

# PhyNOMOnal - Generalized Physical Nomograms

Ernesto Ramirez, Shenli Yuan

Mechanical Engineering, Stanford University

Stanford, CA 94306, USA

{ramireze, shenliy}@stanford.edu

## ABSTRACT

In this work, we created a modular tangible interactive nomogram system that allows user to get results from their customized equations and explore the process of finding a solution. The graphical user interface allows users to select from different types of basic operations (addition, subtraction, multiplication and division), link multiple operations at the same time, and set the range of each parameter in the nomogram; the tangible user interface allows user to change parameters, customize different motions for each nomogram parameter (linear or nonlinear), dynamically manipulate the value of each parameter, which is also synchronized with graphical user interface. This system introduces generality into nomograms that are only representations of certain equations and also exceeds the meaning of traditional definition of nomograms by allowing users faster interaction and exploration.

## AUTHOR KEYWORDS

Nomogram; Nomograph; Tangible user interface; Graphical user interface; Physical interaction; Data visualization

## INTRODUCTION

Nomograms (or Nomographs) are traditionally graphical calculating devices that allow the approximate graphical computation of mathematical functions. Conventional nomograms are known as graphical solutions for certain equations. They are widely used in engineering, medicine, astronomy and other areas to solve problems where theoretical equations are not available and only empirical solutions are available. In these cases, empirical

equations can sometimes be very difficult to solve numerically; and graphical solutions can not only give a good approximation of numerical solution, but also provide an insight on how certain solution compares to others. A few examples in the applications of nomograms include the Moody Chart [2], which is used in fluid dynamics, Smith Chart [7] used in electrical engineering to calculating radio frequencies, and Prostate Cancer Nomograms [8] used to help patients understand their diagnosis.

Most of the conventional nomograms are static diagrams. They generally require user to draw several lines across different axes on the diagram determined by the inputs. It can sometimes be very tedious to use. The biggest problem with most nomograms is, as mentioned above, that they are mostly used only for specific equations.

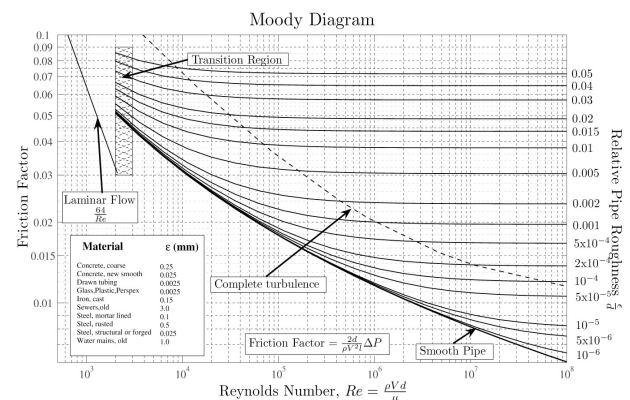


Figure 1 Moody Chart

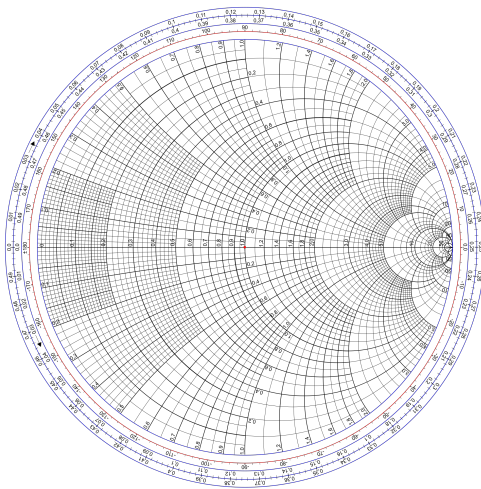


Figure 2 Smith Chart

## RELATED WORK

Howison [5] developed a Java based application for rendering user specified nomograms, an example is show in Figure 3. It allowed users to draw multiple isopleths (the lines connecting axes) and manipulate the isopleths by dragging the values on a certain axis. One weakness to the applet is that user can only manipulate one isopleth at a time. The user is also removed from drawing the line between two axes. We feel that physically drawing the line might be helpful to exploring and would like to retain that feature. The application also requires users to know Java to create the nomograms which removes from the accessibility.

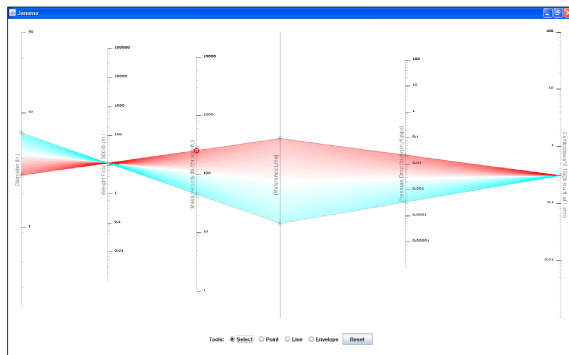


Figure 3 M. Howison, *Constructing Interactive Nomograms*

Jones, Camara, and Marrero [6] developed a Java based application that also allowed users to display a limited number of nomograms and draw multiple isopleths. What's missing from the application is the ability to manipulate isopleths,

the physical interaction, manipulation of multiple points, and the limited number of nomogram types supported.

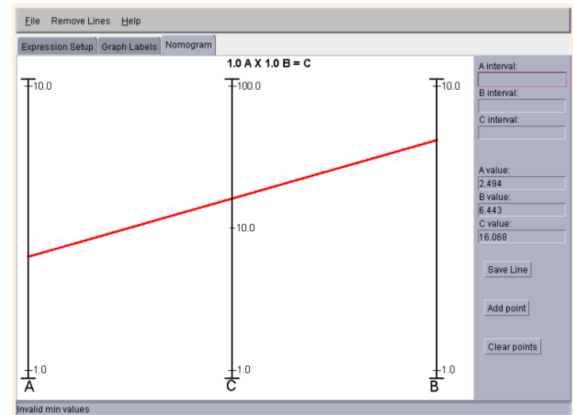


Figure 4 M. Howison, *Constructing Interactive Nomograms*

Tuna [10] developed a system that renders interactive nomograms on the web making interactive nomograms accessible to a wider audience. The system uses an modified PyNOMO [3] to render the application. The system uses also uses a modified PyNOMO syntax so users don't have to be extensive programming experience to use the platform. This system however does not have a means for physical input.

There have not been a lot of related work in this particular area, however all the previous works we found are limited to graphical interaction and visualization. Our work includes graphical user interface as well as tangible user interface. Our system also brings the generality to a higher level by allowing user to add multiple operation modules.

## METHODS

### System

PhyNOMOnal was designed to be an open source platform and so accessibility drove our design choices. PhyNOMOnal at its core is comprised of potentiometers, an Arduino, and a computer with a Chrome web browser.

### Hardware

The tangible user interface consists of a microcontroller (Arduino Uno [4]), linear and rotary potentiometers, customized cases for

potentiometers, customized faceplate with linear or nonlinear slots to constrain the motion, and customized knob-bar to ensure three knobs will be on the same line. The cases, knobs and faceplate are modeled in SolidWorks [9] and were then fabricated by laser cutting, which can be easily customized by the user according to their specific needs. The materials used to build the components in this system were aluminum and acrylic.

The functionality of the tangible interface comes from linear and rotary potentiometers. The linear scale module (Figure 7) provides a linear sliding motion for the user and output analog values to the microcontroller. The nonlinear scale module (Figure 8), which is a rotary potentiometer in combination with linear potentiometer, creates generalized nonlinear motion. By reading the analog outputs from both rotary and linear potentiometers, we are able to measure both the angle and distance of the knob and thereby determine the value for this parameter.

The schematic of electronic components is shown in Figure 5. The analog signals of potentiometers are transmitted through a multiplexer and then to Arduino. In the demo we used three analog pins for simplicity. However, there are only 6 analog I/O pins on Arduino Uno, meaning it can only read the signals from 6 linear/rotary potentiometers. By adding the additional multiplexer, the system will be able to read up to 21 (16 from multiplexer + 5 from Arduino Uno) potentiometer signals. Of course, the system can be further upgraded by adding more multiplexers, or using multiplexers with larger number of input channels.

PhyNOMOnal is designed to be modular, which is consistent to our goal of generality. Each potentiometer module can be easily taking off the face plate (Figure 9), and linear and rotary potentiometers can be easily assembled as one nonlinear module (Figure 10).

The knob-bar ensures the physical position of the knobs to be consistent with the values shown in the graphical user interface. A small tube is embedded in each knob serving as a bearing to ensure a smooth rotation around the knob of potentiometer, which allows user to

move the knobs freely without getting stuck due to the angular position of the bar.

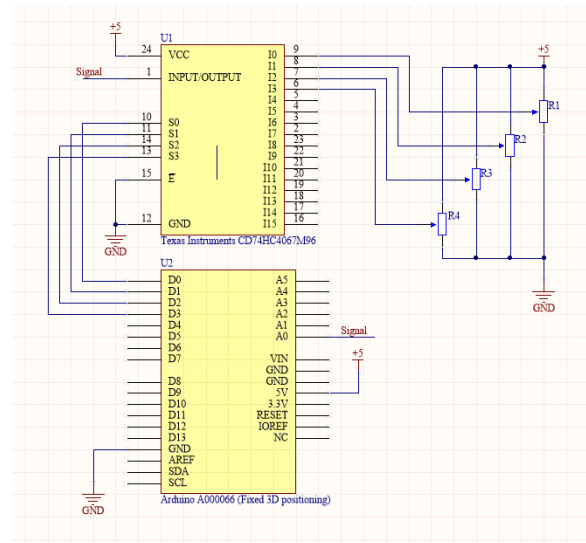


Figure 5 Electronics schematic (no distinguish between linear and rotary potentiometers)

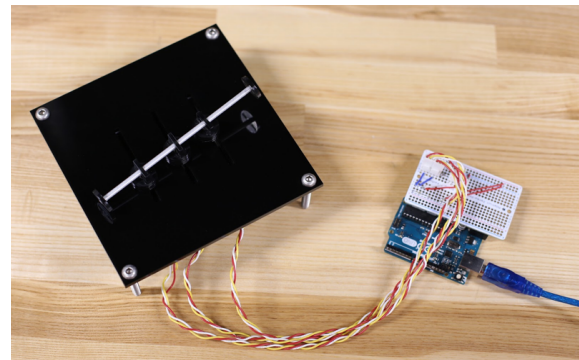


Figure 6 Analog values output from linear potentiometers are transmitted to an Arduino Uno microcontroller

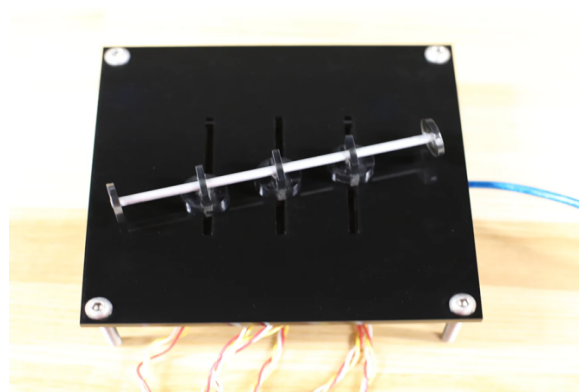


Figure 7 Tangible User Interface (three linear scales, corresponding to single module in Graphical User Interface)

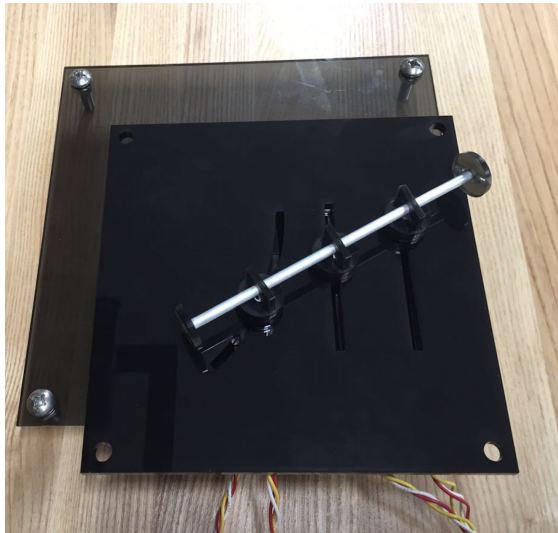


Figure 8 Tangible User Interface (two linear scales and one nonlinear scale)

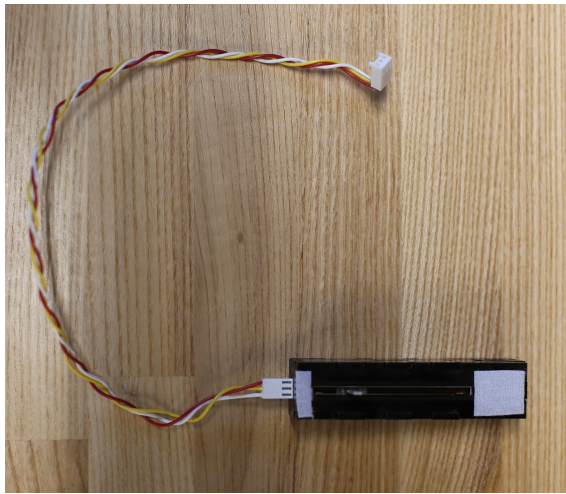


Figure 9 Single linear scale module

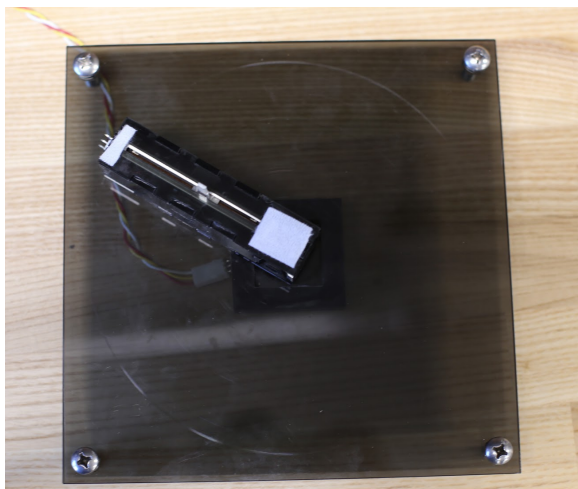


Figure 10 Single nonlinear scale module

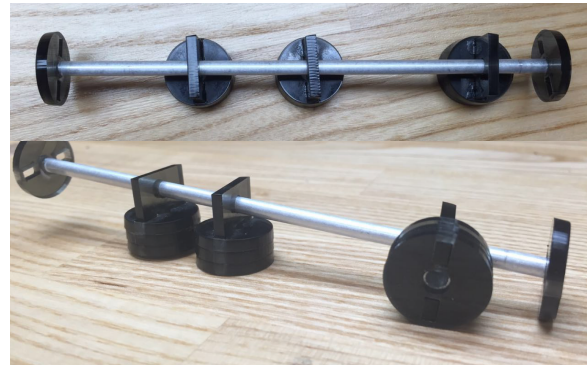


Figure 11 Knob-bar

### Software

The Graphical User Interface was developed in HTML, JavaScript and CSS for accessibility. The Involt framework [11] was used to connect the Arduino and Chrome browser. The Arduino was uploaded with a program that prints the analog values of each pin and sends them through a serial port. Those values were then read from the serial port using JavaScript and processed with d3 [1] to create the interface.

The graphical user interface so far only contains 3-linear-scale modules. Three parameters can be represented by three linear bars. Depending on whether the operation the user chooses is addition/subtraction or multiplication/division, the scales become either linear or logarithmic respectively.

The system relies on event listener and an imposed id convention on the HTML elements to be able to reuse function for multiple elements. Multiple arrays are also used to help in organizing the information required for displaying the components on the screen.

There is an array that contains for all the nomograms being displayed: the type of nomogram, the minimum value for the A input, the maximum value for the A input, the minimum value for the B input, and the maximum value for the B input. The range of the output is determined by performing the desired operation on minimum and maximum values of A and B. The axis types are determined by reading the operation. There is also an array designating for every axis, what circles have had their position locked, an array containing the position of every

circle on the screen, and variable pointing to the current axis selected.

At 20 Hz, circles of the selected axis are updated if the locked flag has not been set and a linear regression is performed on the current circles and that line is shown between the axes of the currently selected nomogram. A regression was chosen to remove the effects of not mechanically constraining the points to be in a line.

When a nomogram was added, the space allocated to each nomogram would be evenly divided and new elements would be appended to the arrays used to create the visualizations. Nomograms could be reordered by clicking on the Nomogram to be removed and dragging it to its new location. This caused the arrays to be resorted and the visualization to be redrawn. Double clicking on a circle would toggle the locked flag for that specific circle.

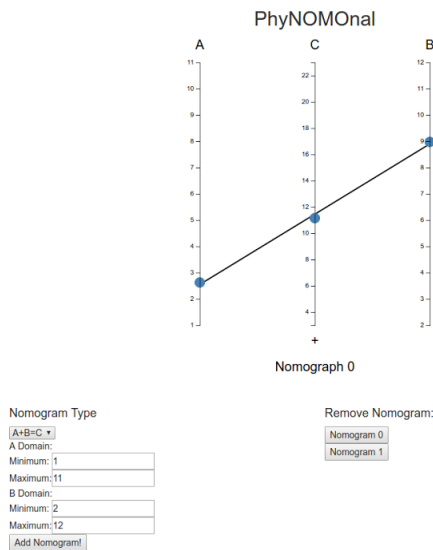


Figure 12 Graphical User Interface with single module

## RESULTS

### Interaction

PhyNOMOnal's equation interaction uses the three parallel axes nomogram. This nomogram type can perform three basic operations: addition, subtraction, multiplication, and division by manipulating axe types, logarithmic or linear, and by reordering the axes. to Multiple nomograms can be linked together to build

complex equations. A user can edit the type of nomogram by selecting the desired nomogram type from a menu and then specify the domain of the inputs to further customize their exploration. To perform an operation, the user physically draws the line across the nomogram by sliding around the linked potentiometers on the Tangible User Interface. The user can draw perform multiple operations by sequentially selecting the nomogram they would like to edit and then using the tangible user interface. PhyNOMOnal also supports connecting more than three potentiometers at once so multiple operations could be performed by nomographic specific modules. After a user performs an operation, the user can then lock that point in by double clicking the circle allowing them to save an operation's result.

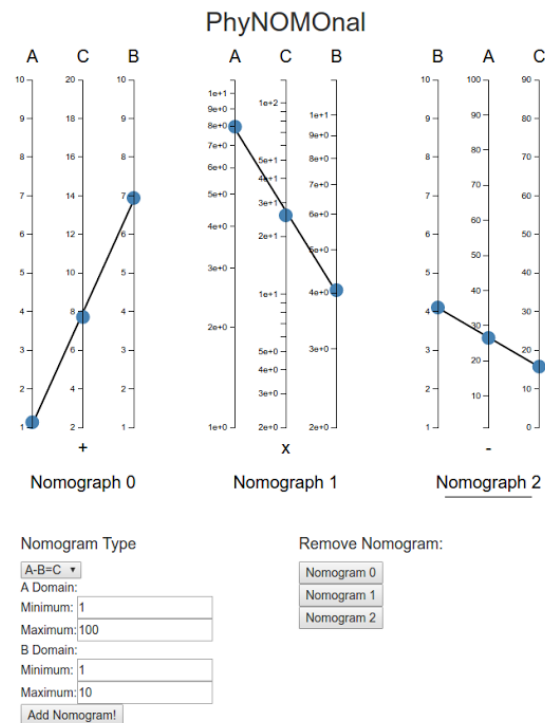


Figure 13 Graphical User Interface with multiple modules

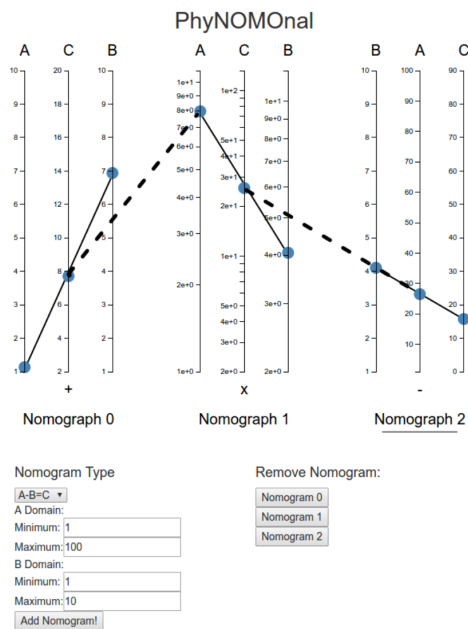


Figure 14 Graphical User Interface (link parameters across different modules, addressed in Future Work)

## DISCUSSION

PhyNOMOnal is an interactive nomogram system consisting of both graphical user interface and tangible user interface. The system surpasses the conventional static nomograms. It allows user to interactively manipulate the input parameters as well as the desired operations between parameters in order to not only obtain the results according to their customized equations, but also explore different possibilities in how to combine different parameters in order to get an expected result. The tangible user interface allows user to change values and receive feedback in a more effective fashion. The combination of the two interfaces gives user more control of their data as well as the entire interaction and exploration process.

## FUTURE WORK

There are a few improvements that we will possibly address in the future. First of all, in the graphical interface, we hope to link parameters across different modules (Figure 14). Currently this process is done by manually calibrating parameters between different modules to the same value. In addition, nonlinear system can be introduced into graphical interface to increase the generality of the system, while currently only

the nonlinear system is limited to our tangible user interface (Figure 8). On the tangible user interface, modularity has already offers us much freedom in customizing different equations; but similarly as graphical interface, it might be helpful to also physically link different nodes across different modules. Finally, currently the ability of fixing a node is only in graphical interface, which will be soon extended to tangible user interface as well.

## ACKNOWLEDGMENT

We sincerely thank Dr. Maneesh Agrawala for offering advises along different phases of this project, and Dr. Sean Follmer for providing hardware components and fabrication tools for this project.

## REFERENCE

- [1] Bostock, Michael. "D3. js." *Data Driven Documents* (2012).
- [2] Chen, NH. "An Explicit Equation for Friction Factor in Pipe - Industrial ..." 1979.
- [3] d'Ocagne, M. "*Traité de nomographie: Théorie des abaaques-applications pratiques*. Gauthiers." 1899.
- [4] D'Ausilio, Alessandro. "Arduino: A low-cost multipurpose lab equipment. "Behavior research methods 44.2 (2012): 305-313.
- [5] Howison, M. "Constructing Interactive Nomograms." 2014.
- [6] Jones, T. B. "An Online tool to create custom, interactive Nomograms (also called ..." 2009. 5 Jun. 2016
- [7] Richards, WF. "An improved theory for microstrip antennas and applications." 1981.
- [8] Ross, PL. "A catalog of prostate cancer nomograms. - NCBI." 2001.
- [9] SolidWorks, Introducing. "Solidworks corporation." *Concord, MA* (2002).
- [10] Tuna, C. "Newmograms: Authoring Interactive Nomograms for the Web." 2014.
- [11] Warzocha, E. <http://involt.github.io/>.