# WIP: A Near Real-Time Circuit Simulation and Analysis Framework for Undergraduate Engineering Education – JSIM 2.0

**Mr. John Francis Simonis, The Ohio State University at Marion**

John Simonis is a Computer Engineering student at The Ohio State University with research interests spanning embedded systems, bioelectronics, circuit simulation, and digital signal processing. His work focuses on developing innovative hardware and software solutions that bridge theoretical learning with practical application. John has contributed to projects in memristive bioelectronics, assistive technologies for individuals with visual impairments, atmospheric water harvesting, and engineering education. His projects often emphasize accessibility, sustainability, and experiential learning through open-source design.

**Dr. Qudsia Tahmina, The Ohio State University at Marion**

Dr. Qudsia Tahmina, The Ohio State University at Marion

Dr. Qudsia Tahmina is an Associate Professor of Practice at The Ohio State University at Marion, where she specializes in instruction for first- and second-year Engineering courses. Beyond the classroom, she serves as the Engineering Program Coordinator, leading strategic scheduling and supporting community outreach initiatives. Dr. Tahmina also plays a pivotal role in curriculum development as the ABET assessment lead, overseeing revisions and accreditation efforts for the Engineering Technology program.

# WIP: A Near Real-Time Circuit Simulation and Analysis Framework for Undergraduate Engineering Education – JSIM 2.0

**Abstract**

A persistent challenge in undergraduate engineering education lies in developing students' intuitive understanding of circuit behavior [1]. Contemporary learners are inherently experiential and benefit from pedagogical approaches that emphasize simulation, experimentation, and programming-based exploration [2]. Traditional instructional methods, which rely heavily on static circuit diagrams and manual analytical techniques, often fail to effectively bridge the gap between theoretical concepts and practical hardware implementation [1].

The original JSIM platform was conceived to address this disconnect by providing an interactive, real-time circuit simulation platform that unites theoretical circuit analysis with applied experimentation. Developed entirely in C++ using only standard libraries, JSIM performed nodal and mesh analyses efficiently, even on highly resource-constrained systems such as low-power microcontrollers.

Through dynamic matrix construction and adaptive memory management, the software sought to enable students to engage with circuit fundamentals in a manner consistent with computational and mathematical frameworks introduced in early engineering curricula.

The last implementation of JSIM supported fundamental circuit elements, including voltage sources, current sources, resistors, capacitors, and inductors, facilitating the modeling and analysis of essential resistive and reactive networks. These capabilities make the platform particularly effective for investigating the frequency and time-domain behavior of fundamental filters, such as low-pass, high-pass, and band-pass configurations.

Informed by educator feedback and prior conference discussions, JSIM 2.0 seeks to incorporate parts of the original framework while integrating basic hardware-based signal acquisition, digital signal processing functionality, and interoperability with other circuit simulators, most notably Falstad [4].

This iteration seeks to incorporate a Fast Fourier Transform (FFT)-based spectral visualization as well as waveform capture at user-defined probe points, enabling near real-time comparison between simulated and experimentally measured circuit responses.

By consolidating simulation, analysis, and measurement within a unified, portable platform, JSIM 2.0 reduces reliance on traditional laboratory instrumentation, streamlines instructional delivery, and enhances experiential learning outcomes. Classroom evaluations of JSIM 2.0 are forthcoming to assess its effectiveness in improving student comprehension, engagement, and instructional efficiency. This integrated approach strengthens the connection between

programming and electrical engineering concepts, fostering a cohesive and application-driven understanding of embedded system design [2].

**Keywords**
Engineering Education, Circuit Simulation, Filter Design, Pedagogical Tools, Circuit Analysis

## Introduction

Introductory circuits courses represent a fundamental component of undergraduate electrical and computer engineering curricula, yet they consistently present conceptual challenges for students [3]. While traditional instruction emphasizes rigorous analytical techniques such as nodal and mesh analysis, basic filter design, and introductory signal processing, many students struggle to develop an intuitive understanding of how circuit behavior evolves in response to changing inputs, component values, and frequency. This disconnect between analytical formalism and physical behavior can persist well beyond the introductory sequence of courses, limiting student confidence and hindering conceptual transfer in subsequent coursework.

This presents an opportunity for alternative instructional approaches. Prior work has shown that students generally respond positively to hands-on activities, particularly when these experiences are combined with simulation-based laboratories [1]. However, in many curricula, these approaches are treated as separate instructional strategies, with laboratory exercises typically emphasizing either simulation or hands-on experimentation, but rarely both in a sufficiently integrated manner [2].

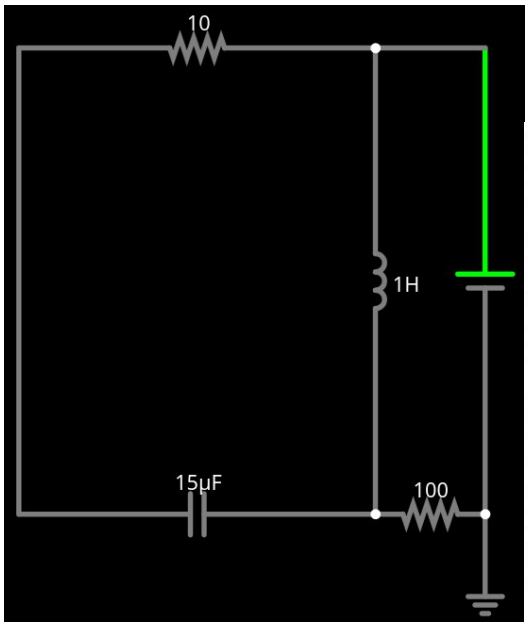
JSIM was originally developed to improve the accessibility of circuit simulation both inside and outside the classroom. While effective in supporting computational exploration, the initial implementation struggled to fully bridge the gap between simulation and hands-on laboratory work. JSIM 2.0 seeks to address these limitations by placing a stronger emphasis on direct comparison between simulated results and real-world measurements obtained from the same hardware, while retaining the accessibility and portability of the original platform.

## Methods

JSIM 2.0 is implemented as a two-part architecture: a host-side translation utility that converts schematic descriptions (e.g., Falstad) into a compact, machine-readable netlist, and a device-side execution engine running on a Raspberry Pi Pico 2 (RP2350). The design goal is to enable rapid iteration between circuit topology authoring (on a desktop) and portable, resource-constrained time-domain simulation and measurement (on the microcontroller), with a consistent representation of circuit connectivity across both modes.

At a high level, the workflow is as follows: (1) a circuit is authored in Falstad; (2) the host utility derives electrical nets from Falstad’s coordinate/wire representation and emits a simplified netlist; (3) the Pico firmware ingests the netlist over USB serial and builds an internal circuit model; and (4) the user selects either simulation (MNA-based transient analysis) or measurement

(analog-to-digital converter (ADC) sampling), optionally applies digital filters, and generates an FFT-based spectrum to compare simulated and measured responses, or exports raw data for Excel-based visualization and analysis.




```
$ 1 0.000005 10.20027730826997 50 5 43 5e-11
r 176 80 384 80 0 10
w 176 80 176 352 0
c 384 352 176 352 4 0.000015 -10 -10 0
l 384 80 384 352 0 1 0 0
v 448 352 448 80 0 0 40 5 0 0 0.5
r 384 352 448 352 0 100
w 384 80 448 80 0
g 448 352 448 400 0 0
o 3 64 0 4099 20 0.05 0 2 3 3
o 2 64 0 4099 20 0.025 1 2 2 3
o 0 64 0 4099 0.3125 0.025 2 2 0 3
38 2 F1 0 0.000001 0.000101 -1 Capacitance
38 3 F1 0 0.01 1.01 -1 Inductance
38 0 F1 0 1 101 -1 Resistance
```

Figure 1: Example of a Falstad Circuit Problem and Raw Coordinate/Wire Representation

```
Falstad circuit (pretty view)
================================
Header: $ 1 0.000005 10.20027730826997 50 5 43 5e-11

Parts list
--------------------------------
V1: Voltage source DC 5 V  (p=GND, n=Net1)
R1: Resistor 10 Ω  (a=Net0, b=Net1)
R2: Resistor 100 Ω  (a=Net2, b=GND)
C1: Capacitor 15uF (1.5e-05 F), Vinit=-10 V  (a=Net2, b=Net0)
L1: Inductor 1H (1 H), Iinit=0 A  (a=Net1, b=Net2)
GND1: Ground symbol -> GND

Nets (from wires)
--------------------------------
Net0
Net1
Net2
GND

Unrecognized lines (preserved)
--------------------------------
o 3 64 0 4099 20 0.05 0 2 3 3
o 2 64 0 4099 20 0.025 1 2 2 3
o 0 64 0 4099 0.3125 0.025 2 2 0 3
38 2 F1 0 0.000001 0.000101 -1 Capacitance
38 3 F1 0 0.01 1.01 -1 Inductance
38 0 F1 0 1 101 -1 Resistance

Pico NETLIST (send this for simulation!)
--------------------------------
BEGIN_NETLIST
# one record per line: TYPE key=value key=value ...
# nets use names: GND or Net<N>
VSRC name=V1 p=GND n=Net1 waveform=DC amp_v=5 freq_hz=NA offset_v=NA phase_deg=NA duty=NA
R name=R1 a=Net0 b=Net1 ohms=10
R name=R2 a=Net2 b=GND ohms=100
C name=C1 a=Net2 b=Net0 farads=1.5e-05 v_init=-10
L name=L1 a=Net1 b=Net2 henries=1 i_init=0
GND name=GND1 net=GND
NET name=Net0
NET name=Net1
NET name=Net2
NET name=GND
END_NETLIST
```

Figure 2: Host-Side Translation of Example Falstad Circuit

Falstad's internal schematic format is inherently geometry-based, representing components by endpoints in a 2D coordinate space and defining connectivity primarily through explicit wire segments [4]. The host translation layer bridges this representation to the Pico by reconstructing electrical nets from wire connectivity and mapping each component terminal to a named net. This translation step is essential because the microcontroller-side solver operates on node indices and net connectivity, not on schematic coordinates or graphical primitives.

The translator produces a compact, line-based netlist framed by **BEGIN_NETLIST** and **END_NETLIST**. Each line specifies a single circuit element using a type identifier (e.g., resistor, capacitor, voltage source) followed by key-value parameters describing connectivity and element values. All schematic-specific details from Falstad, such as coordinates, drawing codes, and wire segments, are removed, leaving only the electrical information required for simulation: element identity, connected nets, and parameter values.

Test points provide a simple mechanism for selecting observation nodes. One test point label is reserved to designate the simulation output node, allowing the solver to identify which node voltage to record. This approach decouples schematic annotation from the solver's internal node indexing and enables the same netlist to support both simulated output selection and experimental probing in hardware.

Transient simulation on the Pico is performed using Modified Nodal Analysis (MNA) [5]. At each time step, the solver constructs and solves a linear system to determine node voltages (with the ground node removed) and currents through independent voltage sources. Circuit elements are incorporated using standard MNA stamping: resistors contribute conductance terms, and independent voltage sources are handled through matrix augmentation with source values evaluated per time step [5]. The simulator supports DC, sinusoidal, noise-based voltage sources, and more.

Reactive elements are modeled using trapezoidal-rule companion models for time-domain analysis. Capacitors and inductors are represented as equivalent conductances in parallel with history-dependent current sources that capture prior state. After each solve, these history terms are updated to preserve continuity across time, enabling stable transient simulation of linear resistive and reactive circuits on resource-constrained hardware [6].

Frequency-domain analysis is provided through an integrated FFT pipeline applied to both simulated and measured time-domain data. Collected samples are first conditioned by removing any DC offset and applying a Hann window to reduce spectral leakage [7]. A radix-2 fast Fourier transform is then computed on the largest power-of-two subset of the data, and window coherent-gain compensation is applied to preserve meaningful amplitude estimates. Spectral magnitudes are reported in both Vrms and dBV (referenced to 1 Vrms), allowing students to directly observe frequency-dependent attenuation and gain without peak normalization masking relative changes. This presentation supports intuitive interpretation of filter behavior, resonance, and bandwidth in both simulated and physical circuits.

The system is operated through a simple serial menu that guides users through selecting simulation or measurement modes, defining sampling parameters, applying digital filters, and generating time- or frequency-domain outputs. Identical processing steps can be applied to simulated and measured data, enabling direct, side-by-side comparison under consistent sampling and analysis conditions. From an educational perspective, this unified workflow reinforces the connection between analytical models, numerical simulation, and real-world circuit behavior. Each functional layer of JSIM 2.0 corresponds to a distinct learning objective: the schematic-to-netlist translation maps to circuit topology and connectivity; MNA-based simulation targets nodal analysis and reactive element modeling; digital filtering reinforces filter design concepts; and FFT-based spectral output develops frequency-domain intuition. By co-locating these capabilities, the platform enables students to experience the full analysis pipeline; from circuit construction to experimental verification; in a single, coherent environment.

**Results**

To evaluate the effectiveness of JSIM 2.0, the simulator was provided with a constructed test circuit consisting of two sinusoidal voltage sources connected in series, each with an amplitude of 5 V and operating at frequencies of 50 Hz and 20 Hz, respectively. These sources were connected in series with two 1 kΩ resistors configured as a simple voltage divider, with the test probe attached at the divider output node.

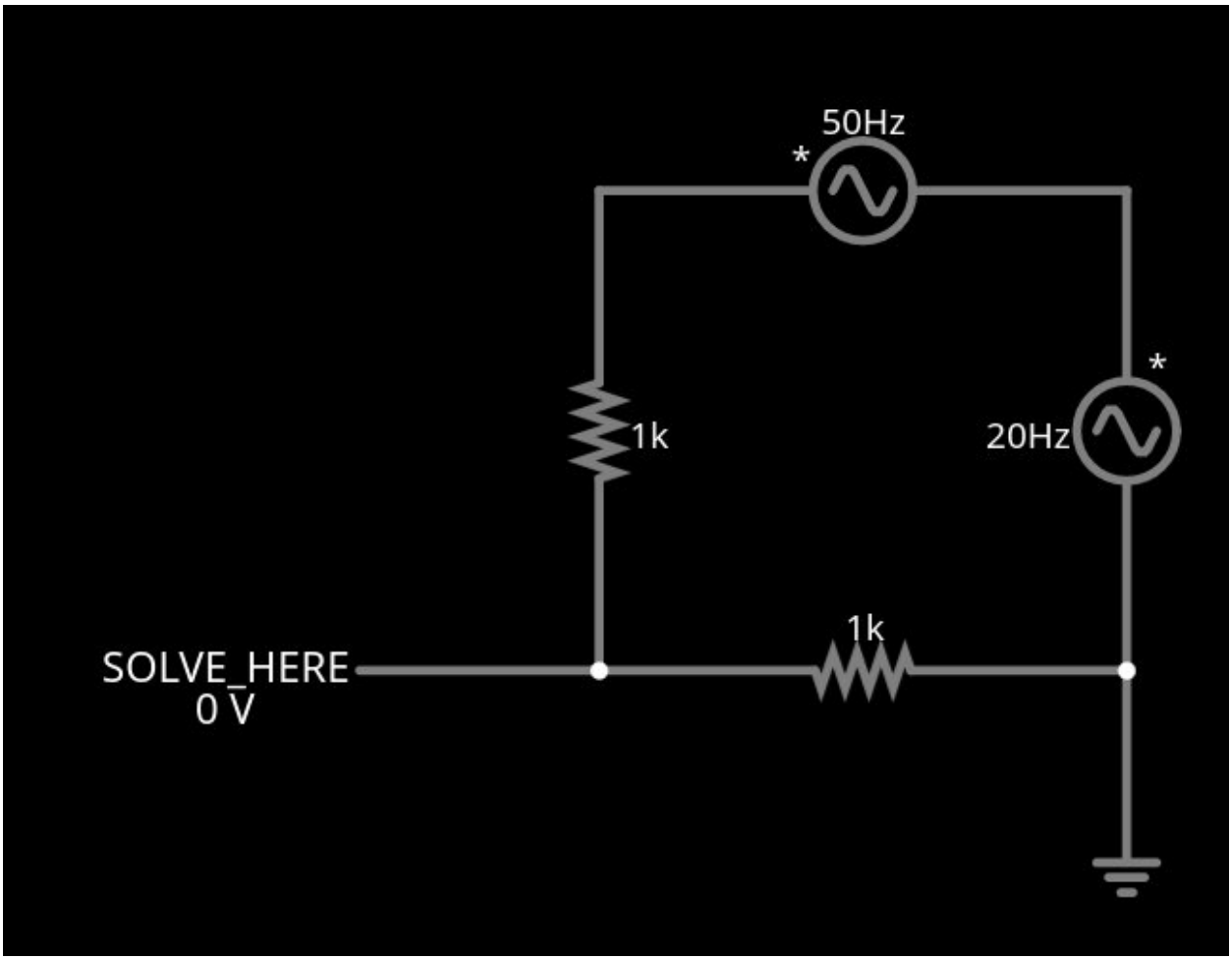



Figure 3. Evaluation Circuit Built in Falstad

The circuit’s coordinate- and wire-based schematic representation was exported from Falstad and used as the input to the host-side translation layer. This representation preserves the original schematic connectivity while enabling automated extraction of electrical nets for subsequent simulation and analysis on the microcontroller platform.

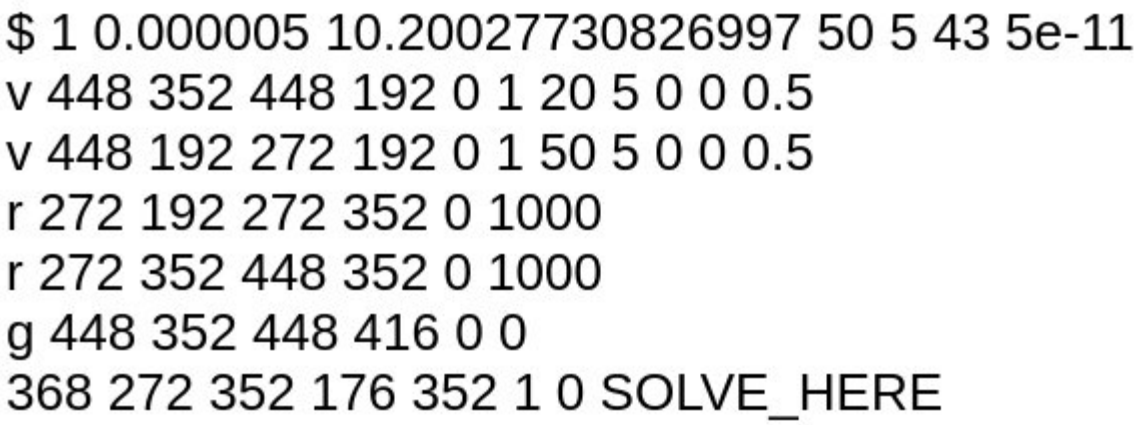



Figure 4. Falstad's Text Output for The Evaluation Circuit

```
Falstad circuit (pretty view)
================================
Header: $ 1 0.000005 10.20027730826997 50 5 43 5e-11

Parts list
--------------------------------
V1: Voltage source SINE amp=5 V, f=20 Hz (offset=0 V, phase=0°)  (p=GND, n=Net3)
V2: Voltage source SINE amp=5 V, f=50 Hz (offset=0 V, phase=0°)  (p=Net3, n=Net0)
R1: Resistor 1000 Ω  (a=Net0, b=Net1)
R2: Resistor 1000 Ω  (a=Net1, b=GND)
GND1: Ground symbol -> GND
TP1: Test point 'SOLVE_HERE' -> Net1

Nets (from wires)
--------------------------------
Net0
Net1
GND
Net3

Unrecognized lines (preserved)
--------------------------------
(none)

Pico NETLIST (send this for simulation!)
--------------------------------
BEGIN_NETLIST
# one record per line: TYPE key=value key=value ...
# nets use names: GND or Net<N>
VSRC name=V1 p=GND n=Net3 waveform=SINE amp_v=5 freq_hz=20 offset_v=0 phase_deg=0 duty=NA
VSRC name=V2 p=Net3 n=Net0 waveform=SINE amp_v=5 freq_hz=50 offset_v=0 phase_deg=0 duty=NA
R name=R1 a=Net0 b=Net1 ohms=1000
R name=R2 a=Net1 b=GND ohms=1000
GND name=GND1 net=GND
TP name=TP1 net=Net1 label=SOLVE_HERE
NET name=Net0
NET name=Net1
NET name=GND
NET name=Net3
END_NETLIST
```

Figure 5: Host-Side Translation of Evaluation Circuit

The resulting machine-readable netlist was transmitted to the Pico over a USB serial connection operating at 115200 baud. Upon successful receipt and parsing of the netlist, the device

initialized its internal circuit model and presented the user with an interactive menu interface, from which simulation and measurement operations could be selected.

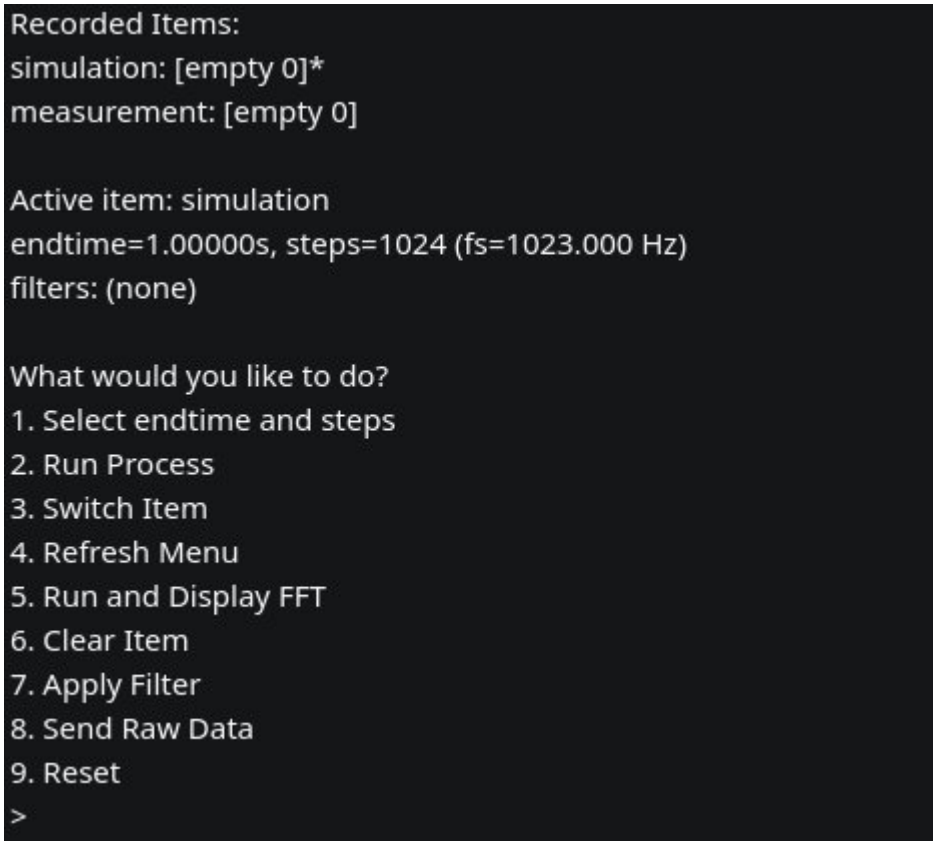



Figure 6: Simple Serial Menu Transmitted from Pico

From this menu, the simulation time base was configured with an end time of 1 s, and an appropriate number of simulation steps was selected. A total of 200 samples was used, as the corresponding Nyquist frequency was sufficient to capture the behavior of the 20 Hz and 50 Hz source components without unnecessary oversampling.

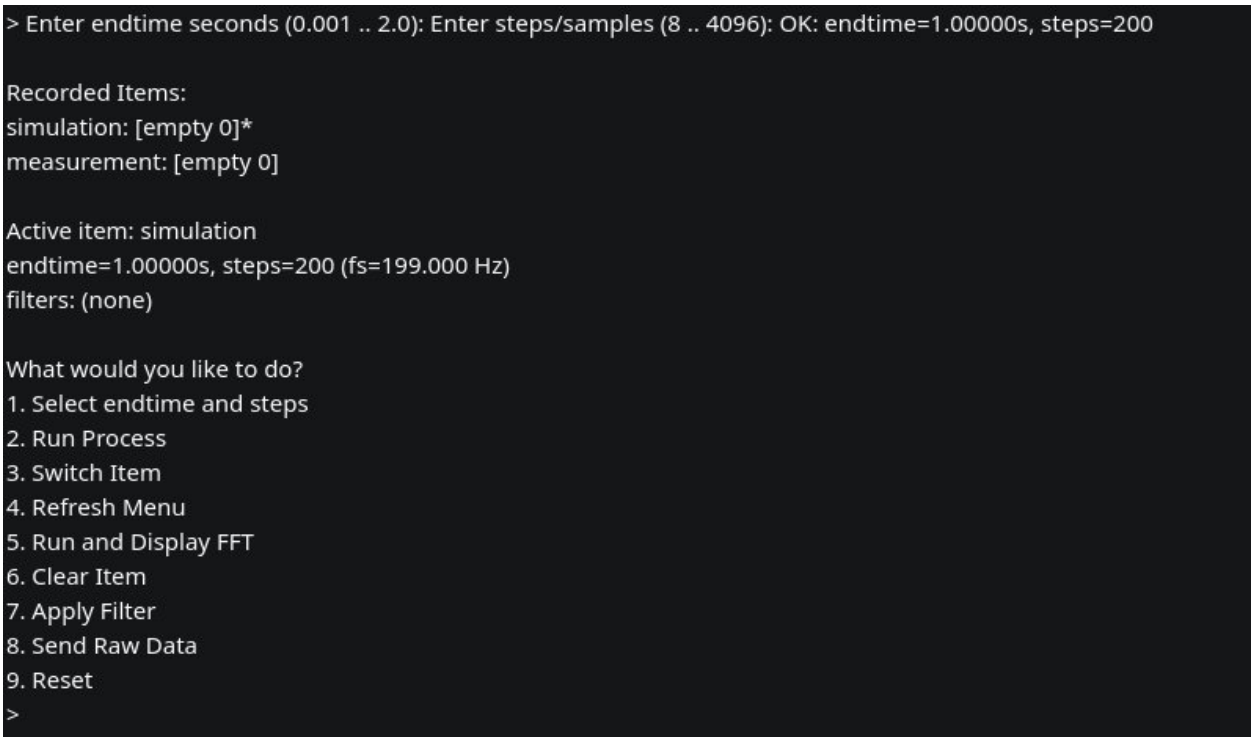



Figure 7: Simple Serial Menu Showing Updated Timing

The simulation was then executed, during which the circuit response was evaluated at each time step and a total of 200 time-domain samples were computed and stored in memory. These samples were retained for subsequent processing, including filtering, frequency-domain analysis, and comparison with measured data.

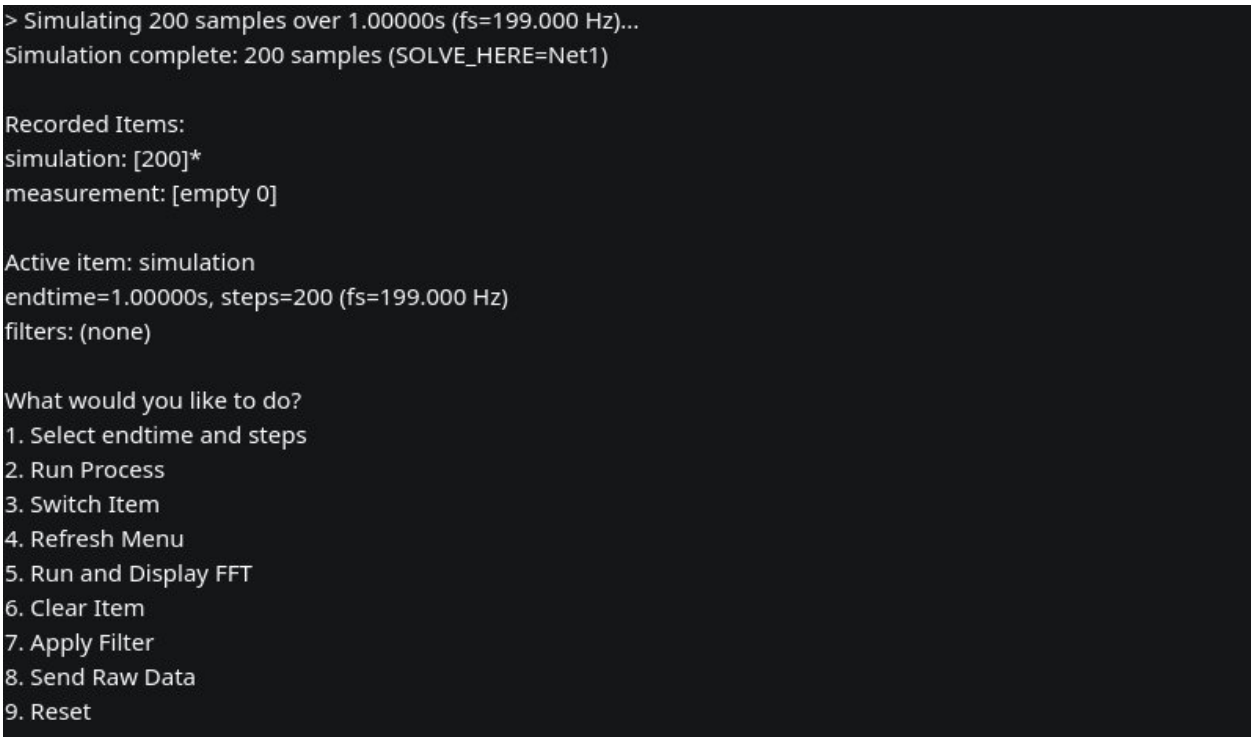



Figure 8: Simple Serial Menu Showing Simulation Completion

An FFT was then computed from the stored time-domain samples to visualize the frequency content of the simulated signal and to identify the spectral components corresponding to the two sinusoidal sources.

```
ASCII Spectrum (0..Nyquist), N=128, fs=199.000 Hz
Each line: freq(Hz) | bar (dBV, 0 dBV = 1 Vrms) | Vrms
     0.0 | ##############################    -28.0 dBV  (0.0396481 Vrms)
     1.6 | ##############################    -34.3 dBV  (0.0192852 Vrms)
     3.1 | ################    -62.9 dBV  (0.000713633 Vrms)
     4.7 | ################    -61.9 dBV  (0.000800839 Vrms)
     6.2 | #################    -59.8 dBV  (0.00102712 Vrms)
     7.8 | ##################    -57.0 dBV  (0.00140814 Vrms)
     9.3 | ####################    -53.8 dBV  (0.00204093 Vrms)
    10.9 | ####################    -50.0 dBV  (0.00317925 Vrms)
    12.4 | ######################    -45.3 dBV  (0.00542483 Vrms)
    14.0 | ########################    -39.4 dBV  (0.0106628 Vrms)
    15.5 | ##############################    -31.5 dBV  (0.0266540 Vrms)
    17.1 | ###################################    -18.8 dBV  (0.114222 Vrms)
    18.7 | ################################################      6.6 dBV  (2.14611 Vrms)
    20.2 | ##########################################################     10.9 dBV  (3.49439 Vrms)
    21.8 | ################################################      3.1 dBV  (1.43660 Vrms)
    23.3 | ################################    -25.1 dBV  (0.0555954 Vrms)
    24.9 | ############################    -36.9 dBV  (0.0142943 Vrms)
    26.4 | ######################    -45.1 dBV  (0.00556981 Vrms)
    28.0 | ####################    -51.4 dBV  (0.00267629 Vrms)
    29.5 | ##################    -56.6 dBV  (0.00148729 Vrms)
    31.1 | #################    -60.6 dBV  (0.000929870 Vrms)
    32.6 | ###############    -63.4 dBV  (0.000673551 Vrms)
    34.2 | ###############    -64.7 dBV  (0.000582251 Vrms)
    35.8 | ###############    -64.2 dBV  (0.000615704 Vrms)
    37.3 | ################    -62.2 dBV  (0.000773335 Vrms)
    38.9 | #################    -59.0 dBV  (0.00112378 Vrms)
    40.4 | ###################    -54.8 dBV  (0.00181112 Vrms)
    42.0 | ######################    -49.7 dBV  (0.00326103 Vrms)
    43.5 | ########################    -43.5 dBV  (0.00669011 Vrms)
    45.1 | ############################    -35.5 dBV  (0.0168577 Vrms)
    46.6 | ##################################    -23.9 dBV  (0.0637763 Vrms)
    48.2 | ################################################      2.8 dBV  (1.37322 Vrms)
    49.8 | ##########################################################     10.8 dBV  (3.47787 Vrms)
    51.3 | ################################################      6.9 dBV  (2.21168 Vrms)
    52.9 | ####################################    -17.2 dBV  (0.138425 Vrms)
    54.4 | ################################    -30.1 dBV  (0.0313280 Vrms)
    56.0 | ############################    -38.2 dBV  (0.0123477 Vrms)
    57.5 | ########################    -44.1 dBV  (0.00621854 Vrms)
    59.1 | ######################    -48.8 dBV  (0.00361306 Vrms)
    60.6 | ####################    -52.7 dBV  (0.00230766 Vrms)
    62.2 | ###################    -56.1 dBV  (0.00157352 Vrms)
    63.7 | ##################    -58.9 dBV  (0.00113014 Vrms)
    65.3 | ################    -61.5 dBV  (0.000843248 Vrms)
    66.9 | ###############    -63.7 dBV  (0.000650164 Vrms)
    68.4 | ##############    -65.8 dBV  (0.000512113 Vrms)
    70.0 | ##############    -67.6 dBV  (0.000416909 Vrms)
    71.5 | #############    -69.3 dBV  (0.000342827 Vrms)
    73.1 | ############    -70.9 dBV  (0.000286160 Vrms)
    74.6 | ############    -72.4 dBV  (0.000240444 Vrms)
    76.2 | ############    -73.6 dBV  (0.000208806 Vrms)
    77.7 | ###########    -74.8 dBV  (0.000181786 Vrms)
    79.3 | ###########    -76.0 dBV  (0.000159193 Vrms)
    80.8 | ##########    -76.9 dBV  (0.000142184 Vrms)
    82.4 | ##########    -77.9 dBV  (0.000127345 Vrms)
    84.0 | ##########    -78.8 dBV  (0.000114282 Vrms)
    85.5 | #########    -79.6 dBV  (0.000105269 Vrms)
    87.1 | #########    -80.3 dBV  (9.65111e-05 Vrms)
    88.6 | #########    -81.0 dBV  (8.91121e-05 Vrms)
    90.2 | ########    -81.3 dBV  (8.62227e-05 Vrms)
    91.7 | ########    -82.1 dBV  (7.81940e-05 Vrms)
    93.3 | ########    -82.0 dBV  (7.93893e-05 Vrms)
    94.8 | ########    -82.9 dBV  (7.19467e-05 Vrms)
    96.4 | ########    -83.1 dBV  (7.01205e-05 Vrms)
    97.9 | ########    -82.7 dBV  (7.33500e-05 Vrms)
```

Figure 9: Output of The FFT Shown Via Pico's ASCII Spectrum Functionality

A high-pass filter with a cutoff frequency of 35 Hz was subsequently applied to the simulated data to demonstrate frequency-dependent attenuation. This filtering operation selectively suppressed the lower-frequency 20 Hz component while preserving the higher-frequency 50 Hz component, enabling clear visualization of the filter's effect in the frequency domain.

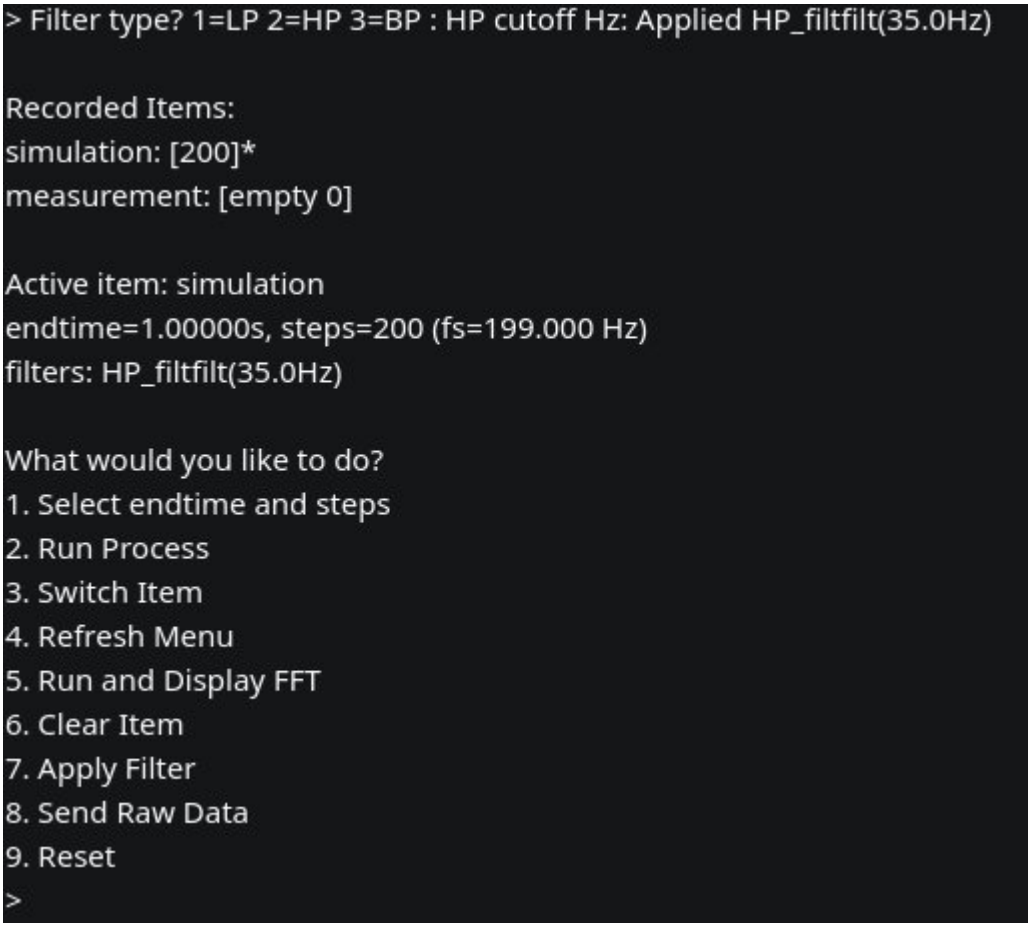


Figure 10: Simple Serial Menu Showing Filter Creation

A second FFT was then generated from the filtered signal to illustrate the resulting attenuation of the lower-frequency component and to highlight the effect of the high-pass filter on the signal's spectral content.

```
ASCII Spectrum (0..Nyquist), N=128, fs=199.000 Hz
Each line: freq(Hz) | bar (dBV, 0 dBV = 1 Vrms) | Vrms
     0.0 | ##############################    -38.4 dBV  (0.0120420 Vrms)
     1.6 | ##########################    -44.0 dBV  (0.00632103 Vrms)
     3.1 | ##########    -78.0 dBV  (0.000126415 Vrms)
     4.7 | ###########    -75.9 dBV  (0.000160529 Vrms)
     6.2 | ############    -74.2 dBV  (0.000195507 Vrms)
     7.8 | ##############    -71.8 dBV  (0.000256059 Vrms)
     9.3 | ###############    -68.7 dBV  (0.000366799 Vrms)
    10.9 | #################    -64.8 dBV  (0.000575662 Vrms)
    12.4 | ###################    -60.0 dBV  (0.00099835 Vrms)
    14.0 | ######################    -54.0 dBV  (0.00199059 Vrms)
    15.5 | ##########################    -46.0 dBV  (0.00502454 Vrms)
    17.1 | ##############################    -33.3 dBV  (0.0216380 Vrms)
    18.7 | ################################################     -7.8 dBV  (0.407151 Vrms)
    20.2 | ##############################################     -3.6 dBV  (0.662869 Vrms)
    21.8 | ##########################################    -11.3 dBV  (0.272550 Vrms)
    23.3 | ############################    -39.5 dBV  (0.0105692 Vrms)
    24.9 | ########################    -51.3 dBV  (0.00273314 Vrms)
    26.4 | ###################    -59.3 dBV  (0.00107780 Vrms)
    28.0 | ################    -65.5 dBV  (0.000529473 Vrms)
    29.5 | ##############    -70.3 dBV  (0.000306803 Vrms)
    31.1 | ############    -73.7 dBV  (0.000205395 Vrms)
    32.6 | ###########    -75.6 dBV  (0.000165372 Vrms)
    34.2 | ###########    -75.8 dBV  (0.000162211 Vrms)
    35.8 | ############    -74.3 dBV  (0.000192378 Vrms)
    37.3 | ##############    -71.6 dBV  (0.000261555 Vrms)
    38.9 | ################    -68.0 dBV  (0.000397557 Vrms)
    40.4 | ##################    -63.7 dBV  (0.000655853 Vrms)
    42.0 | ###################    -58.5 dBV  (0.00119367 Vrms)
    43.5 | ######################    -52.2 dBV  (0.00246091 Vrms)
    45.1 | ##########################    -44.1 dBV  (0.00621309 Vrms)
    46.6 | ################################    -32.6 dBV  (0.0235216 Vrms)
    48.2 | ##############################################     -5.9 dBV  (0.506551 Vrms)
    49.8 | ####################################################      2.2 dBV  (1.28292 Vrms)
    51.3 | ##############################################     -1.8 dBV  (0.815844 Vrms)
    52.9 | ###################################    -25.8 dBV  (0.0510640 Vrms)
    54.4 | ##############################    -38.7 dBV  (0.0115578 Vrms)
    56.0 | ########################    -46.8 dBV  (0.00455602 Vrms)
    57.5 | ########################    -52.8 dBV  (0.00229483 Vrms)
    59.1 | ####################    -57.5 dBV  (0.00133357 Vrms)
    60.6 | ##################    -61.4 dBV  (0.000851925 Vrms)
    62.2 | ################    -64.7 dBV  (0.000580929 Vrms)
    63.7 | ###############    -67.6 dBV  (0.000417279 Vrms)
    65.3 | ##############    -70.1 dBV  (0.000311295 Vrms)
    66.9 | ############    -72.4 dBV  (0.000240022 Vrms)
    68.4 | ###########    -74.5 dBV  (0.000188986 Vrms)
    70.0 | ###########    -76.3 dBV  (0.000153853 Vrms)
    71.5 | ##########    -78.0 dBV  (0.000126487 Vrms)
    73.1 | #########    -79.5 dBV  (0.000105473 Vrms)
    74.6 | ########    -81.1 dBV  (8.85317e-05 Vrms)
    76.2 | ########    -82.3 dBV  (7.68486e-05 Vrms)
    77.7 | #######    -83.5 dBV  (6.67320e-05 Vrms)
    79.3 | #######    -84.6 dBV  (5.87503e-05 Vrms)
    80.8 | #######    -85.7 dBV  (5.21204e-05 Vrms)
    82.4 | #######    -86.6 dBV  (4.67515e-05 Vrms)
    84.0 | ######    -87.6 dBV  (4.18728e-05 Vrms)
    85.5 | ######    -88.3 dBV  (3.85408e-05 Vrms)
    87.1 | #####    -89.0 dBV  (3.53286e-05 Vrms)
    88.6 | #####    -89.7 dBV  (3.25697e-05 Vrms)
    90.2 | #####    -90.0 dBV  (3.15736e-05 Vrms)
    91.7 | ####    -90.9 dBV  (2.84827e-05 Vrms)
    93.3 | #####    -90.7 dBV  (2.91151e-05 Vrms)
    94.8 | ####    -91.6 dBV  (2.61642e-05 Vrms)
    96.4 | ####    -91.9 dBV  (2.54884e-05 Vrms)
    97.9 | ####    -91.4 dBV  (2.67795e-05 Vrms)
```

Figure 11: Output of The Second FFT Shown Via Pico's ASCII Spectrum Functionality

**Discussion**

The results demonstrate that JSIM 2.0 successfully integrates time-domain simulation, digital filtering, and frequency-domain analysis within a single, portable platform. In the evaluation circuit, the simulator accurately captured the superposition of the two sinusoidal sources at 20 Hz and 50 Hz, as evidenced by the initial FFT output, which clearly identified both frequency components. This confirms that the selected sampling configuration was sufficient to resolve the relevant spectral content without unnecessary computational overhead.

Following the application of a 35 Hz high-pass filter, the subsequent FFT illustrated clear attenuation of the 20 Hz component while preserving the 50 Hz signal. The contrast between the pre- and post-filter spectra provides a straightforward visual demonstration of frequency-selective behavior, reinforcing fundamental filter concepts commonly introduced in early circuits and signals coursework. Importantly, these results were generated using the same workflow and analysis tools intended for both simulation and experimental measurement, highlighting the system's ability to support consistent comparative analysis.

From an educational perspective, the results indicate that JSIM 2.0 enables students to move fluidly between circuit construction, simulation, signal processing, and spectral interpretation within a unified environment. The ability to immediately observe how digital filtering affects both time- and frequency-domain representations supports conceptual understanding of signal behavior and filter operation. While the results presented here focus on simulation output, the same pipeline is designed to accommodate measured data, positioning JSIM 2.0 as a bridge between analytical theory, numerical simulation, and physical experimentation. The core features of JSIM 2.0 map explicitly to targeted learning objectives in a typical introductory circuits or signals course. The Falstad-to-netlist translation and MNA-based simulation engine support objectives related to circuit modeling and nodal analysis; the integrated FFT pipeline targets frequency-domain interpretation and filter characterization; and hardware-based ADC measurement reinforces the connection between idealized simulation and real-world signal acquisition. When discrepancies arise between simulated and measured results, students are guided to consider practical sources of error such as ADC quantization noise, component tolerances, non-ideal source impedance, and parasitic effects. Instructors can frame these discrepancies as productive learning opportunities: students are prompted to hypothesize which non-ideal factor is most likely responsible, adjust simulation parameters to approximate the observed behavior, and document their reasoning. This reflective process mirrors engineering design iteration and strengthens critical thinking about model validity and measurement uncertainty.

**Future Work**

With the expanded functionality introduced in JSIM 2.0, the next critical step is formal evaluation in classroom and laboratory settings. While the original version of JSIM demonstrated promise as an accessible simulation tool, its limited integration with hands-on experimentation constrained its pedagogical impact. The addition of hardware-based signal acquisition, digital filtering, and near real-time frequency-domain analysis represents a significant maturation of the platform, making it well-suited for structured instructional use.

Future work will therefore focus on conducting classroom trials to assess the effectiveness of JSIM 2.0 in improving student understanding, engagement, and confidence in introductory circuits and signals topics. Planned evaluations include deployment in undergraduate laboratory and recitation sections, comparison with traditional simulation-only or hardware-only lab

exercises, and collection of both quantitative performance data and qualitative student feedback. These studies will help determine how the integrated simulation-and-measurement workflow influences conceptual learning and supports transfer between analytical and experimental domains. Preliminary informal observations from students who interacted with JSIM 2.0 during development testing are encouraging: students noted that seeing both the simulated and measured FFT spectra side by side made it easier to understand what a filter is "actually doing" to a signal, and several expressed interest in extending the tool to circuits beyond the introductory lab sequence. While these observations are anecdotal and not yet systematic, they motivate the formal evaluation described above and suggest that the integrated workflow resonates with students' intuitions about circuit behavior.

Additional development efforts will explore support for a broader range of circuit elements, enhanced visualization tools, and tighter integration with external analysis environments. However, the immediate priority is to validate the educational impact of JSIM 2.0 through classroom use now that the platform has reached a level of functional maturity beyond its initial prototype.

**References**


[1] M. T. Taher and A. Khan, "Effectiveness of Simulation versus Hands-On Labs: A Case Study for Teaching an Electronics Course," in *Proc. ASEE Annual Conf.*, 2015.

[2] S. Nikolic, "Remote, Simulation or Traditional Engineering Teaching: A Systematic Review," *Eur. J. Eng. Educ.*, vol. 47, no. 2, pp. 1–18, 2022.

[3] C. Furse, S. Wood, and C. Roper, "A Comparison of the Conceptual Understanding of Electrical Engineering Students," *IEEE Trans. Educ.*, vol. 45, no. 3, pp. 270–276, Aug. 2002.

[4] P. Falstad, "Interactive Circuit Simulator," online. Available: https://www.falstad.com/circuit/. Accessed 2025.

[5] C.-W. Ho, A. E. Ruehli, and P. A. Brennan, "The Modified Nodal Approach to Network Analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, Jun. 1975.

[6] J. Vlach and K. Singhal, "Computer Methods for Circuit Analysis and Design," *IEEE Trans. Circuits Syst.*, vol. 30, no. 6, pp. 335–346, Jun. 1983.

[7] A. V. Oppenheim and R. W. Schafer, "Discrete-Time Signal Processing," *Proc. IEEE*, vol. 69, no. 4, pp. 529–541, Apr. 1981.