Designing and Building Powerful, Inexpensive Robots for Evolutionary Research

Terence Soule
University of Idaho
Moscow, ID, USA, 83844
tsoule@cs.uidaho.edu 208-310-2084
www.sigevo.org/gecco-2013/
www.cotsbots.wordpress.com

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

University of Idaho

Instructor

- Terence Soule is an Associate Professor in the Computer Science Department at the University of Idaho. He has been active in the field of Evolutionary Computation for 15 years, with over 50 publications. He is an associate editor for the journal Genetic Programming and Evolvable Machines and was Editor in Chief for GECCO in 2012.
- (Minimal expertise in hardware/electronics.)

College of Engineering University of Idaho

Motivation

- Our research group was performing research in cooperative, co-evolution.
- We wanted to start using real agents.
- We looked at many commercially available robots, but couldn't find any that met our requirements and performance/price point.

College of Engineering University of Idaho

Requirements

- On board and real-time evolution
- Complex representations
- Implies "Serious" CPU and memory
- Inter-bot communication
- Software, not Hardware:
  - Minimal electronics/hardware expertise
  - Languages, IDEs, Libraries
- Reasonable cost < $1000
- Ease of use for multiple (10+) bots

College of Engineering University of Idaho

Requirements

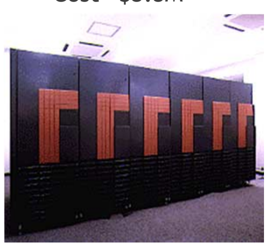
- We couldn't find a commercial robot that met our requirements
- Decided to try a commodity off the shelf (COTS) approach

Commodity Off The Shelf (COTS)

- Use commodity products
- Leverage
  - Mass production
  - Competition
- Get (hopefully)
  - Lower price
  - More features
  - Reliability

COTS Example

Specialty Device
- Cray T3E-1200E
- Cost= $3.5M



=

Commodity Devices
- Network of 32 PCs
- Cost < $100K

EC Has Always Been Cutting Edge

Bennett III, F. and Koza, J. R. and Shipman, J. and Stiffelman, O., "Building a parallel computer system for $18,000 that performs a half peta-flop per day", Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1484–1490, 1999

Our COTSBots Design

- Brain
  - Make decisions, learn, etc.
  - Netbook or smartphone

- Body
  - Moves
  - Hobby or Toy Platform

=

- Spinal Cord
  - Transmits messages from brain to body
  - May transmit sensor data to the brain
  - "Reflexes"(?)

College of Engineering University of Idaho

Not completely original

- Soule, T., Heckendorn, R. B., "COTSBots: computationally powerful, low-cost robots for Computer Science Curriculums", Journal of Computing Sciences in Colleges, 27:1, pp. 180—187, 2011. [Smartphone or netbook powered]
- Bergbreiter, Sarah and Pister, Kristopher SJ, "Cotsbots: An off-the-shelf platform for distributed robotics", Intelligent Robots and Systems, pp. 1632—1637, 2003. [Microcontroller powered cotsbot]
- Friedman, Jonathan and Lee, David and Tsigkogiannis, Ilias and Wong, Sophia and Chao, Dennis and Levin, David and Kaiser, William and Srivastava, Mani, "Ragobot: A new platform for wireless mobile sensor networks", Distributed Computing in Sensor Systems, 2005.
- Kelly, Jonathan and Binney, Jonathan and Pereira, Arvind and Khan, Omair and Sukhatme, Gaurav S, "Just add wheels: leveraging commodity laptop hardware for robotics and ai education", Proceedings of AAAI, 2008 AI Education Colloquium, pp. 50—55, 2008. [Laptop on iRobot/Roomba]
- Nasereddin, Hebah HO and Abdelkarim, Amjad Abdullah, "Smartphone Control Robots Through Bluetooth", International Journal of Research and Reviews in Applied Sciences, 4:4, 2010.

College of Engineering University of Idaho

Not completely original

- Romo (Kickstarter)

- Oddwerx (Kickstarter)
- Smartphone powered NASA Satellite (2013)

College of Engineering University of Idaho

Some projects we've tested

- Color following
- Voice control
- Real-time evolutionary training for color following
- Clicker training
- Tele-operation

College of Engineering University of Idaho

Body/Bodies

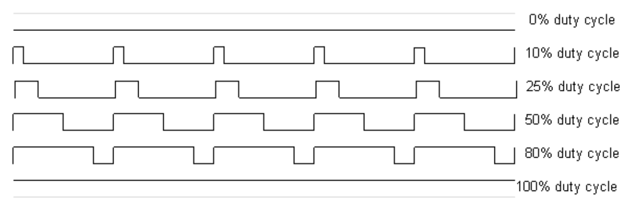
Bodies - Hobby and Toy

Bodies - Tracked

- Typically 2 DC motors

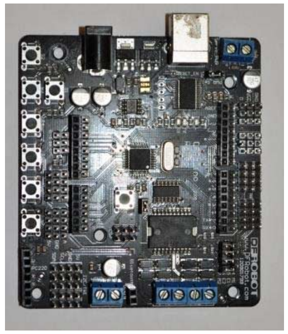
Bodies - DC Motors

- Motor speed is controlled by pulse width modulation (PWM)



Spinal Cord - Arduino

- www.arduino.cc
- Microcontroller
- Atmel ARM processor (84Mhz)
- Multiple I/O pins
- Outputs PWM (and servo signals)



DFRobot Romeo

Spinal Cord - Arduino

- I/O Pins output PWM signals and servo signals (select pins)
- I/O Pins only 3.3 volts max
- Only 40 milliamps
- Requires additional power regulation to run motors

Bodies – DC Motors

- Motor Shield
- Powers 2 motors
- 7-12 volts
- 4 amps

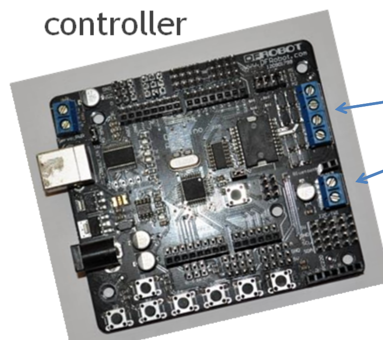
Motor outputs (2)

Power in (7-12 volts)
Can also power the board

Plugs into Arduino

Bodies – DC Motors

- Or DFRobot Romeo w/ built-in motor controller



Motor outputs (2)

Power in (7-12volts)
Can also power the board

DC Motors – Arduino Code

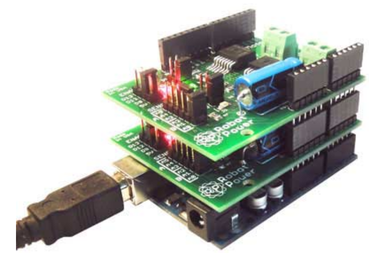
```
//setup
pinMode(leftDrivePin, OUTPUT);
pinMode(leftDirectionPin, OUTPUT);

//loop
digitalWrite(leftDirectionPin,HIGH);
analogWrite(leftDrivePin, dutyCycle);
// dutyCycle is a value from 0 to 255
```

Use one pin for each track

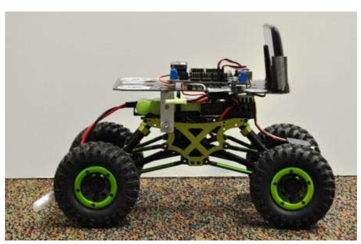
Larger Vehicles

Mega Motor Shield 13A, 5-28V



Bodies - Servos

- Small hobby cars: direct control of steering servo, drive servo goes through an electronic speed control device to increase power.

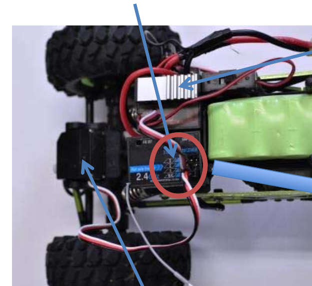


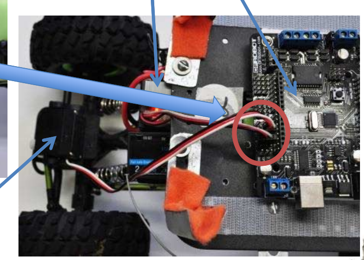
Bodies - Servo

RC Receiver

Electronic Speed Control (ESC)

Arduino

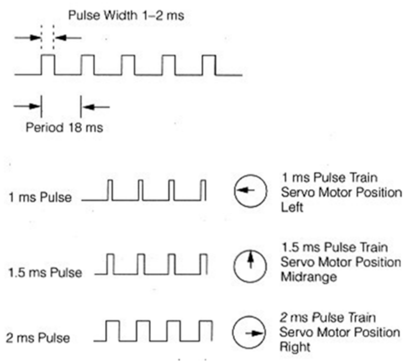




Steering Servo

Disconnect the ESC and steering servo leads from the RC Receiver and plug into servo pins on the Arduino.

Bodies - Servos



Or controls a continual turning motor's speed:
Midrange = Stop
Left = Backwards
Right = Forwards

College of Engineering University of Idaho

Servos – Arduino Code

```
#include <Servo.h>      // servo library
Servo servo1;            // create servo object

// setup
pinMode(pin,OUTPUT);
servo1.attach(pin);   // attach servo object to pin

// loop
servo1.write(angle);  // angle = 0-180
```

College of Engineering University of Idaho

Bodies – Stepper Motors and Encoders

- Arduinos can drive stepper motors
- The Rover 5 has optional motor encoders
- Both options potentially give better control/feedback, but we have not experimented with them

College of Engineering University of Idaho

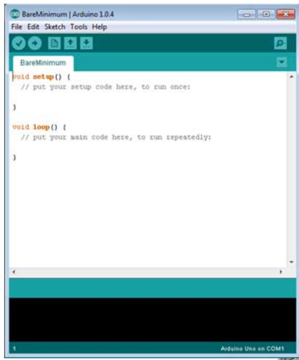
Spinal Cord - Arduino

- Receives commands from the brain via serial connection:
  - USB (netbook)
  - Serial over bluetooth (Smartphone)
  - Android open accessory (USB w/ Smartphone) (http://source.android.com/tech/accessories/index.html)
- Receives data from sensors and transmits to the brain via serial

College of Engineering University of Idaho

Spinal Cord Arduino

- Programmable in C
- Has its own IDE (modeled on Processing)
- Lots of examples
- Useful libraries



College of Engineering

University of Idaho

Basic Arduino Code – Tracked Vehicle

```
const int leftDrivePin = 6;
const int rightDrivePin = 5;
const int leftDirectionPin = 7;
const int rightDirectionPin = 4;

void setup()
{
 // initialize the serial communication:
 Serial.begin(9600);
 pinMode(leftDrivePin, OUTPUT);
 pinMode(rightDrivePin, OUTPUT);
 pinMode(leftDirectionPin, OUTPUT);
 pinMode(rightDirectionPin, OUTPUT);
}

void loop() {
 char input;
 if (Serial.available()) {
  input = Serial.read();

switch(input){
   case 'F':  // forward
    digitalWrite(leftDirectionPin,HIGH);
    digitalWrite(leftDrivePin, HIGH);
    digitalWrite(rightDirectionPin,HIGH);
    digitalWrite(rightDrivePin, HIGH);
    break;
   case 'B':  // backwards
    digitalWrite(leftDirectionPin,LOW);
    digitalWrite(leftDrivePin, HIGH);
    digitalWrite(rightDirectionPin,LOW);
    digitalWrite(rightDrivePin, HIGH);
    break;
   case 'S':  // stop
    digitalWrite(leftDrivePin, LOW);
    digitalWrite(rightDrivePin, LOW);
…
```

(Full code at: www.cotsbots.wordpress.com)

College of Engineering

University of Idaho

Ex: More Complex Arduino Code

Driving using 6 byte packet for variable speed and turning

Packets start with '#' character (byte 0) and end with ';' character (byte 5)

byte 0 is '#' for ALL driving types

byte 1: 'T' for servo turning, 'F' or 'B' for left track direction.

byte 2: 0-180 for servo turning or 0-255 for left track speed.

byte 3: 'P' for servo power, 'F' or 'B' for right track direction.

byte 4: 0-180 for servo speed or 0-255 for right track speed.

byte 5 is ';' for ALL driving types

(Full code at: www.cotsbots.wordpress.com)

College of Engineering

University of Idaho

Brains

- Do all of the heavy lifting
- Computational power for on-board, real-time evolution
- Sensors
- Communication
- Etc.

College of Engineering

University of Idaho

Brains - Netbook

- Serial over USB to drive robot
- Any language, any IDE, lots of libraries
- CPU power, memory
- Wireless communication
- Built-in camera, microphone, speakers, screen, etc.
- Add any USB device
  - stereo camera
  - GPS
  - Etc.

College of Engineering University of Idaho

Brains – Android Smartphone

- Easy start-up (developer.android.com/sdk/index.html)
- Galaxy Note (~$500):
  - Quad-core 1.6 GHz Cortex-A9 CPU
  - 2 GB RAM
  - 16 GB storage
- Come with graphics co-processors (and OpenCV libraries have been tailored for them)

College of Engineering University of Idaho

Brains – Android Smartphone

- Built-in
  - Camera (front and rear) (narrow field of view)
  - Speaker/microphone
  - Accelerometers
  - Compass
  - GPS
  - Touchscreen
  - Wireless and bluetooth

College of Engineering University of Idaho

Android Code - Bluetooth

```
try {
  m_BluetoothDevice =  m_BluetoothAdapter.
                getRemoteDevice(m_Address);
  try {
      m_BlueToothSocket =    m_BluetoothDevice.
    createRfcommSocketToServiceRecord(sf_UUID);
      m_BluetoothAdapter.cancelDiscovery();
      try {
          m_BlueToothSocket.connect();
       } catch (IOException e1) {
            Log.e(sf_TAG, "failed to connect");
            try {
                m_BlueToothSocket.close();
            } catch (IOException e2) {
                Log.e(sf_TAG,
"failed to close, in e1(connectFail)");
            }
       }

       try {
          m_OutStream =
           m_BlueToothSocket.getOutputStream();
          succeeded = true;
       } catch (IOException e2) {
         Log.e(sf_TAG,"Failed to open output stream");
          m_Connected = false;
       }
  } catch (IOException e) {
     Log.e(sf_TAG, "Failed to setup bt socket");
     m_Connected = false;
  }
}
```

College of Engineering University of Idaho

Android Code - Messages

```
private void write(char data) {
    if (m_OutStream != null) {
        try {
            m_OutStream.write(data);
        } catch (IOException e) {
            Log.e(sf_TAG, e.toString());
        }
    }
}
```

College of Engineering University of Idaho

Android Code - Messages

```
private void writeBytes(char[] dataVals) {
    if (m_OutStream != null) {
        try {
            for (char c : dataVals) {
                m_OutStream.write(c);
            }
        } catch (IOException e) {
            Log.e(sf_TAG, e.toString());
        }
    }
}
```

College of Engineering University of Idaho

Android Code – Image Processing

```
while (!m_IsActivityPaused) {
    synchronized (this) {
        if (m_Camera == null)
            break;
        if (!m_Camera.grab())
            break;
// process camera image taken
        processFrame(m_Camera);
    }
    if (mBmpCanvas != null) {
Canvas canvas = m_Holder.lockCanvas();
        if (canvas != null) {
            // Redrawing FPS
canvas.drawColor(android.graphics.Color.
BLACK);
// Draw the camera image to screen
canvas.drawBitmap(mBmpCanvas,
    (canvas.getWidth() -
    mBmpCanvas.getWidth()) / 2,
    (canvas.getHeight() -
    mBmpCanvas.getHeight()) / 2, null);
m_Holder.unlockCanvasAndPost(canvas);
        }
    }
}
```

College of Engineering University of Idaho

Android Code – Image Processing

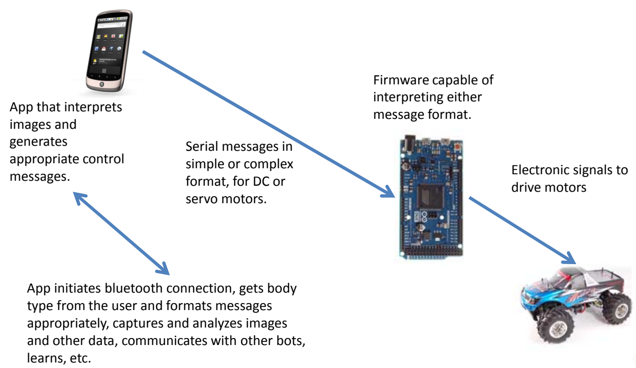
```
protected void processFrame(VideoCapture capture) {
    // capture data from camera
    capture.retrieve(m_Rgba,
Highgui.CV_CAP_ANDROID_COLOR_FRAME_RGBA);

    // Do all processing on m_Rgba here

    Utils.matToBitmap(m_Rgba, mBmpCanvas);
}
```

College of Engineering University of Idaho

Software Architecture



Smart Phone How-To (One Option)

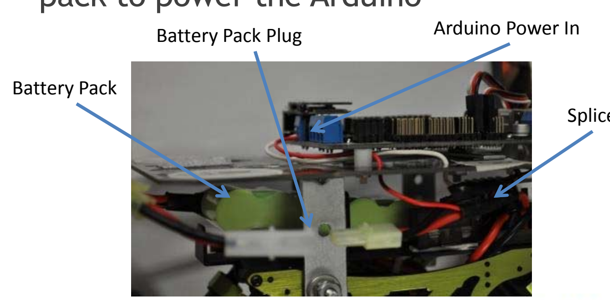
1) Order the Rover 5 Tank Kit (Tracked vehicle, Romeo Board, and bluetooth chip) Dfrobot (www.dfrobot.com/index.php) or local distributor
2) Get an Android phone
3) Install Eclipse and the Android SDK (www.developer.android.com/sdk/index.html) one bundle
4) Install Arduino (select board type and port in tools menu)
5) Install spinal cord code (www.cotsbots.wordpress.com)
6) Install Android app code (www.cotsbots.wordpress.com)
7) Run the app, drive the robot

Power Supply

- For the Rover 5 we replaced the 6 AA battery block with an 8 AA battery block.
- A 2.1mm center-positive plug makes it simple to turn the Rover 5's Arduino on and off.

Power Supply

- For the Rockcrawler we spliced the battery pack to power the Arduino



Cost

- Rover 5 (Tracked platform) + Romeo microcontroller + bluetooth chip, as a kit for $130.
- Rockcrawler (RC car) ($130) + Romeo microcontroller + bluetooth chip ($80)
- Smartphone ($200-$600)

College of Engineering University of Idaho

Features

- 5-20 frames per second of processed images (varies with phone and lighting)
- 2+ hour battery life
  - Extendable with smart phone accessories
- Durable
- All-terrain vehicles
- Fairly small footprint
- 3rd Party accessories
  - Fish-eye lens
  - Waterproof cases
  - Bluetooth keyboards and mice
  - Etc.

College of Engineering University of Idaho

Other Features

- Graphics Co-processor for image processing
- OpenCV libraries tailored to work with smartphone graphics co-processors ["Real-time computer vision with OpenCV" Pulli, K., Baksheev, A., Kornyakov, K.l and Eruhimov, V., Communications of the ACM, 55:6, 61—69, 2012]
- Up-gradable (remove brain, insert next generation phone/netbook)
- Teaches mobile computing

College of Engineering University of Idaho

Other Features

- Easy build ~30 minutes
- No electronics skills needed (e.g. no soldering)
- Can reasonably maintain a swarm of COTSBots

College of Engineering University of Idaho

Issues

- COTSBots are not precision devices
  - Could drive stepper motors
- No array of distance sensors
  - Makes obstacle avoidance difficult
  - Could be added
- Cameras have narrow field of view
- No precise simulation environment

College of Engineering University of Idaho

Arduino Accessories

- "Plug and Play"
  - Ultrasonic range finders
  - Infrared range finders
  - Light sensors
  - Bump sensors
  - Magnetic sensors
  - RFID sensors
  - Temperature and humidity sensors

College of Engineering University of Idaho

Arduino Sensor Code

From the Arduino Examples:

```
/*	Reads an analog input on pin 0, prints the result to the serial monitor.
	 Attach the center pin of a potentiometer to pin A0, and the outside pins to
	+5V and ground.
	This example code is in the public domain.
 */
void setup() {
	Serial.begin(9600);
}

void loop() {
	// read the input on analog pin 0:
	int sensorValue = analogRead(A0);
	// print out the value you read:
	Serial.println(sensorValue);
	delay(1);	// delay in between reads for stability
}
```

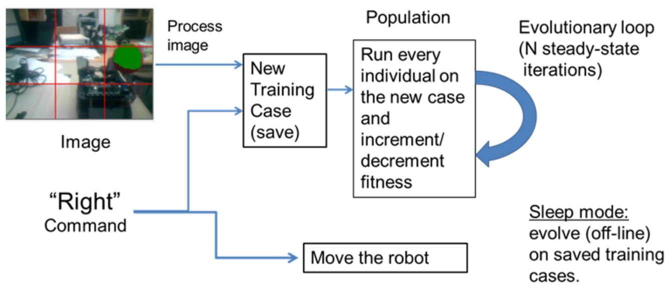
College of Engineering University of Idaho

Projects – Color Following

- Illustrates bots ability to identify each other

College of Engineering University of Idaho

Projects: Evolved Color Following

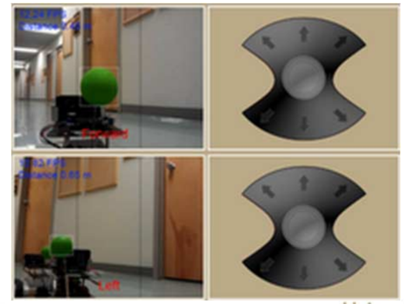


Soule, T. and Heckendorn, R. B., "A Practical Platform for On-Line Genetic Programming for Robotics", Workshop on Genetic Programming Theory and Practice, Springer, 2012.

College of Engineering University of Idaho

Projects: Teleoperation

- Can be used for training



College of Engineering University of Idaho

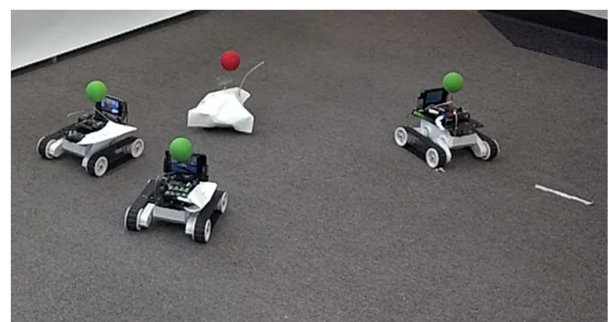
Projects: Clicker Training

- Bot tries random actions
- Human controller gives positive or negative reinforcement via clicks
- We have trained simple tasks: go to green ball, return to red ball.

College of Engineering University of Idaho

Projects: Lions and Gazelle

- Training in simulation. Testing with COTSBots.
- During testing gazelle is controlled by a person.



College of Engineering University of Idaho

Acknowledgements

- Dr. Robert Heckendorn
- Jonathon Lamb
- Dallas Stinger
- Daniel Norris
- Joshua Rubini
- Curtis Bower
- Keith Denney
- Travis DeVault
- Juan Marulanda
- Maxwell McKinnon
- Nate Ebel
- Jayandra Pokharel
- Jafar Al-Gharaibeh
- Trevor Davenport
- Brandon Morton
- Alex Odom
- John Sandmeyer
- Tim Tate
- Tom Zirker
- Mike Simon and Creation Logic

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Thank You!

Questions?

Common Bots – e-Puck

- 8 IR sensors
- 3 axis accelerometer
- Three microphones and a speaker
- 60MHz processor
- Simulation software
- 640x480 color camera (The full flow of information this camera generates cannot be processed by a simple processor like the dsPIC on the robot. Moreover the processor has 8k of RAM, not sufficient to even store one single image.)
- 8 red leds
- ~$850

Common Bots - Khpera

- Khepra III:
  - 9 infrared, 5 ultrasonic sensors
  - 8 hour battery life
  - 60Mhz, 4KB Ram (Upgradable)
  - $3360

iRobot Create (Roomba)

- 30+ sensors
- Write basic scripts of up to 100 Open Interface Commands
- $220
- (An alternative body)

College of Engineering

University of Idaho