Designing a Sketch-based Interface for Electronic Circuit Simulation

Taharka Okai

April 2023

Final year project thesis submitted in support of the degree of Master of Engineering in Computer Science and Electronics

Department of Electrical & Electronic Engineering
University of Bristol

Contents

List of Acronyms

AR augmented reality. 6

CNN convolutional neural network. 5, 6

HDL hardware description language. 5

NN neural network. 5

SDK Software Development Kit. 4

SPICE Simulation Program with Integrated Circuit Emphasis. 4, 5

VR virtual reality. 6

Abstract

1 Introduction

2 Related Work and Key References

2.1 Data Acquisition, Input and Processing

The paper by Bonnici et al. [1] is a review paper that defines the landscape of sketch-based interfaces, with an emphasis on mechanical systems. They list the challenges faced with such approaches, and solutions taken by various authors. In particular, they provide a helpful categorisation of input acquisition and processing techniques. They define 'Paper Sketches' and 'Digital Sketches' which are input techniques that use digital images captured of a non-electronic sketch (i.e., with pen or pencil on paper) and otherwise direct digital input by the use of a peripheral device (i.e., with a digital pen or touch screen).

With regard to processing, Bonnici et al. describe the pipeline of image processing for each input approach, which broadly falls into the following order: Binarisation, Vectorisation and Interpretation. These refer to distinguishing foreground from background elements, smoothing line strokes and creating a physical system respectively.

In the paper by Costa et al. [2], they present SketchyDynamics, an application that takes in user input sketches via a touch screen, builds a system from the sketch and allows simulation of that system. They did not need to acquire any data to interpret user input as they use a gesture recogniser known as CALI to turn user sketches into shapes, interpreted as simulation primitives. They also did not need to have significant development time dedicated to processing the system as they used the Box2D physics engine, which is a rigid body simulation library for games.

Hu et al., [3] use a game engine, Unity3D, and the accompanying physics tools it provides to create a mechanical system simulation. As such, they do not provide a means of data acquisition and input, as the systems and the entities within are 3D modelled in a separate software, 3dmax. However, this reveals the issue of decoupling in mechanical physical simulations – the simulator and 3D modelling steps are separate and often require separate software.

Bergig et al. [4] present a software project that can 'analyse, visualise and simulate mechanical system in 3D'. They capture input from a webcam and display it to the user. This is an example of a 'Paper Sketch' implementation, as the sketch system is prepared on a non-digital medium. They make use of orthographic projection techniques to produce simple systems of the sketches. Rotations, forces, friction and other physical properties are inferred from annotations on the diagram, a feature unique to this paper.

Pichiliani et al. [5] recognise the ease of sketching in the design phase of physical concept and discuss the use of 'Digital Ink' applications, which is an example of the 'Digital Sketches' concept mentioned in the review in [1]. They also propose collaborative input, where multiple engineers can contribute to a schematic at the same time. While this is a useful feature, it is not within scope of the current project and could be considered for future work. Their input method uses the Microsoft Tablet Software Development Kit (SDK) which was shared under a licence agreement. With this, they are able to construct basic shapes for which to do 2D mechanical simulations.

Fang et al., [6] present Sketch3D, which also uses a digital interface in order to retrieve user sketches. They process the data by using a curve estimation algorithm which involves preprocessing the strokes on the screen, finding key points, recognising primitives (shapes), and reconstruction.

2.2 Electronic Systems

There are a variety of robust electronic simulation tools such as Micro-Cap, AWR Design Environment, Simulation Program with Integrated Circuit Emphasis (SPICE)-based tools such as LTSpice and more [7, 8, 9]. However, each of the input modes provided by these applications do not include sketch-based user input. Despite that, their powerful simulation capabilities make them useful to apply to this project, in terms of the algorithms required to simulate a given system and produce time-based output.

For a more targeted case study, SIMULINK uses a drag-and-drop type interface to build block diagrams of electronic systems [10]. This approach removes any need to code and is similar in the approaches taken by the aforementioned electronic system simulation tools. It also provides the ability to link the input and output of models together for use in more complex systems, which allows for a modular design approach.

Certain SPICE-based tools are able to import and export a circuit description file, or netlist [11]. It may be possible, then, to offload simulation tasks to pre-existing software. This turns the problem of sketch-based simulation into generating a valid circuit description that can be used by an external program. This technique is apparent in the web-based tool developed by Kadlec et al. [12], where they allow users to edit subcircuit parameters directly in the model description. Also in this paper, the authors describe the architecture of their software tool, where, instead of a sketch-based interface, there is a graphical user interface accessible through a web browser.

To avoid the introduction of any dependency on external software, this can instead be implemented directly, with the algorithms used by SPICE-based tools, which involves a combination of nodal analysis and numerical methods to solve the system of equations defined by the netlist [13, 14].

2.3 Concept-Adjacent Success

In markedly similar projects to this one proposed, such the one presented by Zamora et al. [15], there is a successful implementation of a sketch-based simulation tool for logic circuits. They make use of digital sketching as their interface and employ a recognition pipeline in the model generation process. They describe the segmentation of input strokes, then the interpretation of these strokes into primitives, which for this project are a set of logic gates. Once the primitives have been defined in the sketch using a classifier, then the circuit is evaluated which is an underlying node graph representation. This is one such example that makes use of the techniques described in the prior sections and represents an outcome similar to the aims of this project.

Similarly, with the project by Alvarado et al. [16], they also include error correction to primitives which attempt to improve the input sketch, by notifying the user that endpoints are not properly connected, for example.

The project presented by Dreijer [17], they produce an idealised version of the type of tool that this project is aiming to produce, using digital ink as their medium. Their implementation consists of a drawing process that accepts strokes until a symbol has been created before creating a new symbol, a clustering process that identifies symbols by finding certain types of line intersection types within a specific region, and a design for a format of a circuit description, similar to SPICE-based netlist files.

Finally, the project described by Majeed et al. [18] takes the ideas by Dreijer and compare a variety of machine learning and deep learning tactics, producing Sketic. Notably, they highlight the fact that deep learning approaches implicitly extract features from the user's sketches, decreasing the dependency on image preprocessing. When compared with a convolutional neural network (CNN) classifier over a regular neural network (NN) classifier, they tout improved accuracy and speed, at the cost of additional overhead that scales with the complexity of the input diagram. Additionally, their software tool has the ability to produce hardware description language (HDL) code (in the form of Verilog [19]) for the logic circuits defined in the sketch, which is a direct analogue to producing a netlist for an electronics circuit developed using a SPICE tool. This paper is the most recent one researched with high relevancy (2020), making use of modern techniques and advancements in the field that some older papers mentioned have not demonstrated.

2.4 Research Summary

Table 1 contains relevant papers discussed prior, and their estimated utility as a foundation for this project. This table uses a Relevance vs. Risk evaluation to rank each paper's contribution to the background knowledge behind this research topic. Relevance describes the amount of directly applicable content the paper has, and Risk is a factor describing complexity (high = low risk).

The preliminary research conducted in the above section has yielded the following useful information:

- Considerations required for different input media types,
- Separate challenges faced simulation electronic and mechanical systems,
- Challenges faced when processing sketches generating primitives,
- Processes of generating a model from a sketch using various classification algorithms,
- Methods of simulating a model using numerical methods,
- Potential to offload simulation tasks to an external program,
- A set of concept-adjacent and concept-aligned software tools as a potential basis.

Paper	Summary
Where do we stand? Bonnici et al. [1]	Review: processing sketches, input/model classification
SketchyDynamics, Costa et al. [2]	A novel sketch-based approach, mechanics simulator
Unity3D approach, Hu et al. [3]	Using Unity3D game engine for simulation, no sketching
Augmented reality (AR)/virtual reality (VR), Bergig et al. [4]	Mechanics simulator with AR/VR capabilities
Collaborative Design, Pichiliani et al. [5]	Collaborative sketch-based mechanics simulator
Sketch3D, Fang et al. [6]	Mechanics simulator, sketch processing techniques
Internet-based Sketch Tool, Hu et al. [12]	Electronic system simulator hosted online
CircuitBoard, Zamora et al. [15]	Sketch-based logic simulator
LogiSketch, Alvarado et al. [16]	Sketch-based logic simulator
Electronics model generator, Dreijer [17]	Sketch-based electronics model generator with detailed input processing techniques
Sketic, Majeed et al. [18]	Sketch-based electronics model simulator making use of CNNs and VHDL code generation

Table 1: Table of research summaries.

3 Method

3.1 Resource Requirements

The following table (Table 2) summarises the expected resource list for this project.

Resource	Type	Description	Availability
Workstation	Hardware	For computation and compilation	Own or Laboratory
Smartphone or Tablet (with camera)	Hardware	For development, testing and demonstration	Own or Laboratory
Machine/Deep Learning Tools	Software	Development library, e.g. PyTorch	Online
Image Processing Tools	Software	Development library, e.g. OpenCV	Online
Software Development Kit and framework	Software	Cross-platform development, e.g. Flutter	Online
Mobile Device emulator	Software	E.g. Android Studio for cross-platform development	Online
Labelled data set	Misc	For training the sketch-to-system model	Own or Office Supplies
Circuit Simulation Guide	Misc	Insight into the algorithms used in simulation	Bristol University Library (Acquired)

Table 2: Table of project resources

4 Results

5 Discussion

6 Conclusion

References

[1] A. Bonnici, A. Akman, G. Calleja, K. P. Camilleri, P. Fehling, A. Ferreira, F. Hermuth, J. H. Israel, T. Landwehr, J. Liu, and et al., “Sketch-based interaction and modeling: where do we stand?” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 33, no. 4, pp. 370–388, 2019.

[2] A. Costa and J. Pereira, “SketchyDynamics: A Library for the Development of Physics Simulation Applications with Sketch-Based Interfaces,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2, pp. 23–30, 09 2013.

[3] W. Hu, Z. Qu, and X. Zhang, “A new approach of mechanics simulation based on game engine,” in *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, 2012, pp. 619–622.

[4] O. Bergig, N. Hagbi, J. El-Sana, and M. Billinghurst, “In-place 3D sketching for authoring and augmenting mechanical systems,” in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, 2009, pp. 87–94.

[5] M. C. Pichiliani and C. M. Hirata, “A Digital Ink Sketch-Based Application for Collaborative Design and Kinematic Simulation,” in *2009 Simposio Brasileiro de Sistemas Colaborativos*, 2009, pp. 1–9.

[6] G. Fang and J. Wang, “An Interactive Sketching Modeling System: Sketch3D,” in *2008 International Conference on Computer Science and Software Engineering*, vol. 2, 2008, pp. 1049–1052.

[7] “ Software - Micro-Cap 12. Analog Simulation, Mixed Mode Simulation, and Digital Simulation Software. SPICE and Pspice® compatible circuit simulator.” Web Article. [Online]. Available: http://www.spectrum-soft.com/index.shtm

[8] C. D. S. Inc, “Awr design environment platform,” 2022. [Online]. Available: https://www.cadence.com/ko_KR/home/tools/system-analysis/rf-microwave-design/awr-design-environment-platform.html

[9] A. D. Inc, “ LTspice simulator — Analog Devices ,” Web Article, 2022. [Online]. Available: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html

[10] Mathworks, “Simulink - Simulation and Model Based Design - MATLAB & Simulink,” Web Article. [Online]. Available: https://uk.mathworks.com/products/simulink.html

[11] C. D. S. Inc, “PSpice Reference Guide,” Web Article, 2000. [Online]. Available: https://www.seas.upenn.edu/~jan/spice/PSpice_ReferenceguideOrCAD.pdf

[12] J. Kadlec, L. Popelka, R. Kuchta, and R. Vrba, “Internet Tool for Electronic Circuit Analysis,” in *2010 Fourth International Conference on Digital Society*, 2010, pp. 98–101.

[13] eCircuit Center, “Spice Algorithm Overview,” Web Article, 2003. [Online]. Available: http://www.ecircuitcenter.com/SpiceTopics/Overview/Overview.htm

[14] F. N. Najm, *Circuit Simulation*. A John Wiley & Sons, Inc, 2010.

[15] E. A. E. Shane W Zamora, “CircuitBoard: Sketch-Based Circuit Design and Analysis,” in *IUI Workshop on Sketch Recognition*, 2009.

[16] C. Alvarado, A. Kearney, A. Keizur, C. Loncaric, M. Parker, J. Peck, K. Sobel, and F. Tay, *LogiSketch: A Free-Sketch Digital Circuit Design and Simulation System.* IEE Proceedings - Circuits Devices and Systems, 07 2015, pp. 83–90.

[17] J. F. Dreijer, "Interactive recognition of hand-drawn circuit diagrams," Master's thesis, University of Stellenbosch, Department of Electronic Engineering, 12 2006.

[18] M. A. Majeed, T. Almousa, M. Alsalman, and A. Yosef, "Sketic: a machine learning-based digital circuit recognition platform," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 4, 2020.

[19] IEEE, "IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language," *IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012)*, pp. 1–1315, 2018.