The Evolution of Robots: TOAD and TAD

Sydney Worley, Bryan Williams, Makai Chambers, Gavin Clark, Zane Evangelista, Anna Jasinsky, Micah Knight, Jordan Malone, Cayden Newport, Myka Salbato, Cade Vigil

Robotics Team: Trinidad State College, 600 Prospect Street Trinidad, CO 81082

Professor Cynthia Clements

Figure 1: TOAD and TAD (right) with past TSC Robots

Abstract

Sir William Ramsay, a Nobel Prize-winning chemist once said, "No process is so perfect that there is not plenty of room for improvement. There is no finality in science. And that which today is a scientific toy may be to-morrow the essential part of an important industry." The idea that there is always room for growth and improvement in science has driven our team this year to look back on our successes and our failures. For this project, our main inspiration was improving upon projects started by our Trinidad State Robotics predecessors to combine multiple aspects of past robots to design TOAD and TAD We have taken features of 2016's S.A.B.L.E., 2022's S.W.A.R.M., and 2023's FROG to create two flip-able robot designs each capable of mapping the courses. As an additional challenge, we decided to implement a drill design inspired by the Mars rover, Curiosity. The idea behind the drill was to collect a soil sample and transport it to a given location. Unlike Curiosity, we did not run tests on the sample since we were only demonstrating a collection system design. Lastly, we decided to continue an unfinished project from the 2022 robotics team, which was to design a system allowing the robot to track its location and map obstacles on the course. Our projects this year were a challenge, but through them we have grown in our understanding and achieved what we had considered impossible.

Materials And Methodology

Over the course of this project, four robots were designed: TOAD versions 1 (V1) and 2 (V2), and TAD versions 1 (V1) and 2 (V2). Additionally, we developed a beacon tower system for 2D mapping and a drill system for soil sample collection.

For our methodology, we followed a systems engineering approach. Each system was first designed and tested, then integrated into its respective robot. There were six components in total brought together to achieve our goal of designing autonomous robots capable of successfully navigating the challenges at the Sand Dunes. The components were: wheels and motors, circuit boards, navigation sensors, mapping and tracking, and soil sample collection.

TOAD: “The Big Robot”

TOAD stands for Teachable Observant Autonomous Driller. This design was the entry into the 4.0 kg weight category with Version 1 of TOAD weighing 3.1 kg and Version 2 weighing 3.6 kg.

The first iteration of TOAD, which was titled TOAD V1 (figure 2), was used for standard navigation of the courses and testing without doing any mapping. TOAD V2 (fig. 3) was the second robot in the TOAD series and was used to map the courses along with collect soil samples. By using TOAD V1 as a testing rover, we were able to explore new ideas and prevent issues that could have occurred with TOAD V2. After conducting tests on TOAD V1, we decided that the design functioned well, therefore we used the same design on both iterations. Some important features were a flip-able design, external USB ports, and primarily 3D printed bodies for both robots.

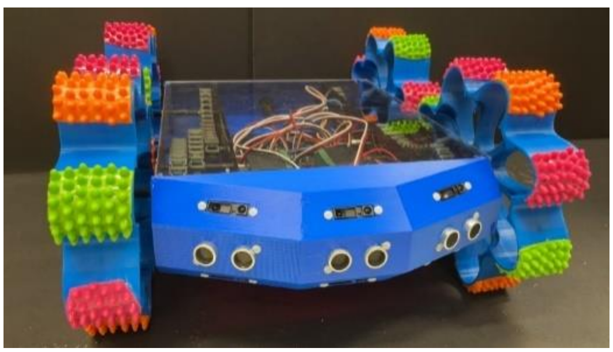

Figure 2: TOAD V1 Completed Body

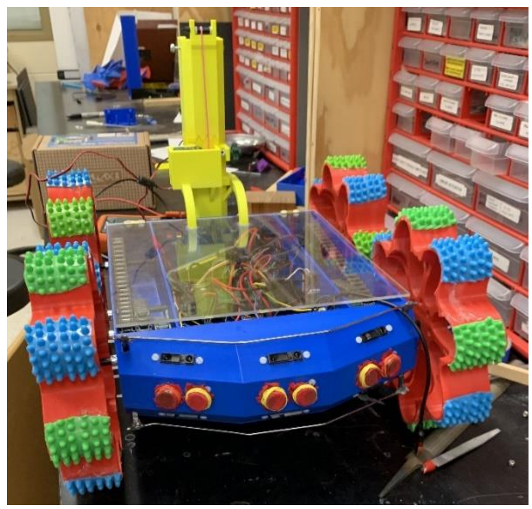

Figure 3: TOAD V2 completed with drill attachment.

TOAD Wheels

At the beginning of the design process for TOAD, we had planned on using a 3D-printed twisted wheel design from 2020’s Mothership (fig. 4). This wheel design was chosen because it had been known to prevent the buildup of sand around the wheels and had better turning capabilities. To test if these were the wheels that we wanted to use, the twisted wheels were scaled down using SolidWorks and 3D printed to fit a test robot.

After mounting the wheels to the test robot, we decided to run them in our indoor sandpit. The wheels did not have sufficient traction and there was no significant difference in the way that they turned. Therefore, we scrapped the idea and decided to update our legacy wheel design (fig. 5).

Because FROG's wheels worked well last year on a similar body, it seemed advantageous to use them again. The wheels feature a wavy design that allows for the sand to be grabbed and scooped under the wheel, propelling the rover forward. The wheels were 3D printed using low-density ABS plastic. For extra traction, textured sensory balls were glued on using cyanoacrylate glue.

After more sand testing with the wheels mounted on TOAD, they worked just as expected. There was a bit of sand pileup atop the robot, which was what we were trying to avoid by using the twisted wheel, but because of the wheel's ability to traverse fine sand, this wheel was the team's choice.

Figure 4: 2020 Twisted Wheel Design

Figure 5: TOAD Official Wheels

TOAD Body

Body Design Drawing for TOAD V1:

Our plan was to 3D print the entire body of TOAD, something we had never done with a large robot before. FROG's design from last year functioned well, but many aspects of it needed alterations. The body needed to be skinnier, have an enclosed area for the ultrasonic and infrared sensors, have built-in angles for the sensors, and have an external USB port for easier programming. On top of that, the robot itself needed a cleaner and higher quality build. The overall measurements of the body were to have a width of 270 mm, a length of 398 mm, and a height of 78 mm. The first drawing (fig.6) did not include any of the inside mechanisms such as gearing or motors.

In the second drawing (fig. 7), we added the interior mechanisms. The width and length remained the same. The height of the walls was increased by 2 mm, totaling 80 mm, providing space for larger gears. The wall thickness was 3 mm, which provided enough strength without being too heavy.

TOAD features a 6:1 gear ratio between the motors and the wheels, which provided enough power to navigate the courses. FROG's motors are in the back of the rover to counterbalance a crane on the front. Because, unlike last year, we do not have a crane, the motors were moved to the front of the robot to counterbalance the drilling mechanism located at the back. There are also supports within the body to help mount the drilling mechanism.

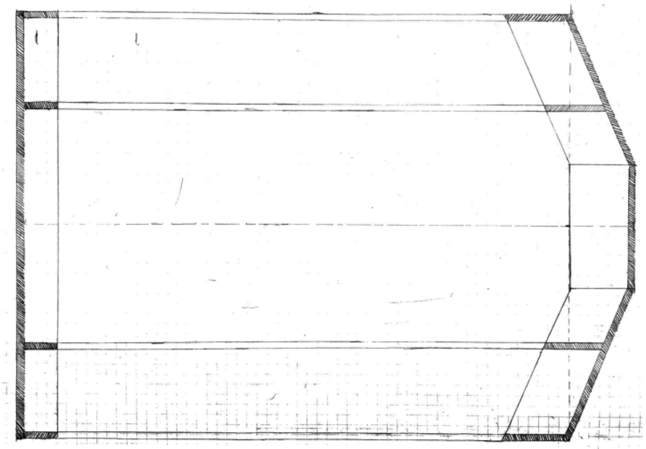

Figure 6: TOAD V1 Drawing

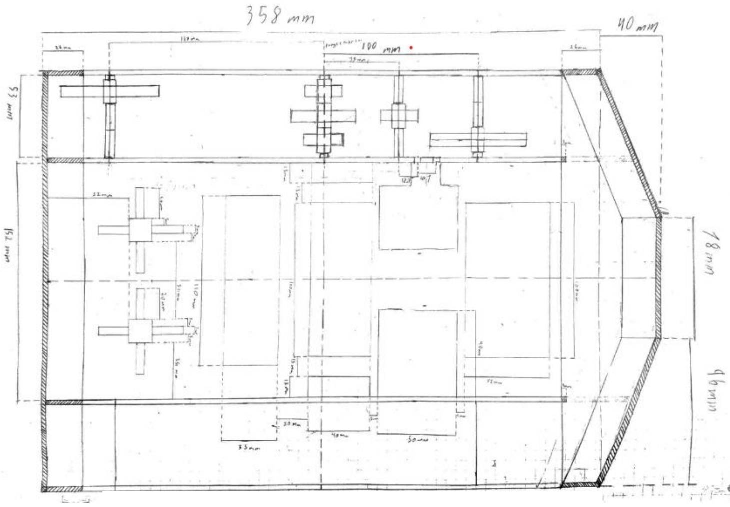

Figure 7: TOAD V2 Drawing

3D Design and Printing of TOAD V1:

TOAD's final body was to be 3D printed almost entirely out of ABS material. Before printing this, the first design included something we titled the "exoskeleton" (fig. 8). Last year, the acrylic panels used to form the robot body were extremely difficult to assemble, so to make the process easier, we designed a 3D printed frame that acrylic panels could be inserted into. The exoskeleton reduced the cost of construction and allowed us to quickly fabricate a prototype that was structurally sound and fully functional. The exoskeleton was comprised of four pieces that snapped together to form the frame. To get the pieces to connect, ship-lapped box joints like puzzle pieces and dadoes were designed into the ends of the four different pieces (fig. 9).

Figure 8: TOAD V1 Exoskeleton

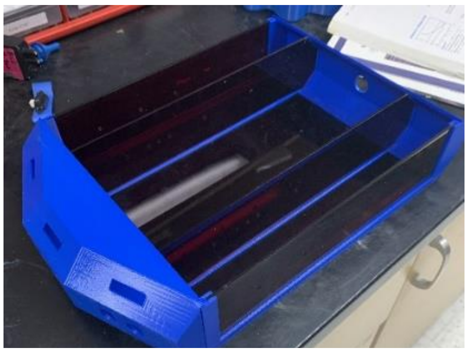

Figure 9: TOAD exoskeleton with acrylic panels.

3D Design and Printing of TOAD V2:

The body of TOAD V2 was made entirely out of 3D print material. Since the printer bed was not large enough to print the body in one piece, it was printed in multiple pieces and designed with an interlocking system of tabs connecting them. Holes were added for the bump sensors, as well as a set of columns to attach the drill at the back (fig. 10). A beacon array (fig. 11) was printed into the bottom of the rover, but later during testing we discovered this was unusable as its design blocked the ultrasonics from properly transmitting.

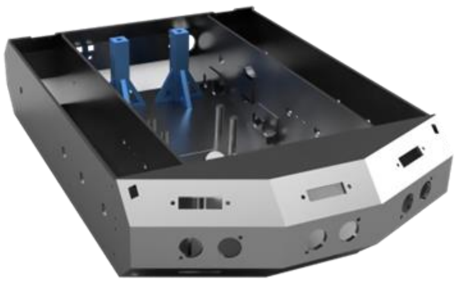

Figure 10: TOAD V2 Rendering with Drill Columns in Blue

Figure 11: TOAD V2 Unused Beacon Array on Undercarriage

TOAD V2 Changes:

After testing TOAD V1, it was time to start designing the second version. In this version, the wheels remained the same, but their position moved slightly forward to improve flipping capabilities. The gear chain was adjusted to accommodate the new placement of the wheels. Furthermore, columns for the drill mechanism to anchor into and an ultrasonic sensor array for the mapping system were added. Lastly, four holes were installed at the front of the rover above and below the sensor arrays to accommodate a bump sensor. Everything else on the rover remained the same.

TOAD (V1 and V2) Circuit Boards

The circuit board for TOAD V1 is a double-sided, plated through-hole board which implements a Propeller Flip Controller and an AD (analog to digital) converter. TOAD uses 6 infrared sensors, 3 ultrasonic (PING))) ™) sensors, 1 gyroscope, 1 radio transceiver, 2 Vex Motors, and 2 Vex Motor Controllers. TOAD is powered using a Thunder Power RC 55c 3 cell 11.1V lithium polymer battery.

TOAD's board design was made in DipTrace, an electronic design software. The circuit board (fig. 12 & 13) features a split design, meaning that the one board is split into two halves that are wired together to fit on either side of the motors. The only major electrical difference between TOAD V1 and V2 is that TOAD V2 has an extra component, the radio transceiver, allowing it send and receive information from the 2D mapping system. TOAD V1 is only meant for completing standard navigation and therefore does not need these components. TOAD V1 did experience brown-outs during sand testing, and the issue was found to be wires that were not thick enough to handle the current being passed through them. The wires were changed from 24 gauge to 16 gauge, and as a result, the brown-outs stopped.

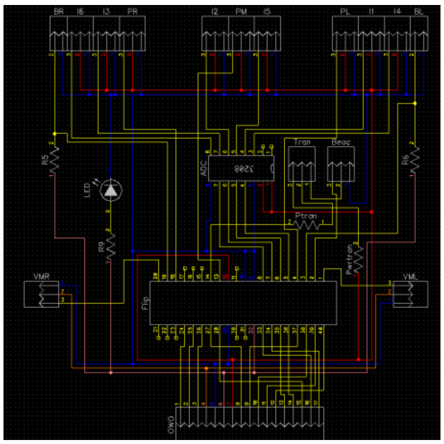

Figure 12: TOAD Circuit Board - Piece 1

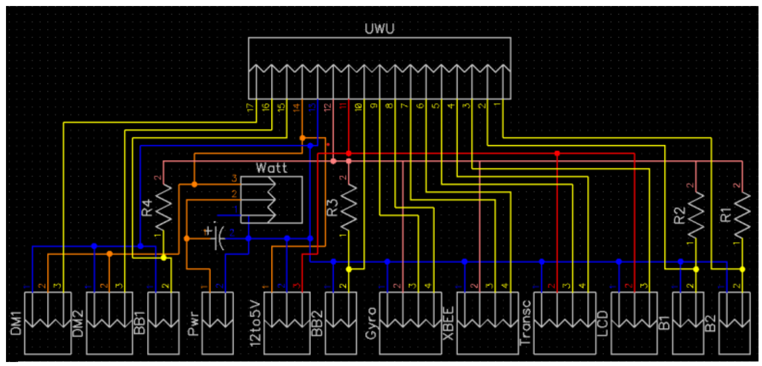

Figure 13. TOAD Circuit Board - Piece 2

TOAD V1 Testing:

Out of all the robots, TOAD V1 had the least number of problems and performed outstandingly from the start. The rover reached all of the goals we had set for it early in the year, including being able to flip and continue to navigate. TOAD completed the courses we set up in our sand pit, which included going over large rocks and avoiding pits. We did realize that we needed a bump sensor, which was added T.O.A.D V2.

The Drills

As an added feature to our large robot TOAD V2, a drilling mechanism was designed, built, and tested (fig. 14). The design is split into two parts: the drill and the drill mechanism. The drilling mechanism uses two motors, a pulley system, a shaft to hold the drill, and a switch which signals the program that the drill is fully retracted.

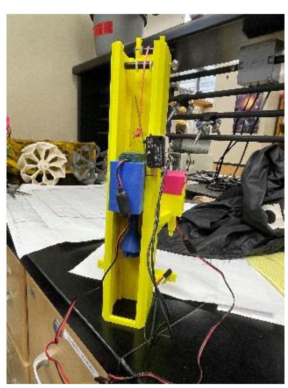

Figure 14: The Drilling Mechanism including: the shaft, the lifting mechanism, and the inner mechanism.

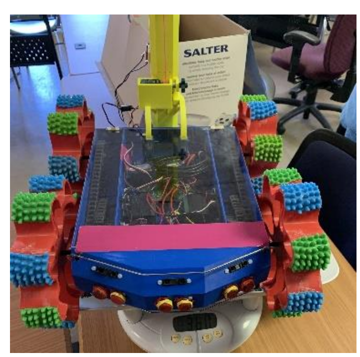

Figure 15: TOAD V2 with drill and posts

Drill #1: Little Timmy

Little Timmy (fig. 16) was the first drill iteration. The drill was 3D printed using ABS material with a diameter of 20 mm and a length of 71 mm. The initial idea for sand collection was to print the drill with a hole that would collect a sample. The hole, located slightly above where the threads of the tip began, measured 5 mm by 5 mm.

Figure 16: Little Timmy - First Drill Iteration

To test "Little Timmy", the drill was attached to a power tool and was taken to our indoor sand pit. Little Timmy failed because the threads were reversed meaning the drill had to be reversed to penetrate the sand. The sand collection hole was not large enough and was slanted in the wrong direction. The shaft was also not durable enough and there was no means for the sand to be removed.

Drill #2: Wide Hexagon:

Drill no. 2, which we named Wide Hexagon (fig. 17) for its hexagonal shape, featured walls that were 6mm thick to increase structural support. The sand collection hole was doubled to 10mm by 10mm, making it easier to collect sand. The testing went well for wide hexagon, but the drill was too large and too heavy. The collection hole also needed to be moved to the base of the threads.

Figure 17: Wide Hexagon - second drill iteration

Drill #3: Fine Lad:

Drill no. 3 was titled Fine Lad and was a redesign of the first drill, Little Timmy. But halfway through the design process of Fine Lad, it was decided to switch to an auger design.

Drill #4: Sandman:

Sandman (fig. 18) was the fourth drill design and first auger design. The idea behind the auger was that the tip of the drill, encased by threads, would push sand upward into an internal cavity. Threads on the outside of the drill were supposed to help the drill penetrate the ground. The casing came down past the start of the auger so that it would keep sand from escaping and push it upward. The issue that occurred with the casing being longer was that it pushed the sand away from the drill instead of collecting it. There was also no method of removing sand from the drill, so this design was scrapped.

Figure 18: Sandman - third drill iteration; first auger design.

Drill #5: Fives:

Fives (fig. 19) was the name of the second and last auger design. This drill was designed so that the threads extend out from the tip of the drill. During testing, it was discovered that with the auger designs, the sand is always going to be pushed down or to the side due to the light weight of the sand. This defeated the purpose of the sand collection drill, and all auger designs were scrapped.

Figure 19: Fives - fifth drill iteration - second auger design.

Drill #6: Fine Lass:

Fine Lass (fig. 20) was made so that instead of having open collection holes in the side of the drill, two holes were printed into the drill with flanges covering them. The flanges pushed the sand away from the collection hole as it burrowed into the ground, but when the direction of the drill was reversed, the flanges acted as shovels and scooped the sand into the drill collection cavity. To remove the sand from the drill collection cavity, a two-way interlocking system (fig. 21) was used at the top of the drill. This locking mechanism works like a key, where the key (the top of the mechanism that connects to the actual drill mechanism) gets placed into the keyhole (the drill sand collection cavity), and is then twisted to lock both into place. The main issue with Fine Lass was that it did not have the proper thread pitch and could not burrow itself into the sand.

Figure 20: Fine Lass- sixth drill iteration; first drill with flanges

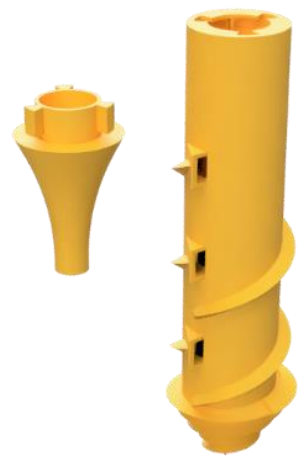

Figure 21: Fine Lass Drill Rendering with two-way interlocking system

Drill #7: Bobert:

Bobert (fig. 22) improved upon previous designs by adding extra threads behind each flange allowing itself to have more traction as it penetrated the ground. A third flange was added to the side of the drill to collect a larger sample size. Flanges were printed on the inside of the sand collection hole as well to prevent sand from escaping out of the bottom collection hole when the drill is lifted. The two-way locking mechanism was installed at the top of this iteration as well. This drill design worked so well that this was the final design of the drill body. The two-way locking system was also used in this design.

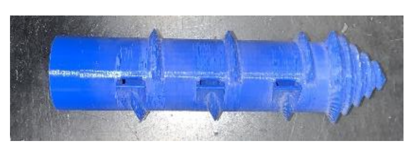

Figure 22: Bobert - seventh drill iteration - more threading

Drill #8: Moe:

Moe (fig. 23) is the final drill and is almost an exact replica of Bobert except for the spacing of the threads. The threads were spaced further apart and were given a slightly more aggressive pitch. The new thread pitch helped the drill dig into the sand quicker and more efficiently than the iterations that came before. The two-way locking system was also used in this drill system to lock the drill to the drilling mechanism as well as close the sand collection cavity.

Figure 23 Moe - eighth and final drill iteration; used in drilling mechanism.

The Drilling Mechanism:

The drilling mechanism (fig. 24) has three primary components: the inner sliding mechanism, the shaft, and two motors. One motor is to spin the drill and the other motor is attached to a pulley to lift and lower the drill.

The Inner Mechanism, Nick:

This system was built using a 3D printed motor casing, a Vex 269 Motor, and an axel rod connected to the two-way interlocking mechanism at the top of the drill. The encased motor is placed inside of the shaft and then connected to a pulley system with an independent motor.

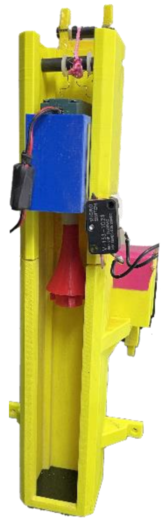

Fig 24: Drilling Mechanism- Inner Mechanism Nick (blue)

The Lifting Mechanism, Uppies:

Uppies is a motor that controls the tension of the string that lifts and lowers the drill and inner mechanism. As the motor spins the axle counterclockwise, the inner mechanism is given slack from the string, allowing the drill to burrow into the ground. To collect the soil sample, the direction of the motor is reversed. After 4.5 seconds, the pulley motor retracts the drill into the shaft. Uppies is secured to the back of the shaft that houses the drill.

The Shaft, Bob:

The 3-sided shaft houses the drill motor, the pulley system, and pulley motor on the backside of TOAD V2. The drill retracts into the shaft as the inner mechanism is lifted and extends out through a hole in the bottom. A limit switch is attached to the outside of the drill to signal TOAD when the drill is fully retracted.

The Drilling Procedure:

The drilling procedure starts with TOAD driving to a specified location autonomously. The inner mechanism is then lowered towards the ground in increments as the drill burrows itself into the ground. Once the drill is fully extended into the ground, it starts rotating in the opposite direction, allowing it to collect a soil sample. Once the drill has collected the sample it continues to spin while the pulley system lifts it out of the ground. Once the limit switch signals full retraction, the process is complete.

TAD: The "Little Robot"

TAD (Fig. 25) also known as the little robot, stands for Tiny Aggravating Device, which is the most appropriate name we could have given this small robot. It was our entry for the 1.5 kg weight category and was inspired by last year's robot F.L.Y.; the biggest difference is that we wanted TAD to be flip-able. As a result, TAD's body went through a complete redesign from its predecessor FLY, looking nothing alike in the end. Initially, the main goal for TAD was to map the courses at the Sand Dunes challenge using the 2D mapping system. TAD would also do standard navigation. Unfortunately, because this small robot had been so troublesome, the duty of 2D mapping was relegated to TOAD.

Figure 25: TAD V1 with TPU wheels.

The T.A.D Bodies:

Initially, the team made a cardboard mockup of TAD. This prototype helped us place the sensors and test our original wheels. This mockup, which we titled "Flammable TAD" (fig 26) worked well in preliminary testing.

TAD's official body was designed in Fusion360 and was 3D printed on the Stratasys F370 in yellow ABS material. Holes were incorporated into the body to accommodate the PING))) ™ sensors, and the IR sensors in the front of the robot. It was manufactured in one piece with a separately printed detachable lid.

Figure 26: Cardboard mockup of TAD body.

Figure 27: TAD V2 – elongated body used at Sand Dunes Challenge

Once TAD was printed, populated, and programmed, sand testing took place. The first issues that we noticed were that the wheels did not work well and that the robot did not have enough clearance to get over rocks. To solve these problems, we went through many redesigns and incorporated different motors and larger wheels (see sections "*TAD Wheels*" and "*More TAD Issues*"). The new larger diameter wheels were bumping into each other, and the front sensors were "seeing" the wheels. To resolve this issue, a new longer body was designed, accommodating the new wheels.

TAD Wheels:

The wheels for TAD have gone through much redesign because of the many problems encountered in testing. Since the beginning of the project, we have known that we wanted to use 3D printed flexible wheels. These flexible wheels printed out of TPU material allow TAD to maneuver over large rocks, have better traction, and avoid piling up sand. Initially these wheels worked well with a cardboard mockup of the small robot but had yet to be put in the sand. When TAD was assembled and sand tested, the robot did not have enough clearance, could not climb over large rocks, and did not have enough traction. To fix this problem, two different kinds of mockup wheels were created: "wood spoon and duct tape TAD" (fig 28) and "coffee can TAD" (fig. 29). These prototypes were created to quickly and easily determine the performance impacts of increasing wheel width and diameter. Initially, concerns were raised that larger wheels may decrease motor power too much, but these wheel tests showed that the increased diameter and width added more traction to the wheels, allowing the rover to maneuver over rocks and swiftly traverse the sand. TAD's final wheel design was named the "Beluga Wheel" (fig. 30) for its large size.

Figure 28: Wooden Spoon and duct tape TAD

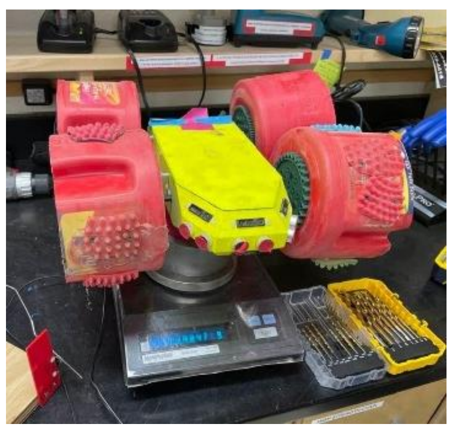

Figure 29: Coffee can TAD

Figure 30: TAD V1 with updated "Beluga" wheels.

Assembling TAD:

TAD V2 (fig. 27) uses 4 Vex 269 motors, a custom circuit board, 1 fuse, 1 switch, 6 IRs, 3 PING))) ™ sensors, 1 Gyroscope, 2Vex Motor controllers, 4 TPU wheels, and 1 lithium polymer battery.

More TAD Issues:

Figure 31: TAD V2 Front View

TAD has been problematic this year. Some other noteworthy issues included dying motors, dying sensors, and improper wheel size. The first problem that came to our attention were the motors. We could not keep TAD running consistently because the gear boxes of the original motors kept breaking teeth. This was fixed by switching to more powerful and better made Vex Motors. Later, we encountered the problem of TAD being switched on but not running. Our electronics team's initial theory was that the wires were too thin, but later we found that the Vex Motors had thermoregulators that would temporarily cut power to the motors. We fixed this by bypassing the thermoregulators.

Furthermore, 13 PINGS))) ™ sensors and 5 IRs were fried because the regulator was installed backwards, sending 12 volts into all the electronics frying TAD twice. The sensors were replaced, and TAD was able to resume in-sand testing.

Programming

Our main goal for programming this year was our mapping system. For the mapping system to work, standard navigation had to be completed. Standard navigation's main goal was to get the robots to their destination by using the information obtained from the sensors in the robot. The programming language used for the robots was Propeller C. Propeller C is a multicore language. It allows the robot to multitask with 8 cores (or cogs). Each cog independently handles a part of the program. The code for each robot allows us to manually select which sensors we use in each course. Both programs contain a navigation cog, a sensor cog, and an ultrasonic sensor cog. TOAD's program also contains a drill program that initiates the drilling procedure. The thresholds and motor speeds also vary depending on the robot. Other cogs include the motor controller cog and the timer cog. The motor controller cog controls the motors, and the timer cog is a universal timer cog that counts how long the program has been progressing in milliseconds.

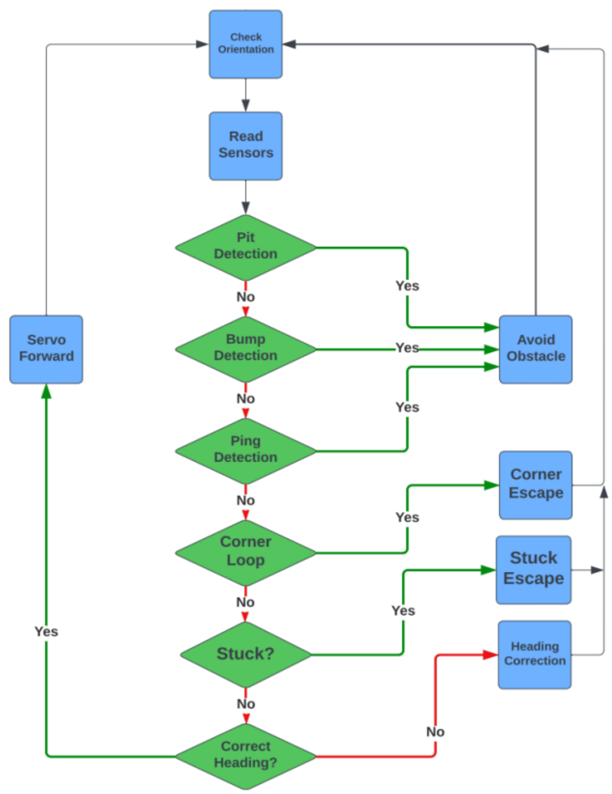

Figure 32: Flow Chart of the Standard Navigation Program

We have two important functions in our programs: the corner function and the stuck function. The corner function is a way to prevent the robot from becoming trapped in a corner. The function works by counting the time the motors are not going forward. If the left and right motors alternate for too long, the corner function chooses a new path based on the information given from the sensors. Additionally, a stuck function was included to assist the robot when it would get high centered on a rock or stuck in a pit. The stuck function initiates every few seconds to make sure it is not trapped. The function determines if the robot is stuck using the gyroscope. First, the robot attempts to move out of the current heading by engaging the motors and then checks the compass to see if the robot's heading has changed. If the robot's heading has changed, it indicates that the robot is not stuck. If the robot's heading has change, it indicates that the robot is stuck. If the robot is stuck, it will initiate a series of motor actions until it is unstuck. If none of the sensors are activated, the robot moves forward until it reaches its destination.

TOAD V2 incorporates the drill program. First, the robot collects a drill heading. This drill heading enables the robot to come back to the starting point after collecting a sample. Subsequently, the robot continues its typical functions: avoiding obstacles and going in the direction set by the compass. This runs for approximately thirty seconds, counted by the timer cog. Once thirty seconds ends, the drill program begins. The robot stops, and then the drill starts spinning. As it spins, it slowly lowers in increments to gain traction and not strain the motors. Once the drill fully lowers, the drill reverses direction to collect a soil sample. While spinning, the drill raises up until the limit switch is triggered, which stops the drill. After the sample is collected, the robot changes its target heading to the drill heading and navigates back to its starting point.

To summarize, in the standard navigation program, the robot checks for obstacles as it is going to its destination right-side up or flipped. Using the 8-core propeller chip, the robot is constantly checking all sensors to assist in this navigation. Using multiple cogs, the robot can successfully navigate through courses.

The Mapping System

The goal of the mapping system was to triangulate the position of a robot as it navigates a course and to map obstacles along the way. This would give the robot a memory and a sense of spatial awareness that could be used for improved course navigation. This project was inspired by the Trinidad State Robotics Team of 2022, which had created a map that was not as robust as they wanted. This year we used a system of three ultrasonic receiver towers to calculate the distance to a robot with ultrasonic transmitters mounted on top.

Preliminary Ultrasonic Testing:

To develop a mapping system, we first had to be able to find the distance between the robot and three separate towers. We considered a variety of ways to achieve this, including using light waves or sound waves from a conventional speaker. Testing found that the most practical method was to use a modified version of the PINGS))) ™. These are ultrasonic sensors that use the speed of sound to measure the distance between themselves and objects by emitting ultrasonic soundwaves and counting the time until the echoes reflect back. To serve our purposes, we removed the transmitters from one set of ultrasonic sensors and removed the receivers from another. To test the effectiveness of this setup, one ultrasonic transmitter was placed several meters away from an ultrasonic receiver. Both ultrasonics were wired to the same Propeller Board (fig 33). The Propeller signaled the transmitter to emit an ultrasonic wave, then it began counting. Once the wave reached the receiver, the propeller would use the time it took to calculate the distance between the two. This test showed that using ultrasonics as a method of measuring distance was accurate to within 1 centimeter when measured against a tape measure and functioned up to 7 meters away, a satisfying result solidifying ultrasonic sensors as our final choice.

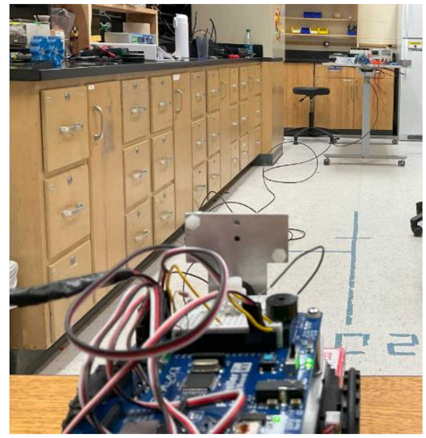

Figure 33: The Wired Test of the Ultrasonics

Means of Data Transmission:

Since the robot and tower cannot be wired together while moving, we had to find a means of wirelessly signaling to the tower that the robot had emitted a sound wave and to begin counting, similar to using the difference in time between lighting and thunder to calculate the distance to a strike. The 2022 Robotics Team had used XBee digital transceivers (fig. 34) to accomplish this, but their erratic buffer time made the distance measurements very inaccurate. This year we used an analog radio transceiver (fig. 34). It was more difficult to program, but it tremendously increased our accuracy by eliminating lag time in the radio signal. Our next test was using two Propeller Boards, instead of just one, and a transceiver on each instead of using a cable. This test, using a radio, proved to be just as accurate as using a cable. Additionally, these same radio transceivers were able to transmit that data back to the robot.

Figure 34: Radio Transceiver (Left), XBee Digital Transceiver (Middle), Propeller Activity Board (Right)

Developing The Receiver Towers:

After a means of accurately measuring the distance between two separate devices had been successful, this technology had to be made into something usable: the receiver towers (fig. 36). Through testing, we found that the lowest distance reading of an ultrasonic over any given arc is always the most accurate measurement in that arc (fig. 35). This meant that to find the distance to an object, the towers' ultrasonic receivers were placed atop servos that would continually scan the field. Using three separate receiver towers, this distance information could be used to triangulate the robot's position.

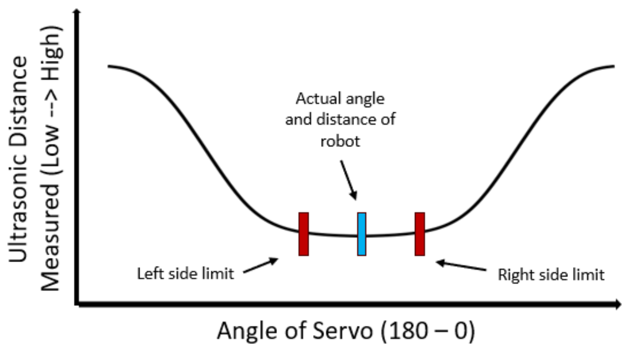

Figure 35: Graph of servo scanning an arc

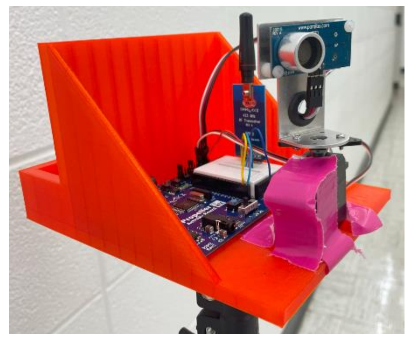

Figure 36: A Receiver Tower

Programming Triangulation:

As the robot navigates the courses, it repeatedly transmits strings of fives. This informs the towers that a soundwave is on the way and to calculate the distance when it arrives. Once the robot needs to know its position, it transmits the numbers 6, 7, and 8. Each number corresponds to the designation of a tower. Since the transceivers all operate on the same frequency, each tower transmits back its distance from the robot as its respective number is called. The program then verifies these results using a checksum, and if they are valid it then calculates the intersection point of three circles (fig. 37) to find the x and y coordinates of the robot:

$$x = \frac{C_1B_2 - C_2B_1}{A_1B_2 - A_2B_1} \qquad y = \frac{A_2C_1 - A_1C_2}{A_2B_1 - A_1B_2}$$

$$A_1 = -2H_1 + 2H_2 \quad B_1 = -2K_1 + 2K_2 \quad C_1 = R1^2 - R2^2 - H1^2 + H2^2 - K1^2 + K2^2$$

$$A_2 = -2H_1 + 2H_3 \quad B_2 = -2K_1 + 2K_3 \quad C_2 = R1^2 - R3^2 - H1^2 + H3^2 - K1^2 + K3^2$$

$H_1 = 0$	$H_2 =$ Distance of X axis from tower A → B
$H_3 = 0$	$K_3 =$ Distance of Y axis from tower A → C
$K_1 = 0$	$R_1 =$ Distance of robot from tower A
$K_2 = 0$	$R_2 =$ Distance of robot from tower B
	$R_3 =$ Distance of robot from tower C

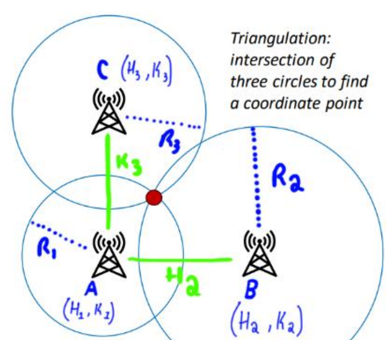

Figure 37

This triangulation was tested without using a real robot, instead just the transmitter array connected to a Propeller Board. Over the course of several weeks, the programming of the three towers and the programming of the transmitter array were fine-tuned such that finding the position of the array became accurate and consistent.

Programming Navigation to Points:

The next logical step in the design process was to take the data from the triangulation program and actually do something with it. This meant being able to, quite literally, move the robot from point A to point B. To accomplish this, the ultrasonic transmitter array was mounted atop a Parallax Activity Bot, a robot that works well indoors. After testing on the Activity Bot, we switched to FLY (fig. 38).

To navigate to a given point, the robot first needs to align its compass to the North of the towers, meaning the compass reads 0 when aligned parallel to the map's Y-axis. This only needs to be done once when the towers are initially set up. The robot then uses its current position and heading to calculate the angle between its current position and target position. Next the robot orients itself in that direction and begins to drive forward. The robot stops occasionally, repeating this process and slowing down as it nears the point. Once the robot is within a set radius of the specified point, it moves on to the next point until it completes its journey.

Creating a Map:

Since the 2022 Trinidad State Robotics Team had done the work of creating a program that would display a series of points on a computer, it was decided that we could reuse this to reduce our workload. To create this series of points, as the robot navigates it stores its position in an array. Displaying this data allows us to see the path the robot followed as it traversed the course (fig. 39).

Figure 38: Navigating to Target Points Using FLY

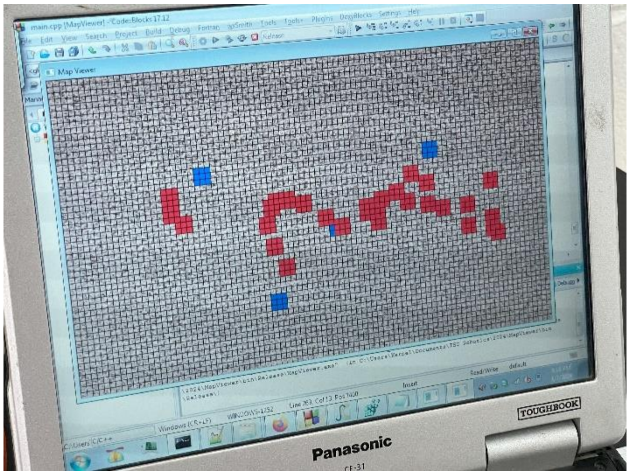

Figure 39: A Map Generated After Navigating A course. Traced Path in Red, Target Points in Blue.

Mapping Obstacles:

The last and final stage of programming the mapping system was to mark and avoid obstacles on the course. To do this, FLY utilized its PING)))™ sensors to calculate the obstacle's distance from the robot and find the obstacle's coordinate position. This was then stored in an array to be displayed later on. This information could also have been used to navigate around known obstacles before the sensors ever saw them by following a precalculated path. At this point, however, the project was running out of time and needed to focus on transferring this system from indoors into an actual sand pit. As of now, all the programming needed to accomplish obstacle avoidance and mapping exists but needs more testing and fine tuning to work consistently.

In-Sand Testing and Final Results:

Getting the mapping system into the sand and mounted to the robot TOAD V2 (fig. 40) was a difficult process. Each element of the mapping program had to be adapted, tested, and fine tuned to work on its new robot. Once this had been achieved, the robot could very consistently navigate to a series of points in the sand. One problem that we did encounter, however, was a lack of power being supplied to the transmitters. This was due to the fact that five ultrasonic transmitters were being powered by a single ultrasonic connection to save space on our robot's main PCB. This oftentimes made it difficult for the towers to receive a good signal from the robot, and thus the robot would spin in a circle until the signal was reacquired.

Figure 40: Mapping Outside Using TOAD

Results & Conclusion

Despite the challenges that we have faced this year, we are proud to say that all six aspects of the projects were completed and are functional. All four robots have met most of the goals that the team set for them. Improvements still need to be made to the 2D-mapping system to get it to the point that we would like, yet tremendous progress has been made on improving the mapping system created in 2022. As the project deadline neared, we had thought that we would not have a functional small robot at the dunes, but TAD exceeded expectations in the last weeks and was, in the end, our best performing small robot thus far. TOAD still amazes us with its drilling, mapping, and navigating capabilities. At the dunes, both robots performed above our expectations, having no major component failures and completing all six courses in under half an hour. We are proud to say our goal of building upon past Trinidad State designs has been achieved and we could not be happier. The team hopes that our project will inspire future teams to continue researching these projects, much like our predecessors' work inspired us.

Acknowledgements

The 2024 Trinidad State College Robotics Team would like to thank the Colorado Space Grant Consortium for their generosity, advice, and resources that they have provided our team. We would not be able to have this great program at Trinidad State College without this grant. We are sincerely grateful to all the many people who so graciously make this possible!

References:

Essays Biological and Chemical, The Great London Chemists, Section I (p. 19), Archibald Constable & Company Ltd. 1908