# Wheastone Bridge Optimaization Using Factorial Experiments

Gavin Gray and Christos Anastasiades

## 1.Circuit Diagram

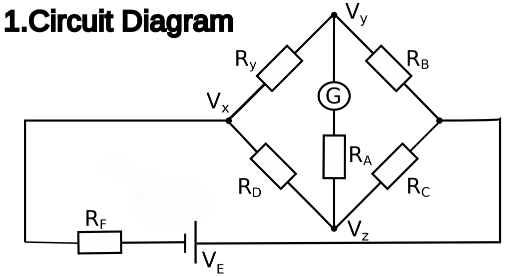



Figure 1: Wheatstone Bridge Circuit Diagram.

Aim: to detect 2Ω resistors as accurately as possible using a Wheatstone Bridge circuit

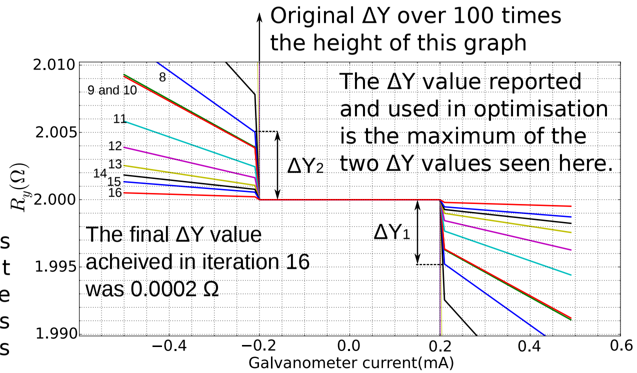
Problem: When galvanometer current is between ±0.2mA, the indicated value is zero. The resistance change that causes currents in this range to flow therefore cannot be detected.

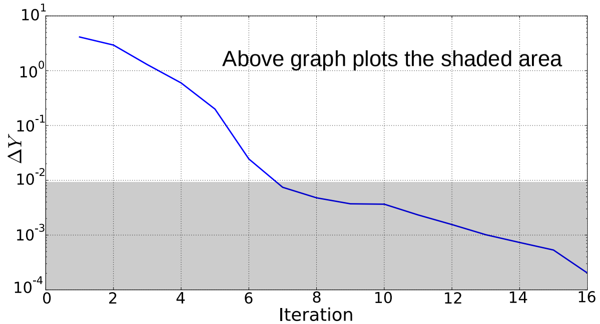
## 2.Iterative Design

The iterations were performed by running experiments varying each control factor by, initially, 10%. The result of these experiments was the ΔY value indicated on the right. Using this result it was then possible to regress the input factors into a model using RS/1 and use this model to find optimum values to minimise ΔY.

On each iteration the optimisation would find factors limited at the edges of the range chosen in the experiments. The range chosen for the next experiment was then based on this observation. These ranges were chosen based on intuition - a percentage of the value the factor was found to be limited at.

The graph on the right shows the ΔY acheived on each iteration. This had to be plotted on a logarithmic scale for the higher iterations to be visible.




## 3.Results

| ΔY: Optimisation 1 |
|---|
| $V_E$=33.810 V |
| $R_A$=1.0000 Ω |
| $R_B$=1.3413 Ω |
| $R_C$=1.2310 Ω |
| $R_D$=1.8356 Ω |
| $R_F$=1.0002 Ω |

| Power: Optimisation 2 |
|---|
| $V_E$=1.500 V |
| $R_A$=1.0006 Ω |
| $R_B$=3.6994 Ω |
| $R_C$=7.2199 Ω |
| $R_D$=3.9033 Ω |
| $R_F$=1.0001 Ω |

| Final Chosen Components | Component Tolerances (%) | Component Power rating (mW) | Max.(6σ) Dissipated power (mW) | Yield (σ) |
|---|---|---|---|---|
| $V_E$=1.500 V | N/A(-20) | | | 6.53 |
| $R_A$=3.00 Ω | ±1 | 00 | N/A | |
| $R_B$=12.70 Ω | ±1 | 00 | 115 | |
| $R_C$=40.20+36.0=76.2Ω | ±1 | 00 | 19.2 | |
| $R_D$=12.00Ω | ±1 | 00 | 3.1 | |
| $R_F$=1.00 Ω | ±1 | 00 | 12.4 | |
| | | | Total power=167 | |

To obtain the greatest accuracy possible the first set of iterations were simply optimising ΔY without considering any other design objectives. These are illustrated in section 2. The high voltage and low resistance made it an impractical design.

A common battery voltage of 1.5 V was chosen as a reasonable $V_E$. The power dissipated in each resistor was constrained below 0.1W in subsequent iterations.

The resulting design had a ΔY of 7 mΩ. Unfortunately, this design required 0.1% resistors to be viable. Only 1% resistors were available from online suppliers in the required sizes.

Setting a yield of 6σ as a design constraint allowed the design to become viable and the final Monte Carlo test produced a worst case 6σ ΔY of 70mΩ.

## Methods

1. Control factor ranges
2. Test specifications
3. Perform experiment
4. Check model fit
5. Optimise variables
6. Confirmation run

A final experiment is always required to validate the optimised design.

10-50% variation on factors was a safe rule of thumb; below which model accuracy was badly affected.

As an arbitrary number of experiments could be run a cubic model was built using a full factorial experiment to increase accuracy.

Fit of the model can be estimated by the reported $R^2$(fit of the model to data) value. This can be improved by applying a y-transform, variance weights or bisquare weights.

$R^2$ is composed of two square sums of observed and predicted responses:
Observed $SS_{obs}$ of real responses
Regressed $SS_{reg}$ of model

$$R^2 = \frac{SS_{reg}}{SS_{obs}}$$

## Calculations

It was necessary to calculate $R_C$ for each circuit to ensure it would balance at 2Ω. This was acheived in the Python code using the following equations, by asserting $R_y$ at 2Ω.

$$R_C = \frac{R_B \times R_D}{R_y} \qquad R_C = \frac{R_B \times R_D}{2}$$

$$R_y = 2\Omega$$

Power dissipation of each resistor and the circuit as a whole was added as a response in the Python code. Thevenin's theorem was applied at node $V_x$ to solve for voltages $V_y$ and $V_z$.

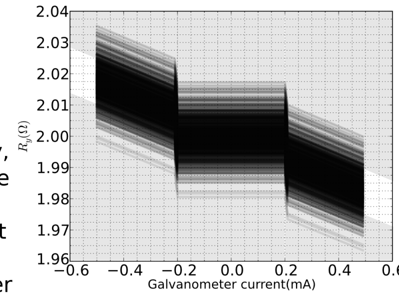
Yield was also added as a additional response, estimating the number of standard deviations able to detect 2Ω.

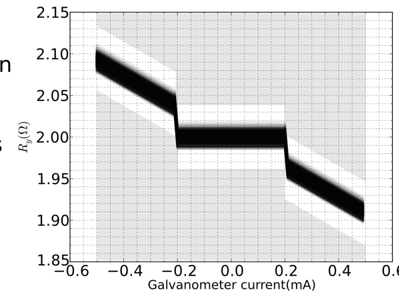
## Statistical Analysis

Using the distrib command it is possible to estimate the effect of resistor tolerances on the model. However, in order to optimise the yield of the circuit easily a Monte Carlo simulator was written in Python. Ideally, this would be run on the model because it involves many experiments.

The region of designs unable to detect a 2Ω resistor is indicated by shading in the top graph - showing the design after optimisation 2. Optimisation of yield was possible through the same regression model approach as in section 2 by using the Monte Carlo script to output worst case 6σ values for power dissipation and ΔY. The bottom graph is plotted in the same manner to the top, but for the final design.

For confirmation, a final Monte Carlo was also run.




**The final circuit is guaranteed to be able to detect a 2Ω resistor to within 70mΩ, while consuming 167mW when balanced and for a resistor cost of £0.047.**