Inquiring into the output of a potentiometer under load

Igor Krzywda

February 24, 2021

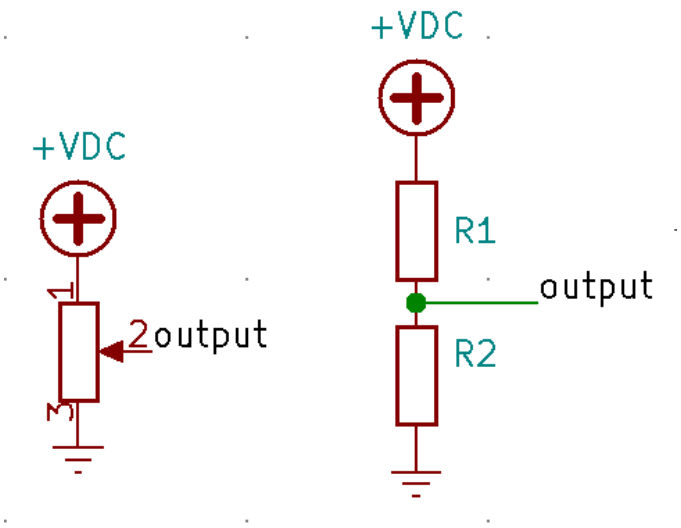
1 Introduction

Potentiometer is an ubiquitous electrical component with its applications ranging from measuring voltage to controlling output voltage supplied to other parts of a circuit. The latter is not as straight forward as it would seem, as the output of loaded potentiometer does not resemble its original output gradient.

2 Potentiometer in detail

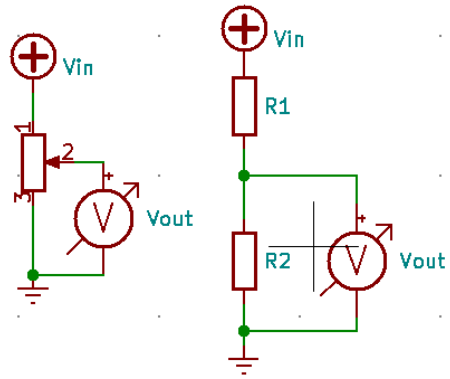
Potentiometer in its essence is a voltage divider. It consists of a resistive wire and a slider. As the slider goes through the wire, the whole potentiometer can be interpred as any vanilla voltage divider with R_1 above the slider and R_2 below the slider (with reference to figure 1). The signal outputted by the potentiometer can be taken with reference to the position of the slider which, in case of a linear potentiometer, can be expressed as either R_1 or R_2 depending on which end of the wire is to be taken as reference point. There are also other types of potentiometers, where the resistance changes logarithmically or exponentially, but simplicity and clarity's sake the potentiometer used in the considerations here is of linear type.

Figure 1:



3 Output voltage under no load

Figure 2:



In order to calculate the output voltage divider will give, we need to use Ohm's law to derive the formula (use figure 2 as reference). Ohm's law states that $V = IR$, and since our output voltage is found across R_2, we can find V_{out} by doing the following (Equation (2))

$$V_{out} = IR_2, I = \frac{V}{R} \tag{1}$$

$$V_{out} = \frac{V_{in}}{R_1 + R_2} \cdot R_2 = V_{in} \frac{R_2}{R_1 + R_2} \tag{2}$$

From the calculations we could say that the output voltage V_{out} is dependent on the ratio of R_2 and the resistance of the whole circuit $R_1 + R_2$, and considering the case of potentiometer, where the net resistance always amounts to some constant value, the function of V_{out} with reference to R_2 is linear. From this point, it could be wrongly said that in order to step down the voltage to power some component, all that needs to be done is to set the dial to desider R_2 and plug in the load. This, however is not the case.

4 Output voltage under load

Figure 3:

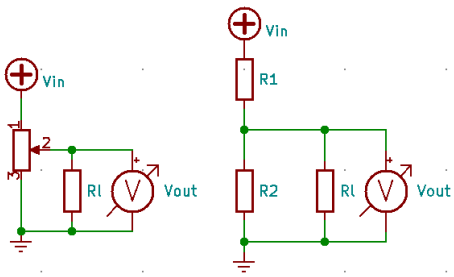


Figure 3 shows our potentiometer with load connected to the output terminal. It is already visible that the load, which has its own resistance is connected in parallel to R_2 rendering the net resistance of R_L and R_2 different, as the resistance calculated for parallel circuit (Equation (3)) will amount to less than the sum of the two resistances.

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + ... + \frac{1}{R_n} \tag{3}$$

We can derive the formula for this case from equation (2) (Equation 6)

$$R_p = (\frac{1}{R_2} + \frac{1}{R_L})^{-1} = \frac{R_2 R_L}{R_2 + R_L} \tag{4}$$

$$V_{out} = \frac{V_{in} \cdot R_p}{(R_1 + R_p)} \tag{5}$$

$$V_{out} = \frac{V_{in}(R_2 R_L)}{R_1(R_2 + R_L) + R_2 R_L} \tag{6}$$

We can see from equation (6) that the load connected to the potentiometer is an integral part to the circuit that cannot be neglected in any way and that it caused the linearity of the voltage output to disappear. Accounting for the potentiometer resistance gain being linear, the output voltage can be expressed by a function (7),

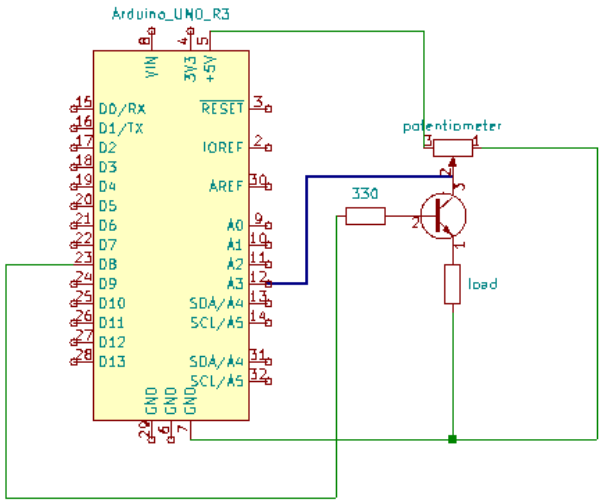
$$V_{out}(R_2) = \frac{V_{in}(R_2 R_L)}{(R_{pot} - R_2)(R_2 + R_L) + R_2 R_L} \tag{7}$$

where R_{pot} is the resistance of the whole potentiometer.

5 Testing numerical predictions against empirical data

5.1 Apparatus

Figure 4:



In order to test the numerical predictions, a test circuit was devised (See Figure 4), which uses an Arduino microcontoller to read analog signal from the output of the loaded potentiometer. Because all analog pins on ATMEGA328p have common ground, the transistor in the circuit disconnects temporarily the load in order to get reference point for unloaded potentiometer which, owing to the potentiometer being linear, can be used as the x axis of the plot. A pair of comma-separated values (position of the dial, output signal) from 0 to 1023 are sent through serial bus 10 times a second to a computer with software on it saving the values to a csv file. Then the file is fed to a Python script mapping the unloaded analog values to resistance between 0 and 50000, and values of loaded potentiometer to voltage between 0 and 5, all of which is later plotted into a graph. (See Appendix A for greater details on the apparatus and software) Sample of raw data is in Appendix B, however, as the files are on average 120 lines, a link to repository is provided for full versions.

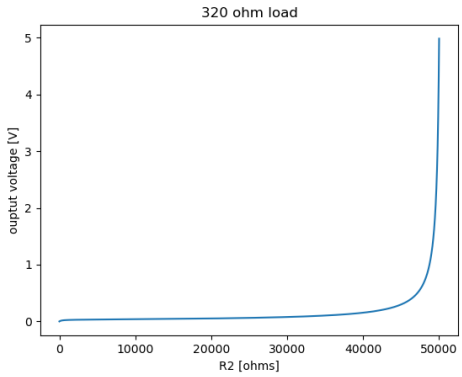
5.2 Method

A set of graphs with ascending load resistance as a variable are generated using a Python script using the function $V_{out}(R_2)$ derived before ((7)). The load resistances are chosen in such way to show the change in the shape of the

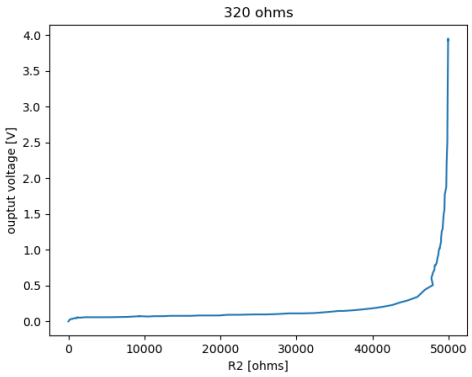
graph gradually. Then, the same resistances are met on the apparatus and the data is gathered and parsed, as described in previous subsection.

5.3 Results

5.3.1 320Ω load

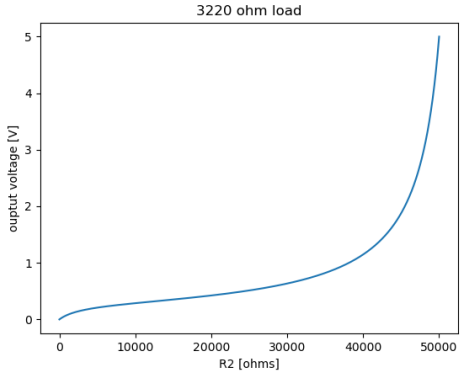


(a) prediction

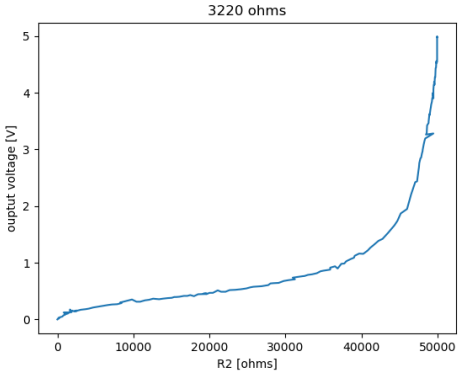


(b) experiment result

5.3.2 3220Ω load

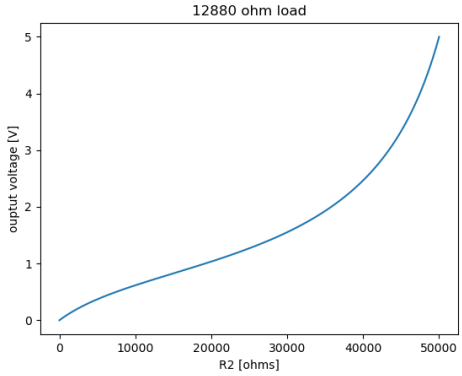


(c) prediction

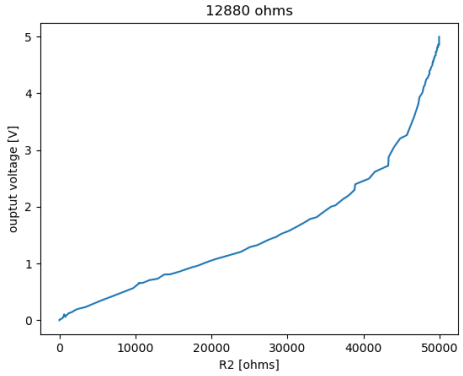


(d) experiment result

5.3.3 12880Ω load

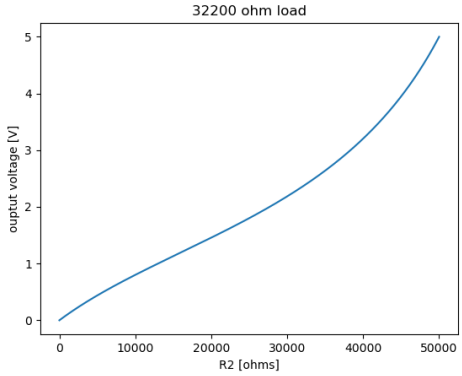


(e) prediction

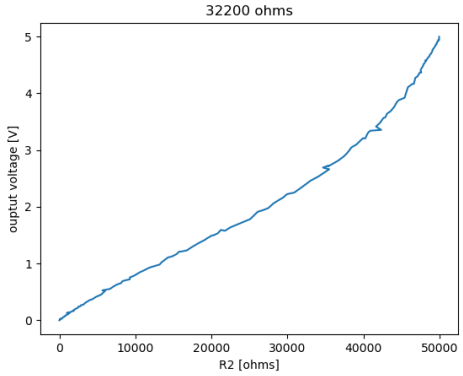


(f) experiment result

5.3.4 32200Ω load



(g) prediction



(h) experiment result

5.4 Evaluation of results

When comparing the plots, the measured plots are identical to their generated counterparts. Since the function describing the outputs of each test is already derived, it can be concluded that output voltage of a potentiometer with any load connected between output terminal and ground, can be approximated to function $V_{out}(R_2)$ ((7)).

6 Appendices

Appendix A

- GitHub repository of all software written for the project

Arduino firmware

```
const int PROBE = A3;
const int TRANS = 8;
bool TRANS_STATE = true;

void setup()
{
    pinMode(TRANS, OUTPUT);
```

```
    Serial.begin(115200);
}

void loop()
{
    unsigned int v_1 = 0;

    digitalWrite(TRANS, !TRANS_STATE);

    v_1 = analogRead(PROBE);

    digitalWrite(TRANS, TRANS_STATE);

    delay(50);

    Serial.print(v_1);
    Serial.print(",");
    Serial.print(analogRead(PROBE));
    Serial.println();

    delay(50);
}
```

Serial bus Python interface

```
#!/usr/bin/python

import sys
import serial

ser = serial.Serial('/dev/ttyACM0', 115200, timeout=None)

def read_data_to_csv(f):
    s = ser.readline()
    print(s)
    f.write(s.decode('utf-8'))

if __name__ == '__main__':
    if len(sys.argv) == 1:
        print('Usage: ser2csv <output filename> \n \
                default settings: \n\
                * port: /dev/ttyACM0 \n\
                * baud rate: 115200 \n\
                * timeout: None\n\
                \rterminate with CTRL+C')
        exit()

    f = open(sys.argv[1], 'w+')
    while 1:
        read_data_to_csv(f)
```

Python script for parsing and plotting data from csv file

```
#!/usr/bin/python
```

```
import sys
import numpy as np
from matplotlib import pyplot as plt

def parse_data(filename):

    data = np.genfromtxt(filename, delimiter=',',names=["x", "y"])

    for i in data:
        i['x'] = ((5/1024) * i['x']) * 10000
        i['y'] = (5/1024) * i['y']

    return data

def plot(data):
    plt.plot(data['x'], data['y'])

    plt.xlabel('R2 [ohms]')
    plt.ylabel('ouptut voltage [V]')
    plt.title(sys.argv[2])

    plt.savefig(sys.argv[2] + '.png')
    plt.show()


if __name__ == '__main__':
    if len(sys.argv) != 3:
        print('compute_data <input csv> <plot name>')
        exit()
    plot(parse_data(sys.argv[1]))
```

Python script for generating theoretical prediction graphs

```
#!/usr/bin/python

import sys
import math
import numpy as np
import matplotlib.pyplot as plt

r_prl = lambda r2, r3 : math.pow((1/r2 + 1/r3), -1)

v_out = lambda r1, r2, r3, vin : vin * (r_prl(r2,r3) / (r1 + r_prl(r2,r3)))

def generate_data(r1,r3,vin):
    x = np.arange(0,r1)
    y = np.zeros(r1)

    for i in range(1, r1):
        y[i] = v_out((r1 - i),i,r3,vin)

    return x,y

if __name__ == "__main__":

    if len(sys.argv) == 1:
```

```
    print('theoretical_prediction <parallel load> <ploc title> <filename>')
    sys.exit()

x,y = generate_data(50000, int(sys.argv[1]), 5)

plt.plot(x,y)
plt.xlabel('R2 [ohms]')
plt.ylabel('ouptut voltage [V]')
plt.title(sys.argv[2])
plt.savefig(sys.argv[3])
```

Appendix B

Sample raw data from 32200Ω experiment. Link to repository with full files `https://github.com/ikrzywda/experiment_utillities/tree/main/potentiometer_under_load/gathered_data`.

7 Bibliography

- "Potentiometer" *Wikipedia, The Free Encyclopedia* Wikimedia Foundation, Inc., 24 February 2021, `https://en.wikipedia.org/wiki/Potentiometer`