Joshua Tanenbaum
IAT 813 – Artificial Intelligence
Week 4 Assignment: Genetic Algorithms and Programming

Imagined System - Cat entertaining robot arm.
My cats love to chase and jump after fuzzy feathery things on the ends of strings. However, as I have learned through years of trying to captivate their attention, it is insufficient to randomly flip the string about in the hope that they will abandon all feline dignity and launch themselves wildly into the air. Indeed, eliciting a satisfactory reaction from my cats involves a diverse assortment of strategies, as well as patient observation and correlation of which string movements result in maximum cat amplitude (MCA). I propose that a genetic program has a good chance of determining an optimal set of automated toy behaviors over time.

Function Set
In order to make sense of the initial function set of the system we must first understand the environment in which it will be applied, and the mechanism by which it will be implemented. There are several components to this system:

- **Robotic Arm –** A full range of toy motions can be produced by a robotic arm with at least two axes of motion. Axis A is capable of pivoting 180 degrees, effectively describing an arc over the center point from one point on the floor to another. Axis B is capable of 360 degrees of motion, allowing the arm to rotate in a complete circle. Combined, these two axes of motion allow it to move the tip of the arm to any point in the dome of space above its base. [Figure 1]

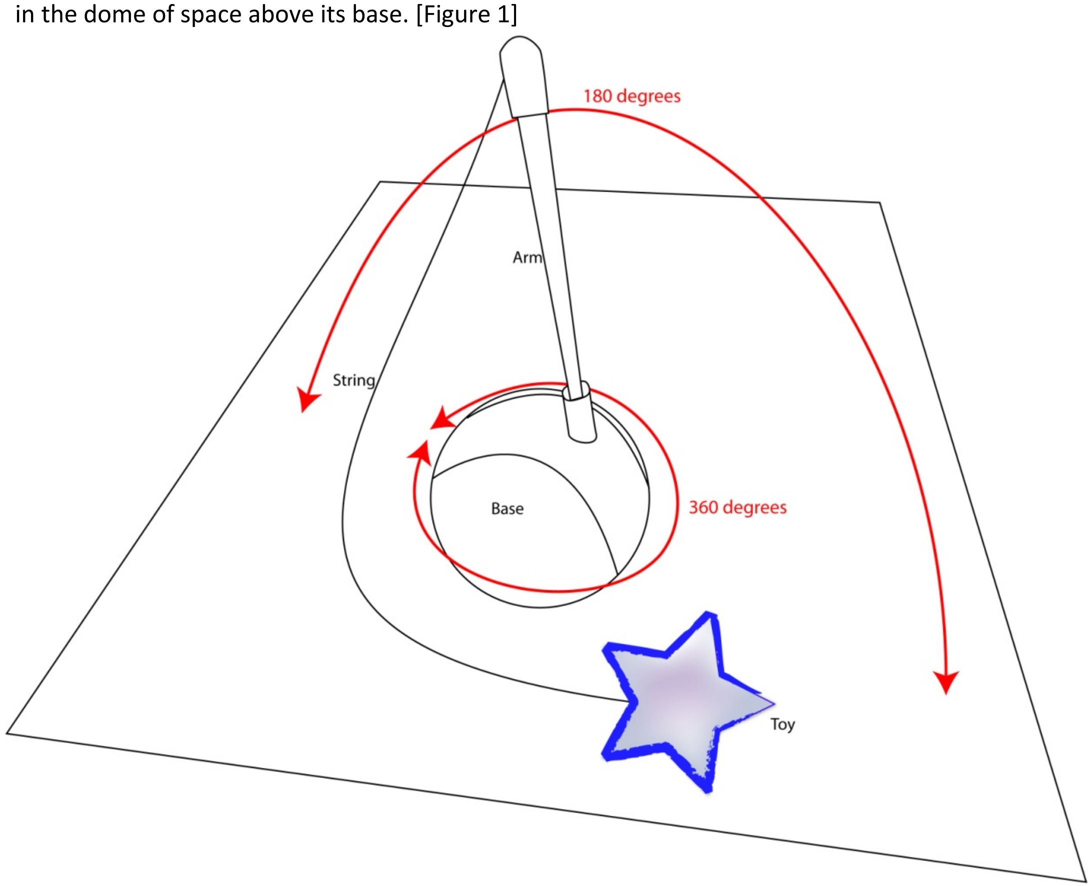

Figure 1 - The Robotic Arm

- **Sensors –** To perceive the behavior of the cats, we need to measure three vectors: contact with the toy (c), time spent airborne (t), and mean ground speed (s). To measure c, we will install a force sensitive resistor at the point where the string connects to the arm. Tension on the string will be translated into a change in voltage, which we can measure digitally. To measure *t* and *s* we will place the entire toy assembly on a simple pressure sensitive floor (using a grid of piezoelectric sensors). *t* will be measured as periods when no pressure is detected. *s* will be determined by sampling a number of floor sensor activations within a given period of time.

Having a basic understanding of the physical environment, we can now discuss the function set.

"Primitive" Functions

1. Rotate Axis B 5^o Left
2. Rotate Axis B 5^o Right
3. Elevate Axis A 5^o(relative to current position)[1]
4. Lower Axis A 5^o(relative to current position)

Compound Functions – These are functions comprised of combinations of primitive functions. Think of them as "macros".

5. "Jerking " - Elevate Axis A 5 ^{o} (relative to current position) and then return to previous position
6. "Flicking" - Rotate Axis B 10^o Left and then 10^o Right
7. "Low 360^o" – Lower Axis A to its lowest position and Revolve Axis B 360^o
8. "High 360^o" – Elevate Axis A 45^o and Revolve Axis B 360^o
9. "Flip Flop" – Reset Axis A to lowest point. Swing Axis A through a complete 180^o arc, and return to starting position

To create individuals to populate each generation, a function sequence chain of a set length (determined ahead of time) will be constructed. Generation size is also determined ahead of time. Because these functions operate in a linear sequential fashion, the notions of "parents", "children", and "generations" exist entirely as predefined heuristics (i.e. each individual is 8 steps long, each generation consists of 20 individuals, etc.)

Fitness Function (Maximum Cat Amplitude)

The fitness function for this system is intended to maximize *t* and *s* and to minimize *c*, that is, to get the cat in the air for the greatest amount of time without allowing the toy to be captured. For the sake of evaluating MCA we can assume that a cat in motion > a cat who has captured the toy, and a cat in the air > a cat on the ground. To guarantee that these are weighted appropriately, we describe *t* on a scale of 0 to 2.0, and *s* and *c* on a scale of 0 to 1.0, where 0 corresponds to a minimal measurement and 1.0 or 2.0 correspond to a theoretical maximum measurement.

Thus a *t* of 0 would be a cat in continuous contact with the ground and a *t* of 2.0 would be a cat that has taken a flying leap. For the sake of calibration, we can say that a *t* of 0.1 is equal to 1/10th of a second of time in the air, and so a *t* of 2.0 would be any leap in which the cat is airborne for 2 seconds or longer.[2]

Likewise, an *s* of 0 would correspond to a cat that is completely stationary, and an *s* of 1.0 would be a cat in fast motion. Our initial heuristic for *s* would make 0.1 equivalent to 1 foot of motion per second and 1.0 equivalent to 10 feet per second or faster.

Finally, a *c* of 0 would correspond to the toy moving freely through space, while a *c* of 1.0 would correspond to a toy with constant tension on it. Intermediary steps can be measured by ratio of tension to freedom within a period of 2 seconds where 0.1 corresponds to 2/10ths of a second of tension, aggregated across a 2 second sample.

Thus a very simple fitness function:

$$MCA = t + s - c$$

with a theoretical maximum possible fitness of 3.0[3].

---

[1] Clearly, some code preventing system from attempting to lower the arm below the level of the floor is needed in conjunction with these functions.

[2] These heuristics would need to fine tuned based on the performance of the individual cat.

Flow Chart and Some Predictions

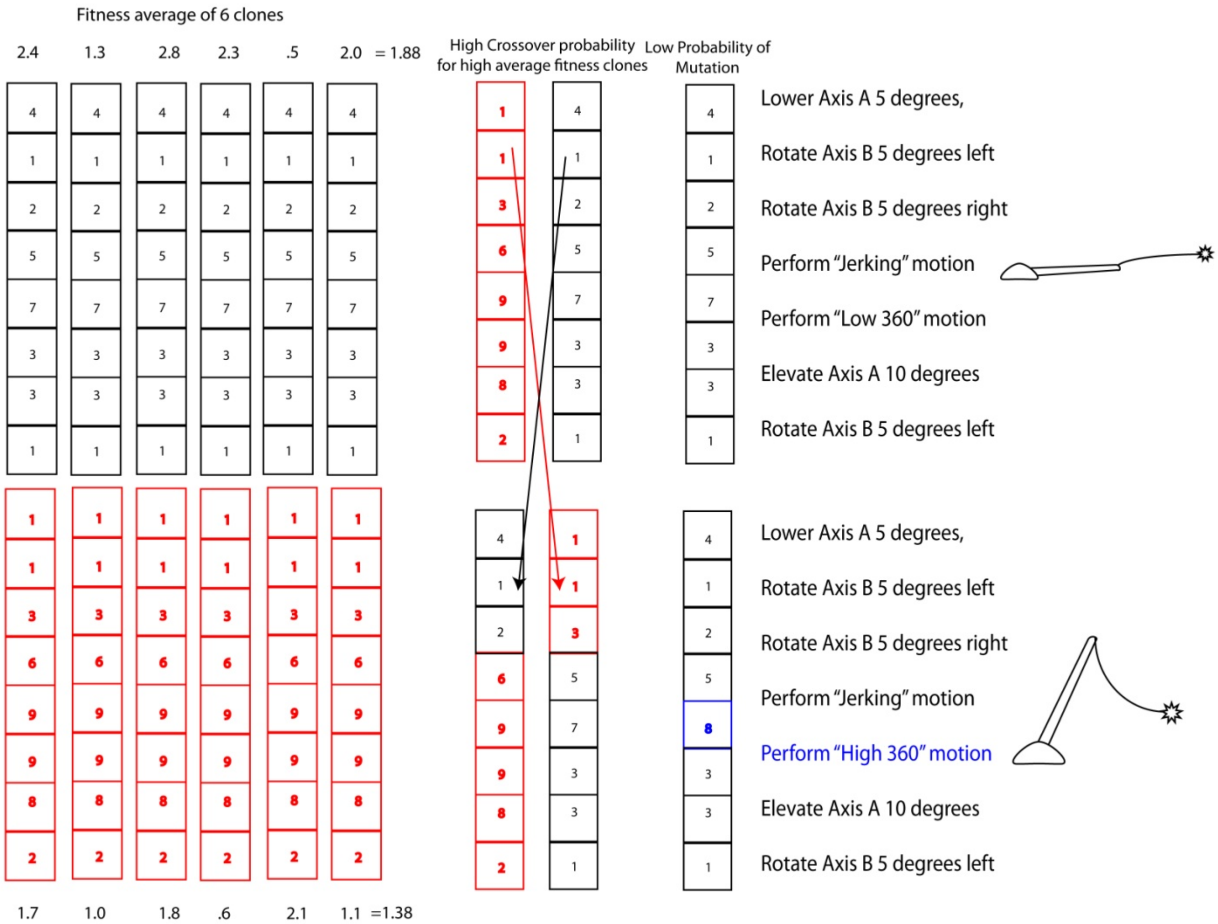

Figure 2

Figure 2 shows how this function set might be recombined into a series of motions, and then evolved towards a set of optimal gestures. We can make some predictions about what types of conditions will lead to an optimal learning environment for this algorithm. Cats are inherently stochastic. The amount of irregularity that the cat introduces into the system makes it much harder for our fitness function to assess the impact of a given maneuver at a given time. For example: perhaps the system executed an excellent gesture, but the cat was distracted and did not react, or perhaps the cat was reacting to some other external stimulus (such as a fly or bird) at the moment that the system executed a motion that ordinarily would not elicit significant response. In order to correct for the unpredictability of the cat, it seems prudent to clone a large number of samples from each generation, in order to maximize confidence over their aggregate fitness. In conjunction with this, I propose a system memory that averages the fitness measure of any group of clones in order to account for feline stocasticity. Mutation rates for this system will probably work best at low probabilities. Crossover probability increases as average fitness for any given family of clones increases.

The example given in Figure 2 is of individuals with 8 function steps. In order to develop sophisticated motion in this system, it might be desirable to string together many more motions than just 8. Experimentation is needed to determine what the optimal individual size is, as well as the optimal population size.

---

[3] This number is technically impossible to attain, due to the way that we are taking measurements. *t* and *s* measurements could conceivably cancel each other out, as a cat leaping a lot (thus attaining a high *t)* might not contact the ground enough to register as a high speed (*s*).