Voice Controlled Injector for Medical Procedures

Nicholas Berger, Tai Lu, Steven Morlan
Northern Illinois University
1425 Lincoln Hwy
Dekalb, IL 60115

Abstract—**During the endoscopic procedures, surgeons need to focus and work with both hands. It becomes difficult for the surgeon to control extra equipment with one hand or relay to an assistant on what exactly they need to be injected. This device helps the surgeon with better control and creates a low-cost device. The surgeon can control the device with voice commands or manually via the physical buttons if need be. This device cuts out the middleman and drastically reduces the chance of mistakes by human error. As a result, endoscopic procedures become safer for patients and more convenient for surgeons. This device will be portable by utilizing a battery as a power supply. The control system of the syringe pump will use a raspberry pi to receive input, perform calculations, and send signals to a motor controller. The motor controller, powered separately from the raspberry pi, will control a stepper-motor which has a 3D printed attachment that will push or pull on the syringe to control the flow.**

I. INTRODUCTION

During endoscopic procedures, it can be difficult for doctors to operate an endoscope while also communicating with a technician who is injecting a dye at the site via syringe. Excess volumes of this dye can lead to pancreatitis within the patient [1]. This device will use voice commands to inject the exact amount of fluid and speed requested by the surgeon. The device will also work with standard syringes readily available in all hospitals. Having a device that will inject exactly how the surgeon wants will help streamline this important procedure.

II. MATERIALS AND METHODS

The program starts with the variables set to 0 and the voice software listening for the command word. At this point, the speed and volume variables are available to be altered by the encoders or voice commands and the motor can be moved via the onboard buttons continuously. Once the "Inject" command is given or the inject button is pressed, the variables are no longer able to be changed and the motor will start to move the pump accordingly. The injection can always be stopped via command or pressing the "inject" button again.
The force required to push the syringe simplified with the pulse. The user gives the command of volume flow rate and will be converted into a pulse and push the syringe under command.

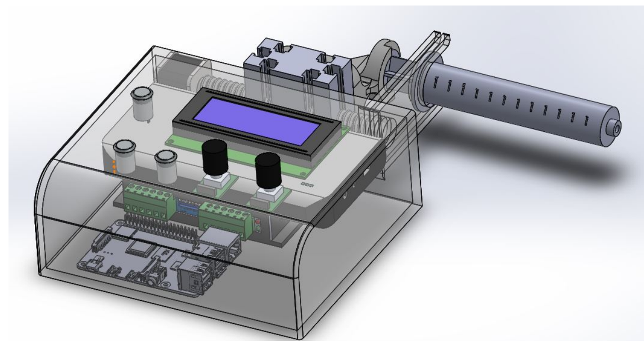

Figure 1. CAD model of final design

Precise control over speed and volume was needed for this project so a stepper motor was chosen with a threaded rod to make the plunger move linearly rather than rotate. Stepper motors stay ridged when not moving, meaning that a patient will not be at risk if the plunger is hit accidentally. A driver was chosen to drive the motor and physical parts were 3-D printed to allow for unique control over the shape of the mounting hardware. Rotary encoders, buttons, and an LCD screen were all used to create a minimalist display that is easy to understand at a quick glance. A battery pack was chosen that can output 5 volts and 24 volts at the same time, allowing both the Raspberry Pi and the motor to operate without needing to be plugged into a wall. It can also be charged at the same time it is outputting, allowing the device to operate through an outlet instead.

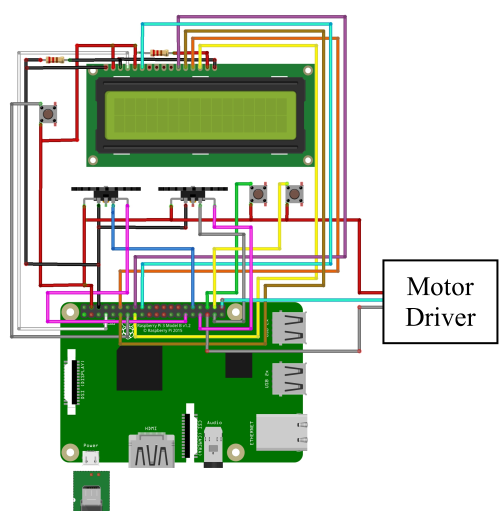

Figure 2. Wiring diagram of device

The voice control system was created by using a multitude of programming and technologies. All voice control related software and processing was located locally on the raspberry pi, allowing the device to work completely standalone and offline. The overall process of the voice control system can be summarized as a few steps. The user first speaks a predetermined wake-word which will trigger the device to listen for commands and make a sound to alert the user it is ready. The user speaks a command and the sound is analyzed by the speech to text program, which sends the command in text form to determine what the intent of the command is. When the intent is identified, it is sent to an intent handler, which executes pre-defined code based on the specified intent.

Voice recognition centered around the software Rhasspy, which is a completely open source voice assistant toolkit. Rhasspy acts like a hub for a variety of voice related tech to work with one another. The wake-word technology used was Porcupine with the default word, "Porcupine," being used. The speech to text program used is called PocketSphinx, which contains its own dictionary and needed no extra training. PocketSphinx would send the translated phrase back to Rhasspy, which would then determine the intent. Intents were labeled by programming phrases with optional or alternative words to allow for leniency within phrases. This programming also labeled a number given as a variable.
Rhasspy sends the intent and variable to the programming software Node-Red through websockets and local host connections. Node-Red would run certain flows of code based on the received intent and variable number.

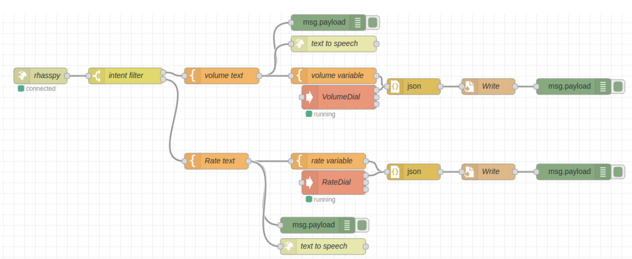

Figure 3. Voice activation code logic

III. RESULTS AND DISCUSSION

The stepper motor required 20 volts to operate. The performance was different than expected due to some technical issues. Finding the correct settings to operate the stepper motor was difficult and it made loud noises when running the command code. Too much current ended up being supplied to the motor and a new one was ordered to replace the melted one. Throughout testing, the motor had no response to the commands, even when no error was given upon execution. A pressure sensor was to be used with a feedback loop to provide a constant flow, changing pressure as the force required to inject at the desired speed changed. The rotary encoders were able to adjust the variables continuously along with the voice activation software by having them both write to a single text file. These encoders are important as they provide an alternate way for a doctor to adjust the settings and bypass spoken commands completely. It was decided that an LCD screen was needed as an interface to inform the user of the current settings and the "ready state" of the device. It was decided that the information on screen should be readable at a glance.

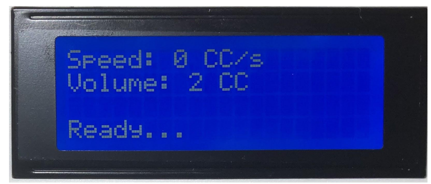

Figure 4. Interface Display

The python code of the LCD, the encoders, buttons, and the force sensor were combined into one script and formatted to all respond in real time. It was decided that two buttons were needed for repositioning the syringe pump and to have the LCD screen say what direction is being activated, if any.

Voice control worked in tandem with manual controls without any issues. Rate and volume variables were changed based on the most recent input and displayed on the LCD. Manual controls had no noticeable delay while voice commands needed a few seconds for processing. The wake-word was very accurate, only rarely not responding to the user, while never responding to any ambient noise or conversations. The wake-word could be detected at both close range and from a distance. Voice recognition would occasionally miss the first part of a command if said too quickly after the wake-word. Individual word recognition was very accurate even with a fan blowing directly on the mic and other ambient noise within the room.

IV. CONCLUSION

The voice-controlled injector takes some of the workload from the doctor. Through local hosting and processing, the unit is portable and completely offline. With high accuracy and stability, the doctor can focus on the surgery without worry about excess dye injections that cause pancreatitis.

Acknowledgement

We would like to thank Hasan Ferdowsi, Simon Kudernatsch, and Justyna Kielar for their guidance in this project. We would also like to thank Northern Illinois University for the tools provided and their flexibility in allowing us to work from home.

References

[1] Erenoğlu, Cengiz, et al. "Do MRI Agents Cause or Worsen Acute Pancreatitis?" *Ulusal Travma Ve Acil Cerrahi Dergisi = Turkish Journal of Trauma & Emergency Surgery : TJTES*, U.S. National Library of Medicine, Jan. 2007, www.ncbi.nlm.nih.gov/pubmed/17310418.