Paper Title

Joshua Gruenstein and Michael Truell

September 11, 2015

Abstract

Our abstract.

1 Introduction

Introduction here.

2 Software Design
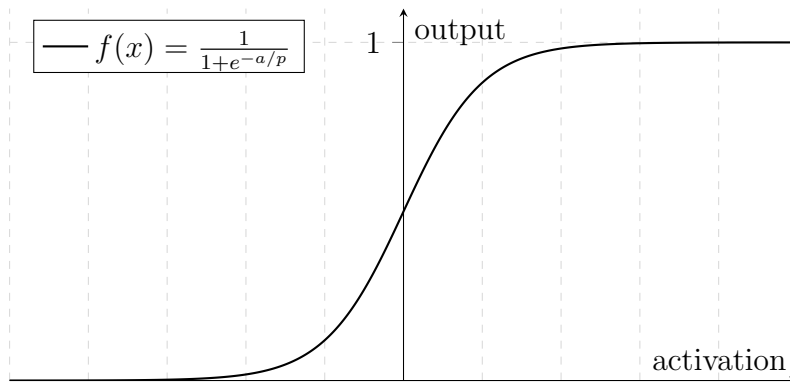
Software design goes here.
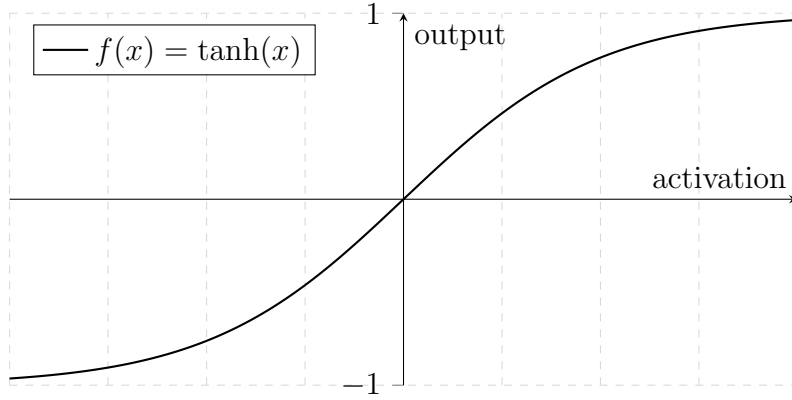

Figure 1: Sigmoid Function Graph

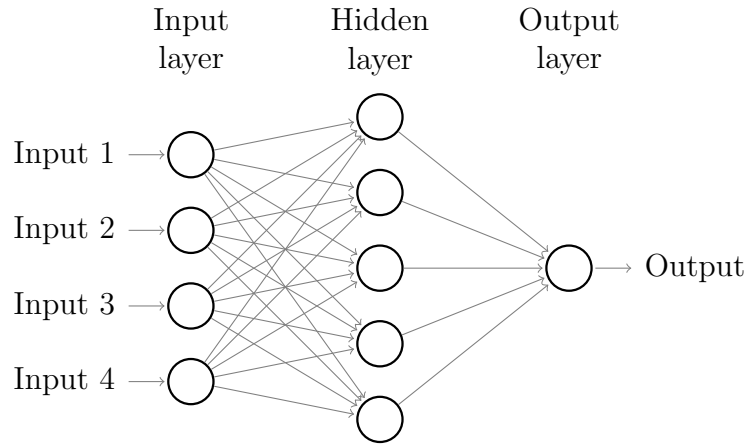Figure 2: Hyperbolic Tangent Function Graph

Figure 3: Single Output Feedforward Network

2.1 Neuron Architecture

2.2 Learning Experimentation

2.3 Q Learning Implementation

2.4 Design and Simulation

3 Physical Design

The hardware for the Fido artificial intelligence was chosen using three main parameters. Firstly, the electrical components had to support and compliment the software library; it would be impractical to refactor the entire code-base.

Algorithm 1 FizzBuzz Algorithm

1: **for** each integer i 1 to 100 **do**
2:     **if** $15 \mid i$ **then**
3:         print "FizzBuzz"
4:     **else if** $3 \mid i$ **then**
5:         print "Fizz"
6:     **else if** $5 \mid i$ **then**
7:         print "Buzz"
8:     **else**
9:         print i
10:     **end if**
11: **end for**

Secondly, the hardware had to be easily trainable and debuggable. Lastly the sensors and design chosen had to facilitate the concept of natural learning, modeling after nature to some degree.

3.1   Preliminary Decisions

The Teensy 3.1 microcontroller development system was chosen as the host platform for Fido's hardware implementation. As the board already has a USB bootloader and an open source software toolchain, it allowed for rapid prototyping of the hardware and software. The core microcontroller of the Teensy sports an ARM 32 bit, 72 MHz Cortex-M4 architecture, giving Fido necessary computational power. While a microcontroller with an integrated floating point unit may have sped up the numerous floating point multiplication operations involved in neural network propagation, it was decided unnecessary for this prototype due to time and cost considerations.
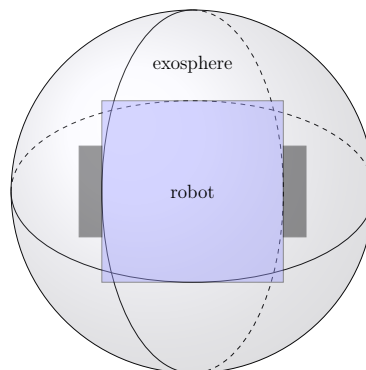

Figure 4: Differential Drive Robot in Exosphere

As Fido relies on inputs and outputs in order to learn and advance its neural networks, the selection of sensors and other peripherals was considered carefully. The sensor outputs needed to be easily modified by a human for training to be feasible: an example is a light sensor being easier to control than a radiation or temperature sensor. An obvious sensor choice was a microphone, allowing Fido to respond differently to sounds of various magnitude and frequency by passing in a sound wave. Another sensor choice was infrared and visible light sensors. While the choice of visible light may seem clear, infrared light was chosen as a one dimensional gradient input (the magnitude of the light) as something that could easily be applied and denied in a training environment. Three dimensional acceleration, rotational velocity, and magnetic field sensors were also chosen to give Fido a sense of spatial awareness: through the accelerometer we could detect forces, through the gyroscope (measures rotational velocity) we could detect orientation, and through the magnetometer we could detect magnets (and the Earth's natural magnetic field). Fido was also given the ability to detect it's own battery voltage to detect states of low and high charge, mimicking nature's state of tiredness.

Outputs were similarly chosen: motors allowed Fido to learn movement, a piezoelectric buzzer served as Fido's voice, and an RGB LED gave yet another output for easy training and debugging. The drive system used by Fido for locomotion is a differential-drive arrangement placed inside of a large, hollow sphere. Differential drive was chosen as a simple to understand kinematic model, while the reasoning for the exosphere was somewhat psychological: a "hidden" drive system gives a greater appearance of Fido learning how to move, as the potential for movement is obfuscated. Additionally the exosphere provides protection against high bursts of acceleration used for training.

3.2 Electrical Implementation

An electret microphone was chosen as a sound sensor to reduce much of complexity traditionally associated with microphones (mainly a polarizing power supply). A simple amplifying circuit uses an OPA344 operational amplifier to increase millivolt level voltages to 3.3 volt levels readable by the Teensy ADC (analog to digital converter).

Visible light, infrared light, and battery level sensors were implemented using traditional voltage dividers: the ratio of R_2 to $R_1 + R_2$ determined the multiplier to the input voltage. This makes the output voltage directly proportional to the change in one of the sensor's resistance, allowing us to read and process values using the Teensy's ADC. For visible light a CdS photoresistor was used as R_1. For infrared light a phototransistor was used
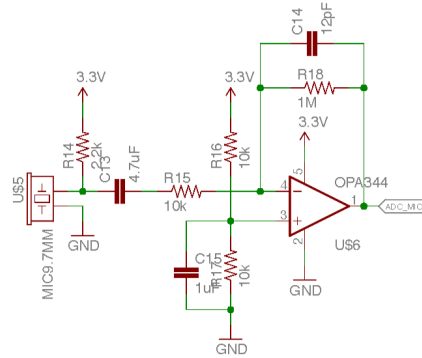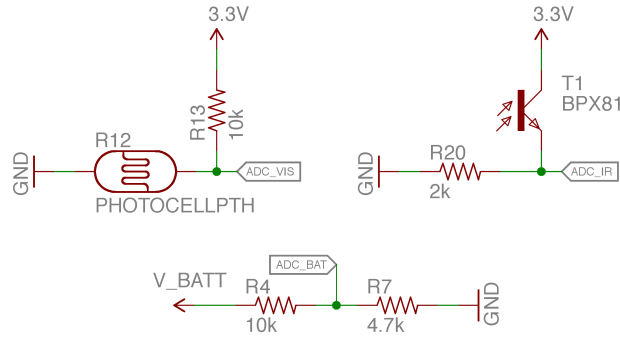
Figure 5: Schematic of Microphone Circuit

Figure 6: Analog Sensor Implementations

instead due to most photoresistors' light sensitivity peaking at 650nm, still in the visible light range.

Fido's motor control was implemented using the TB6612FNG dual motor driver, chosen for its small SMD footprint and integrated flyback diodes to prevent back EMF. When supply voltage is reduced across an inductive load such as a motor, there is often a voltage spike with the capability to damage sensitive electronics. Integrated protection against this phenomenon made this driver a clear choice. Additionally, the driver met current supply capabilities for the chosen DC brushed motors with a constant current of 1.2A per channel and a max current of 3.2A per channel. An RGB LED was added with current limiting resistors to PWM (pulse width modulation) output channels on the Teensy, allowing variable brightness control. A piezoelectric speaker was connected to the Teensy DAC (digital to analog converter) to facilitate clean audio generation. In addition to the noted peripherals a MicroSD slot and Bluetooth 4.0 module were added for purposes of monitoring and debugging:
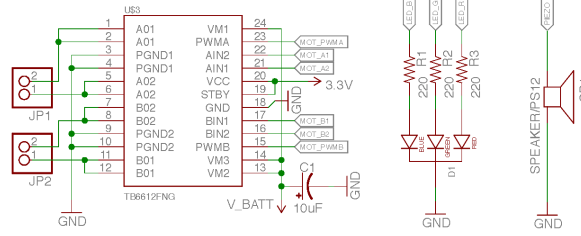
Figure 7: Fido Output Implementations

Neural networks could be stored in text files on the MicroSD, while Fido could be connected to using a mobile application to configuration settings and select neural networks. A complete schematic can be found in the Appendix of this paper.

3.3   Mechanical Implementation and Fabrication
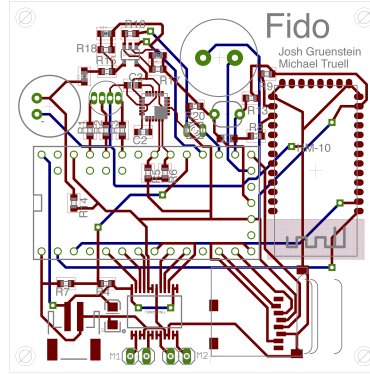
Figure 8: Fido PCB Layout

A printed circuit board was routed in CadSoft EAGLE based on the schematic described above, then manufactured by Smart Prototyping in Hong Kong. The boards were hand assembled (somewhat painstakingly) including all 37 components using standard through-hole and surface-mount soldering techniques. This included use of both a standard soldering pencil and a hot-air rework station with solder paste. The robot chassis was 3D modeled using Autodesk AutoCAD Student, then printed in PLA plastic using a Makerbot Replicator.

4 Results

Results, testing, and applications go here.

4.1 Training Methods

4.2 Findings

4.3 Further Applications

5 Conclusion

Restate, discuss further study, improving experimentation, etc.

6 Appendix

6.1 Electrical Schematic

Dual Motor Driver

U$3

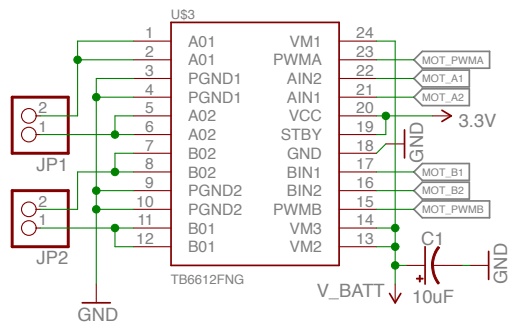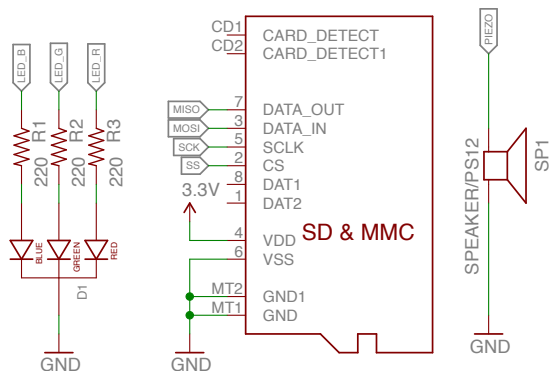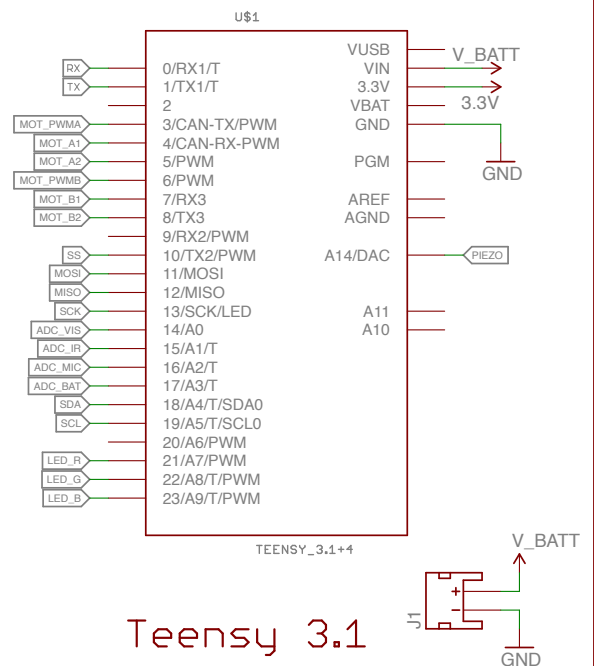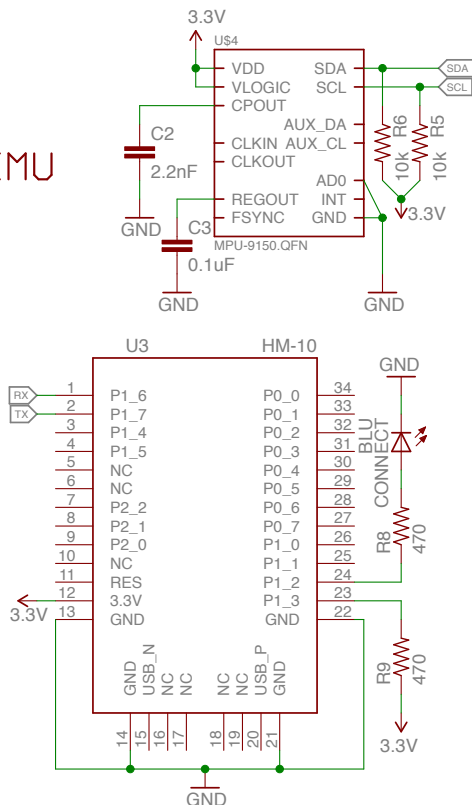| Pin	Name	Name	Pin
1	A01	VM1	24
2	A01	PWMA	23 → MOT_PWMA
3	PGND1	AIN2	22 → MOT_A1
4	PGND1	AIN1	21 → MOT_A2
5	A02	VCC	20 → 3.3V
6	A02	STBY	19
7	B02	GND	18 → GND
8	B02	BIN1	17 → MOT_B1
9	PGND2	BIN2	16 → MOT_B2
10	PGND2	PWMB	15 → MOT_PWMB
11	B01	VM3	14
12	B01	VM2	13

JP1, JP2

TB6612FNG

GND

C1 10uF V_BATT GND

Teensy 3.1

U$1

RX → 0/RX1/T — VUSB → V_BATT
TX → 1/TX1/T — VIN
2 — 3.3V → 3.3V
MOT_PWMA → 3/CAN-TX/PWM — VBAT
MOT_A1 → 4/CAN-RX-PWM — GND → GND
MOT_A2 → 5/PWM
MOT_PWMB → 6/PWM — PGM
MOT_B1 → 7/RX3
MOT_B2 → 8/TX3 — AREF
9/RX2/PWM — AGND
SS → 10/TX2/PWM — A14/DAC → PIEZO
MOSI → 11/MOSI
MISO → 12/MISO
SCK → 13/SCK/LED — A11
ADC_VIS → 14/A0
ADC_IR → 15/A1/T — A10
ADC_MIC → 16/A2/T
ADC_BAT → 17/A3/T
SDA → 18/A4/T/SDA0
SCL → 19/A5/T/SCL0
20/A6/PWM
LED_R → 21/A7/PWM
LED_G → 22/A8/T/PWM
LED_B → 23/A9/T/PWM

TEENSY_3.1+4

J1 V_BATT GND

RGB LED   MicroSD Card

LED_B LED_G LED_R
R1 220 R2 220 R3 220
BLUE GREEN RED
D1
GND

CD1 — CARD_DETECT
CD2 — CARD_DETECT1
MISO → 7 — DATA_OUT
MOSI → 3 — DATA_IN
SCK → 5 — SCLK
CS → 2 — CS
1 — DAT1
8 — DAT2
3.3V
4 — VDD   SD & MMC
6 — VSS
MT2 — GND1
MT1 — GND
GND

SP1
PIEZO
SPEAKER/PS12
GND

IMU

U$4

3.3V
VDD — SDA → SDA
VLOGIC — SCL → SCL
CPOUT
C2 2.2nF
CLKIN — AUX_DA
CLKOUT — AUX_CL
R6 10k R5 10k
REGOUT — AD0
FSYNC — INT
— GND → 3.3V
C3 0.1uF
MPU-9150.QFN
GND

Analog Sensors

C14 12pF
R18 1M
3.3V 3.3V 3.3V
R14 2.2k C13 4.7uF R16 10k
U$5
MIC9.7MM
R15 10k
OPA344 → ADC_MIC
U$6
C15 1uF R17 10k
GND
GND GND GND

3.3V
R13 10k
R12 PHOTOCELLPTH → ADC_VIS
GND

3.3V
T1 BPX81
R20 2k → ADC_IR
GND

V_BATT
ADC_BAT
R4 10k R7 4.7k
GND

HM-10 Bluetooth LE

U3   HM-10

Pin	Name	Name	Pin
1 RX	P1_6	P0_0	34
2 TX	P1_7	P0_1	33
3	P1_4	P0_2	32
4	P1_5	P0_3	31
5	NC	P0_4	30
6	NC	P0_5	29
7	P2_2	P0_6	28
8	P2_1	P0_7	27
9	P2_0	P1_0	26
10	NC	P1_1	25
11	RES	P1_2	24
12	3.3V	P1_3	23
13	GND		22

BLU CONNECT
R8 470
R9 470
3.3V
GND
USB_N NC NC NC NC USB_P GND
14 15 16 17 18 19 20 21
GND

Fido Prototype PCB
Joshua Gruenstein and Michael Truell

TITLE:	FidoBark	
Document Number:		REV: 1.0
Date: 8/30/15 10:18 PM	Sheet:	1/1

References

Dini, S., & Serrano, M. (2012). Combining q-learning with artificial neural networks in an adaptive light seeking robot.

Dudek, G., & Jenkin, M. (2000). *Computational principles of mobile robotics.* New York, NY, USA: Cambridge University Press.

Gaskett, C., Wettergreen, D., & Zelinsky, A. (1999). Q-learning in continuous state and action spaces. In *Australian joint conference on artificial intelligence* (pp. 417–428).

Kim, D. S., & Papagelis, A. J. (n.d.). *Multi-layer perceptron: Artificial neural networks.* `http://www.cse.unsw.edu.au/ cs9417ml/MLP2/`.

MacLeod, C. (2010). *An introduction to practical neural networks and genetic algorithms for engineers and scientists* (Tech. Rep.). Robert Gordon University.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences.