ON THE AESTHETICS OF COMPUTER AIDED TOOLS FOR SIGNAL PROCESSING

Murat Tanyel, Charles Adams
Dordt College

ABSTRACT

"Beauty is in the eye of the beholder" the old saying goes. The wisdom of ages seems to suggest that perception of beauty, a quality that invokes aesthetic pleasure, is arbitrary and this old adage is invoked to express the futility of efforts to reconcile differences in taste. But is perception of beauty really as arbitrary as this saying is taken to imply? Is there a consistent explanation for the aesthetic aspect of objects that would account for the variance of aesthetic pleasure they invoke on different beholders? The first part of this paper will investigate the notions behind aesthetics and will attempt to address the role of aesthetics in engineering design. The second part of the paper will examine one of the authors' preference of a graphical programming environment (LabVIEW) over a textual programming environment (MATLAB) in his Digital Signal Processing and Communication Systems classes from an engineering design aesthetics point of view.

I. INTRODUCTION

In engineering, we are used to making performance comparisons. We usually choose our tools based on quantifiable parameters. Consequently, when we are faced with a decision on which programming environment to use in one of our courses, it is natural for us to ask questions on the robustness, speed, efficiency and cost of the alternatives presented to us. We also consider how widely the candidates are accepted in industry, a pragmatic consideration as well as an indirect measure of their performance. We take into account the fit between the programming environment and the subject at hand. All other factors being equal, or nearly equal, we might finalize our decision on the familiarity of the package to us. In all our deliberations on the choice of a computer tool, we do not think about aesthetics, at least not at the initial stage. It was therefore revealing to realize that the choice of the programming environment for one of the authors' (Tanyel) *Digital Signal Processing* (*DSP*) and *Communication Systems* courses was based on his aesthetic preference. As part of his orientation, he had sat in on Adams' *Technology and Society* class, which undertakes to examine and critique the relationship between technology and other areas of western society, and was particularly impressed with the discussion on aesthetics in that class. Upon the realization that aesthetic preference is a strong factor in the choice of software for these courses in signal processing, we have undertaken to look at the role of aesthetics in engineering design and to examine this choice in light of our discussion on aesthetics. Therefore, this paper will start with an overview of aesthetics, particularly in engineering design. We will then discuss the software packages at Dordt College appropriate for signal processing and proceed to explain why one of the packages won over the other from an aesthetic point of view.

II. ENGINEERING DESIGN & AESTHETICS

What is the role of aesthetics in engineering design? For that matter, what do we mean by aesthetics? This section will be devoted to investigating these questions. However, before we delve into these subjects, we need to reiterate our underlying notions about technology.

It is our primary position that technology is value-laden. In other words, technology proceeds out of the human experience and is therefore affected by the predispositions and commitments of the human beings who shape it. We will leave the elaboration of this position to Monsma et al [1] and Adams [2] as we state the implications of our position. Because technology is value laden, we need to consider what kind of values our technological choices will uphold, which leads us to the question of normativity in technology.

Normativity prescribes how something ought to be. Assuming a holistic outlook on technological artifacts, we assert that they function in all aspects of reality. The question then is how is an artifact to function? In other words, what norms should a technological artifact follow? We espouse the following normative principles for responsible employment of technology [1]:

Cultural Appropriateness: An artifact has to fit the culture in which it is used.

Openness and Communication: Artifacts must be accompanied by proper communication about their operation, their safety and hazards, their maintenance, etc.

Stewardship: The artifact should not waste resources.

Delightful Harmony: This is the norm that deals with aesthetics. The artifact should be a “joy” to use.

Justice: The artifact should promote justice.

Caring: The artifact should promote the well-being of its user.

Dependability: The artifact should function safely and dependably.

Monsma et al associate these principles with design [1] but it is safe to assume that the same may be considered when selecting from among existing technological artifacts. Whereas technological artifacts should satisfy all of these principles, we are going to probe “Delightful Harmony” which deals with aesthetics. Monsma et al [1] claim that this principle implies three qualities: a) the artifact must be effective, or competent; b) it should be “pleasing and satisfying to use;” c) it must promote harmonious relationships. Whereas the first and the third of these qualities are somewhat objective, the second one seems to be directly related with aesthetics, which conjures up notions of arbitrariness. “Beauty is in the eye of the beholder” we say, as if to mean there may be no explanation to aesthetic preferences.

So, what is aesthetics? Can we agree on good engineering design aesthetics? Adams asks these questions and reviews a number of views on aesthetics before he settles for one in his doctoral work [2]. The satisfactory answer comes from Seerveld, who has developed a unique understanding of the subject. According to Seerveld, the aesthetic derives from “playfulness” which incorporates harmony and beauty as distinct qualities [2]. In fact, Seerveld coins the term “allusivity” to describe it. An allusive object is an object that is suggestive of something else in a subtle manner, with nuance. Seerveld suggests that “allusivity is the central core of aesthetic meaning” in [3]. One can therefore equate aesthetic quality with the quality of subtle suggestiveness or nuance.

Adams analyses the Statue of Liberty with Seerveld’s understanding of aesthetics [2]: The female figure symbolizes liberty. The book in her left hand represents “the law”. The torch lifted high is an allusion to the “light of reason”. And her crown suggests that she is sovereign, a subtle hint that liberty is highly valued in this land.

This notion of allusivity helps explain the variation in the perception of beauty. For many people, the smell of aloe vera alludes to freshness, and is a pleasant aroma. In fact, that used to be the case for Tanyel until he had to change diapers. Diaper and wet “wipes” manufacturers have scented their products with the smell of aloe vera to make changing babies a more pleasant experience. However, repeated exposure to this scent while changing babies has conditioned Tanyel to dislike the scent and he now avoids aloe vera scented lotions or shaving creams.

Furthermore, the explanation of aesthetic quality by allusivity implies that some will not appreciate the aesthetic qualities of an object if the allusion eludes them. In order to appreciate the "beauty" of the Chrysler PT Cruiser, one has to remember the cars of the earlier part of the 20th century to which the styling of the PT Cruiser alludes.

If we associate allusivity with aesthetics, does the lack of it imply ugliness? There used to be a portrait of one of the presidents of Drexel University at its library (it may still be there). Every time Tanyel went up to the second floor of Drexel's library, he would be looking straight into the portrait and would think that it is a very bland, unappealing picture. But there was nothing wrong with the picture; in fact, it was a very good replica of a man in suit, almost a photographic image. Looking back at that experience, we can interpret it in this way: in the 20th century, the development of the art and technology of photography had certain implications. We no longer depended on painters to produce exact replicas of objects, that job was delegated to photography. On the contrary, we came to expect and appreciate more subtle suggestions in paintings. So here was this painting with no subtle suggestions, but a direct reference to the object it was depicting. The lack of subtlety made the picture aesthetically bland.

Now that we have discussed aesthetics in the general sense, how do we apply this concept in engineering design? Adams incorporates the principle of delightful harmony and argues that "aesthetically good engineering design is that which embodies technological allusivity" [2]. To clarify, he states "Technological allusivity in engineering design is achieved when the design successfully suggests a (delightfully) harmonious interaction, *at the human-technical interface*, whereby the product dissolves into an extension of the user." In this definition, the suggestion refers to allusivity. The human-technical interface deals with ergonomics while the phrase "dissolves into an extension of the user" emphasizes successful ergonomic design.

An aesthetically well-designed car, first and foremost, dissolves into an extension of the driver's ability to travel. However, as with any object, a car has multiple functions. For most people, it is not just a device to get from point A to point B. It may also be a status symbol. Therefore its allusions to independence, wealth, prestige or power also contribute to its aesthetic appeal. This is most apparent when one sees a very modest car that has been modified to have big wheels, spoilers and throaty-sounding exhaust systems – to give the allusion (which, in most cases, is mere illusion) to power and speed.

III. FROM AESTHETICS TO COMPUTER TOOLS: SOFTWARE PACKAGES FOR SIGNAL PROCESSING

A survey of literature for signal processing and a related field, control systems, will reveal that MATLAB and C are the most prevalent choices for number crunching in these fields [4]. The UNIX operating system and the C programming language on which it is based have been popular in electrical engineering, especially in the areas of signals and systems. One of the authors' doctoral research was done entirely on the UNIX operating system using C [5] while books have been devoted to publicizing algorithms in C [6].

MATLAB is an acronym derived from the words "Matrix Laboratory". The software was originally developed for matrix manipulation, but over time it has acquired capabilities far beyond the original intent and has become an interactive system and also a programming language for scientific and technical computation [7]. Over the years *MATLAB Toolboxes*, sets of functions written in the

MATLAB language that make it convenient to carry out calculations, to build models and to perform analysis in certain areas, have been developed enhancing the study of and research in subjects like control systems, communication systems, and digital signal processing. Books devoted to the study of these subjects using MATLAB are a testimony to the wide acceptance of MATLAB [8-10].

LabVIEW is another acronym from the words “Laboratory Virtual Instrumentation Engineering Workbench.” This software package is based on the concept of data flow programming and is particularly suited to test and measurement applications [11]. The three important components of such applications are data acquisition, data analysis and data visualization. LabVIEW offers an environment which covers these vital components. The full development version of LabVIEW offers a wide range of functions that can be used in signal processing applications [12].

At the time the decision was made to include computer tools for DSP, both MATLAB and LabVIEW were available for the engineering students at Dordt College. In fact, MATLAB was taught in the lab part of the introductory course to engineering (EGR 103) and was used in EGR 221, *Linear Systems* course which all engineers take. LabVIEW’s full development system was also available in the electronics lab, mainly for the purpose of instrument control and data acquisition through IEEE 488 and RS 232. The decision to use LabVIEW in DSP was based on two reasons [12]: (a) Tanyel’s enthusiasm for the software package and his desire to share his “joys” with the class and (b) LabVIEW programming environment’s allusion to block diagrams: “processes that [are best] describe[d in terms of] what happens to various inputs to achieve an output, so easily depicted by block diagrams in control systems, communication systems and DSP, are better candidates for simulation and/or realization in a graphical programming environment than in a textual environment.” [4].

IV. An Aesthetic Look at Software Packages

In 1994, in his introductory section on LabVIEW Tanyel writes [13]:

“Then came LabVIEW. Those of us who had enjoyed programming despite its difficulties instantly fell in love with this new toy. It was different, it was visual. Its original niche, the *virtual instrument* (VI) looked like a[n actual] instrument on the computer screen. No longer did we have to respond to cold, sometimes condescending statements like:

```
>   Please enter the upper cut-off frequency:
>   ?
```

by typing on the blinking question mark. We set our parameters by turning dials, sliding switches and we told the computer to go! The answer came back in the form of familiar meters, pointers, graphs and colorful LEDs. This is the mode of operation in which scientists and engineers have enjoyed working in their labs. The computer finally became a versatile instrument (perhaps another interpretation for the acronym *VI*?).”

Allusivity in the User Interface

We note here that LabVIEW programs are called virtual instruments (VIs). The enthusiasm expressed in the above quotation is for the user interface of the software environment. A well-designed VI’s user interface alludes to an actual instrument that engineers and scientists are so accustomed to see and have come to love in their labs!

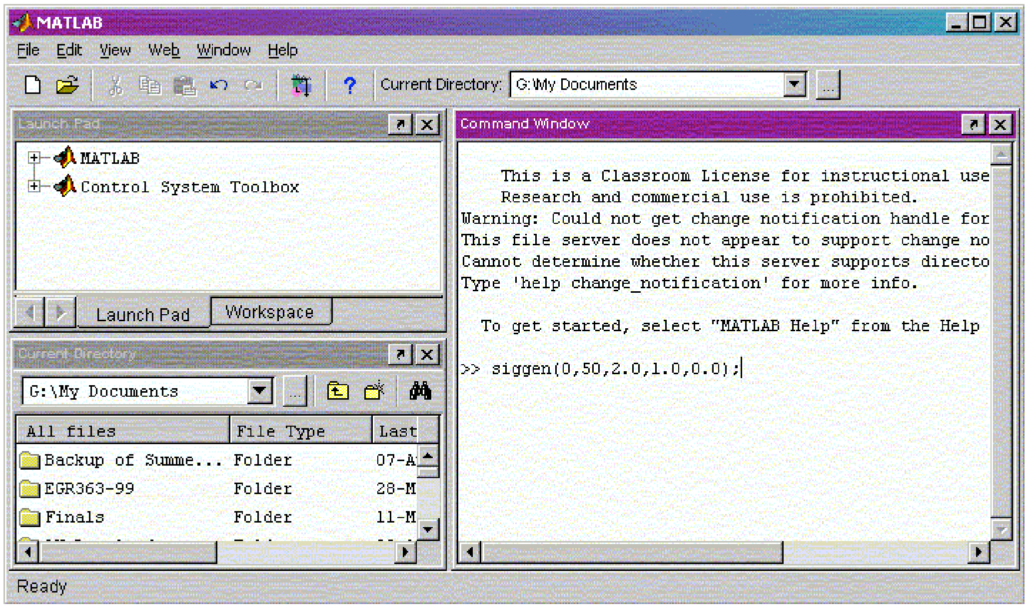

Figure 1: Calling a "signal generator" function in MATLAB.

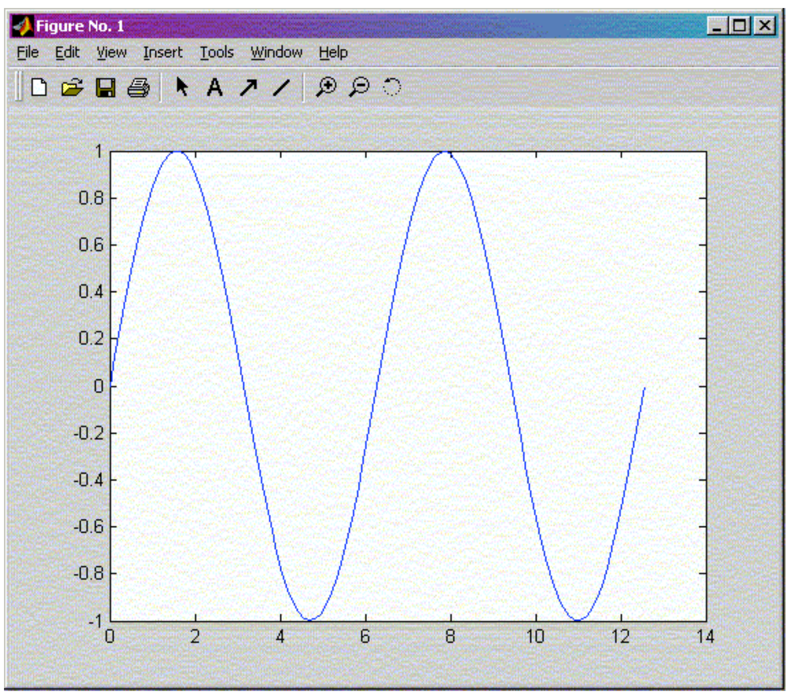

Figure 2: Result of the call in Fig. 1.

Consider a function written in MATLAB that mimics a signal generator. The call to such a function is depicted in Fig. 1 while Fig. 2 displays the result of such a call.

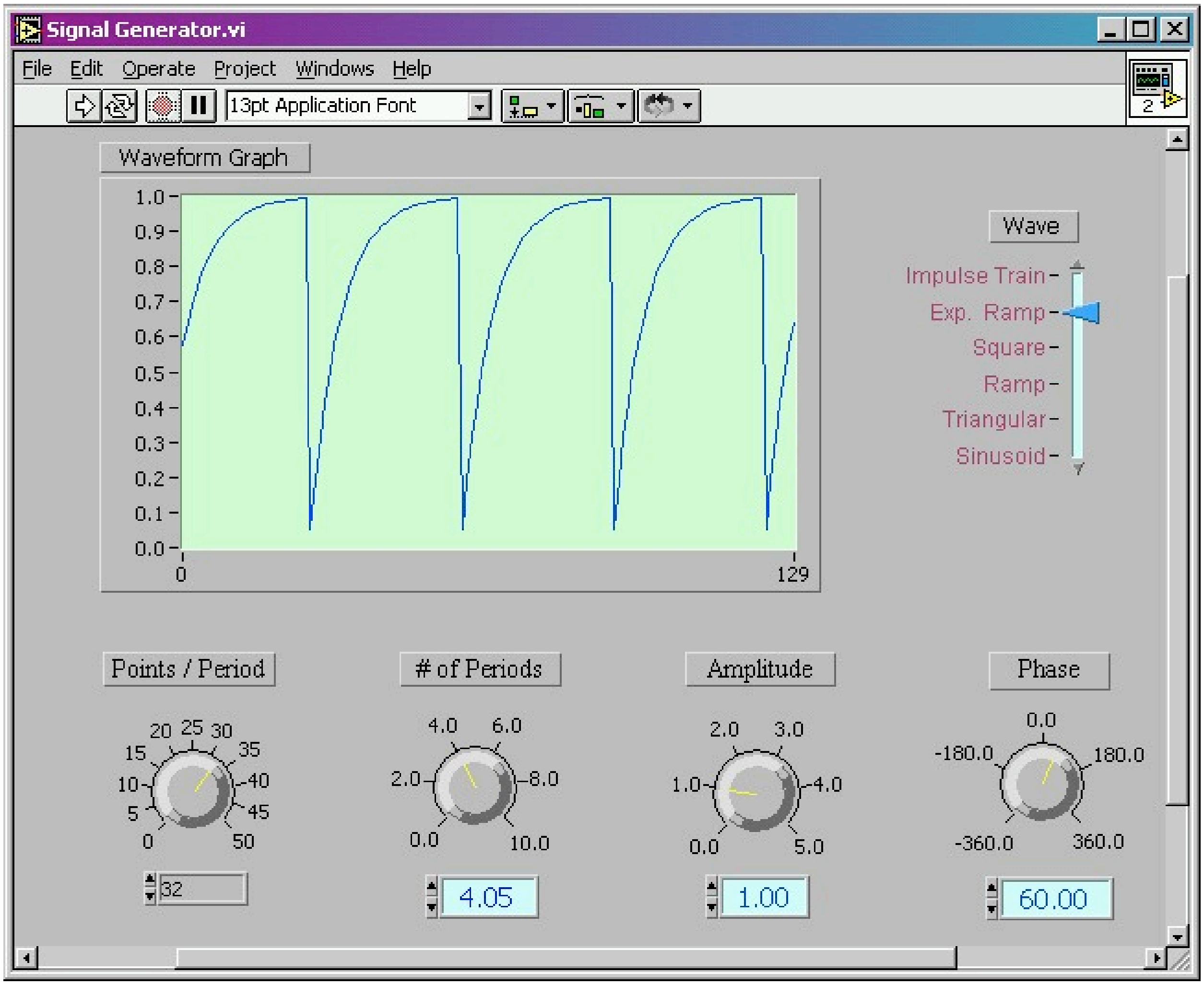

Figure 3: The user interface of a VI that mimics a function generator.

Now we ask you to compare these two figures to the front panel (the user interface) of a VI that mimics a signal generator (Fig. 3). The allusion to an actual instrument has made the program not only aesthetically pleasing, but also easier to understand. By looking at this front panel, we can tell that the program is displaying a periodic Exp[onential] Ramp with the following specifications: 32 points generated per period, 4.05 periods displayed, with amplitude = 1 and a phase angle of 60. Although the unit of the phase angle is not specified, one can infer from the numbers on the "knob" controlling this parameter that the unit is in degrees. Furthermore, we realize that we can also "generate" periodic sinusoidal, triangular, ramp, square and impulses by noticing the selections of the "sliding switch" (a more detailed description of this VI may be found in [14]). The list of arguments in Fig. 1 (0, 50, 2.0, 1.0, 0.0), on the other hand, does not suggest any of the parameters to the average user unless he/she has memorized the order of the variables. The user interface of the VI depicted in Fig. 3, with its allusions to an actual instrument, is more intuitive to use, "suggesting successfully a (delightfully) harmonious interaction, whereby the program dissolves into an extension of the user." We suspect the "dissolving into an extension of the user" bit is a little exaggerated, but it is safe to say that the program is easy to operate. A further feature that has been incorporated into these VIs is their ability to be run in "continuous mode" which is akin to putting the whole VI in a while loop, running it repeatedly until the stop button is pressed. In this mode, the VI continuously monitors its inputs and adjusts its output(s). This feature makes the VI resemble a live instrument, rendering "what if" type explorations much more real. In a static environment like that of Fig. 1, "what if" type questions may be explored by typing the command string over and over with various parameters, but the immediacy is lost. The allusion to an instrument responding to the user's inputs is lost, diminishing the joyful interaction with the program.

Allusivity in Programming

Many processes in communication systems, digital signal processing and control systems are described with block diagrams. Therefore, programming environments that use block diagrams (such as LabVIEW and SIMULINK) have an advantage over textual languages, such as C and MATLAB. Consider the block diagram in Fig. 4 that depicts FM detection. This block diagram may be found in a typical textbook on communication systems [14].

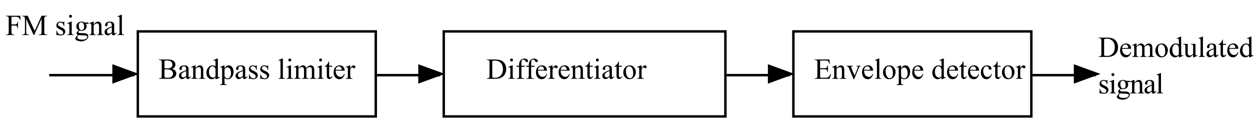

Figure 4: The block diagram that describes FM detection in a typical communication systems textbook.

Fig. 5 is a graphical program that simulates FM detection in LabVIEW. The LabVIEW blocks of Fig. 5, corresponding to the block diagram in Fig. 4 are: `Y[i]=Clip{X[i]}.vi` that performs the function of the limiter, `Derivative x(t).vi` which performs differentiation and `ENV_DET.vi` which performs envelope detection. The first two blocks are part of the full development version of LabVIEW 5.1 and the last block, the envelope detector, is part of the communication systems toolkit described in [14].

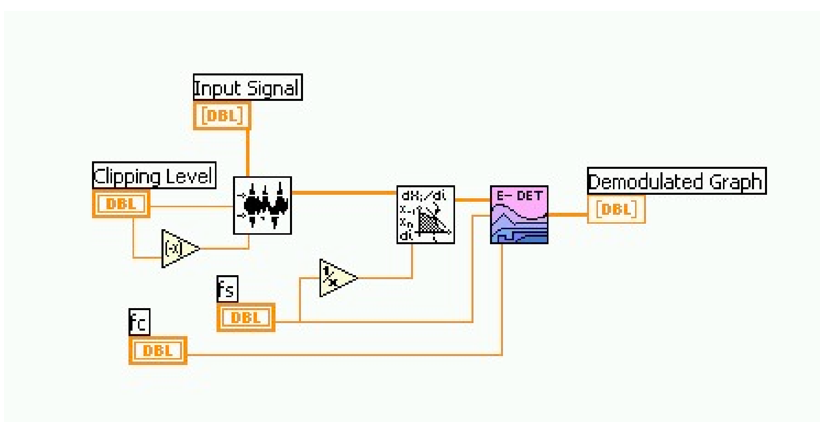

Figure 5: A graphical program that simulates FM detection in LabVIEW.

We note that the program in Fig. 5 suggests the process depicted in Fig. 4 while the suggestion given by the following MATLAB (pseudo)code to Fig. 4 is not as clear:

```
% This is a script file FM_Det.m
clpsgn = clipper(inptsgn, 0.6, -0.6);
di = 1/fs;
difsgn = differ(clpsgn,di);
demdsgn = e_det(difsgn, fs, fc);
plot(demdsgn)
```

Another feature of LabVIEW is the ability to design icons for the various VIs that one writes. The Icon Editor (Fig. 6) works like a mini paint program, allowing the user to create allusions to the processes that the VIs perform. Fig. 6 depicts the Icon Editor at the end of the design of an icon for the VI `PoleZeroDiag.vi`, which displays the poles and zeroes of a discrete system. Anyone familiar with discrete systems can recognize the suggestion by the icon designed here. This feature provides a visual allusion to the function of the VI in addition to the allusion suggested by its name. It thus aids documentation and fosters *Openness and Communication*, one of the normative principles listed in section II.

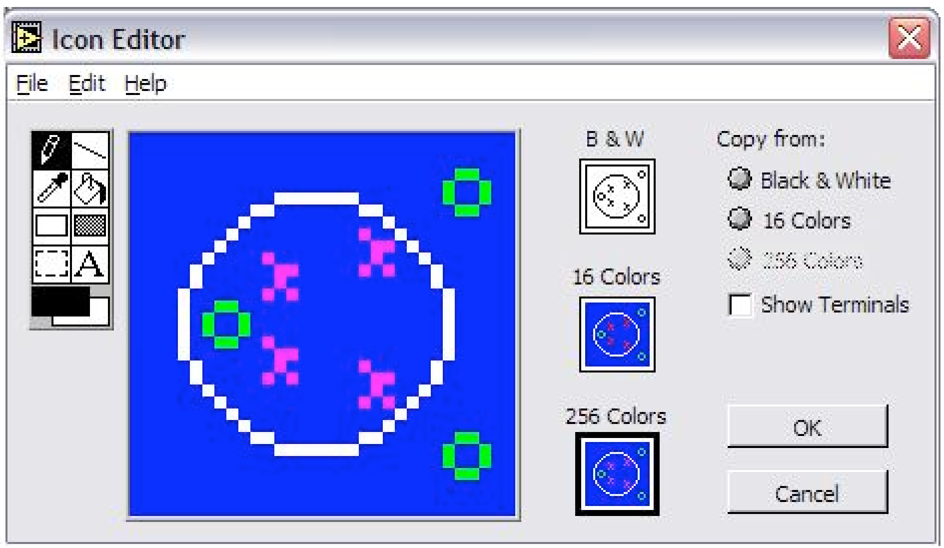

Figure 6: The Icon Editor in LabVIEW.

V. DISCUSSION

We have looked at the choice of computer tools for DSP and communication systems at Dordt College from an aesthetic point of view. To do this, we have stated our view of the role of aesthetics in engineering design and associated aesthetics with allusivity that results in a harmonious relationship between the technological artifact and its user. In section III, we have reviewed the computer tools available in these areas and in section IV, we have shown examples where one package's aesthetic qualities surpass those of the other, rendering the one a "joy" to use. We note that we have compared these packages (LabVIEW and MATLAB) only aesthetically. We would also add with emphasis that our electrical engineers are exposed to both of these packages in the curriculum; the preference noted in this paper reflects only a portion of the curriculum. By no means do we imply that all DSP courses around the country should drop using MATLAB and embrace

LabVIEW. We have simply given an argument as to why one professor finds working with LabVIEW more rewarding.

Students also appreciate the exposure to LabVIEW. We have noted that almost all the senior design projects have incorporated LabVIEW in one way or another this year. In fact, one of the senior design project teams has combined the strengths of both LabVIEW and MATLAB, using the LabVIEW interface to call MATLAB scripts. As Tanyel notes in [12]: "Part of the enthusiasm is due to the aesthetics of LabVIEW virtual instruments. I have seen many a student revisiting and perfecting the way the front panel looks long after his/her VI has achieved its computational goals and long after the class period has ended." If it will make them stay after class of their own volition, it is worth the inclusion in the curriculum.

ACKNOWLEGEMENTS: The LabVIEW software used in this course was purchased through a grant from Johnson Controls in Holland, MI. The authors would like to thank Johnson Controls for the upgrade of the electronics lab which also benefited the courses mentioned in this paper. The authors would also like to thank Kathrine Nguru, who programmed some of the VIs depicted in this paper for the Communication Systems course [14].

BIBLIOGRAPHY

1. Monsma, S. V., Leegwater, A., Christians, C., Schuurman, E., Dykema, E. R. and Van Poolen, L., *Responsible Technology*, Grand Rapids, MI: W. B. Eerdmans Publishing Company (1986).
2. Adams, C. C., *An Analysis and Solution to the Two-Cultures Problem in Undergraduate Engineering Education*, The University of Iowa, Iowa City, IA, July 1996.
3. Seerveld, C., *Rainbows for the Fallen World: Aesthetic Life and Artistic Task*, Toronto, ON: Toronto Tuppence Press (1980).
4. Viss, M. and Tanyel, M. "From Block Diagrams to Graphical Programs in DSP," in *Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition*, Albuquerque, NM, June 24-27 2001.
5. Tanyel, M., *Enhancement of EGGs by Linear Prediction and Nonlinear Filtering*, Doctoral Dissertation, Drexel University, Philadelphia, PA, Aug. 1989.
6. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge: Cambridge University Press (1993)
7. Etter, D. M. and Kuncicky, D. C., *Introduction to MATLAB*, Upper Saddle River, NJ: Prentice Hall (1999).
8. Proakis, J. G., Salehi, M., *Contemporary Communication Systems using MATLAB*, Pacific Grove, CA: Brooks/Cole (2000).
9. Frederick, D., Chow, J., *Feedback Control Systems using MATLAB and the Control Systems Toolbox*, Pacific Grove, CA: Brooks/Cole (2000).

10. Ingle, V. K., Proakis, J. G. *Digital Signal Processing using MATLAB*, Pacific Grove, CA: Brooks/Cole (2000).
11. Chugani, M. L., Samant, A. R., Cerna, M., *LabVIEW Signal Processing*, Upper Saddle River, NJ: Prentice Hall (1998).
12. Tanyel, M. "Enhancing the DSP Toolkit of LabVIEW," in *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*, Montréal, QC, June 16-19 2002.
13. Tanyel, M., *Engineering Explorations with LabVIEW*, Philadelphia, PA: Harcourt Brace Custom Publishers (1994).
14. Tanyel, M., Nguru, K., "Preparation of a Virtual Toolkit for Communication Systems," in *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*, Montréal, QC, June 16-19 2002.

MURAT TANYEL

Murat Tanyel is a professor of engineering at Dordt College. He teaches upper level electrical engineering courses. Prior to teaching at Dordt College, Dr. Tanyel taught at Drexel University where he worked for the *Enhanced Educational Experience for Engineering Students* (E^4) project, setting up and teaching laboratory and hands-on computer experiments for engineering freshmen and sophomores. For one semester, he was also a visiting professor at the United Arab Emirates University in Al-Ain, UAE where he helped set up an innovative introductory engineering curriculum. Dr. Tanyel received his B. S. degree in electrical engineering from Bo_aziçi University, Istanbul, Turkey in 1981, his M. S. degree in electrical engineering from Bucknell University, Lewisburg, PA in 1985 and his Ph. D. in biomedical engineering from Drexel University, Philadelphia, PA in 1990.

CHARLES ADAMS

Charles Adams is a professor of engineering and the dean of Natural Sciences Division at Dordt College. Apart from his duties as dean, he teaches the introductory engineering course and the senior level *Technology and Society* course. Dr. Adams was instrumental in establishing the engineering department at Dordt College and has taught in and headed the department from 1979 to 2000, when he became dean. Prior to teaching at Dordt, he had worked in the aircraft industry as an analytical engineer for three years and was the head of the Science Department at a Christian high school in New Jersey in 1979. Dr. Adams received his B. S. degree in chemical engineering from New Jersey Institute of Technology in 1968, his M. S. degree in mechanical engineering from Rensselaer Polytechnic Institute in 1971, his M. A. in education from Montclair College in 1977, and his Ph. D. in philosophy of engineering education from the University of Iowa, Iowa City, IA in 1996.