

Signal Constellation Design Tool: A Case study in User Interface Synthesis

Gabriel Robins

Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90024, USA
gabriel@vaxb.isi.edu

Abstract

Signal constellation design is a major subtask of constructing an efficient communication system; it essentially entails trading-off error frequency against information throughput, a chief occupation of modem designers. We propose and implement an interactive tool for designing and simulating arbitrary signal constellations. To construct the user interface we have utilized Interface Builder, a new interactive tool that greatly facilitates the synthesis of arbitrary user interfaces through an object-oriented methodology. Using the Interface Builder package and the Signal Constellation Design Tool as the target prototype, we show how an order-of-magnitude improvement can be achieved in the effort required to produce a complex user interface. Our secondary goal is to try to dispel some of the mystique surrounding user interface synthesis on state-of-the-art workstations by describing in detail the construction of an interactive tool for computer-assisted learning.

Keywords: User interfaces, User interface tools, Human computer instruction, Man-machine interaction, Computer-assisted learning, Simulation tools, Object-oriented systems.

1. Introduction

Signal constellation design is a major part of constructing an efficient communication system. This task essentially entails trading off error frequency against information throughput, a chief occupation of modem designers. We propose and implement an interactive tool for designing and simulating arbitrary signal constellations. While the actual code that simulates signal constellations is rather trivial in itself, the user interface to this code is quite complex. To design and construct this user interface we have used Interface Builder, a new interactive tool that greatly facilitates the synthesis of user interfaces through an object-oriented methodology. Using the Interface Builder package and the Signal Constellation Design Tool as the target prototype, we show how an order-of-magnitude improvement can be achieved in the effort required to produce a complex user interface, and then draw some conclusions regarding the synthesis of user interfaces in general.

Our secondary goal is to try to dispel some of the mystique surrounding user interface synthesis on state-of-the-art workstations. Many otherwise informed researchers have very little experience in user-interface design, and consequently view user interface design as some sort of a black art, best left to specialized hackers to dabble in. By user interface design I mean a collection of functionality (running on a bit-mapped display workstation with a mouse) that interacts with the user in a friendly manner via menus, scroll bars, control buttons, icons, mouse clicks, and key strokes.

We intend to show that, quite to the contrary of these myths, given the proper tools and methodology, the synthesis of complex user interfaces could be rather trivial. As a case in point, the user interface described in this document was implemented on a Macintosh, requiring only several days of coding, including the time to read the manuals and learn how to use the software. As a by-product of our inquiry, we have synthesized an interactive tool for computer-assisted learning.

The first half of this document explains signal constellation design in general and how Interface Builder was used to synthesis the user interface; numerous examples and illustrations are given. The rest of this document describes and illustrates the functionality and usage of the resulting signal constellation design tool. The annotated Common LISP source code is available upon request both in hardcopy and on a Macintosh diskette.

2. Signal Constellation Design

In designing an efficient communication scheme for band-limited channels, invariably of chief concern are the effects of noise and other kind of interference on the system [Forney, Gallager, Lang, Longstaff, and Qureshi]. To combat such interference, and while still aiming to achieve high throughput, one must carefully design an appropriate signal constellation [Carlyle] [Schwartz] [Sklar].

The task of signal constellation design essentially entails trading off error frequency against information throughput and is a chief occupation of modem designers. We propose and implement an interactive tool to alleviate the task of designing and simulating arbitrary signal constellations. We would like our tool to graphically display the signal constellation in two dimensions, allowing the user to visually observe the progressing simulation under interactive modifications to the interference parameters of the system.

2.1. The Desired Functionality

In this section we describe in more detail the functionality that we would like our Signal Constellation Design Tool to exhibit. Later we explain how this functionality was actually achieved in the implementation.

First, we would like to allow the user to select any of a number of "canned" standard signal constellations. For example, the user may elect to simulate an N-in-a-circle signal constellation and observe its performance under various levels of noise and distortion. Such selections should be done via mouse and menu interaction. Next the user may wish to select a certain probability distribution that would control the generation of random signal points. For example, the user may wish to select a Gaussian distribution with a specified variance.

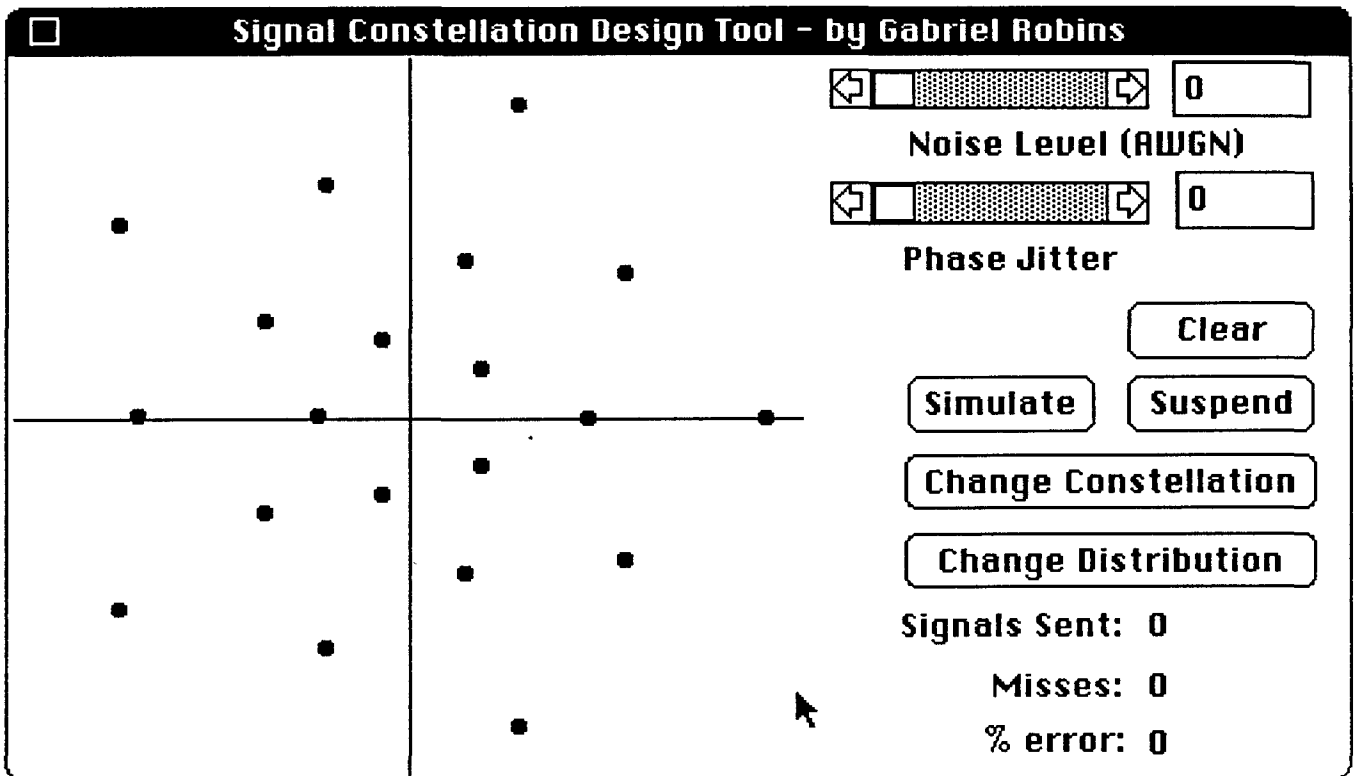
Once the user has selected a particular constellation to simulate/observe, as well as a probability distribution, that constellation should be drawn on the screen and the simulation may proceed. During the simulation, the user may interactively modify a number of system parameters, such as the phase jitter and the additive white Gaussian noise level. This would be accomplished by dragging "scroll-bars" identified with the corresponding parameters, or by directly typing in the desired values.

Using a random number generator, random signals are generated, according to the probability distribution function specified earlier, and are plotted on the signal constellation diagram. After a few minutes, a cumulative scatter-plot of the received signals will become apparent, giving the user an indication of how that signal constellation is performing under the distortion parameter values set previously. A cumulative running total of the number of errors encountered so far should be displayed, as is the empirically derived error-probability (the number of errors divided by the number of signals transmitted.)

The various commands should be also be accessible via clicking appropriate buttons, and alternatively also via menus and keystrokes. In addition, we wish to provide the user with some on-line help and information.

2.2. The Main Panel

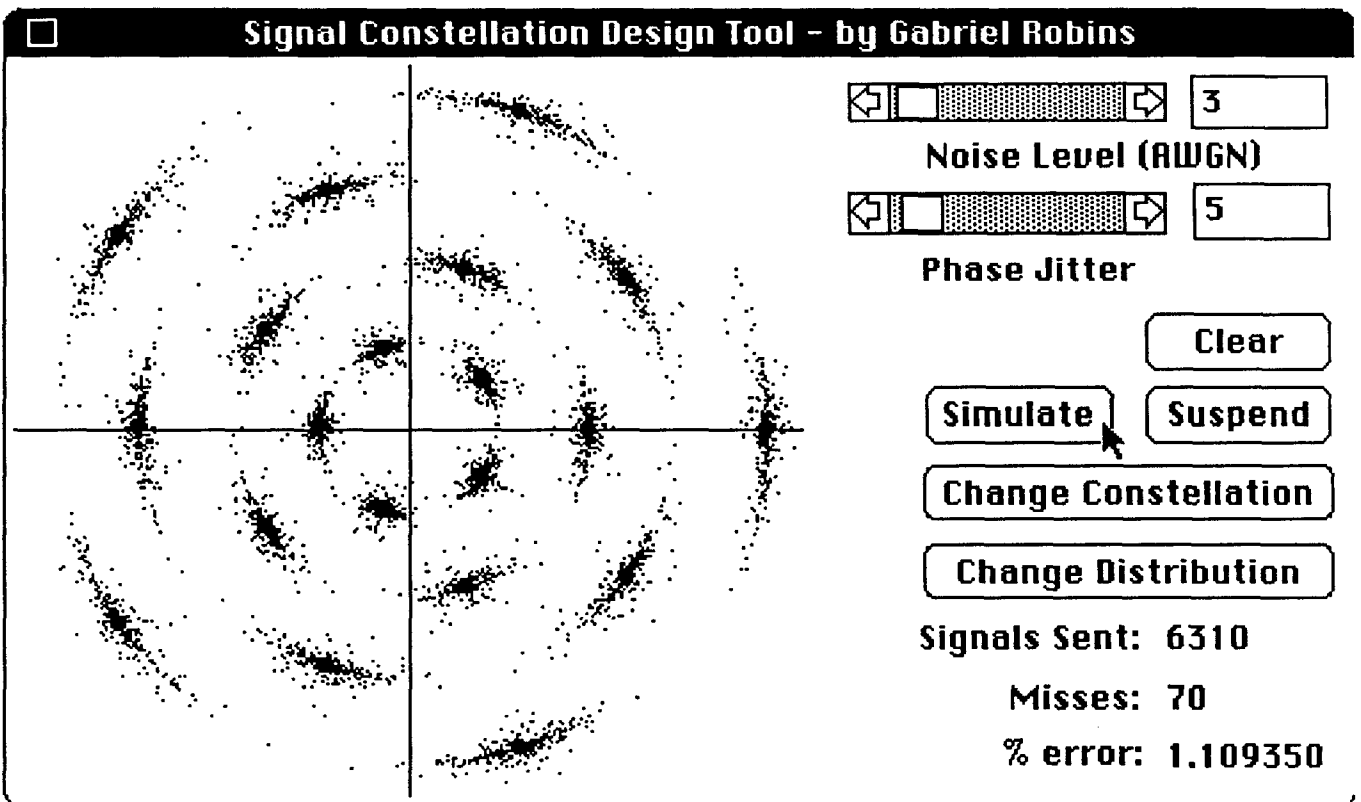
To make the appearance of the user interface more concrete, we give an illustration of how the main panel might appear:



To the left we see the main drawing area where the signal constellation appears; in this case the signal constellation itself consists of 20 points uniformly distributed on 4 concentric circles. At the top right we note the interference parameters, as well as the scroll bars and click boxes used to modify them. Below that we observe several "buttons" each of which will invoke a command if the user clicks it with the mouse. To the lower right we have the running statistics and error-ratio as the simulation progresses.

The user may invoke several operations simply by clicking the corresponding buttons. In addition, all of these commands are also available from the pull-down menus, as well as through keystrokes (i.e. single character keyboard inputs). We may also have at the top a pull-down menu bar, representing the various commands the user may invoke; the menu bar is not visible in this diagram.

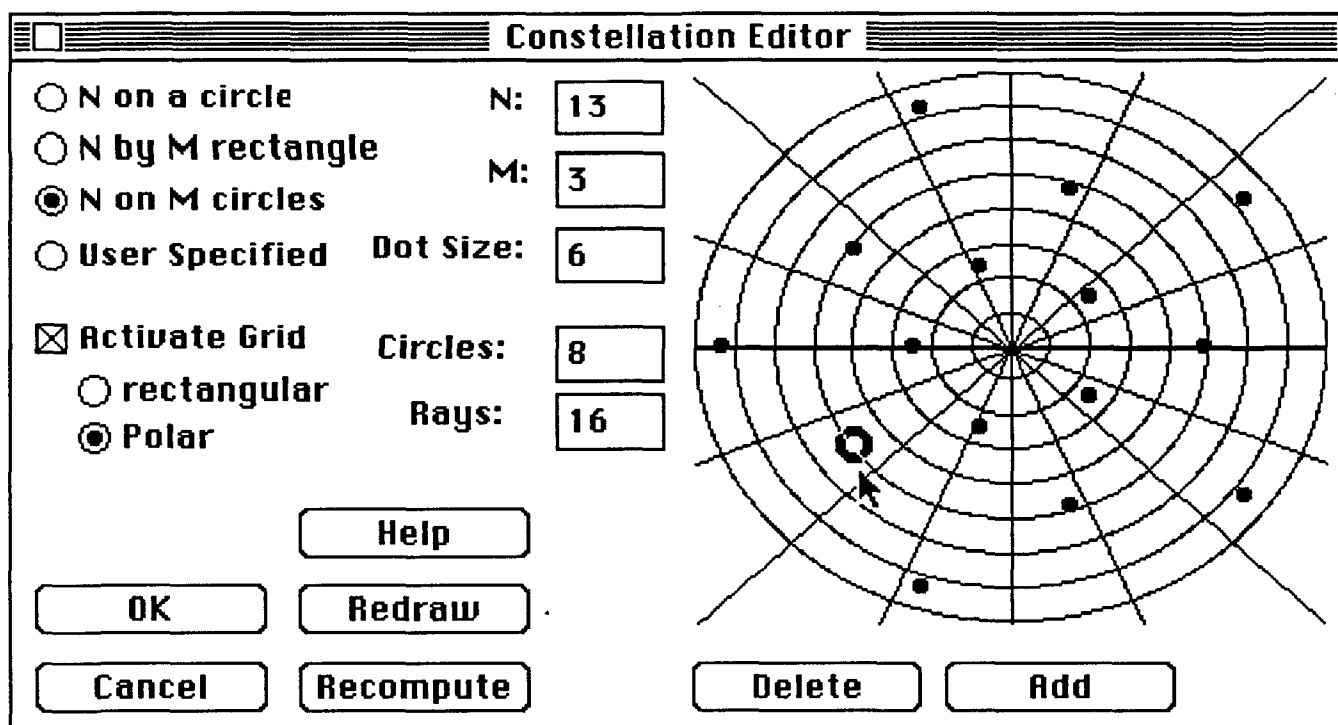
After a simulation has been underway for some time, the main panel might appear as follows:



The clouds around the signal points represent where the randomly generated signals fell around the actual signal constellation points. In this simulation, given the specified noise parameters, we are observing an error rate over over one percent, an undesirable situation.

2.3. The Constellation Editor

The panel that allows the user to select and edit a signal constellation is called the Constellation Editor and may appear as follows:



At the top left the user may select one of several "canned" signal constellations, parametrized by the variables M and N; these parameters are also user-specified: to change them, the user simply clicks in the corresponding box and types in the new value. An optional editing grid is available, and may be either rectangular or polar; the purpose of this grid is to make placement of individual constellation points more precise. The resolution of the grid may be controlled by the user; in the case of the rectangular grid, the number horizontal and vertical lines may be specified, and in the case of the polar grid, the number of circles and rays may be specified.

The user may add or delete constellation points, redraw the display, or obtain on-line help, simply by clicking the corresponding buttons. In addition, all of these commands are also available from the pull-down menus, as well as through keystrokes (i.e. single character keyboard inputs). Note that one of the points of the constellation is highlighted; this is accomplished when the mouse is clicked anywhere in the drawing area, whereupon the closest point to the click becomes highlighted. A "delete" command would subsequently remove the highlighted point, while an "add" command would wait for a new mouse click and a new point would be added to the constellation at the location of that click.

The on-line help consists of several screens of information and will be discussed later. "Ok" saves the current signal constellation and uses it from now on in all future calculations, while "Cancel" reverts back to the signal constellation previously in effect. Had the user selected a rectangular grid instead on a polar grid, the display might have appeared as follows:

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.