

Design of an All-Terrain Rover

New Jersey Governor's School of Engineering and Technology

TANISHQ AGGARWAL
tanishq.aggarwal.11@gmail.com

GARGI SADALGEKAR
g2sadal@gmail.com

WILLIAM LI
superwilliamli@yahoo.com

ANITA ZIRNGIBL
anita.m.zirngibl@gmail.com

ABHINAV RAGHUNATHAN
abhinavr2121@gmail.com

Abstract

Ever since the onset of robotics, humans have been finding increasingly clever ways to traverse harsher terrain. Traditionally, two approaches to robotic locomotion have been used: the use of wheels or the use of "legs". This paper describes the design and construction process of a new type of All-Terrain Rover (ATR) that uses a hybrid design involving both wheels and "legs". The ATR consists of four wheels, with six linear actuators extending radially in a hexagonal pattern from each wheel, as well as two heavy arm-like structures that serve to dynamically alter the center of mass of the vehicle. This design gives the rover significant versatility, allowing it to adapt to both flat terrain (which it can traverse via wheels) and rugged terrain (through the motional flexibility provided by the leg-like actuators.) Using computer-aided design software (CAD), different iterations of the wheel were designed, analyzed using finite element analysis (FEA), and laser cut and put together from wood. A prototype of the wheel has been fully assembled and is ready for testing.

1 Introduction

In today's world, there are two primary methods of robotic locomotion: the use of wheels, and the use of legs, but neither are entirely

sufficient in every situation a rover would experience. While wheeled robots are efficient for crossing large, flat landscapes, they struggle when their path contains obstacles that are greater than the radii of their wheels. Legged robots are far more flexible and are capable of crossing a variety of terrain, but also suffer in terms of their physical and software complexity and speed of locomotion. The All-Terrain Rover, or ATR, combines the ideas seen in both wheeled and legged robots in a "hybrid" system. The design of the All-Terrain Rover consists of a four wheel system, with each wheel having the ability to alter its radius via six linear actuators arranged in a symmetric pattern across its diameter. This design is able to handle various obstacles and can help humans in search and rescue missions by driving through the wreckage of natural disasters without risking more lives. It could also be repurposed and used to tackle the uneven geography of Mars.

This study continues on the heels of many ongoing experimental projects in unconventional robotic locomotion. These projects include BigDog of Boston Dynamics, and HyQ conducted by Claudio Semini and the University of Genoa, both of which focus on optimizing legged robots.

The ATR's wheels include telescopic linear actuators which extend radially from the center of the wheel. The extension and retraction of the actuators act both to move and to stabilize the rover. The rover's body is also able to dy-

namically alter its orientation by manipulating its center of mass. The additional functionality provided by the combination of actuators and movable weights afford the rover maneuvering capabilities that far outstrip those of wheeled and legged rovers.

2 Background

2.1 Shortcomings of Wheeled and Legged Robots

Wheeled robots are far simpler to both build and program than legged ones. Their maneuverability is flexible on a level surface, especially wheels that utilize differential steering and are able to reach high speeds fairly easily. However, the lack of variety in their locomotive ability hinders the robot in situations that require a higher degree of flexibility (i.e. large obstacles or gaps).

As seen in the diagram above, a wheel is unable to clear a height greater than its radius and will only barely be able to clear one equal to its radius.

A wheel is also unable to cross a gap equal to or larger than its diameter without getting stuck. For these reasons, a wheeled-rover is severely crippled on rough terrain.

Legged robots have a distinct advantage over wheeled robots in that they are able to climb steps, cross large gaps, and walk on rough terrain by altering their geometry. However, their ability to handle such obstacles comes at the cost of simplicity; legged robots often require complex designs and programming in order to yield the coordination needed for the limbs to function in concert.

In his essay, *Principles of Robotic Motion*, Böttcher derives an equation for the total number of lift and release event combinations, N , of any legged robot based on the number of its legs, k . Lift and release event combinations refer to the combinations of raised and lowered

legs where the only two positions possibilities for each leg are raised and lowered.

$$N = (2k - 1)! \quad (1)$$

Based on this equation a four legged robot has 5040 leg positions, only including the lifting and releasing mechanism. Additional joints provide even more flexibility which will require more servos and make programming correspondingly more cumbersome.

A legged robot, despite having wide degrees of freedom in its leg positions, is still limited in other capacities. For instance, the length which a legged robot is able to travel is always limited by the length of its legs and how fast it can move them, meaning its speed is fundamentally slower than that of a wheeled robot (whose wheels can spin at an almost arbitrary rate). Similarly, a robot can only climb over or onto obstacles shorter than the height of its fully bent leg. These limitations highlight the discrepancy between the high degree of complexity of a legged robot and its limited actual maneuvering capabilities.

A design, therefore, that enables a rover to alter the radius of its wheels by propping it on to extendable leg-like struts, allows the rover to have several types of locomotive function, and thus through this mesh of linear and rotational components, a robot of reasonable simplicity yet significant complexity and speed can be designed.

2.2 Altering the Rover's Orientation

In addition to the increased maneuverability provided by the linear actuators, a robot that is able to alter its orientation drastically increases both its height and gap clearing capabilities. A robot that can flip over no longer uses only its wheels to overcome a height and instead uses its entire body. This increases both of the distances (height and gap) that the rover is able

to clear immensely.

Image of rover flipping over wall.

Figure 2.1: Height clearing capabilities of a rover that is able to alter its orientation.

In order to minimize the torque necessary for the motor to exert when flipping the rover, the center of mass of the rover must be shifted to one wheel axle, which becomes the axis of rotation. The torque necessary to balance to robot on one wheel axle, in preparation for the orientation alteration, is based on equations defining torque:

$$\tau = \mathbf{F} \cdot \mathbf{d}$$

$$\mathbf{F}_r \cdot \mathbf{d}_r = \mathbf{F}_a \cdot \mathbf{d}_a$$

Taking magnitudes, we have

$$M_r g d_r = M_a g d_a$$

$$M_r d_r = M_a d_a \quad (2)$$

Where M_r, M_a are the masses of the rover and arm, respectively, and d_r, d_a are the distances from the axis of rotation to the center of mass of the rover and arm, respectively.

3 Design Process

3.1 Wheel Design

The original concept for the ATR wheel was designed with six telescopic linear actuators that extend from its center. In order for the rover to traverse on actuator legs without the wheel contacting the ground, the actuator has to extend at least twice the length of the radius (Sanzari). The available linear actuators simply could not do this on their own, but the telescopic mechanism doubled the distance of extension of a linear actuator, and coupling the available actuators to the telescopic ones solved the problem (in theory).

Image of telescopic linear actuator from Mike's paper.

Figure 3.1: Telescopic linear actuator design (Sanzari).

The telescopic linear actuator is a compound pulley system (see above figure) in reverse. It doubles the extension of the inner actuator via the appendages attached around it. As seen above, the telescopic mechanism requires a string wound around an inner and outer tube, both of which surround the actuator. The string is fed through channels in each of the tubes; these channels act as the surface of pulleys in the compound pulley system.

The problem with this radial design was the sheer amount of material and time needed in order for the wheel to be constructed: each telescopic actuator requires (to describe briefly) an inner tube, an outer tube, plates on either end of the tubes to connect them to each other as well as to the actuator, and string. While the string is not expensive, the process of tying it in a configuration taut enough to allow for the telescopic mechanism to work would be a hugely time-consuming task for 24 separate actuators.

As a result, the design for the wheel was heavily modified. The second design for the wheel involved layers of circular plates with a pair of (non-telescopic) linear actuators fitted between two layers, yielding a total of three actuator pairs fitted between four circular plates. Within each layer, the two actuators would be placed in opposite directions, with their lengths spanning the diameter of the plate, and each layer's actuators would form a 60 degree angle with respect to any layer above or below it, so that the overall wheel would have actuators spaced in six symmetrically spaced directions. A foot would be attached to the head of each actuator and fit between the plates. This design was temporarily scrapped (but would later reappear in the final version) due to weaknesses in the foot found via finite element analysis: ow-

ing to the width constraints posed on the foot by the small amount of space available between the plates, the design was considered infeasible.

The third iteration of the wheel consisted of six planks joined in the shape of a hexagon that were sandwiched by two circular plates. Each pair of opposing planks had cutouts that fit around the shafts of a pair of actuators that faced opposite directions and spanned the diameter of the wheel. Attached to the outer face of each plank was a cage that surrounded the actuator body in order to prevent it from falling out. The foot, when attached to the head of the actuator, was intended to fit with the perimeter of the outer plates, so its thickness was no longer significantly limited. However, in order to fit both the foot and the cage within the perimeter of the plates, the diameter of the circular plates needed to be increased. Doing so not only violated materials constraints, but would cause the length that the actuators extended beyond the wheel to decrease to the point that the actuators wouldn't make contact with the ground.

In the final version, the base of each actuator is attached to the edge of the wheel, and its body spans the entire diameter of the wheel. In this regard it is similar to the second iteration, except it contains support tabs that enclose the actuator shaft housing and body, as well as E-shaped support elements around the edges of the wheels.

3.2 Foot Design

The general design of the foot remained constant throughout the design as an isosceles trapezoid with an arc joined to the larger of the bases. The foot attaches to the actuator head and sits within the perimeter of the wheel when the actuator is retracted. When extended, the actuator pushes the foot outside the wheel perimeter where it makes contact with the ground at about a 30 degree angle. As the design for the wheel changed, the design for

the foot had to be adapted. The original design included a solid foot in the aforementioned configuration that fit between the two face plates. In order to save material and fit into the constraints of newer wheel designs, the solid foot design was replaced with a frame design where two end plates shaped in the original configuration are connected by two side plates that were connected using a tab design. The center and top of the foot are hollow and that part of that space is used to house the actuator head and its connecting pieces.

The assembly in the center of the foot is what the head of the actuator connects to and what provides structure to the rest of the frame.

3.3 Arm Design

The original design of the robot included a single arm which provided additional locomotive capabilities. The arm extends from the center of the rover and had 360 degrees of mobility, allowing it to function even when the rover had drastically altered its orientation. The arm is double jointed adding another degree of flexibility. The original arm design functions similar to a limb by adding support when needed and using its rigid structure to help the rover overcome large obstacles.

The revised design for the arm not only alters its design, but is also a major part of its function. In order for the rover to be able to drastically alter its orientation, a certain degree of torque must be applied in order to reduce stress on the arm motor. The arm's new purpose is to provide this relief. Although a variety of designs were considered, it was decided that the new rover design should include two arms. The arms are both single jointed and are each comprised of a weight that can be moved by a pulley. The arms are attached to the two opposing edges of the chassis (in close proximity to the two wheel axles) and pass through slits that span the length of the chassis. When the ATR changes its orientation, the arm attached

closest to the axis of rotation is rotated so that it extends outside the perimeter of the rover. The weight can then be moved to the end of the arm so that it creates the torque required to flip the rover. This should allow the motor to easily rotate the entire body of the rover about a wheel axle.

3.4 Programming

3.4.1 Control Flow of Directional/Actuator Commands

The ATR is built with a series of programmatic components that interact with each other to allow the operator full control of the rover's motion. Firstly, the hardware that directly controls the actuators and the wheels is an Arduino board. The board receives commands from a remote computer that issues directions in Python, and transfers these commands to its domain (wheels and actuators).

3.4.2 Controlling Actuator Extension Length

The ATR's functionality depends heavily on the user's ability to extend and retract the actuators to specific lengths. To facilitate user experience, the code created accepts a desired extension (in millimeters) and extends to that length. To process the extension amount into an analog output to the actuator, the program uses a function created by Arduino's map function to transpose the extension length to an analog value.

After multiple trials of testing the extension, a consistent amount of error was observed. The data collected was graphed (see Graph) and a linear line of best fit with an R-squared value of 0.9994 was observed. Using this trend line to transform the input data before mapping, the actuator code yielded accurate extensions with a margin of error around 1 millimeter.

3.4.3 Python-Arduino Communication

Because the Arduino has limited memory and computing power but is useful for processing and relaying low-level commands, a more capable platform was necessary for programming the ATR. Python was chosen for its versatility and its ability to be easily integrated with Arduino and OpenCV. Python was setup to communicate with the Arduino board via Serial, using a code consisting of a sequence of characters and numbers. The user input is processed through a Graphical User Interface written in python, passed to Arduino via the sequence of an alphanumeric sequence, and executed by the actuators.

3.4.4 OpenCV and Image Processing

Because the wheel does not have built-in motor encoders, it was necessary to devise an alternate method for determining the exact radial position (in degrees) from the origin. Without any rotational sensors or gyroscopic devices, OpenCV was the best choice for such a task. OpenCV (Open Computer Vision) is a computer vision library that allows a script (written in Python) to get and process individual frames from a live video feed (via webcam positioned behind the wheel) and look for specific symbols to denote positions of the wheel. For example, the origin will have a specific shape that the script can identify and label as the origin.

The mechanism that allows the script to make such a conclusion is called "Contour Matching". By reducing shapes to their most basic contours and then counting them, a shape can be identified quickly and accurately. Contour Matching is very efficient, and - especially with OpenCV - is easy to implement. By positioning 12 such symbols around the circumference of the wheel, the Python script can detect the rotational position of the wheel and identify which actuators should be extended/retracted at a given time.

In order to maintain the best accuracy and eliminate the chance of any overlap between the symbols on the back of the wheel, it is important to test the size of the field of vision and to orient the camera in a way that allows it to focus on only one symbol and not its neighboring symbols. In this way, the camera was calibrated to detect only one image that is easily readable.

3.4.5 Graphical User Interface

In addition to the script that actually controls the actuators/wheels, software to easily edit the speed of the wheel and the retraction/extension distance for the actuators is crucial. Continuing the use of Python within the software side, a Graphical User Interface (GUI) was developed. The GUI, written in Python, provides an easy way to select an actuator and extend/retract it with a specified distance. The actuator is located by Wheel Number (#1 through #4) and Actuator Letter (A - F). Furthermore, the GUI contains the trigger for the Python-Arduino interaction (explained in "Python and Arduino Communication"). In addition, the GUI graphically shows the radial position of the wheels with regard to the origin with a degree measure from the OpenCV Python script (explained "OpenCV and Image Processing"). With this information, it becomes possible to correctly select the actuator that needs to be extended/retracted for the robot to continue its motion.

3.5 Electrical Integration

One of the obstacles with building the ATR is the problem of connecting wires from a stationary frame to rotating wheels. A direct, wired connection would result in the twisting of wires, so slip rings are needed for attachment. Slip rings have an inner cylinder and an outer shell that rotate around each other. By keeping the inner cylinder fixed to the wheel axle and allowing the outer shell rotate with the wheel, the

wheel can spin (relative to the ATR) without having its wires becoming entangled.

4 Assembly Process

The majority of the wheel structure is laser cut from baltic birch wood. The assembly of the laser cut wood parts is completed in specific order so that all the parts to assume their proper positions within the structure. First, the inner support structures and inner base structures are inserted into the inner two plates. Then the actuators are inserted before the 'E' support structures are added to the edges of the plates. Finally, the outer plates enclose either side of the wheel. Some problems with constructions are the flaws. When fully assembled, the tight fit of the parts create enough friction to hold the wheel together without any bolts. However, bolts are inserted through the wheel for additional security. The body of the robot was constructed using VEX Robotics parts. Stepper motors were used to power the wheels and arms.

5 Results

5.1 Currently available results

The rover wheel was sufficiently stable even without additional bolts (i.e. it held together due to the friction between puzzle pieces). It was also slightly heavier than expected.

5.2 Further Testing

5.2.1 Height Clearing Capabilities

The rover's height clearing capabilities will be tested by having the rover drive over heights starting at 110 mm, the radius of the wheel, and increasing in increments of 10 mm in order to gauge the maximum height that the rover

can clear. A height would be declared successfully cleared when the rover was able to cross it with a success rate of 90% .

5.2.2 Gap Clearing Capabilities

The rover's gap clearing capabilities will be tested by programming the rover to cross a gaps of various widths. The widths will begin at 220 mm, the diameter of the wheel without extended linear actuators. The widths will be increased at increments of 20 mm until the rover is no longer able to successfully clear the gap. A gap will be declared successfully cleared when the rover is able to clear it with an 90% success rate.

5.2.3 Various Terrains

The rover is designed to be able to navigate a variety of terrains. The success of the rover on each terrain will be measured by the distance that the rover travels in 60 seconds. These terrains will include a flat surface (i.e. a floor), a rough surface (i.e. a bed of gravel), and a soft surface (i.e. mud).

6 Conclusion

The design for a wheel with linear actuators can be constructed using a press fit. The design of the wheel is effective because it resulted in a wheel that is structurally sound on all sides, while allowing the actuators to extend to the necessary length past the wheel.

6.1 Further Steps

The puzzle-like design of the wheels was adopted in order to simplify construction and reduce the number of bolts and screws. While the wheels produced from this concept were fairly sturdy, they are likely not rugged enough for use in field operations. Later iterations of the

wheel need to be redesigned with tougher materials and better attachment methods. In addition, the chassis of the ATR is constructed from VEX robotics parts, but a future version should be constructed from specially designed parts to maximize material efficiency.

From a software point of view, automation is a major step forward in the future. Even if the robot does not become fully autonomous, patterns of code should be developed for predictable scenarios, such as crossing a gap or climbing a cliff. To increase the ATR's capabilities, more sensors need to be connected. For example, distance sensors will help the ATR detect obstacles in its path and predict the correct maneuver to perform. With more sensors, the robot can observe its environment and learn to match maneuvers to scenarios.

7 Acknowledgments

The authors gratefully acknowledge Michael Sanzari of Rutgers University, whose technical experience and knowledge of engineering proved to be extremely essential to this study. The authors also thank the Rutgers University branch of the IEEE for allowing them to use their facilities and resources. The authors appreciate the work of the Makerspace workers who helped laser cut the wheel parts.

The authors would like to take this time to thank Rutgers University's Wireless Information Network Lab (WINLAB), which generously allowed the authors to use their equipment and facilities. The authors sincerely appreciate the efforts of Charles McGrew from Hackerspace, who loaned the linear actuators that made this entire concept possible. The authors would also like to thank Alexander Hobbs, who aided research as project mentor by providing transportation and guidance, and to the New Jersey Governor's School of Engineering and Technology, the program under which this research was conducted. Finally, the authors would like to thank the dean

of the program, Ilene Rosen, and Assistant Dean the program possible.
Jean Antoine Patrick for their support and re-
view of the project through the program, as well
as sponsors of the GSET program for making

8 References