Project Number: P15073   Autonomous IV Stand

Michael Binger
Mechanical Engineering

Caitlin Conway
Electrical Engineering

Nicholas Goddard
Industrial and Systems Engineering

Nicholas Jacobs
Electrical Engineering

Christina Pysher
Biomedical Engineering

Ethan Whritenor
Mechanical Engineering


Figure 1: Functional Prototype

Abstract

An autonomous intravenous (IV) stand is a device that allows for mobile medicine distribution without the need for the patient to maneuver the device. The current IV stand causes patients to use the stand for stability when walking, has a large base that causes trip hazards, and is difficult for weak patients to maneuver. The IV stand became prominent in the 1950's and the design has not experienced much modification since then. This project modifies the existing IV stand by developing and implementing an autonomous system that will follow a patient around a medical facility while avoiding objects. Other outcomes include improvements to the design of the stand such as a smaller base, the ability manually maneuver, the ability to provide stand height adjustability, and the ability to hold more than 4 IV bags.

The autonomous IV stand that we built, displayed in Figure 1, performed well and met our expectations for a first iteration project. We see this being a viable product that could potentially be used in hospitals and other healthcare facilities in the future.

Introduction

It is common for patients to be administered medication or nourishment utilizing an intravenous system due to their medical conditions. While patients spend time in the hospital, they may be required to ambulate a certain amount of time per day or can become bored and want to take a walk. Patients that have IVs must pull an IV stand around the hospital while walking. Patients are often in a weak or fragile state and often times struggle to maneuver and manipulate the IV stands, especially when using a walker. Hospital staff also has their complaints about traditional IV poles, some of which are shared with the patients' opinions. For example, both patients and staff complain about the bulky base of IV stands that can cause issues with maneuvering the stand around the hospital.

Design Process

The scope of this project is to create a functional prototype of an autonomous IV stand that will meet the identified customer requirements. Since this is the first iteration of the project, without a company sponsor, the customer requirements were created through a series of on-site interviews with nurses and other medical professionals to see what problems exist with current IV stands. These requirements include: following a patient, avoiding objects, safe for a hospital environment, must be cordless, holds more than 4 IV bags, and is height adjustable.

From there, the different components/functionalities required and their associated engineering requirements were defined. These include a maximum speed of 5 feet/second, a maximum acceleration of 2 feet/second^2, a maximum distance of 3.5 feet from a patient, 2 hours of run time on a charge, display an error status, respond to a detected object within 50 milliseconds, avoid objects by 2 feet, provide 24 inches of height adjustability, reduce the diameter of the base to 24 inches, and keep the material cost of the unit under $1000. The design process was geared towards fulfilling these requirements.

Motors/Omni Wheels

Figure 2: Base w/ Motors

In order to choose a wheel assembly, multiple designs were generated. Ultimately, the design selected involved omni-directional wheels (onmi wheels) mounted to alternating sides of an octagonal base as shown in Figure 2. The use of omni wheels allows for position control precise enough for tight turns and sudden changes in direction. The wheels also allow for manual maneuvering in all directions. Once the wheel assembly was selected, a motor needed to be selected. The motors were selected based on the calculated required torque, 14 ft-lb, desired speed, 5 ft/sec [3], and the presence of an encoder for use in coding. The primary function of the encoder was to record the speed and provide feedback. With the 8 inch wheels, the motors output 16 lbf and are capable of 240 rpm which will meet the stand to the desired velocity needed to follow patients. The design to connect the motors to the omni wheels, needed to translate the motor torque to the wheel torque. A wheel hub and motor shaft, viewed in Figure 3 were machined out of aluminum and connected with setscrews seen in Figure 5 to ensure the wheels rotate at the same speed as the motors when power is supplied. The motors were secured to the base using the mounts displayed in Figure 4.

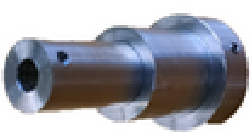

Figure 3: Motor Shaft

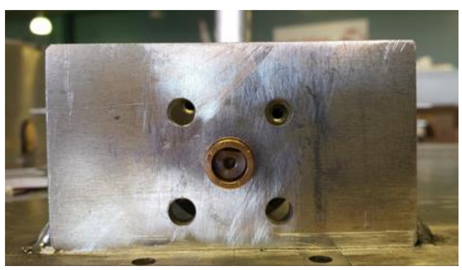

Figure 4: Welded Motor Mount

Figure 5: Setscrews in Wheel Hub

Patient Tracking System

The project requires following a patient through a hospital environment at an optimal distance producing a need to have a method for determining the location of the patient relevant to itself. The design requires the use of a position vector to determine the location of the patient relative to the device. The position vector is the overall distance between the patient and the device with a measurement of angle between the direction of motion perpendicular to the front plane of the device and the actual position of the patient. The patient is tethered to the device via a nylon twine which is attached to a gait belt that the patient will wear around their waist. The position of the patient and the angle measurement are produced by two encoders. The angular measurement does not require a high degree of resolution so the team decided that a potentiometer, depicted in Figure 7, acting as an encoder would produce sufficient results. The distance measurement from the device to the patient is more crucial and the team decided to use a 10-bit rotary encoder from US Digital, depicted in Figure 8, which is the max resolution that the Arduino Mega 2560 is compatible with. The device also includes a fail-safe that will stop the system and alert the patient and nursing staff that there is a problem.

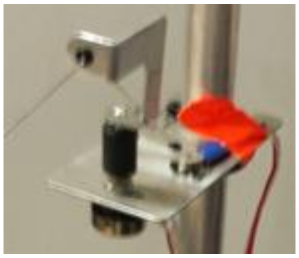

Figure 6: Pole Platform

Figure 7: Potentiometer

Figure 8: Encoder Assembly

Figure 9: Encoder

In order to produce the vector with position and angle, the two components are connected in-line with nylon twine. The nylon twine runs from a retractable tether unit around a pulley wheel which is fixed to the rotary encoder shaft, depicted in Figure 9. The Arduino program reads the 10-bit rotary encoder interpretation and converts it to a measurement in inches. The line is then run up to a platform mounted on the pole, depicted in Figure 6, where the potentiometer is located, which is 35 inches above the ground. This height was found to be the average waist height of males and females 13 years old and above, which we calculated using NHANES anthropometric height data [1] and the waist height coefficient from the link length mannequin [2]. The Arduino interprets the position of the potentiometer shaft as a number from 1 - 1023 (10-bit). The algorithm defines three ranges for these numbers. Each range determines whether the patient is left-of-center, right-of-center or centered, relative to the device. This allows the Arduino program to determine if the device needs to turn to the left or right in order to "face" the patient.

The final design includes a fail-safe that is activated if for some reason the patient becomes too far away from the device. Situations in which this could occur include: the battery dies, the device runs into an obstacle that it could not avoid, or the patient moves too quickly. Though these circumstances are unlikely, the fail-safe will stop the system and provide an alert for the patient to stop moving and for the nurses to respond with assistance. A secondary tether will be attached to the gait belt using a clip and the other end of the tether will be attached to the device using a magnet. Located adjacent to the magnet is a Hall Effect sensor which detects magnetic fields. If the magnet is in place, the device will continue to operate normally. If the magnet has been removed as a result of the patient becoming too far away from the device, the device will stop any attempts at maneuvering and sound an audible alert.

Object Avoidance

The main function needed for the autonomous IV stand is object avoidance which will be achieved through the use of two ping sensors. The ping sensors will be aimed at an angle just to the left and right of where the patient will be standing to evade detecting the patient as an object to avoid. The maximum range of the ping sensor is 10 feet straight out from the sensor. Figures 10 and 11 show the detectable range of the ping sensors. Figure 10 depicts a ping sensor viewing a round object while Figure 11 is viewing a flat object. The stand will alter its course if an object is detected and is within 2 feet of the ping sensor.

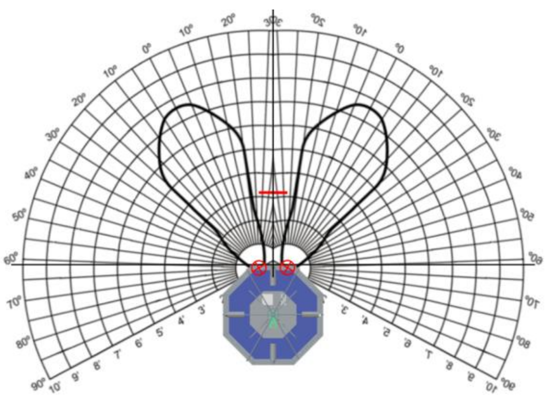

Figure 10: Round object detection

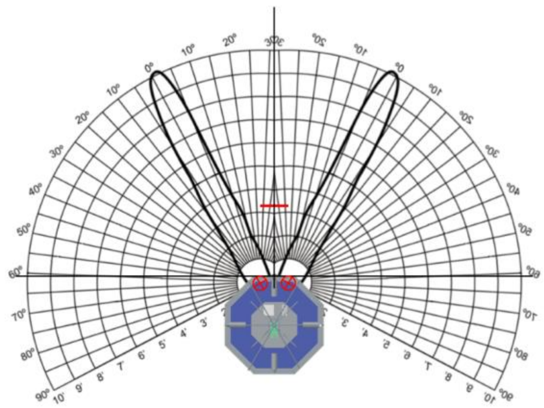

Figure 11: Flat object detection

Battery

Calculations were made to determine the power required to operate all devices on the IV stand, which were then used to determine the proper battery to power the IV stand. Power was calculated by multiplying the voltage required for each device with the current drawn by each device. The device that requires the most power in the circuit at 50.4 Watts (W) is a single motor and totals 201.6W for all four motors combine while all other devices require approximately 30W. Another consideration used to choose the battery was that the motors required 24 Volts (V) in order to operate with the desired functionality needed for the IV stand. To fulfill the above requirements a 24V and 10 Amps (A) battery was selected. Additionally a 24V to 12V step down voltage converter will be used to power the microcontroller and all other devices which require 12V. Another feature of the selected battery is that it has separate discharging and charging terminals which were wired to an ON-OFF-ON rocker switch ensuring that the circuit is not powered simultaneously while the battery is being charged.

Microcontrollers

The logic microcontroller used for this project is the Arduino Mega 2560, Figure 10. This device has 54 digital I/O pins and 16 analog pins to support the various sensors and interfaces that the microcontroller must communicate with to provide proper functionality. Originally an Arduino Uno was sought, however, the sensors take up 22 pins and the Uno lacks this amount. Further thought was given to using the Beaglebone Black controller. This device has more pins and significantly more memory coming in at 4 Gb as opposed to the Arduino Mega's 156 Kb. However, The Beaglebone Black runs using Ubuntu. Seeing as no team members have skill in this language, we decided to use the C-based Arduino model. The trial codes first tried confirmed that the Arduino can handle the logic needed for it to perform the avoidance and follow algorithms.

Additionally, two RoboClaw 2x5A motor controllers are used, Figure 11 and Figure 12. These control the power flow with logic while providing feedback from sensors and encoders. The encoders are attached to the motors and fed into the RoboClaw controllers. These give the position and velocity data needed to properly calibrate and control the motor speed. Also, the RoboClaw has onboard temperature and battery sensors to provide feedback on the condition of the main battery, logic battery, and board temperature. This gives valuable information for status updates and the presence of a terminal condition. The controllers are equipped to process the logic sent to them and have a complete Arduino interface library.

Figure 10: Arduino Microcontroller

Figure 11: Roboclaw Motor Controller

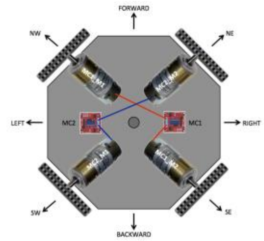

Figure 12: Motor Control Diagram

<u>Programming</u>

To begin the program, the required functionality was pulled from the customer requirements and used to create a pseudo code. This robot has to first pull information from 2 ping sensors, 5 encoders, 1 potentiometer, hall effect sensor, and the RoboClaw motor controllers. This information is put into the logic section of the code and/or displayed onto the LCD information prompt. The information prompt displays the welcome screen, distance traveled, main battery, logic battery, and error status. The distance is determined by reading the encoder value on the motors and dividing the value by the inches of the wheels respective circumference. This gives an overall distance of the forward motion the machine travels. The logic has to incorporate all available data to determine the respective location of the robot in relation to the patient and its surroundings.

The first attempt to make the robot mobile was done using a case number for every instance of sensor values and an action to be done until it came back to the main loop case of searching the other sensors. This inevitably produced very jerky results by having a delay between getting back to equilibrium and searching for the next corrective action. More detrimental however, this process used significantly more memory. This pushed the Arduino past the 156 Kbps available for processing, causing the need for a new method.

This next method uses layers to keep the last state and present state only in the memory. Based on the values of the present state against the previous state, an acceleration and final velocity was chosen for the 4 motors. These values will adjust the speed to bring it near the equilibrium value and have a slight oscillation around the ideal value. These accelerations are controlled by the potentiometer and the encoder attached to the patient string. The potentiometer controls the angle measurement and determines the horizontal motion required to equalize it. The encoder counts revolutions to determine the overall speed of the patient and maintain a distance. This method uses significantly less memory. Furthermore, the effect of this on a calibrated system results in a strafing motion. This means that the front of the robot stays more or less directly facing the patient as the robot moves on slanted axis. This is possible because of the maneuverability offered by the omni-directional wheels. These take the velocities of each wheel and produce a resultant motion. This is controlled by the separate motors to create a fluid maneuver behind the patient as is required by the customer requirements.

The avoidance algorithm plays as an interrupt loop within the previous follow algorithm. It uses the TimerOne interrupt library to constantly check the ping sensors for proximity to an object. This value is read into the program to manipulate the definition of the equilibrium angle. This means that the potentiometer will read an offset value as the center. This is set by the nearness of the wall to the robot. The robot will always maintain a minimum distance from the wall while following the patient. The wall is determined by an object that maintains its presence for a full second. This prevents the algorithm from misidentifying oncoming traffic as a wall. Should the device encounter more obstacles than it can handle or other malfunction the buzzer will sound and the motors all will slow to zero until reset. These methods will meet all the customer requirements while providing a comfortable fluid motion for the patient in question.

Base and Cover

The base plate imaged in Figure 13 was designed to provide enough room for the all electronic components, a cooling system, all four motors, and clearance for the IV pole. The octagonal shape was chosen to accommodate the traditional trajectory of the omni wheels. We selected 0.25" aluminum as the material due to its ability to support the weight of the IV stand. We were originally considering 0.25" UHMW, however FEA confirmed that the UHMW would deform under the stress caused by the weight of the IV stand with 8 attached IV bags.

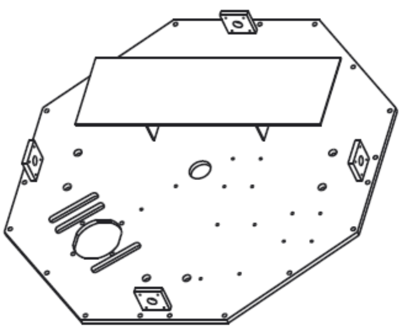

Figure 13: Base Plate

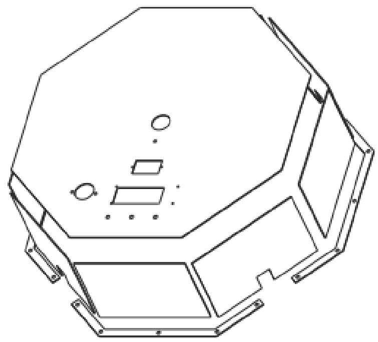

Figure 14: Base Cover

Figure 15: Ping Sensor Placement

The cover for the base was designed to provide a mounting location for the LCD Screen, LEDs, and ping sensors. The positioning of the ping sensors can be seen in Figure 15. It was designed to fit tightly over the base plate by closely following the outline of the base plate. A 0.5" flange surrounding the base allows the cover to be screwed to the baseplate. We selected 0.06" Aluminum sheet metal to allow welding. Sheet metal was selected because the base cover does not hold a significant portion of the weight of the IV stand. It was constructed by water jetting the individual pieces then welding them together. Figure 14 demonstrates how the individual pieces were assembled while Figure ZZ shows the completed structures.

Adjustable IV Pole

A pole system with the ability to adjust at least 24" and carry more than 4 IV bags was required. This was done by using 4 foot telescoping poles and a push button to hold them in place. In order to obtain the large range of heights, six holes were placed in the bottom pole at 2-inch intervals and two holes were placed in the top pole 13 inches apart. This allowed for a maximum height adjustment range while providing a more ergonomic control system. The maximum height is 85 inches and the minimum is 63 inches. The push buttons are pressed down with the user's thumb while raising or lowering the top pole (the inside telescoping pole) to the next hole on the bottom pole (outer telescoping pole). Figure 17 shows the distribution of holes in the poles. In order to increase the capacity for IV bags, Medihooks, temporary bag hooks which can be added at will to the existing hook structure, Figure 16 [4].

Figure 16: Medihooks

Figure 17: Telescoping Poles w/ IV Hook Attachment

Results and Discussion

The first test we completed after full system integration was the battery life test. In the test plan results section in Figure 18 you can see that the stand will be able to operate longer than our desired 2 hours easily with no problems with regard to voltage or current lose. We additionally tested the speed and acceleration of the prototype with a person tethered to it. We did not collect values for these as this is determined by the coding and the weight on the wheels. The motors are capable of meeting our requirements in these areas, but seeing it work we determined it would never need to move that quickly in a hospital setting. If the stand is following a patient, there shouldn't be a need to be following the patient at 5 ft/s. We initially thought we could determine the acceleration and speed of the stand through the motor encoders, but we figured out close to the end of the semester that this would not be possible. Our sensors and feedback worked as we expected and responded almost immediately to the change in sensor values from the potentiometer and encoder. We measured the height change of the pole and it met our requirement of exactly 2 feet of adjustability. We also wanted to decrease the width of the base of the stand. This was measured after it was machined and found to be 26 inches. This is above our target value, but within the tolerance we set in the engineering requirements. The only requirement that was not met was in regards to our budget for the prototype. We received $3,000 from the Simone Center for Innovation, but we wanted to keep our costs below $1,000 due to the cost of currently marketed IV stands. We ended up spending around $2,167 on all parts replacement parts included due to unexpected issues. We calculated that it would cost around $1,450 for all essential materials for one prototype to be built at this stage..

TIme Elapsed (minutes)	Voltage (V)	Notes
0	25.81	
10	25.41	~13 min.
20	25.31	~24 min.
30	25.31	~30 min.
40	25.28	~40 min.
50	25.22	~49 min.
60	25.18	~60 min.
DIff. of Voltage	0.63	

Figure 18: Battery Life test results

Conclusions and recommendations

Obviously we are all very happy that this project was a success. For a first iteration prototype to complete all its requirements is amazing. We did want the budget for the overall prototype to be less, but maybe with a second iteration of this project it can be re-design and meet this goal. We have many things that we would have liked to do differently and things that we recommend for future work. To start with the motor mounts and the shell that covers the base could be re-designed. The motor mounts instead of welding them to the base could be screwed into the plate through the bottom. The shell could be re-designed to be two parts in order to make it easier for electrical component connections and eliminate the need to remove the entire potentiometer subsystem with shaft collar when attaching or removing the current shell. We recognize the need to cover the top of the base plate with an insulator given our many boards that got fried due to lack of insulation.

Besides these things we also have a few recommendations for future work. We recognized need for a way to absorb the shock on the wheels and motors when going over elevated surfaces such as door lips and outlet covers on the floor. Also we had initially desired this prototype to not be tethered to the patient, but since this wasn't implemented we see this as a logical next step to make it more accommodating to a wider range of patients. We also see the potential to design ways to hold other common medical equipment such as: oxygen tanks, infusion pumps, etc. The need for a microcontroller that can perform multiple processes at once can't be overstressed. There are multiple sensors with this prototype and some really need to be monitored at all times to change the direction of the prototype. Lastly, we recognize the need for a computer engineering student for a future iteration of this project. Our electrical engineers did their best with the programming, but it we believe it would be a more robust system if programmed by a CE.

References

[1] Anthropometric Reference Data for Children and Adults: United States, 2007-2010. Web. <http://www.cdc.gov/nchs/data/series/sr_11/sr11_252.pdf>.

[2] *Drillid R. Contini, Body Segment Parameters, New York, New York: Office of Vocational Rehabilitation; 1966 Report No. : No. 1166-03*

[3] *EML2322L – MAE Design and Manufacturing Laboratory* (n.d.): n. pag. Web. 27 Sept. 2014. <http://www2.mae.ufl.edu/designlab/motors/EML2322L%20Drive%20Wheel%20Motor%20Torque%20Calculations.pdf>.

[4] Medihook product Description. NP Medical. Web. <http://www.npmedical.com/products/accessories/medihook>.

Acknowledgments

Michael Zona Project Guide, Richard DeMartino in the Innovation Center at RIT, Professor Slack, Dr. Sahin, Brad Gouldthorpe, John Bonzo in the RIT Brinkman Lab, Jan Maneti and Rob Kraynik in the Machine Shop.