# Jormungand, an Autonomous Robotic Snake

Charles W. Eno, Dr. A. Antonio Arroyo
Machine Intelligence Laboratory
University of Florida
Department of Electrical Engineering

## 1. Introduction

In the Intelligent Machines Design Laboratory (IMDL) at the University of Florida, the best part, in my opinion, is the encouragement one gets to build on other peoples' designs and ideas as well as to come up with your own, new, designs and ideas. No one person would be able to do as much in one semester if this chance to explore previous work was unavailable. Because of this, IMDL keeps getting better and better the longer it is in existence. I started the class wanting to do a snake robot. There had been one previous snake robot (named Monty by Melissa Jones, in Fall '97) and I felt I could do an even better snake. The snake design I came up with was completely different from her design, but I was able to learn from some of the problems she encountered. The design incorporates original my own original programming code, code from the standard IMDL library, as well as from other sources. Many parts of the snake are inspired by previous work, from the IR sensors to the bump switches.

The project presented a challenge: Create a highly flexible, maneuverable robot that could incorporate interesting behaviors without legs or wheels. My design went through several stages where I refined the platform and developed the techniques to control the motion of my robot.

## 2. Integrated System

Jormungand is an autonomous snake robot and has no legs or wheels. The robot is built in an open frame structure that is rugged and flexible. Movements are controlled by nine servos, one at each joint of the body. Forward progress is accomplished through various serpentine movements. Multiple joints provide many ways to manipulate the body of the robotic snake.

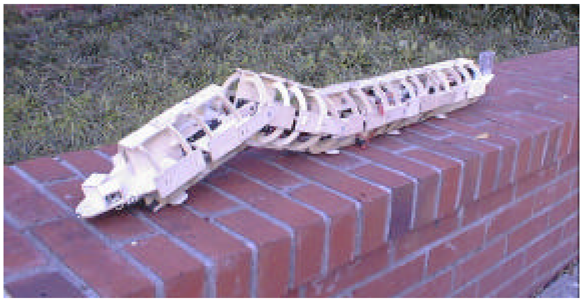

Figure 1. Overall Picture

Jormungand is controlled by two Motorola MC68HC11-E2 based single-chip boards manufactured by Mekatronix. One board serves as a servo controller. The other board acts a brain which controls the sensors and tells the servo controller how to position the servos.

The robot perceives the world through three sets of sensors. IR provides basic obstacle avoidance. Bump sensors initiate various actions. Tilt sensors indicate if the robot is not upright. Jormungand's programming resides in a continuous loop that keeps him moving forward while avoiding obstacles and staying upright.

## 3. Mobile Platform

### 3.1 Overview of the Platform

The snake platform is designed to be simple to assemble with many repeated parts. This is accomplished by each segment of Jormungand being essentially identical. The overall platform was designed to provide the maximum amount of freedom of movement with a minimum amount of complexity. Along with this was the need for an extremely rugged and durable snake to take day to day abuse. (See Figure 1.)

## 3.2 Overall Design

The final design was realized through several revisions. The original design was to be thin walled aluminum or plastic pipe to form the basis of each segment. This proved to be an awkward arrangement as it was apparent that connecting these segments and allowing them sufficient free movement would be difficult. The tubes would require internal structures to mount servos, batteries and other components such as sensors. This internal structure would be hard to mount components to, and once installed, would be nearly impossible to remove or adjust as needed. The solution is to remove the outer tube and leave the internal structure. This provides easy access to components and allows for easy assembly.

## 3.3 Head and Tail Design

The head is designed to provide a large platform for the sensors. Both the IR and bump sensors are mounted on the head with plenty of room for other sensors. The tail is similar to the head and has the battery for the computers mounted on it as well as a rattle to warn the outside world when Jormungand is upset. (See Figure 2.)

## 3.4 Processor Segment Design

The minimum diameter of the snake was dictated by the size of the microprocessor boards. This resulted in a snake 4 inches in diameter. The boards are mounted perpendicularly to the axis of the snake in two special segments. While the design involving the boards functions well, they would be impossible to replace, in the event of failure. Additionally it is difficult to attach plugs to the headers on the board due to the close spacing involved. (See Figure 3.)

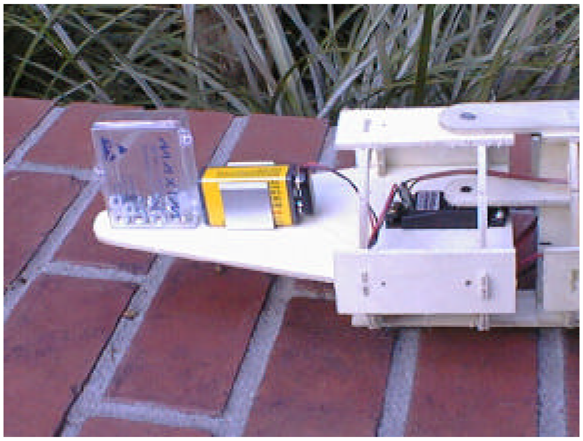

Figure 2. Jormungand’s Tail and Rattle

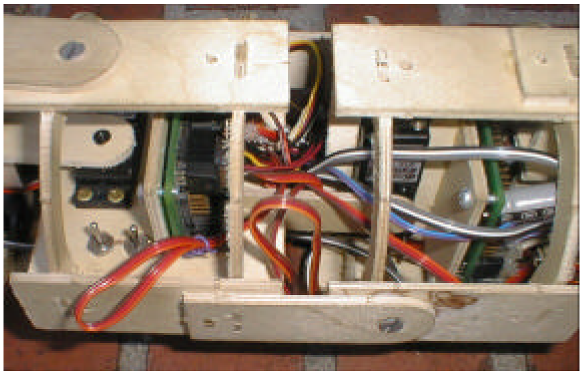

Figure 3. Processor Segments

## 3.5 Joint Design

The initial design called for each joint to have two degrees of freedom. The final design was a compromise by alternating directions of movement on each segment. This reduced the number of servos needed, an important factor in cost. In addition, the weight in each segment was reduced and allowed the overall snake to be longer.

## 3.6 Construction

Jormungand is assembled from 1/8" birch plywood. The pieces were drawn in AutoCAD and milled on a T-Tech machine.

Assembly was done initially with wood glue and later with super glue. Super glue is by far easier and faster to use.  The only hardware involved were small screws to mount the servos and bolts and lock nuts used for the hinges.

## 4. Actuation

### 4.1 Segment Actuation

Actuation is provided by standard RC-type servos.  The servos used were Tower Hobbies' STD TS-53, a 42 oz-in servo that costs $12.  They were chosen for their low cost and have proven reliable.  Each segment consists of a body frame and a servo.  The servo attaches via a wooden arm to the next segment.  Each segment is rotated ninety degrees from the previous and next segments to provide for both up-down and left-right movement.  The segments hinge on wooden arms protruding from the previous section.  The hinge lines up with the pivot of the servo.  The hinges reduce strain on the servo itself and provide for solid connection between segments.

### 4.2 Servo Control

Servos are attached to a MSCC11 single chip board from Mekatronix.  The servos use code written by Drew Bagnell, another MIL student, and a custom serial interface.   The servos require fairly strict time dependent signals to operate correctly.  This is why a separate servo controller is used.  This also freed up program space on the main board for additional movements and behaviors.

### 4.3 Basic Movement

One of the primary goals for Jormungand is to have him move like a snake.  Serpentine movements are interesting and hard to duplicate on a robot.  Snakes are much more flexible than a robot.  Jormungand's movement is primarily an inch-worm style movement.  He lifts his tail, pulls it forward, sets it down, pushes back and then propagates the "wave" created in his body forward to the front of his body.  This pushes him forward at a slow but steady pace.  This movement, while not completely serpent-like, is like a scaled up version of how snakes move.  To turn, all of the left-right segments bend in the direction desired.  This has the effect of guiding the movement in the desired direction.  The scales on the snake's stomach maintain traction.  These allow the snake to slip forward, but keep him from moving backwards. (See Figure 4.)

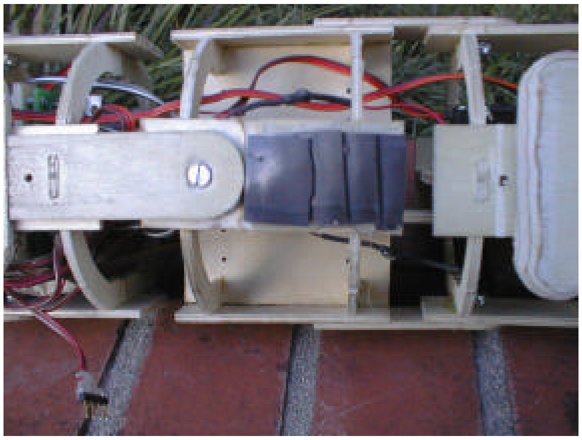

Figure 4. Bottom view showing "scales"

### 4.4 Sidewinder Movement

The sidewinder movement allows the snake to move straight sideways.  This is done by lifting the tail, pulling it to the left or right, setting it down and then moving up the body doing the same with the rest of the segments.

## 5. Sensors

### 5.1 Sensor Overview

The sensor array on Jormungand is small though complete.  The current sensor array consists of two IR emitter/detector pairs, three bump sensors, and two tilt sensors. The sensors provide sufficient input for interesting behaviors.

### 5.2 IR Sensors

The IR detector is a Sharp IR detector that has been hacked to provide basic range finding. The emitters are modulated at the 40kHz frequency required for the detectors. The IR detectors serve as eyes for Jormungand. Their range is approximately 2 feet and provides sufficient sensing for the snake to avoid obstacles. They are mounted on either side of the head looking forward and to the side and provide a good field of view. (See Figure 5.)

### 5.3 Bump Sensors

The bump switches are simple micro push buttons. The nose button causes the snake to become annoyed and rattle his tail. The nose button can be either held down by hand or activated if the snake moves straight into a wall, which occasionally happens. The cheek buttons cause the snake to initiate a sidewinder movement in the opposite direction. The bump switches were wired with a simple pull up scheme to provide the signal to the processor. (See Figure 4.)

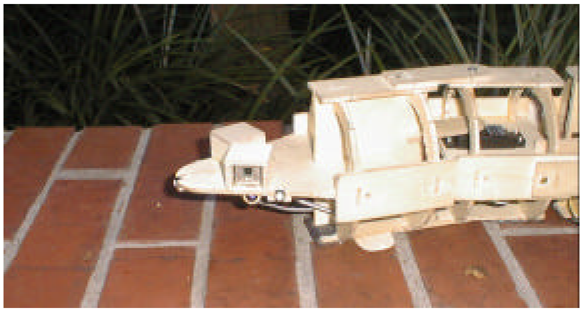

Figure 5. IR emitter/detectors and bump switches

### 5.4 Tilt Sensors

The tilt sensors are mercury switches. They are mounted at an angle inside one of the segments. They are set up so it is possible to tell which side the snake has rolled over on. When the snake tilts over, the mercury in one of the switches slides over onto the two contacts and closes the circuit. The snake then takes appropriate action. The tilt switches were wired up exactly like the bump switches, with a simple pull up scheme. (See Figure 6.)

## 6. Behaviors

### 6.1 Behavior Overview

The goal for the behaviors is to provide an interesting robot. The movement behaviors include an inchworm move as well as a sidewinder move. The additional behaviors are the self-calibration, obstacle avoidance, bump, and tilt-recovery behaviors.

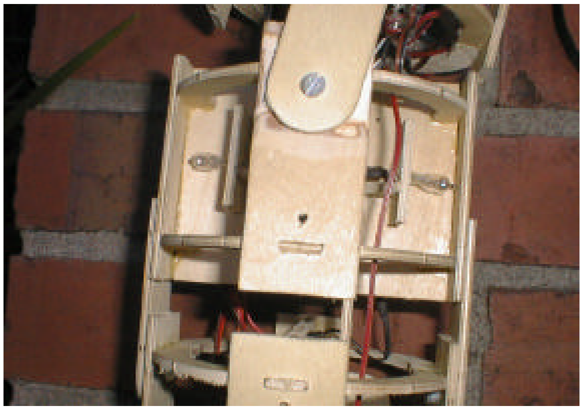

Figure 6. Tilt Sensors

### 6.2 Self-calibration

When the snake is turned on, the first action is a reading from the left IR sensors. If the snake’s left eye is placed a certain distance from a wall, that distance is as close to the wall it get before it wants to turn. If the snake is left out in the open, then as soon it detects a wall it will avoid it.

### 6.3 Obstacle Avoidance

The snake will avoid walls it sees with the IR sensors. As soon as the snake detects a wall it turns away. Then on the next check the snake will turn back if it no longer sees a wall, but at a rate 2/3rds that of the turning away. This minimizes the amount of jerking movement as the snake avoids obstacles. If there are walls on either side, the snake will try to maintain an equal distance from both walls.

### 6.4 Bump

When the nose button is pressed the snake will initiate its rattle. When a cheek button is pressed the snake will perform the sidewinder move.

### 6.5 Tilt Recovery

When the snake realizes it is not upright it initiates a series of moves to right itself. First it curls up a little bit and then uncurls quickly. This has the effect of tossing the robot around a little, and after a few tries generally gets the snake back upright.

## 7. Conclusion

### 7.1 Summary

Jormungand was all in all, a success. He accomplished my goal of a mobile, highly maneuverable robot. Jormungand is a rugged robot with room for expanded sensors and behaviors. His movements are more snake-like than I had hoped and he is quite mobile. His one deficiency, I feel, is his limited range of sensors. With increased sensing ability I think he could perform more actions and exhibit more complex behaviors. While, like everyone in the IMDL, I was unable to complete all I had hoped to do with the robot, I feel I have provided myself, and others who may wish to do so, a basis to build on.

### 7.2 Future Work

-- Replace brain board with a more complete board that has more programming space. I was extremely limited by the 2K EEPROM program space available on the 68'11-E2's
-- Current 42 oz-in servos are cheap but stronger servos means better movement
-- Bigger Batteries needed, not very much run time with current internal batteries (6 AA's)
-- Enlarge diameter to accommodate larger servos and C-cells

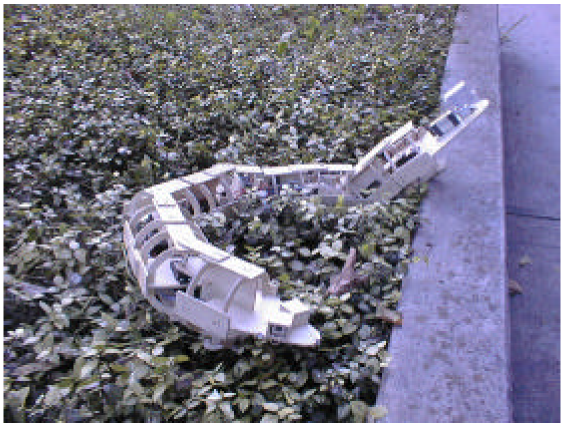

Figure 7. Snake in the grass

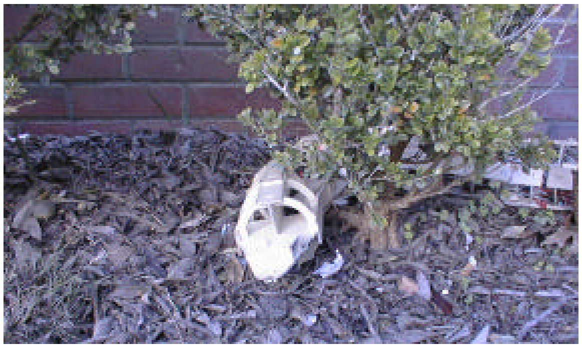

Figure 8. Snake in the bushes

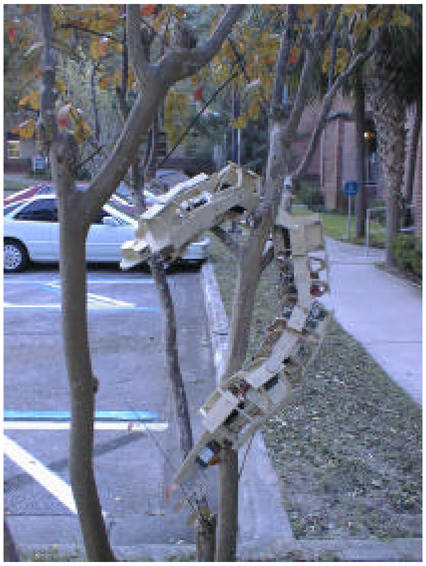

Figure 9. Snake in the tree