Fido: A Universal Robot Control System using Reinforcement Learning with Limited Feedback

Joshua Gruenstein Michael Truell

Horace Mann School

Control System Objectives

Fido was created to fulfill the following goals:

- **Trainability**: Allow both human and autonomous training rather than reprogramming
- **Universality**: Run on any robot, even without prior knowledge of the host

These goals were achieved through the training of artificial neural networks with a wire-fitted moving least squares interpolator following the Q-learning reinforcement algorithm and an action selection policy that utilizes a Boltzmann distribution of probability.

Implementation

Fido was programmed in C++, with no external dependencies. However, the simulator does use the SFML graphics library. The hardware implementation uses the Intel Edison embedded platform, a 3D printed chassis and a differential drive system.

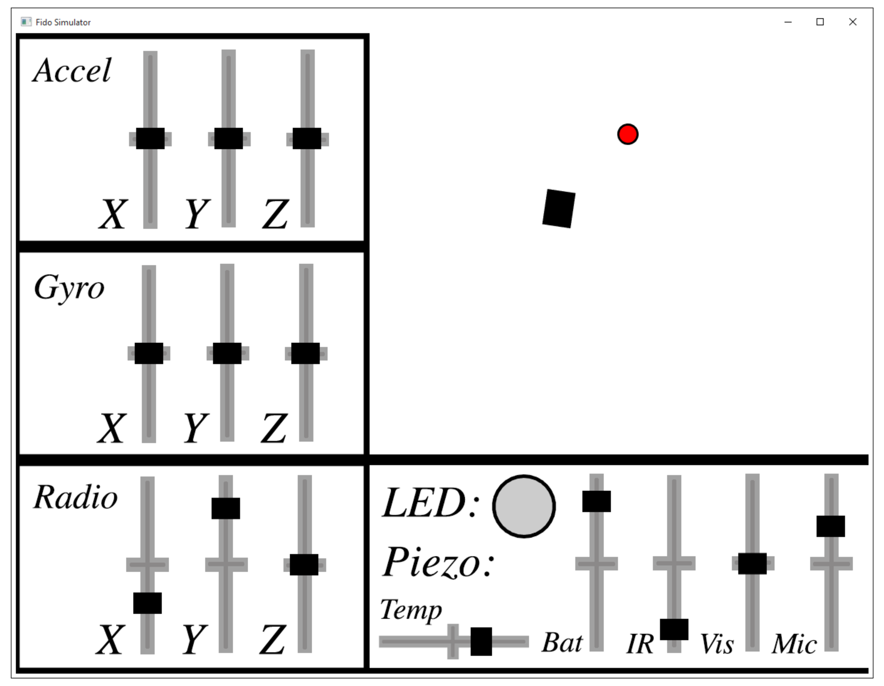

Figure 1: Fido Simulator Graphical User Interface

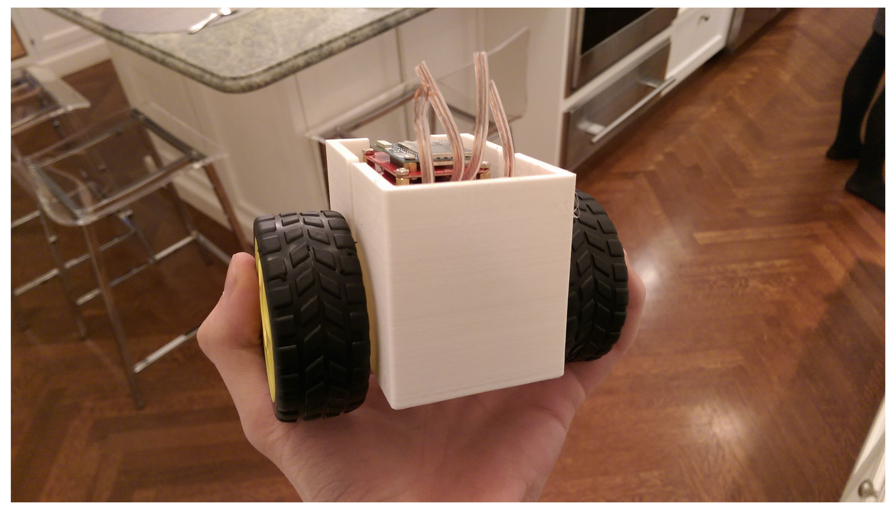

Figure 2: Fido Hardware Implementation

Learning Algorithm

- From a macro perspective, Fido can be viewed as a "black box:" inputs go in and outputs go out, Fido must optimize the relationship of inputs to outputs to maximize reward
- **Reward system:** Trying to determine the expected reward for an action in a given state based on past reward received
  - Must have a scalable, performance-optimized way of storing past state-reward sets and detecting patterns
  - Cannot just pick the action with the greatest expected reward: must "explore" to be trainable and re-trainable

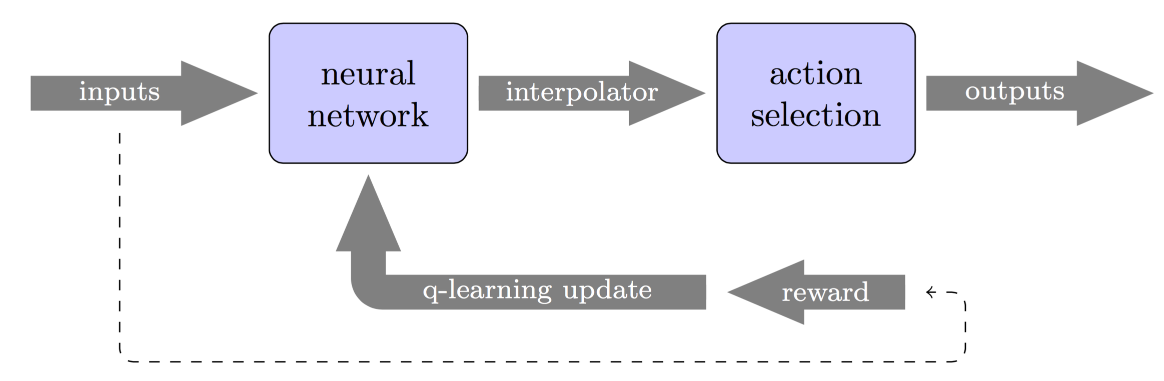

Figure 3: Control System Diagram

Reinforcement Learning

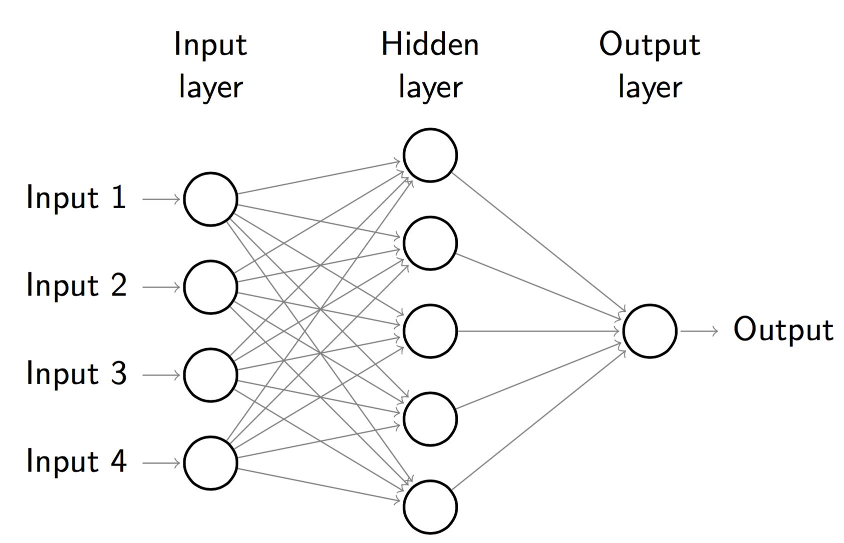

Figure 4: Single Output Feed-forward Neural Network

Action Selection Policy

Boltzmann stuff

Results

Results were gathered both in simulation and from the hardware implementation for a variety of tasks. In simulation Fido was evaluated setting an LED proportionally to light intensity ("Flash"), driving to a point with a direct XY-control drive system ("Float") and a differential drive system ("Drive"), and line following.

Task	Learning Iterations	Action Selection (ms)	Training Time (ms)
Flash	6	0.	6
Float	14	1	6
Drive	17	1	11
Line Follow	21	2	10.

Table 1: Number of Learning Iterations, Action Selection Time, and Training Time Per Iteration for Fido Simulation Tasks

In hardware Fido was tasked with staying still and driving to a point.

Task	Learning Iterations	Action Selection (ms)	Training Time (ms)
Stay Still	3	1	43.5
Drive to Point	18	4	65

Table 2: Number of Learning Iterations, Action Selection Time, and Training Time Per Iteration for Fido Hardware Tasks

Future Development

We would like to experiment with dynamic optimization of hyperparameters, changing factors such as neural network architecture and Boltzman temperature constant to best fit the task at hand. We also plan to package Fido as a machine learning library for embedded electronics and robotics, and build a microcontroller-based hardware implementation to further optimize for resource-limited environments.

References

[1] C. Gaskett, D. Wettergreen, and A. Zelinsky, "Q-learning in continuous state and action spaces," pp. 417–428, 1999.

[2] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," 1974.

[3] S. Dini and M. Serrano, "Combining q-learning with artificial neural networks in an adaptive light seeking robot," 2012.

[4] C. MacLeod, "An introduction to practical neural networks and genetic algorithms for engineers and scientists," Robert Gordon University, Tech. Rep., 2010.

[5] D. S. Kim and A. J. Papagelis, "Multi-layer perceptron: Artificial neural networks," http://www.cse.unsw.edu.au/~cs9417ml/MLP2/.

[6] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. New York, NY, USA: Cambridge University Press, 2000.

[7] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge England, 1989.

[8] G. A. Rummery, "Problem solving with reinforcement learning," Ph.D. dissertation, University of Cambridge Ph. D. dissertation, 1995.

[9] L. C. Baird and A. H. Klopf, "Reinforcement learning with high-dimensional, continuous actions," *Wright Laboratory, Wright-Patterson Air Force Base, Tech. Rep. WL-TR-93-1147*, 1993.

[10] G. Biggs and B. MacDonald, "A survey of robot programming systems," pp. 1–3, 2003.

Acknowledgements

Thank you to Dr. Jeff Weitz of Horace Mann School, who gave us guidance in the art of scientific research.