Mechanical Snake

Christopher Chung
chris.ch.987@gmail.com

Isha Mehrotra
imehrotra15@gmail.com

Owen France
owen15france@gmail.com

Robert Phillips
robertmorganphillips@gmail.com

James Lee
jameslee96@gmail.com

New Jersey Governor's School of Engineering and Technology 2014

Abstract

Most robotic technology tends to move in straight lines with set angles. Many robot designs using unidirectional motion have already been created; however, fewer developments have been made in creating curved patterns. The flexible S-shaped motion of snakes enables them to fit into otherwise inaccessible spaces, leading to potential applications in search and rescue missions and medicine. The addition of Jasper voice recognition technology expands the utility of such technology by integrating new methods of control. The multi-terrain mechanical snake mimics its namesake's locomotion by alternating the movement of each of its segments to create a curved shape. Voice recognition technology still needs further improvement to be successfully implemented in conjunction with the snake. In replicating a snake's locomotion, the multi-terrain robot can be used to complete tasks that may be too dangerous or difficult for people or less versatile robots. Advancements in the design and locomotion enabled the snake to achieve two main types of motion—forward motion and sidewinding.

1. Introduction

The ability to replicate designs from nature has been implemented across many disciplines, from the hypodermic needle shaped after the proboscis of a mosquito to the honeycomb structures used in aircraft.[1] Snake anatomy has much to offer due to its unique shape and associated motion. Within *Serpentes*, the suborder that includes all 3,400 species of snakes, there are species that slither, slide, swim, and glide, traversing extreme environments.[2] Creating a robot that can imitate snake locomotion has several potential applications. Similar biomimicry is already being implemented for a myriad of uses, from exploring the rubble of collapsed buildings to relaying images within the body in preparation for surgery.[3]

Though such innovations are revolutionary, in no way are they perfect; improvement is still needed in order for mechanical snakes to become regularly used. Such robots must be able to navigate through tight spaces, travel undamaged into radioactive areas, and move silently during reconnaissance missions. In addition, its capability to operate wirelessly is crucial. Without a decent range, many of its intended functions would be aimless. Advancement in the design of the mechanical snake was achieved through a series of small improvements, but voice recognition control was not entirely integrated.

2. Background

While the slithering of snakes looks fairly simple, in reality, the movement is complex and relies on the careful collaboration and balance between a variety of forces. There are several different types of serpentine motion, but the most well-known is referred to as lateral undulation. Although no mathematical model of nature is perfect, the most accurate depiction of lateral undulation is as follows:

$$x(s) = \int_0^s \cos(\zeta_\sigma)\, d\sigma$$
$$y(s) = \int_0^s \sin(\zeta_\sigma)\, d\sigma \qquad \text{Figure 1}$$

$\zeta_\sigma = a\cos(b\sigma) + c\sigma$ [4]

> [T]he parameters *a*, *b*, and *c* determine the shape of the serpentine motion...Basically, *a* changes the appearance of the curve, *b* changes the number of phases, and *c* changes the direction.[4]

The friction against the scales of the snake is more difficult to model. When building a mechanical snake, taking these forces of friction into consideration is vital—it provides the forward motion. Without it, the snake would just writhe in place.[5] It is crucial for the snake to achieve a proper balance between its frictional forces because they can either hinder the movement of the snake or allow the snake to propel itself forwards.

The snake was originally created as a senior design project under Professor Stephen Tse by a group of Rutgers Mechanical and Aerospace Engineering seniors. Their main objective was to create a robotic snake which would mimic the serpentine motion of snakes and be able to traverse multiple terrains. They were successful in creating and 3D printing parts for the snake, as well as integrating servos and programming their associated microcontroller. However, based on the video footage of their snake's motion, they accomplished neither climbing nor significant forward motion.[6]

3. Mechanical Snake Design and Locomotion

The general design of the mechanical snake was established by the senior design project. However, several modifications were needed to improve the snake's design and motion. The adjustments to both the program used to run the snake and to the body of the snake itself resulted in an overall stronger and faster snake.

3.1 Design

The snake design consists of four independently-controlled, detachable segments. Each module consists of a black "backbone" segment connected to two ventral claws, one on each side of the backbone. Each claw is attached to the main body segment with a servo to control opening and closing movements. The servos on the claws are attached in a reduced gear ratio that favors torque over speed for more gripping and pulling power. This is partially due to the weight of the snake, but also contributes greatly to the snake's ability to travel on various rougher terrains.

The separate modules are connected by a set of interlocking joints, each powered by a servo motor: one joint controls lateral motion and the other controls vertical motion. All of the servos have a maximum rotational range of ninety degrees. Thus, calibrating them was essential to ensure that they would work properly in tandem. To maximize efficiency, the gear systems used different ratios tailored to the task. For the side-to-side motion and the up-and-down motions, the ratio of the gears from servo to snake part was 1:3. By doing this, the angle of the snake's turn was reduced; however, this also increased the torque of each segment threefold, a tradeoff that greatly increased forward movement. The lowered range of turning was also mitigated by the programmability of the snake's motion; the precise control of each snake module to turn to set angles allowed the snake to use its multiple segments to compensate for the lack of range of each individual segment.

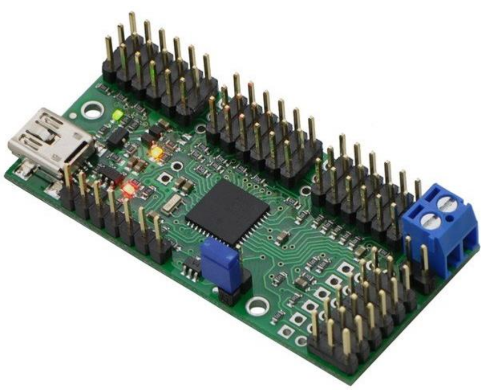


Figure 2: The Pololu Mini Maestro 24-channel servo controller; used to control the motion of the snake

The snake designed by the undergraduates was greatly limited by their programming. Because of the gear ratio used to increase the torque, each joint in the snake has a net turning radius of only thirty degrees. In the new program, the motion of the servos is staggered; they turn in the same direction at different times. This allows the snake to have a net turning radius of ninety degrees by turning multiple separate thirty degree sections, a significant improvement upon its previous configuration. This is important in mimicking the design of an actual snake, because by having it turn ninety degrees, the snake can increase its friction against the surface of the ground, allowing more of the servos' circular motion to convert to forward motion. In a real snake's serpentine motion, the snake relies on shifting its weight so its scales increase friction with the ground when pushing backwards while allowing smooth movement sliding forwards. Thus, the staggered movement helps the snake more closely resemble the movement of an actual snake.

All of the servos are plugged into and controlled by the Pololu Mini Maestro 24-channel USB servo controller, a microcontroller programmed in the Pololu programming language using the Maestro Control Center, which also distributes power to every servo. Servos operate on pulse-width modulation (PWM) signals, which interface directly with the Mini Maestro's output pins; the Maestro sends output signals to each servo and depending on the width of the pulse each servo receives, it moves to a different position. The Mini Maestro runs on a 6V lantern battery which is currently not integrated into the snake's design because of weight and size limitations. However, a smaller and lighter alternative can easily be used as a replacement and attached to the body of the snake.

Figure 3: Mechanical Snake connected to the Pololu Mini Maestro 24 microcontroller

3.1.1 CAD Drafting

All of the components of the snake were designed and drafted in SolidWorks by the undergraduate Rutgers group. These SolidWorks files later proved useful when several parts of the original design failed and had to be reprinted.

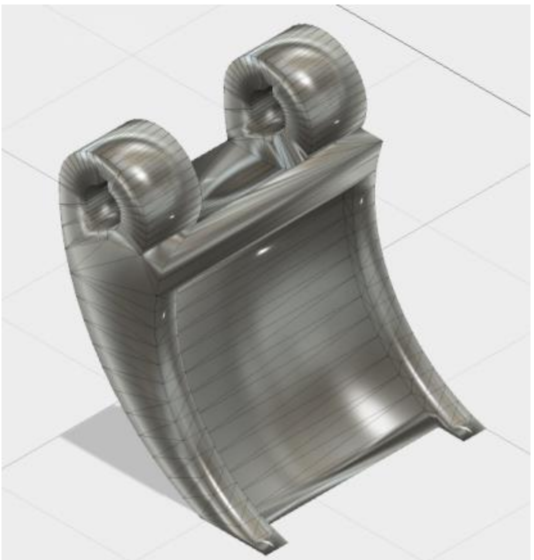


Figure 4: 3D rendering of the claw

3.1.2 3D Printing

The major components of the mechanical snake were already printed and assembled by the undergraduate Rutgers team. 3D printing is accomplished by using a heated ejector; this ejector takes plastic from a supply, melts it down, and ejects it in the form of a thin string. The undergraduates used a coordinate printer, which uses three motors to control the ejector's location in the x, y, and z directions. The printer then fabricates the object by stacking the molten plastic streams, which subsequently solidify in the shape of the part. Properties of the printed part can be adjusted by changing the density of the printed part and the speed at which the printer operates.[7] Although most of the parts functioned adequately, some of the gears seemed too porous and thus too weak for use in the robot. The teeth of some of the gears failed because they could not withstand the weight of the mechanism they were meant to move. Similarly, some of the axles and claws were in poor condition. These parts were reprinted and replaced on the robot.

3.1.3 Assembly

Most of the parts of the mechanical snake were 3D printed using ABS plastic. A great part of the body of the snake was taken from the original senior projects group, which already printed many of the different sections. Stronger newly printed parts were used in conjunction with the original pieces to complete the snake.

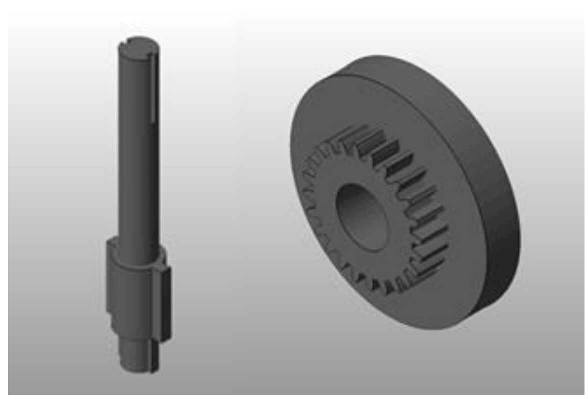


Figure 5: Axle and gear for claw

Reprinting every part would have greatly improved the snake itself; however, this was not possible due to time and cost constraints. Therefore, only a few pieces were reprinted; namely, newly printed parts were focused on the gear systems, axles, and claws, which were central to the motion of the snake and had sustained damage. Most of the main chassis pieces were left as they were. Along with the 3D printed parts, connecting axles made out of metal were used. Although 3D printed parts displayed superb accuracy in shape, there were some areas that received great increased torque and strain. This caused some plastic parts in certain joints of the snake to start bending or even to start cracking. Bending plastic parts led to a lot of energy loss, so in these areas of high strain, stronger material, namely carved metal components, were used instead. This increased the strength and rigidity of the entire snake, allowing the power of the motors to translate more directly to movement without as much power loss, ultimately leading to optimization of the snake's motion.

3.1.4 Dimensions

The snake's length with all four segments is 23 inches. The snake's widths with closed and open claws are 6.625 inches and 10.25 inches, respectively. Its height when the claws are closed is 4.75 inches. The mechanical snake has a relatively large diameter compared to other mechanical snakes. This increased size made it a heavier snake with not only more downforce due to its greater mass, but also more contact area. With more downwards force and a greater area of contact with the ground, there was much more friction in all directions. Although friction is a necessary force for the motion of the snake, there was way more than the optimal amount. This great amount of friction proved to be one of the most significant obstacles faced during the course of the project. Thus, a recommended change in future designs is to decrease the diameter. The increased weight also made it more difficult for the servos to control the snake because the servos can only output a certain amount of force.

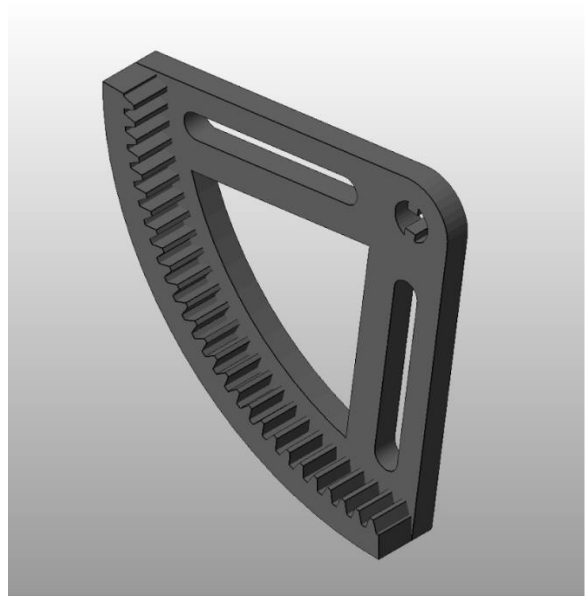


Figure 6: Quarter internal gear

3.2 Locomotion

The Pololu Mini Maestro servo controller is central to the locomotion of the snake. Each segment requires two servos to control its claws and two servos t control its movement with each adjacent segment. In total, eleven servos were utilized to control the motion of the mechanical snake. In order to operate and program all of the servos together and achieve synchronized automatic motion, the servos were connected to the servo controller. The controller can be programmed using the Maestro Control Center. In the program, each servo can be assigned a value and the states of all of the servos at a single time can be saved as a single frame defining each respective position. These frames can be compiled to form a continuous set of instructions telling each servo what position it should be in at any moment.

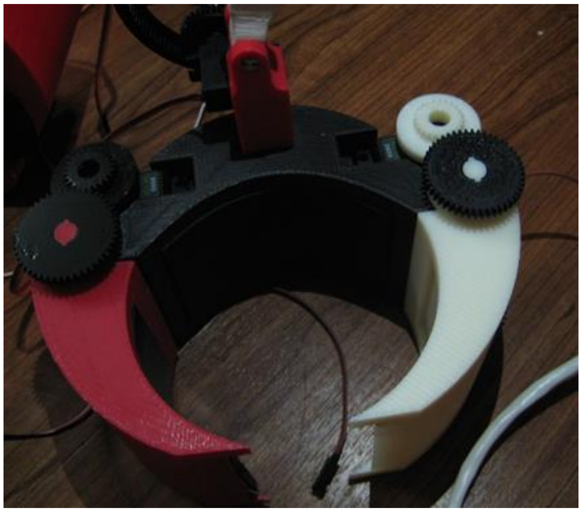


Figure 7: Repair; the white parts were reprinted and used to replace defective parts

3.2.1 Forward Motion

Analysis of the video created by the undergraduate senior project team revealed that the movement of the mechanical snake could greatly be improved and optimized. The snake in the seniors' video barely moved, requiring fast forwarding in order to see any significant locomotion. The program used by the seniors had the segments moving at the same time, causing the segments only to only move back and forth without contributing to forwards motion. This motion is ineffective because a majority of the servo's power went towards lateral motion, leaving no power propelling the snake forwards. Also, having the snake just move back and forth greatly detracted from its serpentine motion. In order to optimize the snake's motion and make it travel in a more serpentine fashion, the program needed to be modified. First, the movement of the segments was staggered. When a snake moves, not all of its segments move at the same time. There is a gradual wave of motion which propels it forwards. By staggering the segments, the motion was more snake-like.

One of the largest obstacles was adjusting the snake's orientation with the ground to optimize the frictional forces for motion. The snake relies upon the force of friction to push itself forward off the ground, but is simultaneously slowed down by friction. Therefore, it is essential that the mechanical snake be able to change its orientation to alter its frictional forces. Real snakes shift their weight in order to vary the force of friction with the ground. However, the mechanical snake used the angle of each section in order to alter the frictional forces. Staggering the movement aligned certain segments parallel to the direction of motion, letting them slide easily, while turning certain segments perpendicular to the direction of motion, which made it possible to gain traction and push the snake forward. This alternating pattern of parallel and perpendicular segments simulates the alternating frictional forces of an actual snake and results in significantly better movement in the snake robot. The program tells each segment to move back and forth, but with a 180 microsecond delay between each segment and the next.

3.2.2 Sidewinding Motion

The next step in developing the snake was programming sidewinding motion, or motion directly sideways. This is because the previously programmed forwards motion does not incorporate any mechanism to turn the snake and thus development of lateral motion was necessary. The claws on the snake were used to achieve this motion. As mentioned before, each segment has two claws, each with its individual servo motor. The snake was programmed to sidewind by alternating opening and closing of each claw. The program written can be broken down into a few steps: opening the claw in the direction intended for movement, closing that same claw and opening the other claw at the same time (effectively shifting the snake's body to the side), and lastly, closing the latter claw. This motion is looped within the program. The servos on the claws are larger than those used for lateral undulation and thus the sidewinding motion offers increased power and torque, making it possible for the snake to move on rougher terrains. This motion, although effective in having the snake move in a lateral direction moving the snake laterally, does not imitate the sidewinding motion of an actual snake, and was dubbed pseudo-sidewinding. Although it does not perfectly mimic nature, the motion was nonetheless an improvement upon the progress of the seniors' project.

4. Integration of Technology

The integration of new technologies into existing ones is crucial for continuous innovation and improvement in industry. Better products are created every day by combining prevailing products with newly developed tools. Examples include nanomaterial-reinforced materials such as concrete and glass and cars powered by solar power and electricity. Many others have accomplished the feat of creating a robotic snake. However, there has been less work on alternative methods of controlling the snake's motion other than by direct feeding of instructions. The second major component of the project was the integration of voice recognition technology to control the snake.

4.1 Jasper Voice Recognition

Jasper, created by Shubro Saha and Charlie Marsh, is "an open-source platform for voice-controlled applications." It enables the user to input vocal commands through a microphone and uses a Raspberry Pi to execute them.[8]

A Raspbian (based on Debian, itself a version of the operating system Linux) disk image was available with Jasper preinstalled, and following the setup instructions on the Jasper website yielded a working device. However, Jasper requires constant Internet access, and it often hears commands incorrectly. Considering these difficulties, it was not possible to implement forward motion by sending vocal commands to the Mini Maestros via the Wixels as planned. However, neither the module nor Jasper overall were used in the final design, largely due to the poor accuracy of Jasper itself.

4.2 Wixel

A wireless communication device was needed to enable the snake to move without a tether. There were multiple options for communicating wirelessly, including Bluetooth, XBees, and Pololu Wixels. Since the Mini Maestro servo controller used to manage the movement of the snake was made by Pololu, Wixels were used to ensure maximum integration between the wireless adapter and the Mini Maestro. The Wixels come as pairs, and operate by communicating with each other through radio. In order to test the capabilities of the Wixels, the blink program was used. One Wixel was connected to the computer and was used to transmit the code; this Wixel will be referred to as the master Wixel. The other Wixel, which will be referred to as the slave Wixel, was connected to a power source and connected to the master Wixel through radio. The blink program was run on the master Wixel, and it was observed that the slave Wixel began running the same code. The slave Wixel also stopped running the blink program after being removed from the range of the master Wixel – proof of connectivity between the Wixels.

Based on the data collected, to utilize all technologies within the snake, one Wixel would be connected to the Raspberry Pi, which would be hooked up to the Jasper system. In theory, this Wixel, connected through a USB, would receive the voice commands from Jasper and relay them through radio waves to the other Wixel, which would take the data, interpret it, and send it to the Maestro servo controller. The Maestro would then interpret the input it receives from the Wixel and implement a specific type of commands, corresponding to a specific type of motion. In this way, voice commands would be relayed to the snake wirelessly. It could not be evaluated whether this would happen though. The Wixel has a range of around fifty feet. With wireless communication technology, users could control the snake's movement freely.

Voice command was not implemented due to the Jasper software's lack of sensitivity in interpreting verbal input. However, if the voice recognition been more accurate, voice control of the snake would be feasible since it was concluded that the Wixels were able to successfully communicate.

Figure 8: Pololu Wixel – wireless communication with the snake

5. Results and Discussion

5.1 Snake

Reworking the code used by the senior group to minimize lateral movement, and using the snake's forces of friction to effectively move the snake forward proved key in optimizing the snake's locomotion. By causing the motors to succeed the previous ones, the movement of the mechanical snake effectively mirrors that of its biological counterpart. Because the snake is relatively heavy, the motion favored less friction over more. When the claws were extended, the rubber on the snake's underside would restrict its forward movement. The snake implementing the reworked program was able to reach speeds of around 1.16 in/sec, significantly faster than the snake using the seniors' code.

5.2 Jasper & Raspberry Pi

The Jasper required a fair amount of troubleshooting, despite having the use of the image with the precompiled software installed. However, tidbits from various support message boards were fully sufficient to resolve the issues initially present.9 Jasper was successfully configured to run upon startup of the Raspberry Pi, recognizing voices, distinguishing words, and responding appropriately. Jasper's functionality was tested by running some of the pre-loaded modules—the weather module, the joke module, and the news module, among others, were run successfully. Jasper was able to identify verbal input and run the corresponding module. However, due to Jasper's poor voice recognition accuracy, it had a difficult time correctly identifying verbal input and could not identify the proper trigger word for the new module. As a result, Jasper was not used for control of the snake.

5.3 Wireless Connectivity

Communication between the two Wixels was established. With the Wixel Configuration Utility, connectivity was achieved by uploading the same code onto each Wixel. As long as they are in range, there is no noticeable delay in communication. One obstacle identified in combining the technologies used in the project would be the differing power requirements for the Mini Maestro and the Wixels (these being approximately 6V and 3.3V, respectively). The use of voltage regulators would resolve this issue so the snake could run under optimal conditions for each of its components. Mounting the component onto the snake proved problematic though and caused difficulties that could not be resolved by the end of the project. The failure of the Jasper to recognize commands and time restraints eventually led to the omission of the Wixels from the final design.

6. Conclusion

Manipulation of the Pololu program greatly improved the snake, optimizing movement and allowing it to move forward much faster than before. In the final design, the snake was able to overcome obstacles with greater ease. Using vocal recognition software to control the snake has great promise as the technology improves. Generally, the robot mimics the movements of actual snakes and gives an understanding of what they have to offer engineers.

Future research can be conducted to integrate both the segments and code in such a way that its locomotion matches that of the actual snake. Beyond that, to match the friction and specific movements that scales allow would result in the most accurately mimicked snake that one could build. Serpentine motion is very difficult to replicate with current technologies, but has great potential for the future.[10]

7. Acknowledgements

Thank you to our mentor, Dr. Stephen Tse of the Mechanical and Aerospace Engineering Department for all of his help. We would also like to thank Priya Deo of the Carnegie Mellon Mechanical Snake project for her guidance. Thank you to the Governor’s School of Engineering and Technology as well as its sponsors: Rutgers University, The State of New Jersey, Morgan Stanley, Lockheed Martin, Silver Line Windows, South Jersey Industries, Inc., The Provident Bank Foundation, and Novo Nordisk. Furthermore, we would like to thank Ilene Rosen, Director of GSET, Dean Jean Patrick Antoine, Assistant Director of GSET and all of the counselors for giving us this opportunity. Also, thank you to Dr. Zhizhong Dong for his assistance. We would especially like to thank Alex Hobbs, our Residential Teaching Associate, for always supporting us and being there for us.

8. References

[1]Y. Bar-Cohen, *Biomimetics: Biologically Inspired Technologies*, 1st Ed. (Taylor and Francis Group, Boca Raton, FL, 2006).

[2]B. Cornett, “How Many Different Types of Snakes are there?”, *Reptile Knowledge*, (2014), <http://reptileknowledge.com/articles/article9.php> (20 July 2014).

[3]Coxworth, Ben. "Surgical Robot Snakes Its Way down the Throat." *Surgical Robot Snakes Its Way down the Throat*, (4 Apr. 2014), <http://www.gizmag.com/flex-system-surgical-snake-robot/31510/>, (10 July 2014.

[4]"Robot Snake." *Northwestern Mechatronics*. Northwestern University, Web. 11 July 2014.

[5]J. Hazel, M. Stone, M.S. Grace, and V.V. Krusuk, Journal of Biomechanics **32** (5), 477 (1999).

[6]A. Hashemi, D. Attia, M. Fung, D. Bhavsar, J. Payan, H. Riddle (unpublished).

[7]3D Printing Industry. "3D Printing Basics: The Free Beginner's Guide." *3D Printing Industry*. 3D Printing Industry, 2013. Web. 22 July 2014.

[8]S. Saha and C. Marsh, "Jasper", *Jasper*, (2013), <http://jasperproject.github.io/> (20 July 2014).

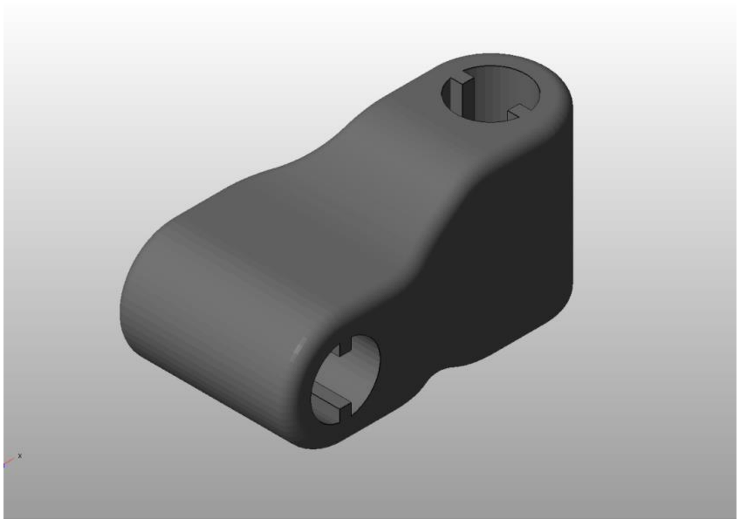
[9]"GitHub", *jasperproject* (2014), <https://github.com/jasperproject/jasper-client/issues> (21 July 2014).

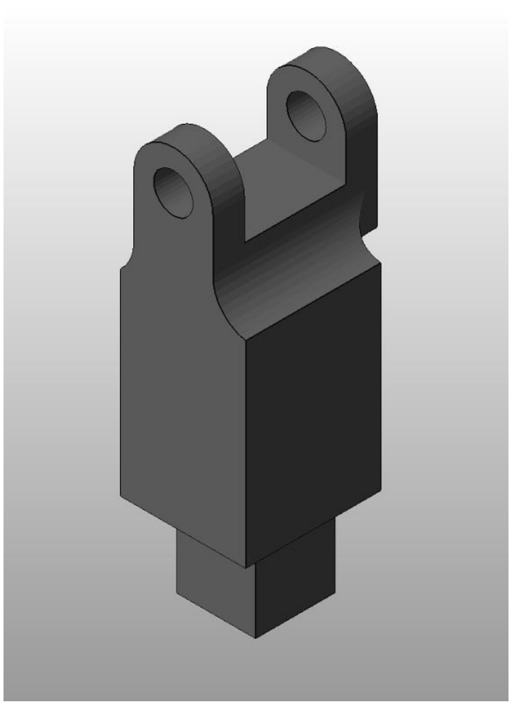
[10]Y. Shan and Y. Koren, IEEE Transaction on Systems, Man, and Cybernetics **23** (4), 1091 (1993).

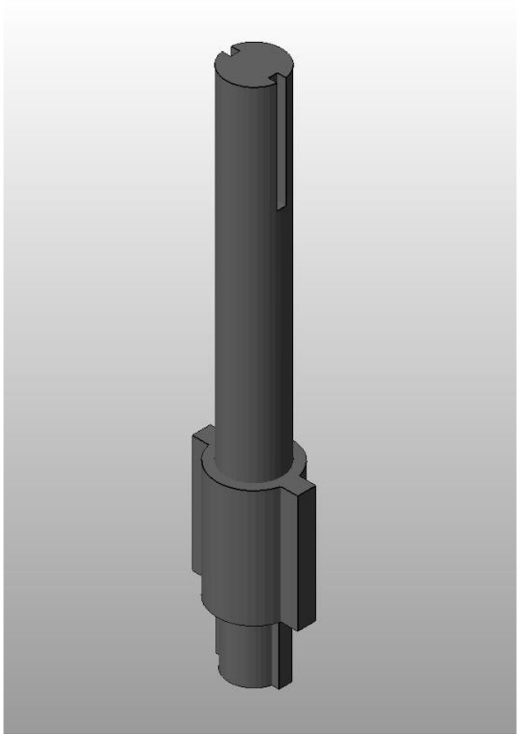
[11]P. Deo (Private Communications, 2014).

9. Appendix

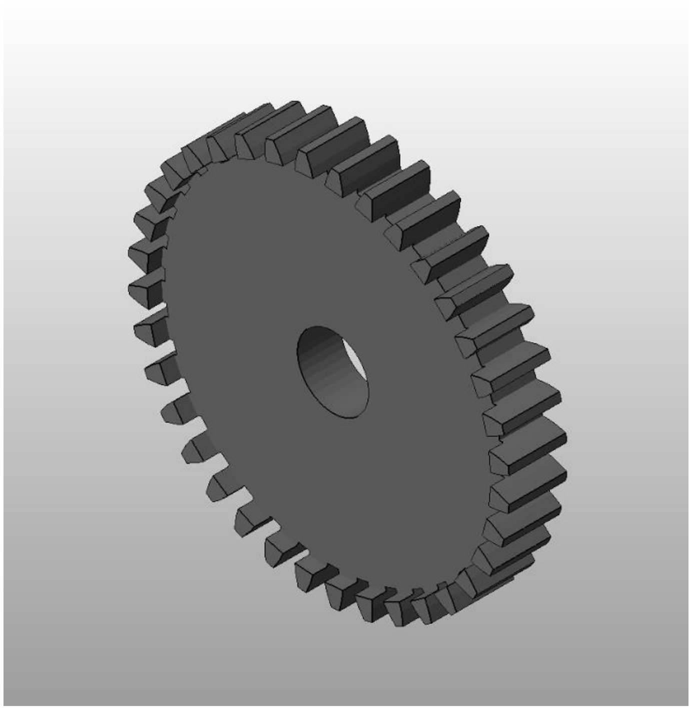
CAD Files:

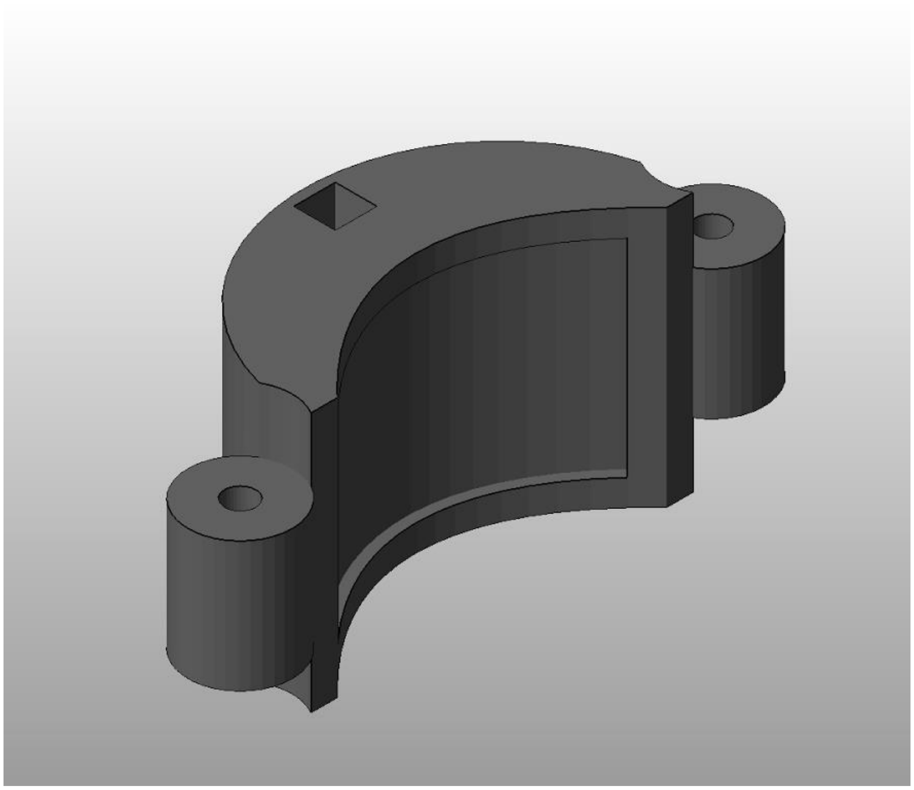

Connector between segments

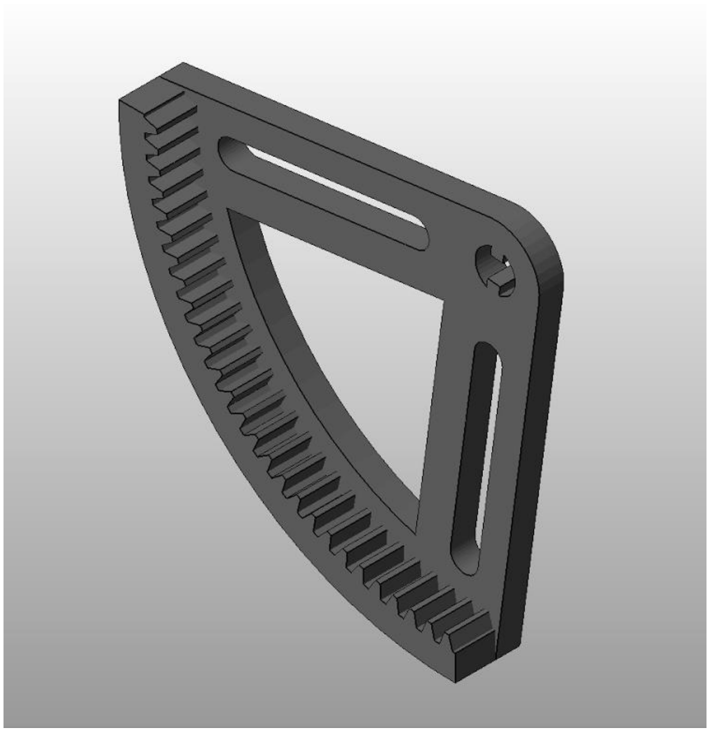

Connector between segments


Claw Axle


Servo gear attatchment


Gear for claws


Snake backbone segment


Quarter internal gear

Snake Code:

```
# SideWinding in a Group
begin
  500 0 8000 3968 8000 3968 8000
  3968 8192 3968 0 0 0
  0 0 0 0 0 0
  0 0 0 0 0 frame_0..8_10..23 # Frame 0
  500 8000 8000 8000 8000 frame_2_4_6_8 # Frame 1
  500 3968 3968 3968 3968 3968 3968
  3968 3968 frame_1..8 # Frame 2
repeat

sub frame_0..8_10..23
  23 servo
  22 servo
  21 servo
  20 servo
  19 servo
  18 servo
  17 servo
  16 servo
  15 servo
  14 servo
  13 servo
  12 servo
  11 servo
  10 servo
  8 servo
  7 servo
  6 servo
  5 servo
  4 servo
  3 servo
  2 servo
  1 servo
  0 servo
  delay
  return

sub frame_2_4_6_8
  8 servo
  6 servo
  4 servo
  2 servo
  delay
  return

sub frame_1..8
  8 servo
  7 servo
  6 servo
  5 servo
  4 servo
  3 servo
  2 servo
  1 servo
  delay
  return
```

```
# Forwards Motion
begin
  180 0 8000 8000 8000 0 0
  0 0 0 0 0 0
  0 0 0 0 0 0
  0 0 0 0 0 frame_0..8_10..23 # Frame 0
  180 3968 frame_1 # Frame 1
  180 3968 frame_2 # Frame 2
  180 3968 frame_3 # Frame 3
  180 8000 frame_1 # Frame 4
  180 8000 frame_2 # Frame 5
repeat

sub frame_0..8_10..23
  23 servo
  22 servo
  21 servo
  20 servo
  19 servo
  18 servo
  17 servo
  16 servo
  15 servo
  14 servo
  13 servo
  12 servo
  11 servo
  10 servo
  8 servo
  7 servo
  6 servo
  5 servo
  4 servo
  3 servo
  2 servo
  1 servo
  0 servo
  delay
  return

sub frame_1
  1 servo
  delay
  return

sub frame_2
  2 servo
  delay
  return

sub frame_3
  3 servo
  delay
  return
```