Name: Shaan Yadav
NetID: ay140
Honor Code: *I have adhered to the Duke Community Standard in completing this assignment.*

Assignment (1):
This is because the waveform generator provides the voltage difference that would exist at the output if the output had a resistance of 50Ω across it. In this case there is a greater resistance than 50Ω across the output therefore the measured voltage is less than expected.

Deliverable (1): For this signal, report the following values as measured from the specified devices:

<u>Multimeter</u>

DCV: -0.0010 V
ACV: 2.84796 V
Frequency: 10 0001 Hz
Period: 99.9995 μs

<u>Oscilloscope</u>

Average: -56.7 mV
Peak-to-Peak: 8.4V
AC RMS: 2.907V
DC RMS: 2.91V
Frequency: 10 001.2 kHz
Period: 100.012 μs

1. $DCV^2 + ACV^2 = 8.1109$
2. $(DCRMS)^2 = 8.4681$

∴ Difference = (2) - (1) = 0.3572

Potential sources of error:
Human measuring error
Equipment resistance not exact
Equipment not calibrated
Values on measuring equipment not completely stabilizing

Deliverable (2): For this signal, report the following values as measured from the specified devices:

Multimeter

DCV: -0.0095 V
ACV: 1.41505 V
Frequency: 10 0001 Hz
Period: 99.9994 μs

Oscilloscope

Average: -66.7 mV
Peak-to-Peak: 4.3V
AC RMS: 1.44 V
DC RMS: 1.44 V
Frequency: 10.0023 kHz
Period: 100.02 μs

1. $DCV^2 + ACV^2 = 2.0025$
2. $(DCRMS)^2 = 2.0736$

∴ Difference = (2) - (1) = 0.0711

Potential sources of error:

Human measuring error
Equipment resistance not exact
Equipment not calibrated
Values on measuring equipment not completely stabilizing

Deliverable (3): For this signal, report the following values as measured from the specified devices:

<u>Multimeter</u>

DCV: 0.0002 V
ACV: 1.98848 V
Frequency: 10 0001 Hz
Period: 99.9995 μs

<u>Oscilloscope</u>

Average: -36.1mV
Peak-to-Peak: 2.04V
AC RMS: 2.040 V
DC RMS: 2.04 V
Frequency: 10.001 kHz
Period: 99.998 μs

1. $DCV^2 + ACV^2 = 3.9541$
2. $(DCRMS)^2 = 4.1616$

∴ Difference = (2) - (1) = 0.2075

Potential sources of error:

Human measuring error
Equipment resistance not exact
Equipment not calibrated
Values on measuring equipment not completely stabilizing

Deliverable (4): For this signal, report the following values as measured from the specified devices:

<u>Multimeter</u>

DCV: 2.0013V
ACV: 1.98474V
Frequency: 10 0001 Hz
Period: 99.9995 μs

<u>Oscilloscope</u>

Average: 1.974V
Peak-to-Peak: 4.34V
AC RMS: 2.039 V
DC RMS: 2.84V
Frequency: 10.000 kHz
Period: 99.998 μs

1. $DCV^2 + ACV^2 = 7.9444$
2. $(DCRMS)^2 = 8.0656$

∴ Difference = (2) - (1) = 0.1212

Potential sources of error:

Human measuring error
Equipment resistance not exact
Equipment not calibrated
Values on measuring equipment not completely stabilizing

Deliverable (5): For this signal, report the following values as measured from the specified devices:

Multimeter

DCV: 0.404 V
ACV: 1.18062
Frequency: 10 0001 Hz
Period: 99.9995 μs

Oscilloscope

Average: 371.2 mV
Peak-to-Peak: 4.30V
AC RMS: 1.2255 V
DC RMS: 1.28 V
Frequency: 10.000 kHz
Period: 99.998 μs

1. $DCV^2 + ACV^2 = 1.5571$
2. $(DCRMS)^2 = 1.6384$

∴ Difference = (2) - (1) = 0.0813

Potential sources of error:

Human measuring error
Equipment resistance not exact
Equipment not calibrated
Values on measuring equipment not completely stabilizing

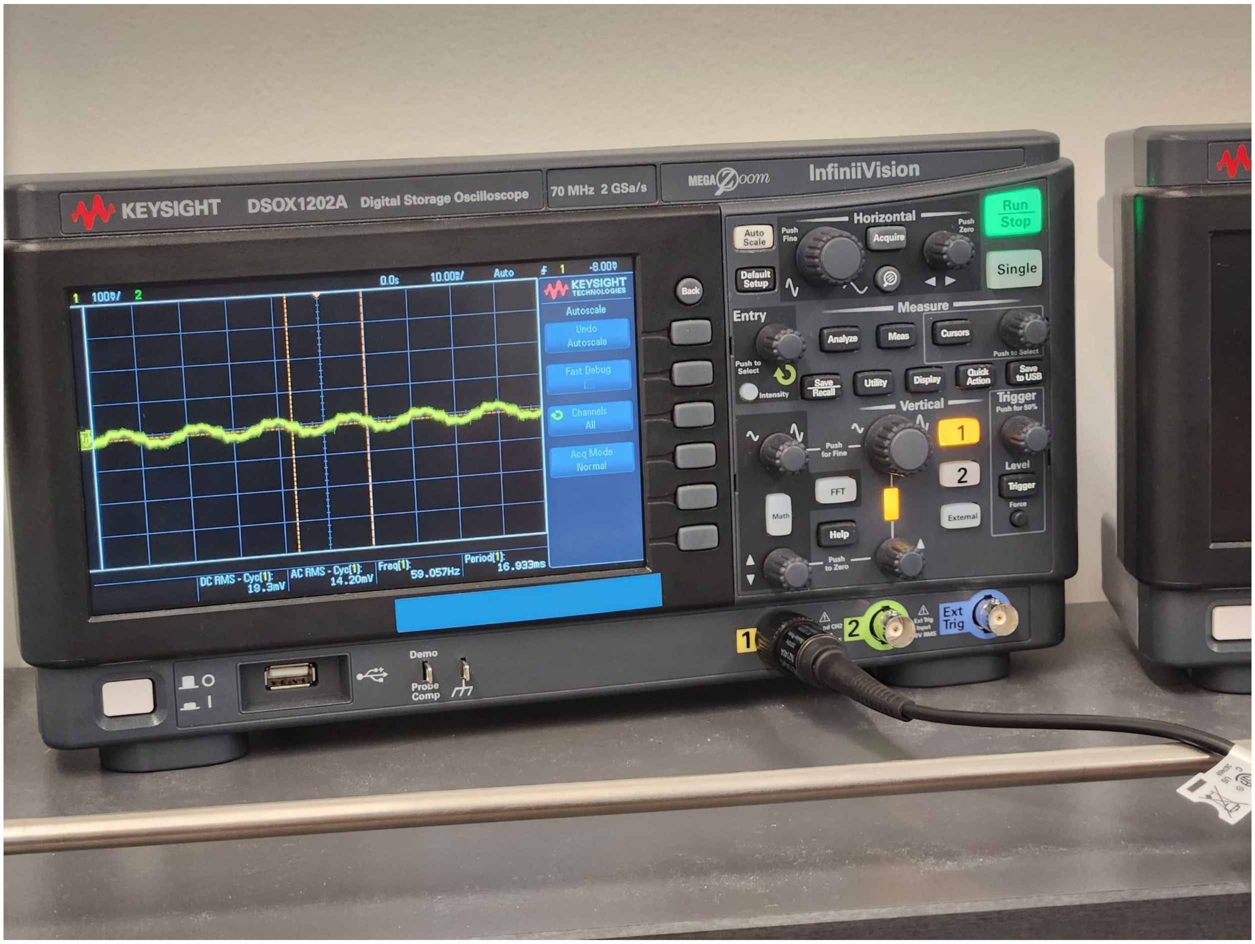
Deliverable (6)

```
#include <Servo.h>
const int pingPin = 35;
Servo servoLeft;
Servo servoRight;

void setup() {
  Serial.begin(9600);
  pinMode(37, OUTPUT);
  pinMode(39, OUTPUT);
  digitalWrite(37, HIGH);
  digitalWrite(39, LOW);

  servoLeft.attach(12);
  servoRight.attach(11);
}

void loop() {
  long duration;
  float inches, cm;
  // short LOW pulse to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  // 2 ms HIGH pulse
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  // back to LOW
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))):
  // a HIGH pulse whose duration is the time (in microseconds)
  // from the sending of the ping to the reception of
  // its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
```

```
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(200);

  if (inches >= 10) {
      Serial.print("Too Far!");
      servoLeft.writeMicroseconds(1700);
      servoRight.writeMicroseconds(1300);
  }

   if (inches < 6) {
      Serial.print("Too Close!");
      servoLeft.writeMicroseconds(1300);
      servoRight.writeMicroseconds(1700);
  }

  if (inches > 6 && inches < 10) {
      Serial.print("Just Right!");
      servoLeft.writeMicroseconds(1500);
      servoRight.writeMicroseconds(1500);
  }
}

float microsecondsToInches(long microseconds) {
  // The speed of sound is about 1125 ft/s
  // Sound takes about 74.074 us to travel 1 in
  return microseconds / 74.074 / 2;
}

float microsecondsToCentimeters(long microseconds) {
  // The speed of sound is about 343 m/s
  // Sound takes about 29.155 us to travel 1 cm
  return microseconds / 29.155 / 2.0;
}
```

Discussion: A way we could improve the code is by reducing the delay between each reading. This would essentially reduce the sampling rate which would make the reaction time of the bot quicker to when it detects a surface is too close or too far.

Deliverable (8):



The waveform is a sinusoid with a measured frequency of 59.057 Hz. This signal is generated by the alternating current in the power cords which is a part of a larger energy grid. The measured frequency is close to 60 Hz which is the frequency the US national power grid runs on.