Translating an electrical circuit schematic into a SIMULINK model

Andreas Boltres

Karlsruhe Institute of Technology

Kaiserstr. 12, 76131 Karlsruhe, Germany

`andreas.boltres@student.kit.edu`

Cem Gülşan

Hamburg University of Technology

Am Schwarzenberg-Campus 1, 21073 Hamburg, Germany

`cem.guelsan@tuhh.de`

Christian Alexander Vecsei

RWTH Aachen

Templergraben 55, 52062 Aachen, Germany

`christian.vecsei@rwth-aachen.de`

Thomas Frei

University of Applied Sciences and Arts Northwestern Switzerland

Bahnhofstrasse 6, 5210 Windisch, Switzerland

`thomas.frei1@students.fhnw.ch`

Abstract

We present a functionality developed within the computing environment MATLAB, which analyses schematics of electrical circuits and transforms them into an electrical circuit model of MATLAB's SIMULINK environment.

1. Introduction

1.1. Motivation

Students and engineers often use schematics while studying or developing. A schematic is a drawing of an electric circuit which should contain all information about the circuit but should also be as simple as possible.

With increasing size, electrical circuits often become more complicated and confuse less experienced engineers, rendering the estimation of the circuit's behaviour hard or impossible.

A program that confidently identifies the components of a schematic with their respective connections and translates them into a user-friendly digital model can substantially accelerate the understanding of the given circuit. Analyzing, modifying and storing a schematic becomes easy once it is properly digitized.

Figure 1. Example circuits

1.2. Program overview

The aim of this student project is, as stated in the abstract, to create a functionality developed within the computing environment MATLAB, analyzing schematics of electrical circuits and transforming them into an electrical circuit model within MATLAB's Simulink environment. There, it can be used to facilitate calculations or behaviour

analyses. The process of obtaining the Simulink model from the given image can be divided into three phases which will be briefly described in the following sentences and in detail in the following sections.

Figure 2. Top: example screenshot of a circuit. Bottom: example photography of a circuit.

Initially, the program expects a digital image to be selected by the user. This image can be either a photography of a schematic on a sheet of paper or the image of a digitally created schematic. If the image is a photography, the program will first detect the boundaries of the sheet of paper in order to then rectify the image such that the sheet of paper lies entirely in the image plane. This implies that the entire paper is clearly included in the photography. After rectification, the photo can be trated in the same way digitally created schematic pictures are to be treated.

Figure 3. ROI selection within a rectified photography.

The user selects a region of interest (ROI) within the provided (rectified) image; the following analysis will be limited to that specific ROI. The first step of the image analysis algorithm is **Optical Character Recognition** (OCR),

as every electrical element comes with electrical properties textually encoded next to them in the image. Recognized text is stored and deleted from the image. Hereby OCR is running in a loop: After each OCR iteration, the program asks the user to confirm the removal of all text within the selected ROI. If this is not the case, he is asked to manually select new regions of interest so that the OCR routine can obtain and remove the remaining text.

Following up is the **retrieval of geometric primitives** within the images which will be further explained in subsection 3.1. Obtained through the binarized and thinned image, these primitives are used to retrieve the scale of the electrical components of the schematic in the image.

Using the knowledge of the components' scale, the **object detection** is done using a sliding window to retrieve the location and orientation of electrical elements in the image. Following up is the **detection of wires** based on the image after deletion of the detected elements. See subsections 3.2 and 3.3 for more detailed descriptions.

Finally, after having detected elements and the connections in between, the **output generation** model uses the objectified results of the image analysis and creates and fills a Simulink model. This model will then be shown and saved for further usage. Refer to section 4 for further imformation.

1.3. LTSpice

TODO 1 TODO auch auf normierung eingehen

Figure 4. Examples of LTSpice-conform circuit elements

TODO 2

2. Preprocessing

Before being able to obtain the electrical circuit elements from the provided image, several steps have to be taken in order to facilitate the information retrieval.

2.1. Rectification

When dealing with photographies, the depicted schematic may be distorted due to the surface not being aligned to the image plane. This has to be taken care of in order to use the object recognition approach presented in this work. Therefore, asserting that the entire surface and its corner points are included in the image, a rectification step is taken to bring the schematic onto the image plane.

TODO explain rectification

Figure 5. Pre/Post-rectification img

2.2. Optical Character Recognition

In depictions of non-distorted or rectified schematics, MATLAB's OCR functionality can be used to retrieve text and its position within the image. After retrieval, the corresponding text regions in the image are erased, simplifying the following recognition of electrical elements and connections.

Figure 6. Some ocr img

TODO OCR explanation

2.3. Binarization, Morphology

After all text has been recognized and erased from the image, a binarized and thinned-out version of the image is prepared to further facilitate the following retrieval of lines, circles, corners and element template matchings. Both operations are done using pre-defined functions within MATLAB.

Figure 7. Left: image region before binarization and line thinning. Right: the same image region after binarization and line thinning.

3. Image Analysis

The goal of the functions within the image analysis package is to retrieve all objects from the pre-processed image. This includes the electrical components themselves as well as the wires connecting them. The approach used for the detection of the electrical components is a sliding-window approach, implying that a robust estimation of the scale electrical components is essential in order to find the objects quickly. This in turn calls for the usage of geometric primitives in order to determine that scale.

3.1. Geometric primitive retrieval

For lines, a parametrized hough transformation is used. The results are rearranged into a data structure including length and angle of all lines, using the given start- and endpoints of the lines.

For circles and corners, which are unused in the current version of the program, predefined MATLAB functionality is used to detect the harris features of the image and the centers and radii of circles are found using a circular hough transform.

In order to retrieve the scale of the elements, skewed lines are used because in rectified schematics conforming to LTSpice, they are used in resistors only. They can be used to retrieve their dimensions, when assuming that every schematic has at least one resistor and every resistor is conforming to the LTSpice regulations. Then, the greater one

Figure 8. Top: pre-processed image region. Bottom: retrieved geometric primitives: circles in blue, lines parallel to the axes in red, skewed lines in green and corners as green crosses.

of the coordinate differences of full-length lines (of which one is highlighted in Image 9) along the axes corresponds to the shorter side of the resistor whereas four times the coordinate difference along the other axis corresponds to the longer side of the resistor.

Figure 9. Resistor template with highlighted full-length line and resistor dimension explanations.

Using this heuristic, the size of a resistor can be estimated within tolerance of a few pixels. Furthermore, including the stretched half-length lines which lie at the end of resistors, the algorithm becomes more robust.

3.2. Object Detection

TODO Cem

3.3. Wire Detection

4. Output Generation

4.1. SIMSCAPE Electrical Model

4.2. Component Generation and Assembly

5. Evaluation

Contributions

Name	Intro	Research
Andreas Boltres	25%	25%
Cem Gülşan	25%	25%
Christian Alexander Vecsei	25%	25%
Thomas Frei	25%	25%
	100%	100%

Table 1. List of members and their contribution