https://doi.org/10.2352/ISSN.2470-1173.2021.17.AVM-214

Design, Implementation, and Evaluation of a Semi-Autonomous, Vision-based, Modular Unmanned Ground Vehicle Prototype

Doncey Albin, Steve Simske
Colorado State University, Fort Collins, Colorado USA 80523

Abstract

In some traditional development processes, engineering teams communicate their subsystem interfaces without much overlap of their respective disciplines and processes. However, for a systems engineering-driven design, a holistic, multidisciplined approach is implemented from the ground up, with considerable overlap between the teams in every phase of the project. Approaching a system from a holistic perspective, rather than an isolated subsystem perspective, is a fundamental component to rapid prototype development and successful system integration. It is also required for full project-level concerns such as the data, security, safety, and sustainability operations. This paper presents the development of a prototype modular unmanned ground vehicle (UGV) used for fire detection and elimination. Taking a systems engineering approach, the mechatronics and control systems designs are performed first, then the system and the important subsystems are built and tested, and finally, the evaluation results are fed back for the next prototype iteration. The goal of this paper is to give engineering students and professionals an example of the process behind holistic development of a semi-autonomous UGV and to begin an inexpensive, readily-modified platform for engineers to build upon.

Introduction

The advent of autonomous robotic systems is just beginning and their utility for modern life is already evident. Today, these systems can perform tasks that are nonideal for most humans, such as work that is dangerous or that requires consistent precision and accuracy [1]. Some of these systems take shape as 6-axis manipulators and legged robots [2], such as Spot Mini from Boston Dynamics [3], but there are many other platforms for systems today. One of the common platforms autonomous robotic systems take on are UGVs [4].

UGVs are versatile and have features that make them a better candidate for certain tasks, in comparison to other platforms. Commonly, UGVs are not legged robots, such as Spot Mini mentioned above, but instead are propelled by wheels [5]. The specialty for UGVs is that they usually don't require additional control for stabilization, and they can run much longer than other mobile systems, as they can easily carry heavier batteries. Additionally, battery exchange and replacement is easier for UGVs, meaning with only minimum interruption they can run continuously.

As engineers work together to design such systems, they will need communication and workflows that simultaneously address each of the various disciplines that go into the project. This is due to the mechatronic nature of robotic systems in general. For example, any changes in the electrical components may change the form factor or mechanical designs to the system, while changes to the structural architecture of the system will affect the software control algorithms.

Having a holistic, transdisciplinary approach to designing UGV systems or any other robotic system increases robustness and system synergy, as engineering integration is optimized. Examples of failed engineering integration include major catastrophes like the explosion of the Space Shuttle Challenger and the Tacoma Narrows bridge collapse [6].

The design, implementation, and evaluation of the UGV presented in this paper utilized a holistic, multi-disciplinary system design approach during each stage of its development. To further augment the ease of the UGV's development and testing, modularity was emphasized throughout the design- including the mechanical, electrical, and software design. Modularity is the ability for a system's components to be separated and recombined, which allows for system flexibility and variety [7].

A system having modularity allowed for quick design modifications, saved resources, and allowed for simple subsystem identification. As the UGV is later modified and improved, its modular design will help engineers implement modifications without lengthy design changes.

This paper is organized as follows: First, the design requirements of the UGV are explained. Then, the methods and manufacturing of the UGV is examined, which includes the design and implementation of mechanical components, electrical components, and software. Next, the process for UGV system functionality and systems engineering evaluation is described. Lastly, the paper discusses results from the development and testing of the UGV, including systems engineering considerations.

Design Requirements

The UGV system presented in this paper will need to meet the following design requirements:

1) Each subsystem needs to incorporate modularity- they need to have "plug and play" ability.
2) The system must be able to detect and track a fire in multiple lighting conditions. The camera sensitivity must be adjustable, just as the rest of the vehicle is modular.
3) The UGV needs to respond to user and sensor inputs with less than a one second delay detect target.
4) A turret with 180 degree panning and tilting capability must be designed and implemented to both hold the vision module as well as the actuator used to extinguish the fire.

These are functional design requirements. In addition, there are data, security, safety, and sustainability requirements. Data requirements for a UGV are tightly coupled with the security requirements, and for a simple, autonomous UGV can be addressed by enforcing encryption for data at rest and in transit, with decryption occurring when code, control, or content is ready to be executed. The safety model for the UGV largely consists of shielding, pointing electrodes and other connectors to the inside of the enclosures, and performing strain relief on wiring through the use of bushing, bundling, tying, and heat shrink. Sustainability is provided both through the use of low-costs materials and through the use of non-toxic materials. Using natural materials – wood, for example – where possible is a fundamental design choice for the UGV as described herein.

Methods and Manufacturing

The methods used to develop the UGV consist of a looped process of (1) design, (2) implementation, and (3) evaluation, as shown in Figure 1. This process was applied to subsystems of the UGV, then to the full UGV system, with evaluation-driven recommendations then being fed back to the subsystems.

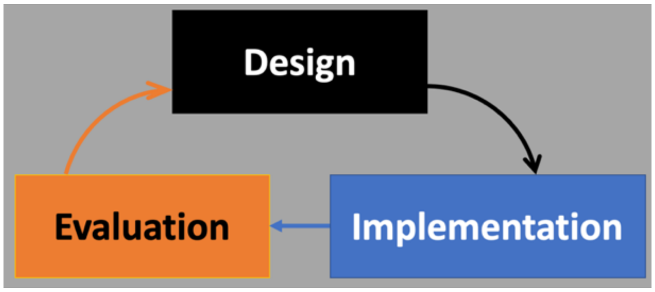


Figure 1: UGV development cycle.

Design Components and Integration

The major UGV design components comprise three system designs: (1) mechanical design, (2) electrical design, and (3) software design. Each of these systems is treated independently to support modularity but is allowed to impact the design of the other two systems. There is always a trade-off between independence of systems for modularity versus dependency of systems for tight coupling and efficiency of integration.

Mechanical Design

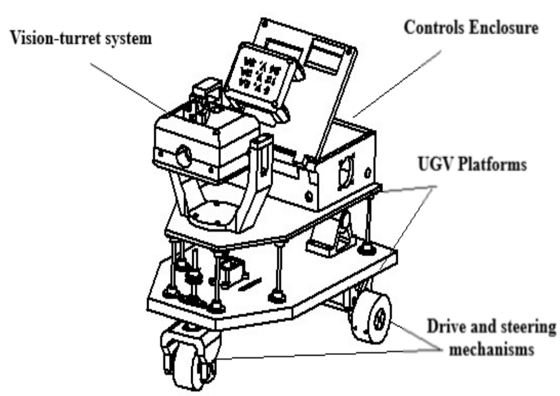


Figure 2: Major Mechanical Features of UGV.

Platform (base)

The double-floored platform of the UGV was a design choice made in order to provide hosting for a drive and steering mechanism, as well as to make available additional volume to place the vision-turret system and a controls enclosure. While a single floored platform would have been achievable with support brackets to lift the vision-turret mechanism and enough space in the center to place the controls enclosure, the second platform allows for increased modularity. This increase in modularity arises as a result of extra space on the UGV and ease of access to subsystems without significant disassembly needed. The major mechanical features of the UGV are shown in Figure 2.

Drive Mechanism

The design of the drive mechanism included a 3D-printed motor mount to hold the motor, a 3D-printed bearing mount supporting a shaft coupled to the motor, two 3D-printed drive rod offsets to level the UGV with the steering mechanism, and a pulley-to-pulley belt drive. This setup is illustrated in Figure 3.

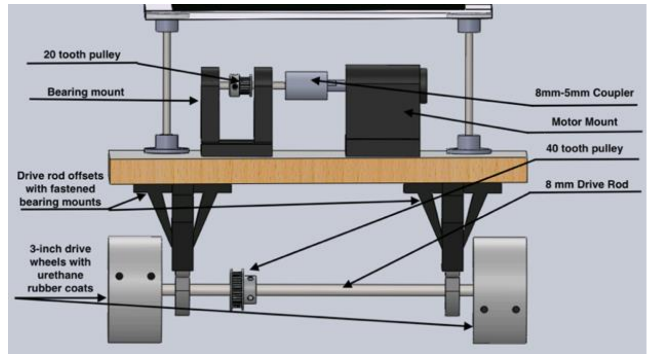


Figure 3: Major features of the UGV Drive system design.

This design for the drive system (Figure 4) allowed easy access to the motor for wiring or swapping out components, along with the ability for the UGV to be configured for single motor locomotion. Modularity for choice of the number of motors desired is ensured, since there is a single drive rod that rotates both drive wheels. This design option also means that no additional encoding is required to ensure symmetric wheel motion or provide additional torque.

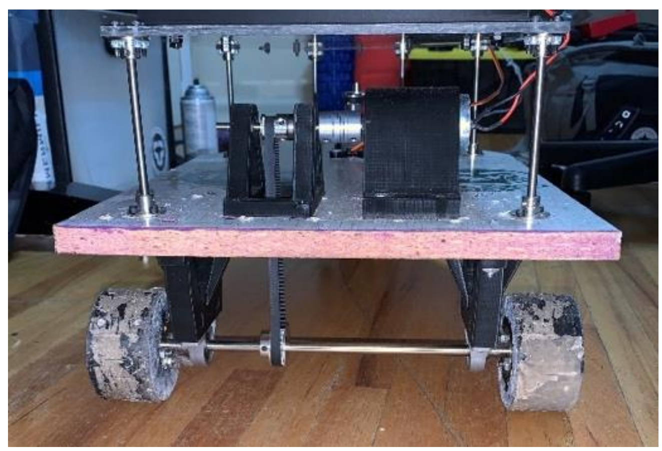


Figure 4: Physically assembled UGV drive system.

Steering Mechanism

The steering mechanism of the UGV system integrates a MG996-R 6-volt servomotor which actuates a roller wheel mechanism via a pulley-belt system, like the drive mechanism. For this system, there are three main parts that were designed and assembled: (1) a servomotor fixed in a mount, (2) two pulleys attached by a rubber belt, and (3) a steering wheel mechanism. The elements of the steering system are provided in Figure 5.

This design allows for modularity in steering, as a variety of motors can easily be mounted and connected to the steering belt. Originally a stepper motor was used to actuate the steering of the UGV; however, it was later decided that a servomotor would be a better option. This is because the signal sent to the panning servomotor of the vision-turret could simultaneously be sent to the steering servomotor during autonomous target following. While a servo is being used for the current design, it would be a quick adjustment to incorporate a different motor.

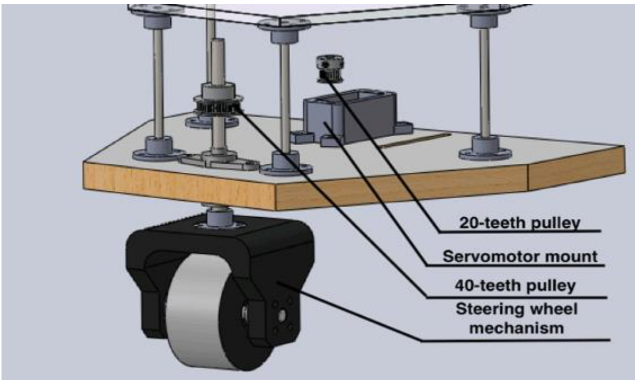


Figure 5: Major features of the UGV steering system design.

After the steering system was designed, parts of it were 3D-printed and purchased, which then were assembled together. Once everything was assembled to the UGV, wiring was completed, and the driving software was uploaded to the Arduino for testing. An assembled steering system is shown in Figure 6.

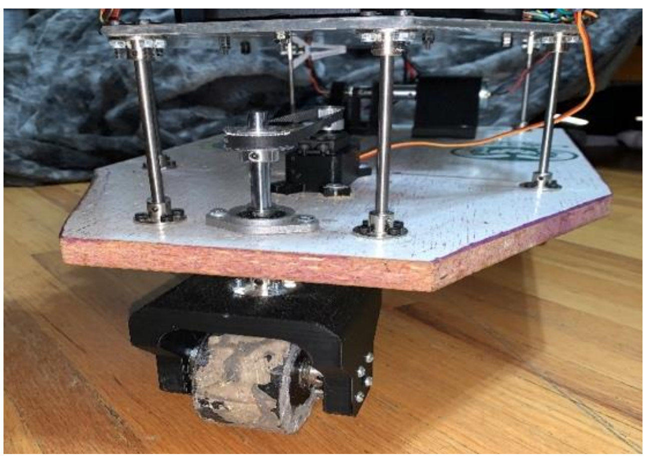


Figure 6: Physically assembled UGV steering system.

Vision-turret Mechanism

In order to extinguish a specifically-targeted fire while providing a pan-tilt system for the camera in the same design component, a vision-turret mechanism was chosen. This included a housing for a Nerf Rival's internal motors, a mount for the vision module and IR sensor to be placed, and a mount to hold a servomotor to actuate the release on a Nerf Rival ball cartridge, as seen on the left in Figure 7.

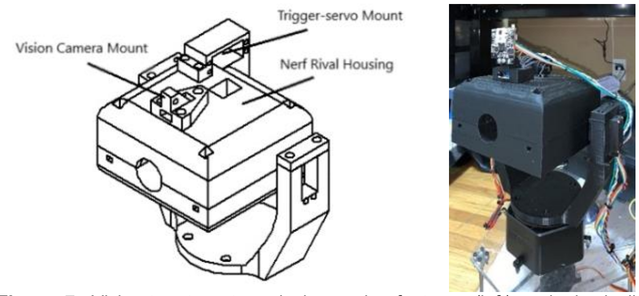


Figure 7: Vision-turret system design major features (left) and physically assembled vision-turret system (right).

The design of the vision-turret mechanism faced the constraint of allowing for tilting and panning functionality. To this purpose, two slots for MG996R servomotors were added for tilting the system and bolt holes were added to the bottom to match the bolt pattern on a rotating servo-motor base that was sourced online.

Controls Enclosure

A controls enclosure was designed (Figure 9) for the UGV system, as there needed to be a compartment to house the on-system electronics. During initial design stages, a control panel was to be sourced and purchased off the internet, but it was more cost and experience effective to simply construct it. This was a COTS (commercial-off-the-shelf) versus self-manufacture trade-off familiar to any systems engineer or program manager.

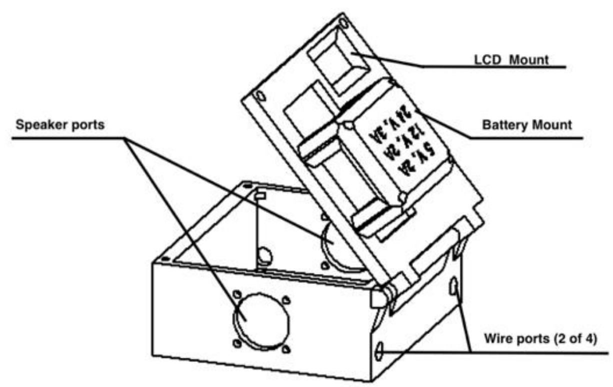


Figure 9: Isometric view of controls enclosure design.

Once the enclosure parts were designed to meet the dimensions for holding the power supply and the internal space to house the electronics, each part was 3D-printed. After printing every part, they were assembled with bolts and a 5mm rod to allow door rotation. Later, electrical components were screwed into the enclosure, wire guides were added, and all wiring was labeled (Figure 10).

User-input Controller

A wireless controller was designed for Bluetooth teleoperation of the UGV. This controller was designed to give the UGV operator a simple but ergonomic handheld enclosure for the user-input controls (see Figure 11). After designing the controller to interface with joystick modules, a HC-05 Bluetooth module, a power switch, and a compartment to house an Arduino Uno, it was then 3D-printed and assembled (Figure 12). The user-input controller was designed with a large compartment space so that other components could be added, if desired or needed later.

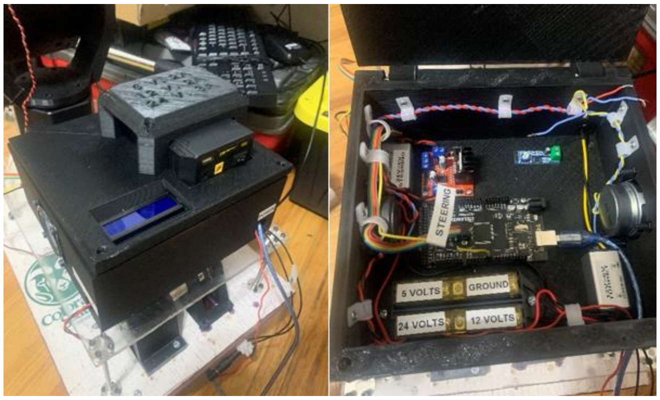


Figure 10: Physical UGV controls enclosure.

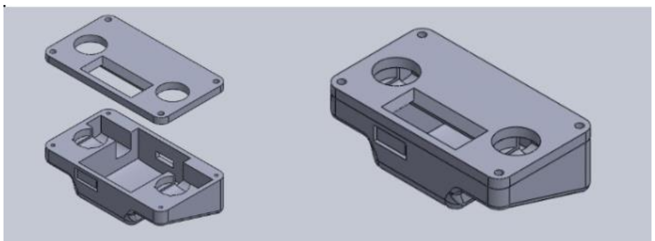


Figure 11: Isometric exploded and assembled view of user-input controller design.

Figure 12: Printed and assembled user-input controller.

Electrical Design

The electrical design of the UGV system (See Appendix) was completed on DigiKey's free SchemeIt electrical design software, and physical routing of wiring was planned on paper. Implementation of electronics and wiring happened on a subcomponent basis and later as a full system.

Power Distribution

Power to all of the UGV's electrical components was supplied from a 3 Volt output 22.4A-hr lithium-ion rechargeable battery. The battery's three different voltage supply ports are: (1) a 24-Volt, 3-Ampere port,(2) a 12-Volt, 2-Ampere port, and (3) a 5-Volt, 2-Ampere port. This battery was chosen after using two NiMH (one 6-Volt and one 12-Volt) batteries to power the system. The 24-Volt battery was chosen afterwards, as it was able to power the 24-Volt DC drive motor chosen and it was an easy alternative to using multiple batteries or voltage dividers.

Vision Camera Module

The vision camera chosen for the UGV was a PixyCam2 vision module. This choice was made for its low cost and open-source platform and suitable technical specifications (Figure 13); however, the PixyCam2 was not designed to detect fires, as it filters out IR light and has been optimized for color-tracking. Despite the limitations of the camera to track fire, it was still decided to be tested and tuned for such functionality.

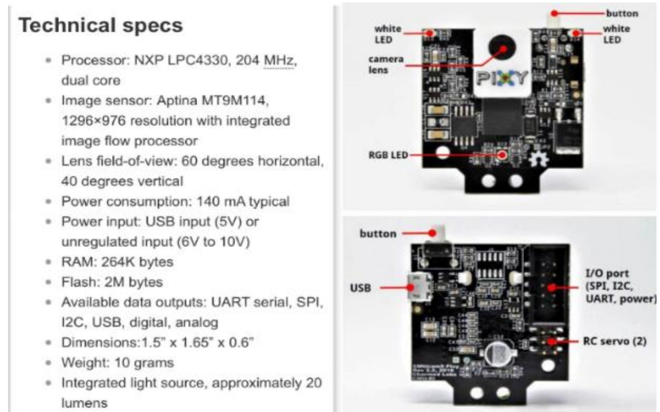


Figure 13: PixyCam2 technical specifications and peripheral identifications.

Arduino UNO (User-input Controller)

An Arduino Uno microcontroller was used to provide the logic for the user-input controller. An Arduino Uno was used for this since there was already one available. If controller size were to become a priority, then a smaller microcontroller, such as an Arduino Nano, could have been used.

Arduino MEGA (UGV)

The controller used for the UGV system was an Arduino MEGA 2560. Originally, an *Arduino Uno* was used as the system's main controller; however, as the system advanced and more peripherals were added, the Arduino MEGA was able to supply the additional I/O ports needed. The MEGA was used for Bluetooth telecommunication with the Uno for user-input control, the drive and steering motor control, tuning the gains for the panning and tilting of the turret mechanism, speaker output, and the distance control during autonomous tracking of target.

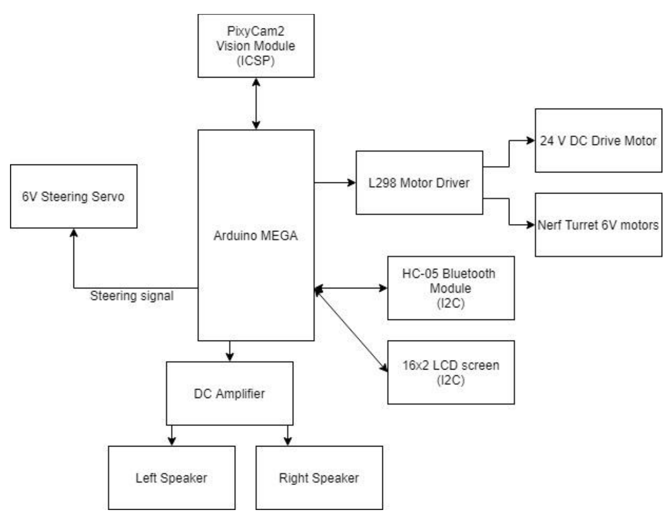


Figure 14: Arduino MEGA module interface.

The Arduino MEGA, along with all other Arduinos, do not have screw terminals, which are the most common electrical control connection method [8]. Screw terminals help constrict wires from slipping out of ports and causing a mess, not to mention possible damage to circuitry. To enable the Arduino MEGA to have screw terminals, a screw terminal breakout board was affixed to the microcontroller (Figure 15).

Figure 15: Arduino Mega screw terminal breakout board.

Software Design

Arduino C

All the software to control I/O and communication protocols on the Arduino Mega were done through Arduino's integrated development environment (IDE) software. It was through the Arduino IDE that control feedback loops were tuned and modified for the desired UGV task.

The UGV is controlled by user-input via Bluetooth telecommunication until the PixyCam2 module detects a target, which then the UGV will be in automatic control (Figures 16 and 17). The user-input control is made possible by Bluetooth communication with two HC-05 Bluetooth modules and two Arduino microcontrollers. When in user-input mode, two analog joysticks control the motion of the UGV- one joystick for driving and one joystick for steering. The outputs from the joysticks are then mapped to the drive and steering motor through linear interpolation, which was later made easier via the *map function* built into the Arduino IDE. This mapping allows for better control of the speed and turning radius of the UGV from the user.

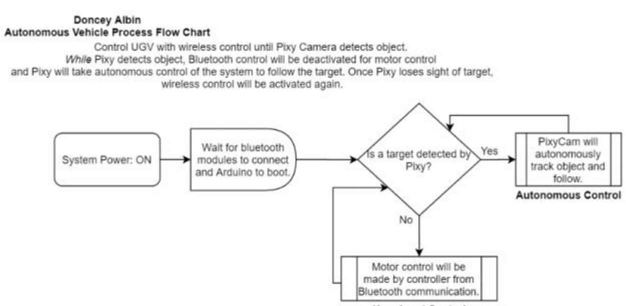


Figure 16: High-level control for Arduino MEGA.

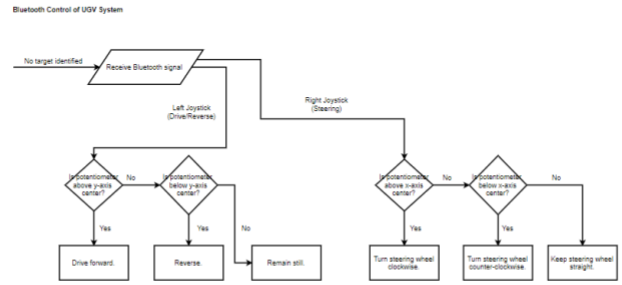


Figure 17: User-input control flow diagram.

The Arduino reads analog inputs at a range of 0-1023, but it outputs analog values from 0-255, which are used to control the voltage to the motor. Also, the inputs to the servomotor are as simple as entering a desired angle, which ranges from 0-180 degrees. For these reasons, it was necessary to scale the analog inputs from the joysticks for both motors (Equations 1 and 2).

$$\theta(t) = \theta_{min} + (Analog(t) - Analog_{min})\frac{Analog_{Range}}{\theta_{Range}} \quad (1)$$

$$MotorV(t) = MotorV_{min} + (Analog(t) - Analog_{min})\frac{Analog_{Range}}{MotorVolt_{Range}} \quad (2)$$

```
void driveControl() //Drive of the rover under autonomous control
{
  int height = pixy.ccc.blocks[0].m_height; // 90 is a good distance
  int heightDes = 90;
  int heightErr = heightDes - height; //-117 to 60

  Serial.println(heightErr);

  if(heightErr < -5){
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    driveMotor = map(heightErr, -5, -117, 100, 255);
  }
  else if(heightErr > 5){
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    driveMotor = map(heightErr, 5, 60, 100, 255);
  }
  else{
    driveMotor = 0;
  }

  analogWrite(enB, driveMotor); // Send PWM signal to motor B
}
```

Figure 18: Arduino C function with proportional control for driving UGV with respect to the PixyCam2 output of the target's height.

When the UGV is in automatic control, the PixyCam2 module tracks the target by controlling the vision-turret with its own servomotor outputs. The target's data is also gathered by the Arduino MEGA from SPI communication with the PixyCam2 module. The Arduino code uses the target's width to maintain a specific distance with proportional control (Figure 18) and the PID error feedback from the *PIDLoop.h* Arduino library from Pixy.com [9] to steer the UGV in the direction of the camera's face.

Once the UGV is at the desired distance from the target – in the UGV described herein – the IR module is used to verify that what the PixyCam2 module is targeting, is in fact a fire. If the target is verified to be a fire, the turret is used to extinguish the fire. If this is not the case, the UGV will reverse, turn off the camera, and return to user-input control.

PixyMon Software

PixyMon2 was the software utility used to tune and train the PixyCam2 module for a target (Figures 19 and 20). PixyMon2 allows an interface with the camera output, settings to tune the camera, and a command line to communicate with the camera hardware.

Figure 19: Image output from PixyMon2 software with tracking enabled. The signature box is labeled "TARGET."

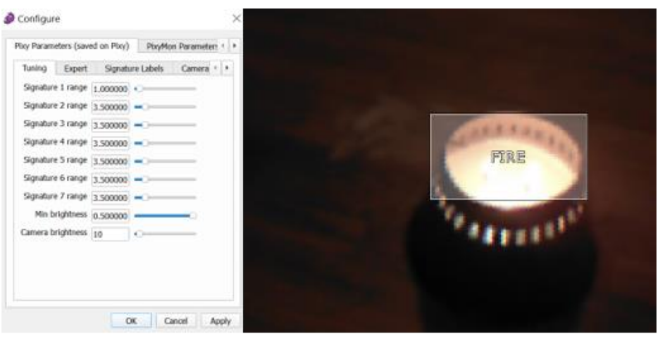


Figure 20: PixyCam2 parameters adjusted to detect fire. The signature box is labeled "FIRE."

Evaluation

The means for system evaluation included hardware testing during each major stage of building the UGV and after any changes were made to the system. The initial UGV evaluation focused on the mechanical aspects of the system (Figure 21). As physical parts were designed, 3D-printed, and later assembled with other fasteners or parts, the newly assembled system was evaluated. If there were any tolerance issues, or ideal design modifications that weren't originally seen, the part's design would be altered and reassembled. After a mechanical system was evaluated, a part of it would usually be assembled to electronics or some actuator(s), then tested with electrical interfacing (wires, drivers, microcontroller) and an Arduino C script to control the signal output from the Arduino Mega.

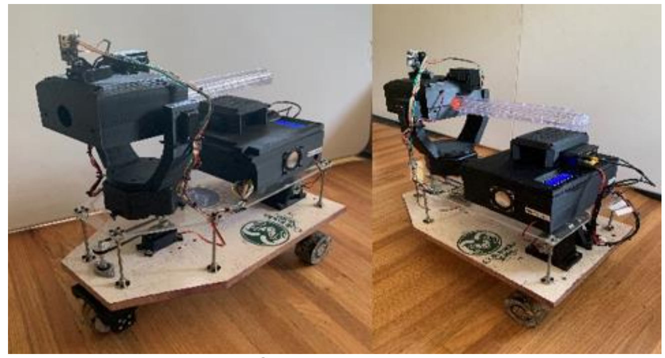


Figure 21: Fully assembled UGV.

One of the crucial functionalities of the UGV is the autonomous control of the system from the vision camera. In order to test the efficacy of the autonomous system control, three control outputs had to be tuned: (1) the PD-control for turret panning to center on a target's width, (2) the PD-control for turret tilting to center on a target's height, and (3) the P-control for the system to maintain a desired amount of pixels for a target.

The pan and tilt PD-control gains were the first control loops to be tuned, since they provide the *raison d'etre* of the UGV. To test these controls, a target was placed in the bottom left corner of the camera's view with an area of 100 pixels. Ultimately, the PD-control algorithm designed implemented error dynamics to center the object's height on the camera's y-axis (at 104 pixels) and the object's width on the camera's x-axis (at 158 pixels). The initial PD gains were set for a smaller, lighter pan-tilt mechanism sold from the Pixy website. However, since the vision-turret mechanism designed for the UGV was heavier than the one sold on the Pixy website, it initially responded with instability for the given gains (Figure 22, left).

After reducing proportional gains for both the tilting and panning PD-control, then adjusting the derivative gains, the response eventually was within a desired boundary (Figure 22, right). The rise time for the panning and tilt control was approximately 0.4 seconds and 0.28 seconds, respectively. The settling times for the pan and tilt control after tuning were 0.88 seconds and 0.8 seconds, respectively. Overshoot was not optimized for either the pan or tilt control, as it did not greatly affect the performance of the tracking.

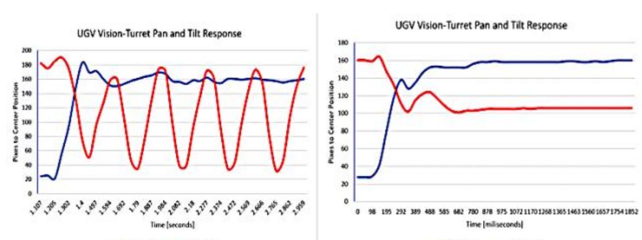


Figure 22: Vision-turret unstable (left) and stabilized (right) system response.

Figure 23: Vision-turret system in motion.

The UGV P-control of distance maintenance was the last autonomous functionality to be evaluated. This was accomplished by setting a desired pixel width for a target and then placing the target further away (Figure 23) or too close. The proportional control implemented was tuned until the system overshoot was hardly noticeable. Achieving critical response to the distance control was hard to achieve, as reducing the proportional gains greatly impacted the settling time. A PID (proportional–integral–derivative) control algorithm will be added to improve these results.

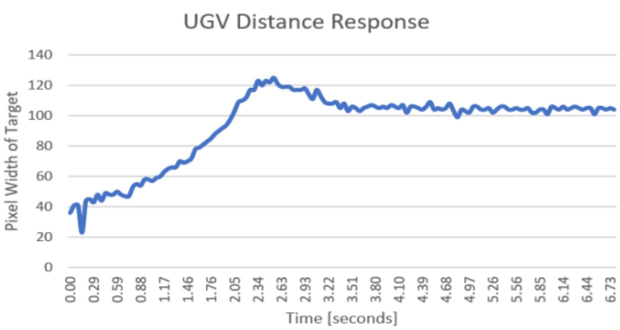


Figure 25: UGV Response to Proportional Distance Control. This example illustrates the response of the system when a target's pixel width is under the setpoint width.

Discussion and Conclusion

This paper presents an example of the development behind a UVG prototype, but there is much more investigative work to be performed for optimizing these inexpensive, easy-to-build, specific-purpose devices. Among relevant additions are:

- PID control for the distance maintenance control will improve autonomous tracking and following.
- A motor per wheel will improve power to propel the device, as well as increased steering control.
- Encryption of all data at rest and in transmission, per the security design, will also increase the amount of processing required.
- Modularity of each of the subsystems described increases the number of interfaces in the UGV, which also increases the security threat surface

As mentioned earlier, there are data, security, safety, and sustainability requirements. Data and security requirements focus on the enforcing of encryption/ Since this increases the amount of processing required both for encryption and decryption, incorporating encryption uniformly leads to a greater preference to transmit the data off of the UGV whenever possible. This requires a shared (e.g. session) key for the encryption with the server or cloud service connected to the UGV for data purposes. The safety model for the UGV was implemented, as shown by the system design diagrams: EMI/RFI (electromagnetic interference and radiofrequency interference) shielding, pointing electrodes and other connectors to the inside of the enclosures, and performing strain relief on wiring through the use of bushing, bundling, tying, and heat shrink were incorporated. Sustainability is provided both through the use of low-costs materials (the total cost of the components was $673.01) and through the use of non-toxic materials (pressboard is evident in all of the figures of the UGV).

The UGV shown also provides a platform for testing the sensitivity of the UGV's functionality to alterations in design or in the components themselves. Particularly interesting is a sensitivity analysis of the functional attributes of the UGV, such as the Response to Proportional Distance Control in Figure 25 and Vision Turret Stability in Figure 22. Parts that have higher variability are usually more likely to be counterfeit, and we need to factor that into the design. This will be an area of functional evaluation that we will perform in the future.

Acknowledgment

We gratefully acknowledge the support for this research given by the NSF EAGER grant with the abstract number 1842577, "Advanced Analytics, Intelligence and Processes for Disrupting Operations of Illicit Supply Networks". Doncey Albin was additionally supported by NSF REU 2028111, associated with the aforementioned grant.

References

[1] K. Kawabata, T. Yamada, N. Shirasaki, and H. Ishiyama, "Designing Test Methods for Running Capabilities of Ground Robots for Nuclear Disaster Response,":in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (AIM), 8-12 July 2019.

[2] S. Kajita and B. Espiau, "Legged Robots," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, eds., Springer-Verlag Berlin, pp. 361-389, 2008.

[3] Boston Dynamics, https://www.bostondynamics.com/robotics, accessed 27 August 2020.

[4] D. W. Gage, Ed., "UGV History 101: A Brief History of Unmanned Ground Vehicle (UGV) Development Efforts," *Unmanned Systems Magazine*, Summer 1995, [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a422845.pdf

[5] F. Rubio, F. Valero, and C. Llopis-Albert, Eds., "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, Apr. 2019. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1729881419839596

[6] J. R. Armstrong. (Jul. 2019). Famous Failures Revisited: A Focus on System Integration. Presented at 29th Annual INCOSE International Symposium, Orlando, Florida, USA. [Online]. Available: https://www.researchgate.net/publication/336109913_Famous_Failures_Revisited_A_Focus_on_System_Integration

[7] C. Y. Baldwin and K. B. Clark, Eds., "Modularity in the Design of Complex Engineering Systems," *in Complex Engineered Systems*, Boston, MA, USA, Springer, 2006, pp. 175-205.

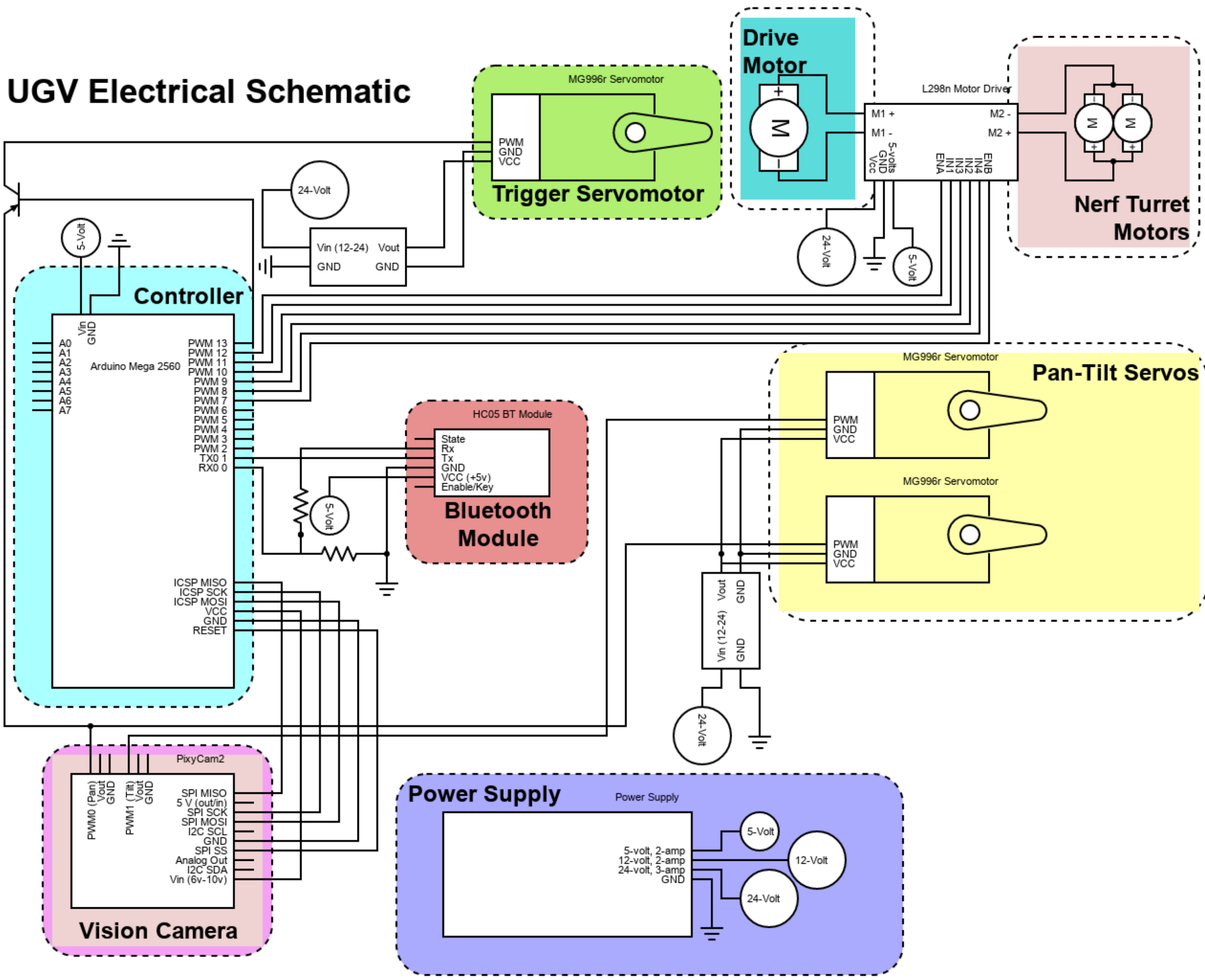
[8] "Basics of Terminal Blocks and Types," Arrow.com, Available: https://www.arrow.com/en/research-and-events/articles/connectors-basics-of-terminal-blocks-and-types

[9] Arduino Library and API, https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:arduino_api, accessed 27 August 2020.

Author Biography

Doncey Albin is a senior mechanical engineering student at Colorado State University in Fort Collins, Colorado, USA. Much of his research and academic interest has been focused on dynamics and controls with the implementation of mechatronics. Once graduated, Doncey looks to develop his education in dynamics and controls engineering at the master's or PhD level.

Appendix: Circuit Diagram (Electrical Schematic) for the UGV Described



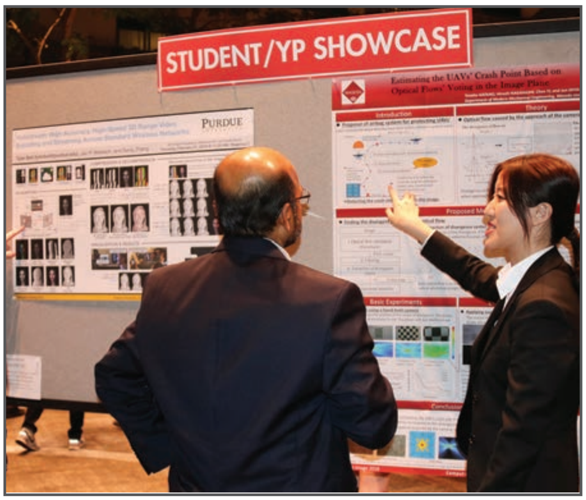
JOIN US AT THE NEXT EI!

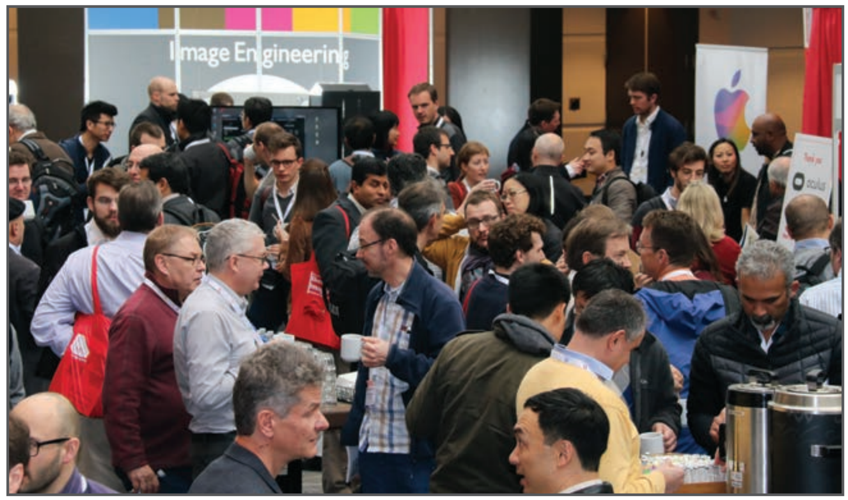
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org