

# MEng Thesis

## Circuit Sensing Breadboard

Josh Gordonson

July 10, 2015

## 1 Motivation

The breadboard has been a staple substrate for electronic construction over the last century. At the dawn of a growing interest in amateur radio, resourceful tinkerers used planks of wood to secure and ruggedize their electrical handiwork. Conductive nodes, such as nails or copper rails, were driven into the non-conductive boards, providing anchors and contact points that were electrically isolated from the rest of the circuit. Components were soldered or wire-wrapped to the nodes, and sometimes secured by non-energized nails or screws. This construction technique provided a lot of artistic freedom in circuit construction, but was time consuming and required relatively heavy hand tools such as a hammer or drill.

The solderless breadboard is the canonical tool given to students taking introductory courses in the field. Rather than driving nodes into arbitrary locations, component leads are inserted into contact points arranged on a grid that allow rapid semi-rigid construction of circuits with no other tools. The layout of a solderless breadboard is designed to be compatible with a plethora of powerful integrated circuits, enabling complex designs. Modern solderless breadboards have come a long way from their namesake wooden ancestors, but there is still room for improvement.

The shortcomings of solderless breadboards lie entirely within the art of breadboarding. The intent of breadboarding is to physically realize a circuit. Often, this involves designing or using a reference schematic to guide construction, but circuit improvisation is not uncommon. Inserting components and jumper wires into contact points is straightforward, but poor contacts, broken wires (inside insulation), and mis-inserting leads can plague designers for hours on end or even destroy components. A meticulous breadboarder can successfully realize a circuit without error, but for the uninitiated it is difficult to justify the additional time and care required to plan and build. Breadboarding is a skill that is learned over time, but small errors can lead to excessive frustration and turn students off to the field. To satisfy the requirements of the Masters in Engineering program, I propose a solution to some of the issues with the solderless breadboard.

## 2 Overview

A confident linkage between the the electrical and mechanical domains is required to construct a circuit. On larger scales, the mechanical structures (eg. contacts, wires, components) that create electrical circuits are visible in plain sight. It is simple and reliable, then, to determine the circuit representation of a mechanical structure that has no hidden connections. Breadboards often obscure connections due to their construction. The regular nature of a breadboard, a grid of contact points arranged in rails, makes it difficult to keep track of which rail is connected to what circuit node. When combined with the opaque nature of most components and the poor reliability of breadboard contacts, visually verifying complex circuits on a breadboard becomes infeasible. Many of the inherent problems with breadboards stem from this open-loop nature of breadboarding, where visual cues are not enough to determine the electrical circuit from the mechanical structure. A symptom of open-loop construction techniques is a mismatch between the mental model and the physical realization of the system at hand. I seek to close this loop by designing and implementing a circuit-sensing breadboard.

At first glance, the proposed problem is straightforward. A standard-issue breadboard has 130 conductive rails. Each of these rails can be connected to an array of ADCs (analog to digital converters) and DACs (digital to analog converters) controlled by a microcontroller. With access to every node in the circuit, it is possible to determine a constructed circuit's topology by iteratively stimulating and probing the circuit. Further, a schematic representation of the circuit topology can be drawn on an available display. This visual display provides an accurate depiction of the electrical circuit at hand, effectively closing the loop on breadboard construction. Designing a circuit-sensing breadboard requires a substantial amount of hardware and software infrastructure. Fortunately, the majority of this infrastructure is useful in its own right.

A hardware test-bench was constructed to interface each of the 130 rails on the breadboard. The test-bench is composed of a pcb-mounted breadboard, an ADC and DAC multiplexing board, a microcontroller, firmware to control sampling and stimulating the breadboard, and software to stream the sampling data to a computer for display. Effectively, the test-bench is a 130-channel digital breadboard-oscilloscope/function generator. Since software is faster to prototype with, the theoretical circuit topology work was first implemented in software. A circuit simulator test-bench was written to provide the circuit-sensing algorithm in development with data. This test-bench allows the circuit-sensing algorithm to probe, stimulate, and ground every node of a randomly generated circuit, as if the circuit were built on a breadboard attached to the hardware test bench mentioned above. The software test-bench also had the ability to print out a circuit network solution to a schematic display.

## 3 Work to Date

### 3.1 Software Test-bench

A software test-bench was written to quickly iterate and test the circuit-sensing algorithm. The test-bench uses NGSpice, an open-source circuit simulator, to simulate a circuit and return only the data that would be accessible to the hardware test bench. It includes a method to generate a random circuit with a specified number of nodes and types of circuit element, a method to insert probes, ground nodes, and voltage sources into an existing circuit, and a method to execute the simulation and format the results into a python-friendly output. [I sincerely regret using python to implement this algorithm and intend to port the algorithm to julia or another, friendlier, language.]

Circuits are described by netlists (text descriptions of a circuit) where each line specifies a new branch in the network. Each branch is composed of a circuit element (resistor, voltage source, etc.) with specified circuit parameter(s) (resistance, voltage, etc.), connected between a set of nodes. Netlists are convenient here because they allow for insertion of circuit elements, voltage sources, probes, and grounds with the addition of simple strings to a text file.

To write a random circuit netlist, the test-bench generates a random symmetric  $n$ -by- $n$  matrix for each desired circuit element type, where  $n$  is the desired number of nodes in the circuit, excluding 'ground'. The circuit elements explored in the scope of this thesis are resistors, capacitors, and inductors. Each matrix is interpreted as a graph connecting the nodes in one dimension of the matrix to the nodes in the other dimension of the matrix. The value at each location in the matrix is the circuit parameter for that particular element. The matrix is used to generate a list of strings that are then inserted into a spice netlist. At this point, the netlist only describes the random circuit generated and contains none of the 'machinery' required to reverse-engineer the circuit.

The circuit sensing algorithm conducts experiments on the circuit at hand and uses the data recorded to reverse-engineer the circuit. These experiments are implemented by simulating the effect of adding voltage sources, grounding nodes, reading node voltages, and reading the current through the voltage sources.

Voltage sources are added by inserting a voltage source into the existing circuit netlist with an AC or a DC voltage. Nodes are grounded by inserting a voltage source with zero DC voltage with respect to ground. In the hardware test bench, nodes are grounded by a MOSFET connecting each rail to ground. On the other hand, voltage sources are driven by a buffered DAC and multiplexed out to all of the nodes. Since we have access to the voltage source, we can measure the current supplied

by the buffered DAC and use that to gain information about the circuit in question. This is a better idea as there are fewer current sensors required.

### 3.2 Impedance network sensing algorithm

Given the ability to probe any node voltage, the ability to add a voltage source with a current meter to any node, and the ability to ground any node, an algorithm to reverse engineer any RLC network was written. The impedance network sensing algorithm works as follows:

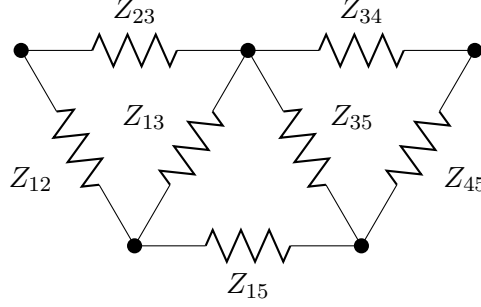


Figure 1: Example Circuit

- 1 - Pick a circuit branch  $[b_1]$  between two nodes  $[n_1]$  and  $[n_2]$  to identify
- 2 - Add a voltage source to  $[n_1]$  and ground every other node in the circuit
- 3 - Divide the voltage at  $[n_1]$ ,  $[V_{n_1}]$ , by the voltage source current  $[I_{VS}]$  to obtain the parallel impedance of all branches connected to  $[n_1]$ ,  $[Z_{1||}]$
- 4 - Add a voltage source to  $[n_2]$  and ground every node in the circuit except for  $[n_1]$  and  $[n_2]$
- 5 - Divide  $[Z_{1||}]$  by the resulting voltage measured,  $[V_{n_1}]$  to obtain the impedance of  $[b_1]$ ,  $[Z_{12}]$
- 6 - Repeat 1-5 for all pairs of nodes in the network

Step [5] is a bit magical, so let's look at a quick example from Figure 1: First, let's pick nodes 1 and 3, apply a voltage source to node 1 and ground all of the other nodes.

The resulting  $Z_{1||} = Z_{13} || (Z_{12} || Z_{15}) = \frac{Z_{13}(Z_{12} || Z_{15})}{Z_{13} + (Z_{12} || Z_{15})}$

Now, we apply a voltage source to node 3, open node 1, and ground all of the other nodes.

The voltage that appears at node 1,  $V_{n_1} = \frac{Z_{12} || Z_{15}}{Z_{13} + Z_{12} || Z_{15}}$

Finally,  $\frac{Z_{1||}}{V_{n_1}} = \frac{\frac{Z_{13}(Z_{12} || Z_{15})}{Z_{13} + (Z_{12} || Z_{15})}}{\frac{Z_{12} || Z_{15}}{Z_{13} + Z_{12} || Z_{15}}} = Z_{13}$

To determine if there are multiple element types [in parallel] between two nodes, we can select a few frequencies to scan through and analyze the resulting change in impedance.

The impedance of a parallel RLC 'tank' circuit can be characterized and analyzed over all frequencies.

$$Z_C = \frac{1}{j\omega C} \quad Z_R = R \quad Z_L = j\omega L \quad (1)$$

$$Z_{RLC} = Z_C || Z_R || Z_L = \frac{1}{j\omega C + \frac{1}{R} + \frac{1}{j\omega L}} = \frac{j\omega RL}{-\omega^2 RLC + j\omega L + R} \quad (2)$$

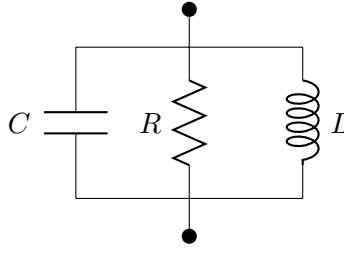
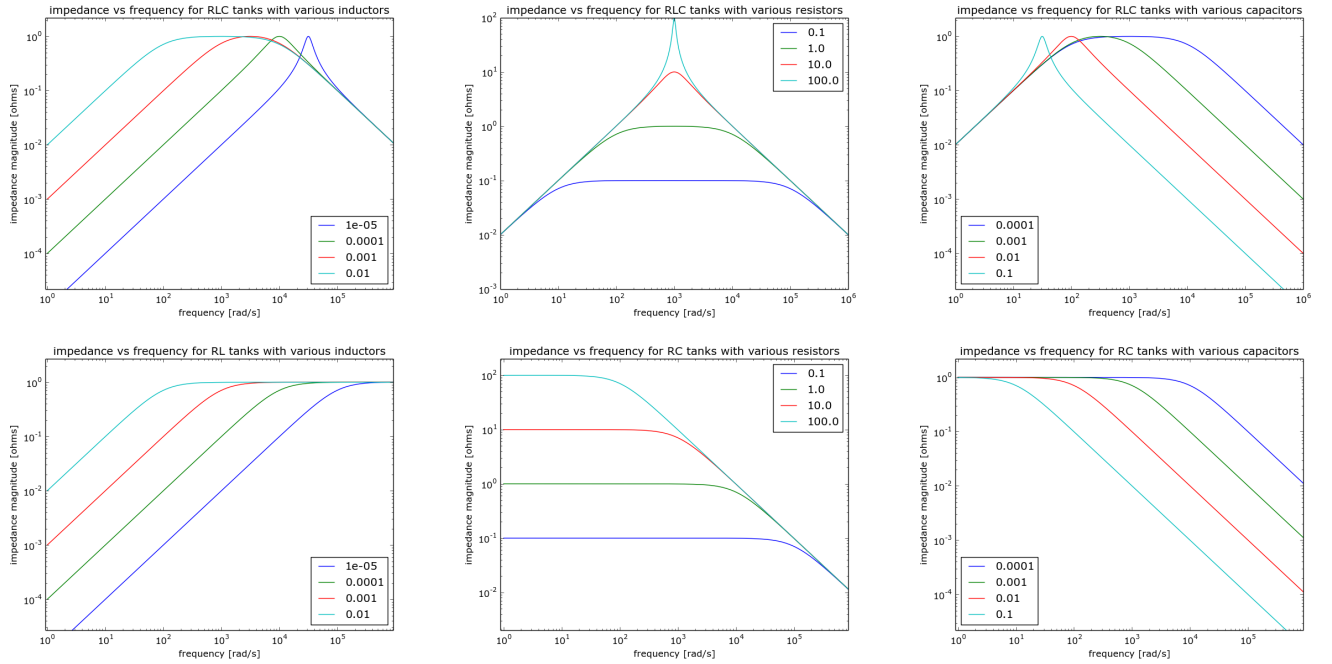


Figure 2: Example RLC Tank Circuit

$Z_{RLC}$  has a zero at  $\frac{1}{RL}$  and two poles at  $\frac{-L \pm \sqrt{L^2 + 4R^2 LC}}{-2RLC}$



If we look at the plots above, we can see that varying the resistance changes the maximum impedance over all frequencies, varying the inductance drops the asymptotic impedance at low frequencies, and varying the capacitance drops the asymptotic impedance at high frequencies. We can imagine there are three regimes on the impedance vs frequency plot that divide the operation of an RLC tank into three elements:

When the slope is  $+1$ , the tank behaves like an inductor  $L = |Z|/(j\omega)$ . When the slope is  $-1$ , the tank behaves like a capacitor  $C = j\omega|Z|$ . When the slope is  $0$ , the tank behaves like a resistor  $R = |Z|$ . This behavior can be exploited to reveal what elements are in parallel with one another. First, sample at the highest possible driving frequency and a decade lower in frequency. If the impedance increases by a decade, then there is a capacitor with capacitance  $C = j\omega|Z|$ . Then, sample at the lowest reasonable driving frequency and a decade higher in frequency. If the impedance increases by a decade, then there is an inductor with inductance  $L = |Z|/(j\omega)$ . If there is an inductor and a capacitor, find the equivalent parallel resistance by sampling at  $\omega = \sqrt{LC}^{-1}$ . If there is only a capacitor, find the equivalent parallel resistance by sampling at the lowest reasonable driving frequency. If there is only an inductor, find the equivalent parallel resistance by sampling at the highest reasonable driving frequency. If there is no capacitor and no inductor, sample mid-band. An open circuit between two nodes will be indicated by an extremely high resistive reading and a low capacitance reading.

Diodes are indicated by asymmetric resistances. For example, if  $R_{12} = 100\Omega$  and  $R_{21} = 1\Omega$ , there is likely a diode between nodes 1 and 2.

## References