Robots for Disability: EEG Operated Water Drinking System

Henry M. Clever
Seunghoon Jeong
EEG Operated Water Drinking System

• Purpose: To assist people with quadriplegia in drinking autonomously.
  – Because people with this severe of a disability are unable to move their arms and legs, they require caregiver assistance to drink and eat.
  – By using an Emotiv EPOC EEG machine, we are able to reliably detect eye blinks through brain signals in the cap.
  – Using a Raspberry pi 2, we constructed a system with an automatic drinking straw that is triggered by a specified series of blinks based on timing intervals between the blink.
  – The drinking system was built conventional mechatronics parts with a Raspberry Pi that has a filtering system on the EEG sensor output to detect blinking
EEG Operated Water Drinking System

• EEG headset device (Emotiv EPOC)
  - 14 EEG channels + 2 references
  - Channel location:
    AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4
  - Uses sequential sampling method, single ADC, at a rate of 128 SPS
Blinking

- Data for all EEG sensors displayed in real time on Raspberry Pi with a Python graphing program for data visualization
**Phase 1:** 7-segment displays “1”  
- Servo in RESTING position (0 deg)  
- Water pump OFF

- 5 blinks in 5 seconds?

**Phase 2:** 7-segment displays “2”  
- Servo in ACTIVE position (180 deg)  
- Water pump OFF

- No signal for 20 seconds? Timeout  
- 4 blinks in 4 seconds?

**Phase 3:** 7-segment displays “3”  
- Servo in ACTIVE position (180 deg)  
- Water pump ON

- No signal for 30 seconds? Timeout  
- 3 blinks in 3 seconds?
Blinking Phase Simulation

• Phase system: Matlab plots show virtual blinking machine
Blinking Phase Python Code

Filter data for blink to establish switch:

```python
if (testmat[9,1]-testmat[2,1]) > 300 and count == 40: # if a blink occurs
  if switcher == True: # this statement reverses switch position when blink occurs
    switcher = False
    GPIO.output(25,GPIO.LOW)
    count = 1
  elif switcher == False:
    switcher = True
    GPIO.output(25,GPIO.HIGH)
    count = 1

for y in range(1, 5):
  timeque[y,0]=timeque[y+1,0]
  timeque[y,1]=timeque[y+1,1]

  timeque[5,0]=testmat[9,0]
  timeque[5,1]=testmat[9,1]
```

Search for sequence to change phase:

```python
if phase == 1: # establish test for servo rotation: look for 5 blinks
  GPIO.output(5,GPIO.LOW)
  GPIO.output(6,GPIO.LOW)
  GPIO.output(12,GPIO.LOW)
  GPIO.output(13,GPIO.HIGH)
  GPIO.output(16,GPIO.LOW)
  GPIO.output(26,GPIO.LOW)
  if timeque[1,0]==0:
    # for all cases where the time que isn't full continue filling
    timeque[0,0]=0
    timeque[0,1]=0
    timeque[1,0]=0
    timeque[1,1]=0
  elif (timeque[5,0]-timeque[1,0]) >= threshold1 and timeque[1,0] > 0:
    # too long, minor timeout. clear queue
    timeque[0,0]=0
    timeque[0,1]=0
    timeque[1,0]=0
    timeque[1,1]=0
  elif (timeque[5,0]-timeque[1,0]) < threshold1 and timeque[1,0] > 0:
    # rotate servo go to next phase of blinking
    phase=2
    turntime=timeque[5,0]
    timeque = numpy.zeros((6,2))
```

Move servo:

```python
if phase == 1:
  duty = float(175) / 10.0 + 2.5
  pwm.ChangeDutyCycle(duty)
elif phase == 2:
  duty = float(5) / 10.0 + 2.5
  pwm.ChangeDutyCycle(duty)
elif phase == 3:
  duty = float(5) / 10.0 + 2.5
  pwm.ChangeDutyCycle(duty)
```
EEG Operated Water Feeder

- System Construction (Hardware)
Display Board

- System Construction (Bread board)
# Project costs

<table>
<thead>
<tr>
<th>Material</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEG headset device</td>
<td>$399</td>
</tr>
<tr>
<td>Raspberry pi 2 with starter kit</td>
<td>$69.99</td>
</tr>
<tr>
<td>Circuit &amp; board components</td>
<td>Free (from ADOL lab)</td>
</tr>
<tr>
<td>Hardware design components_Frame</td>
<td>Free (from Mechatronics lab)</td>
</tr>
<tr>
<td>Hardware design components_Bolt &amp; Nuts</td>
<td>$11.99</td>
</tr>
<tr>
<td>Peristaltic Liquid Pump (Adafruit)</td>
<td>$38.81</td>
</tr>
<tr>
<td>Acetal Resin Tube (Straw)</td>
<td>$11.70</td>
</tr>
<tr>
<td>Water reservoir</td>
<td>Free</td>
</tr>
<tr>
<td>Standard servo (Parallax)</td>
<td>Free (from mechatronics kit)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$531.49</strong></td>
</tr>
</tbody>
</table>