 a). This equipment can be controlled with the software: USB IO-Link Master Tool v1.2.6 (Fig. 17 b) and connects with a mini USB port from the IO-Link Master to a USB A port on our computer. However, when we integrated the software into our system to program the control loop we found that without a PLC (programmable logic controller) we could not control it directly through command or program. Rather than try to add a PLC to the system, we addressed this issue by replacing the original integrated motor with an open-source one, the NEMA 17-type motor. In the figure below (Fig 18), we show the motor the actuator came with and the one we replaced it with.



(a)

USB IO-Link Master Tool (v1.2.6)

File IO-Link Master Tab: ELGS-BS (Festo) Help

Connect to USB IO-Link Master

ELGS-BS (Festo)

▶ Identity

▶ Firmware Update (Profile)

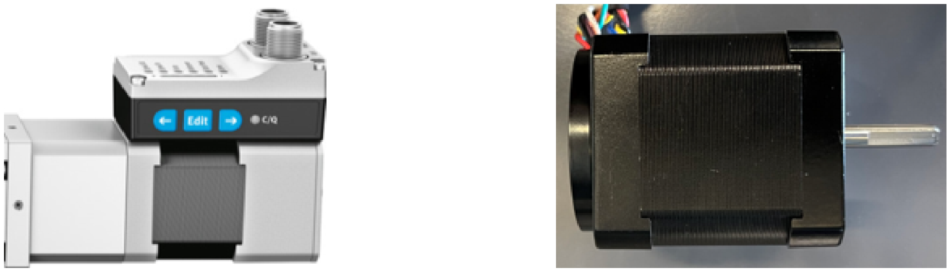
▼ Specialist (User Role)

Parameter	Device	Edit
▼ Process Data		
▼ Process Input Data		
Process Input Data - State In »	invalid	
Process Input Data - State Out »	invalid	
Location Tag »	(?) undefined	32 bytes maximal ('UTF8') Simulate
▼ Parameters		
▼ Control parameters		
Device Access Locks - Local User Interface »	(?) Unlocked	Simulate
Speed "Out" »	(?) 10%	Simulate
Speed "In" »	(?) 10%	Simulate
Force »	(?) 10%	Simulate
Reference »	(?) false	Simulate
Execute "Reference" Movement »		Execute "Reference" Movement
Position "Start Press" »	(?) 0.00 mm	-INF, -3.40E36 .. 3.40E36, NaN, INF Simulate
End Position "Out" »	(?) 0.00 mm	-INF, -3.40E36 .. 3.40E36, NaN, INF Simulate
Intermediate position »	(?) 0.00 mm	-INF, -3.40E36 .. 3.40E36, NaN, INF Simulate
Quit Error »		Quit Error
▼ System commands		
System Command »		Execute "Move In"
System Command »		Execute "Move Intermediate"
System Command »		Execute "Move Out"
System Command »		Stop motion

▶ Events

(b)

Figure 17: USB IO-Link Master (a), and USB IO-Link Master Tool v1.2.6 (b).



Festo integrated motor

NEMA 17-type motor

Figure 18: The included motor (left) and the replacement motor (right)

3.2.2 Feedback Control Loop for Linear Actuator

In this section, we will introduce the control logic used in our system to achieve the tracking movement with the patient's head. Algorithm 2 describes the functions and parameters we need to achieve this control. First, function setup: It starts by defining a function for the control loop that takes '**input.displacement**' as an argument, which is the desired movement input from an external source (in this case, the patient's head movement sensed by the camera). Second, parameter definitions: It sets up the necessary parameters, '**max.position**' and '**min.position**' and defines the boundaries for the actuator to operate. The '**actuator.position**' is the current position, and '**target.displacement**' is the desired movement amount calculated from the '**input.displacement**'. The '**desired.position**' is then calculated based on this '**target.displacement**'. Third, control loop: The main part of the algorithm is a '**While True**' loop, which means it will run continuously until it reaches the target. It reads the current displacement due to head movement (**current.displacement**) and calculates the '**desired.position**' based on this value. It then calculates the direction and magnitude of the required movement and moves the actuator toward the '**desired.position**' within predefined speed limits and updates the '**actuator.position**' after movement.

Algorithm 2 Control Algorithm for Linear Actuator

```
Function Control Loop$Input\_displacement$
  Statement of Parameters
  $max\_position \leftarrow$ Maximum extension of the actuator
  $min\_position \leftarrow$ Minimum retraction of the actuator
  $actuator\_position \leftarrow$ Read current actuator position
  $target\_displacement \leftarrow$ Get target displacement from head movement
  $desired\_position \leftarrow$ Map $target\_displacement$ to actuator range
    While True do
      $current\_displacement \leftarrow$ Read current displacement from head movement
      $desired\_position \leftarrow$ Calculate desired position from $current\_displacement$
        If $desired\_position \neq actuator\_position$
          Determine direction and magnitude of actuator movement
          Move actuator towards $desired\_position$ within speed limits
          $actuator\_position \leftarrow$ Update position after movement
        EndIf
        Safety Check Check for safety limits and adjust $actuator\_position$ if needed
    break if end condition is met
  return $actuator\_position$
```

Fourth, safety check: After each movement, there's a safety check that ensures the actuator hasn't moved beyond its safe operational limits. If it has, it adjusts the '**actuator.position**' to bring it back within limits. Fifth, return value: Once the loop exits, the function returns the final '**actuator.position**'.

3.3 Electrical Components

In this section, we introduced the electrical diagram of our system. An Arduino UNO board is used to read the input from the ultrasonic sensor, send the control command to the NEMA 17 motor, and communicate with the LCD screen to show the height of the EVD device.

3.3.1 Component Overview

First, the center controller is the Arduino UNO board (in Fig 19 ARD 1). We control the NEMA 17 standard-type motor (in Fig 19 right M) through the A4988 stepper motor driver (in Fig 19 right U1). More specifically, we connect the direction and step pin to Digital 4 and 5 to control the stepper motor's direction and step based on the input voltage level. We also need an outer power source ranging from 8 to 24 V to power the stepper motor by the VMOT pin on the A4988 module. To measure the height of the EVD device on the linear axis, we use the HCSR04 ultrasonic sensor (in Fig 19 top U2). We also connect an 8-bit IO expander (in Fig 19 left U4) with the LCD screen (in Fig 19 left COMP1) to enhance the communication from bit-wise to I2C (SCL and SDA pin) with the Arduino UNO board to display the measured height. Two pins are not in the schematic; the integrated encoders are in the NEMA17 stepper motor and are connected

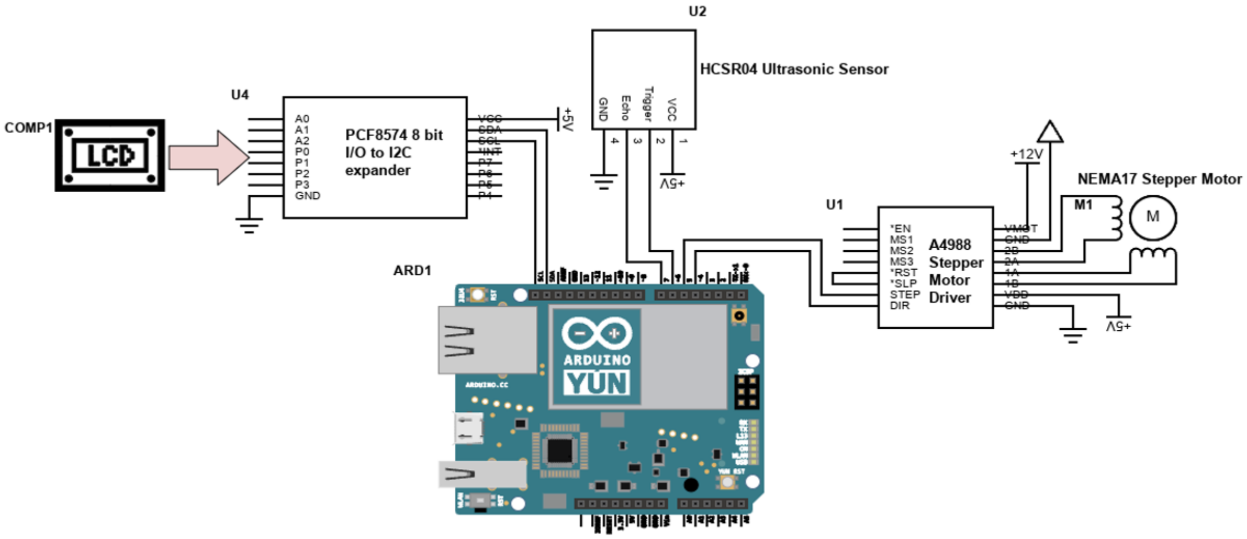


Figure 19: The electrical components diagram

to the Digital IO 2 and 3 since only these two pins in the Arduino UNO are interrupt pins.

3.4 Mechanical Design

Once the actuator, camera, and sensors were selected, a number of components needed to be designed to connect the system. The first of these was a way for the linear actuator system to connect to an IV pole, the second was a way to connect the EVD to the linear actuator, and the third was a casing attached to the actuator that would house all the electrical components. The group decided to 3d print these parts on a Markforged printer using Onyx filament, as 3d printing allowed for faster design iteration than machining, and the Onyx material was strong enough to create a robust prototype.

3.4.1 Clamp Design

Our group designed a clamp to attach the actuator system to the IV pole. The original design seen below in Figure 20 had a V-shaped piece that would push the IV pole into the flat side of the clamp.

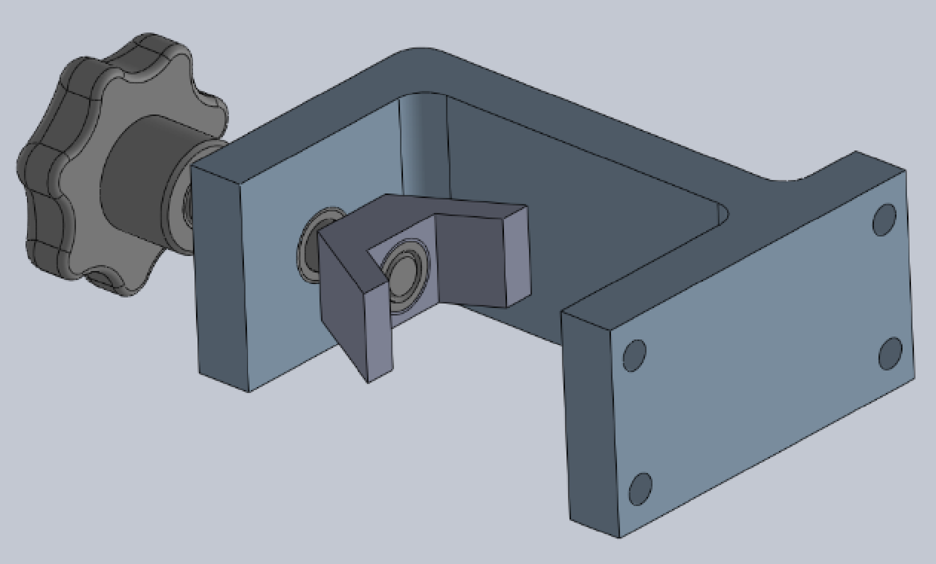


Figure 20: Original Clamp Design

The main advantage of the original shape over a form fitting U-shape was that it could support IV poles of varying diameters. However, this design had a few problems with it. First, the V-shaped piece would need to spin freely in order to be tightened, and the lower surface area of this shape would not grip the pole as well as a U-shape. With these concerns in mind the second iteration was created, seen below in Fig 21.

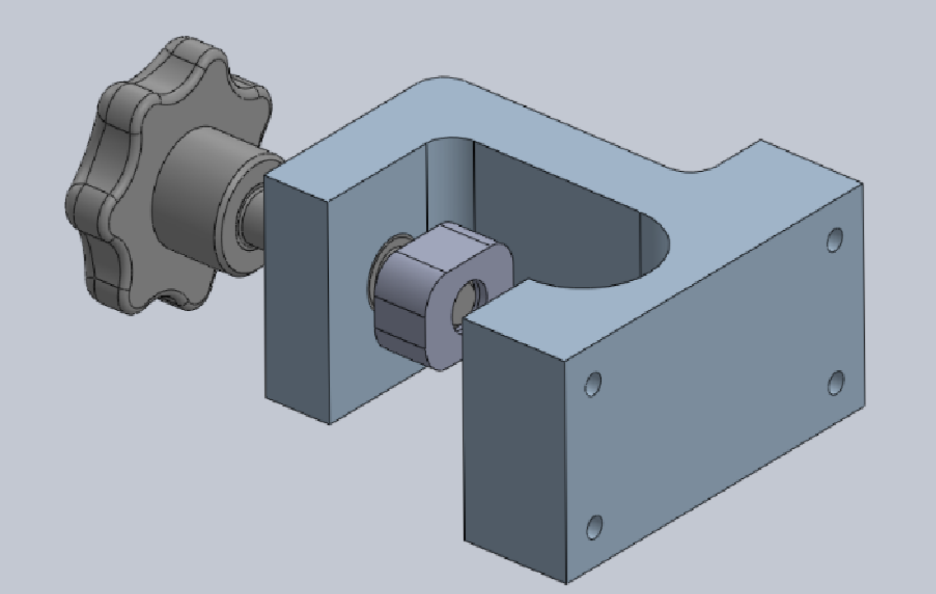


Figure 21: Modified Clamp Design

This design replaced the V-shaped support piece with a flat one and added a U-shaped cut into the main body of the clamp. This increases the surface area contacting the IV pole which increases the weight it can support without sliding. This support piece also does not need to spin freely to be tightened, so it can be attached easily to the knob with a heat-set insert. Initially our group had planned to use two of these clamps to support the actuator at the bottom and top, but a single clamp supporting the actuator from the top proved to be enough to support the system.

3.4.2 EVD Connector Design

Our group had two different EVDs to work with for this project. The first was Integra's Accudrain, seen below on the left in Fig. 22. The second was Medtronic's Duet System, seen below on the right.

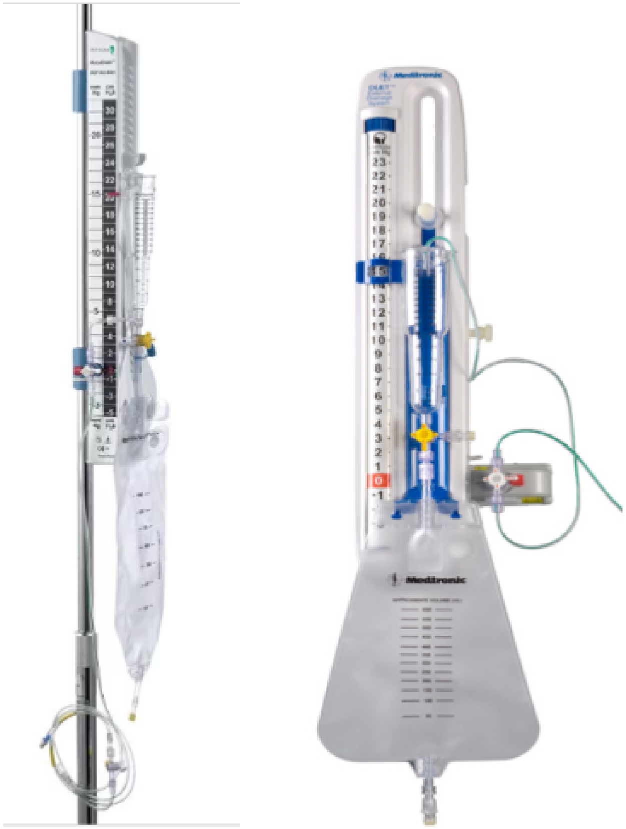


Figure 22: Integra's Accudrain (left) and Medtronic's Duet System (right)

Our group needed to separate the drainage system from the plastic body used to adjust the height so the drainage system could be adjusted by the actuator instead. Given the feedback from our interview, our group had initially hoped to use Integra's EVD in our design. The collection chamber on Integra's design was attached to the main body via a hook, circled in the figure below.

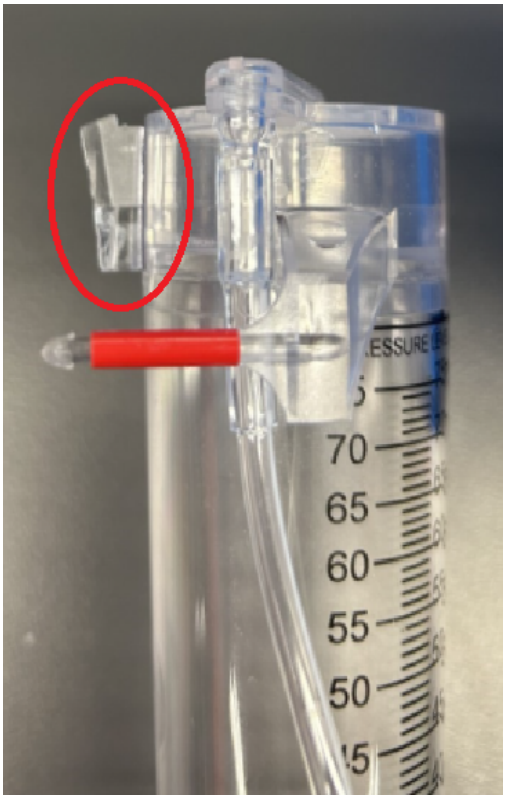


Figure 23: AccuDrain Collection Chamber

A connection piece was designed that would hook on to the collection chamber and screw into the actuator. However, the stopcock on Integra's EVD was glued to the main body, and it proved difficult to remove without damaging the drainage system. Medtronic's EVD had functionality for both ventricular drainage and lumbar drainage, so the stopcock could be easily removed and reconnected to the system depending on the procedure. This made it much easier to disconnect the entire drainage system from the main body and design a part to connect it to the actuator. Therefore, our group decided to build our prototype using Medtronic's EVD instead.

Two parts were designed to connect the EVD to the actuator. The first, seen below in Fig. 24, hooked on to the collection chamber and was screwed into the moving part of the actuator.

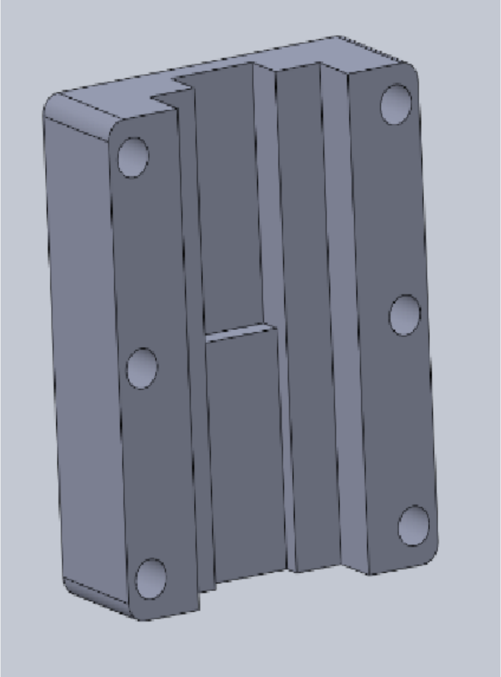


Figure 24: Collection Chamber Connector

The second, seen in Fig 25 below, allowed the stopcock to be connected to the body of the actuator so it would remain static while the collection chamber moved.

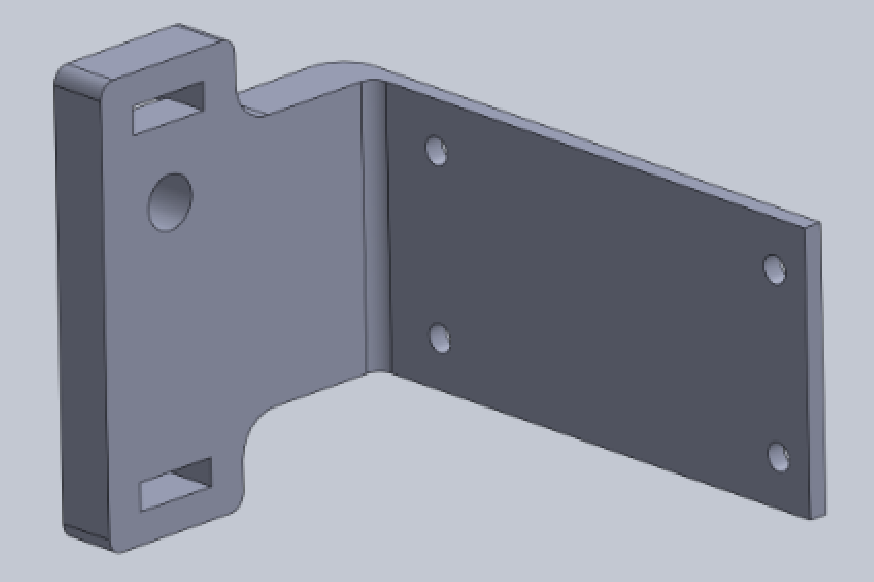


Figure 25: Stopcock Connector

3.4.3 Casing

Overview

The most complex component of the mechanical design was the casing holding the Jetson Nano, Arduino, ultrasonic sensor, camera, and all the necessary wiring for these components. The goal was to design a casing that could hide everything that did not need to be accessed for regular use in a simple shape that would be easy to wipe down after use. The complete casing seen below in Fig. 26 was split into 7 components to allow it to be manufactured.

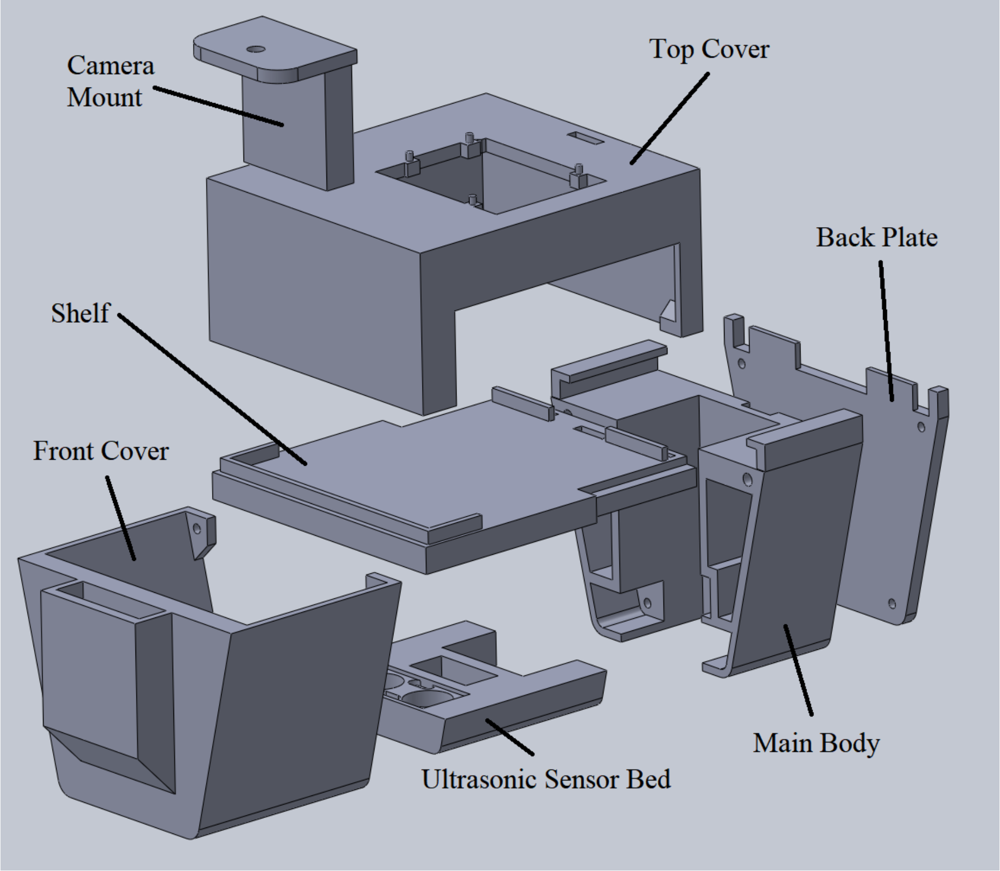


Figure 26: Casing (Exploded View)

Originally our group had intended to control the device entirely with a Jetson Nano, so the casing was designed to allow the wires for the camera, stepper motor, and ultrasonic sensor to be routed internally (the cable channel can be seen in Fig. 27 below running up the back of the main body). Once it was decided that an Arduino would be used instead for demonstration purposes, the casing was modified to allow the Arduino and breadboard to sit on top of the Top Cover. The ultrasonic sensor wires were still routed internally, and the cables for the stepper motor and camera were routed outside the casing.

Main Body

The first of the components making up the casing is the Main Body, seen below in Fig. 27. The Main Body wraps around the linear actuator and stepper motor and connects to the Front Cover, Back Plate, Shelf, and Ultrasonic Sensor Bed. Heat-set inserts were used to allow the Back Plate and Front Cover to be screwed in. The Shelf slides into the groves at the top of the Main Body, and the Ultrasonic Sensor Bed slides into the cutouts at the bottom. The main feature of this part is the cable channel that allows the wires for the Ultrasonic Sensor to be routed up to the Jetson Nano.

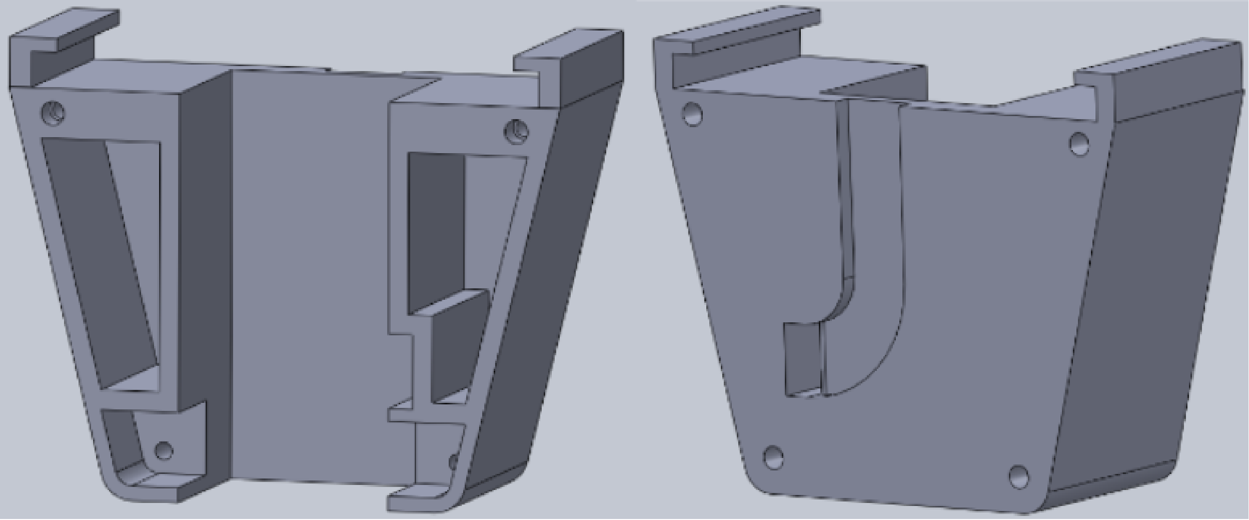


Figure 27: Main Body

Ultrasonic Sensor Bed

The original Ultrasonic Sensor Bed can be seen in Fig. 28 below, and the modified bed is on the right. The original design was going to use 4 heat-set inserts with screws to secure the sensor.

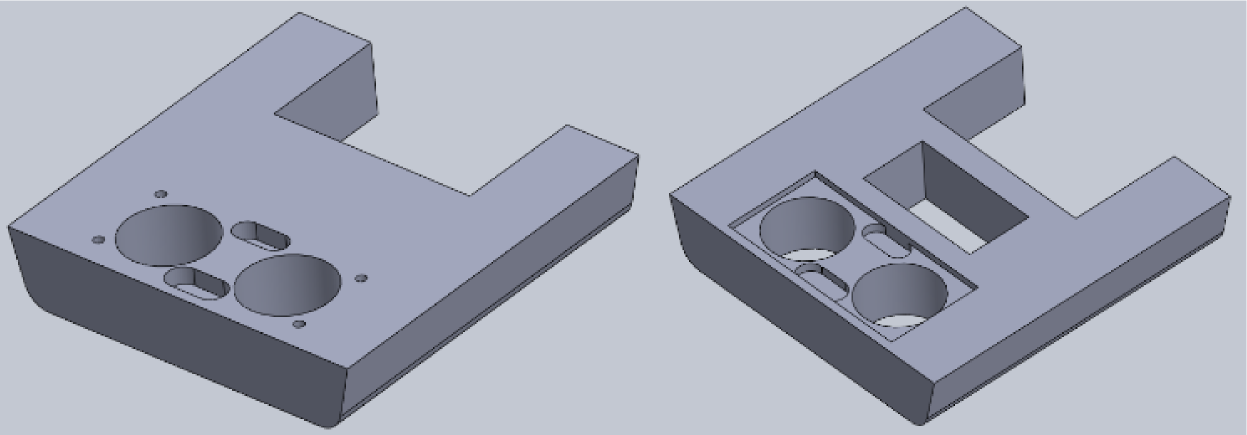


Figure 28: Original Sensor Bed (left) and Modified Sensor Bed (right)

However, the given dimensions for the hole locations in the sensor's CAD model weren't exactly correct, so the holes in the print didn't line up with the sensor. Since the holes were so small there was little room for error on these measurements, so rather than try and get the exact measurements, our group used a different design for the sensor bed that had a recess for the sensor to lie in. A hole was also added to allow the wires from the stepper motor to be routed externally as at this point the Jetson Nano wasn't going to be used.

Front Cover

The front cover seen below in Fig. 29 holds the camera in the back compartment and screws into the Main Body. The large hole at the bottom allows the sensor to see the top of the EVD, and any excess length from the stepper motor wires can be routed in here.

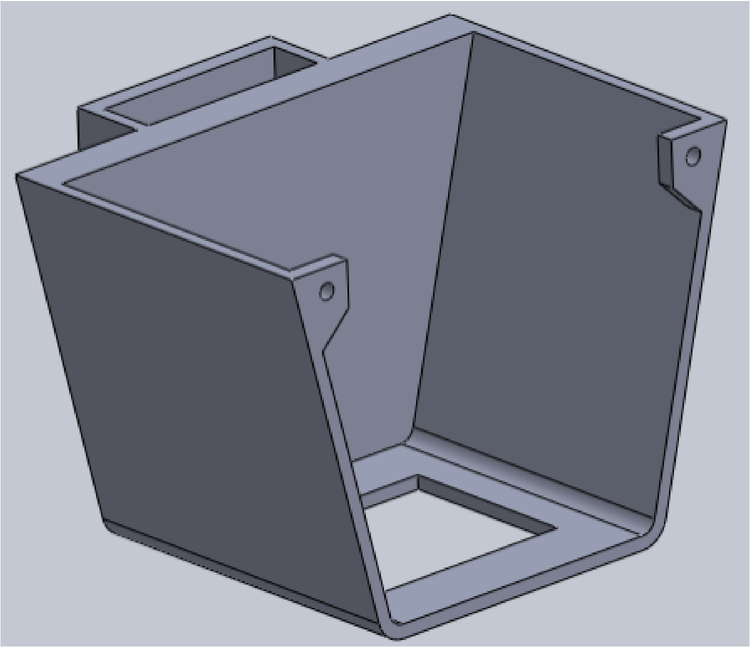


Figure 29: Front Cover

Shelf

The Shelf is supporting the Jetson Nano and needed to connect to the Main Body. The Jetson Nano's frame has small, rectangular cutouts intended for mounting. Our group designed a snap fit joint seen below in Fig. 30, that would be a part of the Shelf and allow the Jetson Nano to clip into the Shelf.

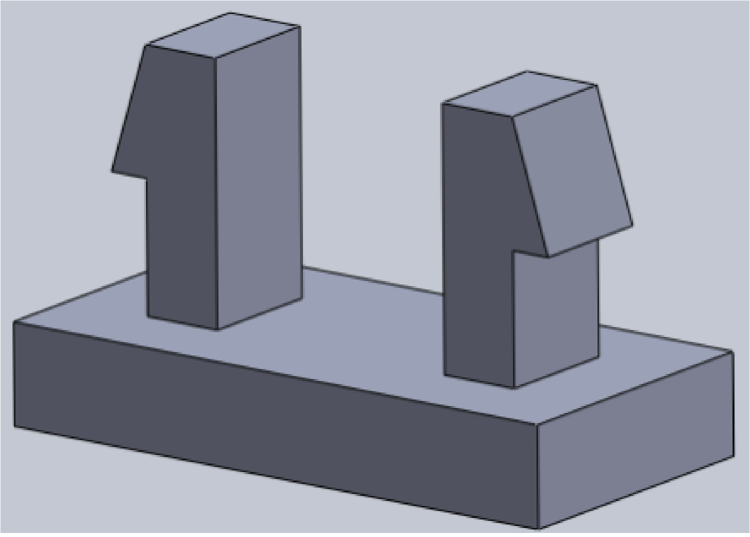


Figure 30: Snap Fit Joint

Unfortunately, this type of joint did not print well at such a small size, so another strategy had to be used. The second iteration seen below in Fig. 31 used small walls to prevent the Jetson Nano from sliding around.

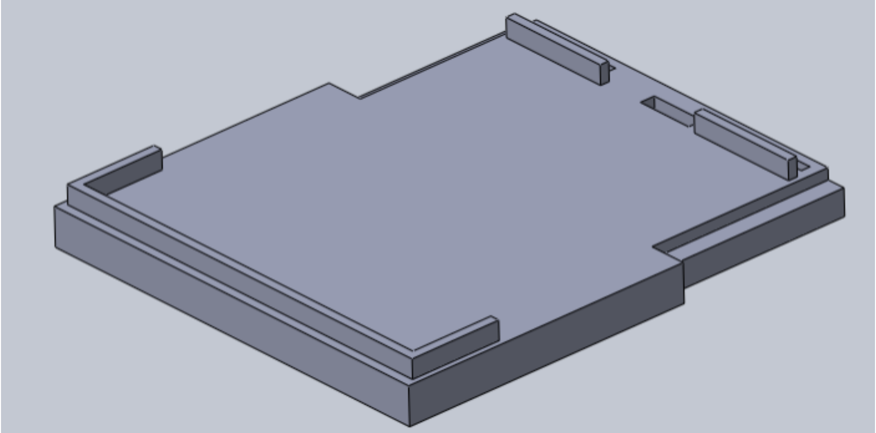


Figure 31: Shelf

Top Cover

The final piece of the casing is the Top Cover, seen in Fig. 32. This was originally intended to just shield the Jetson Nano and allow access to the I/O panel, but was changed to allow the Arduino and breadboard to sit on top of it. A slot was added on the top to allow the Ultrasonic sensor wires to reach the Arduino, and a cutout with mounting pins was created on the top of the cover for the Arduino to rest.

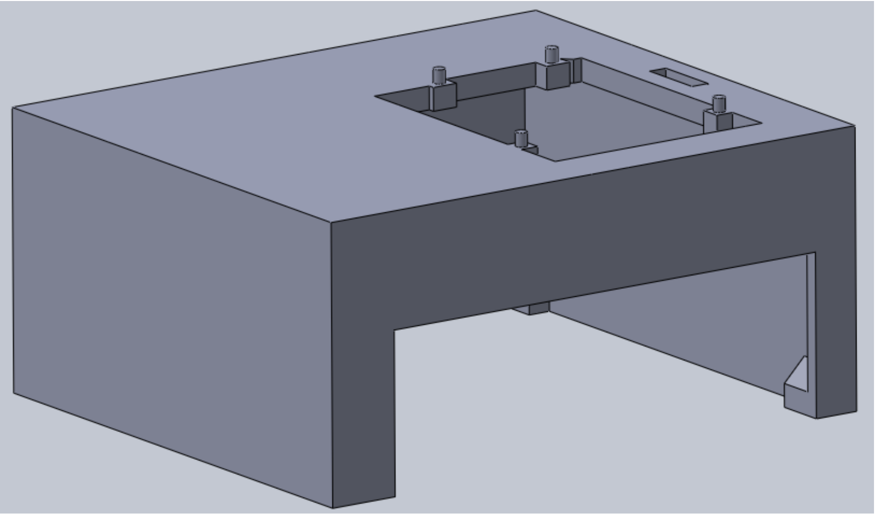


Figure 32: Top Cover

3.5 ICU Requirements

Developing User-Friendly and Aesthetically Pleasing Elements for ICU Use: 1) Designing interfaces optimized for different lighting conditions in the ICU, ensuring readability and high contrast. 2) Choosing materials and creating interfaces that facilitate effortless cleaning and sanitation while complying with ICU hygiene standards.

Supplying Enhanced Features for ICU Situations: 1) Customizing alert settings for individual patient care, such as combining buttons or controllers. 2) Offering timely-response controls for critical situations, an essential feature in ICU environments.

Ensuring Integration with ICU Systems: 1) Guaranteeing compatibility with the existing ICU monitoring systems and electronic health records (EHR). 2) Ensuring consistent and secure data transmission is crucial for seamlessly monitoring patients in real time.

Conducting Testing and Validation in ICU Environments: 1) Conducting careful clinical trials and simulations in simulated ICU settings. 2) Obtaining and incorporating input from ICU professionals to enhance the HMI interface.

4 Experiments and Validation

In this section, we will introduce the procedure and outcomes of validation for our system. The accuracy of our team's motion-tracking algorithm will undergo testing through direct comparison with data from the established VICON motion capture system. Our objective is to refine our algorithm through multiple iterations, aiming for a level of error that falls within an acceptable range (±$30mm$ more details will be shown in the results section).

4.1 VICON Motion Capture System

Vicon is one of the key players in optoelectronic motion capture systems based on markers. The trademark is often used as a proprietary eponym for these systems. Other manufacturers, such as MotionAnalysis (Santa Rosa, CA, USA), Optitrack (Corvallis, OR, USA), or Qualisys (Göteborg, Sweden), also exist [24]. In our case, the system consists of 10 cameras and can measure an object's location with a few millimeters of accuracy. From related articles, we found that in static experiments, the Vicon system had a mean absolute error of 0.15 mm and a variability lower than 0.025 mm [25]. In dynamic experiments, the Vicon system error is less than 2 mm [26].

4.2 Validation Procedure

In this section, we will introduce the setup procedure of our validation process. In general, it can be described in two steps. First initiating motion tracking and setting the desired height. Second, change the height of the head position and the distance from the camera to the head.

The figure (Fig 33) below shows the two steps of setup: one for head position and the other one for camera position. Because of the limitation of the motion capture room, we use the number of pillows to adjust the height of the head position. At first, we initialize the head motion tracking algorithm to recognize the patient's head and start tacking the settled interesting point (in this case, is the right Kocher's point). The second step is to use the Vicon motion capture system to record the current position of the head in the world frame, and the tracking algorithm starts tracking and calculation. After the coordinate is recorded, the third step is to add another pillow to the patient's head, and the motion capture system will record the new position. After that, in the fourth step, the tracking algorithm will record this position and calculate the vertical displacement. In the fifth step, we compare the outputs from the Vicon motion capture system and our tracking algorithm to see the difference.

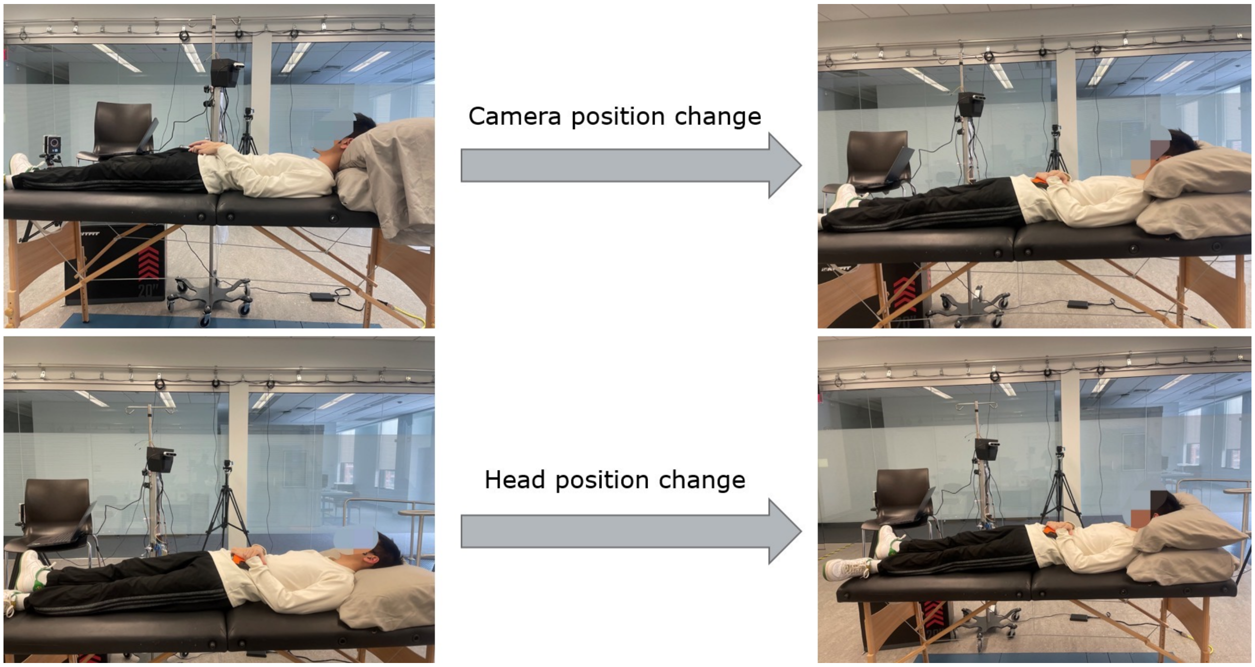


Figure 33: Validation experiment setup

4.3 Validation Results

In this section, we described the outputs from our head motion tracking algorithm and compared them with the ground truth in four different camera positions with the same head vertical displacement. Table 1 compares the output from the current setting, which uses the Intel Realsense D435i depth camera. This camera can measure the depth information (in our case, the distance from the tracking point to the camera) and in theoretically can have a better measurement in vertical displacement based on the aforementioned formula (equation (2)). The results in Table 2, which only uses a Logitech Webcam (2D camera) and depth estimation, prove our hypothesis. The experiment results in the two tables below validated our choice of camera type and the head motion tracking algorithm.

Table 1: Intel Realsense D435i: Different between measured displacement to ground truth displacement, all in millimeters

Distance to Head	Ground Truth Displacement	Measured Displacement
700	83	118
900	88	126
1100	78	122
1300	87	110

In order to maintain the EVD drainage chamber at the same relative height as the patient's Kocher's point, we directly output the height difference from the head motion capture system to the Arduino, which

Table 2: Logitech Webcam: Different between measured displacement to ground truth displacement, all in millimeters

Distance to Head	Ground Truth Displacement	Measured Displacement
700	83	142
900	88	164
1100	78	155
1300	87	172

then drives the motor to move the same displacement. We simply need to make sure that the movement recorded by the head motion tracking system is within an acceptable range because the drainage chamber's movement is confirmed by the differential encoders and ultrasonic sensors. Our method has errors between 20 and 40 mm, according to a comparison between the depth camera motion tracking system and the VICON motion capture system. According to an article, persistently elevated ICP can cause progressive cerebral ischemia, herniation symptoms, or even death. The normal ICP range is 7 to 15 mm Hg.[27], and the dangerous zone is above 22 mm Hg. So, we have a 6 mm Hg window of acceptable misalignment based on the normal ICP range. In our situation, the range of 20 mm to 40 mm corresponds to around 2 to 3 mm Hg in conversion (13.5 mm = 1 mm Hg in scale). So, the validation results satisfied the minimum requirements.

5 Conclusion and Discussion

Our team investigated how existing EVDs function in Neuro-ICU settings and what their primary problems are. From here we developed a design plan for creating a device that would relieve patients and nurses of stress in a way that was easy for nurses to adopt. We present a functional prototype that proves the concept of a vision-based automated EVD leveling system to regulate patient ICP. This prototype has been validated with acceptable outcomes in a simulated environment. The following sections outline areas of future work the team identified for this project to further refine it.

5.1 Discussion on Mechanical Design

The final clamp design functions well for the prototype, but could use some adjustments for a final product. The Onyx filament is strong enough to withstand the bending from the weight of the system, but machining a clamp would be better for long term use. This clamp is also designed to work with 1" diameter IV poles. It could be used on poles smaller than this but would be less effective at gripping the pole, and it cannot be used on IV poles larger than 1" in diameter. A V-shape could be built into the body of the clamp itself instead of the U-shape our group used to allow functionality with IV poles of different diameters.

Rubber pads could also be added to the clamp where it contacts the IV pole to increase the grip strength.

The final designs for the EVD Connector pieces were also effective for the prototype. Being able to adjust the height of the stopcock on the actuator allows the system to work easily on beds of different heights. However, adjusting this height requires loosening four different screws which would not be ideal for a final product. As the design is refined more, creating a slider for the stopcock to allow it to move up and down without needing to adjust multiple screws would make the design much more convenient. Designing a part that allows the EVD to easily be connected and disconnected from the actuator will be important in a final design, as ideally the EVD will be able to be easily added or removed from the system when it is needed.

The casing our group designed was effective enough for a functional prototype, but will need significant adjustments as the electronic components are changed. Once the Jetson Nano is functional, openings can be made externally to allow the fan to ventilate, and internally to allow the camera and stepper motor cables to be routed directly to the controller. Additionally, an I/O shield for the Jetson Nano and its Wi-Fi antennae should be created to prevent any potential damage to the Jetson Nano from these openings. It will also be important to secure the Jetson Nano more permanently to the Shelf, which could be done using screws through the openings where the snap fit joints were going to go.

5.2 Discussion on Head Motion Tracking System

The current head motion tracking system is functional for automatically leveling the EVD, but could be improved in two aspects. First, in the current setting, we only use a fixed camera angle position (the angle between the camera and the mount base on the casing) to do the head motion tracking. This setup can cause an error in the vertical displacement calculation if the camera gets bumped and the angle changes. In future product design, this angle should be considered in the calculation of the vertical displacement so the device can function even if the camera angle is changed. The second aspect is in the algorithm design. The current algorithm will lose the tracking point when the patient's face is occluded or only the side view is visible in the camera. This could be improved by using a more comprehensive facial landmark model (e.g. 194 facial landmarks model) or other machine learning structures (e.g. dlib's MMOD CNN face detector). However, these approaches require more computational resources and RAM (Random-access memory) during operation, necessitating a trade-off with real-time responsiveness.

5.3 Discussion on Electrical Components

Currently we use our laptop to run the head motion tracking algorithm and communicate with the Arduino UNO board through USB to control the linear actuator. In future product development, a Jetson Nano can be considered to control the entire system. Jetson Nano is an embedded computing board from Nvidia that has high RAM and a multithreading processor. This computing board will allow us to deploy larger facial landmark models and machine learning structures. It will also allow the device to function on its own without requiring a laptop to function.

5.4 Discussion on Validation

Because the ICP setting differs from patient to patient, our current setting is simply a parameter converting from height measurement to ICP. In future studies, more verification of the true ICP relationship with vertical displacement is vital to ensure our current findings are appropriate. It needs professional advice from neuro-ICU medical staff and other measurements using pressure sensors.

References

[1] R. Muralidharan, "External ventricular drains: Management and complications," *Surgical neurology international*, vol. 6, no. Suppl 6, p. S271, 2015.

[2] M. Dey, J. Jaffe, A. Stadnik, and I. A. Awad, "External ventricular drainage for intraventricular hemorrhage," *Current neurology and neuroscience reports*, vol. 12, pp. 24–33, 2012.

[3] R. A. Yataco, S. M. Arnold, S. M. Brown, W. David Freeman, C. Carmen Cononie, M. G. Heckman, L. W. Partridge, C. M. Stucky, L. N. Mellon, J. L. Birst, *et al.*, "Early progressive mobilization of patients with external ventricular drains: safety and feasibility," *Neurocritical Care*, vol. 30, pp. 414–420, 2019.

[4] S. Munakomi *et al.*, "Ventriculostomy," 2019.

[5] A. C. Flint, S. Toossi, S. L. Chan, V. A. Rao, and W. Sheridan, "A simple infection control protocol durably reduces external ventricular drain infections to near-zero levels," *World neurosurgery*, vol. 99, pp. 518–523, 2017.

[6] A. L. Estrera, R. Sheinbaum, C. C. Miller, A. Azizzadeh, J.-C. Walkes, T.-Y. Lee, L. Kaiser, and H. J. Safi, "Cerebrospinal fluid drainage during thoracic aortic repair: safety and current management," *The Annals of thoracic surgery*, vol. 88, no. 1, pp. 9–15, 2009.

[7] S. P. Mollan, A. J. Sinclair, and G. Tsermoulas, "Cerebrospinal fluid shunting for idiopathic intracranial hypertension: A systematic review, meta-analysis, and implications for a modern management protocol," *Neurosurgery*, vol. 92, no. 3, pp. e59–e60, 2023.

[8] P. J. Howlett, A. R. Walder, D. R. Lisk, F. Fitzgerald, S. Sevalie, M. Lado, A. N'jai, C. S. Brown, F. Sahr, F. Sesay, *et al.*, "Case series of severe neurologic sequelae of ebola virus disease during epidemic, sierra leone," *Emerging Infectious Diseases*, vol. 24, no. 8, p. 1412, 2018.

[9] A. Bertuccio, S. Marasco, Y. Longhitano, T. Romenskaya, A. Elia, G. Mezzini, M. Vitali, C. Zanza, and A. Barbanera, "External ventricular drainage: a practical guide for neuro-anesthesiologists," *Clinics and Practice*, vol. 13, no. 1, pp. 219–229, 2023.

[10] "Digital external ventricular drain (devd) with data analytics integration." https://weilinstitute.med.umich.edu/product-portfolio/devd. Accessed: 2024-4-24.

[11] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[12] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, pp. 504–513, Springer, 2008.

[13] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *2009 IEEE 12th international conference on computer vision*, pp. 32–39, IEEE, 2009.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

[15] C.-H. Lee, C.-R. Lin, and M.-S. Chen, "Sliding-window filtering: an efficient algorithm for incremental mining," in *Proceedings of the tenth international conference on Information and knowledge management*, pp. 263–270, 2001.

[16] P. Sudowe and B. Leibe, "Efficient use of geometric constraints for sliding-window object detection in video," in *International Conference on Computer Vision Systems*, pp. 11–20, Springer, 2011.

[17] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[18] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2008.

[19] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, 2019.

[20] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "3d constrained local model for rigid and non-rigid facial tracking," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 2610–2617, IEEE, 2012.

[21] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1867–1874, 2014.

[22] N. Boyko, O. Basystiuk, and N. Shakhovska, "Performance evaluation and comparison of software for face recognition, based on dlib and opencv library," in *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pp. 478–482, IEEE, 2018.

[23] "Belt-driven versus ball screw actuator: Which is the best choice for your application?." https://www.isotechinc.com/belt-driven-versus-ball-screw-actuators/#:~:text=Belt-driven%20actuators%20remain%20the,along%20with%20highly%20accurate%20positioning. Accessed: 2024-4-24.

[24] A. Pfister, A. M. West, S. Bronner, and J. A. Noah, "Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis," *Journal of medical engineering & technology*, vol. 38, no. 5, pp. 274–280, 2014.

[25] P.-F. Yang, M. Sanno, G.-P. Brüggemann, and J. Rittweger, "Evaluation of the performance of a motion capture system for small displacement recording and a discussion for its application potential in bone deformation in vivo measurements," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 226, no. 11, pp. 838–847, 2012.

[26] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A study of vicon system positioning performance," *Sensors*, vol. 17, no. 7, p. 1591, 2017.

[27] J. Smith and J. Doe, "Understanding the complexities of kidney transplant procedures," *BMJ*, vol. 378, p. e061960, 2022.