# Design from Zeroth Principles

**Jordan W. Suchow (suchow@berkeley.edu)**
**Michael D. Pacer (mpacer@berkeley.edu)**
**Thomas L. Griffiths (tom_griffiths@berkeley.edu)**
Department of Psychology,
University of California, Berkeley, USA

## Abstract

A successful design accounts for the structure of the problem it is aimed at solving. When it is a human-directed design, this includes the expectations of its users. How do we arrive at such a design? One approach starts from first principles (e.g., simplicity, unity, symmetry, balance) to evaluate the quality of proposed designs. Here, we introduce *design from zeroth principles*, a form of human-in-the-loop computation that synthesizes a design that conforms to its users' expectations. The technique begins by constructing a transmission chain seeded with a random design. Each user in the chain is exposed to the design and then recreates it, passing along their recreation to the next user, who does the same. Through this iterative process, the users' perceptual, inductive, and reconstructive biases directly transform the initial design into one that is better fit to human cognition. Such designs are easier to learn and harder to forget. We evaluated the approach in three domains – stimulus–response mappings, vanity phone numbers, and letter placement in typeset words – and show that it produces a good design in each.

**Keywords:** design, cognitive ergonomics, inductive bias, transmission chain, user interface

## Introduction

Successful design in the rationalist tradition begins by evaluating the problem that a designed object or system aims to solve: the goals, any constraints imposed by the environment or by human factors, and the surrounding context, broadly construed (Simon, 1996). Another tradition appeals to principles that are purported to be universal – simplicity, balance, unity, order, liveliness – rather than to direct considerations of function (Lidwell, Holden, & Butler, 2010; White, 2002). Following these principles will, in theory, lead to successful designs.

But what makes a design successful? Certainly, in cases where the design acts as the interface between a user and a system, the success of the design hinges in part on the user's experience in working with the design. Users have expectations about how to interact with the world to accomplish their goals, and a good design conforms to those expectations — i.e., when humans are the users, good designs fit the human mind. Practices have developed around ensuring this, including techniques to compare variants of a design through statistical hypothesis testing (e.g., A/B testing), to measure performance under user-focused metrics (e.g., usability research), and to elicit feedback from potential users (e.g., focus groups).

In cognitive science, a person's expectations can be described as *perceptual*, *inductive,* and *reconstructive* biases, as they pertain to perception, learning, and memory, respectively. Bias in this sense means merely that the distribution of expected designs is not uniform — some match expectations better than others. Tools and techniques from the domain of cognitive science that identify, extract, and amplify these biases can thus aid designers in their quest to find cognitively fit forms.

Transmission chains are one such technique for extracting and amplifying biases in memory and learning. Originating in early experiments by Frederic Bartlett, transmission chains pass information from one person to the next, much like in the children's game *Telephone* (Bartlett, 1932). At each step of the chain, the transmitted information is transformed. So long as a few technical conditions hold, repeated application of a transformation leads to erasure of the information contained in the input, leaving behind a signature of the transformation process itself.

In this paper, we introduce a method that uses transmission chains to synthesize a design. Our technique begins by constructing a transmission chain seeded with a random design. Each user in the chain is exposed to the design and then recreates it, passing along their recreation to the next user, who does the same. Through this iterative process, the users' inductive and reconstructive biases directly transform the initial design into one that is better fit to human cognition. No formal design principles are assumed. Thus, we call this process *design from zeroth principles*.

The plan for the paper is as follows. We begin with a technical description of the transmission chain technique and its ability to amplify biases in perception, learning, and memory. Next, we apply the technique in three domains: stimulus–response mappings, vanity numbers, and letter placement in typeset words. We conclude with a discussion of how our method can be extended and elaborated, some criteria for design problems that might benefit from the technique, and possible modifications to the transmission-based scheme that relate to other forms of human-in-the-loop computation.

## Transmission chains reveal biases

In a transmission chain, information is passed from one person to the next. In the children's game *Telephone*, for example, a child invents a sentence and whispers it to the next child in line, who then does the same. By the time the sentence reaches the end of the chain, it has changed. Hilarity ensues.

This kind of system can be formally modeled as a Markov chain, a stochastic process in which transformations are defined by a transition matrix specifying the probability of going from one given state to any other state (Kalish, Griffiths, & Lewandowsky, 2007). In *Telephone*, for example, the transition matrix defines the probability that a given sentence will transform into a given another (e.g., that "laid him on the green" becomes "Lady Mondegreen").

If a Markov chain obeys certain requirements,[1] it can be proven that it will eventually converge to a *stationary distribution*, a distribution of states unchanged by the transformation. This means that even if we seed a chain with a random state, after enough steps, the information contained in the input will be lost, leaving behind a signature of the transformation process itself. Identifying the stationary distribution of a Markov chain requires having a model for the probability with which it transitions from one state to another. In the case of transmission chains, one such model is provided by assuming that perception, learning, and memory follow the principles of Bayesian inference. Given an observed stimulus $d$, people consider hypotheses $h$ about its nature, and then produce a reconstruction $d'$. The Bayesian analysis of transmission chains assumes hypotheses are sampled from the posterior distribution $p(h|d) \propto p(d|h)p(h)$, where $p(d|h)$ gives the probability of seeing $d$ if it were generated from $h$ (known as the *likelihood*) and $p(h)$ is the *prior* distribution over hypotheses and encodes people's expectations about the prevalence of different hypotheses. If $d'$ is generated by sampling from the likelihood, then the stationary distribution of this Markov chain is the *prior predictive distribution* $p(d) = \Sigma_h\, p(d|h)p(h)$. Consequently, we should expect the outcome of transmission to reflect people's expectations, as expressed in the form of this prior predictive distribution.

Cognitive scientists have used transmission chains in studies of *serial reproduction* and *iterated learning* to study reconstructive biases in memory and inductive biases in learning. One of the first transmission chain studies was conducted by Bartlett, who asked participants to view an image, wait some time, and then draw what they remembered having seen (Bartlett, 1932). The reproduction was then passed to the next participant in the chain. Over the long run, the drawings transformed (e.g., a drawing of an owl became one of a cat). Serial reproduction experiments like this one allow us to reveal people's perceptual and reconstructive biases.

Iterated learning is similar to serial reproduction, but rather than reproducing what was observed from memory, participants demonstrate what they have learned by generalizing to new examples. For example, in Kalish, Griffiths & Lewandowsky (2007), participants first learned a functional relationship between two magnitudes (the length of a rectangular bar and the width of another) by observing pairs. Notably, participants were then tested on some examples that they had never directly observed. Responding to these novel stimuli requires generalization beyond what they have observed. The authors found that the functional form passed in these transmission chains gradually reverts to a linear relationship regardless of the data that seeds the chain. From this, they concluded that this functional form is what people expect.

If a good design is one that fits the expectations of its users, then any difficulty in perceiving, learning, or remembering a design indicates that it may be inconsistent with the user's cognitive biases. By passing the design through a transmission chain, the users' inductive and reconstructive biases will transform the initial design into one that is better fit to human cognition.

## Experiment 1: Stimulus–response mappings

In which direction should a screw be turned in order to drive it further into wood? Which light switch should be flipped to turn off the patio light? And which knob should be turned to light the front left stove burner? Assigning these mappings are design decisions, and some mappings are better than others. Designs with *stimulus–response compatibility* offer a simple and clear mapping between an action and a response, leading to shorter reaction times and lower rates of error (Fitts & Seeger, 1953; Proctor & Reeve, 1989; Kornblum, Hasbroucq, & Osman, 1990).

In Experiment 1A, we applied design from zeroth principles to stimulus–response mappings between light switches and lights. Experiment 1B evaluated the resulting mappings.

### Methods, 1A

Experiment 1A constructed a transmission chain where participants passed along a mapping between light switches and lights. The chain was seeded with a random mapping.

**Participants.** We recruited 100 participants on Amazon Mechanical Turk, an online crowdsourcing platform. Each participant was paid $0.25 for a few minutes of work.

**Stimulus.** The stimulus was a depiction of a set of six light switches and six lights (Fig. 1). When pressed, the switch turned on one of the lights for 1000 ms.

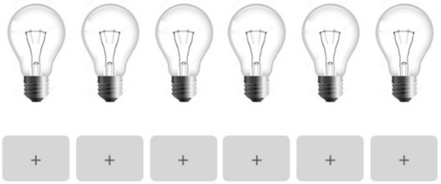

Figure 1. A set of six light switches and lights. Which switch maps to which light?

[1] The condition that needs to be met is that the Markov chain must be ergodic (i.e., starting from any state you can eventually reach any another state; the expected number of steps to reach each other state is finite in expectation; and returning to any one state does not only occur as a multiple of some $k > 1$).

**Procedure.** First, the participant learned the mapping. On each trial, one of the switches was highlighted in green. The participant was instructed to press the switch and observe what happened. Each switch was highlighted once over the course of training, such that the participant observed the entire mapping. Then the participant was tested on what they had learned. On each of six trials, one of lights was highlighted with a bounding box. The participant was instructed to turn on the highlighted light by pressing the corresponding switch. Once pressed, the switch became disabled and could not be reselected. There was no feedback. The order in which the pairings were learned and tested was random. Together, the participant's six responses define a new mapping that was then passed along to the next participant in the chain. There were 10 chains of 10 participants.

### Results & Discussion, 1A

Over time, mappings in the chain became more regular, by the tenth generation coming to resemble the solution where all the switches are mapped to the light directly above them (average Kendall rank correlation coefficient $\tau = 0.96$; Fig. 2). In fact, 6 of 10 chains converged to exactly this solution.

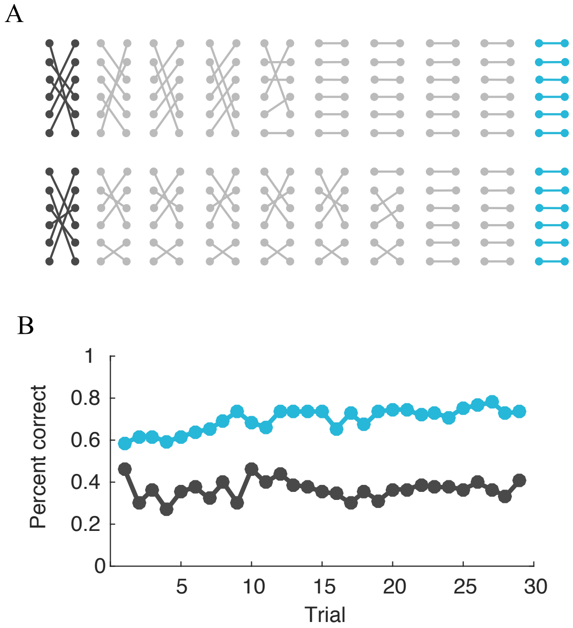

Figure 2. Designing an intuitive mapping between switches and lights from zeroth principles. (A) Two chains that began with random mappings converged to the same design (B). The designs from the chain's first generation (black) are considerably harder to learn than those from the last generation (blue).

### Methods, 1B

Experiment 1B evaluated the designs synthesized through Experiment 1A by comparing the performance characteristics of designs from the beginnings and ends of the chains.

**Participants.** We recruited 200 participants on Amazon Mechanical Turk, an online crowdsourcing platform. Each participant was paid $0.50 for a few minutes of work.

**Stimulus.** The stimulus was the same as in Experiment 1A.

**Procedure.** Participants learned a mapping. On each of 30 trials, one of the lights was highlighted with a bounding box. The participant was instructed to select and press the switch that would turn on the highlighted light. When pressed, light corresponding to the pressed switch turned on, providing the participant with feedback. Half the participants learned the random mappings drawn from the beginning of the chains in Experiment 1A; the other half learned the stopping states of the chains.

### Results & Discussion, 1B

Performance was better for designs from the stopping state of the chain than for designs from its starting state (proportion correct of 0.70 vs. 0.38; independent samples $t$-test, $t(198) = 7.1$, $p < 0.0001$; Fig. 2B).

## Experiment 2: Vanity numbers

A *vanity number* is a telephone number with an easily remembered sequence of digits — e.g., 1 (212) 222-2222, 1 (800) 800-8000, or 1 (202) 456-1111. There is an active market for these numbers, where their pricing depends in part on intuitions for how easily they can be held in mind (Haucap, 2003). Success depends on how memorable the number is, and that success is reflected in the marketplace. Valuable vanity numbers are highly sought after and are sold for prices that are orders of magnitude higher than those without an obvious pattern. Reasons for buying vanity numbers are idiosyncratic. Apple co-founder Steven Wozniak, for example, collected telephone numbers as a hobby, acquiring 888-8888 soon after the 888 exchange went on the market[2]. Businesses often use them in radio and television advertisements, and occasionally, as in the case of 1-800-Flowers.com, Inc., incorporate them.

In Experiment 2, we applied design from zeroth principles to choose memorable vanity phone numbers. We then evaluated the resulting numbers by measuring their memorability and predicting their market value.

### Methods

In Experiment 2, we constructed a transmission chain where participants passed along 10-digit phone numbers. The chain was seeded with a random phone number.

**Participants.** We recruited 40 participants on Amazon Mechanical Turk.

**Stimulus.** Phone numbers were 10-digit strings formatted as `(xxx) xxx–xxxx`. All telephone numbers were sampled ran-

[2] Not only are telephone numbers with strings of repeated digits memorable — they are also easy to press. Wozniak's 888-8888 was soon swamped by calls from children mashing 8 on their family's home phone (Wolf, 1998).

domly from those following the North American Numbering Plan format.

**Procedure.** First, the participant viewed the phone number for 2 seconds. Then there was a retention interval of 4 seconds. Finally, the participant recreated the phone number by typing it on a keyboard. Twenty phone numbers were remembered and tested in this way.

**Estimating a telephone number's value.** We collected 16,000 telephone numbers and their associated prices from phonenumberguy.com, an online marketplace for vanity numbers. The numbers vary widely in price, from \$199 to \$199,999. We also collected 34,000 telephone numbers from Twilio, a communications company. We set the value of a Twilio number to be \$99, midway between \$0 and one dollar less than lowest vanity number, under the logic that any telephone number worth at least the minimum listed market price would not be available for less. To estimate the value of telephone numbers in the transmission chain, none of which were present in the collected data, we constructed a model of telephone number prices. From each listed number we extracted a set of binary features (Table 1). To simplify the analysis, the features considered only the number's digital representation, ignoring value derived from the phonewords. We then regressed log price on these features. The $R^2$ of the resulting model was 0.53.

Table 1. Features used to predict telephone number pricing and their weight in the model. There were no training examples with seven numbers in a row.

| Feature | Example | $\beta$ (log USD) |
|---|---|---|
| Millions | 1 (415) 7**00–0000** | 1.67 |
| Seven in a row | 1 (415) **777–7777** | 3.48 |
| Six in a row | 1 (415) 8**77–7777** | 1.16 |
| Hundred thousands | 1 (415) 87**0–0000** | 1.69 |
| Thousands | 1 (415) 626–8**000** | 0.60 |
| Hundred–thousands | 1 (415) 5**00**–6**000** | 1.20 |
| Double repeater A | 1 (415) **888**–**7777** | 1.96 |
| Double repeater B | 1 (415) 8**66**–**7777** | 0.49 |
| Mid repeater | 1 (415) **888**–2465 | 0.65 |
| № of unique digits | 1 (415) 326–9087 | –4.27 |
| Eight 9s in a row | 1 (41**9**) **999–9999** | 3.70 |
| Repeated sequences | 1 (415) 6**70–7070** | –0.17 |

## Results & Discussion

Over time, telephone numbers in the chain became more memorable and more valuable. Table 2 shows the telephone numbers from one of the chains and their estimated value.

The average number of correctly reported digits per number increased from 6.95 across the first five generations to 9.29 across the final five. Note, however, the unforgiving nature of telephone numbers — with even a single misremembered digit, a call is unlikely to reach its intended target. Thus we also computed performance under a 0–1 loss function, counting only perfectly recalled numbers as having been remembered at all. The proportion of correctly reported numbers rose from 0.29 across the first five generations to 0.76 across the last five.

The average value of a number in the first generation of the chain was \$119, slightly more than the assigned value of a non-vanity number. By the end of the chain, the average value was higher, rising to \$548 (two sample *t*-test on the log values, $t(38) = 4.76$, $p = 1.75 \times 10^{-4}$; Mann-Whitney $U(38) = 40.5$, $p = 6.44 \times 10^{-6}$).

Table 2. Change points from a randomly seeded telephone number transmission chain and estimated values in USD.

| *i* | Number | \$ |
|---|---|---|
| 0 | (603) 639-5026 | 91 |
| 1 | (603) 639-7843 | 90 |
| 2 | (603) 639-0000 | 214 |
| 8 | (603) 693-1234 | 91 |
| 9 | (603) 693-1294 | 91 |
| 10 | (603) 693-0000 | 216 |
| 20 | (800) 963-0000 | 218 |
| 24 | (800) 936-0000 | 217 |

# Experiment 3: Letter Placement

In typography, the shapes of characters are represented by glyphs with dark and light values spread over space. In sequence, these characters can be formed into words. Many typographic factors contribute to the final location of letters in space — we will refer to the total contribution of these factors as *letter placement*.

The success of printed text depends in part on its ability to be read. Letter placement plays a major role in determining that success. If characters are placed too close together, this hurts the text's *legibility* (the ease of distinguishing between letters). If placed too far apart *readability* (the ease of recognizing groupings of letters into words, sentences, and paragraphs) is worsened; words become harder to distinguish from each other, particularly when the spaces between letters rival that between words

When typesetters manually set metal type to design a page for print, they had to decide not only which typeface to use (and which font of that typeface to use), but with what spacing modifications to use when laying them out on the page. While each glyph had a default width, the spacing between characters could be controlled further through *tracking* (alterations to the spacing between all letters), and *kerning*, adjustments of the spacing between specific pairs of letters (e.g., the "T" and "y" in "Type" look better when brought closer together than their default widths allow)[3].

In today's digital typefaces, default glyph width and kerning information is built into font files and generally does not require manual adjustments by typesetters. Because tracking modifies a font's general spacing properties, it still requires

[3] While tracking was accomplished by adding space between every letter, kerning tended to be used to reduce space between letters. Positive kerns were also possible, but those cases were more often dealt with using ligatures or a new single characters that played the role of both ("fi" → "fi"). We do not address ligatures here.

manual adjustment. However, the difference between a good and a poor digital typeface/font-family can often boil down to the decisions the type designer made when considering (or failing to consider) each font's kerning pairs.

Over a lifetime, someone reading an hour a day will see hundreds of millions of typeset words. We argue that, with so much exposure to printed text, people will come to have strong expectations about the letter placement in printed text. In Experiment 3, we use these expectations to derive letter placements through design from zeroth principles.

## Methods

Experiment 3 constructed a transmission chain where participants passed along typeset words. The chain was seeded with randomly spaced words.

**Participants.** We recruited 200 participants on Amazon Mechanical Turk.

**Stimulus.** Fifteen words set in Helvetica were used for the experiments: Typical, frogs, vacuum, hunchback, Chicago, Year, Egypt, the, eye, kiln, milk, WAVE, fjord, Bring, and Pile. These words were chosen because each has at least one pair of adjacent letters that benefit from kerning (e.g. the W and A in WAVE). The position of the final letter was determined by its position when set in Helvetica (Linotype, v10.0d4e1) at 100 points. In the randomly spaced words used to seed the chain, each letter's position was chosen uniformly over the interval between the first and final letter, with the constraint that the letters are correctly ordered. We defined the space between two letters as the center-to-center distance (in pixels) between the letters' minimum bounding boxes.

**Procedure.** The word was presented for 2 s. After a 4 s retention interval, the first and last letters of the word reappeared in their original position. To the left of the first letter was a repository of the letters not yet placed, starting with the second letter of the word. The participant was asked to reposition the letter into its displayed position in the word. Once moved, the next letter appeared. This continued until all the letters had been placed. The participant was able to readjust the letters as much necessary before submitting a response. The participant was able to submit a response only if all the word's letters were arranged in the correct order.

**Task Error.** To measure the fidelity of a participant's recreated word, we measured the space between pairs of adjacent letters[4] and then computed the mean squared error between these spacings and those specified by the font file. We also computed two benchmarks of error — random spacing and equal spacing. Random spacings were drawn in the same manner as the starting states, as described above. Equal spacings were defined with respect to the center of the letters' minimum bounding box.

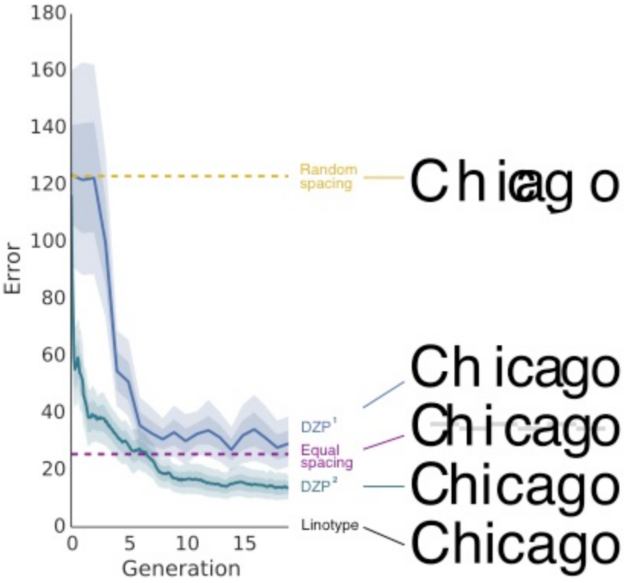

Figure 3. Letter placement in typeset words from zeroth principles, Confidence intervals, indicated by the shaded areas, are ±1 and 2 SE.

### Results & Discussion

The benchmark methods of random spacing and equal spacing produced errors of 120 pixel/word and 24 pixel/word, respectively (Fig. 3, yellow and purple dashes lines). Because the transmission chains are seeded with a random design, the initial performance is identical to that of random spacing. The results soon diverge, however, with design from zeroth principles (blue solid line) outperforming random spacing and approaching the performance of equal spacing. Because the spacing between letters is metric, the method can be improved by aggregating across chains and time. In $\text{DZP}_2$, then, the final design is arrived at by averaging the states visited across all the chains. We found that $\text{DZP}_2$ outperforms equal spacing, with designs more closely resembling those recommended by the Linotype font file.

In addition to the methods reported here, we considered a second approach of framing the problem of the letter placement. In this, the goal was not to position letters correctly between fixed endpoints of the first and last letters, but rather to position each letter, including the last, in sequence. This allowed the length of the word to vary widely. We found that this method did not converge to a good design. In general, letter placements that result from this process were too long. This is counteracted by the fixed-length task that we describe above. Modifications to the method may be able to counteract this lengthening influence.

## General Discussion

These experiments demonstrate that design from zeroth principles can recover good designs without explicit designing. In a series of three experiments, we constructed transmission chains seeded with a random design. Each user in the chain was exposed to the design and then recreated it,

[4] We took the ground truth letter placement to be the centroid of a letter's bounding box when typeset in Adobe Illustrator and transformed into outlines, rounding to the nearest pixel.

passing along their re-creation to the next user, who did the same. Through this iterative process, the users' inductive and reconstructive biases directly transformed the initial design into one that is better fit to human cognition. We demonstrated the technique in three domains: stimulus–response mappings, vanity phone numbers, and letter placement in typeset words.

Our method can be extended in ways inspired by its computational basis in Markov chains. Rather than estimating the prior predictive distribution over states, one can estimate the full transition matrix. This helps compensate for failures of convergence that can occur with short chains. Another method for avoiding undue influence from the starting states is to exclude the *burn in* trials, a standard procedure for discarding initial samples in Markov Chain Monte Carlo simulations (Murphy, 2012). Convergence can be detected using standard diagnostic tools for estimating whether an MCMC sampler has converged (Cowles & Carlin, 1996).

A second direction in which the current method can be extended is to combine the technique with other forms of human-in-the-loop computation. For example, by including an explicit selection layer in which participants evaluate a design and determine whether a solution persists until the next generation, the process can be made more robust to the kinds of errors introduced by various experimental designs. Such an approach would bring the method closer to other forms of human-in-the-loop computation such as interactive evolutionary computation (Takagi, 2001)

Design from zeroth principles can be used on other design problems, too. Consider for example *collation*, which requires choosing a rule for how a set of items will be ordered. Often a well-established convention makes the choice an easy one. For example, alphabetical order is used widely, dictating the arrangement of words in a dictionary, topics in a reference book's index, and quotes in a newspaper's stock table. Other collation methods are conventional in other domains. *The New York Times*' NFL sports standings, for example, are arranged first by division and then by win–loss record, because these features are important to fans who attend to pennant races. *U.S. News & World Report* ranks colleges according to their own quantitative metric of institutional quality, best first, because their readers care about who came out on top. A collation method can be synthesized through by having participants search for items in a list and then try to recall their order, passing along their collation to the next participant.

Some domains of design are unlikely to benefit from design from zeroth principles. One might imagine, for example, that given people's notorious difficulty in estimating their own understanding of the mechanism of helicopter flight and other complex phenomena (Rozenblit & Keil, 2001), a helicopter engine and rotor would be unwise to design in this way. First, the engine and rotor are not subject to much in the way of direct human interaction during use, so there is no reason for its design to be governed by human cognitive biases instead of the laws of aerodynamics, the materials in use, and the atmosphere. (The flight controls and their mapping to the movement of the rotor, however, are a different story.) In addition, we acknowledge that in domains requiring extensive training, there may be different memory biases that manifest in experts and novices[5]. A possible solution is to run separate transmission chains that synthesize a different interface or mode for each group.

## Acknowledgments

This work was funded in part by National Science Foundation grant SPRF-IBSS-1408652 to T.L.G. and J.W.S. and a Berkeley Graduate Fellowship to M.D.P.

## References

Bartlett, F. C. (1932). Remembering: An experimental and social study. Cambridge: Cambridge University.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *4*(1), 55–81.

Cowles, M. K., & Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, *91*(434), 883–904.

Fitts, P. M., & Seeger, C. M. (1953). SR compatibility: spatial characteristics of stimulus and response codes. *Journal of Experimental Psychology*, *46*(3), 199–210.

Haucap, J. (2003). Telephone number allocation: A property rights approach. *European Journal of Law and Economics*, 15(2), 91–109.

Kalish, M. L., Griffiths, T. L., & Lewandowsky, S. (2007). Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin & Review*, 14(2), 288–294.

Kornblum, S., Hasbroucq, T., & Osman, A. (1990). Dimensional overlap: cognitive basis for stimulus–response compatibility — a model and taxonomy. *Psychological Review*, *97*(2), 253–270.

Lidwell, W., Holden, K., & Butler, J. (2010). *Universal principles of design, revised and updated*. Rockport Pub.

Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT Press.

Proctor, R. W., & Reeve, T. G. (Eds.). (1989). *Stimulus-response compatibility: An integrated perspective*. North–Holland.

Simon, H. A. (1996). *The sciences of the artificial*. MIT press.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, *89*(9), 1275-1296.

White, A. W. (2002). *The elements of graphic design: space, unity, page architecture, and type*. Skyhorse Publishing, Inc.

Wolf, G. (1998, September 1). The World According to Woz. Wired Magazine. Retrieved February 1, 2016, from http://www.wired.com/1998/09/woz/

[5] See, e.g., Chase and Simon (1973) regarding chess expertise.