Training sensory-motor behavior in the connectome of an artificial C. elegans

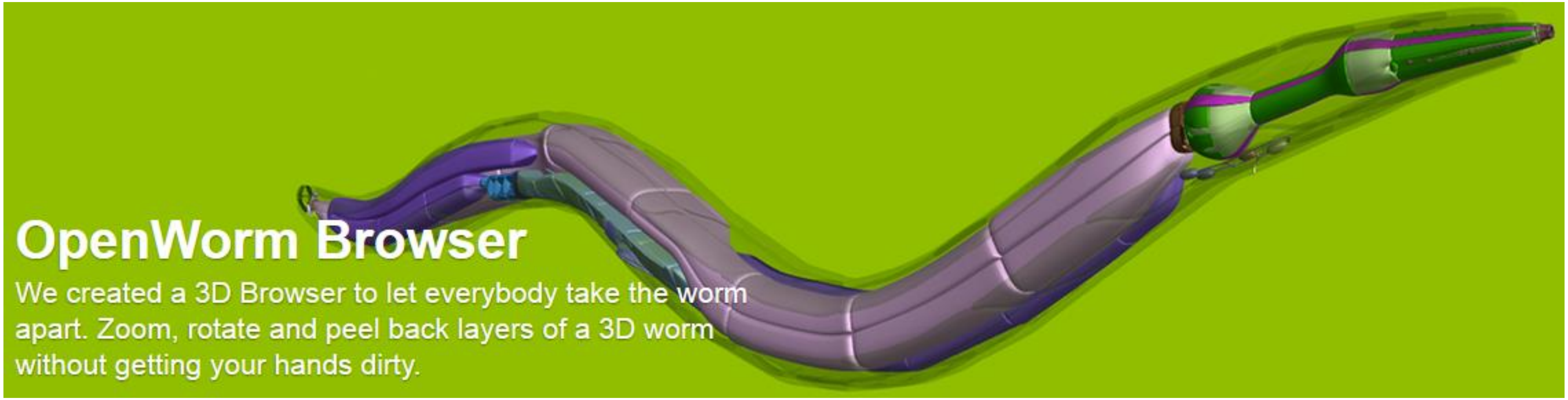
Tom Portegys, portegys@gmail.com

In association with the OpenWorm Project (openworm.org)

C. elegans and the OpenWorm Project

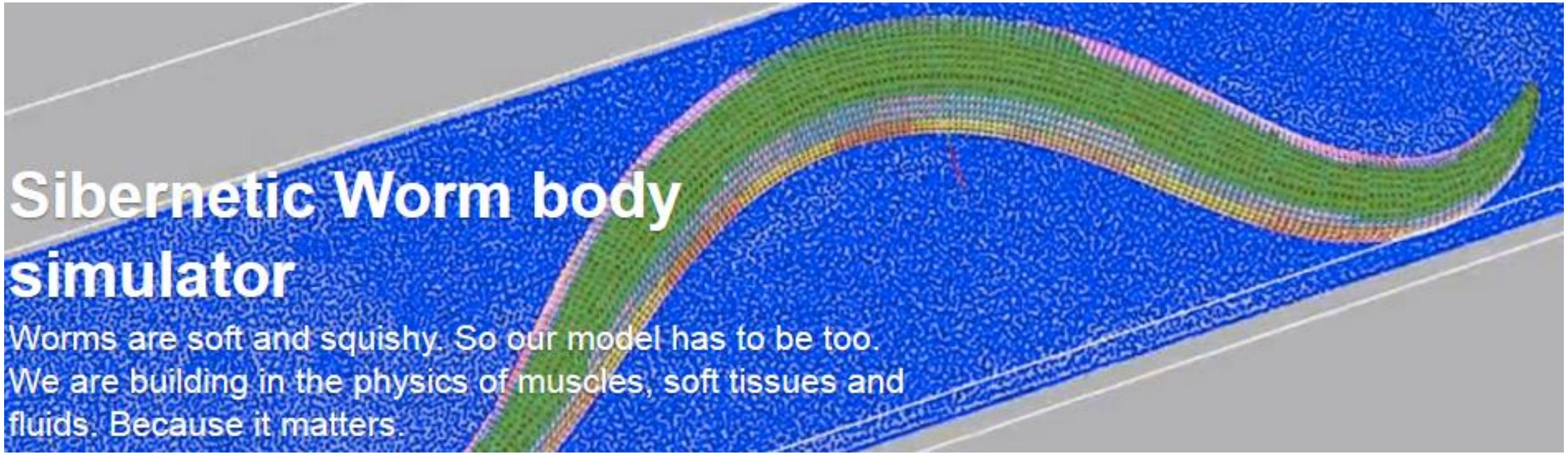
OpenWorm aims to build the first comprehensive computational model of the *Caenorhabditis elegans* (C. elegans), a 1mm transparent nematode worm. With only a thousand cells, it solves basic problems such as feeding, mate-finding and predator avoidance. Despite being extremely well studied in biology, this organism still eludes a deep, principled understanding of its biology.

C. elegans



OpenWorm Browser

We created a 3D Browser to let everybody take the worm apart. Zoom, rotate and peel back layers of a 3D worm without getting your hands dirty.



Sibernetic Worm body simulator

Worms are soft and squishy. So our model has to be too. We are building in the physics of muscles, soft tissues and fluids. Because it matters.

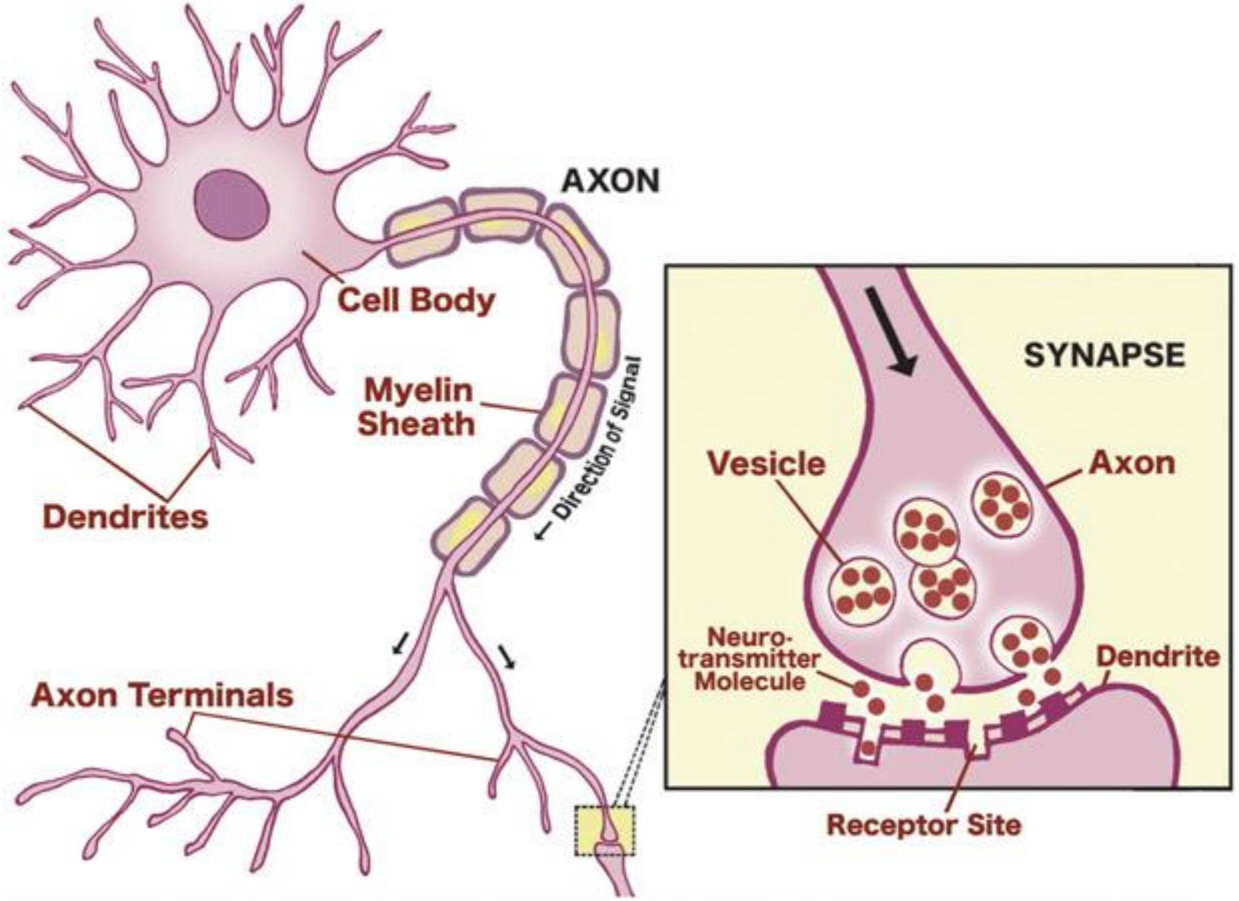
A connectome is a neural network wiring diagram

NeuroML Connectome

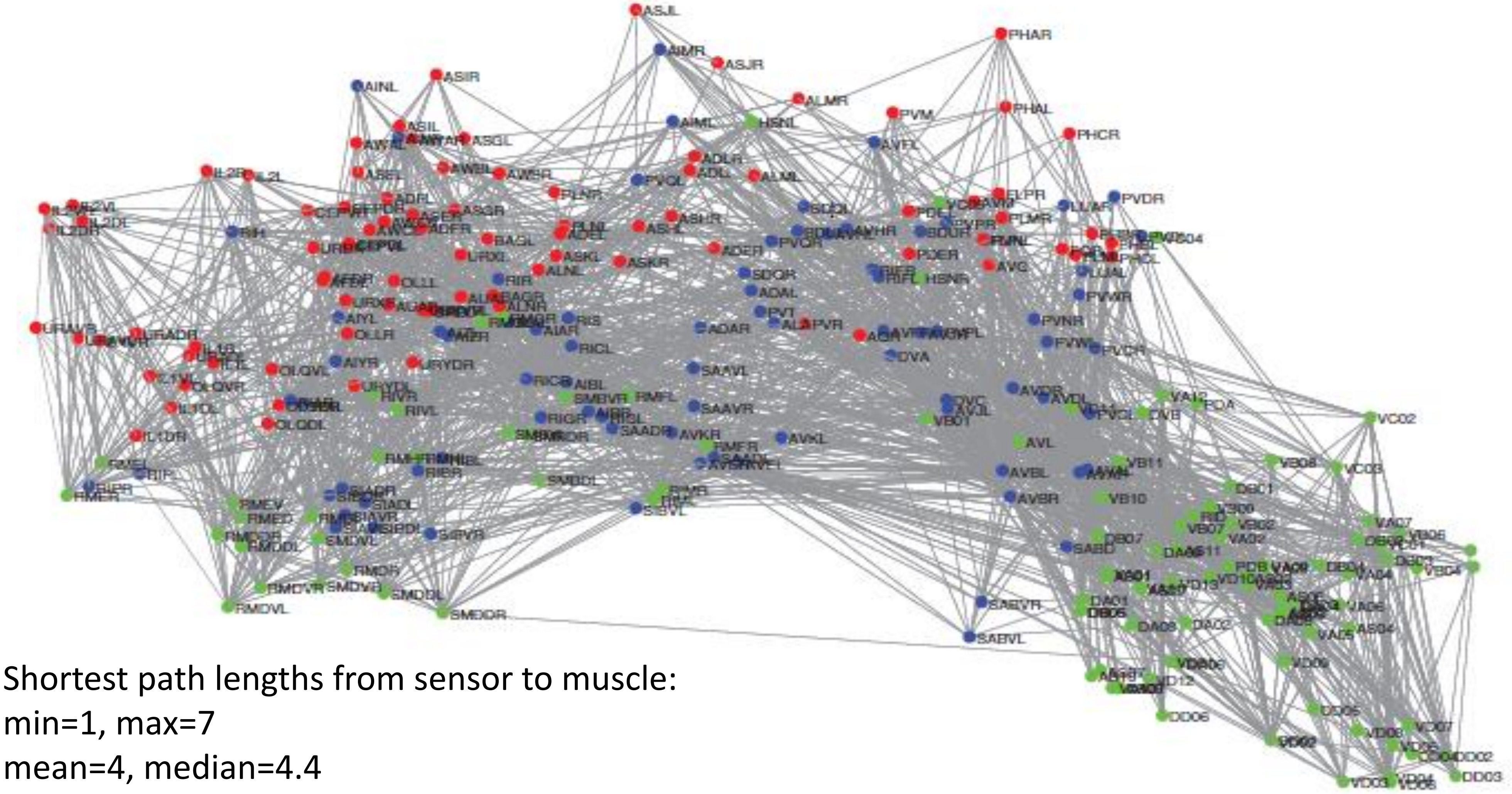
Every neuron the worm has, and how they connect.
Everything in its right place. Now we are bringing it to life.

C. elegans has exactly 302 neurons with 3000+ synaptic connections

Neuron and synapse



The C. elegans connectome graph
(from Bargmann and Marder)



Shortest path lengths from sensor to muscle:
min=1, max=7
mean=4, median=4.4

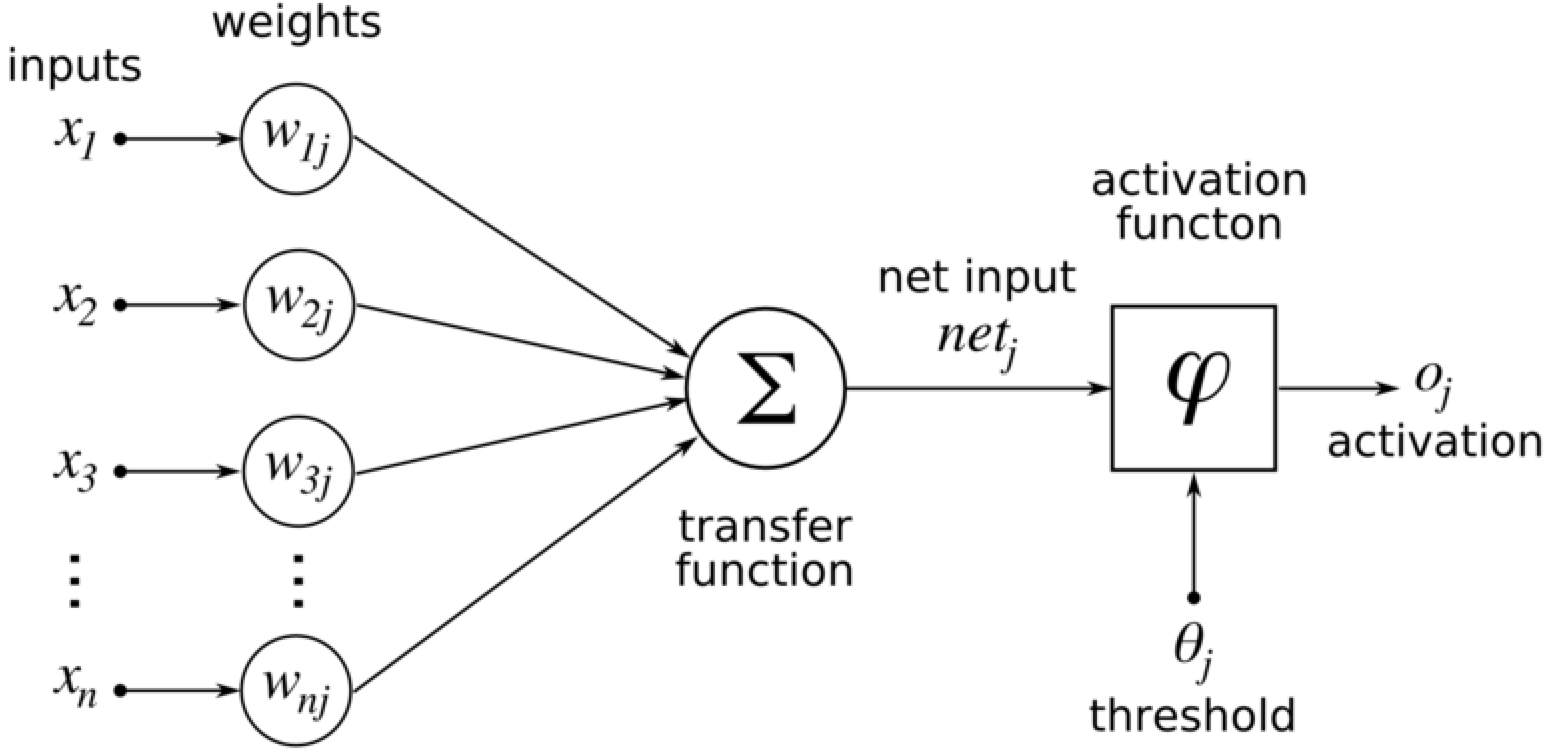
Background

- Although much about the C. elegans connectome is known, including the synaptic types, connectivity, and neurotransmitters, the functional behavior of synapses and neurons is largely unknown.
    - There are efforts to discover this, using luminescent neurons for example.
- Can an artificial C. elegans neural network be “programmed” to behave authentically? E.g. sensory-motor locomotion, etc.
- With 302 neurons and thousands of synapses, training the entire network is daunting; for this reason efforts either reduce, simplify, or augment the network to get it to perform.

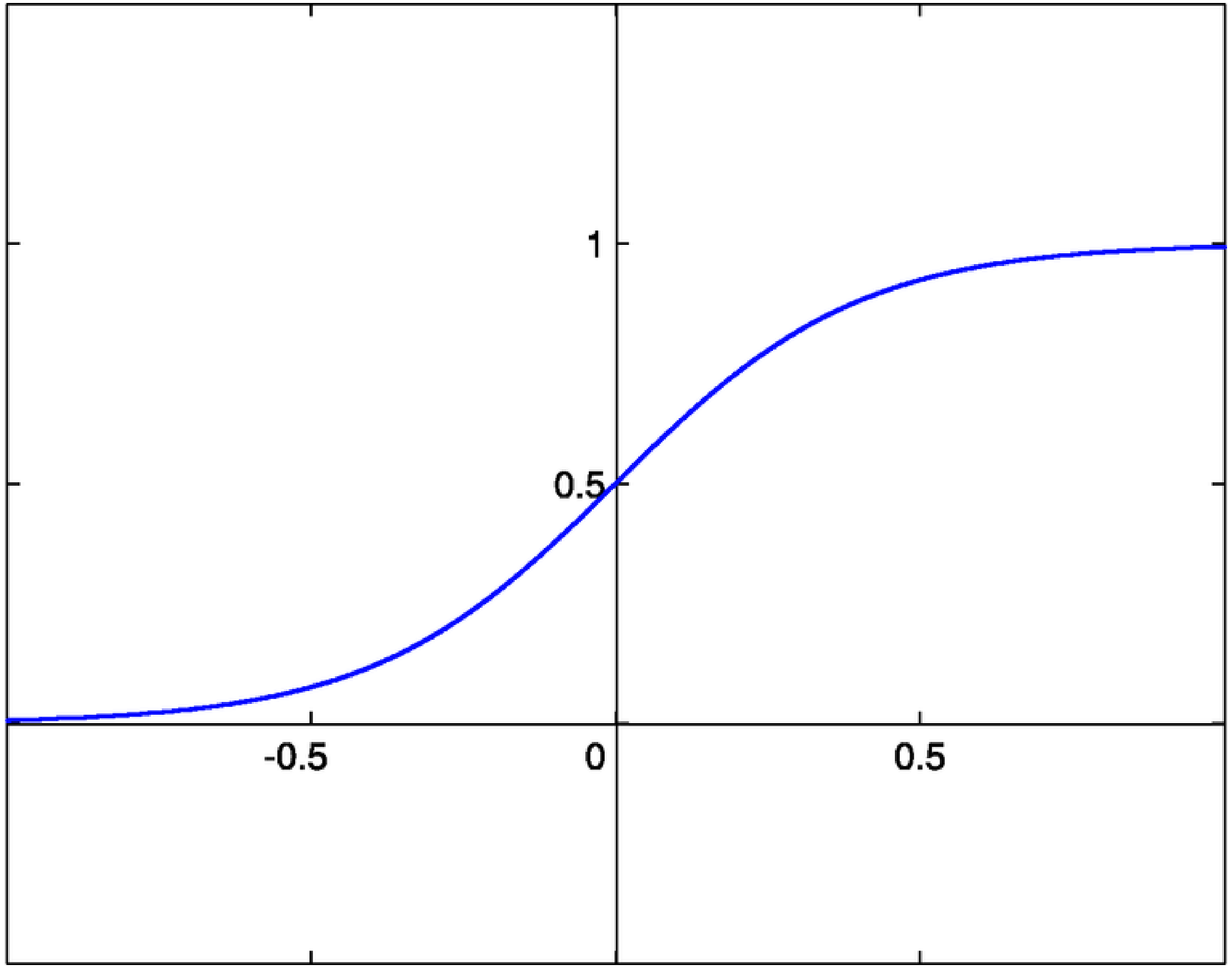
Tasks

- Task 1: using the C. elegans connectome, train the synapse weights to perform feasible input-output sequences.
- Task 2: train the connectome to react to the “light touch” stimulation by starting to undulate.
- Initial simplifications:
  - “Perceptron” (artificial neural network) neurons.
  - Single connection between neurons (many are multiple in connectome).
  - Gap-junction synapses behave the same as gap synapses.
  - GABA is the only inhibitory neurotransmitter
  - Synapse signals propagate uniformly in a step-by-step manner.

Neuron is activated by weighted sum of synaptic inputs



Neuron activation function: logistic sigmoid



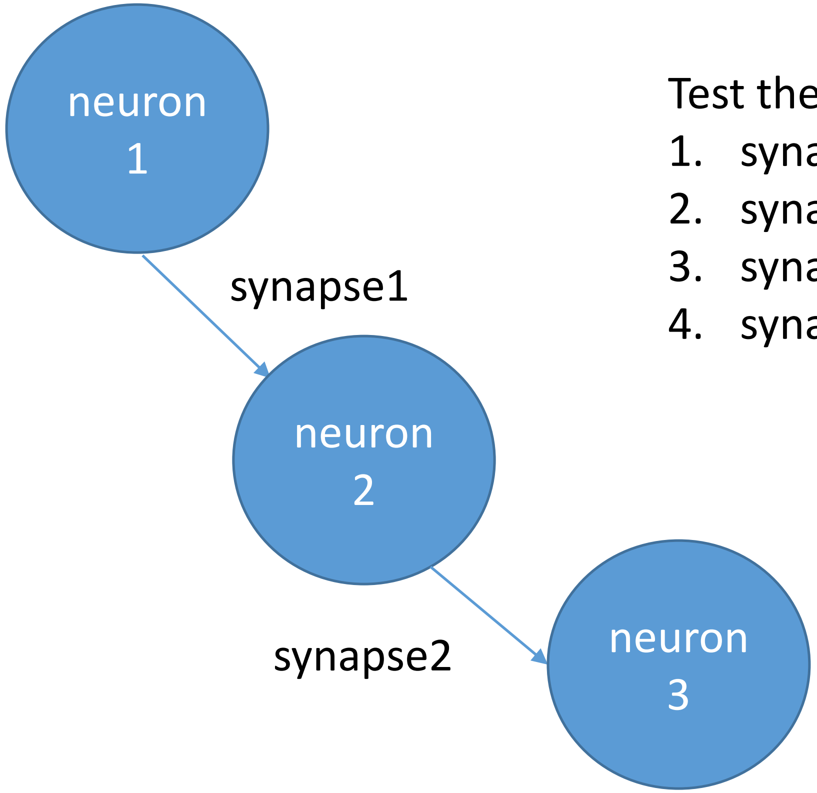
Basic scheme: Genetic algorithm

- In the computer science field of artificial intelligence, genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems.
    - Population members contain a set of “genes”, which are dimensions in search space.
    - Mutation randomly varies genes.
    - Crossover “mates” two parent population members, combining their genes into a child.
    - Survival is based on a “fitness” function.
    - Population evolves over generations.

Algorithm tweaks for C. elegans training

- Also uses “harmonization”, a hill-climbing technique that randomly selects synapse paths in the network to optimize as ensembles. Since C. elegans has relatively shallow paths between sensory and motor neurons, the idea is to optimize these as sets.
- The crossover function selects random neurons from parent networks to place into the child, but neurons connected to a selected neuron probabilistically “stick” to it. The effect is to transfer chains of optimized neighboring neurons into the child.

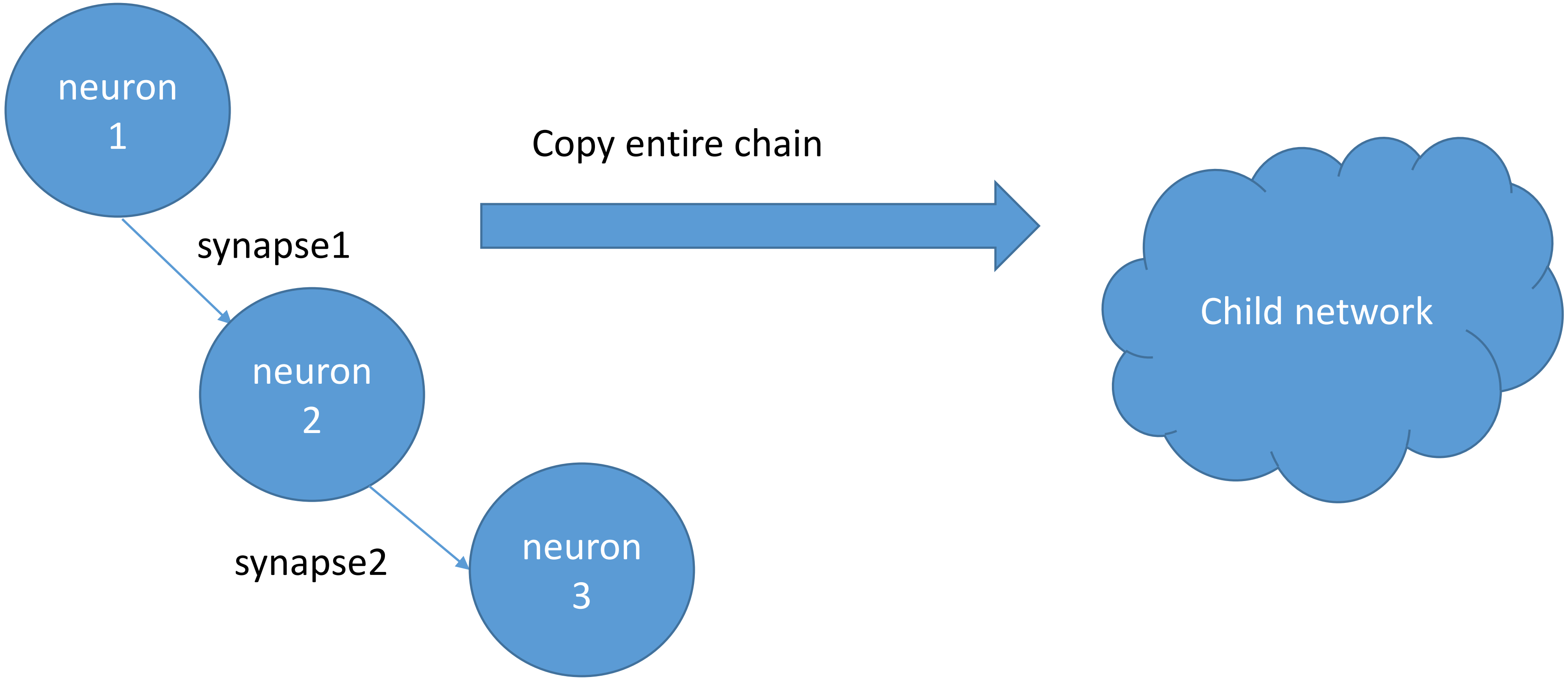
Harmonization: hill-climbing optimization



Test these combinations for fitness improvement:

1. synapse1 + random/synapse 2 + random
2. synapse1 + random/synapse 2 - random
3. synapse1 - random/synapse 2 + random
4. synapse1 – random/synapse 2 - random

Harmonization: crossover using optimized chains



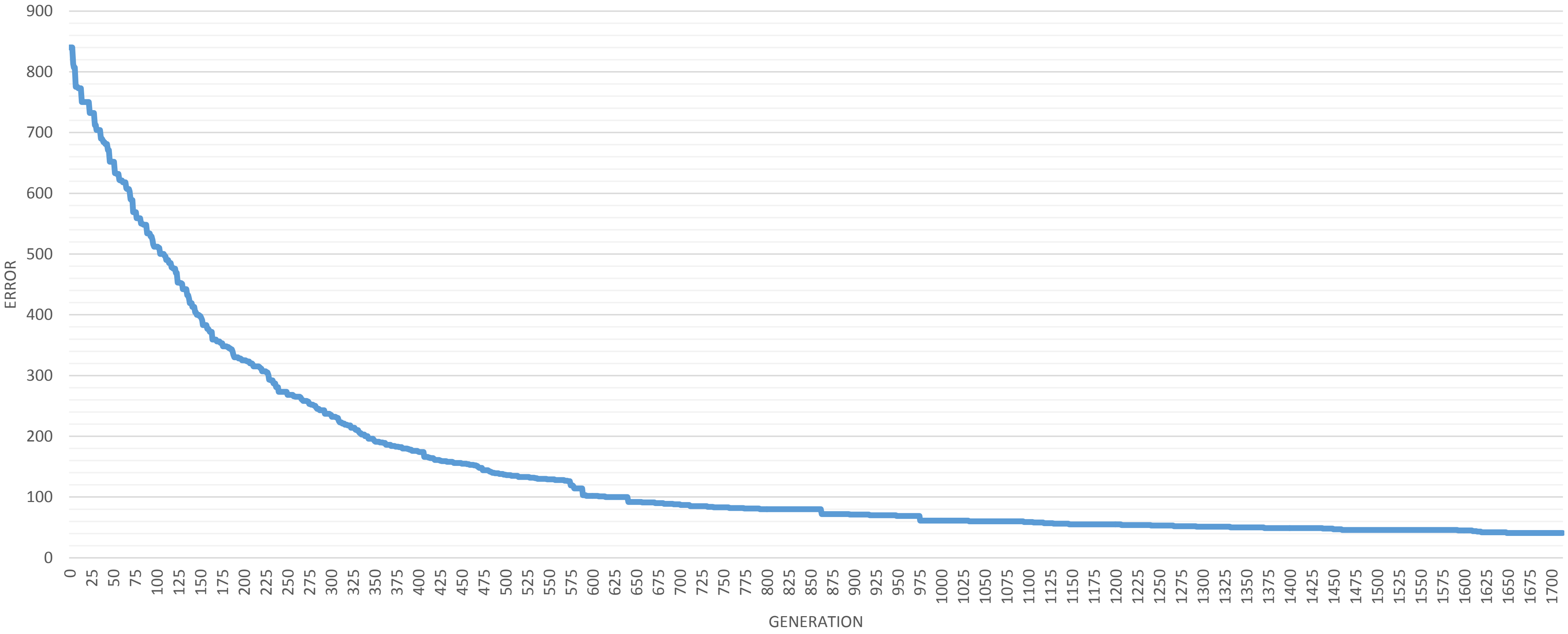
More training details

- Population size=500, offspring per generation=250
- Generations: variable for Task 1, 250 for Task 2.
- Random selection for crossover and optimization.
  - Crossover probability = 80%
  - Optimization consists of mutation and harmonization.
  - Mutation rate = 5%, randomizes synapse weights.
- Elitism: only fittest kept per generation.

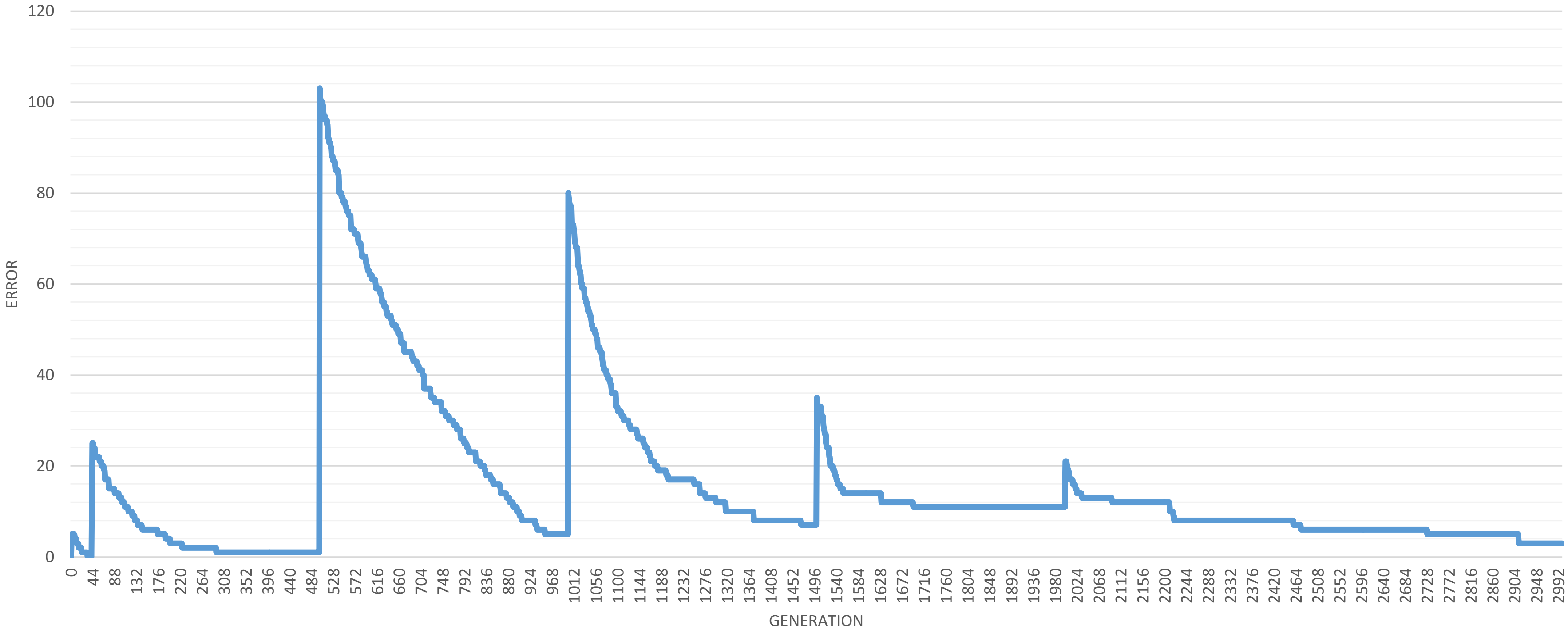
Task 1 method

1. Create a randomly weighted network from the connectome.
2. Feed either a random or cyclical input pattern sequences into the network and record the output sequences.
3. Train other randomly-weighted networks to perform the same input-output sequences.
4. Fitness is based on difference between outputs where an error is a difference of .05 or more.

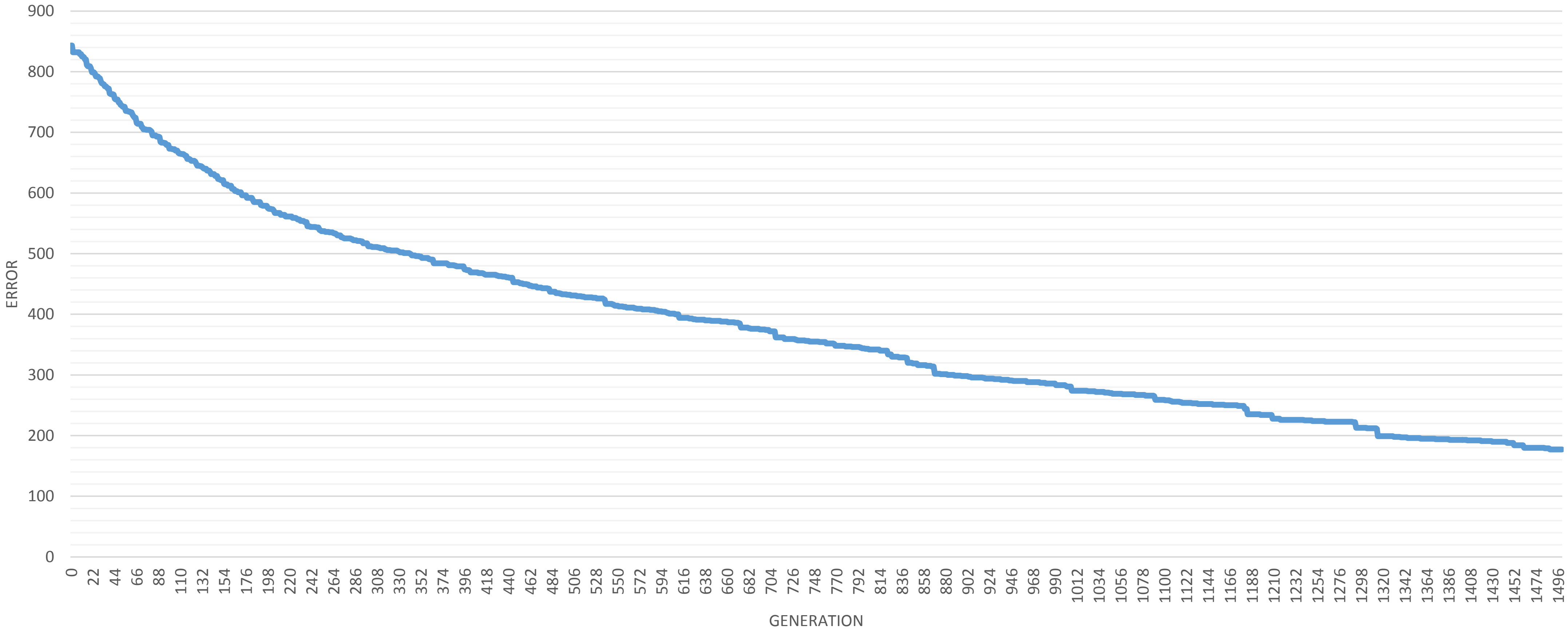
Simultaneous training of two ten-length sensory-motor sequences



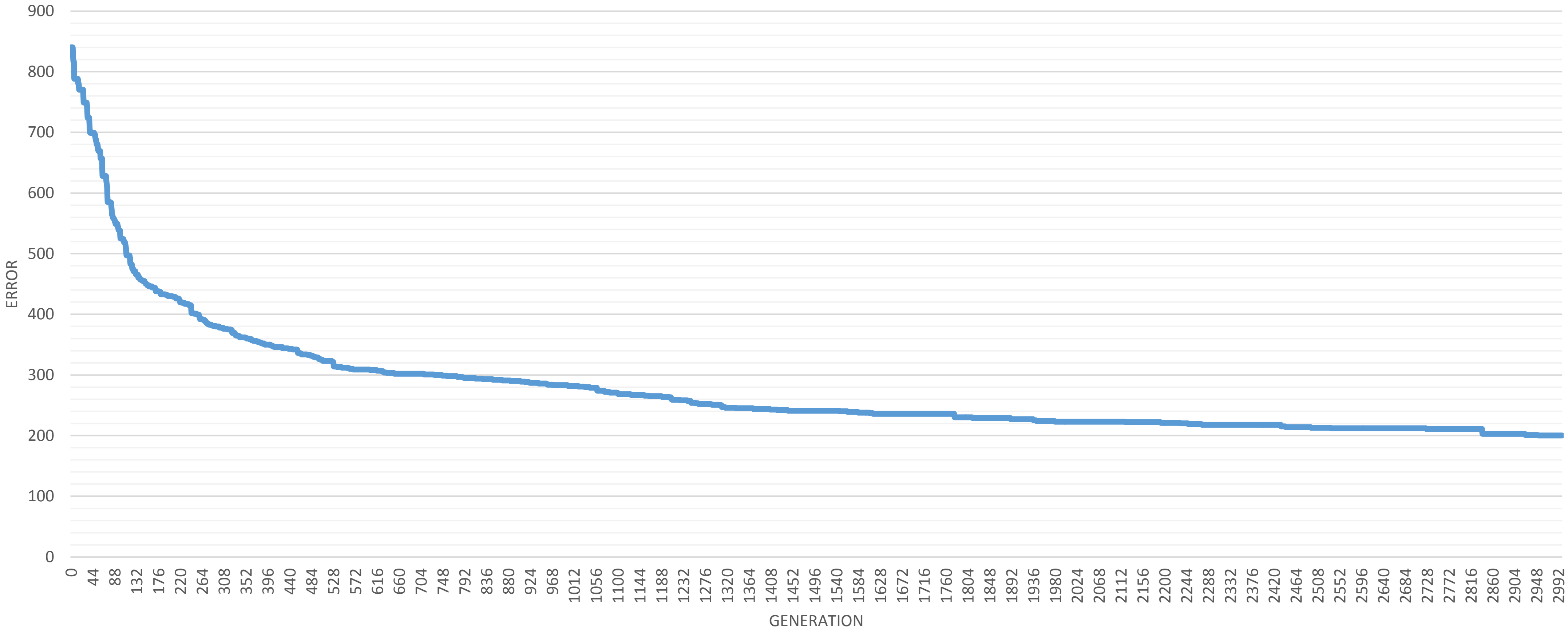
Step-by-step training of two ten-length sensory-motor sequences



Training with only harmonizing, no crossover or mutation



Training with only crossover and mutation, no harmonizing



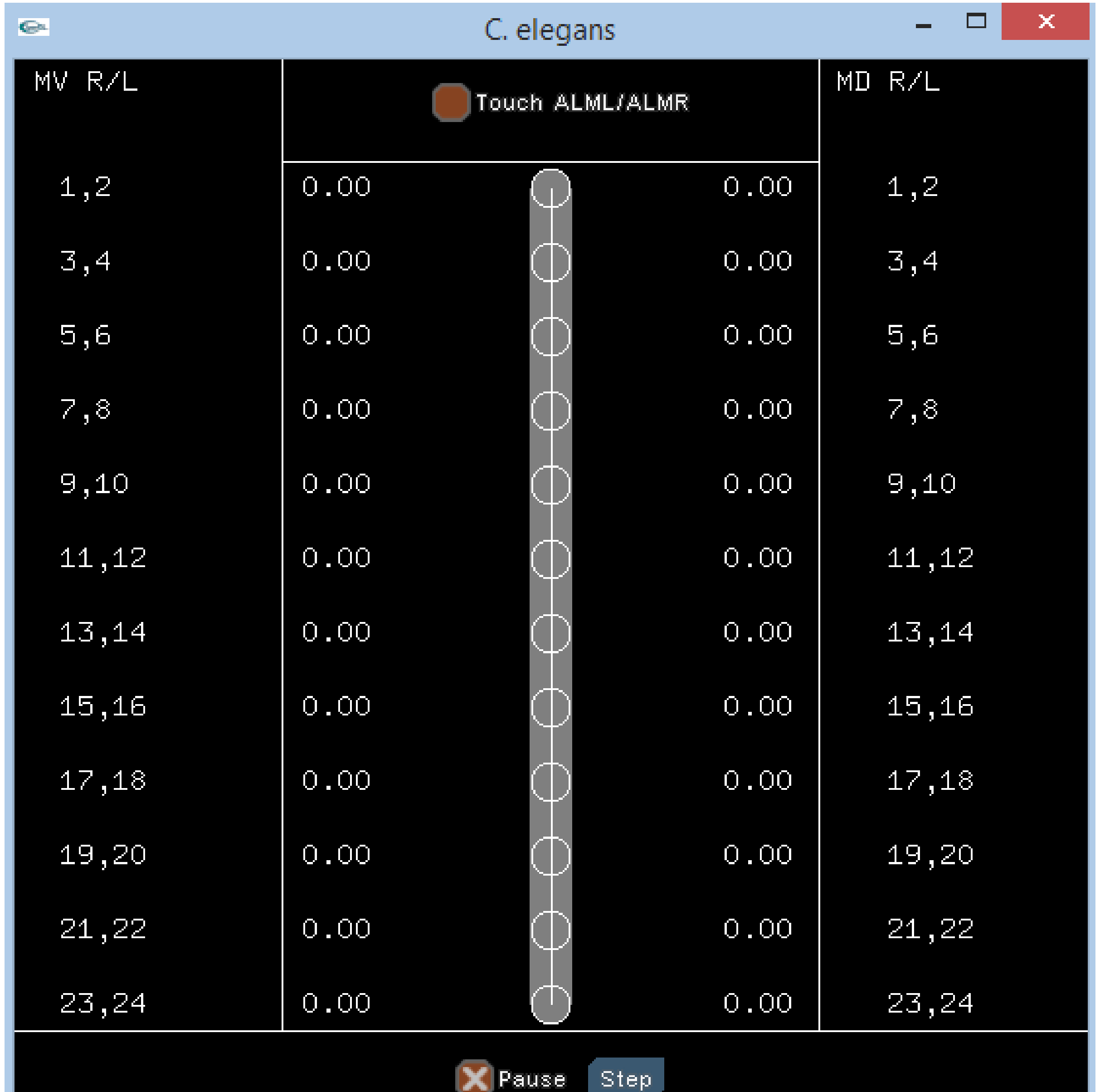
Task 2: light touch stimulation

- Create a randomly weighted network from the connectome.
- Turn off all but the ALML/ALMR light touch sensory neurons.
- Co-train with touch sensors off also to make sure it is responding to them.

Task 2, Part I: min/max fitness

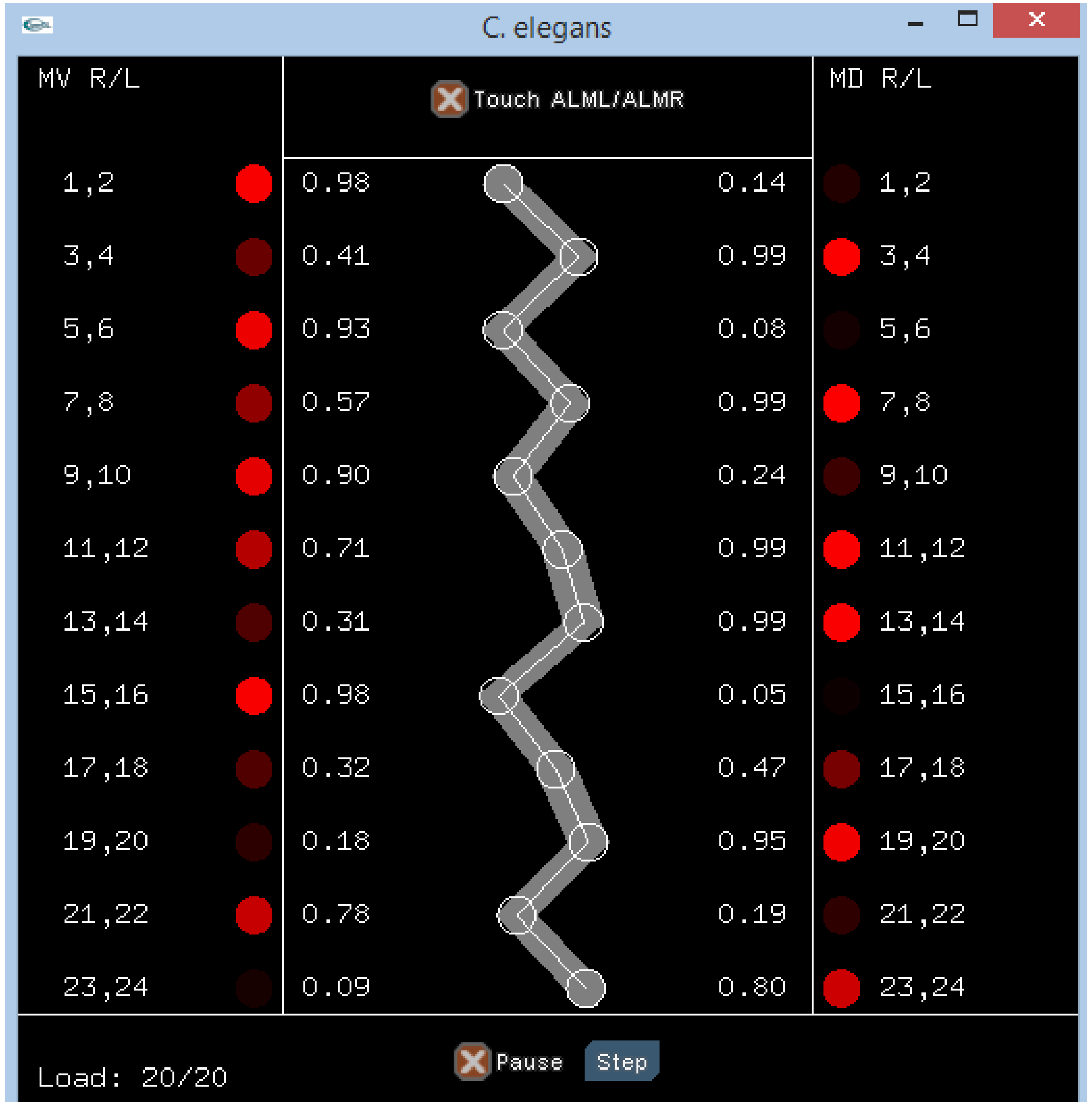
- Train with a fitness function that rewards the 12 body joint muscle groups (MDR/MDL, MVR/MVL) to maximize the number and magnitude of min/max muscle forces on the body.
    - This is admittedly a crude function. There will be a much better one in time with detailed measurements of the body dynamics.
    - It is possible that muscle stretch sensors might perpetuate the undulation once started. This model does not take that into account.

C. Elegans muscle groups (ventral/dorsal motion)



MV R/L	Touch ALML/ALMR		MD R/L
1,2	0.00	0.00	1,2
3,4	0.00	0.00	3,4
5,6	0.00	0.00	5,6
7,8	0.00	0.00	7,8
9,10	0.00	0.00	9,10
11,12	0.00	0.00	11,12
13,14	0.00	0.00	13,14
15,16	0.00	0.00	15,16
17,18	0.00	0.00	17,18
19,20	0.00	0.00	19,20
21,22	0.00	0.00	21,22
23,24	0.00	0.00	23,24

Light touch undulation after training



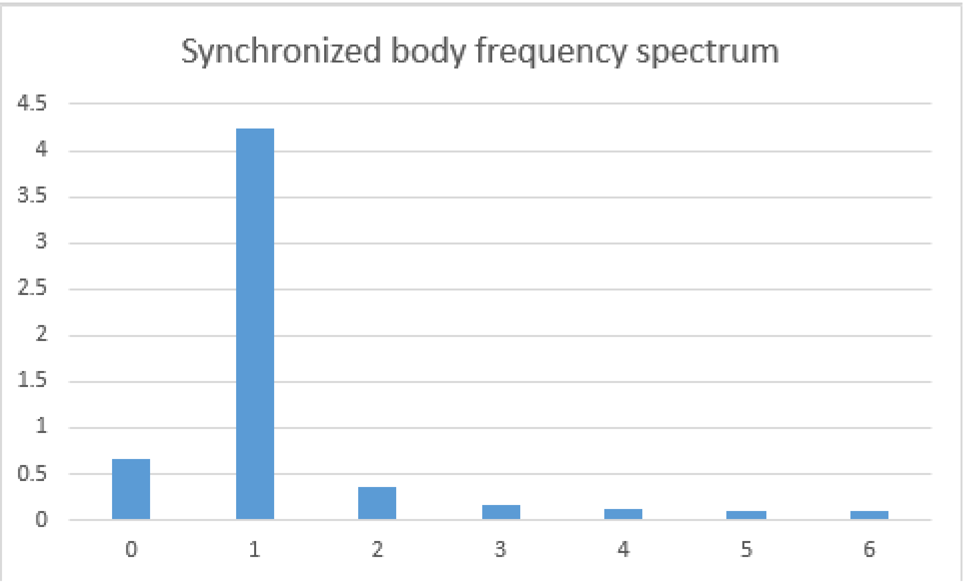
MV R/L			MD R/L
1,2	0.98	0.14	1,2
3,4	0.41	0.99	3,4
5,6	0.93	0.08	5,6
7,8	0.57	0.99	7,8
9,10	0.90	0.24	9,10
11,12	0.71	0.99	11,12
13,14	0.31	0.99	13,14
15,16	0.98	0.05	15,16
17,18	0.32	0.47	17,18
19,20	0.18	0.95	19,20
21,22	0.78	0.19	21,22
23,24	0.09	0.80	23,24

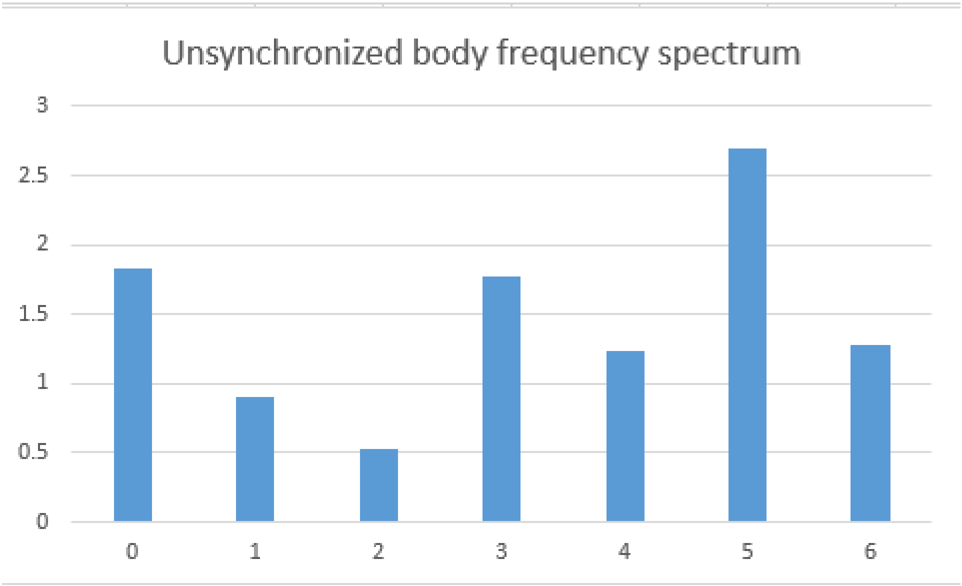
Load: 20/20

Task 2, Part II: Fourier transform fitness

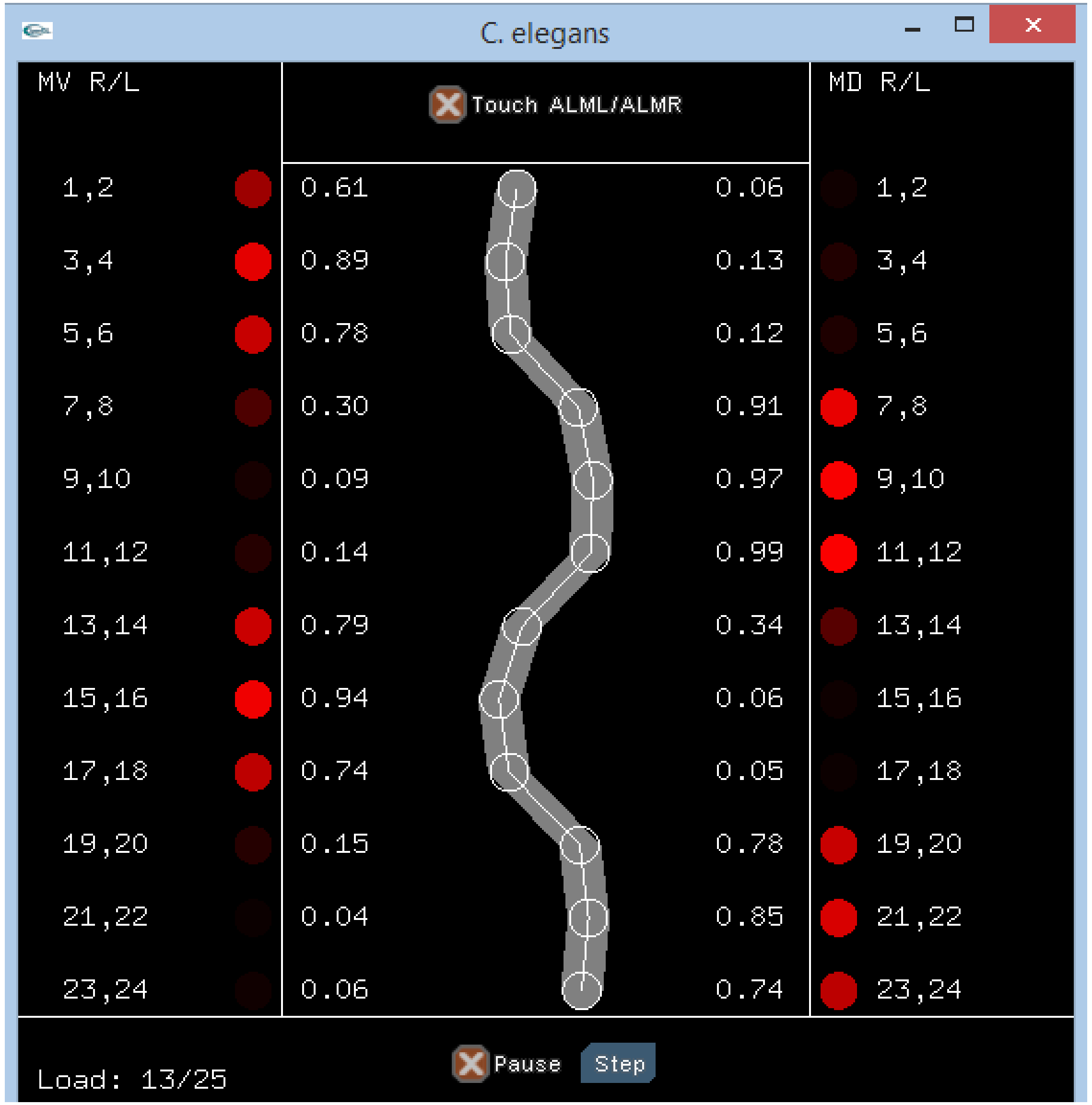
- Train with a fitness function that rewards the 12 body joint muscle groups (MDR/MDL, MVR/MVL) to produce smooth undulation.
- Smooth undulation will have frequency spectrum spike instead of distributed spectrum.

Fitter (left) and less fit (right) frequency spectrums





After training



MV R/L			MD R/L
1,2	0.61	0.06	1,2
3,4	0.89	0.13	3,4
5,6	0.78	0.12	5,6
7,8	0.30	0.91	7,8
9,10	0.09	0.97	9,10
11,12	0.14	0.99	11,12
13,14	0.79	0.34	13,14
15,16	0.94	0.06	15,16
17,18	0.74	0.05	17,18
19,20	0.15	0.78	19,20
21,22	0.04	0.85	21,22
23,24	0.06	0.74	23,24

Load: 13/25

Summary

- That the algorithm trains the entire connectome is unique, to my knowledge. Other efforts have simplified the connectome to reduce complexity.
- The algorithm’s “secret sauce” is the harmonization technique that trains ensembles of neurons together. Without it results are poor.

Future work

- The algorithm only trains to whatever is defined as fitness. This suggests that with higher fidelity networks and more realistic fitness functions, more closely simulated behavior might be possible.
- Focusing on locomotion behavior seems like a good area since the muscle model now exists and detailed movement measurements are being captured.
- Porting has begun to use a higher fidelity biological model using the NeuroConstruct/Neuron tools and the NeuroML language.

Nuts and bolts

- 10,000+ lines of threaded C++, Java, bash/bat
  - Separate program to read connectome spread sheet and create C elegans network.
  - Separate OGL program to view, create, and replay light touch behavior.
  - Builds/runs on Windows (VS) and *nix (make)
  - Uses FFTW3 Fourier transform package (www.fftw.org)
- Code at github.com/portegys/bionet
- Computing platform: NSF/XSEDE “Blacklight” shared memory computer (4096 cores).

References

- “A model of motor control of the nematode C. elegans with neuronal circuits”, Suzuki et. al., 2005
- “Proprioceptive Coupling within Motor Neurons Drives C. elegans Forward Locomotion”, Q. Wen, et. al., 2012.
- “A 3D undulatory locomotion model inspired by C. elegans through DNN approach”, X. Deng and J-X Xu, 2013
- “From the connectome to brain function”, C. Bargmann and E. Marder, 2013.