Aalto University publication series
DOCTORAL THESES 219/2023

Human-in-the-Loop Design Optimization

Yi-Chi Liao

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall T2 of the school on 15 December 2023 at 14.

Aalto University
School of Electrical Engineering
Information and Communications Engineering
Computational Behavior Lab

Supervising professor
Professor Antti Oulasvirta, Aalto University, Finland

Thesis advisor
Professor Antti Oulasvirta, Aalto University, Finland

Preliminary examiners
Dr. Yuki Koyama, National Institute of Advanced Industrial Science and Technology, Japan.
Assistant Professor Seongkook Heo, University of Virginia, U.S.A.

Opponent
Associate Professor Pedro Lopes, University of Chicago, U.S.A.

Aalto University publication series
DOCTORAL THESES 219/2023

ISBN 978-952-64-1579-6 (printed)
ISBN 978-952-64-1580-2 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-1580-2

Unigrafia Oy
Helsinki 2023

Finland

Printed matter
4041-0619

Author
Yi-Chi Liao

Name of the doctoral thesis
Human-in-the-Loop Design Optimization

Publisher School of Electrical Engineering

Unit Information and Communications Engineering

Series Aalto University publication series DOCTORAL THESES 219/2023

Field of research Interactive Systems

Manuscript submitted 4 May 2023 **Date of the defence** 15 December 2023

Permission for public defence granted (date) 29 June 2023 **Language** English

☐ **Monograph** ☒ **Article thesis** ☐ **Essay thesis**

Abstract

This dissertation presents novel computational methods and investigations to enable human-in-the-loop optimization (HILO) for a wider range of realistic applications, allowing designers to efficiently explore the design space of practical problems. Designing effective interaction techniques requires careful consideration of various parameters, that significantly impact user experience and performance. However, optimizing these parameters can be challenging due to the large, multi-dimensional design space, the unclear relationship between parameter settings and user performance, and the complexity of balancing multiple design objectives.

Traditionally, designers perform manual optimization via iterative design processes, which can be time-consuming and effortful, and do not guarantee the best outcome. HILO emerged as a more principled solution for design optimization, using a computational optimizer to intelligently select the next design instance for user testing. Despite some examples of HILO in the human-computer interaction (HCI) field, its application scope is limited to a single objective, for a single user, and for graphical user interfaces. How to extend HILO for multi-objective problems, optimizing for a population, and supporting physical interfaces has remained unclear. Furthermore, conducting HILO does not eliminate the costs arising from human involvement, and practitioners have been reluctant to embrace a technique whose positive and negative qualities are not fully understood.

This dissertation presents a set of computational methods and investigations that address these challenges. Pareto-frontier learning is utilized to handle multi-objective design tasks, and I introduce novel extensions for practical solutions of group-level Bayesian optimization. To reduce the effort and time in prototyping, I propose using physical emulation to render physical design instances, enabling HILO to be applied to the design of physical interactions. The dissertation presents user experiments and a design workshop conducted to enrich the understanding of Bayesian optimization-supported design processes' strengths and limitations. Finally, in light of the resource-intensive nature of user studies, a simulation-based optimization framework is proposed whereby artificial users evaluate design instances.

With the ultimate goal of expanding HILO's utility in realistic and general design tasks, this dissertation opens new directions for future HILO research. One important path for exploration involves more advanced optimization techniques, such as methods that enable greater efficiency and support a high-dimensional design space. The project also spotlights the value of investigating better human-machine collaboration mechanisms in design optimization such that the designers can steer the optimization as required or fine-tune the suggestions proposed by the optimizer. Lastly, simulation-based optimization methods require further validation, and developing human-like models will be a crucial next step.

Keywords Human-in-the-loop optimization, Bayesian optimization, human-computer interaction, computational interaction, machine learning

ISBN (printed) 978-952-64-1579-6 **ISBN (pdf)** 978-952-64-1580-2

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2023

Pages 235 **urn** http://urn.fi/URN:ISBN:978-952-64-1580-2

Preface

Right before embarking on my Ph.D. journey in Finland, I attended ACM CHI 2018 in Montreal, followed by a brief stay in Toronto before catching the connecting flight to Finland. What I have never shared with my friends is that, during this layover, I was completely overwhelmed by the fear of the uncertainty that lay ahead of me. I had no idea what to expect to live in this distant country, Finland, or what the Ph.D. years would hold. As I sat at the gate in Toronto Airport, I even had the sudden idea of returning to Taiwan and abandoning my pursuit of a Ph.D.

Looking back, I'm so grateful that I summoned just enough courage to board that flight and come to Finland. It turned out to be an incredible journey, one that I would not trade for anything. What made it truly amazing were the exceptional people who supported me, guided me, and shared their lives with me. I want to take this opportunity to express my deepest gratitude to these extraordinary friends and colleagues.

First and foremost, I would like to extend my special thanks to my supervisor, Prof. Antti Oulasvirta, for his guidance and support over the entire course of my doctoral studies. Antti provided me with the space and encouragement to explore my academic passions. He taught me the art of thinking and the fun of tackling challenging and complex research problems. It is no exaggeration to say that without his supervision and guidance, I would not have become the person I am today. Beyond research projects, Antti is an excellent friend, an exceptional group leader, a great thinker, and an ambitious mind. He will always be a role model for me in my pursuit as a researcher or, perhaps someday, as a professor.

I would like to thank my pre-examiners, Dr. Yuki Koyama and Prof. Seongkook Heo. I feel truly privileged to have received their insightful and constructive feedback on my dissertation, which significantly helped me in enhancing the overall quality of the work.

I am also greatly honored to have Prof. Pedro Lopes as my opponent. Prof. Lopes might not be aware of this: when I initially started my Ph.D. journey, I had the privilege of attending his Ph.D. defense at UIST 2018.

That experience was truly inspiring and motivating. Now, 5 years later, having Prof. Lopes as my opponent in this final stage of my studies brings a beautiful closure to my Ph.D. journey.

I genuinely thank every member of the User Interfaces Group / Computational Behavior Lab for their friendships and support throughout the years. I want to thank my dear friend, Kashyap, who has supported me in every aspect – academically, emotionally, and far beyond. Without him, I would not have had the chance to do an internship at Meta, become involved in the CHI organization, or enjoy countless other wonderful moments. I must thank Sunjun, who not only endured my constant pestering but also guided me step-by-step on the path to becoming a researcher. I can not thank him more for his patience, kindness, and his unlimited knowledge of buttons and HCI. I would like to thank Aleksi and Markku, who welcomed me with a warm embrace when I first arrived in Finland, making me feel truly at home. My wholehearted gratitude goes to Aurélien, who I am proud to consider almost like an alternative family to me. I will never forget the game nights that we shared, and I am grateful for having you during my challenging time. I want to thank Luis for not only always giving me great advice but also for the joyful moments we played ping pong together.

Further, I would like to express my deep gratitude to Aini, Lena, Joongi, Danqing, Suyog, and Yue with whom I shared the office for an extended period. You hold an irreplaceable place in my memories of my time in the group. I couldn't have asked for better companions. I also need to thank the former Ph.D. colleagues, Anna, Janin, and Morteza, for being outstanding examples for me to follow. I am thankful for having the opportunity to be a colleague of many talented minds: Aida, Jussi, Niraj, Carlos, Michael, Camille, Thomas (van Gemert), Thomas (Grabot), Ai, Yunfei, Christoph, Kristian, etc. I will always cherish the time shared with you.

I could not accomplish any of the publications without my fantastic co-authors. I am fortunate to have had the opportunity to collaborate with Dr. John Dudley on many papers. John is a warm and incredibly dependable person, and his tireless guidance and proofreading of my text have always served as a beacon of light that illuminated my academic journey. I am privileged to have collaborated with Prof. Per Ola Kristensson, Prof. Andrew Howes, and Prof. Liwei Chan. I am glad to receive their wisdom not just in research projects but also in making career decisions. I need to thank Prof. Byungjoo Lee, who guided me in the button project and stands as a role model for me to follow.

I heartily thank Dr. Aditya Acharya and Antti Keurulainen for their persistent support in the affordance project (especially, during the intense deadline week), and George and Chun-Lien, who stood by me through the peaks and valleys in the Bayesian optimization papers. I sincerely thank

Hee-Seung for involving me in the target inference project, being a great friend, and introducing me to the world of Nintendo Switch. I would like to thank Aleksi Ikkala for generously sharing knowledge on biomechanical models with me, and I sincerely look forward to future collaboration.

I had a fantastic internship at Meta Reality Labs, which boosted my growth as a researcher. I would like to thank Dr. Aakar Gupta for giving me this opportunity and for providing support in every detail. I want to thank Dr. Ruta Desai for her strong guidance in the technical aspects of the project. Ruta always asked the right (i.e., most difficult) questions, which I learned so much from. Big thanks to Pierce for aiding with implementations, and Krista for conducting the studies. My days in Seattle and Redmond were filled with memorable moments, largely thanks to the wonderful people, e.g., Sebastian, Naveen, Joao, David, Rishi, Matthias, Tanya, Ting, and many others. These memories are treasured forever.

I must thank all the incredible staff at Aalto University for their assistance and support throughout my Ph.D. studies. Their warmth and efficiency never ceased to amaze me. Special appreciation goes to the E-support team, the HR team, Essi, Sanna, and the Doctoral Studies team.

I want to thank my parents, who encouraged me to be curious and unique when I was a kid, which ultimately led me here. I am grateful for their trust and support in me in making every decision. My deepest thanks go to my grandma, whose kindness and patience have influenced and benefited my whole life. Being the first in my family to venture abroad, I extend my utmost thanks to every member of my family. Knowing that I have a loving family in Taiwan, a place I can always return to, provides me with a sense of security that allows me to continue exploring the world.

Last but certainly not least, I want to thank my amazing partner, Chieh-Ling. Our journey from Taiwan to Finland and beyond has been filled with both challenges and happy memories. Thank you for always being there for me, listening to me, uplifting me when I am down, helping me to find myself when I am lost, and taking care of me in every aspect of my life. Your unconditional love is the strongest driving force behind my exploration of the universe. I can not wait to create more memories with you in the next chapter of our life together.

Helsinki, November 21, 2023,

Yi-Chi Liao

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Yi-Chi Liao, John J Dudley, George B Mo, Chun-Lien Cheng, Liwei Chan, Antti Oulasvirta, Per Ola Kristensson. Interaction Design With Multi-objective Bayesian Optimization. *IEEE Pervasive Computing*, 22, 1, 29-38, January 2023.

II Liwei Chan, Yi-Chi Liao, George B Mo, John J Dudley, Chun-Lien Cheng, Per Ola Kristensson, Antti Oulasvirta. Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques. In *2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, USA, April 2022.

III Yi-Chi Liao, George B Mo, John J Dudley, Chun-Lien Cheng, Liwei Chan, Per Ola Kristensson, Antti Oulasvirta. Practical Approaches to Group-Level Multi-Objective Bayesian Optimization in Interaction Technique Design. Submitted to *ACM Collective Intelligence*, 14, March 2023.

IV Yi-Chi Liao, Sunjun Kim, Byungjoo Lee, Antti Oulasvirta. Button Simulation and Design via FDVV Models. In *2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, April 2020.

V Yi-Chi Liao, Kashyap Todi, Aditya Acharya, Antti Keurulainen, Andrew Howes, Antti Oulasvirta. Rediscovering Affordance: A Reinforcement Learning Perspective. In *2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, USA, April 2022.

Author's Contribution

Publication I: "Interaction Design With Multi-objective Bayesian Optimization"

This paper investigated supporting design exploration with multi-objective Bayesian optimization. I planned and conducted a design workshop and analyzed the results, which formed the basis of this paper. I also developed the optimization with George B. Mo, and I implemented a smartwatch prototype for the workshop. The research ideation process was a team effort involving all co-authors. The writing process was also a collaborative effort, with Dr. John J. Dudley leading the final submission process. Prof. Per Ola Kristensson and Prof. Antti Oulasvirta provided support in the ideation and writing throughout.

Publication II: "Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques"

This research project aimed to investigate the effectiveness of human-in-the-loop Bayesian optimization in comparison to the designer-led approach, under the leadership of Prof. Liwei Chan. I participated in the research ideation phase alongside other co-authors and designed the procedure of the user study. The user study was primarily conducted by Chun-Lien Cheng under Prof. Chan's guidance. Additionally, I implemented Bayesian optimization with George B. Mo and the Unity application with Chun-Lien Cheng. My role in implementation was primarily bridging the optimization algorithm and target application. My contribution to the writing is mainly in Section 4, and Figures 6 and 7. I was also involved in writing and revising other sections throughout the submission process.

Publication III: "Practical Approaches to Group-Level Multi-Objective Bayesian Optimization in Interaction Technique Design"

In this paper, we proposed two practical approaches for achieving group-level multi-objective Bayesian optimization. I led the project ideation and application development, working closely with my co-authors. George B. Mo implemented the Pareto-frontier learning algorithm, which I adapted for various applications. Additionally, Chun-Lien Cheng developed the Unity program for a specific target application. I was involved in planning and conducting two user studies with the support of Prof. Liwei Chan and Chun-Lien Cheng. Throughout the project, all co-authors contributed to the ideation and writing process. Dr. John J. Dudley provided particularly valuable support during the writing and submission of the paper.

Publication IV: "Button Simulation and Design via FDVV Models"

The idea of simulating the tactile characteristics of push-buttons was initially proposed by Prof. Antti Oulasvirta and Prof. Sunjun Kim. In my first attempt to achieve button simulation (published in UIST '18 Adjunct), I followed the traditional Force-Display model approach, but it turned out to be unrealistic for certain types of push-buttons. Therefore, I proposed a more sophisticated FDVV model for capturing button-pressing together with Prof. Sunjun Kim. I further developed an end-to-end button simulation pipeline that starts from profiling a button to rendering its characteristics on a physical simulator. Throughout the development process, Prof. Sunjun Kim provided valuable support in prototyping and implementation. Prof. Byungjoo Lee provided close guidance on the ideation, model development, and formulation of the temporal-pointing application, contributing to the project's overall success. Prof. Antti Oulasvirta was also closely involved in the overall development of the project. The paper was written collaboratively with all co-authors.

Publication V: "Rediscovering Affordance: A Reinforcement Learning Perspective"

I proposed a new theory and a novel model for affordance formulation and perception. I designed and conducted two user studies, and ran a simulation experiment that employed a reinforcement-learning agent. All the authors jointly consolidated the research idea, and I played a key role in proposing the novel theory and model implementation. Prof. Antti Oulasvirta and Prof. Andrew Howes provided valuable perspectives from Cognitive Science and Psychology backgrounds, which is valuable in

theory formulation. Dr. Kashyap Todi and Dr. Aditya Acharya provided their views from design and machine-learning perspectives, which also contributed to the theory. Antti Keurulainen and Dr. Aditya Acharya helped me with the model implementation. Dr. Kashyap Todi provided valuable suggestions for the user study plan. The writing was mainly done in collaboration with Dr. Kashyap Todi and Dr. Aditya Acharya, and all the co-authors were involved in the writing process.

List of Figures

List of Tables

Abbreviations

AF Acquisition function

AR Augmented reality

AutoML Automated machine learning

BO Bayesian optimization

CQ Chapter-level research question

EHVI Expected Hypervolume Improvement

EI Expected Improvement

FD Force-displacement model

FDVV Force-displacement-velocity-vibration model

GP Gaussian process

HCI Human-computer interaction

HILO Human-in-the-loop optimization

IT Information transfer rate

MOBO Multi-objective Bayesian optimization

PI Predicted Improvement

RL Reinforcement learning

RQ Dissertation-level research question

SM Surrogate model

UCB Upper Confidence Bound

UCD User-centered design

UI User interface

VR Virtual reality

Symbols

f a target function to be optimized; in the context of human-in-the-loop optimization, an interaction can be seen as a function that maps a design candidate x to objective function value(s) y, such that $f(x) = y$.

$\mathbb{R}$ real number.

X a design space, which is composed of n design parameters ($X \in \mathbb{R}^n$).

x a specific design candidate or parameter setting; such a design candidate is within the design space X ($x \in X$).

Y an objective-function space, which is composed of m objective functions ($Y \in \mathbb{R}^m$).

y one value or a set of values of the objective function(s); such objective-function value(s) is within the objective-function space Y ($y \in Y$).

$\emptyset$ empty set.

1. Introduction

"Don't ever make the mistake [of thinking] that you can design something better than what you get from ruthless massively parallel trial-and-error with a feedback cycle." — Linus Torvalds

"True optimization is the revolutionary contribution of modern research to decision processes." — George Dantzig

An interaction technique is a system comprising hardware and software elements that enable users to complete tasks [66]. Design parameters define the characteristics of an interaction technique. For instance, among a touchscreen button's design parameters might be its length and width, its color, and the force required to trigger it. Since the values assigned to those parameters have a significant impact on the usability and user experience [2, 86], appropriate exploration of their various possible values is critical in the design process for any given interaction technique [186]. However, exploring the vast design space containing all possible parameter settings is a challenging task. Traditionally, designers perform iterative manual optimization, exemplified by the user-centered design (UCD) workflow [1], which involves the designer proposing a design candidate and developing a prototype. The design that emerges undergoes user testing [142], with the results guiding the designer's iterative tuning of the idea until an acceptable design is reached. While manual optimization in iterative user-centered design is a viable approach for exploring the design space, it does not guarantee an optimal result. The central challenge stems from the large design space that most design tasks bring. For example, even a single physical button, or "push-button," requires a design parameter specifying the travel range, a separate one for the point at which a press gets registered, and at least two defining haptic feedback [56]. Because all these parameters are continuous, the number of possible design combinations is infinite. Even restricting each parameter to 10 discrete levels still entails $10,000$ possible combinations. Hence, exhaustively exploring

all potentially good design candidates proves nearly impossible.

Wrestling with such a large design space is not the only challenge of the manual optimization process. It also takes time and consumes other resources. In our example case of designing a push-button, the interaction designer must develop a proposed design, making assumptions about the expected result along the way, after which a developer or engineer has to create a corresponding prototype [228]. These efforts are followed by those of a user researcher, who conducts a user study and analyzes the data. Finally, the designers have to interpret the results together and propose a new design. The significant time and effort demanded by this multi-phase process renders exploration even more effortful [1]. Moreover, manual design is susceptible to bias and fixation issues, since it relies heavily on the designer's experience and prior knowledge. With multiple, possibly conflicting design objectives, carefully balancing the objectives and predicting the effects of each alteration to the design while exploring the vast design space demands immense mental effort. In consequence, designers often settle for designs that "feel right" instead of pursuing a genuinely optimal solution.

Human-in-the-loop optimization (HILO) offers an alternative: rather than relying on designers to choose the next design for evaluation, the process uses the computational optimizers to intelligently select the design instances while *human subjects* evaluate the design instances by interacting with them. Please refer to Figure 1.4 for an illustration of the concept of HILO. While HILO has demonstrated potential to enter the picture as a more principled optimization procedure [54], it has been employed for a few specific applications only [111]. One significant limit in scope stems from today's confinement of attention to single-objective problems, whereas realistic design challenges often involve tradeoffs among several objectives. For instance, designing any input devices often needs to consider both the efficiency and accuracy of the interaction. Multi-objective optimization is required to fully address this challenge. Another issue is that current HILO methods optimize designs for a single user, while practical design work often must optimize at the population level, for various user groups [165]. Without a population-level optimization, optimization needs to be deployed on every single user, leading to low efficiency. Additionally, HILO thus far has been applied only for user interfaces whose design process does not include physical prototyping, with the main reason being the time-consuming nature of such prototyping [15]. A more general factor holding back progress is that the benefits and drawbacks of applying optimization in the design process have gone unexplored. Accordingly, designers are less strongly motivated to make wider use of HILO procedures. Lastly, the costs remain considerable: HILO still requires user interaction in each iteration. Yet, recruiting user participants and conducting user studies is a costly step, which limits the use of HILO in design practice. It is also worth

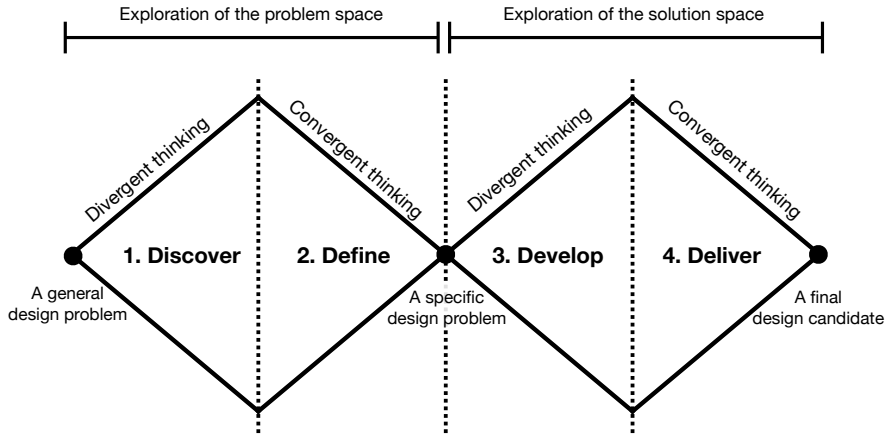


Figure 1.1. Design thinking's double-diamond model for the design process.

exploring the possibility of utilizing simulated human users to boost the efficiency of HILO. It is clear that the aforementioned challenges hamper applying HILO to realistic design problems today.

The primary goal behind this dissertation is to expand the applicability of HILO for design interactions. I developed a set of novel computational methods to this end. Furthermore, I conducted investigations to evaluate these extensions' performance and effectiveness, thus contributing to a comprehensive understanding of the strengths and limitations of HILO. Ultimately, these advances should enable HILO to address a wider range of realistic design problems. With this chapter, I begin my presentation of them by outlining the challenges inherent to design-optimization problems and introducing human-in-the-loop optimization. My research questions then can be situated against this backdrop. Finally, I describe the research objectives, methods, and contributions.

1.1 The Design-Optimization Problem

Practitioners employ various design processes and models; among them, the double-diamond model [162, 13] (see Figure 1.1) applied in "design thinking" is an especially well-established framework describing the general design process. It extends all the way from the initial broad design goal to a final design. My focus is on the latter part of the process, the model's last two steps: developing and exploring possible design solutions and, then, narrowing the set to deliver a final design. Specifically, the dissertation addresses only the **parametric optimization** problem [82], where the design challenge is to determine the optimal set of parameter values for a given interaction. Figure 1.2 illustrates this problem: the designer considers a specific **design space**, within which each design

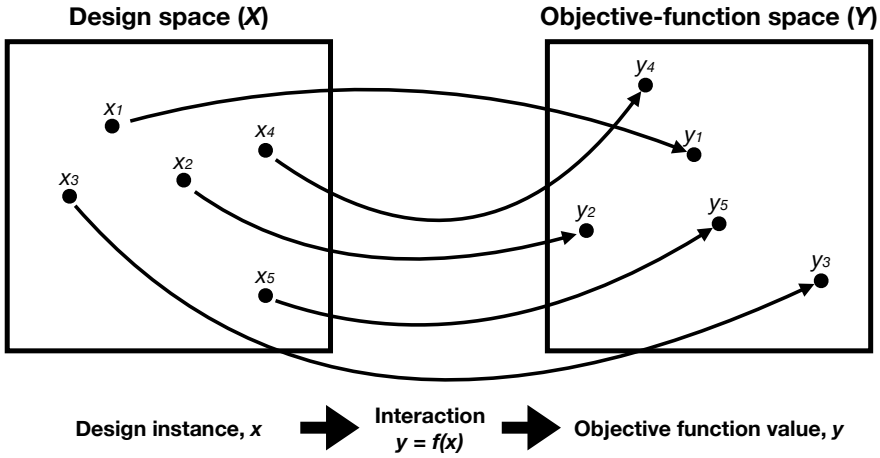


Figure 1.2. The design-optimization problem: within the design space (X), there are many design candidates (x), each constituting one "parameter setting." Particular designs/settings lead to particular user performances, or objective-function values (y), which lie in the objective-function space (Y).

candidate is a unique parameter configuration. Varying the parameter settings produces different objective-function values, which represent user performance; together, all the possible user performances constitute the **objective-function space** [173]. The designer's goal is to identify the best design – i.e., a combination of parameter values that leads to optimal performance. My discussion below refers to said parametric optimization problem as "design optimization" and formulates the research problem on this basis.

1.1.1 Design Parameters and Design Space

The design optimization process revolves around the design space, which contains all the valid design candidates. In this dissertation, the design space is denoted as X. Each design parameter within the space represents a variable to which the designer can assign a value. For example, one of the design parameters in Web-page design could be the font size of the header, with the designer assigning it a certain value. Design tasks usually involve multiple design parameters, and the design space is multi-dimensional. We can mathematically formulate the design space as $X \in \mathbb{R}^n$, where X has n real-number design parameters. For instance, the length, width, and height parameters in our push-button design example might range from 1 cm to 3 cm. Therefore, we can formally describe the design space as $X = [1cm, 3cm]^3$.

A design instance, denoted as x, is a possible set of values within the design space X. For instance, a design instance of the push-button could have a width of $1cm$, a length of $2cm$, and a height of $1.5cm$. This dissertation uses the terms "design instance" and "parameter setting" interchangeably

to refer to a possible design x within the design space X. We can also denote the latter relationship as $x \in X$.

1.1.2 Objective Functions

In design optimization, the objective function represents the performance metric for which the designer aims to optimize. Its value may be minimized or maximized. This metric can reflect either some measurable performance (completion time, success rate, etc.) or a subjective rating (e.g., Likert-scale comfort level). For any interaction, the designer might aim to optimize for a single objective function or several. For instance, in the case of an input device, there might be objective functions for accuracy and efficiency both. Mathematically, we can represent one or several objective function(s) as $y \in \mathbb{R}^m$, where m is the number of objective functions, and all the measured objective value(s) are within a continuous range. Each y contains a set of objective values of an interaction.

All the possible objective function values jointly form a multidimensional objective function space Y, which one may formally represent as $y \in Y$.

1.1.3 Interactions as Black-Box Functions

In design optimization, each design instance x leads to an objective-function value y, which could be measurable or more subjective. To represent the relationship between an interaction's input (x) and its output (y), we can use a function (f) that maps the design instance to the objective function's value. Since we seldom possess prior information about the function, we consider it a black-box function [3]. As depicted in Figure 1.2, we can mathematically express the interaction as $y = f(x)$, where f (the interaction) takes x (a design instance) as input and generates a response y (a value of the objective function(s)).

In the example of a button-pressing interaction, changing the size of the button keycap (x) can affect such aspects of user performance as typing accuracy (y). Although we may have some prior assumptions about the relationship, we generally lack detailed knowledge that would allow us to predict the resulting objective function. Moreover, recall that an interaction can have multiple objective functions. For example, good Web design can improve not only search efficiency (y_1) but also visual comfort (y_2), among other elements. Therefore, solving the design-optimization problem presents several challenges, expanded upon below.

1.1.4 Challenges of Design Optimization

Optimizing design is the process of identifying the optimal design instance(s) x within design space X whereby the optimal objective function(s)

y is/are produced. Several factors make this a difficult task. I present a few of the more prominent ones below.

The Large Design Space
Design optimization often is complicated by the need to deal with large design spaces [104]. With every additional design parameter, the design space grows exponentially and it becomes more difficult to identify the best design instance. For example, when each parameter is discretized to 10 levels, a single-parameter design task involves 10 design candidates in total; in contrast, a task with four design parameters presents more than $10,000$ possible combinations to contend with. The limited time and monetary budget for manual optimization make an exhaustive search impractical; therefore, effective navigation of the large design space necessitates optimization algorithms.

Complex Interaction
The association between design parameters and the objective function is a crucial facet of design optimization. Understanding their interaction can be challenging, however, since it often manifests itself as a black box: one knows only the output value for a given input. This lack of information makes it difficult to optimize the design effectively. To obtain the pairs of output and input values needed for the accurate evaluation of the design, one must carry out user testing, and the manual search for optimal design instances can be mentally and physically demanding.

Tradeoffs and Multiplicity of Objectives
The next issue arises from the frequent need for design optimization to consider multiple objectives [173]. Because optimizing for one objective sometimes comes at another objective's cost, it can be difficult for human designers to make sense of and predict the best tradeoff. For example, improving efficiency may require sacrificing accuracy. Especially when there are more than two objectives, it may be very challenging to identify the optimal design, balancing all of them effectively.

Biases and Design Fixation
Lastly, alongside the challenges rooted in the problem itself, human designers face their own biases [53]. To select the design candidates to be evaluated, designers often rely on their experience and domain knowledge. That experience comes with potential blind spots such as overlooking some options and getting fixated on other, suboptimal designs [98].

In summary, the complexity of the problem, the large design space, and the difficulty of balancing several objective functions render it mentally demanding to seek optimal design instances manually. Since an exhaustive search is not practical in these conditions, designers have to pick the best

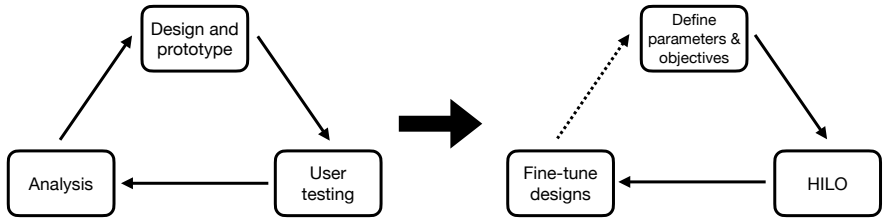


Figure 1.3. At left, a diagram of the traditional UCD workflow for design optimization. The right-hand pane illustrates a human-in-the-loop workflow. Note the dashed line in the HILO workflow, which indicates that adjusting design parameters and objective functions is an optional step. In contrast, UCD is an iterative process, and redesigning is necessary.

design from the small subset of design instances tested. In this process, the optimal one might well end up omitted.

1.2 Human-in-the-Loop Optimization

Human-in-the-loop optimization offers an alternative approach to tackling the challenge of design optimization. In contrast against reliance on human designers to pick each design for evaluation, HILO employs a computational optimizer to generate the next design for user testing [199, 28] (as illustrated in Figure 1.4). By ruling out human biases and fixation, HILO allows for a more principled and systematic search. Moreover, by eliminating the analysis step featured in the traditional UCD process, HILO yields greater efficiency and reduces the number of empirical studies and analyses needed (as illustrated in Figure 1.3).

In the HILO framework, the designers' main task is to set up the optimization, including specification of the design space and objective space, and conduct the HILO. Notwithstanding its various potential advantages, HILO is not widely applied by human–computer interaction (HCI) or design practitioners, mainly because of the many questions that still need to be addressed. Therefore, the main goal I set for this dissertation is to address those open questions, elaborated upon in subsection 1.2.2 ("Research Questions"), thus promoting the use of HILO in realistic design optimization by expanding the application scope. This project for addressing the challenges of design optimization enabled me to propose a series of novel methods based on Bayesian optimization. Next, I introduce Bayesian optimization, which has gained popularity in the HILO domain because of its efficiency and its support for extensive customization.

1.2.1 Bayesian Optimization

Bayesian optimization (BO) is characterized as a powerful optimization method for solving expensive black-box functions [68, 29] (again, problems

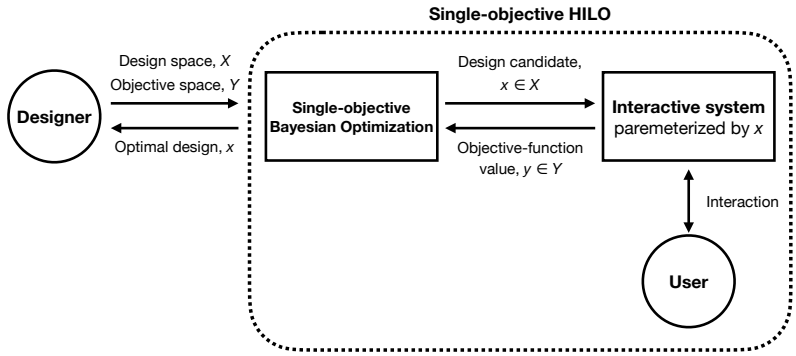


Figure 1.4. The state of the art of HILO using Bayesian optimization. It handles design problems with a single objective, and it optimizes for a single user at a time.

or systems of which we have no knowledge, where the only way to obtain information about what lies within the "box" is by evaluating them and obtaining one pair of input x and output y at a time). Because evaluating black-box functions can be expensive in more than one respect, a grid search over all possible inputs is impractical. As introduced earlier, the interaction between users and interactive systems can be seen as precisely such a function, and evaluating each design usually demands time and effort.

BO is not the only black-box optimization algorithm. While there are evolutionary algorithms [9], genetic algorithms [105], particle-swarm models [107], and others [3], BO exhibits consistently promising performance [24]. Additionally, its customizability represents a great strength that other algorithms do not offer. There are two elements at the core of the BO framework: Gaussian Process regression (GP) and the acquisition function (AF). The former has several hyperparameters, which users can set to achieve the best performance for their particular problems, and various "kernels" (discussed further on). Similarly, there is a wide range of possible AFs. Users can fine-tune each of these by setting hyperparameters. To adjust the BO to case-specific needs, users can even implement their own GP or AF. Thanks to these advantages, BO has become the most mainstream method of human-in-the-loop optimization and formed the core of the dissertation project. I briefly introduce key elements of BO below, with a more detailed description provided further along, in section 2.4 ("Bayesian Optimization").

Surrogate Model (SM)

The surrogate model (SM), a key component both of BO and of other model-based optimization methods, is constructed in line with the observations made as optimization progresses. It serves as a lower-cost alternative to the actual function [92]. In BO, the SM is a Gaussian Process regression (GP) that is fitted with all the observations thus far [222]. By means of

GP, we can predict the output (y) and the variance of the prediction, which represents the level of confidence in the predictions. Exploiting the evaluations of the SM, BO efficiently chooses the next input (x) to evaluate with the function proper. The ability to customize hyperparameters and select from among various kernels in a GP-based SM further enhances flexibility and performance. This has contributed to the technique's popularity as a mainstream method for conducting HILO. The surrogate model in BO plays a vital role in making the optimization process more efficient and effective.

Acquisition function (AF)
On the basis of the information provided by the SM, the acquisition function provides the acquisition value ($AF(x)$) of a particular input x [29] – i.e., the worth of evaluating the actual function with this input x. At the beginning of each iteration, BO samples many design candidates x and expresses queries for the acquisition values $AF(x)$. Then, the design instance x with the highest value is selected to be evaluated in this iteration. Various AFs have been proposed for this core element of BO: the popular upper confidence bound (UCB) function, expected improvement (EI), etc. These seek balance in the exploration–exploitation tradeoff and guide the search toward promising regions of the input space. In addition to these now-standard AFs, researchers have proposed many novel AFs to address specific challenges posed by diverse domains. Some AFs are designed to handle multi-objective optimization, while others are intended specifically to address noisy or dynamic objective functions. These novel AFs often incorporate domain-specific knowledge and heuristics to improve search efficiency and effectiveness.

General Procedure of BO
The general procedure of BO is described below via Algorithm 1, an adaptation from Frazier's introduction paper [68]. Given an expensive target function f and a budget (maximum iteration count) N, BO first initializes a GP instance as the SM. Then, BO randomly samples n_0 data points across the entire prospective design space. Within each optimization iteration, GP updates are performed on the basis of all the observations, the algorithm computes the acquisition values for many x, and it picks the x with the highest acquisition value for function evaluation. This continues until the iteration number reaches N. The x_i that has the optimal y_i function value is returned as the final output.

BO in Human-in-the-Loop Bayesian Optimization
The flexibility and effectiveness of BO have helped it gain popularity in the field of human-in-the-loop optimization. One of the earliest projects to utilize BO in HILO was by Brochu et al. [28], who used BO to identify the

Algorithm 1 General Procedure of Bayesian Optimization

Inputs:
  A target function f.
  The optimization budget N.
Outputs:
  The optimal parameter setting x
Initialize:
  Place a Gaussian Process as the surrogate model.
  Observe f at n_0 points. Set $n = n_0$.
while $n \leq N$ **do**
  Update GP using all observation data.
  Let x_n be a maximizer of the acquisition function over sampled x.
  Observe $y_n = f(x_n)$.
  Increment n.
end while
return x_i that has the optimal y_i function value.

optimal hyperparameter settings for computer rendering of realistic animation. They showed that, with the support of BO, users could efficiently identify a good design. Later, Khajah and colleagues [108] employed BO to maximize user engagement in gaming, while Yamamoto et al. implemented it for efficiently facilitating a photographer's setup process for lighting devices [227]. These projects attest to the effectiveness of BO in HILO and how it can be customized for various optimization problems. Further examples are considered in chapter 3 ("Related Work: HILO in HCI").

1.2.2 Research Questions

While BO-based HILO has demonstrated success in specific cases, certain fundamental limitations continue to restrict its application range. In aims of expanding the usability and application scope of HILO through doctoral research addressing these limitations, I articulated several research questions. The set of questions dealt with in this dissertation (RQs) is structured thus:

RQ 1. How to enable multi-objective human-in-the-loop optimization? Most HILO research has focused on single-objective optimization, but real-world design tasks often involve multiple objectives, and how to extend HILO for approaching such tasks has remained unclear.

RQ 2: What are the benefits and limitations of human-in-the-loop optimization? Although research has shown that Bayesian optimization can assist with the design process in HILO settings, its effectiveness level and designers' perceptions of it are less evident.

RQ 3. How can group-level human-in-the-loop optimization be achieved? All prior work has focused on optimizing for an individual user from scratch. However, design practice often aims at 1) optimizing for a group of users or 2) efficiently personalizing a user-specific design from the starting point of a default setting. Current BO methods are not able to support such aims.

RQ 4. How can we equip HILO for interactions that require physical prototyping? Current HILO methods are restricted to a set of user interfaces for which varying the design instance does not require different physical prototypes. Many physical interfaces (e.g., with a mouse, keyboard, or controller) remain excluded from HILO because prototypes must be fabricated for each iteration.

RQ 5. How could we further reduce the cost of conducting user experiments? Completing full-fledged HILO requires human participants serving as evaluators, which can entail high costs for both designers and these users. Since designers may not always have the budget for such HILO, it is important to find ways to reduce the cost of conducting user experiments.

1.3 Research Objectives and Methods

By means of five research publications, I offer solutions responding to all of these research questions. To address RQ 1, I propose multi-objective Bayesian optimization (MOBO) with Pareto-frontier learning [198] for HILO. Rather than a final design for a single performance metric, MOBO obtains a series of Pareto-optimal designs. This affords designers the flexibility to choose the final design amid the tradeoffs across all objective functions. The corresponding articles are **publications I and II**.

To respond to RQ 2, I conducted several lab-based controlled studies and a workshop to investigate the positive and negative qualities of HILO. These investigations provided insight into the benefits of HILO such as improved performance and shortcomings such as the designer's sense of losing agency and ownership of the design process. **Publications I and II** describe this work too.

For answering RQ 3, I propose the novel concept of group-level HILO. This entails aggregating optimization data across a group of users to arrive at designs optimized for groups or a fast-adaptation model for rapid design customization. The concept is dealt with in **Publication III**.

My answer to RQ 4 involves a proposal to replace physical prototyping with physical emulation. In contrast to the former, a financially prohibitive and time-consuming process, emulation employs a hardware or software system that resembles the various relevant systems in its behavior. Build-

ing an emulating system that can render multiple designs of a physical interface instantly should save time and eliminate the cost of constructing real-world prototypes, thereby enabling HILO for physical interface design. The corresponding paper is **Publication IV**.

Finally, to address RQ 5, I propose a novel simulation-based optimization framework. In this framework, which uses models to simulate users' behaviors and also the human–interface interaction, a reinforcement-learning-based agent learns to interact with a particular interface. The optimizer generates various design instances of the interface, whereby the framework can derive the optimal design in the simulation environment. This contribution is presented in **Publication V**.

1.3.1 Method #1: Optimization

In pursuit of the overarching goal behind the dissertation – to expand the application scope of HILO – I developed novel optimization methods or adapted existing ones to HILO applications. For the response to RQ 1, I worked with established multi-objective Bayesian optimization with Pareto-front learning, implementing the technique for HCI design tasks. The method's applicability for HILO had gone unexplored, and my work filled the gap by demonstrating its effectiveness for handling multiple design objectives.

To address RQ 2, in turn, I explored two extensions based on multi-objective BO for group-level optimization in the context of HILO. The first extension proposed, Global GP, is a unified large model that is constructed to operate from all observations across all of the users. This can be used to derive the group-level optimized design instance(s). The second extension, Warm-Start GP, is a variant of the sparse Gaussian Process method that selects the most representative data points to inform a lightweight model, which can serve as the prior for future optimization. Both extensions improve the scalability and efficiency of HILO for group-level optimization tasks.

1.3.2 Method #2: Emulation, Prototyping, and Modeling

Another key facet of my project is the use of emulation, prototyping, and modeling to enable applying HILO for physical interfaces (under RQ 4). Specifically, for the task of button-pressing, I developed a novel interaction model that captures the nuances of this action. This work was followed by examining a series of prototyping methods (including 3D printing, laser cutting, soldering, and circuit assembly) to employ for the construction of the emulator. Finally, to emulate the button-pressing interaction well, I pioneered an emulation workflow that encompasses data-gathering, signal-processing, and control.

1.3.3 Method #3: Simulation, User Modeling, and Theory

My work addressing RQ 5 involved a simulation framework that employs the MuJoCo engine for a set of optimization applications managed by means of physical simulation. To address the need for a user model to inform the simulation of human behavior within the framework, I utilized policy-based reinforcement-learning methods to construct a user model capable of interacting with various widgets and interfaces. To enable simulated users to learn the correct actions for the various objects, I introduce the final component of this method, a novel theory proposed for explaining the process of forming affordance perceptions. This theory helps shed light on how humans perceive and understand objects' functionality and holds promise for aiding in the creation of more realistic user models in simulation frameworks.

1.3.4 Method #4: User Research

Evaluating the proposed methods' effectiveness took advantage of several user-research methods. Specifically, a series of user studies investigated whether HILO outperforms traditional design approaches. For instance, an empirical study aligned with RQ 2 compared user performance between outputs from HILO and from traditional design approaches. Likewise, a comparison responding to RQ 4 examined the optimized button design vs. several preexisting button designs. In addition, for RQ 3, group-level optimization was compared with state-of-the-art HILO. Finally, I conducted a design workshop to gather subjective feedback from designers (eight designers were assigned two subjects for the completion of a design task). These studies were critical in assessing the proposed methods' effectiveness and guiding further improvements.

1.4 Contributions

The work's central contribution is the development of a set of novel methods and investigations that, by together enabling HILO to explore parameter design spaces across a wider range of realistic applications, enhance its utility for assisting with practical design problems. This dissertation focuses more specifically on developing methods based on Bayesian optimization. New methods and findings connected with design optimization constitute the core of my advances; however, the outcomes also enrich the HCI landscape generally. Several contributions are worthy of specific mention.

Contribution 1 – multi-objective human-in-the-loop design optimization via Pareto-frontier learning (see publications I and II):

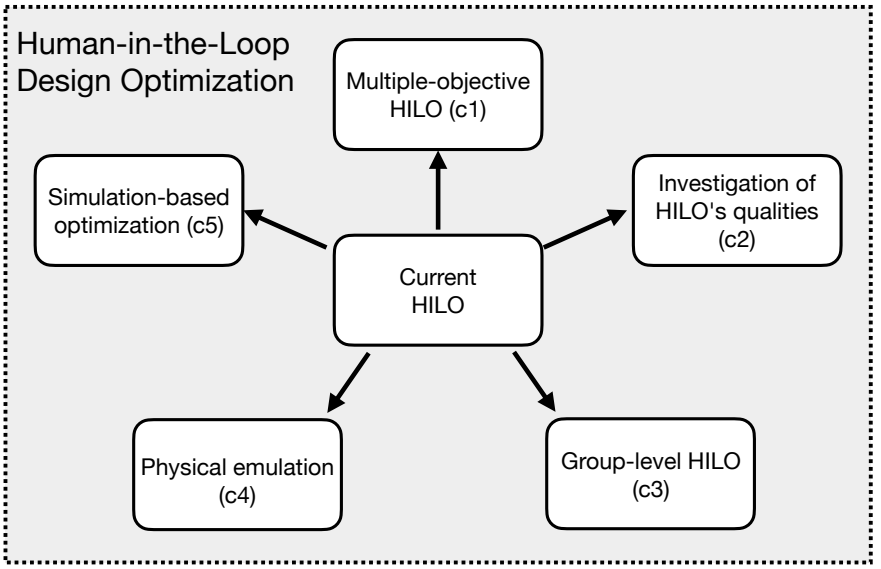


Figure 1.5. The contributions of this dissertation.

While previous work on HILO has focused on a single objective, real-world design challenges usually involve multiple objectives. Therefore, I proposed using Pareto-frontier learning to achieve multi-objective HILO. The evaluation showed that designers using multi-objective HILO engaged in a more variety-rich exploration of the options and ended up with designs that led to better user performance than those produced by traditional approaches.

Contribution 2 – investigating the qualities of multi-objective human-in-the-loop optimization (see publications I and II): Via a series of investigations, I showed that multi-objective HILO reduces the design process's total effort investment and offers a designer the flexibility of trading off between various objectives. However, designers felt less agency and ownership in these conditions relative to the typical, designer-led process.

Contribution 3 – group-level human-in-the-loop optimization (see Publication III): Standard Bayesian optimization devotes lengthy iteration cycles to reaching an optimal design for a single user. However, real-world design tasks usually are intended to produce designs for a user group, not single individuals. The dissertation presents methods that I propose, accordingly, for group-level HILO. The project's research attests that these methods support a) deriving a group-level optimized design and b) implementing a rapidly adapting warm-start setting.

Contribution 4 – human-in-the-loop optimization with emulation (see Publication IV): Heretofore limited to interaction involving a static physical/mechanical form, HILO has not been applicable when fabrication

at each iteration is needed. I have shown that physical emulation mitigates the need for fabrication, thereby enabling HILO's implementation for such interactions. An example use case is push-button design supported by a button-emulation pipeline.

Contribution 5 – a simulation-based human-in-the-loop optimization framework (see Publication V): Finally, I developed a novel framework for moving the entirety of human-in-the-loop design optimization into the simulation domain. The framework enables design optimization without human participants; it utilizes agents as proxy users to interact with the design instances. A preliminary study demonstrated that agent-in-the-loop optimization generates reasonable design results without any users' involvement.

1.5 The structure of the Dissertation

The dissertation's synthesis portion is organized into eight chapters. In Chapter 2, I provide a review of the fundamental background to the work carried out, especially typical design processes and human-in-the-loop optimization. Chapter 3 covers optimization methods, Bayesian optimization, and its applications in HCI work, and (by way of a brief overview) emulation and simulation as used in that field. After that, I turn my attention to the research related to multi-objective optimization, and I investigate its effectiveness. Then, in Chapter 5, I address the challenge of group-level optimization and detail the two novel extensions developed. Chapter 6 details my proposed technique of emulation as a mechanism for overcoming the difficulties involved in the optimization of physical widgets. Continuing the advances in this direction, the seventh chapter introduces simulation-based optimization as a tool that goes further – for optimizing interaction techniques without imposing any need for physical interactions. Finally, Chapter 8 summarizes the key findings from the dissertation project and discusses the opportunities for building upon this work.

2. Background

The dissertation project's contributions have their roots in fundamental work in several fields. This chapter presents a comprehensive review of that background, beginning with an overview of **design processes**, which are crucial in generating ideas and designs. These processes entail designers combing through various design options to identify the optimal solution, typically arrived at via **empirical research methods**. Conventional techniques' manual design optimization exhibits limitations, which highlight the need for automatic and principled optimization methods. Another closely connected discipline, alongside design, is **engineering design optimization**. It employs computational optimizers to aid in decision-making as the course of the optimization procedure progresses. Though the problems tackled in engineering optimization are similar to those in the design field, the latter's involvement of human users renders the cases examined in this dissertation more complex. That issue prompted me to review recent advancements in **human-in-the-loop optimization**, defined as a framework in which an optimizer generates design instances while user participants evaluate them. Because **Bayesian optimization**, a technique in widespread use for HILO, is at the heart of my project, I also provide an in-depth review of key BO concepts and terminology.

2.1 Design Processes and Empirical Research

Designing an interaction is a complex and multifaceted task that requires careful consideration and planning [80, 39]. For a better understanding of the various approaches and methods followed in design, researchers have devoted extensive effort to categorizing and studying the field's many processes. Below, I review two design processes that are particularly relevant to the dissertation's discussion: the double-diamond model of design thinking and user-centered design.

2.1.1 Design Thinking's Double-Diamond Model

In the 1960s, psychologists attempted to summarize humans' procedures for applying creativity to solve complex problems [169, 81]. The 1970s saw design researchers begin documenting how designers solve problems in a more systematic way [48, 5]. That was the era in which the design field recognized the existence of "wicked problems" [190, 30], problems that are ill-defined and have no straightforward solutions. Design thinking emerged in response to the need for solving them nonetheless; the term refers to a set of cognitive, analytical, and practical procedures applied to tackle complex and open-ended design problems [186, 149, 50]. The purpose of design thinking is to obtain a deep understanding of users and problems. Into the 1990s, IDEO continued work on its own version of design-thinking processes and methods [33, 100], which laid the groundwork for the double-diamond model popular today (see Figure 1.1). Among its key concepts are "problem space," a term originally coined by Newell et al. [162], and the notion of the design process's "divergent" phase and "convergent" phases [13]. The double-diamond process itself [47], which ultimately became the most well-known framework for design thinking [178], was fruit of further refinements by the British Design Council.

This process comprises four phases. It begins with the "discover" phase, wherein designers gather information and gain in-depth understanding of the problem and user needs. For comprehensive discovery of the problem, the designers spend time interacting with the actual users. Next, in the model's "define" phase, the designers apply their insight from the previous phase to refine and concretize the problem, moving from exploration of the problem space to focusing on a specific problem. After exploring the problem space in these two phases, the designers enter the "develop" phase, oriented toward seeking diverse solutions and inspiration sources that may answer the question specified. Many possible design candidates get proposed in this stage. Finally, with the "deliver" phase, the designers refine and narrow the set of solutions to arrive at a final design.

Since the last two stages are related to finding the solution, researchers often regard them as exploration of the "solution space." While all four phases are important for the design process, the dissertation focuses on developing and delivering, specifically for design tasks wherein settings for particular parameters determine individual design variations. In this context, we can regard the two solution phases jointly as "manual design optimization."

2.1.2 User-Centered Design

While providing a framework for high-level design principles, the double-diamond model is not furnished with the specific activities necessary for

manual design optimization. To address this gap, I consider another widely known design workflow, that of the UCD process [1]. It emphasizes close collaboration with actual users throughout the design process. Vredenburg et al. [218] define UCD, the origins of which lie in a concept from the early work of Norman [167, 166], as "the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and a multi-disciplinary approach." Detail-level variations in the procedures notwithstanding, all version have several key steps in common, among them gaining understanding of the users and context, designing and prototyping, user testing, and analysis.

The first step in the user-centered design process is to gain **understanding of the users and context**. The designers investigate why the users need the interaction, how they actually engage in it, and in what context they use the interaction (although this step is essential to any design process, it lay beyond the scope delimited by my research questions, involving well-defined design problems only). Then, with a well-defined design space and set goals, which should be aligned with the findings from the first step, the designers carry out **designing and prototyping**. They create a working prototype for communicating with others and conduct user testing. The third step is **user testing**: the designers invite real-world users to interact with the prototype created. This step, aimed at understanding its usability, is also called user testing, a user study, or evaluation. User testing should yield data by means of which the designers can improve the design further. Once the data are gathered, the next step is **analysis** performed to produce useful information. Designers apply appropriate analysis to judge whether the prototype constitutes the final design.

It may need improvement. Designers iterate over these steps until arriving at a final design. Informed by their analysis, they return to previous steps and make improvements until reaching an acceptable design instance. Note that, while these cycles may require revisiting the first step, the aforementioned scope factor precludes delving into it in the dissertation.

2.1.3 Empirical Methods for Design Evaluation

My review of prior work highlighted how crucial user testing is for all design processes as designers conduct experiments and analyze data to uncover the best design instance(s). Empirical research is a well-established quantitative approach to comparisons for particular design qualities [125, 142]. Also, HCI researchers often rely on empirical research and statistical analysis when setting parameters for interaction techniques.

Haptic feedback for button design is among the many examples one could consider. It involves defining parameters such as vibration duration, am-

plitude, and feedback point, all the way to more complex vibration profiles [41]. Even fairly simple interaction such as this demands multiple design choices: Designers search for the optimal design over a high-dimensional design space by means of manual tuning and experiments. Because this can be extremely time-consuming, researchers often select a few parameter sets by referring to experience and experimenting to determine the best ones. They repeat this process until arriving at an acceptable outcome, then iterating at higher level until a single final good design emerges [76]. This can never be the global optimum, however, because of the large number of unexplored designs.

Empirical experiments also serve effectiveness comparisons among various input methods, whether evaluation of desktop input devices such as mouse vs. trackpad [64] or comparison of input methods for mobile devices [168]. Recent studies have even compared particular gaze-input methods' utility for virtual-reality input [44]. Empirical methods also inform settings for parameter values in pilot studies. In all of these cases, a complicating factor rears its head, though: again, the designers can only conduct comparisons for a small portion of the design space, thereby potentially omitting those design instances that are truly/globally optimal. Therefore, we need a more systematic method of selecting promising candidates.

Limitations of Manual Optimization

While the traditional approach allows designers to reach an acceptable solution, it faces several limitations. The most critical of these is that every step is costly: One must ideate a design instance and may have to create a prototype before user testing. Running a user study requires careful preparation, recruitment of participants, and all the effort of conducting the experiment. Then, the designer must analyze the data and extract useful information before being able to derive the next design instance. Often, prohibitively high costs prevent the designers from conducting a full-fledged UCD process in which they could explore every potentially good design sufficiently; instead, they explore only a few design instances in the vast space and then jump to a conclusion. With numerous design candidates possibly remaining untested/undiscovered, the final design is suboptimal. A further practical constraint is that designers often lack the time, technical, and human resources for completing multiple iterations of user testing. Ultimately, the final design depends purely on what "feels right" to the designers and the developers.

The other most critical limitation is that UCD hinges largely on the designer's decision-making. Many sources assist in design decisions, such as the designer's expertise in a specific domain, past experience, and even intuition. Although resources such as expertise and experience can be helpful, they may exert a harmful influence too. For instance, designers may suffer from design fixation and not explore the design space well

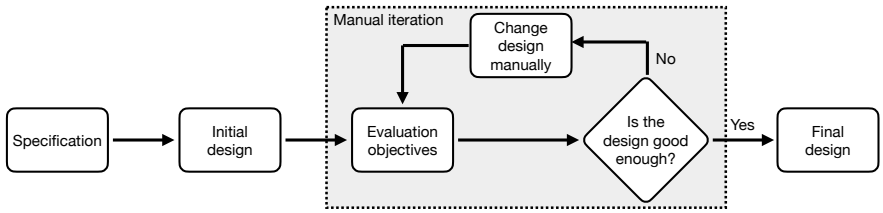


Figure 2.1. Adapted from the figure in Martins and Ning [145]: Illustration of the manual design workflow in engineering fields.

enough. In addition, it is important to recognize that the task of design optimization is quite complicated by nature. As subsection 1.1.4's review of design optimization's challenges attests, the optimization is a reasoning process aimed at understanding a complicated black-box function, whose input and output both may have multiple dimensions.

In summary, this process that is necessary in every design endeavor brings with it limitations and challenges that lead to inefficiency and a suboptimal outcome. In light of this vital issue, the next section reviews how engineering fields have explored the use of computational methods to aid in said disciplines' design-optimization process.

2.2 Engineering Design Optimization

Engineering design optimization is methodology whereby algorithms and tools support engineers' efforts to identify some optimal design solution(s) from among many other alternatives. Martins and Ning's book *Engineering Design Optimization*, which provides an overview of the methods involved [145]. It points out that, compared to the traditional manual iteration process (depicted in Figure 2.1), the computational optimization workflow (presented in Figure 2.2) is a more principled technique. The computational optimization process offers three major benefits: better performance, lower cost, and less uncertainty. It is important to note that certain basic differences exist in design optimization's application between the design field and engineering disciplines. To avoid confusion, I use the term "engineering design optimization" for optimization employed in the latter while "design optimization" denotes optimization in the design or HCI domain.

Engineering design optimization involves three fundamental components: design variables, objective functions, and constraints. **Design variables** are the parameters by which the design space is defined, where particular values assigned to the variables result in mutually distinct design options. The **objective function**, in turn, provides a metric for which the engineer aims to optimize, as introduced above. **Constraints** are the conditions that the optimization process must fulfill.

Engineering design optimization has a rich history and has been ap-

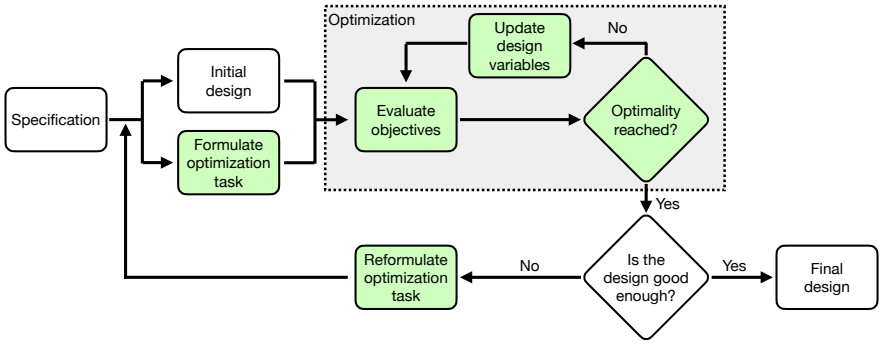


Figure 2.2. Adapted from the figure in Martins and Ning [145]: Illustration of the optimization-driven workflow in engineering fields.

plied extensively in a wide range of professions. Mechanical engineering [184, 192, 141], chemical engineering [19, 161], and architecture design [151, 201] are among the many domains that have employed optimization techniques. With the advent of complex optimization problems at the intersection of multiple domains, multidisciplinary design optimization has attracted attention as a promising avenue [144]. For instance, aircraft optimization is a complex problem that involves optimizing aerodynamics, structures, and controls [8, 84].

At this juncture, it is worth considering the aforementioned notable differences between engineering design optimization and design optimization in HCI. While the fields show similarities in their use of computational algorithms and methods such as gradient-based optimization, black-box optimization, and modeling, developers in engineering often have access to detailed models of physical phenomena, which allows for optimization through simulations. In contrast, HCI and design practitioners rarely have a perfect user model for a specific interaction at their disposal. Since this makes optimization through simulation difficult, human evaluators are typically needed, to interact with each design instance. To address this distinctive factor, the dissertation focuses specifically on applying black-box optimization methods without preexisting user models.

2.3 Human-in-the-Loop Optimization

Robotics and machine-learning researchers have carried out extensive work on the type of optimization that formed the core of my project. I begin this section with a review of HILO in these fields, and its application in design and HCI is covered later, in Chapter 3's discussion of related work.

To build a robotics system that delivers better human–robot interaction, researchers have developed optimization techniques that retain the human aspect of the optimization process [54]. Many of their studies have focused

on wearable robots and exoskeletons. For example, Zhang et al. [233] optimized an assistive exoskeleton system that led to substantial improvements for individuals during walking, and Ding et al. [55] proposed using HILO to optimize hip actions' assistance through a soft exosuit. Fang and Yuan, meanwhile, employed it for adjusting wearable ankle robots to minimize metabolic cost [61], and other, similar work investigated using Bayesian optimization to regulate the step frequency of walking for the same minimization objective [111]. This line of research [63, 220, 97] takes the human as the source for the objective function (again, the measurement result that the systems aim to minimize or otherwise optimize). The system iteratively updates the relevant system parameter setting on the basis of the user's response or performance. My project followed similar lines, with human participants in the process providing the evaluation of objective functions.

The goals and methods characterizing HILO's extensive use in the machine-learning field differ from those in robotics. Machine-learning researchers are more focused on improving how models are trained by exploiting experts' input and domain knowledge [225, 226]. They have employed HILO for data's preprocessing and annotation. For example, Self et al. proposed interactive parameter adjustment informed by human models [197]. Gentile et al. introduced an interactive dictionary-expansion tool based on language models [75]. In other contributions, Zhang et al. summarize ways of efficiently extracting training entities with humans in the loop [234], and Liu et al. proposed using humans in combination with reinforcement-learning agents for efficiently labeling data [138]. Finally, HILO is widely applied in natural-language processing; e.g., humans may aid in parsing the data [88, 230]. While human-in-the-loop methods for machine learning form an important part of the overall picture, their use deviates from the goals and methods central for this dissertation.

2.4 Bayesian Optimization

As an optimization method suited to grappling with black-box functions, BO has proven to be one of the most popular methods for HILO in HCI and design tasks. Below, I summarize the basics of BO. This technique updates a surrogate model iteratively for each observation. Typically, this model is a Gaussian Process regression. Then, BO uses an acquisition function that takes the surrogate model as input and generates the next sample. Bayesian optimization is the most suitable for expensive functions (that is, for conditions in which the function evaluation is time-consuming or costly). Also, it exhibits its greatest effectiveness when there are relatively few parameters (e.g., with a dimension value below 20). For a more detailed introduction of BO, the reader is referred to prior literature [68, 29].

2.4.1 Mathematical Properties of Bayesian Optimization

BO is designed to tackle the optimization of black-box problems. We can define this set of problems as

$$\max_{x \in X} f(x), \tag{2.1}$$

where x is a design instance and X is the design space constructed from all the design parameters. The design parameters and objective functions should follow these principles:

- The input ($x \in \mathrm{R}^d$) for such a function f has d dimensions, where d should be less than 20 (a smaller d enables more efficient and effective searching).
- The objective function (f) should be continuous, so that it can be modeled by GP.
- f does not feature any known special structure (such as concavity or linearity) or need to meet certain requirements.
- When observing f, we observe only the output of the function ($f(x)$), not the first or second derivative. Hence, this is a gradient-free task.
- We regard f as a black box, and BO searches for the global optimum.

The general algorithm for BO is described by Algorithm 1's pseudocode.

2.4.2 The Gaussian Process

Since BO typically uses GP as the surrogate model, I briefly introduce the mathematical definition and properties of GP, again per a concise version adapted from work by Frazier [68] (for more in-depth description of GP, please refer to Williams [222]). This form of regression is a Bayesian statistical method for fitting black-box functions. Assuming we have a set of design instances $[x_1, x_2, x_3, ..., x_n]$ and the corresponding objective function value $[y_1, y_2, y_3, ..., y_n]$, GP takes all these observations and fits them into a multivariate Gaussian distribution with a particular mean vector and a particular covariant matrix. In Bayesian statistics, this distribution is usually referred to as the prior distribution. The mean vector is derived via a **mean function** μ_0 at each x_i; meanwhile, the covariant matrix is derived through a **covariance function** (also commonly known as a kernel), Σ_0, for each pair x_i and x_j, where both x_i and x_j come from the design instances we have gathered. We can formally describe the fitted prior distribution as

$$f(x_{1:k}) \sim Normal(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k})). \tag{2.2}$$

With this prior distribution, we can then infer the function value for $f(x)$ at a particular position x via Bayes rules.

$$f(x)|f(x_{1:n}) \sim Normal(\mu_n(x_{1:n}), \Sigma_n(x_{1:n}, x_{1:n})), \tag{2.3}$$

where the mean function μ_n and the covariance function Σ_n are conditioned by the previous n observations. Such a conditional distribution $f(x)$ is also known as the posterior distribution.

In summary, this process's regression is a probability distribution that is derived from a mean function and a covariance function. We can update the distribution by means of the Bayes rule, and we can also infer the value of a certain point via conditional distribution. The technique's user should select an appropriate kernel and set the hyperparameters before applying BO. Because evaluating the GP is much cheaper (in time and financial cost) than evaluating the actual function (problem), BO uses a special function to search the GP space and identify which point has the greatest value. For this GP evaluation function, we use the term "acquisition function."

2.4.3 The Acquisition Function

The other essential component of BO is the acquisition function [223, 68]. The AF reads in a design instance (x) as input, and the outputs ($AF(x)$) indicates the potential value of evaluating this specific instance. After refitting the surrogate model, BO evaluates samples across the design space X. Each sample results in an acquisition value ($AF(x)$). The design instance x that leads to the highest acquisition value will be selected for evaluation by the actual function (for the problem proper) in the next iteration. The most representative of the commonly used acquisition functions are introduced below. For a more detailed introduction to additional AF types, the reader is referred to Garnett's work [73].

The Upper Confidence Bound:
One of the easiest AFs to handle, the UCB functions calculates an upper bound for each design instance x. The UCB is composed of two parts: predicted mean and variance. A design x with a high predicted mean naturally has greater potential to lead to a higher objective value. Variance too is relevant for this potential: a design x displaying a higher variance value indicates that the surrogate model is rather uncertain of the prediction at this point; hence, there is potential for a high objective value to result irrespective of the mean. We can construct the UCB thus:

$$UCB \equiv \mu(x) + \lambda\sigma(x), \tag{2.4}$$

where $\mu(x)$ and the $\sigma(x)$ are the mean and variance values at x, while λ is the hyperparameter controlling the tradeoff between mean and variance.

Expected Improvement:
The EI value represents the *expected improvement* at a potential design x. At a high level, EI is composed of two elements: the first element is the difference between the predicted value of potential design x and the best objective value observed so far, and the second element is related to the standard deviation at x. Intuitively, a higher EI value indicates greater potential improvement from sampling this point from the real-world function. We can formulate EI as

$$EI \equiv E[f(x) - f_n^*]^+, \tag{2.5}$$

where f_n^* indicates the best observation within the past n iterations and the + sign indicates considering only cases wherein this value is positive. If $f(x)$ is less than f_n^*, leading to a negative value, the EI function will be evaluated as zero.

Probability of Improvements:
The predicted improvement (PI) function calculates the probability of sampling at x and retrieves a value that is better than the current optimal observation. This is similar to EI, but, rather than directly compare the mean value, PI integrates the probability by using the GP model. We can formally describe PI as

$$PI \equiv P(f(x) \geq f_n^*), \tag{2.6}$$

with f_n^* indicating the best observation thus far and P calculating the integrated probability.

2.5 The State of the Art in Summary

With this chapter, I have provided an overview of the fields that developed the foundation for this dissertation. For some of this background, I outlined the typical design processes, with particular attention to design-thinking models and user-centered design. As the design process nears completion, designers need to select the final design instance(s) addressing the design question specified. Traditionally, designers perform design optimization manually with the support of empirical research methods. However, relying on intuition coupled with empirical methods may not lead to the best outcome, since thorough empirical research requires a significant amount of effort. Since, accordingly, designers often end up choosing from among a considerably limited set of tested designs, a more principled approach is sought in optimization guiding the selection of design instances.

This points to a possible way forward via engineering design optimization, wherein researchers employ well-established computational tools to assist engineers in making decisions. However, fundamental differences between design and engineering settings, arising mainly from a lack of solid user models for specific interactions, typically necessitate retaining interaction with human evaluators when computational methods are deployed for design optimization.

The final element of the review, my examination of human-in-the-loop optimization in other contexts – robotics and machine learning – showcases HILO's potential as a general framework applicable for a wide range of applications but also reveals that the approach has not yet seen widespread use in interaction design. Having pinpointed various constraints that currently restrict the application of HILO, I set out to address these gaps.

3. Related Work: HILO in HCI

This chapter provides an overview of work specific to the techniques and domains most relevant to my project. I begin by reviewing human-in-the-loop Bayesian optimization, which formed the core of my research. Then, I examine a wide range of other works on HILO and adaptive user interfaces in HCI. I end the introduction by presenting the optimization toolkits available, in brief.

3.1 Human-in-the-Loop Bayesian Optimization in HCI

This chapter narrows the focus from HILO as introduced in the previous chapter (a general framework to solve parameter-optimization problems in which human participants provide the evaluation functions) and from the BO approach to tackling expensive black-box functions, reviewed there as a promising approach for HILO tasks [199, 24]. Whereas (chapter 2 reviewed the use of HILO in robotics work largely concentrating on refining the parameter settings of wearable systems, we now direct our attention to reviewing research conducted in the HCI domain.

Most HCI work employing BO has used it as a design tool. In various ways, studies have addressed its requirement for the designers to provide objective function evaluation. Brochu et al. [28] demonstrated using Bayesian optimization in conjunction with human designers' subjective ratings for quickly identifying an appropriate hyperparameter setting for realistic animation rendering. Koyama and various colleagues, in turn, proposed applying BO to aid visual designers who optimize for parameters of visualization via sequential linear selections [122] and gallery-based selections [121]. The aforementioned work in which Yamamoto et al. [227] utilized BO for automatically tuning photographic lighting settings is another example of the variety of applications. Other research, by Zhou et al. [237], showcased allowing composers and musicians to generate a melody via BO, and Piovarči et al. [176] used Bayesian optimization to search for design parameters optimized for target friction and vibration

objective metrics.

In experiments following a different technique, the end users act as the objective-function evaluators while the designer's role is to supervise the optimization process. Khajah et al.'s aforementioned work with gamers to fine-tune game-parameter settings for maximal user engagement [108] is one example. Another such application of BO is work in which Kadner et al. [102] customized font designs to optimize the reading speed of individual users. Nielsen et al. showcased fine-tuning hearing devices with BO [164], while Snoek extended the gaze to other assistive technologies [204]. Also worth noting is Dudley et al.'s work [57], which utilized BO to optimize 2D map design directly via a group of end users.

Researchers have dedicated considerable effort to investigating means of making BO more suitable for optimizing interaction or design. An important stumbling block to BO's higher efficiency is that it typically learns "from scratch." Accordingly, Brochu et al. sought more efficient optimization by attempting to transfer the kernel learned from previous tasks to the current task [28]. Studies also have explored using crowdsourcing to refine design-parameter values quickly, as opposed to relying on a single user's evaluation [57, 122]. For addressing high-dimensional design challenges, the Koyama teams' publications considering the visual realm propose line search wherein the user makes one design judgment at a time. Taking a different task, Koyama and Masataka investigated having BO act as more of an assistant; here, the designer has greater flexibility to take or ignore the BO-produced design suggestions [120]. Finally, several studies have examined the use case for preferential BO in design tasks (i.e., the designer simply comparing two possible designs at a time) [227, 120].

These studies' success notwithstanding, application of human-in-the-loop BO is still limited to the scope identified in the first chapter: its focus today remains on single-objective optimization cases, customizing for a single user, and interactions entirely confined to graphical user interfaces. It is precisely these limitations that motivated me to extend human-in-the-loop BO for other applications.

3.2 Online Optimization Systems in HCI

Besides BO, prior work in HCI has explored other computational tools for HILO and online optimization. To provide context, this section revisits the most representative of those tools.

Bandit Systems

An alternative approach still closely related to Bayesian optimization is the formulation of design-space optimization as a multi-armed bandit problem. This formalization refers to the task of selecting from among

several alternatives offering uncertain outcomes so as to maximize some gain – in the example of a row of slot machines, which machine should be played next if one wishes to maximize one's winnings. Multi-armed bandits have been proposed and indeed demonstrated as a tool for assisting in interface design where the selection problem becomes one of choosing an interface-design alternative that maximizes some utility [137, 139, 4]. The same exploitation–exploration tradeoff is evident, in that the solution involves balancing learning more about potential alternatives vs. consistently preferring a known good design. It is possible to apply a Bayesian treatment [195, 199] to this selection problem, in which case the approach closely resembles Bayesian optimization.

Adaptive User Interfaces

Adaptive user interface are intended to provide means by which the interface or interaction technique adjusts to varying user capability, interest, and behavior. Extensive research into adaptive user interfaces has considered diverse computational techniques, across many devices and domains; I review only the most representative work here. Many adaptive interfaces are based on heuristics and logic. For example, Puerta et al. proposed a system that automatically generates design based on the user model and a set of "design rules" [182], and Gobert et al.'s system adapts menu in accordance with adaptation policies and the user's interactions [78]. These methods are limited in their adaptation scope, however. In more complicated cases, the heuristics or rules cannot fully cover the whole spectrum of possible scenarios. Moreover, these systems require a set of known rules, which is not a workable assumption for all interactions.

Looking beyond heuristics, researchers have also investigated exploiting computational solvers or optimizers to identify the design that optimizes best for given objective functions. For example, Belo et al. [60] proposed a toolkit for adaptive user interfaces that applies solvers to find the optimal design. AutoGain [126] is an optimization method that fine-tunes the transfer functions of input devices on the basis of user aim errors. The gain functions are iteratively updated during the user's interaction. Bailly et al. [11] proposed an online menu-optimizer with an objective focused on predicted user performance. For sketching, meanwhile, Todi et al. introduced an optimization-supported interface [214] that updates the positions, color, and sizes of user-interface elements as interaction with the designer progresses. The objective functions for this optimization were based on design principles.

While my project is aligned with this direction of research, in which a computational optimizer is applied for some given objective function(s), the scope of my work is delimited more narrowly. Firstly, while prior work has explored a wide range of solvers, from grid search [140] to evolutionary algorithms [11], the dissertation focuses principally on applications of BO.

One important reason for this focus on BO is that it offers a general framework that can be extended to a broad range of scenarios. For example, other optimization solvers may not be extended for such aims as my goal of extending HILO from a single objective to multiple objectives. In contrast, BO supports altering the acquisition function, whereby the optimization is guided to explore all the optimal designs in a high-dimensional objective space. The other crucial difference between my work and that on general adaptive user interfaces lies in the overall goal. This dissertation examines only parametric optimization tasks, while previous publications have tackled different aims. For example, some studies have focused on assignment problems (menu items and keyboard assignments), problems better served by methods such as combinatorial optimizations [172]. Other research has considered the effects of adaptation, such as learning and fatigue, over a longer interaction span [213, 46, 65, 83]. The optimization problems examined in this dissertation are specific to shorter episodes, not long-term human adaptation.

3.3 Preexisting Optimization Tools

The dissertation project employed several varieties of Bayesian optimization. Some optimization tools providing similar functionality are available for ready use by developers with programming abilities. Single-objective optimization is provided through established libraries such as scikit-optimize[1] for Python and the Statistics and Machine Learning Toolbox[2] for MATLAB.

With regard to the greater complexity that multi-objective optimization entails, current packaged implementations vary in their level of maturity and capabilities. BoTorch [12] is a relatively full-featured and actively developed library implementing a particular variant of multi-objective Bayesian optimization [51]. Among the other packages that support multi-objective BO are GPflowOpt [117] (in the process of being revamped into Trieste[3]) and MOBOpt [72]. TS-EMO [26] is a MATLAB library implementation employing a closely related technique. These various projects chiefly target developers familiar with the techniques and, correspondingly, offer them good configurability. In contrast, with Bayesian Compass my research team seeks to offer the same core functionality but at a higher level of abstraction that provides better support for HCI researchers and developers naïve to the underlying techniques.

While we apply alternative optimization techniques, our goal is similar to that behind the implementations of Sağ and Çunkaş [193], Google

[1] Available via `https://scikit-optimize.github.io/`.

[2] Presented at `https://www.mathworks.com/products/statistics.html`.

[3] See `https://github.com/secondmind-labs/trieste`.

Vizier [79], and pymoo [22]. These efforts give greater focus to the visualization of the optimization process and its results, in conjunction with the aim of offering the user greater abstraction. Bayesian Compass is targeted specifically at HCI researchers and developers who wish to apply advanced multi-objective optimization in their design problems. To this end and in contrast to prior work, we seek to provide inspection tools along with relevant scaffolding and guidance specific to the HCI domain. Furthermore, we aim to build a MOBO-exploiting design workflow that is readily usable and applicable for a wide variety of tasks and that, in comparison to traditional design methods, offers the design process demonstrable benefits.

4. From a Single Objective to Multiple Objectives

"There are no solutions; there are only trade-offs." — Thomas Sowell

While work on human-in-the-loop optimization has restricted itself mainly to solving single-objective problems, the prevalence of real-world applications that may even involve competing objectives highlights a need for feasible multi-objective approaches. For instance, a good input-device design should balance accuracy and efficiency while also accounting for other relevant metrics. A design optimized exclusively for accuracy may not be efficient, in that it could allow highly reliable and precise selection but at the cost of significant time and effort. Conversely, a design that prioritizes efficiency may lead to low accuracy.

One commonplace approach for straightforwardly addressing multiple objectives in optimization problems is to aggregate several objective functions into a single scalar objective function by assigning weights to each objective. We can formally describe the weighted-sum approach thus:

$$Weighted\_Sum\_Objective = \sum_{i=1}^{n} weight(i) \cdot Old\_Objective(i), \qquad (4.1)$$

where i is the index of the original objective functions and $weight(i)$ is the weight assigned for a particular objective function. By aggregating multiple objective functions into one, we can perform single-objective optimization to solve the problem.

However, such an approach has several major drawbacks. The first is that objective functions differ in their units, so comparison of their magnitudes becomes difficult. For example, task-completion time, a popular metric for efficiency, is measured in seconds or milliseconds, while accuracy in tasks such as pointing is judged in terms of distance, with centimeters or millimeters as the unit. This makes it challenging to interpret their aggregate-level meaning. Additionally, aggregating multiple objective functions into a single one results in information loss, in that the original individual objective functions' values do not get considered in

the optimization process. The optimizer observes only the weighted-sum value, which may not accurately reflect the true performance of the system. Moreover, the difficulty of a designer or developer predefining good weights for all the objective functions may lead to a suboptimal user experience. Another limitation of the approach is that the problems must have a small number of objective functions, since weighting multiple objectives becomes increasingly complex as their number rises. Lastly, aggregating several objectives into a single objective function exacerbates the issue of nonlinear tradeoffs. For example, when greater accuracy entails decreased efficiency or when higher user satisfaction is accompanied by poor system performance, the relationship between the two factors may be far from obvious. Since a simple weighting system cannot capture this nonlinear relationship between objectives, suboptimal solutions may result. In short, conducting single-objective optimization with a weighted sum is not a principled solution.

With the discussion below, I aim to answer two chapter-level research questions (CQs):

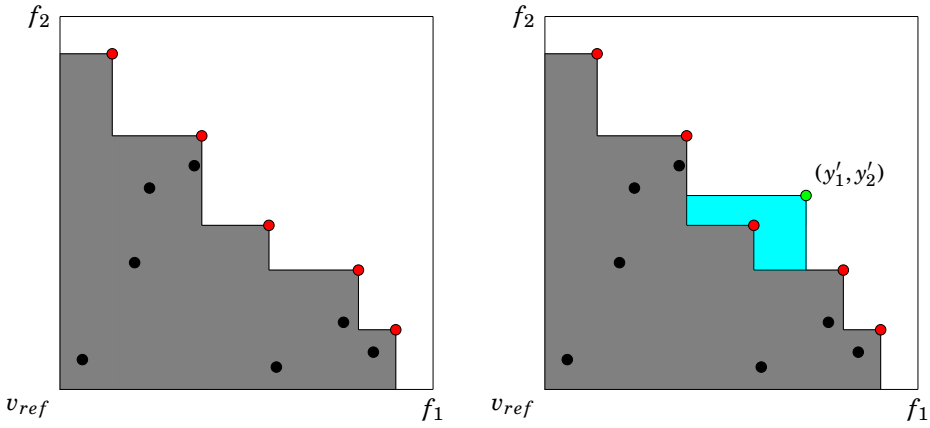
CQ 1: How could we enable HILO to tackle multi-objective design tasks systematically?

CQ 2: What are the qualities and performance traits of multi-objective human-in-the-loop optimization as compared to traditional design approaches?

4.1 Multi-Objective Optimization

To address the multi-objective design problem, a more principled solution than taking a naïve weighted-sum approach is to employ multi-objective optimization with **Pareto-front learning**. When conducting single-objective optimization, all we need to do is compare distinct designs by one metric, concluding with a final design that has the highest single-objective value. On the other hand, when performing multi-objective optimization, we have a set of objectives. In this case, instead of seeking a single optimum, we search for a series of design points that represent optimal balances of several objective functions. The set of design points is also known as the Pareto frontier, Pareto efficiency, or Pareto-optimal designs. Figure 4.1 gives an example with the Pareto-optimal design set shown as the points in red; these points exhibit the optimal tradeoffs between the two objectives, f_1 and f_2.

Clearly, a single optimum cannot be identified. As we progress along the Pareto front, compliance with one objective diminishes as that with the other grows. This pattern highlights the tradeoffs. We can formally describe Pareto-optimal designs thus: Consider an interaction technique with the function $f : X \to \mathbb{R}^m$, where X is the design space, which has



(a) Current Pareto hypervolume **(b)** Increase in Pareto hypervolume

Figure 4.1. Two example objective functions, illustrating the Pareto-optimal design set and the Pareto hypervolume. At left, points $\{(f_1(x_i), f_2(x_i)\}_{i=1}^{12}$ are shown as dots, with red ones representing the Pareto-optimal design set and black ones representing dominated points. The gray region is the current Pareto hypervolume with respect to the reference point v_{ref}. With pane b, a new observation is made at $(y'_1, y'_2) = (f_1(x'), f_2(x'))$, which dominates one point that was previously Pareto-optimal. The cyan region is the Pareto hypervolume increase after the observation (the green point). If the new observation is dominated by some previously observed point, there would be zero change in Pareto hypervolume.

the parameter space $\mathbb{R}^n$, and Y is the set of feasible objective-function vectors in $\mathbb{R}^m$ such that $Y = \{y \in \mathbb{R}^m : y = f(x), x \in X\}$. Specifically, m here corresponds to the number of design parameters and n refers to the number of objectives. An objective vector $y'' \in \mathbb{R}^m$ is preferred over (i.e., strictly dominates) another vector $y' \in \mathbb{R}^m$ when all of its elements are greater than the second vector; this is denoted as $y'' \succ y'$. Formally, the Pareto front is expressed as the set $P(Y) = \{y' \in Y : \{y'' \in Y : y'' \succ y', y'' \neq y'\} = \emptyset\}$.

From an intuitive standpoint, one can regard the Pareto front as capturing the need to sacrifice satisfaction of a specific objective function when pursuing improved performance for another objective function. In our example of an input-device design with two objective functions (efficiency and accuracy), if we randomly select two designs on the Pareto front, one of them must exhibit higher efficiency but lower accuracy than the other. The idea behind employing multi-objective Bayesian optimization is to search for the Pareto-optimal designs, the designs that lead to the Pareto-optimal objective functions.

How, then, do we enable BO to search for the Pareto-optimal design? For an answer, we have to introduce the concept of Pareto-front learning.

4.1.1 Pareto-Front Learning

In essence, Pareto-front learning is searching for the Pareto-optimal designs for a multi-objective optimization problem. Prior work has demonstrated how to reach this goal by maximizing the Pareto hypervolume. To understand the latter concept, consider some reference design $\mathbf{v}_{ref}$ that is inferior to all of the Pareto-optimal designs. In practice, this can be taken as the point corresponding to the lower bounds of each of the m objectives. We define the Pareto hypervolume with respect to $\mathbf{v}_{ref}$ as the hypervolume bounded above by the Pareto-optimal design set and bounded below by $\mathbf{v}_{ref}$. A new design point that improves upon at least one of the Pareto-optimal designs would therefore yield an increase in hypervolume. This concept is illustrated in Figure 4.1 with two objectives. Pareto hypervolume can be used as a measure of how good the current estimate of the Pareto-optimal design set is since the more dominant the Pareto front, the larger the Pareto hypervolume. Thus, in multi-objective optimization, the Pareto hypervolume functions as a good proxy for the quality of a set of Pareto-optimal points.

Intuitively, we would like the next point sampled to increase the Pareto hypervolume as much as possible, since that would correspond to a significant improvement in the estimate of the Pareto front. There are various ways of reaching this target. For example, Yang et al. and Daulton et al. proposed differential hypervolume increase [229, 51]. Shah et al. [198], in turn, proposed an acquisition function indicating the approximate expected improvement in Pareto hypervolume in the case where objectives are assumed to be correlated, which they referred to as the correlated expected improvement in Pareto hypervolume. There are other implementations [58, 72, 52], but they share the same central idea – to maximize the hypervolume. Pseudocode for Pareto-front learning is presented below, as Algorithm 2.

For the next two sections of this chapter, I applied MOBO to two distinct design cases. In the work presented here, I relied mainly on the methods of Daulton et al. [51] and Shah et al. [198]. Future work should explore the efficacy of other implementations and benchmark the resulting performance.

4.2 Designing with Multi-Objective HILO

MOBO with Pareto-front learning offers a principled solution for multi-objective human-in-the-loop optimization, thus potentially speaking to CQ 1. Furthermore, to assess the efficacy of MOBO and the experience of using it in design tasks (per CQ 2), I invited designers to work on a design-optimization task (optimizing the design of tactile icons) in the context of

Algorithm 2 Pseudocode for multi-objective Bayesian optimization

Inputs:
Take a given d, v_{ref}, $n \leftarrow 1$, and iteration count N.
Initialize:
Randomly sample d parameter sets for querying the user, and set $\mathscr{D} = \{x_i, \mathbf{y}_i\}_{i=1}^{d}$.
for $n \leq N$ **do**
  Construct a Gaussian Process from $\mathscr{D}$.
  Compute the current Pareto-optimal design set.
  Use Sobol sampling to obtain $x_{new} = \arg\max_x EHVI(x|\mathscr{D})$.
  Query the user to obtain $\{x_{new}, \mathbf{y}_{new}\}$.

  $\mathscr{D} = \mathscr{D} \cup \{x_{new}, \mathbf{y}_{new}\}$
end for
return x_n with the optimal y_n

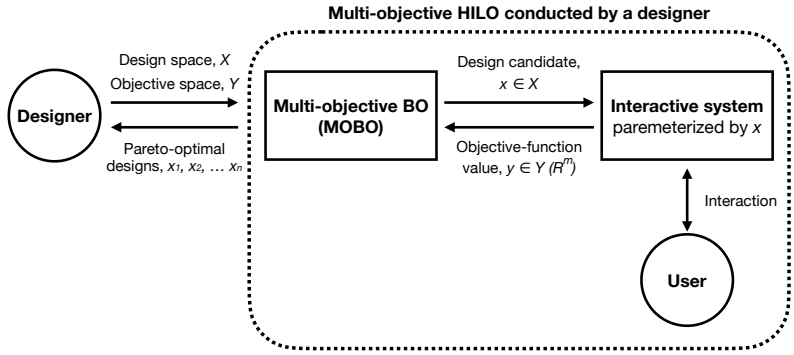


Figure 4.2. Multi-objective HILO conducted by a designer.

a workshop I organized. The workshop anchored in CQ 2 was oriented toward two goals: 1) understanding the designers' experience of working with MOBO and 2) understanding how MOBO-based design differs from typical manual optimization procedures. The main finding emerging from the workshop is that multi-objective optimization significantly reduces the total effort required for making decisions in the course of a complex, multi-objective design task. My workshop also revealed that MOBO can effectively assist designers in managing tradeoffs between objectives.

4.2.1 Interaction: Tactile Icons

The design task in the workshop setting was to optimize the set of haptic feedback from a haptic display. One can convey information to the user in a simple manner with a single tactor by using unique vibration cues (combinations of vibration duration and amplitude) to represent each distinct message. Naturally, a set of vibrations that features more unique

combinations carries more information, but it becomes more difficult to distinguish between individual cues as the number of unique vibration cues increases. To constrain the dimensionality of the design space, I limited the task to four parameters: the designer had to specify the vibration duration and amplitude's minimum values and set the number of distinct levels for duration and amplitude.

While the first two of these design parameters (representing the minimum duration and minimum amplitude of vibration) were specified by the designer, the maximum duration of vibration (1 s) and the maximum amplitude (1.45 g) were set in advance. After the designer's decisions on lower bounds came the selection of the number of levels for vibration duration, N, and amplitudes, M. The result was N duration levels, equally spaced between the minimum and maximum, and M levels of amplitude, distributed analogously. These conditions yield $N \times M$ distinct possible vibration cues in total. Two objectives were taken into account: the information-transfer rate (IT) represents the channel's capacity, and accuracy was judged in terms of recognition difficulty.

The Background to Tactile Icons

Investigating and optimizing ways of transmitting information via skin sensations has been an important enduring goal for haptics researchers, and it gained still greater relevance with the emergence of smartwatches [74, 116, 130, 147]. Prior work has investigated generating vibrations with various durations, frequencies, and amplitudes with a single vibration tactor [209, 211, 211, 27]; transmission of spatial patterns by means of multiple tactors [130, 128, 133]; and the use of motor-driven skin-drag displays for continuous spatial patterns [96].

These efforts notwithstanding, scholarship has produced only extremely limited conclusive guidance as to what constitutes the "best" design for a given context of use [208]. One critical reason for this lack of clarity is that optimizing a tactile display depends not on one performance metric but on several. The goal of most studies has been to minimize recognition error and maximize IT. These are mutually conflicting objectives, yet prior research has not effectively accounted for multiple objectives. According to information theory, there must be an increase in entropy – corresponding to a large set of possible haptic cues – if the rate of information transfer is to increase; however, a larger number of stimuli also leads to lower recognition accuracy, as noted above. In the absence of a principled algorithm for efficiently seeking Pareto optima, the methods available usually consider only one objective or involve manually eliminating non-suitable designs [140, 37, 41, 133]. Because of the time and effort involved in conducting empirical experiments, most efforts have curtailed the exploration of design alternatives after a handful of iterations. Given these "budget constraints," Bayesian optimization offers a clear advantage in facilitating

Table 4.1. Design parameterization for the haptic display.

Design parameter		Range or set
x_1:	Minimum duration of vibration	[50 ms, 950 ms]
x_2:	Number of discrete vibration-duration levels	$\{1,2,3,4\}$
x_3:	Minimum amplitude of vibration	[0 g, 1.45 g]
x_4:	Number of discrete amplitude levels	$\{1,2,3,4\}$

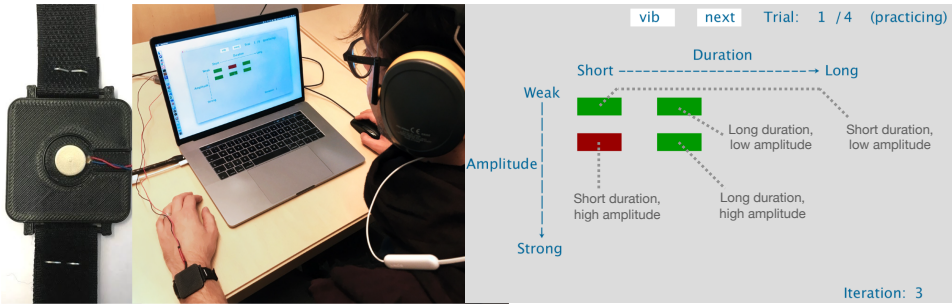
an efficient search of the design space.

Design Parameters
The ranges for the four design parameters adjusted in the interface study (the minimum duration, number of duration levels, minimum amplitude, and number of amplitude levels) are listed in Table 4.1. Again, the discrete nature of the two parameters for the number of levels (x_2 and x_4) is worthy of note: the intervals were assigned via even segmenting of the continuous space for the specified number of levels.

Objective Functions: Information Transfer and Recognition Accuracy
Information transfer (IT) represents an estimate of the channel's capacity to communicate with a given set of stimuli – i.e., the information, in bits, successfully transferred per stimulus. A standard way to measure and compute the IT value for a certain communication channel is to conduct an absolute identification study. For details of IT computation, I refer the reader to section II A of the paper by Tan et al. on this topic [208]. While IT is a general metric for the effectiveness of a communication channel, one should take recognition accuracy into account also, to mitigate error effects. Recognition accuracy over a set of stimuli is calculated as $Accuracy = \frac{n_{correct}}{n}$, where $n_{correct}$ is the number of correct trials (the user's response is matched with the stimulus); n is the total number of trials.

4.2.2 The Workshop

I conducted the workshop to understand how the MOBO-assisted procedure compares to a traditional one for manual design optimization. The eight participants in the workshop all were interaction designers with industrial-design experience. Each designer was assigned two other participants, who acted in the role of users when the designer needed to test designs. Figure 4.2 illustrates the MOBO framework that was employed to assist in the design process. For clarity's sake, my description below refers to the participants as "designers" when they acted in that capacity and as "proxy users" when they were testing another participant's designs.



(a) **(b)** **(c)**

Figure 4.3. The prototype (a), experiment setup (b), and interface (c). Publication I provides more details of the setting.

The Experiment Design

The experiment is a within-subject design with one independent variable. There were two conditions: the design procedure supported by MOBO, which is referred to here as the MOBO-assisted procedure, and the procedure of the designer's choice, which I refer to as the designer-led procedure. For the experiment, I counterbalanced these two conditions.

The Apparatus and Prototype

The team built a 3D-printed smartwatch prototype ($4 \times 4 \times 0.5$ cm) with a single vibration motor, a Precision Microdrives 310-113 unit,[1] as shown in Figure 4.3a. The user interface, depicted in the figure's pane c, was developed in the application Processing.[2] The interface is composed of a grid of boxes, where each box represents a distinct vibration cue corresponding to a given amplitude and duration. In this interface, the boxes are arranged in accordance with amplitude and duration such that there are x_2 columns and x_4 rows.

Procedure

The designers were each provided with a prototype smartwatch (see Figure 4.3a) and two proxy users. The study setup was as shown in Figure 4.3b. In the procedure for the users' identification task, the interface (shown in Figure 4.3c) displays the vibration designs to the proxy users in each iteration, with each box in the interface representing a unique vibration pattern. These vibration designs are generated by either the designer or MOBO. Each iteration starts with "practice mode," in which the proxy users are presented with a unique vibration cue at random and the corresponding box in the interface is marked in red. Practice mode was designed to familiarize the proxy users with the vibration set in question. It is followed by "identification mode," an evaluation setting in which each vibration cue

[1] See `https://www.precisionmicrodrives.com/product/310-113-10mm-vibration-motor-3mm-type`.
[2] See the site `https://processing.org/`.

is displayed to the proxy user, who then has to identify it. A span of three hours was assigned for each condition.

The designer-led procedure: The designers were asked to assign values directly for the four design parameters and present the resulting vibration set to the proxy users. In each design iteration, when the work in identification mode (after practice mode as presented above) was completed, the designer viewed the achieved recognition accuracy and the number of cues. After the three hours assigned to the designers had elapsed, they were asked to determine one final design.

The MOBO-assisted procedure: The MOBO implementation utilized was based on the method proposed by Shah et al. [198]. Designers configured the MOBO, then commenced the human-in-the-loop optimization involving the two proxy users. In this procedure too, the designers had three hours for completing the task. They had to select one final design from the Pareto frontier as the final outcome.

After the experiment (i.e., the tasks in both conditions), the designers were presented with the designs they had derived and the corresponding user-performance results for each. To understand their experience, the designers were asked to fill in the NASA-TLX and the System Usability Scale (SUS) questionnaire.

4.2.3 Results

Both the final designs and the user-performance levels achieved proved very similar between the designer-led procedure and the MOBO-assisted one. The mean values of the recognition accuracy and the number of distinct vibration cues arrived at via the manual procedure were, 0.863 ($SD = 0.078$) and 6.125 ($SD = 1.36$), respectively. The corresponding performances for the MOBO-assisted procedure were 0.883 ($SD = 0.08$) and 6.125 ($SD = 2.031$), respectively.

Analysis of the Design Strategies in the Designer-Led Procedure
I noticed that the designers created various strategies to tackle the design problem in the designer-led procedure. This is indicative of the challenge, complexity, and higher mental load in this condition.

Strategy 1 – divide-and-conquer and then an increase in the vibration cues' complexity: Two designers undertook a divide-and-conquer approach by only tuning certain design parameters as the first step. Afterward, these designers gradually increased the complexity of the vibrations (i.e., with additional unique vibrations) in the course of the optimization process until both metrics met their expectations.

Strategy 2 – divide-and-conquer followed by decreasing the complexity of the vibration cues: In contrast, two of the designers applied a divide-and-conquer strategy similar to strategy 1 but then gradually

decreased the complexity rather than increasing it.

Strategy 3 – divide-and-conquer and local search: Two designers identified a reasonable starting design via testing with the proxy users. These reasonably set initial values were fairly close to their final designs. The designers then performed a "local search" strategy in which they slowly fine-tuned the design parameters until they had identified a set of satisfactory cues.

Strategy 4 – a self-evaluation approach: One designer evaluated the designs in a largely independent manner at first. After an hour of self-testing, this designer narrowed the field to three design candidates. The designer then invited the two proxy users to evaluate these candidates and selected the final one.

Strategy 5 – a focus group: The last designer's work involved a small "focus group." This designer invited both proxy users to spend five minutes creating their preferred designs independently. Then, each of the three evaluated all of the designs created by the others. The group then jointly discussed the approach to improving the design, then embarked on another round of iteration and evaluation use. Afterward, the group selected two final design instances.

Analysis of Usability and Workload

Wilcoxon signed-ranked testing was applied to analyze overall usability and workload on the basis of the questionnaires' results. Figure 4.4 shows the results for the individual questions. The mean SUS score was 54.375 ($sd = 15.51$) for the designer-led procedure and 64.375 ($sd = 13.48$) for the MOBO-assisted procedure. A Wilcoxon signed-rank test showed no statistically significant difference in overall usability between the two conditions ($Z = -1.022, p > 0.05$). Still, significant differences were found in the responses to specific questions. Differences were particularly evident for item 1 ($Z = -2.06, p < 0.05$) and item 4 ($Z = -2.97, p < 0.05$). This suggests that the designers felt both that they would like to work with MOBO more (item 1) and that they would need technical support in the meantime when designing with the optimizer (item 4).

Figure 4.5 shows the results for individual questions in the NASA-TLX questionnaire. The mean workload rating for the manual procedure was 62.67 ($sd = 16.36$), and it was 45.17 ($sd = 12.4$) for the MOBO-assisted procedure. A Wilcoxon signed-rank test revealed a significant difference between the workloads perceived for the two conditions ($Z = -2.38, p < 0.05$), indicating that the MOBO-assisted procedure reduced the overall workload. Examining the individual questions showed that MOBO decreased the task's mental demands ($Z = -2.197, p < 0.05$), physical demands ($Z = -2.06, p < 0.05$), temporal requirements ($Z = -2.366, p < 0.05$), and frustration ($Z = -2.527, p < 0.05$) specifically.

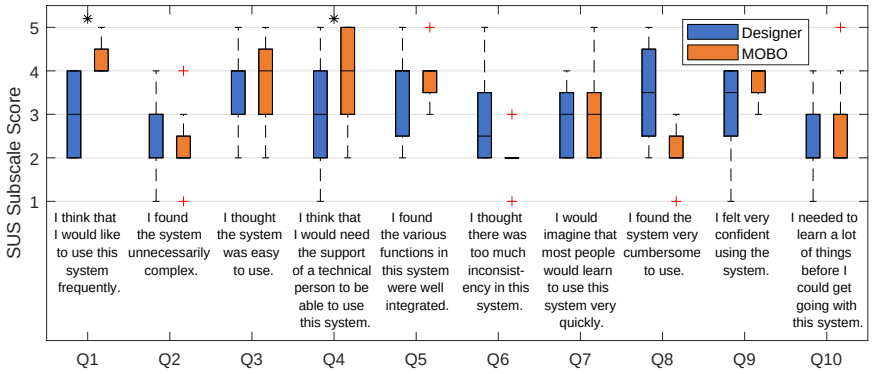


Figure 4.4. Boxplots showing the ratings given by the designers in the SUS questionnaire. Publication I provides details.

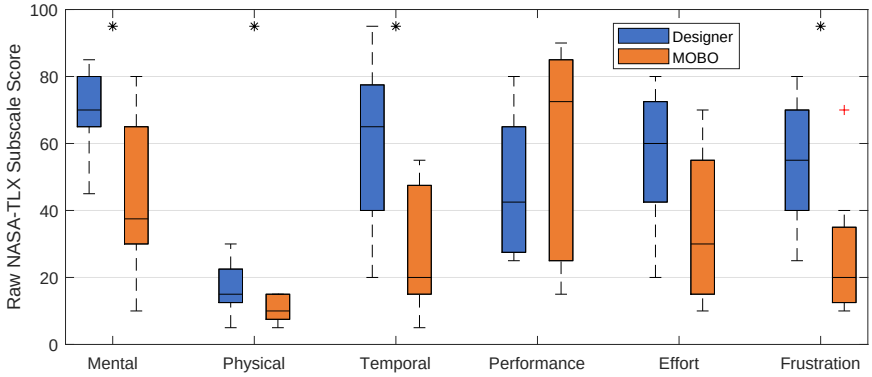


Figure 4.5. Boxplots presenting the eight designers' ratings from the NASA-TLX instrument. Publication I provides further details.

Qualitative Analysis

While the MOBO-assisted procedure and designer-led procedure generally reached similar outcomes, MOBO's assistance significantly reduced the designer's overall effort (mental and physical). The qualitative analysis pinpointed explanations for this.

With the designer-led procedure, designers must carefully create a strategy for tackling the design problem. Doing so requires extra mental effort. In addition, the designers had to expend time on "interpreting" and "making sense of" the results from each iteration and then selecting the next set of vibration cues. The MOBO reduced the designer effort required for determining the next design instance. Designers' responses in the interviews led to the same conclusions. One designer (D4) stated that the designer-led search "can go on forever. I can always change something and lead to a different performance. I always feel uncertain, not knowing if this change will improve [things] or not, and this is frustrating. Also, because I

need to deliver a design within a certain amount of time, so I was somehow stressed." Along similar lines, D1 shared that "I was not sure whether this design is good enough, so I felt it to be more temporally demanding. On the other hand, when using [MOBO], I simply needed to assign one hour [...] to each participant and collected the results. It is much simpler and relaxing."

From the user standpoint, all designers pointed out the benefits of having the derived Pareto-optimal designs. One (D1) observed, "If I changed my weights of the objectives and wanted to search for another design, I might need to invest another 30 minutes to reach that point. [The MOBO procedure's output visualization] showed all the designs along the line (Pareto front) and I could just pick one from them. From this perspective, I find [the MOBO procedure] much more efficient because it searches not just one final outcome but multiple." Going into greater depth, D5 explained, "I set some kind of priority at the beginning of the design. For example, the recognition rate is more important than the information transfer, and I want to achieve 95% accuracy. However, during the [manual] process, I might gain new knowledge about the interaction and would like to change the weight of the two objectives, which would force me to change the direction of the search. The [MOBO procedure] can avoid this kind of a hassle because it explores all the directions and provides all the possibilities."

The Workshop Overall

The workshop component of the research contributed to answering CQ 2. Two major findings emerged from the qualitative data: Firstly, MOBO reduced the effort of searching and of proposing the next design candidate. Secondly, the Pareto frontier gave the designer more flexibility to determine a final design. In contrast, the designer-led procedure requires the designer to have an implicit direction (apply some sort of weight) during the search process. Such internal weighting may change as the process progresses, thus necessitating more effort.

4.3 Investigating Performance in Multi-Objective HILO Conditions

Though the workshop-based research demonstrated that the MOBO effectively reduced the workload for designers, it did not fully address CQ 2, particularly with regard to comparing the ultimate user performance yielded by MOBO with that obtained through traditional approaches. Also, the first study did not investigate how MOBO affects creative activities in the design process. Neither did it demonstrate how search behaviors diverge between MOBO and designer-led optimization procedures. Therefore, a user study was designed with the specific aim of addressing these

outstanding questions.

4.3.1 3D Touch Interaction

The study involved a complex interaction, 3D touch, which encompasses parameters related to input and haptic output both (see Figure 4.6). In virtual-reality (VR) or augmented-reality (AR) applications, interactive objects (or targets) are often placed beyond the range within immediate reach of the physical hand. For instance, a user who is browsing a virtual living room might need access to an object situated on the other side of the room. To enable selecting targets at various distances, one can employ a transfer function to translate the physical hand's position to a cursor position. This concept is widely applied in 2D interaction; for example, the transfer function of a mouse maps hand position to the location of the cursor on the monitor. For this study, I chose **the Go-Go technique**, a classic mechanism intended to address precisely this problem, as the base interaction. To bring the problem even closer to off-the-shelf devices, I added a vibrotactile motor supplying tactile feedback. Also, two further parameters were introduced, to control the vibration feedback.

The Background to 3D Touch Interaction
Pointing at a target is an essential and ubiquitous interaction in any VR or AR interface [7]. Hence, extensive research has explored a wide range of pointing and selection methods for VR and AR [25, 180]. As any input interaction should, well-designed 3D selection affords action that is both fast and accurate. Research attests that there is a vast range of control-to-display transfer functions, all with slight differences in their design space [7, 179, 69, 150, 118]. Since searching a high-dimensional design space while evaluating the user's performance is complex and challenging, prior work has relied either on a tremendous amount of trial-and-error [35] or on heuristics [160, 232]. As for the technique chosen for this study, within certain bounds the Go-Go technique [179] follows a 1-to-1 linear mapping, so the virtual hand (i.e., cursor in virtual reality) moves linearly with the physical hand's movements. Exceeding the given threshold, the technique follows a nonlinear mapping; the virtual hand's motion away is scaled quadratically from the physical movements. This mechanism lets users stably select the targets that are relatively close to the body yet also enables reaching those targets that are farther away. The Go-Go technique operates with two parameters.

Design Parameters
The Go-Go technique's two parameters are D and k. The first of these is the threshold for ranges between the linear and nonlinear parts of the transfer function. When the physical hand is within the range D, the

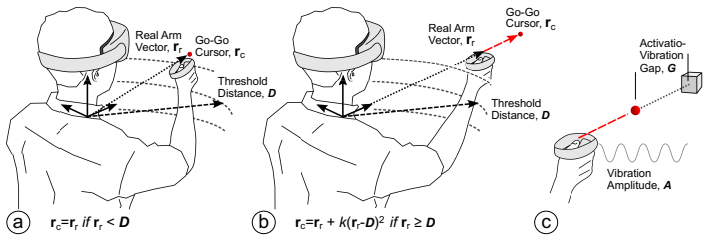


Figure 4.6. A diagram of the 3D touch interaction. (a) illustrates the cursor position when the length of the real arm vector is less than the threshold distance D. (b) showcases the cursor position when the real arm vector is beyond the threshold distance D. (c) shows the vibration feedback. More details are given in Publication II.

transfer function is linear; otherwise, it is quadratic. Secondly, k dictates the scale of the quadratic transfer function. The paper introducing the technique provides a detailed explanation [179]. I retained D and k as the design parameters for the study's 3D touch interaction. In light of the pilot-study results, the parameters were set in these ranges: $D \in [0,1]$ and $k \in [0,0.5]$.

In addition to the transfer function, I introduced tactile feedback to the interaction. To enhance user performance, I strove to offer a tactile signal when the target was reached. Hence, there are two parameters for vibrotactile feedback: the activation–vibration distance, denoted as G, and the vibration amplitude, or A, as shown in Figure 4.6 (pane c). To avoid further complications, I chose a fixed vibration duration of 300 ms. On the basis of the pilot study's results, I set the ranges of these two parameters to $G \in [15cm, -5cm]$ (15 cm before and 5 cm after touching a target) and $A \in [2.6g, 0g]$.

Objective Functions

Similarly to those for general input devices, the two objective functions considered here are connected with efficiency and accuracy. The first addresses completion time, here the average time between the moment at which the user starts moving the cursor and when the selection action is complete. The second objective involves spatial error; the function refers to the maximum overshoot range.

For a roughly commensurate range between these two objective functions, I converted the aforementioned measurements into two metrics: **speed** and **accuracy**. In essence, the completion times and spatial error were normalized linearly to speed and accuracy. I transformed the completion-time range [1,600 ms, 900 ms] into speed range [-1, 1], and I transformed the spatial error range [1 cm, 0 cm] into the accuracy range [-1, 1]. After normalization, 1 is the best value and -1 denotes the worst performance; hence, the problem has become a case of maximization.

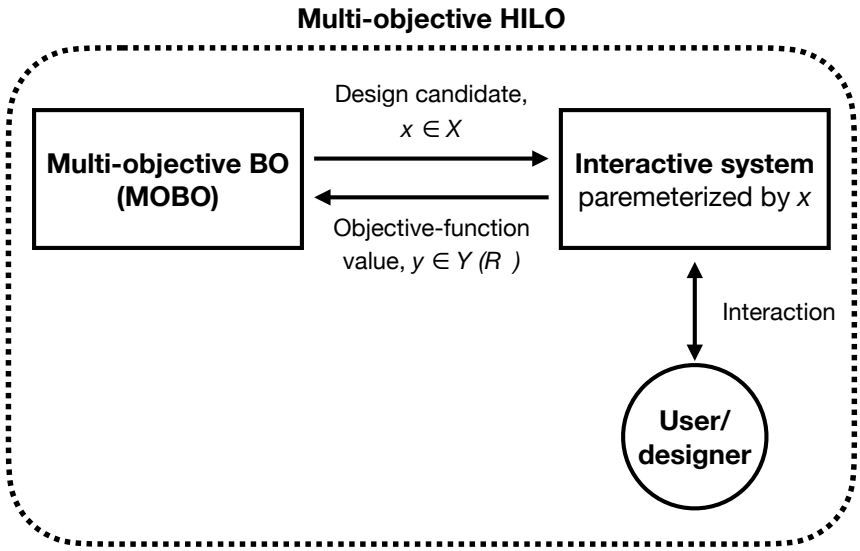


Figure 4.7. In the study, the designer was also the user who observed the final results.

4.3.2 The User Study

As the workshop described above was, this study was carried out to compare two conditions. Where the second study differs markedly from the aforementioned workshop with the MOBO-driven procedure and the designer-led procedure is that there was not a separate role of designer in this design. Instead, the designers themselves were the designs' users (see Figure 4.7). They had to fine-tune the design parameters for maximizing their own performance. In the optimization-led procedure, the study participants engaged in HILO; that is, the optimizer chose how to set the design parameters in each iteration. After the optimization procedure was completed, the participant was presented with a set of Pareto-optimal designs. In the manual procedure's implementation, search was driven by manual exploration; the participant had to decide on the next design candidate, unaided.

The Experiment Design
The study employed a between-subjects design. There was one independent variable with two conditions, involving the optimization procedures introduced above. Each participant was assigned to either the group for the designer-led condition or that for the optimization-driven one. Three sets of measurements were gathered: user performance (completion time and spatial error), perceived creativity (gauged via the Creativity Support Index, or CSI [43]) and workload (probed with the NASA-TLX instrument [85]), and the results of search-behavior analysis (for search distance and

hypercubes visited).

The Apparatus and Prototype
The 3D touch interaction was implemented with Unity 3D[3] and deployed on the Oculus Quest 2[4]. For the vibration motor added to the system,[5] the vibration was driven by a DRV2605L driver board powered by an Arduino Uno microprocessor. The target arrangement primarily followed the lines of previous Fitts' law tasks [38].

To support the participants in the designer-led process, I provided them with a panel of parameter sliders, with which they could easily and intuitively tune the parameters' values. The new value was applied to the interaction immediately, so the participants could try it out without any lag. If they reached a design that they deemed worthy of formal evaluation, they could press a button labeled "evaluation." Doing this initiated a full evaluation of the design, consisting of 36 selections. Once formal evaluation was complete, two charts were shown to the participant to report on the resulting performance.

Procedure
Members of the designer-led group were instructed to tune the design manually and told that they needed to arrive at three optimal designs with the tools offered by the research team. Participants who were instead exposed to the optimization-led condition worked with an optimizer. After the optimization was complete, these participants likewise were asked to pick three final designs from among the Pareto-optimal designs presented. Finally, I evaluated the final performance of the designs developed, in a separate session.

4.3.3 Results

The analysis of the results comprised three elements: examining user performance, assessing the subjective ratings for the experience, and comparing search behavior between the two conditions.

User Performance
As Figure 4.8's pane a shows, for the designer-led and optimization-driven procedure, respectively, the mean completion times were 1,120 ms (sd = 119.4) and 1,185 ms (sd = 97.2), and the mean values for spatial error were 2.2 cm (sd = 1.2) and 1.5 cm (sd = 0.7). From t-tests performed on completion-time and spatial-error data, a significant difference in spatial error ($t(38) = 2.237, p < 0.05$) became evident, indicating that the

[3] Further information available at `https://unity.com/`.
[4] See `https://www.oculus.com`.
[5] The model is presented at `https://www.precisionmicrodrives.com/product/310-117-10mm-vibration-motor-3mm-type`.

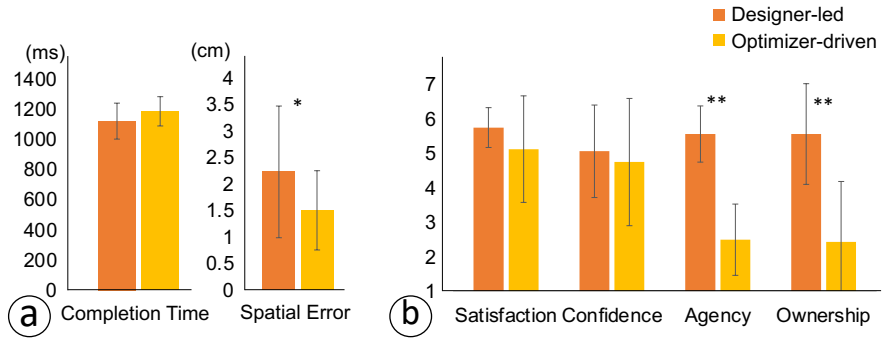


Figure 4.8. Average completion time and spatial error (a) and the ratings of the general experience for its sense of satisfaction, confidence, agency, and ownership. The result indicates that the optimizer-driven procedure derives better user performance, but sacrifices the designer's sense of agency and ownership. More details are presented in Publication II.

optimization-driven approach brought higher accuracy.

Experience and Workload

Figure 4.8, pane b, shows the perceived satisfaction, confidence, agency, and ownership for the two optimization procedures. Further analysis performed with a Mann–Whitney U-test revealed significant differences in the Agency ($t(38) = -5.523, p < 0.001$) and Ownership ($t(38) = -3.892, p < 0.001$) factors. To analyze the level of perceived creativity support further, I examined the responses to the CSI questionnaire. I found a significant difference in the overall CSI score ($t(38) = -2.503, p < 0.05$). For each question, a statistically significant difference in the Expressiveness factor was evident ($t(38) = -3.222, p < 0.001$).

Examining the workload perceived by the participants as probed via the NASA-TLX questionnaire, I found significant differences between the two approaches in both Mental Demand and Effort metrics ($p < 0.05$ for both scales). This sense that the designer-led procedure imposed a greater mental burden is consistent with the takeaway from the workshop discussed in the preceding section of the chapter.

Search Behavior

Lastly, I analyzed the search behavior within the two procedures. The designer-led group took, on average, 51.8 minutes ($sd = 10.0$) to complete the task, while the equivalent figure for the optimizer-driven procedure is 78.0 minutes ($sd = 6.3$). Furthermore, the designer-led procedure included visits to 271 distinct designs, on average ($sd = 192.4$). In all, the participants exposed to that condition tried out 259 designs ($sd = 194.5$) and formally evaluated 12.5 ($sd = 5.5$). Meanwhile, the optimizer-driven procedure visited only 40 designs. It is especially interesting that the designer-led procedure consumed less time for the full process yet visited

6.7 times more design points than the optimizer-led procedure did. This finding demonstrates that humans have a mechanism for rapidly distinguishing between a design that merits full-fledged evaluation and one that does not.

The final quantitative assessment examined how human designers and the optimization system explored the design space, by calculating how many hypercubes were visited. The design space for this case was $[0,1]^4$. We can split each dimension evenly into m levels by means of a division parameter m. This separates the space into m^4 hypercubes of equal size.

I defined a hypercube as "visited" if any set of design parameters fell within the range of said hypercube's bounds. The team conducted t-tests to check for inter-condition differences in the number of hypercubes visited. Analysis identified significant differences for both $m = 2$ ($p = 0.0001$) and $m = 3$ ($p = 0.0019$) segments. This result indicates that the optimization-driven procedure consistently explored more areas of the design space than the designer-led process. This is one explanation for the optimizer achieving better performance; the human designers were not able to explore the design space with as much breadth as the optimizer, so some optimal design(s) may have gone unexplored.

Qualitative Analysis

Six designers stated that they would like to have more agency and the opportunity to express their ideas when designing with the optimizer, especially in cases of disagreement with the design suggestions generated by it. As one participant explained, "I knew what I wanted. I wanted the gap [value] to be reduced, but the AI didn't give me that design." Another participant offered this suggestion: "I wish I can just tell the AI I don't like it [the design]." With similar comments, a designer highlighted the frustration of dealing with "bad" designs proposed by the optimizer – an evaluation "trying out a design that I knew wouldn't work is a waste of time." Generally, the participants were not satisfied with the diminished sense of ownership they experienced in the optimization-driven process. They described feeling as if they had been "working for the AI on those trials," having felt "bored," and finding that this was "not intellectual work."

Irrespective of the shortcomings, optimization did reduce the participant effort consumed in the design process. Participants cited the difficulty of optimizing this interaction themselves, offering as examples "the need to figure out how each parameter works" and "trying to further increase the performance." They also reflected on the challenge of handling two objectives, with comments such as "in fine-tuning, I tended to work on reducing completion time more than spatial errors." Participants assisted by the optimizer felt that they had expended significantly less mental effort: "I feel relaxed as the AI is doing the design part."

4.4 The Work in Summary

In response to previous human-in-the-loop optimization work, which focused mainly on tasks with only one objective, this chapter lays out how to extend HILO from a single objective to multiple objectives. To this end, I raised and tackled two questions: how to enable multi-objective HILO (CQ 1) and what qualities characterize performing multi-objective HILO (CQ 2).

To respond to the former, I proposed applying Pareto-front learning to achieve multi-objective HILO. Instead of searching for a single design that yields the best user performance, Pareto-front learning identifies a series of designs that lead to the Pareto-optimal performance levels, representing the best tradeoffs of multiple objectives. Although publications in the machine-learning field have proposed various implementations of MOBO and these have been evaluated by means of test functions, no prior work has implemented multi-objective optimization in a human-in-the-loop setting. Filling this gap, two of the dissertation's component publications demonstrate the efficacy of Pareto-front learning: Publication I describes a design workshop for the task of designing a set of vibration cues, which highlighted the designer-perceived workload and the range of design strategies exhibited. Publication II presents an empirical study conducted in the context of designing a 3D touch interaction. That study focused more specifically on examining the ultimate performance, perceived creativity support, and the search patterns.

Together, the two publications answer CQ 2. Analysis showed that the designers taking part experienced **significantly less effort and a lower workload with MOBO** than with manual (or designer-led) procedures. There are several reasons. Firstly, they did not need to come up with strategies to tackle the complex design challenge. As participants reported, exploring a large design space on their own was mentally taxing. Furthermore, they had to balance multiple objectives, which further complicates matters. In contrast, MOBO offers extensive flexibility for dynamically picking a final design on the Pareto frontier. Performance analysis revealed that MOBO yielded performance at least comparable to that obtained by means of the designer-led procedure. This can be attributed to MOBO's ability to search a wider range of areas in the design space in a principled manner, thereby generating a greater variety of designs.

However, the workshop and follow-on study also illuminated **the designer-led procedure's greater sense of agency and ownership**, stemming from the designer's direct control over the exploration; participants articulated **greater perceived expressiveness** too. Participating designers recommended a mixed design method in which the human designer could rely on optimization for effectively exploring the design space but take control as necessary (e.g., skipping bad designs). I would encourage future

work exploring this direction. It is worth stressing the widespread applicability that such advances afford: the avenues presented are suitable, for instance, for single-objective cases as well as the multi-objective scenarios examined here.

The two publications described in this chapter jointly inform the dissertation's first and second contribution: **multi-objective human-in-the-loop design optimization via Pareto-frontier learning** and **investigation of the perceived qualities of multi-objective HILO**, respectively.

5. From Individual to Population

"Design must always consider the needs of the mass, the majority, the group - not just the individual - for it is in satisfying the needs of the group that the needs of the individual are satisfied." — Don Norman

Following on from the introduction of the method enabling multi-objective HILO, this chapter speaks to the next aim: extending HILO to reach optimization beyond the individual user. In design practice, designers often seek to address the needs of a group or population rather than an individual user. Two major issues arise in shifting from optimization for individuals to group-level optimization.

The first of them is **transferability**: Can a design optimized for a single user carry over to other users? This depends greatly on the interaction and on behavioral diversity across users. If all users show extensive similarity in their preferences and performance, the optimized design may be transferable; that is, a design that is "good" for one user should work well for others. However, if there are considerable variations among users, the optimal designs for two users are likely to be different. In that case, the optimal design cannot be transferred non-problematically, and a design that excels for one person might serve another user poorly. The methods encountered thus far in the field have not been able to address this problem. Therefore, there is a need to identify and hone a method able to generate design instances that are optimal for a group/population – i.e., the best-compromise designs for all individuals involved.

Another issue that has to be addressed is the **time-efficiency** of the optimization process. Previous studies have shown that a full-fledged optimization procedure can take an hour or even more to identify the Pareto-optimal designs for an individual. The issue becomes even more pressing in cases of design tasks that involve larger numbers of parameters or objectives, which require more iterations and time before convergence. Such lengthy optimization procedures are impractical for end users, who cannot afford to spend an extended period of time on finding an optimal setting before starting to use the interaction technique. The reason for

the protracted nature of the optimization is that BO's learning typically does not utilize any prior information. One possible way to boost efficiency is to take advantage of previously gathered optimization data to form an informative prior. The practicality of this suggestion has remained unclear, though, with its further investigation posing a fundamental challenge. Therefore, the research presented in this chapter pursued a time-efficient optimization method that exploits group-level data.

With this chapter, I aim to answer two central questions arising from the issues described above.

CQ 1: How could we derive the optimized design for a group of users?

CQ 2: How could we generate an initial model that can efficiently adapt to an individual user by making use of group-level optimization data?

5.1 Group-Level Optimization

My response to these questions takes the form of a novel approach referred to as group-level human-in-the-loop optimization. Group-level optimization differs from typical HILO in that the optimization process does not simply consider an individual user's data; it works with data from a group of users. To achieve this, I have introduced two extensions to the HILO framework. The first of these, **Global GP**, involves a model of aggregated group-level data that can be inspected to identify Pareto-optimal designs for the whole group. The second extension is **Warm-Start GP**. Its model relies on a set of optimization data for initialization that affords efficient design customization for new users. Table 5.1 presents a fuller comparison between standard BO and my group-level approach.

5.1.1 Procedure of Group-Level Optimization

As given earlier, the central idea for these two extensions is to aggregate the data gathered from the optimization of a population user group. Different from a standard HILO where an individual user simply goes through an optimization process on their own, group-level HILO has a different procedure:

1. Running individual optimizations on a group of users;

2. Constructing the group-level models (the Global GP or the Warm-Start GP). If the goal is group-level optimization, use the Global GP to generate the group-level optimal designs.

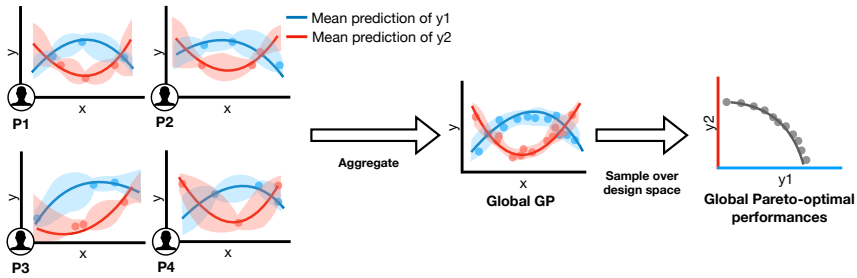


Figure 5.1. Illustration of the mechanism of the Global GP.

3. Deploying the derived design (Global GP) or the derived Wram-start GP on the new users.

The proposed group-level HILO extensions support both single-objective and multi-objective purposes. However, it is required to be consistent throughout the steps. The group-level surrogate models should share the same number of objective functions as in the deployment step. In the following content, I only focus on multi-objective cases.

5.1.2 The Global GP: Deriving Group-Level Optimal Designs

The first extension, Global GP, is intended to address CQ 1, which involves deriving the Pareto-optimal designs for a group of users. Once I have constructed a global model, supplying all observed design parameters and objective values from all participants so as to form a single GP model, we can query that GP model to obtain the predicted mean and variance of a given design. This prediction can be viewed as the group-level expected performance for a given design instance. After constructing the model, I derive the optimal design instances by making predictions via the group-level surrogate model. To identify the Pareto-optimal designs generated by Global GP, I conduct a fine-grid search of the entire design space. Assigning c equally spaced points along each dimension gives us c^n points in total. We can query Global GP for all of these c^n points and store all the output. Finally, I identify the Pareto-optimal designs from among these c^n predictions. For the experiments reported upon in this chapter, the research team chose $c = 16$.

This method is computationally expensive but since it runs as a post-processing step, it is practical and feasible to perform and provides a comprehensive summary of the optimal design parameter sets. Figure 5.1 illustrates the high-level process of constructing the Global GP and extracting the Pareto-optimal designs.

Table 5.1. The differences between standard Bayesian optimization and Group-in-the-loop optimization.

Features	**Standard HILO**	**Group-level HILO**
Objective	Best design for an individual	Best design or model initialization for a group
Best design	Maximum given by the GP	Best compromise among users (Global GP)
Model initialization	Learn from scratch	Exploit previous users' data (Warm-Start GP)
User sample	Flexibly defined	Group specified in advance

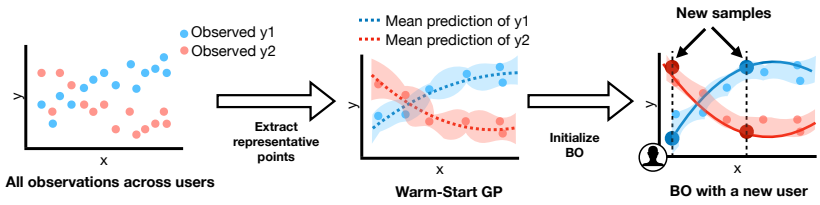


Figure 5.2. Illustration of the mechanism of the Warm-Start GP.

5.1.3 Warm-Start GP: A Rapidly Adapting Surrogate Model

The other novel extension responds to CQ 2, which pertains to the time-efficiency of HILO. Standard HILO usually begins without any informative prior distribution. This leads to random exploration of design options in the early stages, sometimes called the "cold-start" problem. To overcome this issue, I explored creating a "warm-start" GP model to serve the initial stages. We can implement the core principle (i.e., exploiting previously collected optimization data to form an informative distribution that captures the interaction characteristics) in practice by carefully selecting a subset of data points from the individual-level optimization processes and fitting them to our Warm-Start GP model. This SM can provide useful prior knowledge for efficient optimization in subsequent individual-level optimization processes. Figure 5.2 illustrates the general procedure of using Warm-Start GP in HILO.

For the most part, the extension follows the steps implemented in **sparse GP**. We adapted the method proposed by Titsias [212]; they applied an approximation of the marginal likelihood for the entire dataset with a subset of size K in accordance with the work of Seeger et al. [196]. The main idea here is to pick the most "informative" K data points by adding one training point from the full dataset at a time with the aim of maximizing the approximate marginal likelihood over the complete dataset. To reduce the computation complexity, some heuristics are introduced: Instead of directly including the full dataset as initial candidates for the sparse subset, the procedure considers only a reduced subset of candidates; such a subset is sampled randomly. In our application (detailed further along), that randomly selected subset had 100 data points; i.e., it was half the size of the full 200-data-point set. Within this reduced dataset, the likelihood-maximization process functions for greedy selection of the candidate point for the sparse subset of size K, which was set to be 5.

In summary, our method is designed to generate a warm-start prior that can adapt to a new user and, thereby, obtain personalized optimal design instances sample-efficiently. Our method follows sparse Gaussian Processes [17, 34, 32] to obtain the representative data points from the full dataset.

Table 5.2. The parameters of the simulation tasks: the corresponding objective functions of each parameter and the parameters' ranges.

Parameters	Corresponding Objective Functions	Range
x_1	Branin (y_1)	[0, 5]
x_2	Branin (y_1), Sphere (y_2)	[0, 5]
x_3	Sphere (y_2)	[0, 5]
x_3	Sphere (y_2)	[0, 5]

5.2 Evaluating Group-Level Optimization via Simulations

While our methods provide a means to obtain group-level optimal designs and initializations, they assume the users in a group to have some shared characteristics. Different levels of similarities within the group affect the performances of the proposed methods. Before conducting the experiment with a human-in-the-loop setup, I validate the Global GP's and the Warm-Start GP's performances when facing various diversity levels using simulations. Furthermore, I compare these performances to the conditions that are without our enhancements. The results showed that when the functions within a group are relatively similar, our methods can effectively achieve higher objective performance and more efficient adaptation. Even when the functions are quite diverse, our methods still provide marginal benefits or comparable performances to the baseline.

5.2.1 Test Functions

All the following simulations share a base function; I shifted the base function differently to simulate different users. The base function has four parameters and two objective functions. We adopted two widely used test functions: Branin[1] (2-dimensional input) and Sphere[2] (3-dimensional input) as our objective functions. The first two parameters (x_1, x_2) contribute to the value of the Branin function, while the last three parameters (x_2, x_3, x_4) contribute to the value of the Sphere function. Note that parameter x_2 contributes to both the Branin and the Sphere values. This was deliberately set to create a trade-off scenario: there is not an x_2 value that can maximize two functions at the same time, and hence, MOBO must search for the Pareto frontier rather than an optimal design. We normalized these two functions so that all the parameter values are bounded in the range of $[0, 5]$ and all the objective values are bounded in the range of $[-1.1, 1.1]$. Further details of the parameters and the objective functions can be found in Table 5.2 and Table 5.3.

[1] https://www.sfu.ca/~ssurjano/branin.html

[2] https://www.sfu.ca/~ssurjano/spheref.html

Table 5.3. The simulation task's objective functions and their ranges.

Objective functions	Name	Range
y_1	Branin	[-1.1, 1]
y_2	Sphere	[-1.1, 1.1]

5.2.2 Setting up Simulations

To simulate the differences between the users, I need to generate a set of functions that have different Pareto-optimal parameter values. We achieve that by randomly shifting the base function. Each shifted function can be seen as a unique user because it has its own set of Pareto-optimal design parameter values. The shifting happens in all the parameters (x_1, x_2, x_3, x_4), and the amount of the shifting is sampled from a uniform distribution of certain sample ranges. Specifically, the shifting is denoted as $x'_n = x_n + \delta$. x_n is the original parameter value, x'_n is the new parameter value after shifting, $n \in [1,4]$, and $\delta \sim U(-\frac{range}{2}, \frac{range}{2})$. If the sample range is big, the shifts of this group will be farther away from each other and result in more diverse functions. If the range is small, the resulting shifts and functions will be more similar.

We then simulate various groups of users by varying the sample ranges. We have 5 groups of functions (users), whose sample ranges are 0.1, 1, 2, 3, and 4, respectively. The smallest sample range (0.1) naturally results in a group of functions that are highly similar to each other. The largest sample range (4) leads to a group of high-diversity functions.

For each sample range (i.e., 0.1, 1, 2, 3, and 4), I randomly generated 20 unique functions (users). 10 of them were used to run individual MOBO, and the optimization data was used to generate the Global GP and the Warm-Start GP; I named this group of functions "the GP groups". The remaining 10 functions (users) are then used to evaluate the Global GP and Warm-Start GP; I named this group "the evaluation groups". For running the individual optimization, MOBO is set to have 3 initial random samplings followed by 10 optimization iterations.

To evaluate the Global GP, I first constructed the Global GP, and then performed a grid-search over the whole design space to identify the design that has the highest average of the two objectives. We evaluated the obtained parameter sets within the evaluation groups. Then, I created a baseline group, which determined the parameter values randomly.

To evaluate the Warm-Start GP, I derive 10 sparse GP points from the GP groups' data. Then I ran 5 iterations on the evaluation groups with the Warm-Start GP and recorded the hypervolume. The baseline group is running MOBO from scratch with 2 initial samplings and 3 optimization iterations.

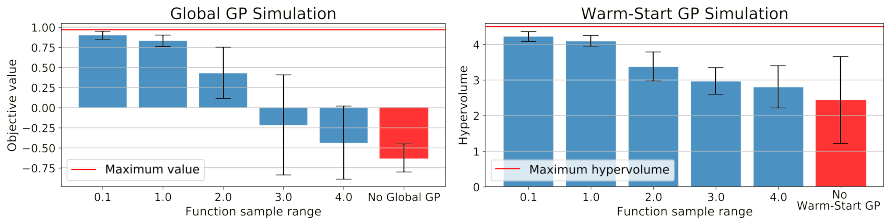


Figure 5.3. The Global GP aggregates all observations from the user-specific optimization processes. The consolidated model can thereby estimate the group's average performance at any given design parameter value. After constructing the Global GP, I perform a fine-grid sampling of the design space to identify the global Pareto-optimal design instances.

5.2.3 Simulation Results

The simulation results are shown in Figure 5.3. Both Global GP and Warm-Start GP resulted in promising performances when the sample ranges are small (e.g., groups 0.1, 1, and 2) This indicates that our methods would bring advantages to the users if the group has a certain level of similarity among users, which is similar to our findings in the following user studies. Even when the groups share only little similarity (e.g., groups 3 and 4), the final performances are slightly better or comparable to the baseline conditions. This suggests that our methods will at least provide marginal benefits or deliver similar performance to standard Bayesian optimization. The results further suggest that group-level optimization can be applied when the users may share certain characteristics.

With these positive simulation results, I then conducted the two studies, as shown in the following sections. These studies showed that both the Global GP and the Warm-Start GP lead to significantly better performances than baselines, which confirms that the users have some shared characteristics in the selected interactions. Further investigation is needed to understand the levels of similarities among users within different interactions. If there is an interaction where the users have no similarities at all, it is naturally impossible to derive a group-optimized design regardless of the methods because such a group-optimized design simply does not exist. Under such cases, only individual optimization will be suitable to address the problem, which requires a longer optimization duration.

5.3 Designing with Global GP

The first user study validated the Global GP method for deriving a set of Pareto-optimal designs that represent a group of users. The study involved three steps: A series of individual-level optimizations was conducted, resulting in observations of <design instance, objective values> pairs. Secondly, a Global GP model was constructed with all the data

gathered, and a fine-resolution grid search was performed to derive the group-level Pareto-optimal designs. Finally, the Pareto-optimal designs were evaluated with a group of users and compared with the baseline condition. The results showed that the proposed technique produced better user performance.

5.3.1 Group-Level Optimized 3D Touch

The target interaction was the Go-Go technique [179], and the setting of the interaction was the same as that described in subsection 4.3.1.

5.3.2 The User Study

The three steps employed for the user study generally followed the description in subsection 5.1.1. Firstly, I ran individual-level optimizations for all users in a given group. After that, I constructed a Global GP model with the data obtained from those optimizations and, on that basis, derived the group-level Pareto-optimal design. Finally, the group-level optimal designs were evaluated with users. The design parameters and objective function remained the same as outlined in subsection 4.3.1.

Participants
The research team recruited 20 participants for the study and assigned them at random to two groups. The first group participated in every step of the study, and I subsequently labeled it the "experienced" group accordingly. The second group, involved in only the final step, was labeled as the "novice" group.

Apparatus
The apparatus was similar to what is presented in subsection 4.3.1. However, since the participants were not required to select design instances manually, all relevant interfaces related to manual optimization were absent.

Procedure
Step 1 (user-specific optimization): The first step consisted of individual-level optimization for all members of the experienced group. There were 40 iterations in total. Each iteration provided a pair of <design-parameter values, objective-function values>. The full procedure took 90 minutes per participant. From Step 1, 400 data points (10 participants × 40 data points) were gathered.

Step 2 (Global GP and group-level optimal designs): We constructed the Global GP model with all data points gathered in Step 1. Then, a fine-grained grid search over the whole design space was performed. Each design parameter was discretized evenly to 16 points, resulting in 16^4 de-

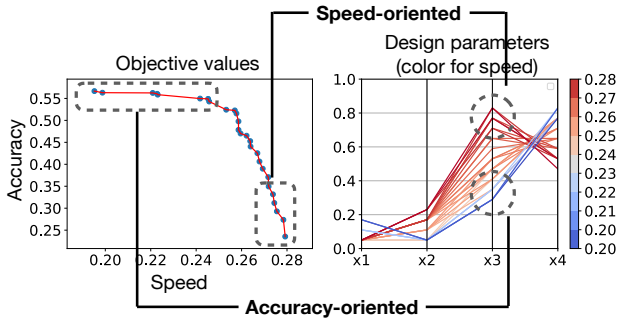


Figure 5.4. The predicted Pareto-optimal objective values from Global GP and the global Pareto-optimal designs.

sign instances. Global GP made predictions for all 16^4 instances. Lastly, I derived the set of Pareto-optimal designs within this set of 16^4 points. The final set of Pareto-optimal global designs is presented in Figure 5.4. These global designs were subdivided into two subsidiary sets: a speed-oriented subset and an accuracy-oriented subset. The speed-oriented designs prioritized efficiency (completion time) over accuracy (spatial errors), with the accuracy-oriented designs' priorities aligned the other way around. We synthesized the results into two final designs, one speed-oriented and the other accuracy-oriented, by averaging over all parameter values from these subsets.

Step 3 (evaluation of the group-level optimal designs): The goal for the final phase of the study was to evaluate the performance of the group-level optimal designs against the baseline Go-Go technique configuration. To investigate the efficacy of the global-level optimal design thoroughly, I evaluated the designs with the aforementioned two groups of users: the experienced group (included in the first step) from which initial data were obtained, and the novice group (completely naïve to the interaction). The evaluation setting was structured as a 3 (designs) × 2 (groups of participants) mixed-design experiment with two independent variables. Each participant tested all of the design instances, so the design factor was within-subject. The participant group was a between-subjects factor.

Note that the Go-Go technique as originally conceived does not feature vibration feedback. For a fair comparison, I augmented the technique by issuing a vibration cue as the user contacts the target with the virtual cursor ($x_3 = 0$ cm). This is a common default setting in many VR interactions. The vibration amplitude for the Go-Go technique (x_4) was set to 1 g, the amplitude that was most strongly preferred in pilot testing. With this setting, there are three conditions to be evaluated; Table 5.4 outlines them.

Table 5.4. The three design conditions evaluated in Phase 2.

Condition	x_1	x_2	x_3	x_4
Speed-oriented	0.05	0.098	5.77 cm	2 g
Accuracy-oriented	0.092	0.037	10.76 cm	0.91 g
Go-Go Technique	0.667	0.167	0 cm	1 g

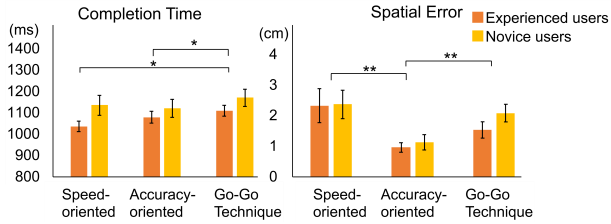


Figure 5.5. Results of the comparative study on three designs. The result indicates that the designs generated by the Global GP outperform the Go-Go Technique in completion time. Further, the accuracy-oriented design also outperforms the other designs. Please refer to Publication 3 for further details.

5.3.3 Results

The mean completion time and spatial error for each condition and participant group are presented in Figure 5.5. The experienced users' mean completion time for the speed-oriented design, the accuracy-oriented design, and the Go-Go technique were 1,038 ms ($sd = 77.64$), 1,081 ms ($sd = 87.72$), and 1,111 ms ($sd = 80.32$), respectively. Novice users' corresponding mean completion times for the three conditions, respectively, were 1,124 ms ($sd = 126.28$), 1,117 ms ($sd = 127.10$), and 1,167 ms ($sd = 120.88$). As for the mean spatial error, the experienced users' averages for the speed-oriented design, accuracy-oriented design, and Go-Go technique were 2.34 cm ($sd = 1.76$), 0.97 cm ($sd = 0.49$), and 1.55 cm ($sd = 0.85$), respectively. The equivalent figures for the mean spatial error of the novice users were 2.03 cm ($sd = 1.17$), 0.96 cm ($sd = 0.68$), and 1.83 cm ($sd = 0.78$), respectively.

Mixed-design ANOVA was performed to examine the effect of the interfaces and user-experience levels. Significant within-subject effects were found for both completion time ($F(2,36) = 7.483, p < 0.005$) and spatial errors ($F(1.432,25.781) = 19.284, p < 0.001$). Analysis of the between-subjects effects found no differences connected with user-experience levels (all $p > 0.05$). Examining completion time showed that both the speed-oriented and the accuracy-oriented design outperformed the Go-Go one (all $p < 0.05$). In the analysis of spatial errors, the accuracy-oriented design emerged as significantly better than the other designs (all $p < 0.001$); however, the speed-oriented design did not prove useful for reducing spatial errors. Both group-level optimal designs can be characterized as yielding better or

comparable performance levels.

It is important to note that, in this project, the presented approach aggregated all the previously gathered data points into one unified GP model. Future work should also consider more advanced ways of aggregating data and generating group-level designs. One direction is to store each user's data as separate surrogate models, which will then lead to a group of GP models. Then, for each particular design instance, we can generate a group of predictions, one derived from a single model. With this extension, future work can further derive diverse kinds of global-level optimal designs, such as "best-case" (prioritizing the optimization for the expert user group with the best performance), "worst-case" (prioritizing the optimization for the novice user group with the worst performance), and "average-case" (prioritizing the optimization for the average user group) designs.

5.4 Designing with the Warm-Start GP

Another study was conducted to validate the efficacy of the Warm-Start GP extension by a touch-button design task. Similar to the previous study, there were three major steps. First, I conducted individual optimizations on a group of users. With the gathered observations, I derived the Warm-Start GP via the method described earlier. Finally, I assess the effectiveness of the Warm-Start by comparing it to the standard BO process.

5.4.1 The Fast-Adaptation Touch-Button

The design task involved touch-button-pressing, which is a fundamental mechanism of interaction with touch-sensitive devices. Interacting in this manner involves the user's finger and a button on a touchscreen. Contact triggers a function attached to the button, alongside generation of a key-click vibration to inform the user of the activation. In order to provide context for considering this interaction, the subsections below discuss its design parameterization and target objective functions.

The Background to Touch-Button Interaction
Research has revealed that the optimal point for triggering a button is not at the exact moment when the finger makes contact with the button surface [134, 115] but somewhere within its travel range. Additionally, there has been extensive research into the design of haptics for touch-buttons, various aspects of which have been proven to influence the user's typing speed on a soft keyboard [236, 177]. Designing appropriate haptic feedback for target selection is complex, in that the feedback designs can lead to widely varying perceived sensations and user performance. For instance, the ideal moment for vibration might not be identical to that for

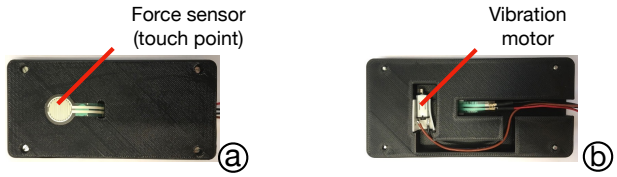


Figure 5.6. The smartphone prototype with a touch-sensitive button. (a) The smartphone prototype and the touch point. (b) The vibration motor is placed inside the prototype for emitting vibration feedback. Please refer to Publication III for more details.

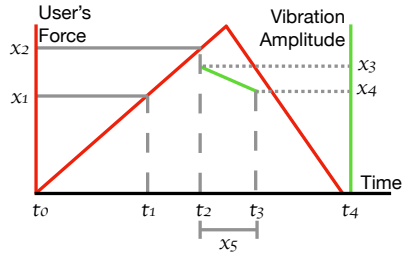


Figure 5.7. Illustrative design example of the target interaction. Please refer to Publication III for more details.

activation (as Figure 5.7 illustrates).

While previous research has demonstrated optimization of button design with a single objective [41, 189, 132, 236], the relevant efforts were not conducted with a human-in-the-loop setup or the results served only one specific objective. In contrast, the aim of this study was to derive a fast-adaptation model of a touch-button for **temporal pointing** interaction. Temporal pointing tasks require users to provide certain discrete inputs within a narrow time window [127]; for instance, they might have to activate a function at a particular moment in a game or synchronize input experiences in day-to-day use [224].

Design Parameters

The study includes five design parameters, shown in Table 5.5 and Figure 5.7. The button-activation threshold (x_1) is the force level that activates the touch-button, and the vibration threshold (x_2) is the force level that triggers the device to emit the vibration signal. The rest of the parameters were added to create a rich variety of haptic cues. Initial vibration amplitude (x_3) and final vibration amplitude (x_4) dictate the amplitude of the vibrations, and x_5 determines the vibration's duration. If x_3 and x_4 are not identical, the vibration is set to linearly increase or decrease over the span of x_5.

Table 5.5. Design parameterization of the touch-button.

Design Parameter		Range
x_1:	Button activation force level	[15 g, 1515 g]
x_2:	Vibration activation force level	[15 g, 1515 g]
x_3:	Initial vibration amplitude	[0 g, 3.2 g]
x_4:	Final vibration amplitude	[0 g, 3.2 g]
x_5:	Vibration duration	[0 s, 1.5 s]

Table 5.6. The design objectives of the touch-button.

Objective	Description
Temporal Error Mean	The temporal pointing is more accurate if this value is smaller.
Temporal Error Standard Deviation	The temporal pointing is more precise if this value is smaller.
Subjective User Rating	The vibration cue matches the click interaction more if this value is higher. Values from 0 to 100.

Objective Functions
Three objective functions considered the temporal performance and the user's subjective rating. Two separate objective functions were measured regarding the temporal performance: the mean value of temporal errors and the standard deviation of the temporal errors. The participant's subjective rating was provided by the participants after one iteration was complete. The three objectives are detailed in Table 5.6.

5.4.2 The User Study

With this study, I sought to compare efficacy between Warm-Start GP and the baseline (standard BO), with the design task described above. The results indicate that optimization efficiency indeed improves significantly when Warm-Start GP provides a starting point.

Participants
In total, 22 people were recruited for the study. The participants were randomly allocated to two groups thus: Ten participants completed all phases of the study. In other words, their data were used to generate the Warm-Start GP model, and they also were involved in the final evaluation phase. I refer to this group of users as the "experienced user group." The remaining 12 participants took part in only the final evaluation, so their data did not inform construction of the Warm-Start GP model. I denote this group as the "novice user group."

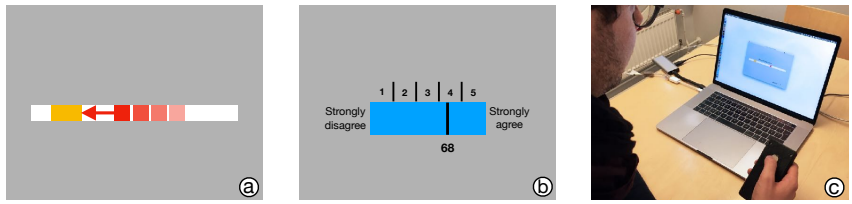


Figure 5.8. A simplified sketch of the study interface during button-pressing (a): the participant is asked to activate the button when the red bullet reaches the yellow target area, at which point the bullet turns blue. After 24 presses, the user is asked to rate the vibration cue (b). Pane c shows the study's interaction in action.

Apparatus

I implemented a touch-sensitive prototype device (6 cm × 12.5 cm × 1 cm). The most important parts of this prototype were a force-sensing resistor (FSR 402[3]) and a vibration motor (Precision Microdrives 308-102[4]). The device is shown in Figure 5.6. Its vibration motor was controlled by a driver (SparkFun DRV2605L[5]) and an Arduino microprocessor. The study interface, implemented in Processing, is presented in Figure 5.8, panes a and b.

General Tasks

As in the previous study, there were three main steps. The first and the third involved optimizations that required participants' involvement; they were instructed to perform a temporal pointing task. The leftmost pane of Figure 5.8 depicts the interface. A red "moving bullet" flies from the right side of the interface to the left side, along the bar. Participants were instructed to "activate the button when the bullet reaches the center of the yellow target zone." The red bullet turns blue when the button is activated, to notify the participant about its status.

In each iteration, participants interacted with the system at two levels of difficulty: "Easy" (with the bullet's movement speed set to 625 pixels/second) and "Hard" (with a movement rate of 1,000 pixels/second). Users encountered the two difficulty levels in random order. At each difficulty level, the participant performed 12 button presses.

At the end of each iteration, the participants rated the statement "The vibration cue synchronizes (matches) with the button pressing interaction," which was displayed via the interface. This statement was presented with the scale depicted in Figure 5.8's pane b: each participant gave a subjective rating from 0 ("Strongly disagree") to 100 ("Strongly agree").

[3] Details are available at `https://www.interlinkelectronics.com/fsr-402`.

[4] Presented at `https://www.precisionmicrodrives.com/product/308-102-8mm-vibration-motor-15mm-type`.

[5] See `https://www.sparkfun.com/products/14538`.

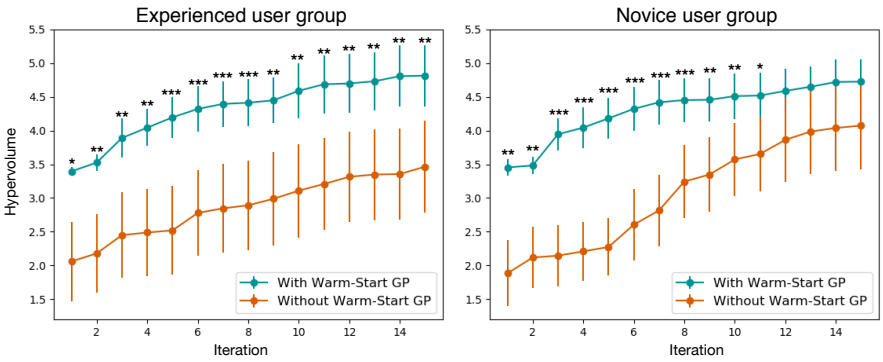


Figure 5.9. The results from the evaluation of Warm-Start GP. More details are provided in Publication III.

Procedure

Step 1 (user-specific optimization): Participants were asked to complete the standard MOBO procedure, and Step 2 – generation of the warm-start model – applied the data collected from all these participants: 500 observations (50 design instances × 10 participants).

Step 2 (deriving the Warm-Start GP model): The research team applied the Warm-Start GP approach to derive a subset of K data points for forming a Warm-Start initialization model.

The size of K strongly affects the efficacy of the adaptation: if there are too few points, the prior would not be helpful in the later optimization, yet were there too many points to form a prior, initialization would start with a very large initial hypervolume. This is potentially problematic since the data points selected might completely dominate new observations in the next step, in which case there would not be any adaptation at all. To pick a reasonable number of points, I created three warm-start models, with 5, 10, and 15 initial data points, and I then compared their performance by means of a simulation. The results showed that using five Warm-Start data points offers the best setting – it resulted in the greatest hypervolume increase within 15 iterations.

Step 3 (evaluation of Warm-Start GP): Both groups of users, the experienced user group and the novice user group, were involved in Step 3 of the study. Both sets of participants were exposed to two conditions, the standard BO and the BO with Warm-Start GP as the initial model. The tasks given to the participants were the same as in Step 1.

5.4.3 Results

Figure 5.9 plots the hypervolume increases produced in both conditions. Two-way repeated-measures ANOVAs were run to analyze the effect of two independent variables – **initialization** (with or without Warm-Start GP)

and **iterations** – on the hypervolume increase for both user groups.

For the experienced user group, there was no significant interaction between the effect of initialization and that of iterations ($F(14,126) = 0.38$, $p > 0.05$). Simple main-effects analysis showed that the hypervolume was significantly larger with Warm-Start GP than without it ($F(1,9) = 23.43$, $p < 0.001$). This analysis also revealed significant differences between iterations for the experienced user group ($F(14,126) = 31.88,\ p < 0.001$). Pairwise T-tests were conducted to compare the hypervolume between the two initialization conditions at every iteration. Significant differences evident across all iterations (all $p < 0.05$) indicate that Warm-Start GP initialization led to a consistently larger hypervolume.

For the novice user group, a significant interaction between the effect of initialization and of iterations ($F(14,154) = 8.25,\ p < 0.001$) was found. Simple main-effects analysis showed the hypervolume to be significantly larger when the procedure started with Warm-Start GP than without it ($F(1,11) = 19.24$, $p < 0.001$). Also, there was a significant effect for iterations ($F(14,154) = 32.17,\ p < 0.001$). Pairwise T-tests were run to compare the hypervolume between the two initialization conditions at every iteration. Warm-Start GP produced a significantly larger hypervolume from the first to the 11th iteration (all $p < 0.05$). This is evidence that Warm-Start GP effectively supported faster adaptation for the novice user group too.

5.5 The Work in Summary

For this chapter, I explored another important direction of HILO, **extending it to support group-level optimization**. To this end, I specified two important goals: 1) group-level optimal designs and 2) a group-level rapidly adapting the initial model. For reaching these goals, I propose methods that gather optimization data from a group of users and aggregate said data to derive designs or models.

The first method presented, Global GP, involves fitting all observations from user-specific optimizations into a single GP model to form a group-level representative model, after which a grid search is conducted to identify group-level Pareto-optimal design instances. I was able to demonstrate this method's efficacy for 3D touch interaction, wherein the Pareto-optimal design resulted in better user performance than the baseline condition. The second method, Warm-Start GP, uses a subset of data from user-specific optimizations to create an SM that serves as a more efficient initial model for the optimization process. My evaluation of this method with a touch-button design task showed the hypervolume increase to be significantly greater with Warm-Start GP than in its absence.

These methods extend HILO from single-objective cases to multi-objective cases and expands the scope of optimization from a single user to a group of

users. Thereby, they enable HILO for more realistic design problems. That outcome constitutes the third contribution identified for this dissertation, **group-level human-in-the-loop optimization**.

6. From Physical Prototyping to Emulation

"If a picture is worth a thousand words, a prototype is worth a thousand meetings."
— IDEO.org

In this chapter, I address another significant challenge of HILO – namely, the high cost of physical prototyping. For a better understanding of this challenge, a review of the typical design processes is in order. Prototyping is a crucial step in most design processes. In other words, prototyping involves transforming a design idea into a physical representation for user testing and evaluation.

I can consider prototyping in terms of two classes of activity: physical prototyping and non-physical prototyping. Many design prototypes do not involve a physical prototype; for instance, one can usually render various design instances for graphical user interfaces in real-time without any physical constructs. The preceding chapters demonstrate how HILO can tackle such applications; e.g., the optimizer can change the properties of the interaction instantly. In contrast, physical prototyping is significantly more costly and time-consuming. Creating a physical prototype involves transforming a design idea into a complete 3D model, then investing time in building the prototype. Soldering the circuit, assembling the digital elements into a working device, etc. requires substantial effort and can take hours to days. On account of the high cost of physical prototyping, it is impractical to conduct HILO for interaction conditions that require fabrication. These interactions have remained within reach only for traditional design processes such as UCD, and design exploration has been limited to a small portion of the design space.

The main research question addressed in this chapter is **how to facilitate HILO in design tasks that currently require physical prototyping (CQ 1)**. To address this challenge, I propose the use of **physical emulation**, a software and/or hardware system able to simulate the responses of a physical device or system. Physical emulation eliminates the need for fabricating a physical prototype at each iteration, thus significantly reducing the financial outlay and time required for prototyping. I

begin the discussion with background information on physical interaction specifically with buttons, physical prototyping, and emulation in HCI research. Then, taking push-button interaction as an example, I demonstrate how physical emulation can function as a cost-effective tool for HILO.

6.1 Background

Physical Buttons (Push-Buttons)
Physical buttons are devices that translate mechanical force into an electrical signal. This chapter focuses specifically on push-buttons as used in keyboards or key panels. Design parameters such as the physical properties of the keycap (width, angle, and key depth) and the materials used (e.g., plastics) are among the many factors influencing the button's haptic and tactile characteristics, commonly known as **tactility** [110, 143]. While tactility is crucial for the typing experience and performance of professional gamers, programmers, typists, and hobbyists alike [131, 2, 49, 174], designing push-buttons can be challenging, because of the high cost of creating the physical prototypes needed in testing their tactility, for which the mechanical structure is vital. This chapter presents physical emulation as a possible solution. It holds potential for enabling human-in-the-loop optimization of push-button design.

Physical Prototyping
Prototyping plays a crucial role in the HCI and design fields, enabling designers to express their ideas, compare design instances, and test hypotheses [221]. Accordingly, research has delved into the functionality of prototypes [90] and the experience of working with them [31], and advances in technology have enabled the emergence of rapid prototyping as a tool for designers' communication and demonstration of sophisticated concepts [228]. Techniques such as 3D printing [200] and laser cutting [109] have made it possible for designers to fabricate physical forms at a lower cost, while microprocessors and IDEs [6] have made it easier for engineers and developers to work with circuits, sensors, and actuators, thereby facilitating work with fully functional prototypes and even "personal fabrication" [16].

Traditionally, designing a new push-button requires fabrication in full, inclusive of internal circuits and mechanical parts (spring and structure). Despite the relevance of push-buttons in HCI, the discipline has devoted very little effort to physical prototyping of buttons, with one possible reason being the complexity of the prototyping process. The most relevant research in this area has focused on prototyping devices that deliver "passive haptics." For example, Lin et al. [136] used planar-compliant structures to

create various types of passive haptic feedback when an object is pressed, and metamaterials that alter the structure's geometry can deliver several sorts of passive feedback and functions [95]. He et al. proposed 3D-printed spring designs for provision of varying passive haptic feedback [87]. It is important to stress that, although the aforementioned research has opened some paths to physically prototyping push-buttons, it remains extremely difficult to replicate all the haptic feedback that physical buttons can offer; the subtle "clicky" tactile feedback is especially hard to generate. Crafting prototypes of push-buttons continues to be a costly process that requires significant expertise in working with 3D/2D modeling, devices for printing or cutting out the parts, etc. These issues highlight the need for emulation.

Physical Emulation and Haptic Rendering

"Emulation" generally refers to using a hardware and/or software system to imitate the responses of another system. While it is often addressed in connection with the gaming domain's software emulation [135, 157], my focus is on physical emulation, in which a system (the emulator) imitates the physical and mechanical responses of various objects/interfaces. There has been extensive research exploring shape-changing interfaces – devices that are able to render different shapes [185]. For instance, inForm [67] is a device able to do this in real time, and inForce [158] is an extension to inForm that can render force feedback with the same form factor.

While prior research has pursued rendering of rich and realistic haptic feedback via emulation devices [106, 14, 181], the body of work on emulating push-buttons remains quite limited. Here, I address this important void by demonstrating the use of a button emulator to aid in push-button design. As a button gets pressed, the rapid compression applied to the internal mechanical structure causes rich (force and tactile) feedback. Researchers have attempted to utilize the Phantom device, a six-DOF pen-type force-rendering platform [146, 191, 71], to generate rich force feedback. Phantom can emulate various levels of resisting force and softness of materials, to generate such force feedback, but it lacks the ability to generate vibrotactile feedback, and its relatively low rendering rate (60 Hz) makes it unsuitable for emulating push-buttons. Softness displays [155, 205, 159] and pseudo-force devices [70, 93, 103, 207] have been explored as alternatives. These too are not suitable for directly emulating push-buttons, since their response rate is insufficiently high. The work closest to this application has involved using vibrotactile feedback to emulate various force responses [175, 113]. At this juncture, it is worth noting that prior research into emulation, with its emphasis on understanding the mechanism of haptic perception or emulating certain feedback, has not utilized emulation to support HILO.

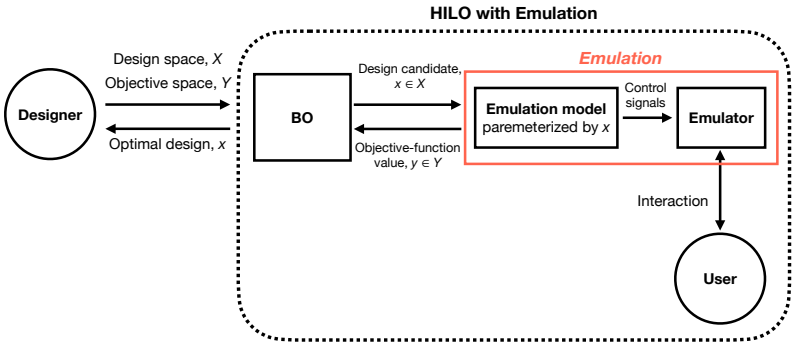


Figure 6.1. The workflow of HILO with physical emulation.

6.2 Emulation: The Case of a Push-Button Emulation Pipeline

This section of the chapter presents the example case of optimizing push-buttons. Thus, I demonstrate the efficacy of employing HILO with physical emulation. The approach to realistic button emulation must address the aforementioned effect of a push-button generating rich feedback within a very short time period. There are two elements to any accurate physical emulation. The first is a **model**, required for capturing and describing the physical phenomenon. In the case of a button model, it should accurately describe the level of resisting force when the button is pressed 1 cm downward etc. The model informs the goals for the simulation at any given point. Secondly, an **emulation workflow** is needed, to deliver the output requested from the model. Emulation workflows differ in their level of complexity. In a simpler case, a physical prototype that is able to generate the correct response suffices as the emulator. Other scenarios demand control methods for guaranteeing that the output is near the intended target specified by the model.

The novel technique I introduce below applies a force–displacement–vibration–velocity (FDVV) model to capture button-pressing. After describing this, I introduce an emulation pipeline that, in conjunction with the modeling, provides for accurate emulation. Lastly, I report on evaluation conducted to validate the realism of the emulation.

6.2.1 FDVV Modeling

Accuracy in capturing the tactility of a button requires a model that accurately describes the physical characteristics that are relevant to the process of button-pressing. Below, I outline preexisting force–displacement (FD) models, then present the improved approach.

FD Models and Their Limitations

Push-buttons have traditionally been described by way of their force–displacement function or force curve, with separate curves for actuation and release [56, 183, 131]. The FD curve can influence sensations, joint kinematics, muscle activity, and user performance [99, 110, 183, 187, 174]. **Linear buttons**' internal structure is composed mainly of a spring, so they do not offer tactile "bump" when pressing or releasing. **Tactile buttons**, in contrast, have a mechanical structure that generates such a bump or a "snap." In another variation, some tactile buttons emit a clearly audible clicky sound when reaching the snap point. Other important properties of a button are the travel distance (i.e., the total distance between the initial and the bottommost state of the keycap) and the activation point, or the depth at which the button is activated [112]. Although FD models can capture these properties, they fail to consider velocity-dependency and vibration characteristics, which are critical factors.

Another way to conceptualize push-buttons is as a mass–spring–damper system. The inertia caused by the button's mass means that the resisting force depends not on displacement alone but also on velocity and acceleration. Previous studies have modeled the softness required for contact with the target surface at various velocities and accelerations, thereby demonstrating the importance of considering velocity- and acceleration dependence. Another problem with FD modeling lies in its limited ability to capture the aforementioned snap sensation caused by the button's unique internal structure. The force–displacement–velocity–vibration model proposed here addresses that limitation too.

The FDVV Approach

The novel modeling method I propose as an extension of FD models is illustrated in Figure 6.2. It captures the velocity dependency relation by incorporating several FD profiles, for various speeds. In addition, vibration is obtained by a microphone during a button press. To accompany the FDVV model, I developed a novel end-to-end emulation workflow covering the operations from capturing a model to emulating it.

6.2.2 The Emulation Pipeline

Step 1 (button capture): The technique begins with a novel approach to measuring the FDVV characteristics of push-buttons. Firstly, to capture the velocity-dependence of button pressing, one measure presses in several velocity conditions. Secondly, instead of a rigid, static-velocity probing object such as a mechanical probe, a human finger does the pressing. This provides a more realistic response envelope of the sort encountered in users' real-world button-pressing. Thirdly, vibrations during presses are recorded.

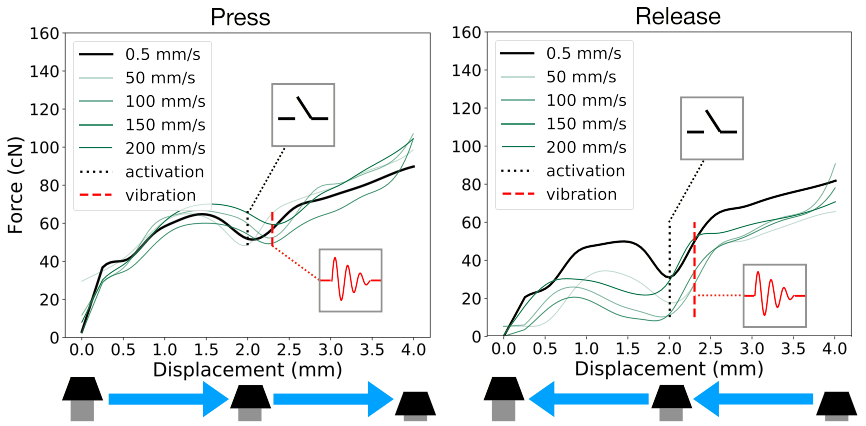


Figure 6.2. A depiction of what the force–displacement–vibration–velocity model represents, detailed more fully in Publication IV.

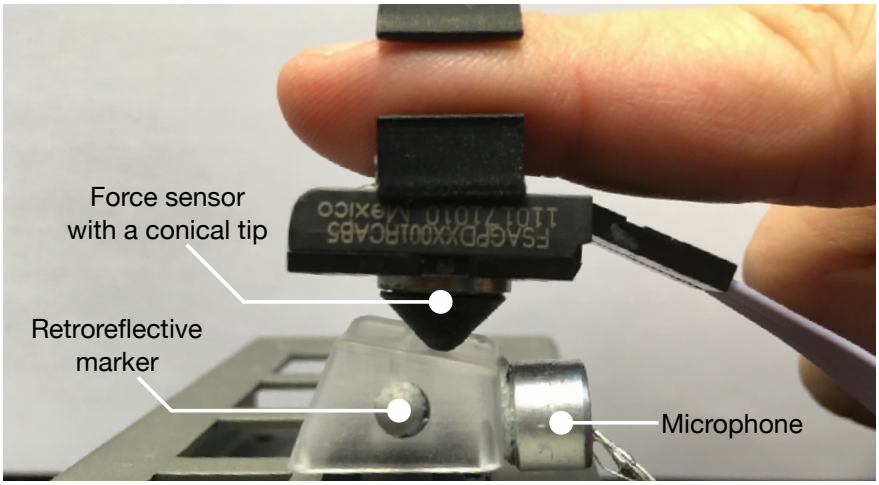


Figure 6.3. Capturing the profile of a button.

Step 2 (FDVV modeling): To obtain a lower-dimensional representation and enable efficient design and optimization, the procedure transforms the raw measurements from the previous step into an FDVV model. However, the raw measurements are inherently noisy and potentially require high-dimensional parameterization. For overcoming this issue, the data undergo several preparatory operations. The FDVV modeling comprises several steps of preprocessing, accordingly. The first one is **filtering**, which entails the application of an electrical low-pass filter during data acquisition and a Gaussian filter after the acquisition is complete. The second step is **synchronization** addressing the fact that data were gathered from the microprocessor and motion-tracker at different rates (1,000 Hz and 256 Hz, respectively). To synchronize the timestamps, the procedure up-sampled

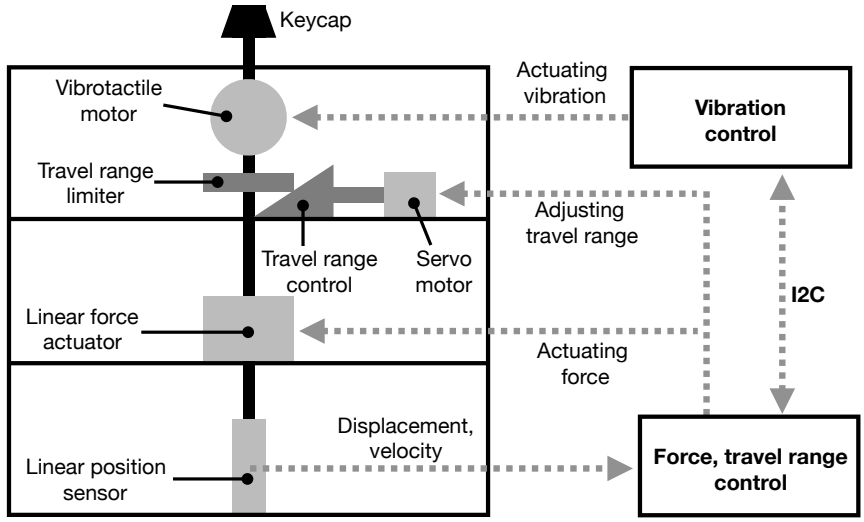


Figure 6.4. The button emulator.

the motion-tracker data. After synchronization, several smoothing operations were performed in succession. The next activity involved fitting the data to a low-dimensional B-spline model. The selection of this model type stemmed from its ability to approximate the data with minimal error while keeping the number of parameters low. For determining the optimal number of B-spline knots, the procedure minimized the Bayesian Information Criteria [119]. This allowed for a lower-dimensional representation of the data, reducing the complexity of the model and enabling more efficient design and optimization.

Step 3 (using the physical emulator): Figure 6.4 presents the physical emulator, which has four major components: a linear force actuator, a linear position sensor, a voice coil acting as a vibrotactile motor, and a servo motor. A microprocessor (Adafruit ItsyBitsy M0) drove the force actuator, the sensor, and the servo motor. To adjust the travel range, this microprocessor directed the servo motor to change the location of the travel-range controller, which further modified the travel range of the button. Another microprocessor (Arduino Uno) drove the vibrotactile voice coil with a wave shield. When emitting the vibrotactile cue was needed, the ItsyBitsy board requested Arduino Uno to trigger the vibrotactile motor (voice coil) such that the corresponding wave files play.

Step 4 (iterative compensation control): Making sure the force response of a push-button gets emulated accurately requires one to account for the transfer function of the force actuator. Yet no prior work on button or force emulation has considered this issue. To compensate for the transfer function, I propose an iterative compensation method, as depicted in Figure 6.5. The central idea is to adjust the signal amplitude associated with

force-actuation at each displacement point until the target resisting force is detected by the sensor as the key is being pressed against the keycap. This iterative process allows us to achieve accurate emulation of the force response in a manner that factors in the unique transfer function of the force actuator. The iterative compensation process can be mathematically described as

$$u_{k+1}(p) = u_k(p) + \Gamma(error_k)(y_d(p) - y_k(p)), \quad p \in [1, n], \tag{6.1}$$

where $u_k(p)$ is the actuation signal at displacement point p in the current iteration. $u_{k+1}(p)$ is the actuation signal in the next iteration at the same displacement; $y_k(p)$ is the force detected by the sensor, and $y_d(p)$ is the target force level at the given displacement point. Finally, $\Gamma(error)$ represents the proportion of adjustment of the actuation signal that must be applied, arrived at from the error value in this iteration ($error_k$). The error from the current iteration is defined as

$$error_k = \alpha \cdot \frac{\Sigma_{p=1}^{n} |y_d(p) - y_k(p)|}{n} + (1-\alpha) \cdot \max_{p \in [1,n]} |y_d(p) - y_k(p)|. \tag{6.2}$$

In this expression, two terms constitute the error. One is the average difference between the target FD curve and the measured curve at each measuring displacement. The other is the largest error between the target FD signal and the measured FD signal. The weight applied between the two is α.

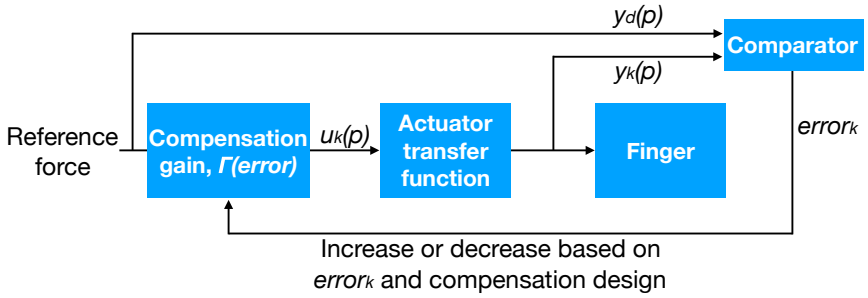


Figure 6.5. Iterative compensation finds a way to render an FDVV model via a given simulator plant.

6.2.3 Evaluation of the Emulation

To evaluate the perceived realism of the emulation, a controlled study employed ABX testing, which is a commonly used method in psychophysics to compare two possible sensory stimuli to a target [148, 45, 114, 156]. Participants experienced a real reference button (X), after which they were asked to press two synthetic buttons (A and B) and indicate which one offered a more realistic rendering of the target. Both A and B were

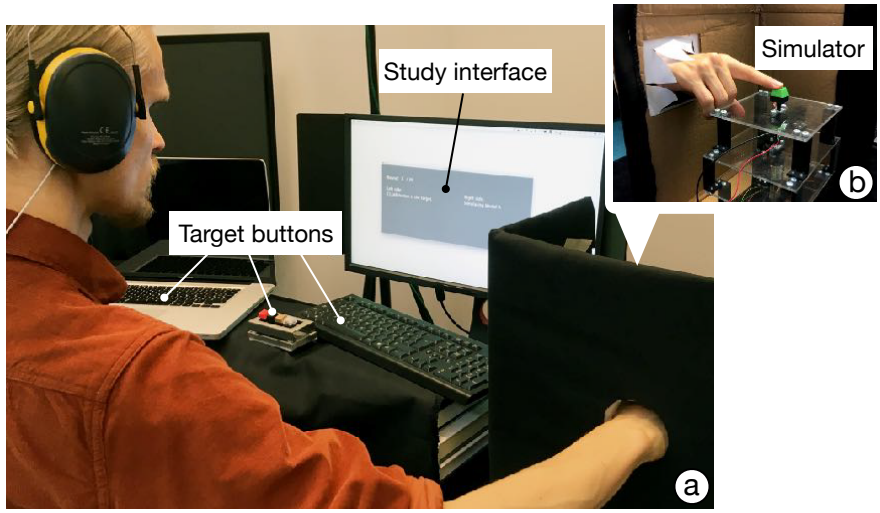


Figure 6.6. The setup for the study.

rendered by means of the emulator described above. The study compared FDVV-based models against traditional FD ones.

Participants
For the study, I recruited 12 participants (6 of them female).

The Task and Apparatus
The study compared the realism of six target push-buttons. I captured all these buttons in both traditional FD models and FDVV models. The emulator was placed inside a box. The participants had to reach into the box for performing the pressing action (see Figure 6.6). This prevented the users from being biased by visual cues. An interface informed the participants which of the two buttons ("A" or "B") was the current rendering.

Procedure and Experiment Design
In each round, which exposed the participant to a specific reference button and two emulated buttons, the participants were informed that there were two emulated buttons: button A and button B. They were then instructed to freely experience the reference button by pressing it at different velocities. Then, they were instructed to experience the two emulations, buttons A and B. Participants were told that, when they were ready to provide their judgment, they needed to indicate which provided greater realism (A, B, or neither). Further, they needed to provide the perceived realism of the two emulated buttons by a seven-point Likert scale. The FDVV and FD models were randomly assigned labels A and B.

Results
The FDVV modeling delivered greater perceived realism. In their choices between models, the participants determined the FDVV model to be more realistic 77.31% of the time. Wilcoxon signed-rank tests afforded further analysis. For all buttons presented, there were statistically significant differences between the FDVV models and the traditional FD models. I refer the reader to Publication IV for more details regarding the analysis.

6.3 An Example of Optimization: HILO for Button Design

After evaluating the realism produced by the physical emulation pipeline, I set out to demonstrate the potential for human-in-the-loop button optimization using physical emulation with a specific design task of temporal pointing [127]. Temporal pointing tasks require user activation of a certain feature within a predefined time window, which is a commonplace action in many interactive applications (games etc.). With this optimization process, I did not consider the velocity-dependent properties of button activation; I assumed a consistent pressing speed across all participants.

6.3.1 The Design Parameters and Objective Function

There were eight design parameters in all. Six of them contributed to specifying the button model: $x1$, $x2$, and $x3$ (displacements of these three control points) and $f1$, $f2$, and $f3$ (actuation signals of these three control points). The ranges of these six parameters were $x1 \in [0,1), x2 \in [1,3), x3 \in [3,6.2)$ and $f1, f2, f3 \in [20,180]$. The final two parameters were the activation point, p_a, and the vibration point, p_v. Their ranges were $p_a, p_v \in [0.5,5.5]$.

The scenario was a case of single-objective optimization in which the objective was to minimize a user's temporal error, i.e., the mean asynchrony [127, 188].

6.3.2 The Study Setting

I recruited 10 participants (4 of them female) for this optimization study. The participants were presented with the button emulator and an LED strip. They were instructed to press the button when the yellow bullet on the LED strip had reached the center of the blue target zone (Figure 6.7). There were two levels of task difficulty: "Easy" (the bullet moves slower) and "Difficult" (the bullet moves faster). For each iteration of each difficulty level, 27 trials were collected and then the system computed the mean asynchrony. The optimizer then updated the design based on the observed performance. Comparing against the optimized button, I selected a series of

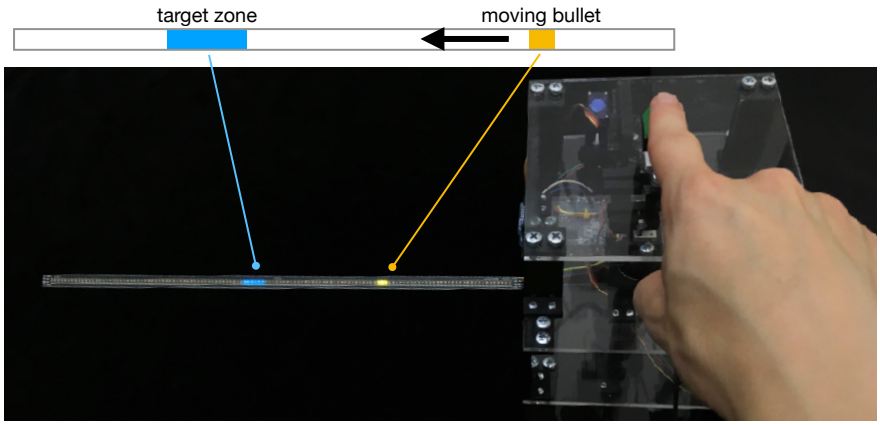


Figure 6.7. The setting for the push-button optimization.

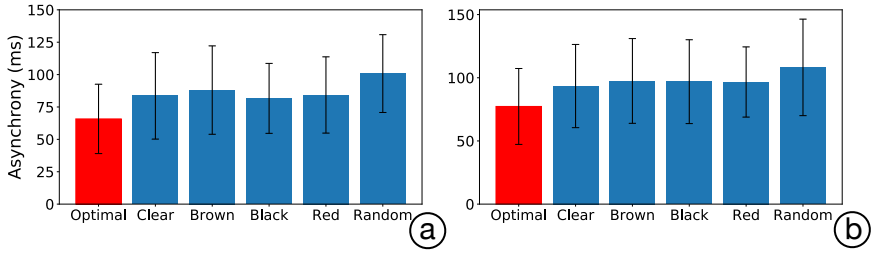


Figure 6.8. The results of the push-button optimization. The optimal button is the personalized optimal design. The Clear, Brown, Black, and Red switches are 4 mm mechanical Cherry MX switches rendered by our emulator. Lastly, the random button is a design generated randomly within the design space.

Cherry MX 4 mm switches[1] and a randomly generated button as baselines.

6.3.3 Results

For the Easy level, the mean asynchronies produced for the optimal, clear, brown, black, red, and random buttons were 65.8 ms (sd = 6.15), 83.6 (sd = 7.65), 88.04 (sd = 7.82), 81.65 (sd = 6.19), 84.28 (sd = 6.75), and 100.78 (sd = 6.89), respectively. For the Difficult level, the asynchrony's mean values, in corresponding order, were 77.3 ms (sd = 6.89), 93.43 (sd = 7.56), 97.48 (sd = 7.69), 96.9 (sd = 7.61), 96.65 (sd = 6.36), and 108.22 (sd = 8.77). Two-way repeated-measures ANOVA revealed the existence of a significant main effect of buttons on mean asynchronies, $F(5,95) = 10.724$, with $p < 0.001$. Pairwise *post-hoc* comparisons with Bonferroni correction showed that the optimal button design indeed outperformed the rest ($p < 0.05$).

[1] See `https://www.cherrymx.de/en/cherry-mx.html`.

6.4 Summary and Discussion

Designing physical interactions often entails optimizing their physical and mechanical properties. However, varying the physical properties requires sets of physical prototypes, which may display a host of variations, for communication and testing of the ideas involved. Because the ensuing resource demands may make HILO difficult to apply for such design tasks, I explored emulation as a potential solution, thereby striving to answer CQ 1. Instead of creating a physical prototype for each iteration, the solution developed imitates the mechanical properties of particular designs via emulation. This work accounted for both of the important aspects to emulation: the model of the physical phenomenon and a workflow to support the emulation, which usually involves control methods.

To demonstrate the concept's workability in practice, I implemented this novel approach for emulation in the form of FDVV models for button-pressing and an end-to-end emulation pipeline. I investigated the proposed method's efficacy further by taking push-buttons as an example and conducting an evaluation experiment that demonstrated reaching greater realism than previously implemented approaches have produced. Finally, I conducted HILO with the emulator to optimize push-buttons for a temporal pointing task. The optimal button designs that emerged outperformed the baseline preexisting button designs.

This chapter ties in with the dissertation's fourth contribution,(**human-in-the-loop optimization with emulation**, through the proposed novel approach to applying HILO with emulation. The chapter also points to several interesting directions for future research. On one potential path, scholars could embrace physical emulation as a design tool rather than merely a means of generating specific feedback. This could involve exploring how designers might use emulation to prototype and iterate on physical designs more efficiently and effectively. Additionally, future research could investigate a wider range of applications for HILO with emulation, beyond push-button design alone. This work could include exploring possible uses of emulation to optimize other types of physical interactions or to support the design of more complex systems.

7. From the Real World toward Simulation

"What I cannot create, I do not understand." — Richard Feynman

"Science is what we understand well enough to explain to a computer; art is everything else." — Donald E. Knuth

My attention thus far in the dissertation has been directed to expanding the application scope of HILO to multi-objective cases, group-level optimization, and HILO's practical implementation with physical emulation. In this chapter, my aim is to tackle the issue of the cost and effort associated with conducting human evaluations within the HILO process. Evaluation is costly for both the designers and the participating users: the designers need to plan the study, recruit people to participate, and conduct the HILO, while users have to dedicate immense effort to repetitive tedious tasks. Additionally, as the earlier chapters elucidate, performing HILO in itself is a time-consuming process.

The central question I set out to answer in this chapter is **CQ 1: Can we free the HILO procedures of human efforts?** To this end, I extend the definition of the term "human" in Human-in-the-Loop Optimization in this chapter: Previously, the word "human" referred to human participants, and here, the word refers to both "actual human participants" and also "synthetic human participants." With this extension, I propose a novel **simulation-based optimization framework** to automate the evaluation process. Within this framework, there is an agent governed by a user model, which produces human-like behaviors to inform the evaluation of each design's quality in a simulation environment. An optimizer then iteratively proposes the next design instance that is likely to yield the best possible performance of the agent. Since no human users are involved, this framework offers the designer high efficiency and automation. Furthermore, since the agent's performance does not deteriorate over time (there are no fatigue effects), the framework provides greater robustness.

To demonstrate the efficacy of this framework here, I present a use case of

optimizing a 3D pointing interaction by means of the proposed simulation-based optimization framework. The results attest that the framework supports optimizing the pointing interaction with highly efficient automation. Overall, this framework represents a promising route to eliminating the cost and effort associated with employing human evaluations in the design process. Future research could investigate the scalability and generalizability of this framework with various design tasks and domains.

7.1 A Simulation-Based HILO Framework

This section of the chapter deals with the novel framework I propose for the simulation-based optimization of design interfaces. The discussion begins with an orientation to prior work. Shifting the optimization process from the real world to a simulation environment is not a new idea.

7.1.1 Background and Related Work

As is evident from the review in section 2.2, simulation-based optimization is commonplace when engineering practitioners find physical evaluation overly expensive or otherwise impractical [144]. Engineers have applied it for optimizing buildings [163], urban transportation design [170], aircraft [210], the aerodynamics of rocket design [202], and other designs. However, these engineering tasks display a fundamental difference from optimization of interface design: as noted earlier on, the former often relies on well-established models of the problem.

While HCI researchers have attempted to optimize interfaces by using simulations, their efforts are plagued by the fact that many interface interactions lack precise models. In fact, this is the main challenge they face. For example, optimization of layout-assignment problems often utilizes the well-established Fitts' law model, which is suitable primarily for simple tasks, such as optimizing soft-keyboard layout [20], physical keyboards' layout [62], and keyboard layout for gaze-based interaction [18]. Other HCI optimization work relies on assumptions about user behavior (e.g., the frequency of using particular functions), in areas such as applying combinatorial optimization methods to graphical layouts [171] and optimizing menus via reinforcement learning (RL) [213]. Likewise, some display optimization is based on heuristics rather than user models. For example, Luzhnica and Veas optimized a tactile pattern by avoiding two vibration points identified as too close to each other [140].

Frequently, these techniques' dependence on well-established models or simple user-behavior assumptions renders them unable to generalize to different tasks or handle more complex settings that require physical motor control and interaction. These shortcomings highlight the need for

a simulation-based optimization framework that can handle a broad range of HILO tasks even in the absence of precise models or assumptions about user behavior.

7.1.2 The Simulation-Based Optimization Framework

In light of the need described above, I propose a novel simulation-based HILO framework intended both to 1) be generalizable to a wider range of tasks and 2) handle complex and realistic interfaces. Figure 7.1 illustrates the framework. The core concept is moving the HILO from the real world to the simulation domain. This framework has three key components:

- **A physical simulation environment**: A realistic environment that can simulate physical responses and phenomena is needed. This must cope with a wide range of tasks.

- **An RL-based user model**: Reinforcement learning is a general framework in which agents learn the policy for interaction with the environment via trial and error [206]. An RL-based user model permits interaction with a broad range of tasks, even complicated and realistic ones.

- **Bayesian optimization**: Since BO does not require assumptions about or knowledge of the problem or the user model, it is generalizable to numerous sorts of design-optimization problems.

In a divergence from previous simulation work in the HCI field, most of which has targeted a specific interaction, the elements in the framework I propose are not limited to certain interactions or tasks. Rather, they are generic solutions for a wide range of problems. Against the backdrop of the preceding chapters' thorough introduction to Bayesian optimization, the discussion below focuses on presenting the simulation environment and the RL-based modeling involved.

It is worth mentioning that the proposed simulation-based optimization framework differs from the previous model-based optimization, such as optimizing layout based on Fitts' law, in the form of evaluation. In the traditional approach where simplistic models are applied, the model directly generates the statistical evaluation result of a design instance. For instance, one can apply Fitts' law to predict the completion time of a selection. Yet, such predictions are not based on moment-by-moment movements. On the other hand, the proposed simulation-based optimization framework generates moment-by-moment motions and interactions and then the system calculates the performance metrics. It permits more realistic replication of user evaluation and also allows for higher generalizability.

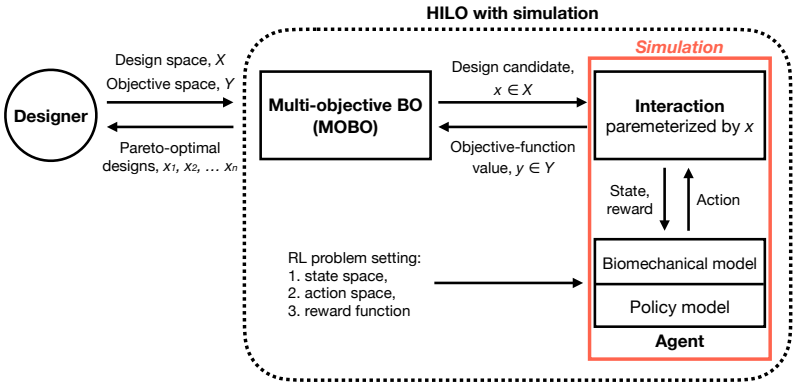


Figure 7.1. The simulation-based HILO framework.

7.1.3 Physics Simulation

Physics simulation and engines have revolutionized the study and analysis of complex systems, across various fields [124]. Recent years have seen them gain particular importance for the training of robots and deep-RL policies. Of the various physics engines introduced (DART [129], PhyX[1] ODE [203], etc. [59]), MuJoCo [215] stands out for its exceptional realism and efficiency. For the work presented in this chapter, I employed the simulation-based HILO framework by means of that leading physics engine.[2]

7.1.4 Reinforcement-Learning-Based User Models

There is a long history of constructing user models to improve understanding and prediction of user behaviors [21], and recently RL has emerged as a generic framework for modeling a wide range of tasks [42, 152, 101]. In this general framework, an agent learns to select a series of actions for the optimal accumulated reward. Typically, RL is modeled as a Markov decision process with the following key elements:

- **The state space**, S: All possible statuses of the agent and the environment
- **The action space**, A: All actions (choices) available for the agent to select
- **A reward function**, $R(s,s')$: A function that generates the utility of performing a certain action from a given state

[1] See `https://developer.nvidia.com/physx-sdk`.
[2] For more information about MuJoCo, please consult its official Web site, `https://mujoco.org/`.

For a more detailed introduction to reinforcement learning, please refer to the presentation by Sutton and Barto [206]. Below, I present an example case wherein RL as described here can be utilized to model and simulate humans' motor behaviors. Specifically, I model affordance perceptions through recognizing motions, which are acquired via the mechanism of RL. A preliminary case of design optimization utilizing RL models is presented after the affordance perception modeling in section 7.2.

7.1.5 Reinforcement Learning for Understanding of Affordance

An important step toward agent-in-the-loop optimization is to allow the agents to interact with the target interface as envisioned. That is, the agent should be able to perceive affordance. Despite its importance in HCI and a rich corpus from ecological psychology, researchers have never produced a theory that explains the formation of affordance. Simultaneously, the numerous supervised-learning-based computational models available remain largely detached from the world of motion planning. Therefore, such affordance models cannot be directly applied to guide agents' motions. Addressing these gaps, I developed a theory explaining the formation of affordance. I argue that the perception of affordance is obtained via reinforcement learning. The key elements of this theory can be articulated thus:

1. Affordances are learned when reinforcement signals are provided in response to motions. An organism tries out different possible motions to learn which motion results in the highest reinforcement signal (i.e., the greatest reward). In the future, a rational agent will then avoid a motion that leads to little reward and repeats/favors a motion that brings a large reward [40].

2. Affordance perception is guided by predicted rewards. The solution to the problem of delayed feedback is dependent on prediction, which is grounded in reinforcement learning [206]. To decide on an optimal action at the moment, the brain must learn to predict the outcome of a given action [77].

3. Affordances are learned by exploring and exploiting. Based on the theory of reinforcement learning, a rational agent intelligently balances exploration and exploitation, aiming to maximize the reward.

4. Affordance perception generalizes to unseen instances of a category. An organism generalizes policies via feature similarities. In other words, the motions toward two similar objects are similar.

5. Humans learn to associate linguistic categories (labels) with affordances. Nearly always, affordance theory has presumed linguistic categorization of actions [216, 36]. Such linguistic labeling allows us to reason and communicate, which further boosts the development of cognitive representations.

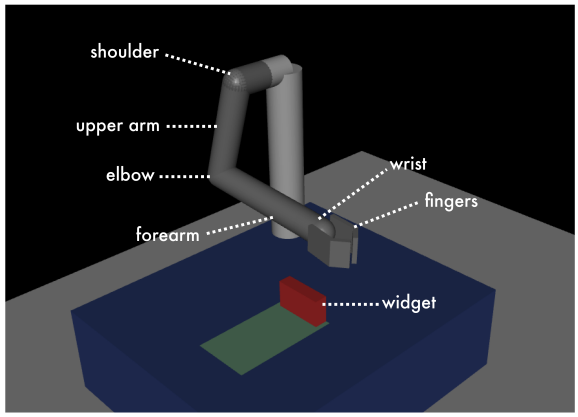


Figure 7.2. The team developed a computational model of the affordance theory, implemented in the MuJoCo physics engine. It enables a virtual robot to interact with widgets.

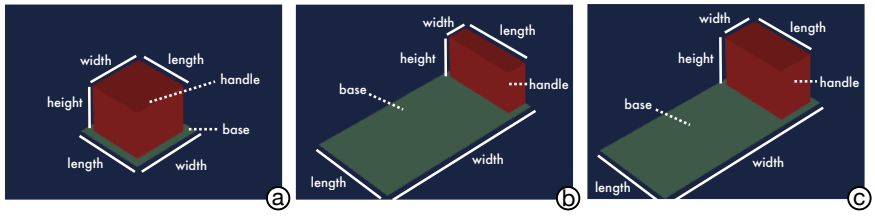


Figure 7.3. The virtual robot interacted with widgets of several types: (a) button widgets; (b) slider widgets; and (c) a deceptive widget that, while resembling a slider, allows only push actions, similar to a button.

Evaluating the Affordance Model via Deceptive Widgets

For the evaluation, I devised a simulation implementing a robot agent, which is essentially a robotic arm with characteristics similar to the human (Figure 7.2). There were two types of widgets for the agent to interact: buttons (in Figure 7.3, pane a) and sliders (in pane b). Each widget affords a particular action – a button provides the possibility of "pressing" and a slider provides the possibility of "sliding." In addition, I implemented a "deceptive widget," which has the appearance of a slider but it only affords "pressing"; this widget (shown in the figure's pane c) was deployed only in testing. When successfully triggering a widget, the agent receives a reward signal.

In the training phase, the agent interacts with the buttons and the sliders, via the process depicted in Figure 7.4, pane a. Evaluation was carried out after training, to confirm the robot's ability to use these two widgets correctly. In 1,000 trials with each widget, the agent interacted with the button and achieved a success rate of 91.3%; meanwhile, the agent interacted with the slider and achieved a success rate of 94.5%.

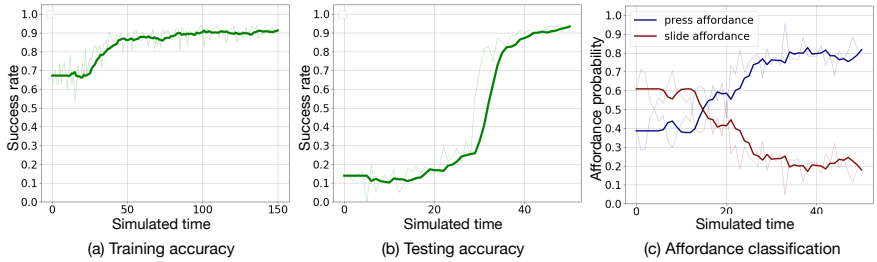


Figure 7.4. The results of the model training and testing, detailed more fully in Publication V.

Next, I labeled 1,000 successful movement trajectories with each widget: the trajectories for activating the button were labeled as "press", and the trajectories for activating the slider were labeled as "slide." With these labeled data, I proceeded to train a motion classifier. Here, 80% of the data items were randomly assigned to the training set, 10% to the validation set, and 10% to the testing set. The motion classifier achieved 85.8% validation accuracy and 88.1% testing accuracy, indicative of high-quality recognition.

In the last step, I evaluated the model by using the deceptive widget. The robot interacted with the widget and continued discovering the right action through successive interactions. The results, presented in Figure 7.4 (b and c), show the evolution of progress in the agent's affordance perception. The agent initially perceived a greater affordance for sliding and, consequently, experienced low success rates. Through interactions and reinforcement learning, however, the robot gradually adjusted its policy and came to perceive the right affordance (i.e., pressing).

7.2 A Preliminary Example Case: 3D Touch Optimization

This section describes the validation of the efficacy of the framework whose core elements are outlined above. That simulation-based optimization framework was validated for the goal of optimizing the parameter settings of a transfer function for 3D touch interaction. The validation application followed three phases:

1. **Agent training** – training the agent to acquire the policy for the target interaction via reinforcement learning

2. **Optimization** – deploying the optimizer to derive the Pareto-optimal parameter setting(s)

3. **Evaluation** – comparing the performance when given the optimized

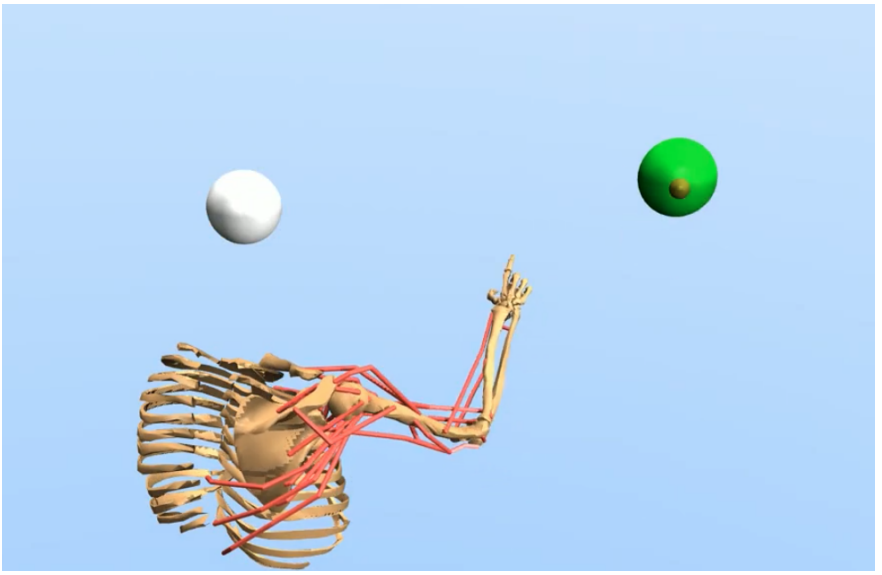


Figure 7.5. The simulation of 3D touch.

parameter setting and a random parameter setting

7.2.1 Configuration for the 3D Touch Interaction and Optimization

The interaction setup was established in the manner introduced in subsection 4.3.1. Again, the research team chose the Go-Go technique to serve as the transfer function. It makes use of two parameters, D and k (for details of D, k, and the definition of operation distance, please refer back to Figure 4.6). For the present setting, there were a few subtle modifications: Firstly, since there is no controller in the simulation, I used the fingertip position in place of the controller position; hence, the Go-Go technique computation's figure for the distance between the controller and the center of the body uses the distance from the fingertip to the center of the body. Secondly, because there is no button for the agent's completion of the selection when the cursor reaches the target, dwelling for five steps (timestamp increments) denoted making the selection. Figure 7.5 shows the interaction with the target in the simulation.

Iteration and Target Placement
The agent-training phase utilized targets randomly sampled from the [1,2] operation distances range. A uniform distribution was used. Publication II supplies details related to the operation distance in the study conducted. For the optimization phase, I evenly discretized the target range into 21

distances (i.e., operation distances of $[1, 1.05, 1.1, ..., 2]$). In each optimization iteration, every discrete distance was sampled 200 times, for a total of $21(distances) \times 200 times$ selections.

Design Parameters
For the parameters, I used the settings $d \in [0.3, 0.4]$ and $k \in [0.5, 1.5]$.

Objective Functions
Two objective functions were considered, for the hit rate and the efficiency. As noted above, there were $4,200$ selections per optimization iteration. The hit rate, defined as the number of successful selections in each iteration divided by 4,200, has the range $[0, 1]$. Efficiency was a normalization function of completion time. Because there is no convenient time unit (second, minute, etc.) in the simulation environment, I took the simulation step as the unit for time. Specifically, I converted $[160, 20]$ steps to a $[0, 1]$ range for the efficiency metric. Therefore, for both objective functions, a higher value reflected better performance.

Settings for Bayesian Optimization
I set up MOBO to use expected hypervolume improvement (EHVI) [52]. The initial sampling, with a value set to 5, was followed by 30 optimization iterations.

7.2.2 Agent Settings

In the system developed, the agent has two components: the biomechanical model and the policy model. The biomechanical model chosen is the one introduced and used for User-in-the-Box [94].

State
Here, the state is composed of three elements together constituting the observation of the biomechanical model: The first is **vision**: The model has a fixed forward-facing visual field. I took the full set of RGB values from the 2D view as part of the state. The second element is **proprioception**, represented via the 3D coordinates of the fingertip position and the cursor position. The final component, **the target position**, refers to the 3D coordinates for the center of the target.

Action
The action space is identical to that in the User-in-the-Box work It includes all the muscles of the right arm of the agent.

The Reward Function
When the agent carries out an action a, the environment generates a reward $r(s, a)$. The reward has two parts, a distance penalty and task-

completion reward. The **distance penalty** is based on the distance from the center of the cursor to the center of the target. This distance is multiplied by a scalar and normalized to the value $\in [-0.01, 0]$. The closer the cursor is to the target, the smaller the penalty received. When the target is successfully selected (i.e., once the cursor has reached the target and stayed there for five steps), the agent receives a **task-completion reward** of value 10; otherwise, the reward is 0.

Training the Policy Model
I employed Proximal Policy Optimization [194] to teach the policy of this interaction. I directly used the implementation in the stable-baseline3 library.[3] The library's documentation[4] supplies further details of the hyperparameter settings.

To derive a generalized policy that is able to interact with a range of potential design settings, at the beginning of each iteration, a set of new design parameter (d and k) values are uniformly sampled from the given range (as shown in *Design Parameters*).

I set the maximum step of each episode (the horizon) to 160 steps. If the agent fails to select the target within 160 steps, it does not receive the task-completion reward, and the environment is reset.

7.2.3 Results

Below, I present the results in terms of the sequence followed. I begin with the first phase, addressing the training progress of the agent model. Then, I present the optimization results and the optimized parameter setting. The subsection ends with a comparison between the optimized setting and random settings.

Phase 1: Agent Training
I trained the agent for 10^8 episodes. Figure 7.6 presents the progress of training. At the left is the episode length over the course of the training. Episode length gradually decreased. This is evidence that the agent learned to complete the task more efficiently. The right-hand part of the figure shows the episodic reward per episode, which rose throughout the training process.

Phase 2: Optimization
Figure 7.7 presents the objective functions resulting from the optimization. There are two objective functions in our application (hit range and efficiency), and we can see a strong correlation between them. Namely, a design that allows for better access to the targets (higher hit rate) also

[3] Available via `https://github.com/hill-a/stable-baselines`.
[4] Available at `https://stable-baselines.readthedocs.io/en/master/modules/ppo1.html`.

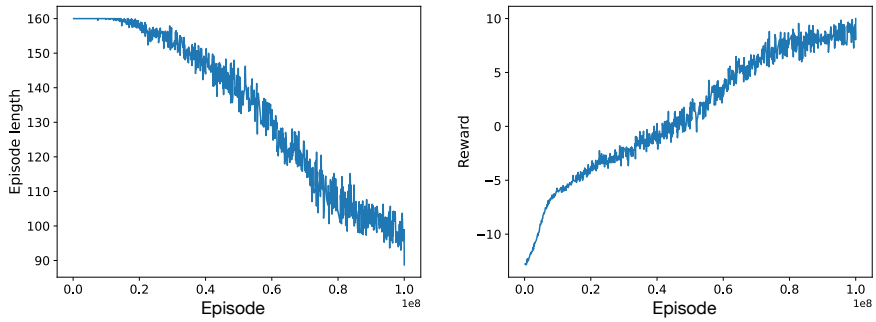


Figure 7.6. The progress of agent training.

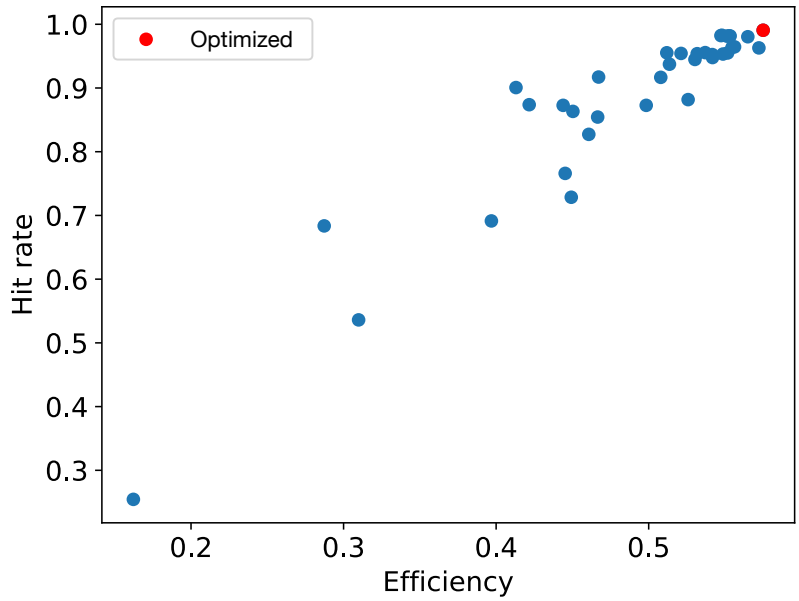


Figure 7.7. The achieved performances during the optimization step.

allows for a more efficient selection. Although multi-objective optimization was employed, there is only one optimized design on the Pareto frontier, which is the design that yields the highest hit rate as well as the greatest efficiency. The optimized design is $d = 0.351$ and $k = 1.23$.

Also note that, although we did not observe the formation of a Pareto frontier in the outcome of the optimization, we can see that the performances of the sampled designs are generally located in the upper right corner, but not scattered over the whole objective function space. This shows that the optimization is effective as the derived designs lead to positive but not random performances.

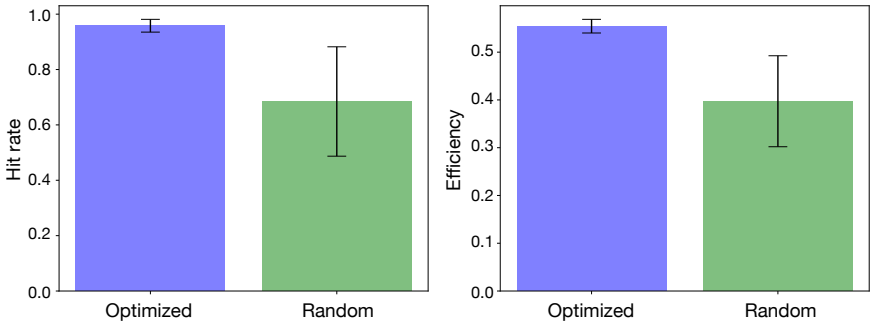


Figure 7.8. The results of the evaluation. The left plot shows the achieved hit rates, and the right plot shows the achieved efficiency. The error bar shows one standard deviation.

Phase 3: Evaluation

I evaluated the performance of the optimized design ($d = 0.351$ and $k = 1.23$) and randomly sampled designs for 10 iterations. For the optimized-design condition, the same parameter setting was applied over all 10 iterations. The sampling for the random condition took a new parameter setting for every iteration. The results are presented in Figure 7.8. For hit rate, the optimized design reached 0.961 ($s.d. = 0.035$) while the random designs reached 0.656 ($s.d. = 0.16$). The optimized design reached an efficiency value of 0.54 ($s.d. = 0.019$), and the random designs' value was 0.383 ($s.d. = 0.88$). In t-tests run to compare the performance levels, significant differences emerged for both objective functions (both $p < 0.05$). This analysis showed that the optimized design outperformed the baseline conditions in both efficiency and hit rate.

Discussion and Future Work

The positive results of the preliminary evaluation notwithstanding, more work is needed. An critical next step is to employ more diverse design objective functions so that the optimized results form a **Pareto frontier** instead of a single optimal design. The evaluation of different tradeoffs along the Pareto front will then inform us of the efficacy of generating optimal designs for different objective preferences.

Another important topic for consideration is **evaluating the optimized design in the real world**. At present, the optimized design has been evaluated only with the agent; the differences between the agent and human participants have not been fully investigated. The design that proved optimal for it might not be the optimum for certain users. It is important to investigate the discrepancy between the synthetic performance of the agents and the actual performance of the human participants. Along this line of research, future work should also investigate how to generate more realistic synthetic behaviors.

Also, the agent in the simulation conducted had a fixed set of physical characteristics. Users in the real world exhibit variations in numerous physical features – limb length, muscle strength, etc. Furthermore, users may differ in their strategies for approaching the task. The simulation did not consider these between-user variances. An important next step would be to train a policy model that is generalized for different user types, which should allow for more generalized optimal designs.

Last but not least, future research should consider applying this simulation-based optimization framework to other interaction techniques. The use of RL allows for modeling a potentially wide range of interactions. However, as the complexity increases, it is more challenging for the RL-based approach to generate human-like behaviors due to the difficulties in crafting realistic reward functions and physical characteristics. It is important to understand the limitation of RL-based agents for simulating human motor behaviors, which will then be useful for us to fully understand the application scope of such simulation-based optimization.

7.3 The Work in Summary

For the chapter-level goal of addressing how to make HILO more accessible to practitioners by eliminating the cost of human user studies, I have presented a novel simulation-based HILO framework that supports the efficient optimization of interaction designs by deploying an agent as the evaluator of the design instance. This work forms part of the last contribution listed for the dissertation, **a simulation-based human-in-the-loop optimization framework**. The key difference between this framework and previously introduced optimization systems using simulation is that it is not constrained to any specific interaction or user model. It can be applied to a wide range of interaction tasks.

The framework factored in three key elements: a physics simulation, an RL-based user model, and BO. The workflow is another important aspect of the work. In the example case presented, the first step of the workflow was to train the user model to acquire a policy model capable of performing the target interaction, the second step was to deploy an optimizer to derive the optimal designs, and the evaluation of the designs derived formed the final step. While further evaluation of the framework is necessary, for a better understanding of its limitations, the results of the preliminary study clearly show that the optimized design for 3D touch interaction outperforms the baseline.

Overall, the work constitutes a promising avenue for practitioners' efforts to optimize interaction designs without incurring the high cost and effort of running human studies. Through the power of physics simulations and RL-based user models, this simulation-based HILO framework offers an

efficient and accessible approach to interaction-design optimization.

8. Discussion and Conclusion

Design optimization involves exploring a large design space to identify the optimal parameter setting. This necessary search task remains complex and challenging. A realistic design challenge usually involves multiple objective functions that a designer must consider during the optimization process. Furthermore, the design parameters are often continuous, presenting a nearly infinite set of possible combinations. The relationship between the resulting objective function and a particular design instance is often unknown, and evaluating the design instance with human participants is an expensive undertaking. Manual optimization, a process in which designers seek a good design by means of various design methods, intuition, and exhaustive trial and error, is a highly demanding and effortful approach to this problem, and it still does not guarantee an optimal outcome.

HILO holds great potential as a solution for design optimization, offering more principled, generic, and automated applications. However, several critical limits have constrained its application scope thus far.

The first constraint arises from HILO having been applied exclusively to single-objective rather than multi-objective design tasks. To address this limitation, I have proposed applying multi-objective BO based on Pareto-frontier learning in HILO. This process identifies a set of designs that lie on the Pareto frontier.

Secondly, researchers have not comprehensively investigated HILO's strengths and limitations. In the dissertation project, a workshop and user study informed understanding of how designers perceive the HILO process. The findings indicate that the participating designers experienced significantly less mental and physical effort in searching for the optimal designs with HILO, relative to manual approaches.

In addition, HILO has been applied with regard to individual users only, as opposed to a group of users. To address this restriction in scope, I have proposed group-level HILO, which aggregates optimization observations from a group of users and derives a model accordingly. Simulations and two user studies attest to the efficacy of the two extensions devised for this

purpose, Global GP and Warm-Start GP.

The fourth important restriction is that HILO must be implemented for the interaction in such a way that the design process need not involve fabrication. To overcome this shortcoming, the project utilized physical emulation for HILO. Through this mechanism, designers need not fabricate or prototype each design instance.

Finally, my research addressed the issues arising from the requirement for every iteration in HILO to include a human evaluator, which is costly and effortful for both the designers and the other participants. The work behind the dissertation extended HILO from the real world to physical simulations to circumvent the hassle and effort of human evaluation.

In conclusion, this dissertation contributes to the field of human-in-the-loop optimization by introducing a set of methods that broaden the application scope of HILO. The methods presented here address the critical constraints of HILO, paving the way for future work on applying HILO to a wider range of design problems.

8.1 The Findings Overall and Their Implications

In summary, the dissertation demonstrates the effectiveness of HILO for reducing designer effort and arriving at better design outcomes. Furthermore, the project has expanded the scope for HILO's application by implementing various computational methods that perform better than traditional design processes. Together, these showcase HILO's potential as a versatile solution: the findings suggest that automating design optimization via computational methods is a feasible and reasonable way forward. State-of-the-art methods such as Bayesian optimization are able to overcome the considerable challenges arising from the complexity and noise associated with human–computer interactions, and they can lead to promising user performance in response to diverse design challenges. There is a clear implication that design practitioners should embrace these means of computational optimization. Additionally, the dissertation highlights expansion in the space ripe for HILO-related research, in that the optimization step has already become automated in many branches of engineering. With advances to computing capability and machine-learning tools, it is now possible to tackle interaction optimization via these techniques. The flexibility of BO invites further enhancements, and future research could push the boundaries of HILO even further.

8.2 Limitations and Future Work

The research presented here contributes to work on several topics. With this fertile ground come several limitations that future steps must address.

8.2.1 Advanced Optimization Methods

To enhance the efficacy of the HILO process further, researchers should strive toward more advanced optimization algorithms. Bayesian optimization is a general framework within which there is ample room for such enhancements.

More Efficient Optimization
One critical limitation of HILO is that it can be time-consuming. For example, with the 3D touch interaction presented in Chapter 4. a two-objective problem featuring four design parameters, it took 60 to 90 minutes (40 iterations) for multi-objective BO to arrive at final designs. A need for more iterations and time is to be expected if the problem has more parameters or objective functions. Alongside the Warm-Start GP demonstration provided in the dissertation, extensive research has examined how to enable more efficient BO [10], with one mainstream approach being meta-learning [217]. Meta-learning in the context of BO is a set of techniques designed to boost the efficiency of solving an unseen optimization problem by incorporating auxiliary data from similar tasks. Several implementations of BO that employ meta-learning have been proposed, some including multi-task GP, deep neural models, etc. I encourage researchers to look into the opportunity of applying them to HILO problems.

High-Dimensional Optimization
One important restriction still hampering state-of-the-art BO-based HILO is its highly limited ability to handle a larger number of design parameters. For instance, design tasks that have more than five parameters prove computationally expensive, and it may become challenging to address a task with more than 10 parameters at all. Real-world design problems often require the designer to deal with a larger number of design parameters. To address the associated limitation, researchers recently have introduced high-dimensional BO approaches [154], which typically involve learning a low-dimensional latent parameter space that BO can operate on, thus enabling the optimization of designs that have more parameters. It is worth exploring whether such techniques can be extended to high-dimensionality HILO problems, thereby enhancing the ability of the approach to handle more complex design problems.

Automated Machine Learning for BO
One step inherent to HILO is setting up the optimizer, which involves tuning hyperparameters at various levels of abstraction. This unavoidable step may discourage designers who do not have a programming background. To make HILO more accessible to practitioners despite this challenge, we can draw inspiration from the field of automated machine learning. Researchers in this emerging field, also known as AutoML, aim to automate the machine-learning pipeline, including hyperparameters' tuning [89]. In essence, AutoML algorithms search for the best machine-learning models and hyperparameters for the particular task and data given. We might be able to reduce the difficulty of setting hyperparameters by integrating AutoML techniques into HILO; further automating the optimization process in this manner could render HILO more accessible to a wider range of practitioners while also aligning it with the trend of automating machine-learning workflows more broadly.

Hierarchical Group-Level Optimization
The Global GP implementation presented in the dissertation has shown promise for generating Pareto-optimal designs suited to a larger group of users. In many realistic design scenarios, though, the target user population may consist of several groups, each with a unique set of needs and preferences. We could approach these groups via a hierarchical structure, with subgroups and, in turn, individual users within each of those. In such cases, it is important to consider group-level optimization for generating hierarchical group-optimized designs via HILO. This would involve developing a framework that can capture the design requirements and preferences distinct to each group well, then generate optimized designs that satisfy those specific needs, all while guaranteeing general feasibility and compatibility. One possible path to this end is through the use of a multi-level covariance kernel [231] and multi-task GP [23], which can capture hierarchical structures in the data and enable efficient group-level optimization. Such an approach possesses the potential to enhance the practical utility of HILO in real-world design applications significantly. Further research and development in this area, which could encompass exploring other hierarchical machine-learning methods, could inform a more comprehensive and robust HILO framework, one able to cater to a wider range of design problems and user populations while still exploiting the power of Global GP.

8.2.2 Making HILO More Usable for Designers

While the HILO framework has demonstrated its effectiveness for varied design tasks, several areas remain in which the framework could be made more usable for designers. Future research plans should consider

enhancing HILO's practicality/usefulness by means of more realistic and complexity-attuned user studies. Better human–machine collaboration is another vital goal.

Investigation of More Realistic Design Challenges
The workshop and user studies conducted in connection with the dissertation shed valuable light on the effectiveness of HILO in a controlled environment with a limited number of users. For a full understanding of the practicality and possible benefits of HILO, future research should aim for more realistic, in-the-wild experiments. Some experiments might involve pairing designers with a larger number of users and allowing them to explore various applications of HILO. Evaluating the subjective feedback of designers and users in a real-world setting would more readily permit judging HILO's effectiveness. Furthermore, it is important to evaluate designer perceptions of the approach's usefulness in practical design conditions (which lack extensive technical support etc.). Such experiments could yield valuable insight for both understanding the practicality of HILO and refining the framework such that it responds better to real-world design problems and the needs of actual user populations.

Better Designer–Optimizer Collaboration
From the qualitative analysis of the studies, it became clear that the designers sometimes felt disconnected from the design process and that the optimizer was driving the design decisions, not *vice versa*. At the same time, the studies showcased human designers' unique ability to quickly identify areas of the design space that do not merit exploring, which is difficult for the optimizer to do. This prompts us to ask an important question: how we can improve collaboration between human designers and optimizers? Answering this demands two lines of attack. Firstly, we need to develop collaborative interactions that permit humans and optimizers to make decisions jointly, and, secondly, we must identify a computational method that supports such collaboration appropriately. This should bring a deeper understanding of how human designers make decisions and how we can effectively integrate their knowledge and preferences into the optimization process. Among the possible approaches are to develop more interactive optimization methods that allow human input in real-time and to create hybrid optimization methods that blend human decision-making with automated optimization. Further research in this area could help bridge the gap between human designers and optimization algorithms, thus leading to more effective and efficient design processes.

8.2.3 Work toward More Realistic Simulation and Emulation

The dissertation provides a starting point via the proposed simulation-based and emulation-based HILO, accompanied by reports on a few applications and preliminary studies. To take the work further, researchers should consider investigating other techniques to address the gulf between simulation/emulation and the real world.

Evaluating the Simulation-Based Optimized Designs with Human Participants
The simulation-based optimization framework proposed in this dissertation was evaluated in a preliminary experiment using a synthetic user agent. While this provided valuable insight, studies with human participants are vital for enriching our understanding of the framework's functionality in real-world scenarios. Importantly, the possibility of real users' behavior and interactions deviating from synthetic users' necessitates testing the framework with a broad spectrum of users. Conducting a full-fledged study with human participants should help to furnish researchers with a more comprehensive evaluation of the proposed framework and its potential impact on practical design applications.

Advanced User Models
A crucial direction for future research is to develop more human-oriented models for design optimization. The models employed in the dissertation project rely on policy-based reinforcement learning, which may generate movements that are not entirely similar to those of humans. Researchers could address this issue by employing reward-shaping [123] and imitation RL [91], for closer alignment of current models' behavior with human behavior. However, since humans adapt much more quickly than RL agents, meta-learning [219] or model-based learning approaches [153] are needed, for higher efficiency. In addition, factors such as learning and fatigue affect human performance. These are not easily replicated by RL agents. Therefore, it is important to develop realistic models that can better capture human behavior and its variations in real-world design scenarios.

Transferring Designs from Simulation/Emulation to Products
Simulation- or emulation-based optimization work wrestles with the critical issue of translating the optimized design from the simulated or emulated environment into the real world, a process commonly known as "sim-to-real transfer" [235]. The ultimate goal of this process is to realize the optimized design in the physical world, such that end users can use it. This simple goal notwithstanding, transferring designs from simulation to reality is a complex process faced with various technical and practical challenges. Among them are accounting for uncertainties, inaccuracies,

and divergences between the simulation domain (the simulation world's physical environment and equipment) and the real world. Addressing these challenges requires developing robust, effective sim-to-real transfer methods that can assure the optimized design's functionality and reliability in real-world settings. Further research in this area could help strengthen the connection between simulation and reality, thus facilitating the efficient and effective realization of optimized designs in practical applications.

Exploring Other High-Precision Emulation
Lastly, there is room for further work associated with the emulation-supported HILO approach I have proposed for the button-pressing task. Exploring other applications is necessary for evaluating the effectiveness and versatility of the approach suggested. One critical factor for successful emulation-based HILO is high-precision emulation, which can be achieved through the development of more accurate models, better emulators, and/or more powerful control methods. Further advances in this direction could promote identifying other applications that could benefit from the proposed approach. Such work too would pave the way for the development of more accurate and efficient emulation-based HILO frameworks.

8.3 Conclusion

With this dissertation, I aim to address the complex and challenging task of design optimization through the use of HILO. While HILO has shown potential as a general solution for design optimization, its scope of application has remained restricted thus far, on account of several limitations. To overcome them, I have introduced a set of computational methods with the potential to augment HILO's capabilities. Work applying Pareto-frontier learning for multi-objective HILO problems demonstrated both positive and negative qualities of HILO. While demonstrating the methods' effectiveness, the research showed that incorporating HILO can reduce the designer's sense of agency and ownership. Another technique I examined is group-level optimization; this generates group-optimized designs and rapidly adapts group-level warm-start models. My work also demonstrated the potential of emulation deployed as an alternative solution to reduce the cost of fabrication, coupled with a simulation-based optimization framework to eliminate the costs entailed by human evaluation. This simulation-based framework's three generic elements allow it to be applied with ease to other design optimization tasks. All of these efforts contribute to enhanced applicability in the design domain.

The outcomes presented here advance the fields of design and HCI through the following contributions:

1. Knowledge: The dissertation showed that HILO is a viable and effective solution for design optimization problems. Various experiments and studies carried out attest that HILO produces better design instances than traditional design methods do. The knowledge produced can serve to inform and guide future research in the field of design optimization.

2. Methods: The computational methods proposed in this dissertation expand the application scope of HILO. By enabling HILO to tackle multi-objective and group-level optimization problems, these methods make it a more versatile and powerful tool for designers. Additionally, the use of physical emulation and simulation has showcased HILO's benefits and potential in relation to the time and other resource demands of prototyping and user studies.

3. Use cases: The use cases presented exemplify the proposed computational methods' suitability for a range of optimization tasks that span such design domains as input devices, wearable haptic interfaces, VR/AR interactions, and physical interfaces. By demonstrating the effectiveness and applicability of HILO for a wide range of design-optimization scenarios, these applications and my presentation of them together lay a foundation for solid research and development in the field.

This dissertation contributes demonstrably to the fields of HCI, design, and machine learning through valuable insight enriching HILO research. The computational methods introduced attest to HILO's potential for various design-optimization tasks, and the use cases' documentation attests to their effectiveness in extending HILO further. The findings and results reported upon in the dissertation open many interesting research questions for further discussion, providing useful information and inspiration for future research in this area.

References

[1] Chadia Abras, Diane Maloney-Krichmar, and Jenny Preece. User-centered design. In W. Bainbridge, editor, *Encyclopedia of human–computer interaction*, pages 445–456. SAGE, Thousand Oaks, CA, 2004.

[2] Kenichi Akagi. A computer keyboard key feel study in performance and preference. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 36(5):523–527, 1992.

[3] Stéphane Alarie, Charles Audet, Aïmen E. Gheribi, Michael Kokkolaras, and Sébastien Le Digabel. Two decades of blackbox optimization applications. *EURO Journal on Computational Optimization*, 9(2):Paper 100011, 2021.

[4] Zahra Amiri and Yoones A. Sekhavat. Intelligent adjustment of game properties at run time using multi-armed bandits. *The Computer Games Journal*, 8(3-4):143–156, 2019.

[5] L. Bruce Archer. Whatever became of design methodology? *Design Studies*, 1(1):17–18, 1979.

[6] Store Arduino Arduino. Arduino. *Arduino LLC*, 372, 2015.

[7] Ferran Argelaguet Sanz and Carlos Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, 2013.

[8] Holt Ashley. On making things the best – aeronautical uses of optimization. *Journal of Aircraft*, 19(1):5–28, 1982.

[9] Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[10] Tianyi Bai, Yang Li, Yu Shen, Xinyi Zhang, Wentao Zhang, and Bin Cui. Transfer learning for Bayesian optimization: A survey. *arXiv preprint arXiv:2302.05927*, 2023.

[11] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. MenuOptimizer: Interactive optimization of menu systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology – UIST '13*, pages 331–342, New York, NY, 2013. Association for Computing Machinery.

[12] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *NIPS'20:*

Proceedings of the 34th International Conference on Neural Information Processing Systems, pages 21524–21538, 2020.

[13] Bela H. Banathy. *Designing social systems in a changing world*. Springer Science & Business Media, 2013.

[14] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. TeslaTouch: Electrovibration for touch surfaces. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, pages 283–292, New York, NY, 2010. Association for Computing Machinery.

[15] Patrick Baudisch. Personal fabrication in HCI: Trends and challenges. In *AVI '16: Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 1–2, New York, NY, 2016. Association for Computing Machinery.

[16] Patrick Baudisch and Stefanie Mueller. Personal fabrication. *Foundations and Trends® in Human–Computer Interaction*, 10(3–4):165–293, 2017.

[17] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Proceedings of the 30th International Conference on Neural Information Processing Systems – volume 1*, 2016.

[18] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. SliceType: Fast gaze typing with a merging keyboard. *Journal on Multimodal User Interfaces*, 13:321–334, 2019.

[19] V. Bhaskar, Santosh K. Gupta, and Ajay K. Ray. Applications of multiobjective optimization in chemical engineering. *Reviews in Chemical Engineering*, 16(1):1–54, 2000.

[20] Xiaojun Bi, Barton A. Smith, and Shumin Zhai. Quasi-Qwerty soft keyboard optimization. In *CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 283–286, New York, NY, 2010. Association for Computing Machinery.

[21] Pradipta Biswas and Peter Robinson. A brief survey on user modelling in HCI. In *Proceedings of the International Conference on Intelligent Human Computer Interaction (IHCI)*, 2010.

[22] J. Blank and K. Deb. Pymoo: Multi-objective optimization in Python. *IEEE Access*, 8:89497–89509, 2020.

[23] Edwin V. Bonilla, Kian Chai, and Christopher Williams. Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems*, 20:153–160, 2007.

[24] Ali Borji and Laurent Itti. Bayesian optimization explains human active search. *Advances in Neural Information Processing Systems*, 26:55–63, 2013.

[25] Doug A. Bowman, Donald B. Johnson, and Larry F. Hodges. Testbed evaluation of virtual environment interaction techniques. In *VRST '99: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 26–33, New York, NY, 1999. Association for Computing Machinery.

[26] Eric Bradford, Artur M. Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *Journal of Global Optimization*, 71(2):407–438, 2018.

[27] Stephen Brewster and Lorna M. Brown. Tactons: Structured tactile messages for non-visual information display. In *Proceedings of the Fifth Conference on Australasian User Interface – volume 28*, pages 15–23. Australian Computer Society, 2004.

[28] Eric Brochu, Tyson Brochu, and Nando de Freitas. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. Eurographics Association, 2010.

[29] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599 [cs]*, December 2010. arXiv: 1012.2599.

[30] Richard Buchanan. Wicked problems in design thinking. *Design Issues*, 8(2):5–21, 1992.

[31] Marion Buchenau and Jane Fulton Suri. Experience prototyping. In *DIS '00: Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, pages 424–433, New York, NY, 2000. Association for Computing Machinery.

[32] David R. Burt, Carl Edward Rasmussen, and Mark van der Wilk. Convergence of sparse variational inference in Gaussian processes regression. *Journal of Machine Learning Research*, 21(1):5120–5182, 2020.

[33] Maria Camacho. David Kelley: From design to design thinking at Stanford and IDEO. *She Ji: The Journal of Design, Economics, and Innovation*, 2(1):88–101, 2016.

[34] Yanshuai Cao, Marcus A. Brubaker, David J. Fleet, and Aaron Hertzmann. Efficient optimization for sparse Gaussian process regression. In *Proceedings of the 26th International Conference on Neural Information Processing Systems – volume 1*, pages 1097–1105. Curran Associates, 2013.

[35] Géry Casiez and Nicolas Roussel. No more bricolage! Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 603–614, New York, NY, 2011. Association for Computing Machinery.

[36] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo. Using object affordances to improve object recognition. *IEEE Transactions on Autonomous Mental Development*, 3(3):207–215, 2011.

[37] Jessica R. Cauchard, Janette L. Cheng, Thomas Pietrzak, and James A. Landay. ActiVibe: Design and evaluation of vibrations for progress monitoring. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3261–3271, New York, NY, 2016. Association for Computing Machinery.

[38] Yeonjoo Cha and Rohae Myung. Extended Fitts' law for 3D pointing tasks using 3D target arrangements. *International Journal of Industrial Ergonomics*, 43(4):350–355, 2013.

[39] Gong Chao. Human–computer interaction: Process and principles of human–computer interface design. In *2009 International Conference on Computer and Automation Engineering*, pages 230–233, 2009.

[40] Nick Chater. Rational and mechanistic perspectives on reinforcement learning. *Cognition*, 113(3):350–364, 2009.

[41] H. Chen, J. Park, S. Dai, and H. Z. Tan. Design and evaluation of identifiable key-click signals for mobile devices. *IEEE Transactions on Haptics*, 4(4):229–241, 2011.

[42] Xiuli Chen, Aditya Acharya, Antti Oulasvirta, and Andrew Howes. An adaptive model of gaze-based selection. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2021. Association for Computing Machinery.

[43] Erin Cherry and Celine Latulipe. Quantifying the creativity support of digital tools through the Creativity Support Index. *ACM Transactions on Computer–Human Interaction*, 21(4), 2014.

[44] Mungyeong Choe, Yeongcheol Choi, Jaehyun Park, and Hyun K. Kim. Comparison of gaze cursor input methods for virtual reality devices. *International Journal of Human–Computer Interaction*, 35(7):620–629, 2019.

[45] David Clark. High-resolution subjective testing using a double-blind comparator. *Journal of the Audio Engineering Society*, 30(5):330–338, 1982.

[46] Andy Cockburn, Carl Gutwin, and Saul Greenberg. A predictive model of menu performance. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 627–636, 2007.

[47] Design Council. Eleven lessons: Managing design in eleven global brands. a study of the design process, 2005.

[48] Nigel Cross. Design research: A disciplined conversation. *Design Issues*, 15(2):5–10, 1999.

[49] Matthew J. C. Crump and Gordon D. Logan. Warning: This keyboard will deconstruct – the role of the keyboard in skilled typewriting. *Psychonomic Bulletin & Review*, 17(3):394–399, 2010.

[50] Alma L. Culén and Asbjørn Følstad. Innovation in HCI: What can we learn from design thinking? In *Proceedings of the 8th Nordic Conference on Human–Computer Interaction: Fun, Fast, Foundational*, pages 849–852, New York, NY, 2014. Association for Computing Machinery.

[51] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

[52] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34:2187–2200, 2021.

[53] Nicola Dell, Vidya Vaidyanathan, Indrani Medhi, Edward Cutrell, and William Thies. "yours is better!" Participant response bias in HCI. In *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1321–1330, 2012.

[54] María Alejandra Díaz, Matthias Voß, Arnau Dillen, Bruno Tassignon, Louis Flynn, Joost Geeroms, Romain Meeusen, Tom Verstraten, Jan Babič, Philipp Beckerle, et al. Human-in-the-loop optimization of wearable robotic devices to improve human–robot interaction: A systematic review. *IEEE Transactions on Cybernetics*, page Paper 14, 2022.

[55] Ye Ding, Myunghee Kim, Scott Kuindersma, and Conor J. Walsh. Human-in-the-loop optimization of hip assistance with a soft exosuit during walking. *Science Robotics*, 3(15):Paper eaar5438, 2018.

[56] C. Doerrer and R. Werthschuetzky. Simulating push-buttons using a haptic display: Requirements on force resolution and force-displacement curve, 2002.

[57] John J. Dudley, Jason T. Jacques, and Per Ola Kristensson. Crowdsourcing interface feature design with Bayesian optimization. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2019. Association for Computing Machinery.

[58] Michael Emmerich and Jan-Willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Rapport Technique, Leiden University*, 34:7–3, 2008.

[59] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. *Proceedings – IEEE International Conference on Robotics and Automation*, 2015(June):4397–4404, 2015.

[60] João Marcelo Evangelista Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. AUIT – the adaptive user interfaces toolkit for designing XR applications. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, 2022. Association for Computing Machinery.

[61] Jing Fang and Yuan Yuan. Human-in-the-loop optimization of wearable robots to reduce the human metabolic energy cost in physical movements. *Robotics and Autonomous Systems*, 127:Paper 103495, 2020.

[62] Anna Maria Feit, Mathieu Nancel, Maximilian John, Andreas Karrenbauer, Daryl Weir, and Antti Oulasvirta. AZERTY amélioré: Computational design on a national scale. *Communications of the ACM*, 64(2):48–58, 2021.

[63] Wyatt Felt, Jessica C. Selinger, J. Maxwell Donelan, and C. David Remy. "Body-in-the-loop": Optimizing device parameters using measures of instantaneous energetic cost. *PLoS ONE*, 10(8):Paper e0135342, 2015.

[64] Elisabeth Fernström and Mats O. Ericson. Computer mouse or Trackpoint – effects on muscular load and operator experience. 28(5):347–354, 1997.

[65] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1655–1664, 2009.

[66] James D. Foley, Andries Van Dam, Steven K. Feiner, and John F. Hughes. *Computer graphics: Principles and practice*. Addison-Wesley Professional, 1996.

[67] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. InFORM: Dynamic physical affordances and constraints through shape and object actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology – UIST '13*, pages 417–426, New York, NY, 2013. Association for Computing Machinery.

[68] Peter I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[69] Scott Frees, G. Drew Kessler, and Edwin Kay. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer–Human Interaction*, 14(1):2–es, 2007.

[70] K. Fujita. A new softness display interface by dynamic fingertip contact area control. In *5th World Multiconference on Systemics, Cybernetics and Informatics, 2001*, pages 78–82, 2001.

[71] Nico Galoppo, Serhat Tekin, Miguel A. Otaduy, Markus Gross, and Ming C. Lin. Interactive haptic rendering of high-resolution deformable objects. In Randall Shumaker, editor, *Virtual reality*, pages 215–223, Berlin/Heidelberg, Germany, 2007. Springer.

[72] Paulo Paneque Galuzio, Emerson Hochsteiner de Vasconcelos Segundo, Leandro dos Santos Coelho, and Viviana Cocco Mariani. MOBOpt – multi-objective Bayesian optimization. *SoftwareX*, 12:Paper 100520, 2020.

[73] Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

[74] Robert H. Gault. Progress in experiments on tactual interpretation of oral speech. *The Journal of Abnormal Psychology and Social Psychology*, 19(2):155–159, 1924.

[75] Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. Explore and exploit: Dictionary expansion with human-in-the-loop. In *The Semantic Web: 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2–6, 2019, Proceedings 16*, pages 131–145. Springer, 2019.

[76] Darren Gergle and Desney S. Tan. Experimental research in HCI. In Judith S. Olson and Wendy A. Kellogg, editors, *Ways of knowing in HCI*, pages 191–227. Springer, New York, NY, 2014.

[77] Samuel J. Gershman and Nathaniel D. Daw. Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual Review of Psychology*, 68(1):101–128, 2017.

[78] Camille Gobert, Kashyap Todi, Gilles Bailly, and Antti Oulasvirta. SAM: A modular framework for self-adapting web menus. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 481–484, New York, NY, 2019. Association for Computing Machinery.

[79] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D. Sculley. Google Vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1487–1495, New York, NY, 2017. Association for Computing Machinery.

[80] Elizabeth Goodman, Erik Stolterman, and Ron Wakkary. Understanding interaction design practices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, pages 1061–1070, New York, NY, USA, 2011. Association for Computing Machinery.

[81] William J. J. Gordon. *Synectics: The development of creative capacity.* Harper, 1961.

[82] Abhijit Gosavi. *Simulation-based optimization*. Springer, 2015.

[83] Saul Greenberg and Ian H. Witten. Adaptive personalized interfaces – a question of viability. *Behaviour & Information Technology*, 4(1):31–45, 1985.

[84] Bernard Grossman, Z. Gurdal, G. J. Strauch, W. M. Eppard, and Raphael T. Haftka. Integrated aerodynamic/structural design of a sailplane wing. *Journal of Aircraft*, 25(9):855–860, 1988.

[85] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human mental workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland, 1988.

[86] Rex Hartson and Pardha S. Pyla. *The UX book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.

[87] Liang He, Huaishu Peng, Michelle Lin, Ravikanth Konjeti, François Guimbretière, and Jon E. Froehlich. Ondulé: Designing and controlling 3D printable springs. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 739–750, 2019.

[88] Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. Human-in-the-loop parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2337–2342, 2016.

[89] Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:Paper 106622, 2021.

[90] Stephanie Houde and Charles Hill. What do prototypes prototype? In *Handbook of human–computer interaction*, pages 367–381. Elsevier, 1997.

[91] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[92] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pages 507–523. Springer, 2011.

[93] Yoshiaki Ikeda and Kinya Fujita. Display of [a] soft elastic object by simultaneous control of fingertip contact area and reaction force. *Transactions of the Virtual Reality Society of Japan*, 9(2):187–194, 2004.

[94] Aleksi Ikkala, Florian Fischer, Markus Klar, Miroslav Bachinski, Arthur Fleig, Andrew Howes, Perttu Hämäläinen, Jörg Müller, Roderick Murray-Smith, and Antti Oulasvirta. Breathing life into biomechanical user models. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, 2022. Association for Computing Machinery.

[95] Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. Understanding metamaterial mechanisms. In *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[96] Alexandra Ion, Edward Jay Wang, and Patrick Baudisch. Skin drag displays: Dragging a physical tactor across the user's skin produces a stronger tactile stimulus than vibrotactile. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2501–2504, New York, NY, 2015. Association for Computing Machinery.

[97] Rachel W. Jackson and Steven H. Collins. Heuristic-based ankle exoskeleton control for co-adaptive assistance of human locomotion. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(10):2059–2069, 2019.

[98] David G. Jansson and Steven M. Smith. Design fixation. *Design Studies*, 12(1):3–11, 1991.

[99] Devin L. Jindrich, Aruna D. Balakrishnan, and Jack T. Dennerlein. Effects of keyswitch design and finger posture on finger joint kinematics and dynamics during tapping on computer keyswitches. *Clinical Biomechanics*, 19(6):600–608, 2004.

[100] Ulla Johansson-Sköldberg, Jill Woodilla, and Mehves Çetinkaya. Design thinking: Past, present and possible futures. *Creativity and Innovation Management*, 22(2):121–146, 2013.

[101] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. Touchscreen typing as optimal supervisory control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2021. Association for Computing Machinery.

[102] Florian Kadner, Yannik Keller, and Constantin Rothkopf. AdaptiFont: Increasing individuals' reading speed with a generative font model and Bayesian optimization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2021. Association for Computing Machinery.

[103] Hiroyuki Kajimoto, Naoki Kawakami, Taro Maeda, and Susumu Tachi. Electro-tactile display with force feedback. In *EuroHaptics 2010: Haptics: Generating and Perceiving Tangible Sensations*, pages 285–291, 2001.

[104] Eunsuk Kang, Ethan Jackson, and Wolfram Schulte. An approach for effective design space exploration. In *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems: 16th Monterey Workshop 2010, Redmond, WA, USA, March 31–April 2, 2010, revised selected papers 16*, pages 33–54. Springer, 2011.

[105] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm[s]: Past, present, and future. *Multimedia Tools and Applications*, 80:8091–8126, 2021.

[106] D. Katz. *The world of touch*. Psychology Press, 1989.

[107] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

[108] Mohammad M. Khajah, Brett D. Roads, Robert V. Lindsey, Yun-En Liu, and Michael C. Mozer. Designing engaging games using Bayesian optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5571–5582, New York, NY, 2016. Association for Computing Machinery.

[109] Pankaj Khatak. Laser cutting technique: A literature review. *Materials Today: Proceedings*, 56:2484–2489, 2022.

[110] Jeong Ho Kim, Lovenoor Aulck, Michael C. Bartha, Christy A. Harper, and Peter W. Johnson. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied Ergonomics*, 45(6):1406–1413, 2014.

[111] Myunghee Kim, Ye Ding, Philippe Malcolm, Jozefien Speeckaert, Christoper J. Siviy, Conor J. Walsh, and Scott Kuindersma. Human-in-the-loop Bayesian optimization of wearable device parameters. *PLoS ONE*, 12(9):Paper e0184054, 2017.

[112] Sunjun Kim, Byungjoo Lee, and Antti Oulasvirta. Impact activation improves rapid button pressing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 571:1–571:8, New York, NY, 2018. Association for Computing Machinery.

[113] Sunjun Kim and Geehyuk Lee. Haptic feedback design for a virtual button along force-displacement curves. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 91–96, New York, NY, USA, 2013. Association for Computing Machinery.

[114] Sunjun Kim and Geehyuk Lee. Haptic feedback design for a virtual button along force-displacement curves. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology – UIST '13*, pages 91–96, New York, NY, 2013. Association for Computing Machinery.

[115] Sunjun Kim, Jeongmin Son, Geehyuk Lee, Hwan Kim, and Woohun Lee. TapBoard: Making a touch screen keyboard more touchable. In *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 553–562, New York, NY, 2013. Association for Computing Machinery.

[116] Jacob H. Kirman. Tactile perception of computer-derived formant patterns from voiced speech. *The Journal of the Acoustical Society of America*, 55(1):163–169, 1974.

[117] Nicolas Knudde, Joachim van der Herten, Tom Dhaene, and Ivo Couckuyt. GPFlowOpt: A Bayesian optimization library using TensorFlow. *arXiv preprint – arXiv:1711.03845*, 2017.

[118] Werner A. König, Jens Gerken, Stefan Dierdorf, and Harald Reiterer. Adaptive pointing – design and evaluation of a precision enhancing technique for absolute pointing devices. In *Human–Computer Interaction – INTERACT 2009*, pages 658–671. Springer, 2009.

[119] Sadanori Konishi and Genshiro Kitagawa. *Information criteria and statistical modeling*. Springer, 1st edition, 2007.

[120] Yuki Koyama and Masataka Goto. BO as assistant: Using Bayesian optimization for asynchronously generating design suggestions. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, 2022. Association for Computing Machinery.

[121] Yuki Koyama, Issei Sato, and Masataka Goto. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics*, 39(4):Paper 88, 2020.

[122] Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics*, 36(4):Paper 48, 2017.

[123] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[124] Alexander Lavin, David Krakauer, Hector Zenil, Justin Gottschlich, Tim Mattson, Johann Brehmer, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, et al. Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint arXiv:2112.03235*, 2021.

[125] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research methods in human–computer interaction*. Morgan Kaufmann, 2017.

[126] Byungjoo Lee, Mathieu Nancel, Sunjun Kim, and Antti Oulasvirta. AutoGain: Gain function adaptation with submovement efficiency optimization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2020. Association for Computing Machinery.

[127] Byungjoo Lee and Antti Oulasvirta. Modelling error rates in temporal pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1857–1868, New York, NY, 2016. Association for Computing Machinery.

[128] Jaeyeon Lee, Jaehyun Han, and Geehyuk Lee. Investigating the information transfer efficiency of a 3x3 watch-back tactile display. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1229–1232, New York, NY, 2015. Association for Computing Machinery.

[129] Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):Paper 500, 2018.

[130] Seungyon "Claire" Lee and Thad Starner. BuzzWear: Alert perception in wearable tactile displays on the wrist. In *CHI: '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 433–442, New York, NY, 2010. Association for Computing Machinery.

[131] James R. Lewis, Kathleen M. Potosnak, and Regis L. Magyar. Keys and keyboards. In *Handbook of human–computer interaction*, pages 1285–1315. Elsevier, 1997.

[132] Yi-Chi Liao, Yen-Chiu Chen, Liwei Chan, and Bing-Yu Chen. Dwell+: Multi-level mode selection using vibrotactile cues. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 5–16, New York, NY, 2017. Association for Computing Machinery.

[133] Yi-Chi Liao, Yi-Ling Chen, Jo-Yu Lo, Rong-Hao Liang, Liwei Chan, and Bing-Yu Chen. EdgeVib: Effective alphanumeric character output using a wrist-worn tactile display. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 595–601, New York, NY, 2016. Association for Computing Machinery.

[134] Yi-Chi Liao, Sunjun Kim, Byungjoo Lee, and Antti Oulasvirta. Button simulation and design via FDVV models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2020. Association for Computing Machinery.

[135] Jeffrey S. Libby. The best games in life are free: Videogame emulation in a copyrighted world. *Suffolk University Law Review*, 36(3):843–861, 2002.

[136] Hongnan Lin, Liang He, Fangli Song, Yifan Li, Tingyu Cheng, Clement Zheng, Wei Wang, and HyunJoo Oh. FlexHaptics: A design method for passive haptic inputs using planar compliant structures. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2022. Association for Computing Machinery.

[137] Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 161–168, 2014.

[138] Zimo Liu, Jingya Wang, Shaogang Gong, Huchuan Lu, and Dacheng Tao. Deep reinforcement active learning for human-in-the-loop person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6122–6131, 2019.

[139] J. Derek Lomas, Jodi Forlizzi, Nikhil Poonwala, Nirmal Patel, Sharan Shodhan, Kishan Patel, Ken Koedinger, and Emma Brunskill. Interface design optimization as a multi-armed bandit problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4142–4153, New York, NY, 2016. Association for Computing Machinery.

[140] Granit Luzhnica and Eduardo Veas. Optimising encoding for vibrotactile skin reading. In *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2019. Association for Computing Machinery.

[141] Karam Y. Maalawi and Mervat A. Badr. Design optimization of mechanical elements and structures: A review with application. *Journal of Applied Sciences Research*, 5(2):221–231, 2009.

[142] I. Scott MacKenzie. *Human–computer interaction: An empirical research perspective*. Morgan Kaufmann, San Francisco, CA, 1st edition, 2013.

[143] Richard W. Marklin and Mark L. Nagurka. Measurement of stiffness and damping characateristics of computer keyboard keys. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 44(6):678–681, 2000.

[144] Joaquim R. R. A. Martins and Andrew B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013.

[145] Joaquim R. R. A. Martins and Andrew Ning. *Engineering design optimization*. Cambridge University Press, 2021.

[146] Thomas H. Massie and J. K. Salisbury. The PHANToM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Dynamic Systems and Control Division*, pages 295–301, 1994.

[147] M. Matscheko, A. Ferscha, A. Riener, and M. Lehner. Tactor placement in wrist worn wearables. In *International Symposium on Wearable Computers (ISWC) 2010*, pages 1–8, 2010.

[148] M. Meilgaard and B. Carr. *Sensory evaluation techniques*. CRC Press, Boca Raton, FL, 4th edition, 2007.

[149] Christoph Meinel and Larry Leifer. *Design thinking research*. Springer, 2012.

[150] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95(3):340–370, 1988.

[151] Jeremy Michalek, Ruchi Choudhary, and Panos Papalambros. Architectural layout design optimization. *Engineering Optimization*, 34(5):461–484, 2002.

[152] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[153] Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[154] Riccardo Moriconi, Marc Peter Deisenroth, and K. S. Sesh Kumar. High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109:1925–1943, 2020.

[155] G. Moy, C. Wagner, and R. S. Fearing. A compliant tactile display for teletaction. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 4, pages 3409–3415, 2000.

[156] W. A. Munson and Mark B. Gardner. Standardizing auditory tests. *The Journal of the Acoustical Society of America*, 22(5):675–675, 1950.

[157] David Murphy. Hacking public memory: Understanding the multiple arcade machine emulator. *Games and Culture*, 8(1):43–53, 2013.

[158] Ken Nakagaki, Daniel Fitzgerald, Zhiyao (John) Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. InFORCE: Bi-directional "force" shape display for haptic interaction. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 615–623, New York, NY, 2019. Association for Computing Machinery.

[159] Ken Nakagaki, Daniel Fitzgerald, Zhiyao (John) Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. inFORCE: Bi-directional "force" shape display for haptic interaction. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 615–623, New York, NY, 2019. Association for Computing Machinery.

[160] Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon. Mid-air pointing on ultra-walls. *ACM Transactions on Computer–Human Interaction*, 22(5):Paper 21, 2015.

[161] E. Bruce Nauman. *Chemical reactor design, optimization, and scaleup*. John Wiley & Sons, 2008.

[162] Allen Newell, J. C. Shaw, and Herbert A. Simon. The process of creative thinking. In M. Wertheimer H. Gruber, G. Terrell, editor, *Contemporary approaches to creative thinking*. Atherton Press, New York, NY, 1967.

[163] Anh-Tuan Nguyen, Sigrid Reiter, and Philippe Rigo. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 113:1043–1058, 2014.

[164] Jens Brehm Bagger Nielsen, Jakob Nielsen, and Jan Larsen. Perception-based personalization of hearing aids using Gaussian processes and active learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):162–173, 2015.

[165] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic Books, 2013.

[166] Donald A. Norman. *The psychology of everyday things*. Basic Books, New York, NY, 1988.

[167] Donald A. Norman and Stephen W. Draper. *User centered system design; New perspectives on human–computer interaction*. Lawrence Erlbaum Associates, US, 1986.

[168] Aleks Oniszczak and I. Scott MacKenzie. A comparison of two input methods for keypads on mobile devices. In *Proceedings of the Third Nordic Conference on Human–Computer Interaction*, pages 101–104, 2004.

[169] Alex F. Osborn. *Applied imagination: Principles and procedures of creative thinking*. Scribner, New York, NY, 1953.

[170] Carolina Osorio and Michel Bierlaire. A simulation-based optimization framework for urban transportation problems. *Operations Research*, 61(6):1333–1345, 2013.

[171] A. Oulasvirta, N. R. Dayama, M. Shiripour, M. John, and A. Karrenbauer. Combinatorial optimization of graphical user interface designs. *Proceedings of the IEEE*, 108(3):434–464, 2020.

[172] Antti Oulasvirta. User interface design with combinatorial optimization. *Computer*, 50(1):40–47, 2017.

[173] Antti Oulasvirta, Niraj Ramesh Dayama, Morteza Shiripour, Maximilian John, and Andreas Karrenbauer. Combinatorial optimization of graphical user interface designs. *Proceedings of the IEEE*, 108(3):434–464, 2020.

[174] Antti Oulasvirta, Sunjun Kim, and Byungjoo Lee. Neuromechanics of a button press. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 508:1–508:13, New York, NY, 2018. Association for Computing Machinery.

[175] Chaeyong Park, Jeongwoo Kim, Dong-Geun Kim, Seungjae Oh, and Seungmoon Choi. Vibration-augmented buttons: Information transmission capacity and application to interaction design. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2022. Association for Computing Machinery.

[176] Michal Piovarči, Danny M. Kaufman, David I. W. Levin, and Piotr Didyk. Fabrication-in-the-loop co-optimization of surfaces and styli for drawing haptics. *ACM Transactions on Graphics*, 39(4):Paper 116, 2020.

[177] Matthew J. Pitts, Mark A. Williams, Tom Wellings, and Alex Attridge. Assessing subjective response to haptic feedback in automotive touchscreens. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 11–18, New York, NY, 2009. Association for Computing Machinery.

[178] Hasso Plattner, Christoph Meinel, and Larry Leifer. *Design thinking: Understand–improve–apply*. Springer Science & Business Media, 2010.

[179] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The Go-Go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, pages 79–80, New York, NY, 1996. Association for Computing Machinery.

[180] Ivan Poupyrev and Tadao Ichikawa. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages and Computing*, 10(1):19–35, 1999.

[181] William R. Provancher and Nicholas D. Sylvester. Fingerpad skin stretch increases the perception of virtual friction. *IEEE Transactions on Haptics*, 2(4):212–223, 2009.

[182] Angel R. Puerta, Henrik Eriksson, John H. Gennari, and Mark A. Musen. Model-based automated generation of user interfaces. In *Proceedings of the National Conference on Artificial Intelligence, 1994*, pages 471–477, 1994.

[183] Robert G. Radwin and One-Jang Jeng. Activation force and travel effects on overexertion in repetitive key tapping. *Human Factors*, 39(1):130–140, 1997.

[184] R. Venkata Rao, Vimal J. Savsani, and D. P. Vakharia. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3):303–315, 2011.

[185] Majken K. Rasmussen, Esben W. Pedersen, Marianne G. Petersen, and Kasper Hornbæk. Shape-changing interfaces: A review of the design space and open research questions. In *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 735–744, New York, NY, 2012. Association for Computing Machinery.

[186] Rim Razzouk and Valerie Shute. What is design thinking and why is it important? *Review of Educational Research*, 82(3):330–348, 2012.

[187] David Rempel, Elaine Serina, Edward Klinenberg, Bernard J. Martin, Thomas J. Armstrong, James A. Foulke, and Sivakumaran Natarajan. The effect of keyboard keyswitch make force on applied force and finger flexor muscle activity. *Ergonomics*, 40(8):800–808, 1997.

[188] Bruno H. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969–992, 2005.

[189] Hendrik Richter, Ronald Ecker, Christopher Deisler, and Andreas Butz. HapTouch and the 2+1 state model: Potentials of haptic feedback on touch based in-vehicle information systems. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 72–79, New York, NY, 2010. Association for Computing Machinery.

[190] Horst W. Rittel and Melvin M. Webber. Wicked problems. *Man-Made Futures*, 26(1):272–280, 1974.

[191] J. K. Salisbury and M. A. Srinivasan. Phantom-based haptic interaction with virtual objects. *IEEE Computer Graphics and Applications*, 17(5):6–10, 1997.

[192] E. Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, 112(2):223–229, 1990.

[193] Tahir Sağ and Mehmet Çunkaş. A tool for multiobjective evolutionary algorithms. *Advances in Engineering Software*, 40(9):902–912, 2009.

[194] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *Clinical Orthopaedics and Related Research*, 467:1074–1082, 2017.

[195] Steven L. Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[196] Matthias W. Seeger, C. K. Williams, and N. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pages 254–261, 2003.

[197] Jessica Zeitz Self, Radha Krishnan Vinayagam, James Thomas Fry, and Chris North. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *HILDA '16: Proceedings of the Workshop on Human-in-the-Loop Data Analytics*, 2016.

[198] Amar Shah and Zoubin Ghahramani. Pareto frontier learning with expensive correlated objectives. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1919–1927, 2016.

[199] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[200] Nurhalida Shahrubudin, Te Chuan Lee, and R. J. P. M. Ramlan. An overview on 3D printing technology: Technological, materials, and applications. *Procedia Manufacturing*, 35:1286–1296, 2019.

[201] Xing Shi, Zhichao Tian, Wenqiang Chen, Binghui Si, and Xing Jin. A review on building energy efficient design optimization from the perspective of architects. *Renewable and Sustainable Energy Reviews*, 65:872–884, 2016.

[202] Wei Shyy, Nilay Papila, Rajkumar Vaidyanathan, and Kevin Tucker. Global design optimization for aerodynamics and rocket propulsion components. *Progress in Aerospace Sciences*, 37(1):59–118, 2001.

[203] Russell Smith. Open Dynamics Engine, 2007.

[204] Jasper Snoek. *Bayesian optimization and semiparametric models with applications to assistive technology*. PhD thesis, University of Toronto, Ontario, CA, 2013.

[205] Aiguo Song, Jia Liu, and Juan Wu. Softness haptic display device for human–computer interaction. In Ioannis Pavlidis, editor, *Human computer interaction*, chapter 16. IntechOpen, Rijeka, Croatia, 2008.

[206] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[207] Seiya Takei, Ryo Watanabe, Ryuta Okazaki, Taku Hachisu, and Hiroyuki Kajimoto. Presentation of softness using film-type electro-tactile display and pressure distribution measurement. In Ki-Uk Kyung Hiroyuki Kajimoto, Hideyuki Ando, editor, *Haptic interaction: Perception, devices and applications*, pages 91–96. Springer Japan, Tokyo, JP, 2015.

[208] H. Z. Tan, S. Choi, F. W. Y. Lau, and F. Abnousi. Methodology for maximizing information transmission of haptic devices: A survey. *Proceedings of the IEEE*, 108(6):945–965, 2020.

[209] Hong Z. Tan, Nathaniel I. Durlach, William M. Rabinowitz, Charlotte M. Reed, and Jonathan R. Santos. Reception of Morse code through motional, vibrotactile, and auditory stimulation. In *Perception and Psychophysics*, pages 1004–1017, 1997.

[210] Ravindra V. Tappeta, Somanath Nagendra, and John E. Renaud. A multidisciplinary design optimization approach for high temperature aircraft engine components. *Structural Optimization*, 18:134–145, 1999.

[211] David Ternes and Karon E. Maclean. Designing large sets of haptic icons with rhythm. In *Proceedings of the 6th International Conference on Haptics: Perception, Devices and Scenarios*, pages 199–208, Berlin/Heidelberg, Germany, 2008. Springer-Verlag.

[212] Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

[213] Kashyap Todi, Gilles Bailly, Luis Leiva, and Antti Oulasvirta. Adapting user interfaces with model-based reinforcement learning. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, 2021. Association for Computing Machinery.

[214] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. Sketchplore: Sketch and explore with a layout optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 543–555, New York, NY, 2016. Association for Computing Machinery.

[215] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[216] Emre Ugur, Erol Sahin, and Erhan Öztop. Affordance learning from range data for multi-step planning. In *Proceedings of the Ninth International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 177–184, 2009.

[217] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

[218] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. A survey of user-centered design practice. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 471–478, New York, NY, 2002. Association for Computing Machinery.

[219] Jane X. Wang. Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences*, 38:90–95, 2021.

[220] Cara Gonzalez Welker, Alexandra S. Voloshina, Vincent L. Chiu, and Steven H. Collins. Shortcomings of human-in-the-loop optimization of an ankle–foot prosthesis emulator: A case series. *Royal Society Open Science*, 8(5):Paper 202020, 2021.

[221] Stephan Wensveen and Ben Matthews. Prototypes and prototyping in design research. In *The Routledge companion to design research*, pages 262–276. Routledge, 2014.

[222] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, 2006.

[223] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for Bayesian optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[224] Alan Wing and A. B. Kristofferson. Timing of interresponse intervals. *Attention, Perception, and Psychophysics*, 13:455–460, 1973.

[225] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022.

[226] Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. Accelerating human-in-the-loop machine learning: Challenges and opportunities. In *DEEM'18: Proceedings of the Second Workshop on Data Management for End-to-End Machine Learning*, 2018.

[227] Kenta Yamamoto, Yuki Koyama, and Yoichi Ochiai. Photographic lighting design with photographer-in-the-loop Bayesian optimization. In *UIST '22: Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, 2022. Association for Computing Machinery.

[228] Xue Yan and P. Gu. A review of rapid prototyping technologies and systems. *Computer-Aided Design*, 28(4):307–318, 1996.

[229] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44:945–956, 2019.

[230] Ziyu Yao, Xiujun Li, Jianfeng Gao, Brian Sadler, and Huan Sun. Interactive semantic parsing for if-then recipes via hierarchical reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2547–2554, 2019.

[231] Dani Yogatama and Gideon Mann. Efficient transfer learning method for automatic hyperparameter tuning. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, ABSTATS 2014*, pages 1077–1085, 2014.

[232] Jaesik Yun, Youn-kyung Lim, Kee-Eung Kim, and Seokyoung Song. Interactivity Crafter: An interactive input–output transfer function design tool for interaction designers. *Archives of Design Research*, 28(3):21–37, 2015.

[233] Juanjuan Zhang, Pieter Fiers, Kirby A. Witte, Rachel W. Jackson, Katherine L. Poggensee, Christopher G. Atkeson, and Steven H. Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2017.

[234] Shanshan Zhang, Lihong He, Eduard Dragut, and Slobodan Vucetic. How to invest my time: Lessons from human-in-the-loop entity extraction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2305–2313, 2019.

[235] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE, 2020.

[236] Zhaoyuan Ma, D. Edge, L. Findlater, and H. Z. Tan. Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard. In *2015 IEEE World Haptics Conference (WHC)*, pages 220–227, 2015.

[237] Yijun Zhou, Yuki Koyama, Masataka Goto, and Takeo Igarashi. Generative melody composition with human-in-the-loop Bayesian optimization. *arXiv preprint arXiv:2010.03190*, 2020.

Errata

Publication I

Yi-Chi Liao, John J Dudley, George B Mo, Chun-Lien Cheng, Liwei Chan, Antti Oulasvirta, Per Ola Kristensson. Interaction Design With Multi-objective Bayesian Optimization. *IEEE Pervasive Computing*, 22, 1, 29-38, January 2023.

Interaction Design with Multi-Objective Bayesian Optimization

Yi-Chi Liao
Aalto University

John J. Dudley
University of Cambridge

George B. Mo
University of Cambridge

Chun-Lien Cheng
National Yang Ming Chiao Tung University

Liwei Chan
National Yang Ming Chiao Tung University

Antti Oulasvirta
Aalto University

Per Ola Kristensson
University of Cambridge

***Abstract*—Interaction design typically involves challenging decision making that requires designers to consider multiple parameters and careful trade-offs between various objectives. This article examines how AI can facilitate the process of interaction design by offloading some of the complex decision making required of designers. We study how multi-objective Bayesian optimization can be used to support designers when creating a tactile display for smartwatches. We present the results of a study that explores how such human-AI collaboration afforded by multi-objective Bayesian optimization can be exploited by designers, and the advantages and disadvantages this solution offers over conventional design practice.**

INTERACTION DESIGN is challenging and a part of this challenge is the complexity of the design space, which is only exacerbated in pervasive computing applications. The user's experience and performance when interacting with a system is often governed by a large number of configurable design parameters. Adding still further complication is the fact that design objectives, such as performance, accuracy or comfort, may be in tension or in direct conflict with each other and thus demand explicit or implicit trade-offs to be decided by the designer. However, exhaustively examining the design space and assessing the impact of various design configurations is rarely feasible and in practice designers rely on their past experiences, design know-how and established conventions to arrive at a particular design.

For example, consider the task of designing a distinguishable set of vibration-based notifica-

 Published by the IEEE Computer Society

tions on a smartwatch. Intuitively, a distinguishable set of vibration cues could be delivered by simply selecting distinct combinations of vibration durations and amplitudes. If we assume that the objectives in this design problem are to maximize cue recognition accuracy as well as the total number of distinguishable cues, how exactly should one go about methodically exploring the space of possible designs?

One conventional ad-hoc approach to design space exploration is to manually select promising design candidates and sequentially evaluate these with participants. However, this process is difficult to perform systematically, in particular when the design space is large and may contain multiple competing objectives. Further, there is a risk that the design space exploration process inadvertently absorbs the biases and subjective preferences of the designer.

This article explores an alternative approach where the designer is partnered with an AI agent that intelligently proposes designs for evaluation. We study designers' experiences when interacting with such an AI agent to design a pervasive computing user interface—a tactile display for a smartwatch. We focus on the designer's experience, as opposed to the end-user's experience, as we see a critical need to preserve a designer's appreciation of the design space. This focus reflects the fact that novel interactions may be designed in isolation but must typically be integrated into an actual application. At this integration stage it is useful if the designer can exploit their appreciation of the design space to understand how particular design decisions might be influenced by the much broader demands of the application.

Among a potential range of techniques that may be suitable for AI-assisted interaction design we investigate multi-objective Bayesian optimization (MOBO). Bayesian optimization is a method for performing optimization on black-box functions. In interaction design we can view the mapping between the design parameters and the quality or performance of the design as such a black-box function. Bayesian optimization constructs a surrogate model of this unknown function and leverages this model to intelligently determine a promising new point in the design space to evaluate. Each new observation of the design space serves to improve the surrogate model. As such, Bayesian optimization might be particularly suitable for guiding interaction design for three reasons. First, it efficiently pursues promising designs while ignoring demonstrably poor regions of the design space. Second, it relies on very few initial assumptions compared to other optimization methods. Third, it is able to accommodate the high levels of noise typically inherent in observations of human behavior.

However, despite its suitability, MOBO is not widely used in interaction design. To address this gap, and to highlight the potential of MOBO in interaction design, we study the experience and performance of designers working in collaboration with MOBO. Further, we go beyond simply exploring whether MOBO is useful by also investigating the ease or difficulty with which designers can conceptualize and integrate MOBO into a design problem. We examine the relative merits and deficiencies of MOBO-supported human-AI collaboration by asking designers to complete the same design task using both their own preferred design approach as well as with the assistance from MOBO. This approach enables the study participants to directly reflect on the experience of using MOBO compared to the design process that they might otherwise use. The design problem we pose to designers is to design a maximally expressive and distinguishable set of vibration cues for delivering smartwatch notifications. A successful design will rely on the ability of the designer to carefully balance competing objectives and a relatively large design space.

Three key takeaways emerge from the results of this study. First, a MOBO procedure results in designs that exhibit very similar performance as designs generated by designers using their own self-elected design strategies. Second, MOBO significantly reduces designers' perceived overall workload while successfully assisting the designers in identifying promising design candidates. Third, designers may become detached from the design process when typical aspects of their role are subsumed by MOBO.

Overall, MOBO appears to be a promising complementary AI-assisted design method suitable when design problems are complex and have multiple competing objectives. However, HCI research should study methods that help designers retain ownership and agency in the process.

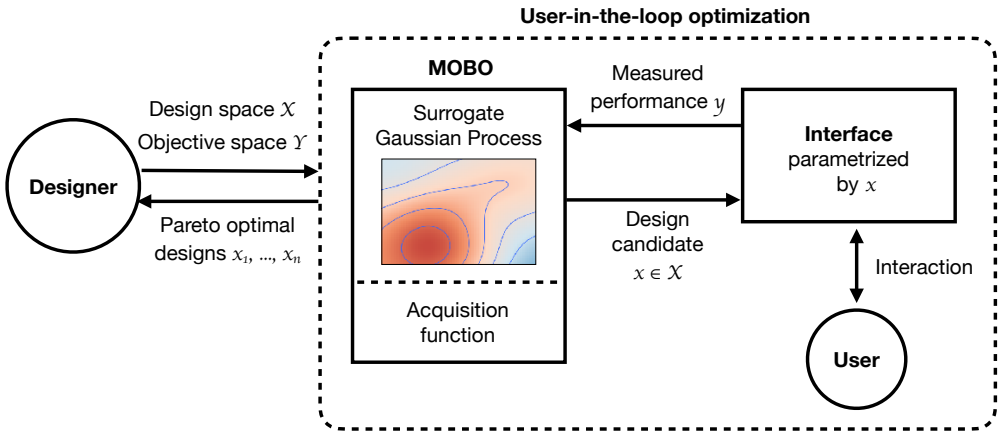


Figure 1: Illustration of the multi-objective Bayesian optimization procedure. The designer initializes the process by defining the design parameters (X) and the objectives (Y) and then commences the user-in-the-loop optimization. The MOBO procedure will propose design candidates, $x \in X$, for evaluation in the search for the Pareto-optimal parameter settings. The interaction technique behaviors are updated for each x and the user's performance and/or subjective experience is measured and translated into objective values y. The multi-objective Bayesian optimization procedure updates its proxy model—a Gaussian Process model based on the observed $\{x, y\}$ sets—which it then uses to propose a new design candidate. Once the optimization is complete, the designer gathers and selects from among the Pareto-optimal designs.

BACKGROUND

Bayesian optimization is a machine learning technique for performing optimization of black box functions. It works by constructing a surrogate model of the unknown function and selecting new points to evaluate with reference to this model. Each new observation improves the estimate of the model and thereby the understanding of where promising points are likely to lie. Bayesian optimization has been successfully applied to HCI design problems to refine interfaces [1] and to support image [2, 3] and animation customization [4]. Our work sits within a much broader body of research focused on AI-supported decision making [5, 6, 7].

These prior studies used Bayesian optimization for a single objective. However, it is also possible to perform multi-objective Bayesian optimization. As is true for all forms of multi-objective optimization, the outcome is no longer a single optimum but rather a set of optima representing various trade-offs between the different objectives. This set of optima is referred to as the Pareto front. In the context of interaction design, the Pareto front represents a set of possible designs for which one objective cannot be improved without degradation of another objective. For example, a design involving a large set of distinct vibration-based notifications will maximize the potential for transferring information but may also give rise to more frequent misrecognitions. Conversely, a design with fewer distinct cues may support a high recognition accuracy but may limit the amount of information that can be transferred. Both designs may sit on the Pareto front and reflect optimal operating points exposing different trade-offs in the design objectives.

The general form of the multi-objective Bayesian optimization procedure is summarized in Figure 1. The designer initializes the procedure by determining a suitable parameterization for the design problem as well as the relevant design objectives. When the user-in-the-loop optimization process begins, new designs are presented and evaluated by the user. The results of these evaluations are fed back to refine the surrogate

model and improve the selection of new designs. After some fixed number of iterations or set time period, the designer can inspect the surrogate model and extract the designs corresponding to the Pareto front.

The study presented in this article applies multi-objective Bayesian optimization to the task of designing a wearable haptic display. Investigating and optimizing the information transmission via haptic sensation has been a long-standing goal in haptic research. The problem has gained increasing attention with the emergence of smartwatches [8, 9, 10, 11]. Prior work has investigated a single vibrotactor generating vibrations with various durations, frequencies, and amplitudes [12, 13, 13, 14] and delivering temporal-spatial patterns on skin with 2D tactors [10, 15, 16].

Recent work by [17] conducted a related user study in which 40 novice designers were asked to create optimal designs for a 3D touch interaction either manually or by an optimizer-led approach using MOBO. Although the research protocols between this workshop study and [17] are similar, there are several major differences in the experimental setup and the study goals. [17] investigated the benefits of MOBO for novice designers, where they searched for Pareto-optimal designs by assessing the performance of the interaction designs generated on themselves without external study participants. However, our study examines MOBO when applied in a scenario closer to what a typical designer would do in terms of evaluating an interaction design with several study participants. We only invited experienced designers and HCI researchers to take on the role of the designer, and we provided study participants to the designers during the entire design process for the designer to evaluate the designs generated.

STUDY

The study sought to answer two key questions. First, to what extent can MOBO be applied by designers to support their design work? Second, how does a MOBO procedure compare to the standard practice of designers? The study, which was conducted as a workshop, engaged participants in the task of designing the vibration cues for a haptic wearable display. The target participant group for the workshop was individuals with some experience of interaction design. For clarity we subsequently refer to these participants as *designers*. Each designer was allocated two further participants who served as a proxy for users that the designer could use to test designs with.

The workshop was structured such that each designer completed the same design exercise twice: first using their own preferred design strategy, and second in collaboration with MOBO. Given the workshop approach, we focus primarily on observations regarding the designers' experience of the design procedure as opposed to the quality of the design outcomes.

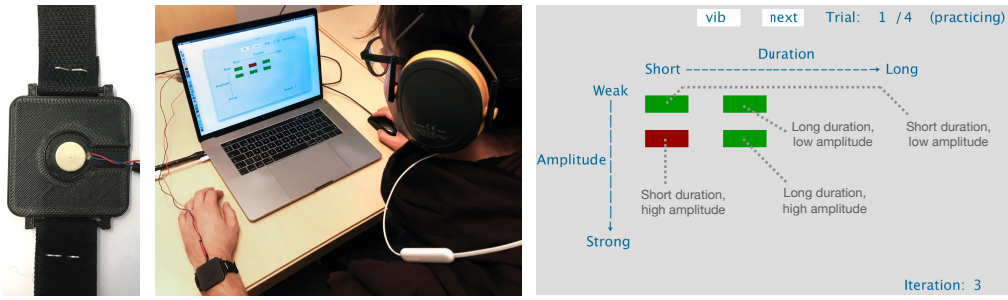
Design Task

Designers were asked to tackle a classic problem in human-computer interaction—designing vibration cues for a haptic wearable display. One possibility for conveying different messages to the user with a single-tactor is designing a set of vibrations that contain unique combinations of vibration duration and amplitude. More unique combinations allows for more messages to be conveyed. However, at some point the different messages become difficult to distinguish. As messages become more difficult to distinguish, more recognition errors will occur.

We purposefully constrained this design space by restricting the design problem to selecting an appropriate range for vibration duration and amplitude, as well as the number of distinct levels of duration and amplitude over that range. We fixed the maximum duration of vibration to 1 second and the maximum amplitude to 1.45 g. We then parameterized the design of the wearable tactile display according to the four design parameters (X) (and permitted ranges/values) summarized below:

- Minimum duration time of the vibration [50 ms, 950 ms]
- Number of discrete vibration duration levels $\{1, 2, 3, 4\}$
- Minimum amplitude of vibration [0 g, 1.45 g]
- Number of discrete amplitude levels $\{1, 2, 3, 4\}$

Designers participating in the workshop were presented with the following design brief: "*You are asked to design the vibration cues for a newly released smartwatch. Your task is to use*



(a) (b) (c)

Figure 2: a) The haptic display prototype. b) Study setup with participant wearing the prototype and interacting with the user interface. c) Detail of the user interface.

the four design parameters available to create a set of vibrations that can achieve both high recognition accuracy and high information transfer rate. Other products on the market deliver up to three different vibration cues, and your final design is expected to at least outperform these competitors."

This brief refers to the two objectives (Y) which are to govern the optimization process: the Information Transfer (IT) rate and recognition accuracy. IT represents an estimate of the channel capacity for a specific stimuli set, that is, the effective bits of information transferred per stimuli (see Tan et al. [18, section II, A] for the calculation of IT). We calculate recognition accuracy as $\frac{n_{correct}}{n}$ where $n_{correct}$ is the number of trials in which the correct response is matched with the given stimulus, and n is the total number of trials.

Participants

Eight designers were recruited for the study (age 23–31; five females). Four designers were Ph.D. students recruited from local universities, and all of them conducted research in human-computer interaction. One designer worked as a professional user experience designer and had rich experience in conducting user research and data analysis. The remaining three designers were master's students majoring in design programs at local universities and they all had prior experience in running user studies.

16 proxy users (age 21–33; seven females) were also recruited for the workshop so that each designer was assigned two dedicated study participants. Both the designers and the study participants were compensated based on the number of hours they participated in the workshop. The hourly compensation was €11.

Procedure

The designers completed the same design exercise using both the MOBO procedure and their own chosen design procedure. For clarity, we refer to these two alternative design procedures as distinct conditions, even though in practice the designer-elected workflow may have been different for each designer. Four of the designers completed the workshop using the MOBO procedure first and their chosen design procedure second, while the remaining designers undertook the conditions in reverse order. This ensured that the conditions were counterbalanced in an attempt to control for learning effects.

The designers were each given a prototype of the wearable haptic smartwatch (see Figure 2a) and assigned two dedicated additional study participants who served as proxy end-users to test their designs with. The study setup was as pictured in Figure 2b. The user interface shown in Figure 2c allowed designers to both instruct their two study participants about the mapping between the vibration cues and the notification intent, as well as to capture the ability of the two study participants to recognize cues. The set of possible cues for the current designs was displayed as a grid of boxes. A box further towards the right represented a cue with a longer vibration duration while a box further towards the bottom represented a cue with a larger vibration

amplitude. The interface supported both a practice and a test mode. In the practice mode, participants were randomly presented with a specific cue and the corresponding box in the interface would turn red. In the test mode, participants clicked on the box that they believed corresponded to the cue presented.

The designers were given three hours for each condition. If needed, they could ask the workshop facilitator (one of the authors) for technical support and clarification about the design brief. The procedure differed slightly for each condition as detailed below.

Designer-Elected Procedure. The designers could directly choose a particular set of parameter values in the interface and present these to their two study participants. Each design iteration would include both practice and test modes. When the test session of a particular design configuration was completed, the designer was shown the recognition accuracy and the number of cues. The designers familiarized themselves with the task and devised a study plan in the first hour of the session. The study plan was then executed in the remaining two hours. At completion, the designers were asked to specify their preferred final design.

MOBO Procedure. The MOBO procedure illustrated in Figure 1 was implemented as a set of API calls that could be used by designers. The MOBO procedure itself employed the CEIPV (Correlated Expected Improvement in Pareto hyperVolume) acquisition function proposed by Shah and Ghahramani [19]. Hyperpameters were tuned at each step of the optimization process by maximizing the log likelihood. Designers configured the procedure by specifying the parameterization, objectives and some basic hyperparameters and then initiated the user-in-the-loop optimization process. We provided a simple method that could combine the observations obtained from the two dedicated participants into a single model for extraction of the Pareto optimal designs. The workshop facilitator (one of the authors) spent an hour introducing designers to the MOBO procedure and guiding them through the setup of the optimization process. The designers then had two hours to complete the design task, at the end of which they selected one final design from the Pareto frontier. The final Pareto frontier was generated from all collected data and presented as a 2D plot with each axis corresponding to one of the objectives.

After completing both conditions the designers were presented with their derived designs and their performances. The designers then completed a NASA-TLX and a System Usability Scale (SUS) questionnaire. Finally, the designers participated in an interview in which they were invited to reflect on the advantages and disadvantages in the two conditions' different procedures. In total, each designer was engaged for approximately seven hours.

RESULTS

Overall the designs arrived at using either a designer-elected procedure or the MOBO procedure were very similar. The mean accuracy and number of distinct vibration cues for the designs produced in the designer-elected workflow were 0.863 ($SD = 0.078$) and 6.125 ($SD = 1.36$), respectively. The mean accuracy and number of distinct vibration cues for the designs produced in the MOBO procedure were 0.883 ($SD = 0.08$) and 6.125 ($SD = 2.031$), respectively.

A major point of contrast between the MOBO procedure and designer-elected procedure is that the Pareto frontier generated by MOBO enables a structured interpretation of the influence of the design parameters. We observed that the optimal designs identified by MOBO had minimum vibration durations and minimum vibration amplitudes at the lower end of the feasible range. This aligns with intuition given that lower minimum vibration durations and amplitudes will produce more distinct individual cues. We also observed that the precise balance between accuracy and information transfer was chiefly influenced by the combined variation of the number of vibration duration levels and the number of vibration amplitude levels: reducing the number of levels for both parameters produced more accurate designs.

The designers used a variety of design strategies when choosing their own design procedure, which highlights the complexity of tackling the design problem using a conventional design procedure. Below we summarize the various design strategies that were applied by designers in the non-MOBO condition.

Divide-and-conquer and subsequently in-

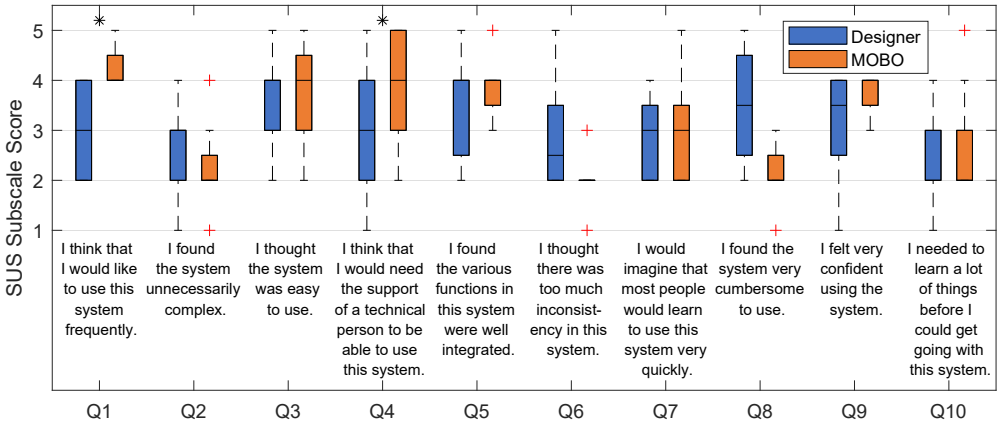


Figure 3: Boxplots showing the ratings from the eight designers for the SUS questionnaire, comparing the design-elected workflow with the MOBO procedure. The red crosses mark outliers, which are defined as beyond $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$. The star $(*)$ symbol indicates significant difference at $p < 0.05$. Significant differences were observed in Q1 and Q4 suggesting that designers would be more willing to use the MOBO procedure frequently than their own selected design strategy, but that they would be more likely to require technical support.

creasing the complexity of the task. Two designers (*D1* and *D2*) applied a divide-and-conquer approach by focusing on certain design parameters in isolation first and subsequently increasing the vibration cue count. They involved their two study participants throughout this process. Both designers determined the acceptable minimum duration and amplitude through testing with the two study participants as the first step. Then they tested 1 (duration level) $\times$ 2 (amplitude level) and 2 (duration level) $\times$ 1 (amplitude level) designs with the participants, which yielded a near perfect recognition rate but a relatively low cue count. The designers then gradually increased the number of vibration cues in the design until both the recognition rate and the cue count met the requirements.

Divide-and-conquer and subsequently decreasing the complexity of the task. D3 and *D6* applied a divide-and-conquer approach as described above, but then subsequently gradually decreased the vibration cue count. First, they derived the minimum duration and amplitude by testing with their two study participants. Then, they tested the largest possible vibration set, 4 (duration level) $\times$ 4 (amplitude levels), yielding a high cue count but a low recognition rate. The designers then incrementally decreased the number of vibration cues and tested each design generated with the participants for every change. The design process stopped when a satisfactory recognition rate was achieved.

Divide-and-conquer and local search. D5 and *D7* spent 30 minutes deriving a starting design with a medium number of vibration cues: one designer started with a 3 (duration level) $\times$ 2 (amplitude level) design and the other designer started with a 3 (duration level) $\times$ 3 (amplitude level) design. These initial designs were relatively close to their final designs in terms of their design parameters. The designers followed a strategy similar to performing a "local search" where they fine-tuned the design parameters until they were satisfied.

Self-evaluating approach. D8 largely evaluated the generated designs without involving their two study participants. After an hour of self-testing, *D8* derived three final design candidates. *D8* then invited the two study participants to evaluate these final design candidates and thereafter selected the design candidate with the highest preference.

Focus group: D4 adopted a "focus group"

approach. The designer allocated both the two study participants and themselves five minutes to create their own designs independently. Then, all three people (the two study participants and the designer) evaluated all the designs made by the others. Then the group discussed how to improve the design, followed by another round of evaluation using the same approach. After two iterations, the group narrowed down the selection to two final designs.

The total number of designs evaluated varied across designers (*D1*: 6; *D2*: 6; *D3*: 5; *D4*: 8; *D5*: 9; *D6*: 6; *D7*: 5; *D8*: 3).

Usability and Workload

We assess significant differences in the overall and subscale ratings for usability and perceived workload using a Wilcoxon Signed-Rank Test based on participant matched samples. Figure 3 summarizes the results of the SUS questionnaire. The mean SUS score for the designer-elected procedure was 54.375 ($sd = 15.51$) and 64.375 ($sd = 13.48$) for the MOBO procedure. A Wilcoxon Signed-Rank Test revealed no statistical difference between the overall SUS scores ($Z = -1.022, p > 0.05$).

Although the overall scores were not significantly different, there were statistical differences when examining individual questions. Based on a Wilcoxon Signed-Rank Test, there were statistically significant differences in the responses to Q1 ($Z = -2.06, p < 0.05$) and Q4 ($Z = -2.97, p < 0.05$). This suggests that designers would indeed like to use the MOBO procedure (Q1) but that they would require more technical support (Q4).

Figure 4 summarizes the results of the NASA-TLX questionnaire. The mean workload for the designer-elected procedure was 62.67 ($sd = 16.36$) and 45.17 ($sd = 12.4$) for the MOBO procedure. A Wilcoxon Signed-Rank Test revealed a significant difference between the overall workloads for the two conditions ($Z = -2.38, p < 0.05$). In other words, the MOBO procedure significantly reduced mental workload. Further, based on Wilcoxon Signed-Rank Tests, the designer-elected strategy elicited statistically higher mental demand ($Z = -2.197, p < 0.05$), physical demand ($Z = -2.06, p < 0.05$), temporal demand ($Z = -2.366, p < 0.05$), and frustration ($Z = -2.527, p < 0.05$).

The SUS and NASA-TLX results show that the MOBO procedure delivered a usable alternative design process. Further, the MOBO procedure generally induced a lower cognitive load, frustration and mental and physical demand than the designer-elected procedures.

User Experience

All designers agreed that the MOBO-assisted design procedure largely reduced the effort involved in interpreting the data and making decisions throughout the design process. As noted by *D1*: "The design space is very large. The manual design process would take a lot of time and effort to explore until reaching an acceptable design. [...] [MOBO] removed that effort of making decisions and trial-and-errors." *D8* shared a similar perspective: "In the manual design process, I needed to carefully consider tuning the design so that there will be an improvement. It is a demanding process and I constantly felt uncertain. However, MOBO just did that for me and I'm happy with the final results." *D2* also pointed out that the MOBO procedure helped to reduce not only the mental load but also the physical load: "[With MOBO] I did not need to manipulate the interface and the device, nor interact with the participants much. I simply needed to instruct the participants what to do, and the results would be generated, which is a big advantage."

The designers agreed with the benefits of having a series of proposed designs, as represented by the Pareto front. *D1* observed: "If I changed my weights of the objectives and wanted to search for another design, I might need to invest another 30 minutes to reach that point. [The output of the MOBO procedure] showed all the designs along the line (Pareto front) and I could just pick one from them. From this perspective, I find [the MOBO procedure] much more efficient because it searches not just one final outcome but multiple." *D5* further mentioned: "I set some kind of priority at the beginning of the design. For example, the recognition rate is more important than the information transfer, and I want to achieve 95% of accuracy. However, during the [designer-elected] design process, I might gain new knowledge about the interaction, and would like to change the weight of the two objectives, which would

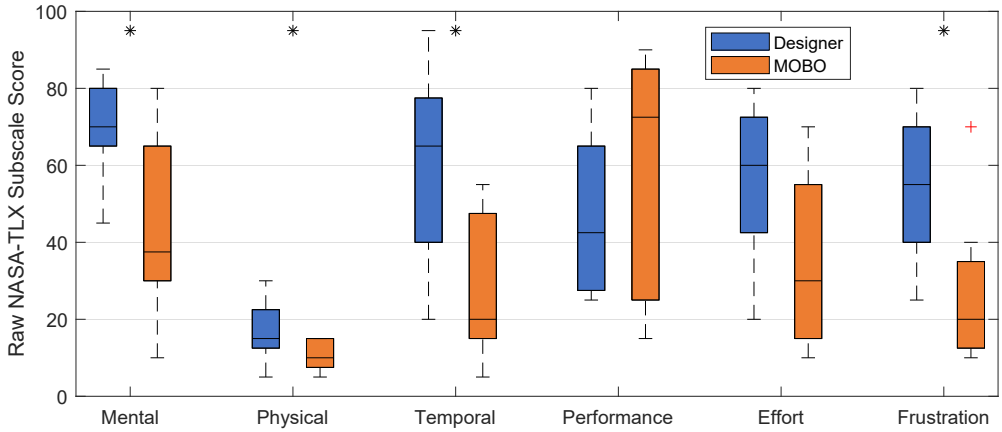


Figure 4: Boxplots showing the ratings from the eight designers for the NASA-TLX questionnaire, comparing the manual designer-elected workflow with the MOBO procedure. The red crosses mark outliers, which are defined as beyond $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$. The one-star $(*)$ symbol indicates significant difference at $p < 0.05$. The MOBO procedure yielded significantly lower mental, physical and temporal demand and significantly lower frustration.

force me to change the direction of search. The [MOBO procedure] can avoid this kind of hassle because it explores all the directions and provides all the possibilities."

All designers pointed out that their self-selected approaches induced higher frustration and temporal demand. *D4* stated: "This search can go on forever. I can always change something and lead to a different performance. I always feel uncertain, not knowing if this change will improve or not, and this is frustrating. Also, because I need to deliver a design within the certain amount of time, so I was somehow stressed." *D1* also shared that: "I was not sure if this design is good enough, so I felt it is more temporal demanding. On the other hand, when using [MOBO], I simply needed to assign one hour [...] to each participant and collected the results. It is much simpler and relaxing."

Overall, all the designers provided positive feedback to the final designs derived by the MOBO procedure. *D5* said, "I am surprised to see the results [produced by MOBO] are so good. It is well aligned with my own design. Also, the whole Pareto front looks promising to me and in line with my expectation." Further, *D7* used the Pareto-optimal designs generated by the MOBO procedure as a baseline to compare to: "Checking the designs made by [the MOBO procedure] is more like a reassuring step. To be honest, I trust the results made by the system more than the ones made by me. The Pareto front indicated a very systematical search."

In addition to the positive aspects of the MOBO procedure, the designers also highlighted several drawbacks. First, setting up the MOBO procedure required some level of programming experience. However, this issue can be mitigated if the designer is supported by a developer or potentially resolved by developing more elaborate tools for MOBO-assisted design in the future. As *D3* pointed out: "As a developer, I do not find major difficulty of using [MOBO], but I would assume a designer without coding background will need some kind of technical support." *D8* expressed a very similar view.

Another major identified drawback of adopting the MOBO procedure is losing the opportunity to receive qualitative feedback from users regarding a particular design. *D2* mentioned: "I might want to learn the feedback from the participants, such as how do they feel about this design, and how can I improve from that. But [the MOBO procedure] did not give me this possibility." *D5* also mentioned: "I felt I lost the involvement if I fully rely on the [MOBO-assisted procedure]. I

fully trusted the final results; they look promising and reasonable to me. Still, I would appreciate to talk to the participants and learn from them how they felt."

DISCUSSION AND CONCLUSIONS

Overall, the final designs produced by the MOBO procedure were found to be comparable in terms of performance to those generated by a designer-elected procedure. However, we found that the MOBO procedure significantly reduced the designers' perceived workload, which was also echoed by the qualitative data we gathered from the interviews.

The variety of approaches taken by the designers when they were allowed to choose their own design strategy demonstrates that there is no single obvious design strategy that can generate designs that guarantee any performance specification. We conjecture, as a consequence, the designers had to spend time and effort in devising a specific strategy for the design problem. In contrast, the MOBO procedure provided a single systematic approach for tackling the design problem. The designer was thus freed from the burden of conceiving a strategy and selecting a specific study plan, as the optimizer lead the generation of new designs to explore. Although the results of this study are broadly in line with Chan et al. [17], the investigation presented in this article of MOBO versus the designer's own selected design strategy highlights the additional cognitive burden the designer encounters when devising a custom experimental methodology. Further, we more precisely examine the experience of the designer as opposed to the merged designer/end-user that was the subject of Chan et al.'s study.

There were primarily two downsides to using the MOBO procedure. First, it requires some experience in programming or the ability to call on a developer to support the setup of the system. This added complexity may in itself have negatively impacted the experience of using the MOBO procedure. Second, fully relying on a MOBO procedure may lead to the designer becoming detached from study participants and unable to utilize subjective feedback to drive the design process. Our work therefore provides further motivation for the research community to develop human-AI interactions that promote positive synergy [20].

The MOBO procedure in this article is an example of how AI can be fruitfully used as a partner with a designer to exploit a complex design space with competing objectives for a pervasive computing application. We view it as highly encouraging that the AI-assisted designs closely match the outcomes of the designers' plethora of self-elected approaches in the study, which is a strong indication that partnering a designer with an AI can at the very least result in comparable results and with a significantly reduced perceived workload. However, we also note the many avenues for future work.

First, we see fruitful future work in *tool design* that alleviates the need for programming expertise. Further, such tools should ideally incorporate techniques for clearly explaining designs proposed to users, their trade-offs and implications, and the inherent uncertainty associated with their measured performance.

Second, there is a challenge in avoiding the MOBO procedure resulting in the designer becoming too detached from study participants. This well-known problem in automation is expected but will need to be tackled for a MOBO procedure to ultimately achieve widespread adoption.

Third, pervasive computing user interface designs, in general, are particularly challenging as they often rely on context and/or uncertain sensing. It would be interesting to consider a more complex design problem, for example, an interface that, in part, relies on working within a specific environment for successful interaction.

ACKNOWLEDGMENTS

The authors would like to thank all the participants for attending the user study. John J. Dudley and Per Ola Kristensson were supported by a grant from the Engineering and Physical Sciences Research Council (EPSRC EP/S027432/1).

REFERENCES

1. J. J. Dudley, J. T. Jacques, and P. O. Kristensson, "Crowdsourcing Interface Feature Design with Bayesian Optimization," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery,

May 2019, pp. 1–12. [Online]. Available: https://doi.org/10.1145/3290605.3300482
2. Y. Koyama, I. Sato, D. Sakamoto, and T. Igarashi, "Sequential line search for efficient visual design optimization by crowds," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 48:1–48:11, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073598
3. Y. Koyama, I. Sato, and M. Goto, "Sequential gallery for interactive visual design optimization," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 88:88:1–88:88:12, Jul. 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392444
4. E. Brochu, T. Brochu, and N. de Freitas, "A Bayesian interactive optimization approach to procedural animation design," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '10. Goslar, DEU: Eurographics Association, Jul. 2010, pp. 103–112.
5. V. Lai, C. Chen, Q. V. Liao, A. Smith-Renner, and C. Tan, "Towards a science of human-ai decision making: A survey of empirical studies," 2021. [Online]. Available: https://arxiv.org/abs/2112.11471
6. H. Liu, V. Lai, and C. Tan, "Understanding the effect of out-of-distribution examples and interactive explanations on human-ai decision making," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW2, oct 2021. [Online]. Available: https://doi.org/10.1145/3479552
7. Y. Zhang, Q. V. Liao, and R. K. E. Bellamy, "Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, ser. FAT* '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 295–305. [Online]. Available: https://doi.org/10.1145/3351095.3372852
8. R. H. Gault, "Progress in experiments on tactual interpretation of oral speech," *The Journal of Abnormal Psychology and Social Psychology*, vol. 19, no. 2, pp. 155 – 159, 1924.
9. J. H. Kirman, "Tactile perception of computer-derived formant patterns from voiced speech," *The Journal of the Acoustical Society of America*, vol. 55, no. 1, pp. 163–169, 1974. [Online]. Available: https://doi.org/10.1121/1.1928145
10. S. C. Lee and T. Starner, "Buzzwear: Alert perception in wearable tactile displays on the wrist," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 433–442. [Online]. Available: https://doi.org/10.1145/1753326.1753392
11. M. Matscheko, A. Ferscha, A. Riener, and M. Lehner, "Tactor placement in wrist worn wearables," in *International Symposium on Wearable Computers (ISWC) 2010*, 2010, pp. 1–8.
12. H. Z. Tan, N. I. Durlach, W. M. Rabinowitz, C. M. Reed, and J. R. Santos, "Reception of morse code through motional, vibrotactile, and auditory stimulation," in *Perception and Psychophysics*, 1997, p. 1004–1017.
13. D. Ternes and K. E. Maclean, "Designing large sets of haptic icons with rhythm," in *Proceedings of the 6th International Conference on Haptics: Perception, Devices and Scenarios*, ser. EuroHaptics '08. Berlin, Heidelberg: Springer-Verlag, 2008, p. 199–208. [Online]. Available: https://doi.org/10.1007/978-3-540-69057-3_24
14. S. Brewster and L. M. Brown, "Tactons: Structured tactile messages for non-visual information display," in *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28*, ser. AUIC '04. AUS: Australian Computer Society, Inc., 2004, p. 15–23.
15. J. Lee, J. Han, and G. Lee, "Investigating the information transfer efficiency of a 3x3 watch-back tactile display," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1229–1232. [Online]. Available: https://doi.org/10.1145/2702123.2702530
16. Y.-C. Liao, Y.-L. Chen, J.-Y. Lo, R.-H. Liang, L. Chan, and B.-Y. Chen, "Edgevib: Effective

alphanumeric character output using a wrist-worn tactile display," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 595–601. [Online]. Available: https://doi.org/10.1145/2984511.2984522

17. L. Chan, Y.-C. Liao, G. B. Mo, J. J. Dudley, C.-L. Cheng, P. O. Kristensson, and A. Oulasvirta, "Investigating positive and negative qualities of human-in-the-loop optimization for designing interaction techniques," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2022. [Online]. Available: https://doi.org/10.1145/3491102.3501850
18. H. Z. Tan, S. Choi, F. W. Y. Lau, and F. Abnousi, "Methodology for maximizing information transmission of haptic devices: A survey," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 945–965, 2020.
19. A. Shah and Z. Ghahramani, "Pareto Frontier Learning with Expensive Correlated Objectives," in *International Conference on Machine Learning*. PMLR, Jun. 2016, pp. 1919–1927, iSSN: 1938-7228. [Online]. Available: http://proceedings.mlr.press/v48/shahc16.html
20. A. Campero, M. Vaccaro, J. Song, H. Wen, A. Almaatouq, and T. W. Malone, "A test for evaluating performance in human-computer systems," *arXiv preprint arXiv:2206.12390*, 2022.

Yi-Chi Liao is a Ph.D. candidate in the User Interfaces group at Aalto University, Finland. His research focuses on using Bayesian Optimization to assist UI design and reinforcement learning to model human interactions with physical interfaces. He has previously interned at Meta Reality Labs and holds Bachelor's and Master's degrees from National Taiwan University. Contact him at yi-chi.liao@aalto.fi or visit his personal website at http://yichiliao.com for more information about his research.

John J. Dudley is an associate teaching professor in the Department of Engineering at the University of Cambridge and a postdoctoral associate of Jesus College, Cambridge. His research focuses on the design of interactive systems that dynamically adapt to user needs and behaviors. Contact him at jjd50@cam.ac.uk.

George B. Mo recently graduated with a BA and MEng in Engineering from the University of Cambridge. He was a member of the Intelligent Interactive Systems group at the University of Cambridge during the execution of this work.

Chun-Lien Cheng was a student at the National Yang Ming Chiao Tung University, Taiwan during the execution of this work.

Liwei Chan is an associate professor in the Department of Computer Science at the National Yang Ming Chiao Tung University, Taiwan. His research interests include human-computer interaction, interaction design for AR/VR, and haptic user interfaces. Contact him at liweichan@cs.nycu.edu.tw.

Antti Oulasvirta is a professor of user interfaces with Aalto University, Finland. He leads the Interactive AI programme at the Finnish Center for AI. His work focuses on computational methods in human-computer interaction, including interactive ML, user modeling, and simulation. Contact him at antti.oulasvirta@aalto.fi.

Per Ola Kristensson is Professor of Interactive Systems Engineering in the Department of Engineering at the University of Cambridge and a Fellow of Trinity College, Cambridge. He leads the Intelligent Interactive Systems group, which belongs to the Engineering Design Centre. He is also a co-founder and co-director of the Centre for Human-Inspired Artificial Intelligence at the University of Cambridge. Contact him at pok21@cam.ac.uk.

Publication II

Liwei Chan, Yi-Chi Liao, George B Mo, John J Dudley, Chun-Lien Cheng, Per Ola Kristensson, Antti Oulasvirta. Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques. In *2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, USA, April 2022.

Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques

Liwei Chan
liweichan@nycu.edu.tw
National Yang Ming Chiao Tung University
Taiwan

Yi-Chi Liao
yi-chi.liao@aalto.fi
Aalto University
Finland

George B. Mo
gm621@cam.ac.uk
University of Cambridge
United Kingdom

John J. Dudley
jjd50@cam.ac.uk
University of Cambridge
United Kingdom

Chun-Lien Cheng
liencc.cs08@nycu.edu.tw
National Yang Ming Chiao Tung University
Taiwan

Per Ola Kristensson
pok21@cam.ac.uk
University of Cambridge
United Kingdom

Antti Oulasvirta
antti.oulasvirta@aalto.fi
Aalto University
Finland

ABSTRACT

Designers reportedly struggle with design optimization tasks where they are asked to find a combination of design parameters that maximizes a given set of objectives. In HCI, design optimization problems are often exceedingly complex, involving multiple objectives and expensive empirical evaluations. Model-based computational design algorithms assist designers by generating design examples during design, however they assume a model of the interaction domain. Black box methods for assistance, on the other hand, can work with any design problem. However, virtually all empirical studies of this human-in-the-loop approach have been carried out by either researchers or end-users. The question stands out if such methods can help designers in realistic tasks. In this paper, we study Bayesian optimization as an algorithmic method to guide the design optimization process. It operates by proposing to a designer which design candidate to try next, given previous observations. We report observations from a comparative study with 40 novice designers who were tasked to optimize a complex 3D touch interaction technique. The optimizer helped designers explore larger proportions of the design space and arrive at a better solution, however they reported lower agency and expressiveness. Designers guided by an optimizer reported lower mental effort but also felt less creative and less in charge of the progress. We conclude that human-in-the-loop optimization can support novice designers in cases where agency is not critical.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501850

CCS CONCEPTS

• **Human-centered computing → Systems and tools for interaction design**.

KEYWORDS

Interface Design; Bayesian Optimization; Human-in-the-loop Optimization; Multi-objective Optimization; Haptics; Touch

ACM Reference Format:
Liwei Chan, Yi-Chi Liao, George B. Mo, John J. Dudley, Chun-Lien Cheng, Per Ola Kristensson, and Antti Oulasvirta. 2022. Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3491102.3501850

1 INTRODUCTION

One central problem in design is that of finding a satisfactory operating point in a multidimensional design space, one that balances trade-offs between relevant design objectives (e.g., [7, 13]). Such an operating point can be obtained using different strategies. A common strategy is relying on prior experience, intuition, and a bit of trial and error. Under such a strategy, the designer explores the space by gradually searching for suitable parameter values and assessing the observed trade-offs between the objectives. This approach can be effective when the design space is simple or familiar. However, as a method, it is not reliable. It is sensitive to the level of skill and prior-experience of the designer as well as the complexity of the design problem at hand. Moreover, it scales poorly and offers no guarantees that all reasonable options have been considered. An emerging alternative strategy which we study in this paper is to use an optimization-driven design method in which exploration is guided by a search algorithm [3, 15, 24, 46, 47, 52]. An optimization-driven design method guides the designer in their design space exploration and may offer various tools to inform final

design selection. In this paper we contrast these two approaches in an empirical study in order to report on the various positive and negative qualities of human-in-the-loop optimization.

Both the designer-led and optimization-driven strategies have conceivable advantages and disadvantages, thereby offering a rich collection of hypotheses worthy of examination. With complete freedom over the exploration of the design space, the designer is likely to have a stronger sense of agency which may deliver greater engagement in the task. A recent study of designers' expectations about data-driven design raised the loss of agency as a concern [22]. On the other hand, a potential drawback of the designer-led approach is that exploration of the design space is either consciously or subconsciously constrained by preconceived notions held by the designer. These preconceptions may be accurate, in which case constraints applied on exploration yield greater efficiency. Empirical research has exposed biases that limit the creative capability, such as confirmation bias [26], as well as a tendency for *fixation*, or 'blind adherence with a solution' [30, 56], which the literature suggests is hard to break [1]. Promising regions of the design space can be missed and outcomes fail to deviate significantly from those arrived at early in the process by the designer. We hypothesize that optimization-driven design may help to address problems such as design fixation but at the cost of designer agency and engagement. Optimization-driven design also serves to mitigate sensitivity to the expertise and prior experience of the individual designer which in turn may deliver more consistent outcomes when engaging a group of designers of different skill and experience levels.

This paper contributes to empirical research on computational methods for designers. Our focus is on a HCI-related design task relevant for the development of interactive systems and interaction techniques. Our anecdotal evidence is that relatively few papers presenting interactive systems at CHI, the premier venue of the HCI field, explore their parameter spaces systematically. We recognized the three following strategies described below. First, potential design parameters can be assigned or eliminated by extrapolating from evidence presented in the literature. HiveFive [38], for example, is a VR visualization technique that was optimized by first referencing a biological theory of bee swarming to substantially narrow down the search range for each parameter, and second, fixing values in a pilot study (with three people). Second, a divide and conquer strategy can be employed in which parameters are tackled one by one. For example, in Body Follows Eye [50], an interaction technique that guides users' posture change in VR was optimized over a series of six sub-studies where each determined the threshold for one of the six motion types. Third, sometimes the dimensionality of the problem is simplified with a mathematical model. For example, ErgonomicsTouch [53] exploits the so-called Hermite curve to amplify the user's hand movements into a larger movement for increasing physical comfort while preserving ownership. It was optimized by reducing the dimensionality of the problem by identifying four parameters that determine the mapping curve, with respect to the objectives of accuracy, comfort, and ownership. Lower and upper bounds for amplifications were then determined empirically with a pilot study with five users.

To better understand the pitfalls and perks of optimization-driven design in contrast to a designer-led approach, we conducted a study with 40 novice designers. We hypothesized that novices might benefit the most from computational assistance, especially to achieve a degree of directedness and organization when exploring designs [11]. To account for learning effects across the two conditions, we used a between-subjects protocol assigning 20 participants to each condition and examined both the quality of design outcomes and the designers' subjective experience of designing. The specific optimization technique we employed in the optimization-driven condition was Multi-Objective Bayesian Optimization (MOBO). Bayesian optimization has shown significant potential in HCI design problems and offers an efficient method for exploring design spaces that are poorly understood by the designer at the outset. To make this investigation concrete, designers are given a non-trivial design task involving the selection of parameters characterizing the behavior and haptic feedback of a 3D touch interaction in virtual reality to maximize efficiency and accuracy. This design task involves two competing objectives for which the relationship to the controllable design parameters is unclear. It therefore ensures a degree of challenge for designers and MOBO alike.

In summary, the core contribution in this paper is the empirical investigation of the positive and negative qualities of designer-led and optimization-driven design in a study with novice designers. We found that the optimization-driven design of the 3D touch interaction technique delivers a superior outcome in terms of reducing spatial error but at the cost of the subjective experience of agency and ownership. Furthermore, optimization-driven design using MOBO promotes wider exploration of the design space helping to mitigate detrimental design fixation.

2 RELATED WORK

Designing better interaction techniques is a long-standing topic within the HCI researcher and practitioner community. This has motivated the development of various strategies and tools to support the designer in this process. Papers in this vein in HCI typically demonstrate their new method or tool by highlighting improvements in the design outcomes but less commonly examine the secondary impact on the design process and the designer's experience. In this paper we seek to understand how the interaction technique design process is influenced by the tools made available to the designer. Specifically, we examine the advantages and disadvantages provided by human-in-the-loop optimization using Bayesian methods.

Below, we briefly review the related work to provide insight into the design process involving optimization methods within HCI. We first cover the broader topic of data-driven optimization before examining interaction design with human-in-the-loop optimization and multi-objective optimization. Finally we review prior work utilizing Bayesian optimization specifically to support the design process.

2.1 Data-Driven Optimization

One viable approach to improving interaction techniques is to leverage data collected on the whole or sub-tasks involved. An example of this approach is provided by Feit et al. [18] who collected eye tracking data from 80 people performing a calibration task. Feit et al. [18] demonstrated an optimization procedure leveraging this

data to select optimal filter parameters and inform the design of gaze interfaces in terms of target sizes.

Captured data may also be combined with relatively simple empirical models such as Fitts' Law to optimize various interactive elements such as hierarchical menus [19, 42] and keyboard layouts [4, 17, 58]. SUPPLE [21] takes a related approach in optimizing interface designs based on specified device constraints and user activity traces. Deep neural networks modelling user performance when interacting with vertical menus [39] have also been leveraged to drive optimization [14]. These various approaches may involve a degree of designer involvement to determine the feasible design space and interpret outputs, but the optimization process itself is largely offloaded to the computer.

Although not necessarily involving explicit optimization, data-driven methods leveraging deep learning have shown recent promise. GUIGAN [59] employs a generative adversarial network (GAN) fed with a large dataset of real Android application graphical user interfaces (GUIs) to construct a generative model for creating novel application GUIs. The quality of GUIGAN-generated GUIs is evaluated in the paper but there is no investigation of how the generative model can assist or influence the design process for designers. Also employing deep learning, Guo et al. [24] introduce Vinci which applies a variational autoencoder to construct a generative model for advertising posters. Critically, the Vinci system takes user input in the form of a product category, product image, and tagline text. These inputs condition the generative process and are incorporated into the generated poster. Various features of the Vinci system were evaluated with both novice and expert designers with generally favorable outcomes, particularly in terms of the tool's efficiency in generating a large number of design alternatives. Nevertheless, concerns were raised by designers in terms of the "controllability, comprehensibility, and predictability" of the design process using Vinci.

2.2 Human-in-the-Loop Optimization and Multi-objective Optimization

Human-in-the-loop optimization refers to the process in which the optimization process is steered by human input, for instance through training feedback and observed human behavior to a set of input parameters. This process has been extensively applied to HCI design tasks, for example in MenuOptimizer [3] where the designer is assisted during the task of combinatorial optimization of menus, and DesignScape [46] where layout suggestions for position, scale, and alignment of elements are interactively suggested to the designer. Other design tools that have a human-in-the-loop aspect include Sketchplore [52] where real-time design optimization is integrated into a sketching tool; Forte [9], in which designers can directly iterate on fabrication shape design through topology optimization; in Kapoor et al. [32], where the behavior of classification systems can be iteratively refined by designers to support more intuitive behavior; and in Lomas et al. [41], where the arrangement of game elements is iteratively adjusted for increased user performance. Overall, these tools all feature the central aspect of human interaction where the human actively participates during the optimization process to generate better designs. In broad terms, this human-in-the-loop paradigm of design is an evolution of the line of work introduced by [44] which aims to enhance the efficiency of the interface design process by automatically generating the code for the interface after demonstration of the interface specifications.

Yannakakis et al. [55] introduce the concept of player modeling in which a computational model is constructed of the cognitive, behavioral, and affective states of the player of a game. This constructed model may be dynamically updated in-game based on observations of user inputs and, in turn, used to drive changes in gameplay and game content. This general approach has been used to adjust game mechanics to maintain a challenging gaming experience for players [12, 54]. With a focus on designers as opposed to players, Guzdial et al. [25] explore co-creation with an agent for game level design and identify various potential roles for an agent in this design process, e.g., the agent portrayed as a friend, collaborator, student or manager. Liapis et al. [40] provide a review of related mixed-initiative methods applied to procedural content generation in game design.

Multi-objective optimization for interaction design serves as a special case for optimization-based design where instead of one objective to optimize over, there are now multiple objectives. As there is no longer one defined optimum for multiple objectives, the concept of Pareto optimality is important, where a design is considered to be Pareto optimal if no individual objective can be enhanced by changing the design parameters without resulting in at least one individual objective worse off. Multi-objective optimization aims to search for Pareto optimal designs so that an optimal trade-off between competing objectives is found. In HCI, multi-objective optimization has been applied to touchscreen keyboard design to trade-off speed, familiarity, and improved spell checking [17], multi-finger input for mid-air text entry [51], and linkage design for a haptic interface [28]. Many algorithms and computational methods have been applied for multi-objective optimization, including aggregating the different objectives into one via a linear weighted sum [51], grid-based methods [17], evolutionary-based methods [34], and Bayesian optimization [29]. In this paper, we seek to assess one specific multi-objective optimization algorithm, namely Bayesian optimization, in a human-in-the-loop context to explore the benefits and drawbacks as compared to the designer-led process, as it shows great potential in HCI design as detailed in Section 2.3.

2.3 Bayesian Optimization

Bayesian optimization is a machine learning technique for facilitating the optimization of unknown and/or difficult-to-evaluate functions. It works by iteratively refining a surrogate model representing the function and intelligently selecting new test points to evaluate by balancing between exploration of the design space and exploitation of regions where the designs are particularly promising. A major strength of Bayesian optimization is that the surrogate model is leveraged to ensure efficient search of the design space. Bayesian optimization is therefore well suited to interaction technique design problems where the relationship between design parameters and user performance and/or subject experience is unknown or easily modeled.

Bayesian optimization has been employed in HCI to tackle various design problems as a human-in-the-loop optimization method.

Early work by Brochu et al. [6] demonstrated how Bayesian optimization can incorporate direct feedback from users in a preference gallery to help determine desired parameters governing the appearance of animations. Koyama et al. [36, 37] use a similar approach to allow users to rapidly adjust the visual appearance of an image in line with some desired aesthetic. Bayesian optimization has also been used as a tool to determine game mechanic settings to maximize engagement [33], adjust font parameters to maximize reading speed [31] and adjust interface and interaction features to minimize task completion time [15]. These various studies serve to highlight how Bayesian optimization provides an effective tool to support design tasks in HCI. What is lacking, however, is a clear understanding of how design driven by this mechanism is experienced by or impacts the designer.

2.4 Summary

The various research efforts reviewed above offer a range of alternative tools and techniques for optimizing user interfaces and interactions. Lacking, however, is a clear understanding of how these various tools and strategies influence the design process and experience for designers. This paper seeks to address this gap in the literature by comparing the outcomes and experience of designing with and without assistance from Bayesian optimization. We focus on Bayesian optimization as the tool offered to designers given the significant advantages that have been demonstrated within the HCI domain in terms of its efficiency and its ability to handle black box optimization problems.

3 CASE: 3D TOUCH INTERACTION

Our empirical study focuses on a complex and realistic interaction technique case – 3D touch interaction – which is ubiquitously applied in virtual reality. Here, we compare two approaches: the designer-led and the optimizer-driven approach, and in this section, we outline the background of the interaction, the design space parameterization, and the design objective functions. In particular, we specifically chose this task as 1) target acquisition in 3D is an important problem in the domain of virtual reality, 2) the resulting performance of the interaction is easily observable to the user as the design parameters vary, and 3) it serves as a classic multi-objective design problem in HCI as we will detail in Section 3.1.

3.1 Background of 3D Touch Interaction

Target selection is a crucial, if not the most important, task for a virtual reality (VR) or an augmented reality (AR) application [2]. A great variety of VR and AR selection methods have been proposed [5] in mind with the challenge of the trade-off between speed and accuracy that was identified in early works [2]. Poupyrev categorized such selection techniques into the use of a virtual pointer or virtual hand metaphor [49]. A good 3D selection design should allow selection to be fast and accurate; however, searching for the good design candidates while satisfying both objectives is known to be a challenging design problem. Moreover, previous works showed that the control-to-display transfer function (including 2D and 3D selection) requires different numbers of parameters, which can range from two to ten [2, 20, 35, 43, 48]. Thus, the high-dimensional design space makes searching a promising design instance especially time-consuming and costly. For instance, previous approaches applied for designing 2D transfer functions are either based on a great amount of trial-and-error [7], which is a costly process, or by heuristics [45, 57], which requires prior domain expertise.

The *Go-Go technique* is a well-known 3D touch interaction design proposed by Poupyrev, which has been widely applied in VR and AR interactions [48]. Essentially, a hybrid control-to-display transfer function determines the virtual hand's position according to the physical hand's movement. Within a certain range, the transfer function follows a linear mapping, in which the virtual hand moves linearly based on the physical hand's position. Beyond this range, the transfer function follows a non-linear mapping, in which the virtual hand moves quadratically away according to the physical hand's position. This combination enables users to stably touch the objects that are closer to the body while being able to hit the targets that are beyond the physical hand's reach. Two parameters determine the switch of the mapping methods and the degree of the nonlinearity in the nonlinear schema.

Despite the number of the design parameters being relatively low, exhaustively searching the design space for the optimal design instance is not practical due to the challenges discussed above. While the 3D touch interaction design is timely and increasingly important, optimization of its design either based on manual parameter tuning done by a designer or an optimization algorithm has not been well documented or explored. For example, the Go-Go technique as described in the original paper recommends parameter settings without a proper rigorous justification [48]. In the following experiment, we selected the Go-Go technique as a base example to compare a human designer's iterative search and the Bayesian optimization workflow, with some add-on design parameters to allow greater selection accuracy and speed.

3.2 Parameterizing 3D Touch Interaction and the Objective Functions

3.2.1 3D Touch Interaction. The 3D touch interaction used in the later experiment is built on the original Go-Go technique with some modifications. The original Go-Go technique decided the chest position as the reference origin point. The arm vector r_r was obtained by subtracting the physical hand position to the chest then translating to the hand's coordinate and direction. In our experiment, we shifted the reference point to the shoulder, which captures more natural hand movements, as shown in Figure 1. We further defined 1 unit of the "operation range" as the distance between the origin (which is shoulder of the operating hand) and the hand when the arm is fully extended. The Go-Go technique's transfer function was then applied to calculate the virtual hand's position.

3.2.2 Design Parameters. There are two parameters in the original Go-Go technique – D and k – which jointly form the hybrid transfer function. D is the range which divides the linear and non-linear mapping, and k determines the scale of the nonlinear component. If the physical hand's distance is within the range D, the transfer function linearly maps the user's physical hand to the virtual hand along the same direction, where the real arm vector r_r is assigned to the Go-Go cursor r_c (Figure 1a). Once the physical hand moves

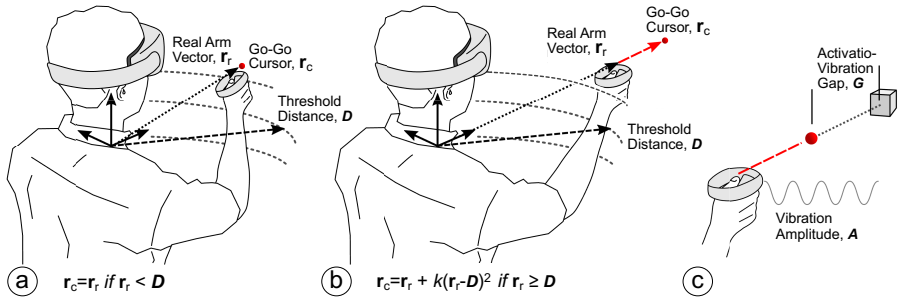


Figure 1: Our empirical study focuses on the task of improving the transfer function of the Go-Go technique. The technique calculates the virtual hand's position with the parameters D and k. (a) It maps the position linearly when the physical hand's distance is within the range D, or (b) non-linearly by a factor controlled by k when it moves beyond the range D. (c) In addition, the two parameters G and A for the activation-vibration gap and the vibration amplitude determine the vibrotactile feedback when the target is reached.

beyond the distance D, the nonlinear mapping allows the virtual hand to move much faster away from the origin (shoulder) along the direction of the physical hand by a factor controlled by k, with which the Go-Go cursor r_c is computed as $r_r + k(r_r - D)^2$ (Figure 1b). We directly took D and k as the design parameters for our 3D touch interaction, and set the ranges of these two parameters to be $D \in [0, 1]$ and $k \in [0, 0.5]$.

However, there are other parameters that will affect the 3D selection performance, including a vibration cue. This has been proven effective for enhancing efficiency and accuracy, and it has been applied to commercial devices. Following this direction, we look to add the simplest and most pervasive haptic feedback when the target is reached to enhance user performance—a vibrotactile cue. For a balanced design, we selected two parameters for vibrotactile feedback: the activation-vibration point, G, and the vibration intensity, A, as shown in Figure 1c. The duration of the feedback was fixed at 300 ms. We set the range of the activation-vibration point to activate at any point in the range of 15 cm before and 5 cm after touching a target. We also set the vibration amplitude to be within the maximum voltage level (3.1V), which led the vibration amplitude to be within 2.6g. All design parameters are summarized in Table 1.

3.2.3 Objective Functions. The objective functions refer to the metrics we aim to maximize or minimize during the design process. Following the discussion above, we considered two design metrics — completion time (speed) and spatial error (accuracy) in target acquisition — as our objective functions to be minimized. The first objective function, completion time, refers to the average duration between the moment a target is shown in the 3D experimental environment and the moment it is successfully touched by the virtual hand. The second objective, spatial error, is the maximum overshoot distance, which is the maximum Euclidean distance between the virtual hand and the target's 3D position if the virtual hand moves beyond the range of the target. If a participant touches the target without any overshoot occurring (the cursor did not go beyond the range of the target at all), the spatial error will remain zero.

Because the values of the completion time and spatial error have their own ranges, normalization is required before the optimization process. We converted these two metrics into two values which we refer to as *speed* and *accuracy* by linearly transforming the completion time ranged [1,600 ms, 900 ms] into to *speed* ranged [-1, 1] , and the spatial error ranged [1 cm, 0 cm] into the *accuracy* ranged [-1, 1]. Note that after the conversion, both the speed and accuracy objectives are now functions to be maximized instead (the higher value indicates better performance). The ranges of completion time and spatial error were decided from a pilot test conducted with eight participants.

3.2.4 Hyperparameter Setup for Bayesian Optimization. The Bayesian optimization in our implementation is built upon BoTorch[1], a PyTorch-enabled Bayesian Optimization library. This library is commonly used in many research projects, and it offers reliable performance and the flexibility of picking the Gaussian Process models and acquisition functions. The Gaussian Process we applied in the later experiment is the multi-output Gaussian Process. The acquisition function we applied is qEHVI, which represents the expected hypervolume increase, where we set $q = 1$ to ensure that after each iteration, a batch of size one is selected to be given to the designer for testing. Other hyperparameter settings include using 10 optimization restarts during the optimization of the acquisition function, 1024 as the number of restart candidates for the acquisition function optimization, and 512 as the number of Monte Carlo samples to approximate the acquisition function. These were selected to ensure good computational efficiency for each iteration of the optimization process.

4 EXPERIMENTAL METHOD

The goal of the experiment is to investigate positive and negative aspects of human-in-the-loop optimization by contrasting it to the designer-led approach. The metrics we used to analyze the results cover the design outcomes and a wide range of designer experiences including the perceived creativity and workload. The optimization

[1]https://botorch.org/

Table 1: The four design parameters for the 3D touch interaction design, with the ranges. All four design parameters are continuous.

Design Parameter	Description	Range
x_1: Distance Threshold, D	Division between linear and non-linear mappings.	$[0, 1]$
x_2: Scale Factor, k	Scale of the non-linear component.	$[0, 0.5]$
x_3: Activation-Vibration Gap, G	Cues when the target is reached.	[15 cm, -5 cm]
x_4: Vibration Amplitude, A	Vibrotactile feedback intensity.	[0 g, 2.6 g]

task consists of four design parameters left undetermined and two objectives to which the 3D touch interaction is set to be optimized during the design process.

In the designer-led condition, the search is progressed manually by actively exploring and refining design candidates. In contrast, the optimizer-driven condition follows a human-in-the-loop process in which a Bayesian optimizer leads the search for the designer; at the end, the designer determines the optimal designs from a set of Pareto optimal designs suggested by the optimizer. To avoid learning effects on the design target across experiment conditions, the experiment followed a between-subjects design. We measured the performance of designs produced in the two conditions, quantified the perceived creativity and workload using the Creativity Support Index [10] and NASA-TLX [27], and collected user feedback with a semi-structured interview. With the mixed-methods approach, we looked to understand the trade-offs for human-in-the-loop optimization as compared to the designer-led process.

4.1 Participants

We recruited 40 novice designers (20 F, 20 M), with a mean age of 22.2 years (sd: 2.4), via snowball sampling and through a Facebook group page dedicated to recruiting participants from a local university. Most participants were enrolled in a master's program with their expertise covering engineering, architecture, interaction, and education. Following the between-subjects design, they were randomly divided into the groups for the designer-led or optimizer-driven processes. All volunteered under informed consent and agreed to the recording and anonymized publication of results. They were compensated 20€ for their participation.

4.2 Apparatus

The apparatus mainly consisted of the 3D touch interaction. However, the interface to support the optimization process was customized according to the experimental condition for the designer-led or optimizer-driven processes.

4.2.1 The 3D Touch Interaction and Prototype. We built the 3D touch interaction in Unity 3D[2] with the Oculus Quest 2[3] and the companion hand controllers, as shown in Figure 2. Our prototype implementation matches closely to the original one in [49] with the minor changes listed in Section 3.2.2. To provide vibrotactile feedback on the controllers that can be precisely controlled, we added a vibration motor, Precision Microdrives 310-117[4] (rise time 97 ms), on the controller such that users can easily rest their thumb on the motor. The vibrotactile feedback was controlled via a DRV2605L driver and an Arduino Uno microprocessor. During the optimization task, participants were asked to sit on a legged chair so that a polar coordinate system can be easily maintained. We followed task arrangements used in [8] for 3D target acquisition. The three variables that determined target locations are: the inclination angle (30°, 45°, and 60°); the azimuth angle (0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°); and the radial distance to the target (0.5 units, 1 unit, 1.5 units, 2 units of the operation range). The fourth variable determines target widths (3 cm, 4 cm, and 5 cm). In total, there were 288 (3 inclination angles × 8 azimuth angles × 4 distances × 3 target widths) variations of movement trials, as illustrated in Figure 2d.

4.2.2 The Parameter Sliders and Evaluation Button. We offer parameter sliders and an evaluation button as shown in Figure 3a, which participants in the designer-led group use to adjust parameters for a new design and to initiate a formal evaluation of the design, respectively. Four parameter sliders are located at the lower right-hand side of the participant in VR, whose values correspond to the four parameters of the interaction. Any adjustment of the slider values directly applies the design parameters to the interaction. Since there is always a random target presented in the virtual space, participants can test the current design by simply selecting the target; subsequently, the next target appears for further testing. To initiate a formal evaluation of the current design, the participant presses the evaluation button below the parameter sliders. This enters a dedicated mode where these widgets disappear and the participant starts to follow a series of 36 trials randomly selected from the 288 variations while keeping an equal sampling across target distance and target width. The evaluation was completed when 36 trials were finished. Then, the averaged completion time and spatial error of the trials were computed and indicated on the objectives chart (detailed in subsection 4.2.3).

4.2.3 The Parameters and Objectives Charts. The parameters chart and objectives chart allow designers to keep track of all the designs that have gone through formal evaluation. The parameters chart contains a parallel coordinate plot of the designs evaluated, and the objectives chart contains 2D scatter plots of the corresponding objectives calculated from their formal evaluations. Once a formal evaluation is completed, the two charts are brought up for the participant to visualize the performance of the design under evaluation (Figure 3b). The data point in dark blue in the objectives chart indicates the most recent evaluation. Pressing the controller's menu button dismisses or invokes the charts. These charts also support interactive functions. For example, the two charts are interlinked: on selection of a data point, indicated in red in the objectives chart, the corresponding design in the parameters chart is highlighted in

[2]https://unity.com/
[3]https://www.oculus.com
[4]https://www.precisionmicrodrives.com/product/310-117-10mm-vibration-motor-3mm-type

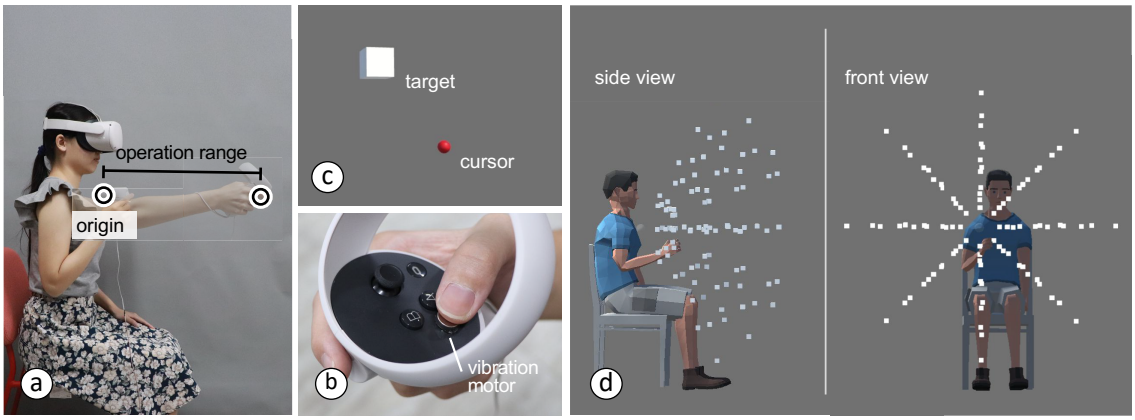


Figure 2: (a) The experiment setup for the 3D touch interaction adapted from the original Go-Go technique, and (b) the interaction enhanced with vibrotactile feedback via the vibrator added to the controller. (c) Participants acquire the target using a cursor (e.g., the virtual hand) with dwell-based selection. (d) All possible locations of targets.

red, and vice versa. Two floating text fields appear beside the selection to show detailed data of the evaluation. In addition, the charts also directly apply the selected design to the parameter sliders and thus the interaction, allowing designers to easily revisit previously evaluated designs.

4.2.4 The Bayesian Optimizer. In the optimizer-driven group, participants worked with the optimizer to determine optimal designs. The Bayesian optimizer was configured for optimizing the 3D touch interaction as described in Section 3.2.4.

4.3 Task

We created a realistic brief for proposing 3D touch interaction designs in the form of a one-page description with background and goals. Participants were prescribed as designers and were tasked to propose three optimal designs as the outcome of the design optimization.

In the designer-led group, participants led the design process by actively testing and evaluating designs using the parameter sliders, evaluation button, and the charts. They were instructed to conclude the designs within a time limit of 60 minutes. However, they could propose to end early when they were satisfied with the design outcome.

In the optimizer-driven group, participants worked with the optimizer in two stages – the design and decision stages – to conclude three optimal designs. In the design stage, the optimizer would propose in total forty designs; each required the participant to complete a formal evaluation by selecting 36 trials in sequence. After completing each evaluation, the design parameters and the design performance were displayed to the participant on the charts. The initial ten designs were randomly sampled by the Bayesian optimizer for optimization seeding. Completing the forty design evaluations entered the decision stage, where the participant was presented with the Pareto optimal designs (e.g., the designs connected by the red line on the objectives chart in Figure 4). They could test each of the Pareto optimal designs by selecting it. Then, they concluded the optimization process by selecting three designs from the Pareto optimal designs. As a result, the number of Pareto optimal designs could be fewer than three instances, in which case re-selection was allowed. In other words, if there was only one Pareto optimal design proposed, the three selected designs would be the same Pareto optimal design. From our study, the average number of Pareto optimal designs proposed is 3.3 ($sd = 1.5$) by the optimizer across participants.

4.4 Procedure

Figure 5 illustrates the study procedure. After briefing the study, the experimenter helped the participants wear the VR device, explained the parameters of the interaction, and allowed them to adjust the design parameters to observe the interaction behavior so as to familiarize the participants with the setup. According to the participant's experimental condition, the experimenter introduced the interface and the overall procedure. In design optimization, the designer-led group was tasked to propose three optimal designs within 60 minutes. The optimizer-driven group was told they would be working with an optimizer, which could take 60 minutes or longer depending on the situation.

Once participants concluded their three designs, we again collected the performance data from them on those three designs in a separate session. Since the participants' skill on the interaction may grow over time, this separate session was intended to ensure equal influence on the three designs' evaluation. In this session, the three designs were presented in random order to the participant, each with a formal evaluation containing 36 trials to acquire their averaged performance. Participants did not know which design among the three designs was under evaluation.

4.4.1 Questionnaires. We collected their subjective experience regarding the design process with three question sets. The overall

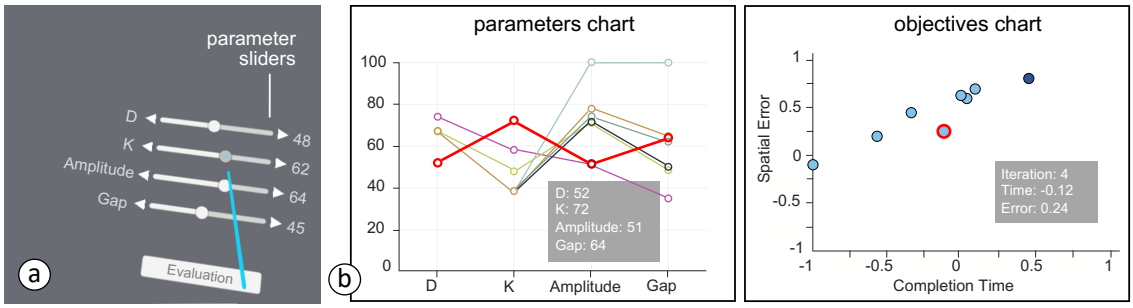


Figure 3: (a) In the designer-led condition, the designers can adjust the 3D touch interaction's parameters using parameter sliders, and initiate a formal evaluation containing 36 trials on the current design with the evaluation button. (b) On completion of a formal evaluation, the parameters and objectives charts are brought up to show the evaluation results. The latest evaluation is indicated in dark blue, and the selected evaluation in red.

experience set contained four 7-point Likert scale questions regarding (1) Satisfaction: how much they were satisfied with the final design, (2) Confidence: how confident they felt the final designs proposed were optimal designs, (3) Agency: how much they felt they were conducting the design, and (4) Ownership: how much they felt they owned the final designs. We used the Creativity Support Index (CSI) [10], a standardized psychometric tool for assessing the perceived creativity support of a tool. It takes into account aspects of perceived creativity including exploration, expressiveness, results worth effort, enjoyment, immersion, and collaboration. We also used NASA-TLX [27], a widely used assessment tool that rates the perceived workload of a task by looking at Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration.

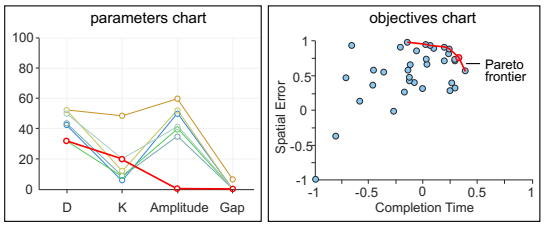


Figure 4: In the optimizer-driven condition, after the 40 formal evaluations, participants were allowed to test the Pareto optimal designs in the Pareto frontier, indicated in red.

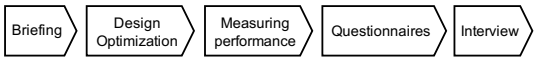


Figure 5: Diagram showing the study procedure: *Briefing*; *Design Optimization* where designers conclude three optimal designs; *Measuring Performance* where design performance on the three designs is re-collected on designers; *Questionnaires*; and *Interview*

4.4.2 Semi-structured Interviews. At the end of each experiment, we conducted a semi-structured interview focusing on experience, perceived issues, and how the participant values the design process and learns about the design space. The interview was audio-recorded. The procedure took about 2 hours in total per participant.

5 RESULTS

5.1 Quantitative Results

5.1.1 Design Performance. Figure 6a shows the averaged completion time and spatial error of the three designs concluded by participant designers in each group. The average completion times were 1120 ms (sd = 119.4) and 1185 ms (sd = 97.2), and the averaged spatial errors were 2.2 cm (sd = 1.2) and 1.5 cm (sd = 0.7), in the designer-led and optimizer-driven groups, respectively. For statistical analysis, we initially log-transformed the completion time data, and confirmed the homogeneity of variances was not violated using *Levene's Test* for both transformed completion time and spatial error data. Then, unpaired t-tests were run on completion time and spatial error data to investigate if any significant differences exist between the groups. The analysis reported significant differences on spatial error ($t(38) = 2.237, p < 0.05$) but not on completion time. This indicates the optimizer-driven method outperformed the designer-led approach in terms of the accuracy of the designs generated.

5.1.2 Designer Performance. Notably, designers in the designer-led group spent 0.6 times less time in design optimization, but visited 6.7 times more design instances than those in the optimizer-driven group. The designer-led group participants spent on average 51.8 minutes (sd = 10.0) on the design, compared to 78.0 minutes (sd = 6.3) in the optimizer-driven group, comprising on average 75.8 and 2.2 minutes respectively in the design and decision stages.

5.1.3 Experience and Workload. Figure 6b displays user ratings on Satisfaction, Confidence, Agency, and Ownership as well as the statistical analyses between the two groups. We ran *Mann-Whitney U Test* on each scale to investigate if significant differences exist. The analysis reported differences existed on Agency ($t(38)$ =

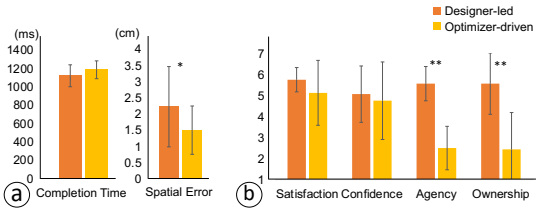


Figure 6: (a) The averaged completion time and spatial error of the designs concluded in the designer-led and optimizer-driven groups. (b) The ratings of general experience on Satisfaction, Confidence, Agency, and Ownership. The error bars denote 1 standard deviation. The one-star (∗) and two-star (∗∗) symbols indicate $p < 0.05$ and $p < 0.001$ significant differences, respectively.

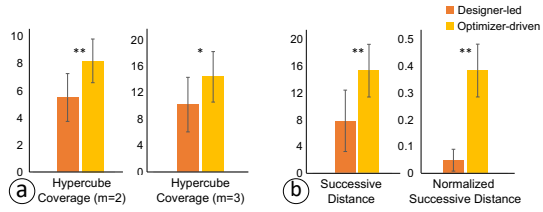


Figure 7: (a) The number of hypercubes covered for both optimizer-driven and designer-led methods for $m = 2$ and $m = 3$. (b) The total successive distance for both optimizer-driven and designer-led processes for the unnormalized case and the normalized case. The error bars denote 1 standard deviation. The one-star (∗) and two-star (∗∗) symbols indicate $p \leq 0.001$ and $p \leq 0.0001$ significant differences, respectively.

$-5.523, p < 0.001$) and Ownership ($t(38) = -3.892, p < 0.001$), but not on Satisfaction and Confidence.

Table 2 summarizes the CSI scores and the statistics analysis between the groups. The *Mann-Whitney U Test* was applied on the overall CSI score and each factor comprising the CSI. The analysis shows a significant difference on the overall CSI score ($t(38) = -2.503, p < 0.05$), suggesting that perceived creativity support was higher in the designer-led group than that in the optimizer-driven group. Comparing each of the factors, significant differences were only found on the Expressiveness factor ($t(38) = -3.222, p < 0.001$). No differences were found on Exploration, Result Worth Effort, Immersion, and Collaboration.

The NASA-TLX scores and the statistical analysis between the groups are summarized in Table 3. The *Mann-Whitney U Test* was applied on the overall NASA-TLX score and each factor of the NASA-TLX. The analysis shows no difference in the overall score. Looking into each factor, significant differences were found only on the Mental Demand and Effort (both $p < 0.05$). No differences were found for the Physical Demands, Temporal Demands, Performance, and Frustration. We found rationales that suggest the factor ratings in each group are distinct and worth discussion. In the following subsection, we will discuss the results and rationales between groups by factor.

	Designer-led		Optimizer-driven		Sig.
Factor	Score	sd	Score	sd	p
Exploration	53.5	16.9	49.3	12.5	.149
Expressiveness	44.9	23.2	23.0	18.9	**.001**
Worth Effort.	55.7	22.9	48.6	26.2	.301
Enjoyment	44.0	28.1	40.8	35.6	.678
Immersion	21.4	21.0	28.2	18.5	.183
Collaboration	6.4	10.2	9.3	15.8	.718
CSI	75.3	13.0	65.4	12.7	**.011**

Table 2: User ratings on Creativity Support Index (CSI).

5.1.4 Exploration and Exploitation during Design. In terms of design exploration, the designer-led group on average visited 271 different designs ($sd = 192.4$), in which testing contributed on average 259 designs ($sd = 194.5$) and formal evaluations contributed on average 12.5 designs ($sd = 5.5$). In comparison, the optimizer-driven group visited only 40 designs selected by the optimizer. We further assessed how designers explored the design space in both conditions. To this end, we came up with the metric of finding how many hypercubes are covered. For our specific application, the total design space is $[0, 1]^4$ and for a given division parameter m, we divide up the space into m^4 hypercubes. We assign a hypercube as being covered if there exists a design parameter set obtained that lies within the hypercube bounds, lower bounds inclusive and upper bounds exclusive. We have the upper bound being inclusive for the special case if the design parameter includes a parameter having the value of 1. We assessed the hypercube coverage of both design methods with $m = 2$ and $m = 3$, each having 16 and 81 hypercubes respectively in Figure 7a. We see that for both values of m, the number of hypercubes covered is greater for the optimizer-driven process as compared to the designer-led method. Figure 8 shows the hypercube coverage for the worst and best performances from the participants for both optimizer-driven and designer-led processes for $m = 2$. The figure illustrates that the worst-case and best-case coverage for the designer-led process covers less of the

	Designer-led		Optimizer-driven		Sig.
Factor	Score	sd	Score	sd	p
Mental.	14.9	8.3	8.4	9.8	**.011**
Physical.	31.8	21.5	38.5	32.2	.242
Temporal.	12.2	20.6	12.6	19.9	.242
Performance	25.1	19.8	15.7	12.7	.398
Effort	24.9	15.3	13.7	11.7	**.040**
Frustration	8.5	12.7	10.0	15.8	1.00
NASA-TLX	57.6	24.4	49.6	28.3	.758

Table 3: User ratings on workloads (NASA-TLX).

design space than that of the optimizer-driven process. Furthermore, we conducted an unpaired t-test to assess whether the means of the two independent conditions are different, and we achieve a p-value of 0.0001 for $m = 2$ and 0.0019 for $m = 3$, both indicating very statistically significant results. Therefore, this shows that optimizer-driven process is able to explore more of the design space consistently as opposed to the designer-led process and hence able to come up with more diverse design candidates. This helps the designer in exploring more different candidates which can alleviate the problems of over-exploitation of a region in the design space.

We also extended the hypercube coverage analysis for various levels of m for the Pareto-optimal designs achieved by each participant. For $m = 2$, the mean hypercube coverage for the designer-led method is 1.4 (sd = 0.6) and for the optimizer-driven method is 1.5 (sd = 0.5). There is no statistical significance in the difference of the means through an unpaired t-test through these two groups ($p = 0.3950$). For $m = 3$, the mean hypercube coverage for the designer-led method is 1.7 (sd = 0.8) and for the optimizer-driven method is 2.0 (sd = 0.9), with no statistical significance in the difference of means ($p = 0.1766$). However, as m increases, the difference in the means becomes statistically significant as for $m = 4$, the mean for the designer-led method is 1.7 (sd = 0.6) whereas it is 2.3 (sd = 0.9) for the optimizer-driven method with p-value of 0.0214, and for $m = 5$, the means for the designer-led and optimizer-driven method are 1.7 (sd = 0.7) and 2.4 (sd = 1.2) respectively with a p-value of 0.0380. This shows the advantage of changing m as a coarseness parameter in determining the level of exploration for different methods of interaction design, as m increases, the hypercubes we considered to be covered become smaller in volume. The above analysis suggests that the optimizer-driven design may be better in determining a wider variety of Pareto-optimal designs with a statistically significant greater coverage of hypercubes as m increases. However, the region of the Pareto-optimal designs can also largely depend on the nature of the problem itself. For instance in our application, certain parameters lead in general to better accuracy and speed trade-offs, and also variation between individual performances of different users.

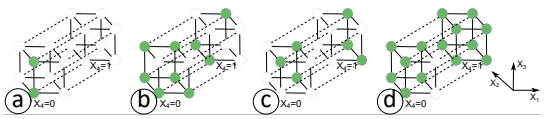


Figure 8: Figure showing the best and worst performance of hypercube coverage for both the designer-led and optimizer-driven conditions from the 40 participants. A hypercube is colored green if it is explored during the optimizer-driven or designer-led process. (a) and (b) show the worst and best coverage for the designer-led processes and (c) and (d) show the worst and best coverage for the optimizer-driven processes.

Next, we assessed explicitly how much designers exploit narrow regions of the design space. We used the metric of the total successive distance—the sum of the Euclidean distances between successive design parameters tried for consecutive design iterations—to measure this. If a designer is over-exploiting or fixated, the successive distance between designs would be small as opposed to a designer who is exploring many very different design candidates. More specifically, a designer that would be fixated would focus on a smaller region of the design space, yielding design instances that are clustered to each other. This results in a smaller successive distance between consecutive iterations and hence a smaller total successive distance. If there was more exploration done on the design space, then the design instances would be in more disparate regions of the design space, yielding a greater successive distance between consecutive iterations and hence a greater total successive distance. In addition, for the designer-led group, there are cases where the total number of design parameters attempted is very large (up to 830 iterations for both exploring and testing), whereas for the optimizer-driven group, the total number is set to be 40 iterations. To account for the variation in the total number of design iterations, we also normalized the total successive distance over the total number of design iterations. This metric would help eliminate the increase in the total successive distance due to simply more design iterations attempted.

The results for the successive distances are shown in Figure 7b. We see that for both normalized and unnormalized successive distances, the optimizer-driven process has a higher value than the designer-led method. It is also worth noting that for the designer-led method, the variance in total unnormalized successive distance is similar to that of the optimizer-driven method, suggesting that both methods yield a similar level of exploration with respect to its mean successive distance. Furthermore, we conducted an unpaired t-test to assess whether the means of the two independent conditions are different, and we achieve a p-value of < 0.0001 for both the unnormalized and normalized total successive distances, both indicating statistical significance. This shows that the optimizer-driven process leads to less fixation on a specific design region and with greater variation in terms of design exploration due to the greater discrepancy in the designs generated in consecutive iterations. Therefore, this indicates that the optimizer-driven process is a useful tool for designers in order to cover more diverse design instances.

5.2 Qualitative Results

5.2.1 Exploration. 18 out of 20 participants in the designer-led group stated the tool as intuitive, calling it "*straightforward*" (P3, P7, P14) and "*easy to learn*" (P2, P5, P17). Six designers stated the tool allowed them to be efficient at exploration (P3: "*testing a design allowed me to gain some idea about the design before going into full evaluation*") especially "*when you want to quickly test alternatives around a design*" (P7). However, six participants reported some sort of anchoring bias, stating "*I invested most of the time in fine-tuning.*" (P3, P5), and that they were aware that "*many [alternatives] were left unvisited*". P12 stated being stuck: "*I think I can push further the [completion] time, but I can't find how*". P14 expressed dissatisfaction but was also resistant to re-initiate the search, saying "*I may start over with any different design, but that would be another long investment*". Designers in the optimizer-driven group perceived the exploration differently. Four participants stated it was interesting to watch "*what designs the AI will bring up to me*" (P22, P24, P34,

P38). P21 mentioned "*it was obvious to me there were many different designs*" and stated he got to know the design space and established what constituted smooth interactions in the process. These comments were echoed by P27 who commented: "*experiencing bad and good designs is helpful in gauging how parameters gave good interaction.*"

5.2.2 Explainability and Reflection. Most participants stated the optimizer generally led them to better designs over time. However, there were those moments they would become confused when "*the new proposal suddenly appeared to be worse*" (P30). P24 mentioned "*I thought I was doing good with the AI, but then it seemed to steer into a very different design direction*". Some blamed the confusion on the AI side, thinking it was "*broken*", and "*got lost*". Others attributed the confusion to themselves, saying "*I wondered if it was my [bad] performance that caused the AI to bring the design*" (P22). Ten participants stated they looked to have some form of explanation from the AI. In most cases, participants realized the optimizer steered them back on good-performance designs and could regain their satisfaction with the AI. Otherwise, two designers who ranked low satisfaction and confidence, commented "*the AI was limited*" (P22, P34).

5.2.3 Agency and Expressiveness . In contrast to the designer-led group, the designers in the optimizer-driven group generally expressed low agency and low expressiveness. Six designers stated they wanted to have some form of agency and express their ideas to the optimizer, especially when they disagreed with designs offered by it. For instance, P24 mentioned, "*I knew what I wanted. I wanted the gap [value] to be reduced, but the AI didn't give me that design*". He suggested a feature of recommending the direction of adjustment, taking the gap as an example. Also, P32 suggested a feature for inputting preference on the design to AI, saying: "*I wish I can just tell the AI I don't like it [the design]*". P33 wanted to skip evaluations where he thought "*trying out [in an evaluation] on a design that I knew wouldn't work is a waste of time.*"

5.2.4 Ownership and Adaptability . The optimizer-driven group received on average low ownership about the design outcome. However, participants reported mixed opinions, reflecting the relatively high variance in the ratings. Six participants attributed low ownership to low enjoyment, calling it "*felt like working for the AI on those trials.*" (P22), "*bored*", and "*not intellectual work*". In addition, P28 commented on no sense of adaptation, stating "*the outcome seemed not to reflect who I am*", thinking others would also get the same design. Since the optimization algorithm leads the design, P30 stated, "*the AI takes the responsibility of the design outcome*". Four designers who gave high ratings commented about the concept of relatedness. For instance, P24 stated "*I realized the AI was adapting design for me when I found the design is getting useful with increasing performance, that I felt I am part of the design*". P35 mentioned the sense of relatedness saying "*the AI was watching closely on those designs I performed well, and providing designs related, and it felt related to me*". P38 attributed the ownership to the effort invested, "*the AI cannot go on designing without me working out those trials*".

5.2.5 Enjoyment and Engagement. The rationales that suggest enjoyment are distinct between groups. In the designer-led group, participants enjoyed advancing the design outcome with their active exploration, saying "*it resembles gaming*" (P12), and in particular, "*seeing my adjustment result in progress is stimulating and helping me engage*" (P11). P4 said "*although it's simple and repetitive, I don't get bored on iterating.*" By contrast, designers in the optimizer-driven group attributed their enjoyment to curiosity and unexpectedness. Three participants stated, "*an interesting way to learn design possibilities*" (P29) and, "*fun to feel like working with the AI*" (P26). Four participants stated being suspicious, for instance saying "*I was doubting it would work out*" (P22) but then felt excitement when seeing progress. Three said "*you don't know what's coming up next until you get to try it*" (P23, P28), so "*each time got me something to expect*" (P38). P33 stated "*adapting myself to a new design is the fun part and sometimes challenging*". However, the enjoyment seemed to not last long; most participants mentioned the enjoyment reduced in later half rounds owing to long design time.

5.2.6 Effort and Responsibility. The designer-led group perceived higher mental demands and effort invested than the optimizer-driven group. Four designers attributed the effort to "*the need to figure out how each parameter works*" (P3), and "*trying to further increase the performance*" (P14). Three participants stated it is challenging to handle two objectives, such that P18 commented "*in fine-tuning, I tended to work on reducing completion time more than spatial errors.*" In the optimizer-driven group, participants reported mental effort was little. P22 stated "*I feel relaxed as the AI is doing the design part*". 18 out of 20 participants ran overtime (more than 60 minutes). However, most reported little pressure of time. P24 mentioned "*it was overtime but I didn't feel it took that long*". Two participants stated they did not feel responsible for the design outcome, saying "*the AI took the lead and should take the responsibility*" (P32, P34).

6 DISCUSSION

Our experimental results expose previously unreported trade-offs when using human-in-the-loop optimization to design interaction techniques. Differences found between the designer-led and optimizer-driven conditions are summarized in Table 4. The results demonstrate that Bayesian optimization enables designers to explore the design space more broadly. In our study, optimizer-driven designers had around 1.5 times more extensive coverage when measured as hypercube coverage than when designers explore on their own. The optimizer-driven group also ended up with somewhat better designs. Their final designs better accounted for the balance of the two objectives with less effort, while designers without optimization assistance focused more on selection speed at the expense of accuracy. However, on the negative side, optimizer-driven designers reported lower expressiveness and agency as well as lower ownership of the design outcomes. The low expressiveness and low agency are likely attributed to the fact that designers are 'dictated to' by the optimizer resulting in a reduced sense of creativity. However, an observed benefit of this 'hand-holding' is that designers felt less effort: some attributed this to being more relaxed, while others felt less time pressure and less stress related to the design outcomes.

Table 4: Summary of differences found between designer-led and optimizer-driven conditions.

Factor	Designer-led	Optimizer-driven
Completion Time	Equal	Equal
Spatial Error	Worse	Better
Agency	Better	Worse
Ownership	Better	Worse
Exploration	Equal	Equal
Expressiveness	Better	Worse
Creativity Support	Better	Worse
Mental Demands	Worse (High)	Better (Low)
Effort	Worse (High)	Better (Low)

6.1 Four Challenges to Improve Human-in-the-loop Optimization

The results inform the development of better methods for human-in-the-loop optimization, which in our view must converge proper interaction techniques with commensurate developments on the algorithmic side.

Challenge 1: Steering the optimizer with partial ideas. Our results suggest that Bayesian optimization is effective when exploring a vast design space. A previous study on a system called Vinci, which used generative models to propose design suggestions interactively [24], reported that designers felt a lack of diversity in important design dimensions. However, our participants felt that loss of agency and expressiveness when being led by the Bayesian optimizer. We see this as an opportunity to develop interaction techniques that allow steering Bayesian optimization.

A key aspect of this challenge is to enable designers to express partial (vague) ideas that the optimizer could explore for them. In our study, designers commented that once they had constructed an internal model of the requirements for a 'good' interaction design, they wanted to be able to express these ideas to the optimizer. This was mostly strongly felt when they found themselves disagreeing with subsequent designs offered by optimizer. Reflecting on this feedback, interaction techniques are needed that allow users to express priorities in design dimensions, or directions where to look at next. However, such developments need commensurate developments in how the Bayesian optimization works, especially in the acquisition function.

Challenge 2: Mixed-initiative interaction. Another direction to improve interactivity is to push the optimizer to the background, making its suggestions recommendations and not dictations as in our study. In a mixed-initiative fashion, it could make suggestions when it sees a significant opportunity. For instance, the Bayesian optimizer could patiently construct a surrogate model of the design space in the background using only the evaluations the designers have encountered in the design process. If the optimizer observes that the designer is spending excess time examining a well-explored region of the design space, the optimizer can suggest alternative design candidates in less well-explored regions. This assistance could also be initiated by designers, for example by pressing a button to request a recommendation from the optimizer when they are stuck for ideas on how to improve the current design. Further, distinct support for exploitation and exploration could be offered for triggering recommendations that respectively aim for local improvements in regions of design space known to be promising or that aim to obtain new insight about unvisited or uncertain regions of the design space.

Challenge 3: Improving transparency. Our designers expressed wanting the optimizer to be more transparent about the proposals. This finding is consistent with general observations within related research areas such as Interactive Machine Learning [16] and Explainable AI [23]. User feedback indicated that designers expect monotonicity during the design process, meaning that designers expect that each new design proposed by the optimizer yields some improvement over the previous iteration. Confusion occurs when they experience the optimizer presenting designs that are then found to perform worse than preceding designs. This confusion in part stems from the users' lack of knowledge about the inner workings of Bayesian optimization. It iteratively refines a surrogate model and leverages an acquisition function to drive the proposition of new points to test, in an exploration and exploitation trade-off. Exploitation seeks to sample where the surrogate model predicts a good objective while exploration samples where the uncertainty is high. Transparency of the method could be improved simply by communicating in which mode it is currently operating so that designers then know they are assisting the optimizer in evaluating uncertain territory where high risk or opportunity is presumed.

Challenge 4: Supporting exploration/exploitation decisions. Our data suggests that user engagement comes from two sources: first, in exploitation where incremental improvement in performance can be expected, and second, in exploration where a fresh unfamiliar design attracts user attention. Human-in-the-loop optimization should help designers take these perspectives when needed. A recent study [60] has explored this concept by allowing users to control sampling behavior in Bayesian optimization determined by acquisition functions so as to adjust the balance between exploration and exploitation. Furthermore, the participants commented that the exploitation process resembled computer games. In the optimizer-driven condition users linked unexpectedness to enjoyment. This observation suggests that it may be fruitful to encourage periodic switching between exploitation and exploration in order to improve engagement under both designer-led and optimization-driven strategies. Such a control may be optionally applied to the Bayesian optimizer by simply assigning a minimum and maximum number of iterations spent in each of the exploitation or exploration modes before mode switching occurs.

6.2 Limitations and Future Work

Our findings are drawn from an empirical study on 3D touch interaction, of which the two objectives for optimization are clearly observable for human designers. Other types of interaction techniques that are not as perceivable to human designers may lead to different techniques to improve the optimization process, which calls for more experimentation. In addition, the results of the empirical study are potentially subject to interpersonal differences due

to the between-subjects protocol used. More experimentation is needed to validate reliability of the differences reported.

7 CONCLUSION

This paper has reported novel observations from a comparative study where two groups of novice designers, one optimized-led and the other self-led, completed a realistic interaction design optimization task. Our main finding is that optimization-led design can help novices identify better designs, but at the expense of agency and expressiveness. When led by an optimizer, designers report lower mental effort but also feel less creative and less in charge of what happens. The results have a practical implication: designers who know a design domain poorly can benefit from Bayesian optimization when optimizing a design. However, more effort is needed to make optimization methods truly interactive, in particular in such ways that can help designers without compromising their agency over the process. We have proposed several ideas to this end in the previous discussion section.

8 OPEN SCIENCE

The Bayesian optimizer and the collected (anonymized) data are released on our project page: https://userinterfaces.aalto.fi/dit. Instructions for the prototype studied in the empirical part will be released, including the installation instructions and the computer program.

ACKNOWLEDGMENTS

The research was supported by the Ministry of Science and Technology of Taiwan (MOST109-2628-E-009-010-MY3), Department of Communications and Networking (Aalto University), the Finnish Center for Artifcial Intelligence (FCAI), Academy of Finland (grants 'OptiHAFE' and 'BAD'), and the Engineering and Physical Sciences Research Council (EPSRC EP/S027432/1). George B. Mo was additionally supported by a Trinity College Summer Studentship Fund.

REFERENCES

[1] Marine Agogué, Nicolas Poirel, Arlette Pineau, Olivier Houdé, and Mathieu Cassotti. 2014. The impact of age and training on creativity: A design-theory approach to study fixation effects. *Thinking Skills and Creativity* 11 (2014), 33–41.

[2] Ferran Argelaguet Sanz and Carlos Andujar. 2013. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers and Graphics* 37, 3 (May 2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[3] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: interactive optimization of menu systems. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*. ACM Press, St. Andrews, Scotland, United Kingdom, 331–342. https://doi.org/10.1145/2501988.2502024

[4] Xiaojun Bi, Barton A. Smith, and Shumin Zhai. 2010. *Quasi-Qwerty Soft Keyboard Optimization*. Association for Computing Machinery, New York, NY, USA, 283–286. https://doi.org/10.1145/1753326.1753367

[5] Doug A. Bowman, Donald B. Johnson, and Larry F. Hodges. 1999. Testbed Evaluation of Virtual Environment Interaction Techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (London, United Kingdom) *(VRST '99)*. Association for Computing Machinery, New York, NY, USA, 26–33. https://doi.org/10.1145/323663.323667

[6] Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 103–112.

[7] Géry Casiez and Nicolas Roussel. 2011. No More Bricolage! Methods and Tools to Characterize, Replicate and Compare Pointing Transfer Functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 603–614. https://doi.org/10.1145/2047196.2047276

[8] Yeonjoo Cha and Rohae Myung. 2013. Extended Fitts' law for 3D pointing tasks using 3D target arrangements. *International Journal of Industrial Ergonomics* 43, 4 (2013), 350 – 355. https://doi.org/10.1016/j.ergon.2013.05.005

[9] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. *Forte: User-Driven Generative Design*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174070

[10] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (June 2014), 25 pages. https://doi.org/10.1145/2617588

[11] Shanna R Daly, Robin S Adams, and George M Bodner. 2012. What does it mean to design? A qualitative investigation of design professionals' experiences. *Journal of Engineering Education* 101, 2 (2012), 187–219.

[12] Alena Denisova and Paul Cairns. 2015. Adaptation in Digital Games: The Effect of Challenge Adjustment on Player Performance and Experience. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* (London, United Kingdom) *(CHI PLAY '15)*. Association for Computing Machinery, New York, NY, USA, 97–101. https://doi.org/10.1145/2793107.2793141

[13] Kees Dorst. 2004. On the problem of design problems-problem solving and design expertise. *Journal of design research* 4, 2 (2004), 185–196.

[14] Peitong Duan, Casimir Wierzynski, and Lama Nachman. 2020. *Optimizing User Interface Layouts via Gradient Descent*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376589

[15] John J. Dudley, Jason T. Jacques, and Per Ola Kristensson. 2019. Crowdsourcing Interface Feature Design with Bayesian Optimization. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300482

[16] John J. Dudley and Per Ola Kristensson. 2018. A Review of User Interface Design for Interactive Machine Learning. *ACM Trans. Interact. Intell. Syst.* 8, 2, Article 8 (June 2018), 37 pages. https://doi.org/10.1145/3185517

[17] Mark Dunlop and John Levine. 2012. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2669–2678. https://doi.org/10.1145/2207676.2208659

[18] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. 2017. *Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design*. Association for Computing Machinery, New York, NY, USA, 1118–1130. https://doi.org/10.1145/3025453.3025599

[19] Gregory Francis. 2000. Designing multifunction displays: An optimization approach. *International Journal of Cognitive Ergonomics* 4, 2 (2000), 107–124.

[20] Scott Frees, G Drew Kessler, and Edwin Kay. 2007. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)* 14, 1 (2007), 2–es.

[21] Krzysztof Gajos and Daniel S. Weld. 2004. SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces* (Funchal, Madeira, Portugal) *(IUI '04)*. Association for Computing Machinery, New York, NY, USA, 93–100. https://doi.org/10.1145/964442.964461

[22] Katerina Gorkovenko, Daniel J Burnett, James K Thorp, Daniel Richards, and Dave Murray-Rust. 2020. Exploring the future of data-driven product design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[23] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. 2019. XAI—Explainable artificial intelligence. *Science Robotics* (2019).

[24] Shunan Guo, Zhuochen Jin, Fuling Sun, Jingwen Li, Zhaorui Li, Yang Shi, and Nan Cao. 2021. *Vinci: An Intelligent Graphic Design System for Generating Advertising Posters*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445117

[25] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O. Riedl. 2019. *Friend, Collaborator, Student, Manager: How Design of an AI-Driven Game Level Editor Affects Creators*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300854

[26] Gregory M Hallihan, Hyunmin Cheong, and LH Shu. 2012. Confirmation and cognitive bias in design cognition. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 45066. American Society of Mechanical Engineers, 913–924.

[27] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[28] Vincent Hayward, Jehangir Choksi, Gonzalo Lanvin, and Christophe Ramstein. 1994. Design and Multi-Objective Optimization of a Linkage for a Haptic Interface. In *Advances in Robot Kinematics and Computational Geometry*, Jadran Lenarčič

and Bahram Ravani (Eds.). Springer Netherlands, Dordrecht, 359–368. https://doi.org/10.1007/978-94-015-8348-0_36

[29] Daniel Hernandez-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. 2016. Predictive Entropy Search for Multi-objective Bayesian Optimization. In *International Conference on Machine Learning*. PMLR, 1492–1501. http://proceedings.mlr.press/v48/hernandez-lobatoa16.html ISSN: 1938-7228.

[30] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.

[31] Florian Kadner, Yannik Keller, and Constantin Rothkopf. 2021. AdaptiFont: Increasing Individuals' Reading Speed with a Generative Font Model and Bayesian Optimization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 585, 11 pages. https://doi.org/10.1145/3411764.3445140

[32] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. *Interactive Optimization for Steering Machine Classification*. Association for Computing Machinery, New York, NY, USA, 1343–1352. https://doi.org/10.1145/1753326.1753529

[33] Mohammad M. Khajah, Brett D. Roads, Robert V. Lindsey, Yun-En Liu, and Michael C. Mozer. 2016. Designing Engaging Games Using Bayesian Optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5571–5582. https://doi.org/10.1145/2858036.2858253

[34] J. Knowles. 2006. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10, 1 (Feb. 2006), 50–66. https://doi.org/10.1109/TEVC.2005.851274 Conference Name: IEEE Transactions on Evolutionary Computation.

[35] Werner A König, Jens Gerken, Stefan Dierdorf, and Harald Reiterer. 2009. Adaptive pointing–design and evaluation of a precision enhancing technique for absolute pointing devices. In *IFIP Conference on Human-Computer Interaction*. Springer, 658–671.

[36] Yuki Koyama, Issei Sato, and Masataka Goto. 2020. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics* 39, 4 (July 2020), 88:88:1–88:88:12. https://doi.org/10.1145/3386569.3392444

[37] Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics* 36, 4 (July 2017), 48:1–48:11. https://doi.org/10.1145/3072959.3073598

[38] Daniel Lange, Tim Claudius Stratmann, Uwe Gruenefeld, and Susanne Boll. 2020. *HiveFive: Immersion Preserving Attention Guidance in Virtual Reality*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376803

[39] Yang Li, Samy Bengio, and Gilles Bailly. 2018. *Predicting Human Performance in Vertical Menu Selection Using Deep Learning*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3173574.3173603

[40] Antonios Liapis, Gillian Smith, and Noor Shaker. 2016. Mixed-initiative Content Creation. In *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Noor Shaker, Julian Togelius, and Mark J. Nelson (Eds.). Springer, 195–214.

[41] J. Derek Lomas, Jodi Forlizzi, Nikhil Poonwala, Nirmal Patel, Sharan Shodhan, Kishan Patel, Ken Koedinger, and Emma Brunskill. 2016. Interface Design Optimization as a Multi-Armed Bandit Problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 4142–4153. https://doi.org/10.1145/2858036.2858425

[42] Shouichi Matsui and Seiji Yamada. 2008. Genetic Algorithm Can Optimize Hierarchical Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1385–1388. https://doi.org/10.1145/1357054.1357271

[43] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 95, 3 (1988), 340.

[44] Brad A. Myers and William Buxton. 1986. Creating Highly-Interactive and Graphical User Interfaces by Demonstration. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. Association for Computing Machinery, New York, NY, USA, 249–258. https://doi.org/10.1145/15922.15914

[45] Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon. 2015. Mid-Air Pointing on Ultra-Walls. *ACM Trans. Comput.-Hum. Interact.* 22, 5, Article 21 (Aug. 2015), 62 pages. https://doi.org/10.1145/2766448

[46] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. DesignScape: Design with Interactive Layout Suggestions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1221–1224. https://doi.org/10.1145/2702123.2702149

[47] Antti Oulasvirta, Niraj Ramesh Dayama, Morteza Shiripour, Maximilian John, and Andreas Karrenbauer. 2020. Combinatorial optimization of graphical user interface designs. *Proc. IEEE* 108, 3 (2020), 434–464.

[48] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (Seattle, Washington, USA) *(UIST '96)*. Association for Computing Machinery, New York, NY, USA, 79–80. https://doi.org/10.1145/237091.237102

[49] IVAN POUPYREV and TADAO ICHIKAWA. 1999. Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques. *Journal of Visual Languages and Computing* 10, 1 (1999), 19 – 35. https://doi.org/10.1006/jvlc.1998.0112

[50] Joon Gi Shin, Doheon Kim, Chaehan So, and Daniel Saakes. 2020. *Body Follows Eye: Unobtrusive Posture Manipulation Through a Dynamic Content Position in Virtual Reality*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376794

[51] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, Seoul, Republic of Korea, 3643–3652. https://doi.org/10.1145/2702123.2702136

[52] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 543–555. https://doi.org/10.1145/2901790.2901817

[53] Johann Wentzel, Greg d'Eon, and Daniel Vogel. 2020. *Improving Virtual Reality Ergonomics Through Reach-Bounded Non-Linear Input Amplification*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376687

[54] Georgios N. Yannakakis and John Hallam. 2008. Real-time adaptation of augmented-reality games for optimizing player satisfaction. In *2008 IEEE Symposium On Computational Intelligence and Games*. 103–110. https://doi.org/10.1109/CIG.2008.5035627

[55] Georgios N Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. 2013. Player modeling. (2013).

[56] Robert J. Youmans and Thomaz Arciszewski. 2014. Design fixation: Classifications and modern methods of prevention. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28, 2 (2014), 129–137. https://doi.org/10.1017/S0890060414000043

[57] Jaesik Yun, Youn-kyung Lim, Kee-Eung Kim, and Seokyoung Song. 2015. Interactivity Crafter: An Interactive Input-Output Transfer Function Design Tool for Interaction Designers. *Archives of Design Research* 28 (08 2015), 21–37. https://doi.org/10.15187/adr.2015.08.28.3.21

[58] Shumin Zhai, Michael Hunter, and Barton A. Smith. 2000. The Metropolis Keyboard - an Exploration of Quantitative Techniques for Virtual Keyboard Design. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (San Diego, California, USA) *(UIST '00)*. Association for Computing Machinery, New York, NY, USA, 119–128. https://doi.org/10.1145/354401.354424

[59] Tianming Zhao, Chunyang Chen, Yuanning Liu, and Xiaodong Zhu. 2021. GUIGAN: Learning to Generate GUI Designs Using Generative Adversarial Networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. 748–760. https://doi.org/10.1109/ICSE43902.2021.00074

[60] Yijun Zhou, Yuki Koyama, Masataka Goto, and Takeo Igarashi. 2021. Interactive Exploration-Exploitation Balancing for Generative Melody Composition. In *26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 43–47. https://doi.org/10.1145/3397481.3450663

Publication III

Yi-Chi Liao, George B Mo, John J Dudley, Chun-Lien Cheng, Liwei Chan, Per Ola Kristensson, Antti Oulasvirta. Practical Approaches to Group-Level Multi-Objective Bayesian Optimization in Interaction Technique Design. Submitted to *ACM Collective Intelligence*, 14, March 2023.

Collective Intelligence
XX(X):1–15

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Practical Approaches to Group-Level Multi-Objective Bayesian Optimization in Interaction Technique Design

Yi-Chi Liao[1]*, George B. Mo[2]*, John J. Dudley[2], Chun-Lien Cheng[3], Liwei Chan[3], Per Ola Kristensson[2], and Antti Oulasvirta[1]

Abstract

Designing interaction techniques for end-users often involves exploring vast design spaces while balancing many objectives. Bayesian optimization offers a principled human-in-the-loop method for selecting designs for evaluation to efficiently explore such design spaces. To date, the application of Bayesian optimization in a human-in-the-loop setting has largely been restricted to optimization, or *customization*, of interaction techniques for individual user needs. In practice, interaction techniques are typically designed for a target population or group of users, with the goal is to produce a design that works *well* for *most* users. To accommodate this common use case in interaction technique design, we introduce two practical approaches that facilitate multi-objective Bayesian optimization at the group level. Specifically, our approaches streamline the process of: (1) deriving designs suitable for a group of users from data collected in individual user evaluations; and (2) deriving an initialization from group data to improve the efficiency of design optimization for new users. We demonstrate the advantages of these practical approaches in two multi-phase user studies involving the design of non-trivial interaction techniques.

Keywords

Human-in-the-Loop Optimization; Interaction Technique; Interface Design; Optimization; Design Optimization; Bayesian optimization; Pointing; Haptics; Input; Touch

Introduction

Developing an interaction technique is hard. Technically, it involves setting the values of various design parameters that eventually shape the performance and experience of users. These configurable attributes implicitly define a multi-dimensional design space with theoretically infinite feasible operating points. A simple design task with three configurable design parameters, each with ten possible levels, already has 1,000 feasible operating points and most practical tasks face even larger design spaces. Unfortunately, the relationship between particular design choices and their outcomes for users is complex and rarely predictable. Even a small change in a design parameter that improves one aspect of an interaction technique can have unwanted side effects on the others. Therefore, selecting the 'best', or even a 'good enough' operating point poses a central and nontrivial challenge to the design of interaction techniques.

Complicating matters further is the fact that seeking to design for more than a single objective, i.e., a measurable performance metric, exposes the challenge of *Pareto-optimality*. Pareto-optimality refers to the idea that given more than one design objective, there is no longer a single best operating point. For example, a particular interaction configuration may yield good task performance but receive mediocre subjective user ratings, or vice versa. For a given Pareto-optimal design, no individual objective can be improved by adjusting the design parameters without making at least one other individual objective worse off. For instance, a well-known trade-off in interaction technique design is balancing speed and accuracy. A design instance that favors speed may lead to higher error rates, and vice versa. Such trade-offs are difficult to navigate without some degree of subjectivity and/or introducing secondary constraints.

The conventional way of tackling this challenge has been via empirical evaluation of manually selected operating points: a promising set of combinations is chosen and compared in an experiment (1). However, because of the significant time and effort involved, a truly exhaustive study is rarely conducted, and the designer must often select a design by other means, for instance, by basing a decision on previous results (2; 3). This approach is not only prone to missing good designs, but potentially biased based on prior experience and personal preferences. Recently, human-in-the-loop optimization has emerged as a more systematic and unbiased design exploration process. Bayesian optimization is a particularly strong candidate for human-in-the-loop optimization given that it achieves high sample-efficiency by leveraging an iteratively refined model of the design space (4). Underlying this is a probabilistic surrogate model, such as a Gaussian

[1]Aalto University
[2]University of Cambridge
[3]National Yang Ming Chiao Tung University

Corresponding author:
Yi-Chi Liao Department of Information and Communications Engineering, School of Electrical Engineering Espoo, Finland.
Email: yi-chi.liao@aalto.fi
*The authors contribute equally to this paper.

process (GP), which offers a robust probabilistic estimate of the latent function relating the design parameters to measurable objectives. As more observations are collected, the quality of the GP's estimate is improved, which in turn enables the optimizer to make informed choices about where to sample next. Recently, researchers have investigated the advantages and disadvantages of using *multi-objective Bayesian optimization* (MOBO) in assisting design exploration (5; 6).

Prior work has applied Bayesian optimization in the context of human-in-the-loop interface and interaction design to determine: game mechanics that maximize user engagement (7); font features that maximize user reading speed (8); interface features that minimize interface search time (9); interaction settings that minimize selection time and maximize accuracy (5); and animation and image adjustments to efficiently match some desired appearance (10; 11; 12).

These prior applications of Bayesian optimization are either limited to a single objective and/or focus on customization of the design to the individual. In practice, the design intent for interaction techniques is often to address the needs of a group or population of users, rather than an individual. Accommodating multiple objectives further complicates such an optimization process given the concept of Pareto-optimality and the absence of a single best operating point. In this current work, we seek to bridge the gap between the demonstrable efficiency of Bayesian optimization for interaction technique design, and the practical need to focus on group requirements in contrast to individual requirements.

In this paper, we introduce two practical approaches that streamline the process of working with group data for MOBO in interaction technique design. Both approaches leverage MOBO to independently identify optimal designs for a group of individuals and subsequently offer methods for aggregating the data from these multiple individuals to assemble a performance model representative of the group. These aggregated models can then be inspected to identify Pareto-optimal designs for the group or used to derive an efficient initialization for subsequent design customization for new users. We refer to these two approaches as: (i) the *Global GP*, which computes group-level optimal designs from data obtained from individual users; and (ii) the *Warm-Start GP*, which provides an initialization for the MOBO process using group-level data to support more rapid individual design optimization. We evaluate each capability in challenging and realistic interaction design tasks across two multi-phase user studies: (i) designing a 3D touch interaction in virtual reality, exposing a complex trade-off between selection speed and accuracy; and (ii) tuning the vibration feedback for a touchscreen button to achieve an effective compromise between temporal accuracy, consistency and user comfort.

In summary, this paper makes three key contributions:

1. We introduce the *Global GP* as a practical method for generating Pareto-optimal design instances based on user-specific optimizations performed by a group of users.
2. We introduce the *Warm-Start GP* as a practical method for deriving initializations from group-level data in order to facilitate more rapid optimization for a new user.
3. We demonstrate both the *Global GP* and the *Warm-Start GP* in two representative and challenging interaction design tasks. This provides a valuable reference on how to apply group-level multi-objective optimization more broadly to HCI design problems.

Related Work

Computational methods for assisting designers have attracted substantial recent attention. In this section we review the literature introducing and demonstrating computational approaches to interaction technique design.

Computational One-Shot Design

Computational one-shot design relies on prior knowledge of the function relating design choices to performance objectives. For example, one can construct a model describing the impact reducing the size or separation between buttons has on selection time. Discrete and continuous optimization methods have been extensively explored for one-shot design of user interfaces (13). These approaches have been successfully applied in a variety of HCI applications such as widget layouts (14), keyboards (15), and context-aware interfaces in AR (16). These approaches generally assume a *predictive model* as given or learned from a pre-existing dataset. This established model can then be queried to guide the search over the design space. Often, however, it is not possible to derive or acquire a predictive model relevant to a novel design problem and so such approaches do not generalize well.

Bayesian Approaches for Single-Objective Problems

Bayesian optimization is a machine learning method that performs efficient exploration of complex or black-box objective functions to identify an optimum point. Bayesian optimization eliminates the need for an established predictive model or prior exploration of the design space. The approach is well suited to applications where the objective functions are expensive or difficult to evaluate due to the required time or effort. Bayesian optimization, therefore, has good utility in supporting interface and interaction design since many objectives can only be evaluated by conducting a test with a user. Prior work outside of HCI has also shown that Bayesian optimization can outperform other black-box optimization methods for one-dimensional design problems (17). Bayesian optimization has been applied to a wide variety of optimization problems, but we subsequently constrain the scope of our review to design tasks involving a user. For a more general overview please see Shahriari et al. (4).

Khajah et al. (7) applied Bayesian optimization to assign parameter values dictating the mechanics of a game in order to maximize user engagement. Also using Bayesian optimization, Kadner et al. (8) sought to customize font designs for individuals to maximize reading speed. Dudley et al. (9) also leveraged crowdsourcing to quickly access a large number of users but evaluated a more traditional

metric based on task completion time to refine design parameters for a range of simple user interfaces. Users were also given the opportunity to subjectively rate interface designs but this was not integrated into the optimization process. Brochu et al. (12) demonstrated a technique for allowing designers to quickly determine appropriate values for smoke animation while Koyama et al. (10; 11) sought to streamline user editing of photographs to achieve a desired visual appearance. The ability to tightly integrate the user into the procedure makes Bayesian optimization well suited to customizing settings to an individual. This capability has been exploited to tune hearing devices (18) and other assistive technologies (19). Piovarci et al. (20) used an approach influenced by Bayesian optimization to explicitly search for design parameters in surfaces and styli for drawing haptics that exhibit target friction and vibration objective values. However, the goal of Piovarci et al. (20) was to obtain parameter values that yield predetermined friction and vibration qualities and so they do not strictly perform multi-objective optimization over the design space. Chan et al. (5) reported the results of a between-subjects experiment which investigated the advantages and disadvantages of using Bayesian Optimization in interaction technique design versus manual design space exploration. The core focus of Chan et al. (5) was investigating when designers generate design candidates and evaluate theses designs on themselves, as opposed to with a group of end users.

Multi-Objective Bayesian Optimization

Many methods have been proposed for multi-objective optimization of black-box objectives from evolutionary approaches (21) to Bayesian approaches (22; 23; 24). Multi-objective optimization has been employed in many engineering problems, such as in user interface design. It has been used to optimize linkages for haptic interfaces (25), mid-air text entry (26), and keyboard layout optimization (27). These methods either relied on reducing the multiple objectives into a single objective by a linearized weighted sum (26) or by variations on grid search or trial-and-error (25; 27). Feit et al. (28) provided an extensive overview of the challenges and methods available in applying multi-objective optimization to keyboard design. However, none of those methods use Bayesian optimization to converge to the Pareto-optimal parameters, without applying heuristics, by optimizing over black-box objective functions. Most of these methods assumed that the multiple objectives are independent, however, Shah and Ghahramani (29) described an approach that incorporates the correlation between expensive objectives. In this paper, we build on the formulation introduced by Shah and Ghahramani (29) and apply it to interaction design for groups.

Practical Approaches to MOBO: the Global GP and the Warm-start GP

When optimizing a single objective it is possible to determine a single 'best' design*. By contrast, when performing multi-objective optimization there is no single optimum but rather the set of operating points that represent optimal trade-offs between the design objectives. For example, consider three hypothetical designs *A*, *B* and *C*: *A* delivers high speed but poor accuracy; *B* delivers high accuracy but poor speed; and *C* delivers moderate accuracy and moderate speed. All of these designs may be considered optimal if their performance in one of the two objectives cannot be improved without making the other objective worse. The set of operating points for which one objective cannot be increased without the other objectives decreasing is referred to as the set of Pareto-optimal designs or the Pareto front.

In Bayesian optimization, the goal is to optimize over black-box objective functions by sequentially choosing new test points at which to evaluate those objectives. As new samples are collected, a surrogate model relating the design parameters to their approximate objective function values is updated. It is typical to use a Gaussian process (GP) as this surrogate model. A GP is a non-parametric method that models functions, giving uncertainty estimates about function values and often allowing analytically tractable inference. An acquisition function is consulted to determine which point should be sampled next. Acquisition functions propose sampling points by trading off exploration (sampling where the inference uncertainty is high) and exploitation (sampling where the surrogate models predicts high objective values). For multi-objective Bayesian optimization, the acquisition assessment is typically performed with reference to the Pareto hypervolume (30). The Pareto hypervolume is the volume bounded by a fixed reference point on one side and the multidimensional Pareto front on the other side. The acquisition function effectively seeks to sample a new design point that will increase the Pareto hypervolume as this corresponds to a new point that advances the Pareto front. We leverage the acquisition function called CEIPV (Correlated Expected Improvement in Pareto hyperVolume), proposed by Shah and Ghahramani (29).

Previous works have applied MOBO to interface design problems, but they have focused solely on optimizing for individual users (5; 6). In this paper, we demonstrate two methods that facilitate the application and broaden the utility of MOBO by addressing two unique requirements commonly encountered in HCI design problems. The first requirement is the common goal in interaction design to arrive at a configuration that performs *well* for *most* users. This reflects the need to release products that are acceptable to a broad user base. The second requirement is the goal of supporting efficient customization methods from some broadly acceptable initial configuration. This reflects a desirable quality for an interaction technique to perform generally well from the outset but to also quickly adapt to individual user needs. We refer to the two proposed methods as the *Global GP* and the *Warm-Start GP* and introduce each below.

Global GP: Concise User Group Modeling

Running the standard MOBO procedure produces a distinct set of Pareto-optimal designs for each individual user. Each Pareto front obtained potentially reflects the unique

*Strictly, several designs may be equally good if they all achieve the same performance in terms of the chosen objective.

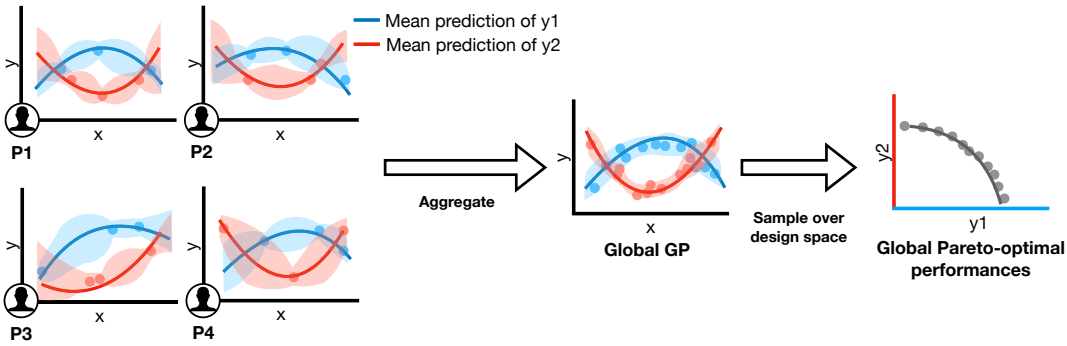


Figure 1. The Global GP aggregates all observations from the user-specific optimization processes. The consolidated model can thereby estimate the group's average performance at any given design parameter value. After constructing the Global GP, we perform a fine-grid sampling of the design space to identify the global Pareto-optimal design instances.

preferences and abilities of these individual users. In interaction design we typically want to find a configuration that is suitable for a broad user base rather than a design that works well for one person but poorly for the majority. Therefore, we wish to combine the Pareto-optimal designs obtained for all users sampled individually into a single 'global' model that reflects the broader preferences and abilities of the user group. Here we use the term 'global' to refer to the user group as distinct from individual users.

To fulfill this task, we construct a *Global GP* that incorporates all the data from all users. A GP is a probability distribution over possible functions and estimates the model relating input values, x, to function values, y. In the context of interaction design, x refers to design parameter values while y is the measured performance. Since a GP captures the probability distribution over all possible functions, one can derive the means of the functions and the variances to indicate the confidence of the predictions. The Global GP is constructed by providing all observed pairings of design parameters and objectives from all participants to form a single GP model. We can then inspect the Global GP to obtain the estimated mean and variance of y (i.e., performance objectives) at any x (i.e., design parameter settings) and use this to predict the expected performance of the group given a particular design instance. To obtain the Pareto-optimal designs from this Global GP, we conduct a fine-grid sampling of the design parameter space. Given appropriate bounds for each parameter and normalization, the design parameter space is a hypercube in $\mathbb{R}^n$ and so we do an exhaustive fine-grid sampling on that hypercube with the resolution specifying the coarseness, c, of the sampling. Given c, we divide each dimension of the parameter space $\mathcal{X}$ into c equally spaced grid points, 0 and 1 inclusive. After sampling, we have c^n samples, which we use to output the set of global Pareto-optimal designs. We set $c = 16$ for our applications which was determined by the empirical trade-off between design parameter specificity and computation time.

This method is computationally expensive but since it runs as a post-processing step, it is practical and feasible to perform and provides a comprehensive summary of the optimal design parameter sets. Figure 1 illustrates the high-level process of constructing the Global GP and extracting the Pareto-optimal designs.

Besides using GPs as base models, other approaches, such as regression models and deep neural nets, may be applied to constructing a global-level model. However, these methods have various requirements and limitations. A deep neural net requires a large amount of data and high-dimensional input. That typically demands a large number of users, which is not usually feasible for human-in-the-loop optimization. Parametric regressions, such as linear regressions, require assumptions of the model (e.g., the number of degrees and the landscape of input and output), which are also not always viable for HCI problems. Furthermore, deep neural nets and parametric regression models are not capable of capturing the uncertainty of the predictions. In comparison, a non-parametric GP is a more general and natural choice, and it effectively models the uncertainty of the prediction.

Warm-Start GP: Efficient Initialization for User-Specific Customizations

Another common use case encountered in interaction design is optimizing the parameters of a technique to the particular abilities and preferences of an individual. This customization or personalization process can also be efficiently performed with the aid of multi-objective Bayesian optimization. Ideally, this process should be as fast as possible and one way to enhance efficiency is to initialize the MOBO procedure with a generalized appreciation of the design space. In practice, this can be achieved by selecting some subset of the data collected from all users to initialize the Bayesian optimization procedure when personalizing a technique to a new user. The problem then is to select a sparse subset of size K from the whole dataset that is representative of the whole dataset so as to quickly adapt to the needs of the new user. These selected points are used to construct a *Warm-Start GP* which then provides the initialization for running Bayesian optimization with the new user. Figure 2 illustrates this general procedure.

We simplify and adapt the greedy selection approach taken by Titsias et al. (31) by using the approximation to the

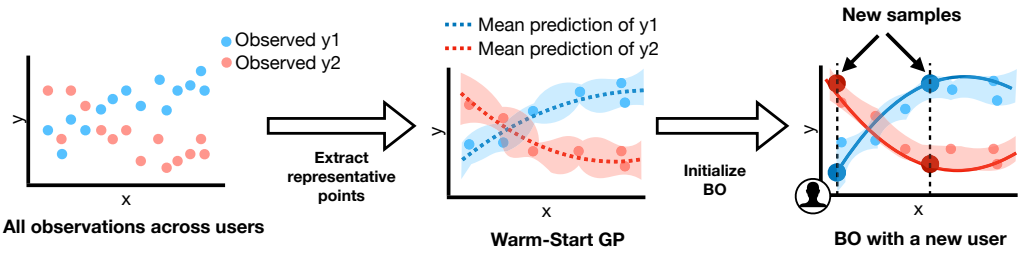


Figure 2. We extract the most representative observations from the current pool of users to form a Warm-Start GP. This Warm-Start GP serves as an informative prior, allowing rapid adaptation to individuals with fewer BO iterations.

marginal likelihood of the entire dataset with the size K subset given by Seeger et al. (32). We refer the reader to those papers for more details on the greedy algorithm in (31) and the approximation and hyperparameter tuning in (32). The central idea is to select the K points by iteratively adding a training point greedily from the complete dataset that maximizes the approximate marginal likelihood of the complete dataset. This approximate marginal likelihood of the complete dataset is determined from the GP constructed from the sparse subset and using the model to compute the approximate marginal likelihood over the complete dataset. We also apply the following heuristic to reduce computation time. First, instead of including the entire dataset as initial candidates for the sparse subset, we first reduce the set of all possible candidates to a randomly selected subset. For the touch-button temporal pointing task described in Study 2, we set the size of that randomly selected subset to be 100, corresponding to half of the size of the total number of data points collected (200). Next, from that reduced dataset, we apply the likelihood maximization process as stated above to greedily select the candidate point to be in the sparse subset of size K. We set K to 5 in the application described in Study 2. This method produces an appropriate prior which can adapt to a new user from newly given samples to obtain a personalized optimal design parameter set. Although there have been alternative methods proposed for sparsifying GPs (33; 34; 35), we pursued this computationally efficient approach that retains representative data points from the original dataset.

Study 1: Individual-to-Group Design with the Global GP

In this study, we seek to validate our Global GP method for deriving a set of optimal designs that are representative of a group of users. We do this within the context of a design problem in 3D touch interaction loosely based on the Go-Go technique (36). In the first phase of the study, participants performed 3D touch selections while the MOBO procedure seeks to identify their individual set of Pareto optimal designs. The data from the individual participants was then used to generate a set of global Pareto-optimal designs using our Global GP method. In the second phase of the study, we evaluated the performance of two designs taken from this global Pareto-optimal set against a baseline

Table 1. Design parameterization of the 3D touch interaction.

Design Parameter		Range
x_1:	Distance threshold, D	$[0, 1]$
x_2:	Scale factor, k	$[0, 0.5]$
x_3:	Activation-vibration gap	[15 cm, -5 cm]
x_4:	Vibration amplitude	[0 g, 2.6 g]

configuration roughly based on the design of the original Go-Go technique (36).

3D touch interaction is a subclass of 3D object selection based on the virtual hand metaphor. A wide array of selection techniques have been proposed and tailored to different applications by trading off between accuracy and speed (37; 38; 39). Depending on the metaphor and the implementation of the interaction, the number of design parameters can range from three to ten (37; 40; 41; 42). A more detailed summary of 3D selection and pointing techniques can be found in Sanz and Andujar (37).

The *Go-Go technique* (36) is a classic work which enables users to touch virtual targets appearing beyond their physical reach. It employs a control-display gain approach that selectively applies a linear or nonlinear scaling on the virtual hand according to the physical position of the real hand. Two parameters determine the selection of the mapping schema and the degree of the nonlinearity. The general task of finding ideal control-display gain function parameters can be both challenging and time-consuming. Previously, the gain function of pointing devices has been decided by either extensive trial-and-error (43) or by heuristic iteration (44; 45). These approaches are costly in terms of time and effort and difficult to conduct without prior expertise. Perhaps as a consequence of this difficulty, the Go-Go technique as described in (36) recommends parameter settings without providing clear rationale. Many similar interaction techniques presented in the literature also contain parameter values that were arbitrarily chosen or derived from informal pilot testing. As an illustration of an alternative approach, this study demonstrates how MOBO can efficiently and systematically guide the identification of design parameters most suitable for a sampled user population.

Design Parameterization and Objectives

The original Go-Go technique scaled hand motions by computing offsets with respect to the user's chest. To better

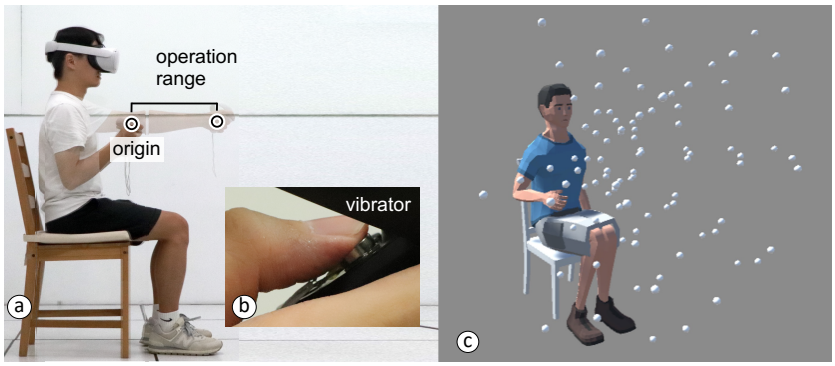


Figure 3. The experiment setup for the 3D touch task. (a) The reference origin and operation range as adapted from the original Go-Go technique design. (b) The interaction is enhanced with vibrotactile feedback via the vibrator added to the controller. (c) All possible locations of targets.

capture the direction and dynamic range of this motion, we relocated the reference frame to the hand's position when fully retracted to the shoulder, as shown in Figure 3. We defined the distal bound of the *operation range* as the distance between the origin and the hand when the arm is fully extended, as shown in Figure 3a. The Go-Go technique's scaling mechanism was applied over this operation range.

The 3D touch design was parameterized according to four variables as described in Table 1. There are two parameters, x_1 and x_2, that determine the resulting virtual hand position in 3D space. The first parameter, x_1, is referred to as D in the original Go-Go technique publication and describes the normalized distance in the operation range at which the mapping transitions from linear to non-linear scaling. The second parameter, x_2, is the non-linear scaling factor and is referred to as k in the original publication. Appropriate parameter ranges were determined by pilot testing. We set the parameter $x_1 \in [0, 1]$ and the parameter $x_2 \in [0, 0.5]$.

In an effort to further improve selection performance, we augment the original Go-Go technique by introducing a haptic cue when the target is reached. Previous works have shown that vibration can effectively assist selection (46; 47), and it is widely employed in commercial VR controllers. We selected two parameters to describe this vibrotactile feedback: the activation-vibration gap, x_3, and the vibration amplitude, x_4. The activation-vibration gap is the distance from the target at which the cue is activated and was bound to values between 15 cm before and 5 cm after the target. The vibration amplitude is the intensity of the cue and was bound between 0 and the maximum voltage level (3.1 V, 2.6g). The duration of the vibration feedback was fixed at 300 ms.

Input selection techniques are characterized by a trade-off between speed and accuracy. We therefore chose two proxy measures of speed and accuracy to guide the optimization process: completion time and spatial errors in target acquisition. Completion time refers to the average duration between the moment of the first movement and the moment the target is successfully selected. Spatial error is the maximum overshoot distance, that is, the 3D Euclidean distance between the cursor represented by a virtual hand and the target's position. Both completion time and spatial error are minimization metrics, i.e., a smaller value indicates better performance. We convert these metrics into two objectives which we subsequently refer to as *speed* and *accuracy*, before passing them into the optimizer. Based on pilot testing, we linearly map the completion time ranged [1,600 ms, 900 ms] to speed ranged [-1, 1], and linearly map the spatial error ranged [2 cm, 0 cm] to accuracy ranged [-1, 1].

Phase 1: User-Specific Optimizations

In this first phase of the study, participants were exposed to the standard MOBO procedure and the design parameters were optimized at the individual level. A set of global designs were then derived using the Global GP method, providing the basis for Phase 2 described later in Section .

Participants: In total, we recruited 20 paid participants (9 female) from our university for the whole of Study 1. Their average age was 23.3 (sd = 0.8). We randomly divided them into two groups. The group who participated in the first study will subsequently labeled as the 'experienced' group in Phase 2 of the study. The 'novice' group only participated in Phase 2. Participants in the experienced group received €20, and participants in the novice group received €10 as a token of appreciation for their involvement.

Task: Participants performed a 3D touch task in a VR setup where the completion time and overshoot error were measured. Selection was performed based on a dwell threshold (0.5 s) with a cursor representing the virtual hand.

Apparatus and Prototype: We built the 3D touch interaction application in Unity 3D. Participants wore an Meta Quest and performed the task with the companion hand controllers. These controllers were modified to include the custom vibration motors as shown in Figure 3b.

Setup and Procedure: We followed task arrangements used in (48) for 3D target acquisition. Each iteration of the task contained 36 trials (selecting a single target), drawn randomly from the radial distances, target widths, the azimuth, and inclination angles, to ensure the index of difficulty across trials was well-distributed. The possible target locations are shown in Figure 3c. A five-minute break

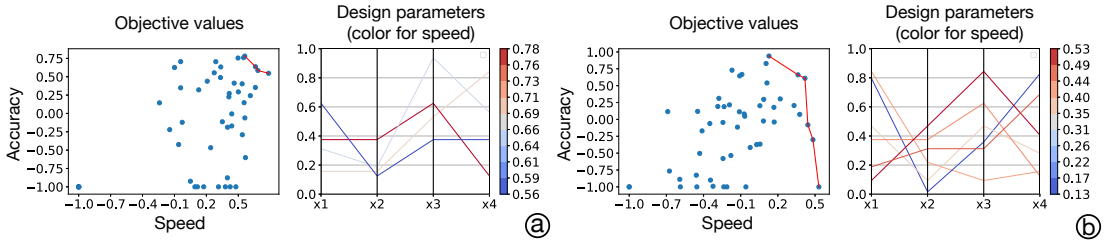


Figure 4. The Pareto front and Pareto-optimal designs obtained for two illustrative participants (a and b).

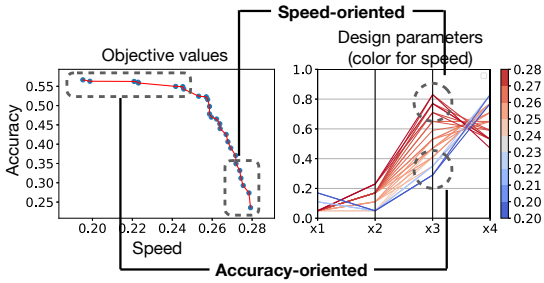


Figure 5. The predicted Pareto-optimal objective values from the Global GP and the global Pareto-optimal designs.

was given every ten iterations. The whole procedure lasted approximately 90 minutes.

MOBO Hyperparameters: The design configurations used in the first 10 task iterations of the experiment were randomly selected. The subsequent 40 task iterations utlized design configurations proposed by the MOBO procedure. The design space was discretized into 40 possible cue configurations.

Results of the User-Specific Optimizations. Participants exhibited natural differences in their performance and this is reflected in the identified optimal designs. Figure 4 shows the Pareto front and Pareto-optimal designs obtained for two illustrative participants. The optimal designs obtained for the first participant (left two plots) generally show superior performance in both objectives compared with the second participant. The two participants yielded rather distinct parameter designs suggesting that the MOBO procedure has successfully captured user-specific optimal designs.

Phase 2: Evaluation of the Global Designs from the Global GP

In this second phase of the study, we evaluated the performances of the global designs derived from the data collected in Phase 1 against a baseline design configuration. The purpose of this evaluation was to assess the quality of the designs produced by the Global GP method. If designs extracted by the Global GP perform as well or better than the baseline configuration, this indicates that the approach can effectively produce designs suitable for a group of users.

This evaluation was performed by both an *experienced group* who participated in Phase 1 and a *novice group* who were completely new to the study. There was a two day gap between Phase 1 and Phase 2. The structure of the evaluation was a 5 (designs) $\times$ 2 (groups of participants) mixed-design experiment with two independent variables. Each participant tested all of the design instances; thus, this factor is within-subjects. The two groups of participants is a between-subjects factor.

Table 2. The three design conditions evaluated in Phase 2.

Condition	x_1	x_2	x_3	x_4
Speed-oriented	0.05	0.098	5.77 cm	2 g
Accuracy-oriented	0.092	0.037	10.76 cm	0.91 g
Go-Go Technique	0.667	0.167	0 cm	1 g

Generating Global Pareto-Optimal Designs We used the Global GP method based on all observations from all the participants in Phase 1 to generate a set of global Pareto-optimal designs. Each design parameter was equally divided into 16 levels for grid search in the Global GP method. The set of derived Pareto-optimal global designs is presented in Figure 5.

We grouped the global designs into a speed-oriented subset and an accuracy-oriented subset. The speed-oriented subset prioritized completion time over spatial errors, while the reverse is true for the accuracy-oriented subset. We then generated two final designs for evaluation by averaging over the individual parameter values in each subset.

These two final designs were then compared against the baseline Go-Go technique. As noted earlier, the original Go-Go technique does not include vibration feedback. To ensure a fair comparison, we augmented the standard Go-Go technique so that the vibration cue will be generated as the user contacts the target with the virtual hand ($x_3 = 0$ cm). This reflects a common setting used in VR interactions (49). The vibration amplitude (x_4) was set to 1 g, which was the most preferred and effective amplitude among four alternatives (0.5 g, 1 g, 1.5 g, 2 g) presented in a pilot test. The three conditions evaluated in Phase 2 are summarized in Table 2.

Evaluation Setup The three design conditions were presented in four rounds, where each round consisted of 36 trials (target selections). The condition order was counterbalanced with a Latin square. The evaluation phase lasted approximately fifteen minutes for each participant.

Performance of the Global Designs Figure 6 shows the mean completion time and spatial error for each condition

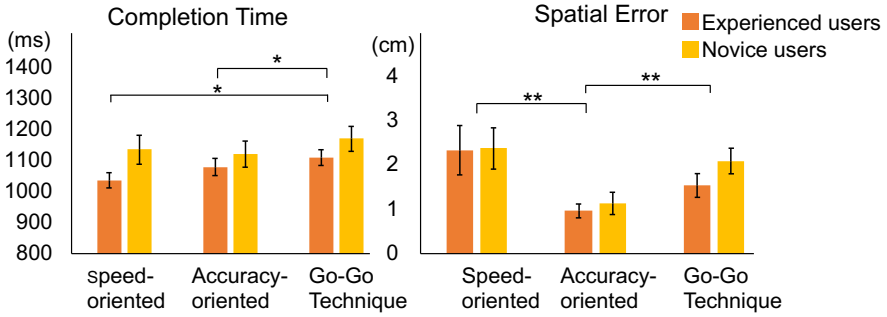


Figure 6. Results of the comparative study on three designs (two global designs and Go-Go technique) and over experienced and novice user groups. The error bars denote 1 standard deviation. The one-star ($*$) and two-star ($**$) symbols indicate $p < 0.05$ and $p < 0.001$ significant differences, respectively.

and participant group. For the mean *completion time* of the *experienced users*, the speed-oriented design, the accuracy-oriented design, and Go-Go Technique were 1,038 ms ($sd = 77.64$), 1,081 ms ($sd = 87.72$), and 1,111 ms ($sd = 80.32$), respectively. For the mean completion time of the *novice users*, the speed-oriented design, the accuracy-oriented design, and Go-Go Technique were 1,124 ms ($sd = 126.28$), 1,117 ms ($sd = 127.10$), and 1,167 ms ($sd = 120.88$), respectively. For the mean *spatial error* of the *experienced users*, the speed-oriented design, the accuracy-oriented design, and Go-Go Technique were 2.34 cm ($sd = 1.76$), 0.97 cm ($sd = 0.49$), and 1.55 cm ($sd = 0.85$), respectively. For the mean spatial error of the *novice users*, the speed-oriented design, the accuracy-oriented design, and Go-Go Technique were 2.03 cm ($sd = 1.17$), 0.96 cm ($sd = 0.68$), and 1.83 cm ($sd = 0.78$), respectively.

We conducted a mixed-design analysis of variance (mixed ANOVA) to examine the effect of interfaces and user experience levels. Sphericity was assessed with Mauchley's test, and if violated, Greenhouse-Geisser corrections were employed. The results revealed significant within-subject effects for both completion time ($F(2, 36) = 7.483, p < 0.005$) and spatial errors ($F(1.432, 25.781) = 19.284, p < 0.001$). Tests of between-subjects effects indicated there were no differences found between user experience levels (all $p > 0.05$). However, the generally higher completion times for novice users suggests a learning effect.

Pairwise comparisons were run for all conditions on both completion time and spatial errors and the significant differences are noted in Figure 6. For completion time, both the speed-oriented and the accuracy-oriented designs outperformed the baseline design (all $p < 0.05$). No significant difference was found between the two global designs. With respect to spatial errors, the accuracy-oriented design was shown to be significantly better than the speed-oriented design and the baseline design (all $p < 0.001$). However, the speed-oriented design was not superior to the baseline in reducing spatial errors. Overall, both global designs brought significantly better or comparable performances to the users for both metrics. As expected, the speed-oriented design successfully delivered shorter completion times than the baseline, and the accuracy-oriented design significantly reduced spatial errors.

Summary

In this study we showed the efficacy of deriving global Pareto-optimal design instances from user-specific observations using the Global GP method. The results of the comparative evaluation show that our global designs bring better or comparable performances for the user group compared with the baseline. This approach highlights how MOBO and the Global GP method can eliminate much of the design labor that is typically required to aggregate the preferences and behaviors of multiple individuals into a single sound design.

Study 2: Group-to-Individual Design with the Warm-start GP

This second study validates our method for initializing the multi-objective Bayesian optimization procedure in order to enhance the efficiency of interaction optimization at the individual level. We refer to this initialization process as constructing a Warm-Start GP. This demonstration of the Warm-Start GP is contextualized by the design challenge of producing an *adaptive* touch-button for a temporal pointing task. To further highlight the capabilities of the MOBO procedure, we tackle this problem using a design parameterization of five variables and with respect to three objectives. One of these design objectives is based on a subjective user rating which has high relevance to many HCI design problems.

The approach we employed in this study to validate the Warm-Start GP method involved two phases. In Phase 1, participants were exposed to the standard MOBO procedure. The dataset generated in Phase 1 was then used to produce the MOBO initialization for Phase 2. In Phase 2, the same group of participants (which we refer to as the *experienced user group*) and a new group of users (which we refer to as the *novice user group*) completed the MOBO procedure in two different variants: once with the warm-start initialization and once with the default initialization. Using this protocol, we investigate whether our Warm-Start GP model can effectively leverage previously collected information on user performances and deliver more rapid adaptation to individual users.

Table 3. Design parameterization of the touch-button.

Design Parameter		Range
x_1:	Button activation force level	[15 g, 1515 g]
x_2:	Vibration activation force level	[15 g, 1515 g]
x_3:	Initial vibration amplitude	[0 g, 3.2 g]
x_4:	Final vibration amplitude	[0 g, 3.2 g]
x_5:	Vibration duration	[0 s, 1.5 s]

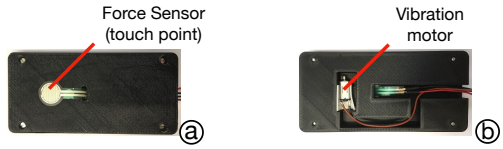


Figure 7. (a) The smartphone prototype: Users were instructed to touch the center of the force sensor. (b) The vibration motor is mounted inside the smartphone prototype.

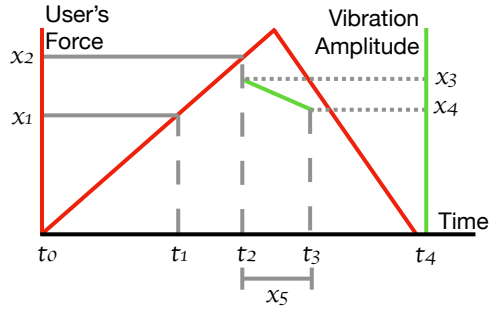


Figure 8. Illustrative design example where x_1 (Button-Activation Point) is a lower force threshold than x_2 (Vibration Point). The user starts pushing the button at t_0. The detected force exceeds the activation point at t_1 and the button is activated. At t_2, the force reaches the vibration point and the tactile cue is triggered. The initial vibration amplitude is set at x_3 and the vibration linearly decays until the amplitude becomes x_4. The vibration duration is x_5, and thus the vibration stops at t_3. The user's finger is completely lifted from the sensor at t_4, and the button is reset.

The design problem examined has high relevance given that pressing a touch-button is a fundamental interaction on touchscreen devices. When the finger contacts a button, a key-click vibration signal is generated to notify the user of the activation of the matching function. Such a key-click signal affects the user's typing speed and errors on a soft keyboard (50), and subjective preferences (51). The design of a touch-button is not a trivial task, though. Previous research has shown that an optimal point to trigger a button is not as the finger makes contact with the button (52; 53). Rather, it is somewhere within the travel range. As our first study shows, determining proper haptic feedback for target selection is also not straightforward. Various haptic feedback leads to different sensations and performances. Further, the optimal point to render the haptic cue is not at the same point as where the selection happens (Figure 5). Additionally, vibration feedback is a continuous cue which lasts for a certain duration (54), and yet, the button triggering is momentary. Determining when and how to render a continuous cue to match a momentary event has not been previously explored. Despite the prevalence of touch-buttons in our daily experiences and the challenges of designing them, there have been few attempts to investigate and iterate its design. Previous research attempted to optimize single objectives with iterative experimentation, including maximizing the button's information communication (3; 55; 56), minimizing typing error (50), and creating realistic physical-button sensations (54). However, iterative experimentation is not conducive to the efficient exploration of a multi-dimensional design space (3; 55; 57; 58) and risks omitting promising designs. Liao et al. (52) applied Bayesian optimization to the task of designing the haptic characteristics of a push-button. However, Liao et al. (52)'s study is on physical buttons and the method is limited to optimizing a single objective.

In this study, we sought to derive an adaptive model of touch-button pressing for temporal pointing tasks. Temporal pointing refers to tasks consisting of entering certain discrete inputs within a short time window (59). It is not only a common interaction for games in which a function must be elicited at a particular moment (for instance, to attack an enemy); it is also a synchronization task (60) that occurs in daily input experiences. To our knowledge, no prior work has applied multi-objective optimization methods to search for the optimal button design for such a task.

Design Parameterization and Objectives

Prior work has demonstrated that the activation point of the push-button affects typing speed (61), and that the vibration emission timing impacts temporal errors (52). We translate the depth sensing of a push-button to the pressing force level on a touch-button. This approach is illustrated in Figure 8, where the activation of the button functionality and the vibration may occur at different times and after the user's first initial contact with the button. Each event is triggered when a certain level of pressing force is detected. In this illustrated example, the force threshold for activating the button is lower than the force threshold for activating vibration, so the button would be activated prior to the vibration cue being generated.

We examine five design parameters as summarized in Table 3. x_1 (Button-Activation Point) and x_2 (Vibration Point) were explained in the previous paragraph and are illustrated in Figure 8. x_3 (Initial Vibration Amplitude) and x_4 (Final Vibration Amplitude) define the amplitude level of the vibration cue after the detected force exceeds the vibration point—that is, the moment that the vibration should be generated. When x_3 and x_4 have different values, the amplitude linearly increases or decreases over the Vibration Duration, which is defined as x_5.

We aim to maximize the temporal performance (59) of button pressing and the user's subjective rating. The temporal performance is assessed by two separate objective measures: the mean value and the standard deviation of the temporal errors of all the presses. The three objectives that govern the optimization process are summarized in Table 4.

Table 4. The design objectives of the touch-button.

Objective	Description
Temporal Error Mean	The temporal pointing is more accurate if this value is smaller.
Temporal Error Standard Deviation	The temporal pointing is more precise if this value is smaller.
Subjective User Rating	The vibration cue matches the click interaction more if this value is higher. Values from 0 to 100.

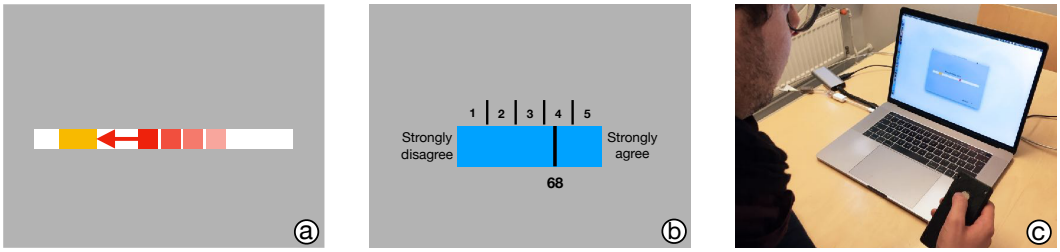


Figure 9. (a) A simplified sketch of the study interface during button pressing. The participant is asked to activate the button (the red bullet turns blue) when the red bullet reaches the yellow target area. (b) After 24 presses, the user is then asked to rate the vibration cue. (c) The study interaction.

Phase 1: User-Specific Optimizations

In Phase 1 of the study, participants completed the standard MOBO procedure. The data collected from all participants in Phase 1 was then used to generate the warm-start initialization. This initialization was subsequently evaluated in Phase 2 as described later in Section .

Participants: In total, 22 participants were recruited from our local institution for the whole study. Among them, 10 participants completed both phases, which were performed on different days (3 female, average age = 23.5, sd = 5). This group of users is referred to as the "experienced user group". The total duration for these 10 participants was 90 minutes, and participants received two movie tickets, worth €24, in appreciation for their involvement. 12 additional participants were invited to participate *only in the second phase* (4 female, average age = 22.9, sd = 2.4). The duration for these participants was about 30 minutes, and they received one movie ticket, worth €12. This group of users is referred to as the "novice user group".

Task: Participants (the ones in the experienced user group) were asked to perform a temporal pointing task. The graphical interface for the task is shown in Figure 9a. A red 'bullet' moves from right to left along the white bar as illustrated in Figure 9a. Participants were instructed to activate the button when the bullet reached the center of the yellow target zone. When the button is activated, the red bullet turns blue.

Apparatus and Prototype: We implemented a smartphone prototype (6 cm × 12.5 cm × 1 cm) with a force sensing resistor (FSR 402[†]) and an embedded vibration motor (Precision Microdrives 308-102[‡], rise time 21 ms) as shown in Figure 7a and b. The vibration motor was driven by a motor driver (Sparkfun DRV2605L[§]), and the motor and the sensor were controlled by an Arduino Uno. The study interface shown in Figure 9a and b was implemented in Processing.

Setup and Procedure: In each iteration of the task, participants were presented with two levels of difficulty: easy (bullet moving at 625 pixels/second rate) and hard (1000 pixels/second). The two difficulty levels were presented in random order. Both difficulty levels required the participant to complete 12 trials (or presses). The first five presses at a given level were allocated as familiarization trials and their data was not used for performance calculation. The remaining seven presses were used to calculate the mean and standard deviation of the temporal error. After all presses were completed at both difficulty levels, participants were asked to rate the vibration design iteration they had just experienced. The statement, *"The vibration cue synchronizes (matches) with the button pressing interaction"*, was presented to participants as illustrated in Figure 9b. Participants were asked to submit their subjective rating on a scale from 0 to 100; 100 for strongly agree and 0 for strongly disagree. Five levels $[1, 2, 3, 4, 5]$ were shown above the continuous slider to provide a coarse reference frame. A two-minute break was given after every 15 iterations of the task to avoid fatigue. Phase 1 lasted for approximately 60 minutes per participant.

MOBO Hyperparameters: The design configurations used in the first five task iterations were randomly selected. The subsequent 45 task iterations used design configurations proposed by the MOBO procedure. The design space was discretized into 45 possible cue configurations.

Phase 2: Evaluation of the Warm-Start GP

A total of 500 (50 design configurations × 10 participants) observations were collected in Phase 1 of the study. We applied the Warm-Start GP method on this dataset to select

[†] https://www.interlinkelectronics.com/fsr-402
[‡] https://www.precisionmicrodrives.com/product/308-102-8mm-vibration-motor-15mm-type
[§] https://www.sparkfun.com/products/14538

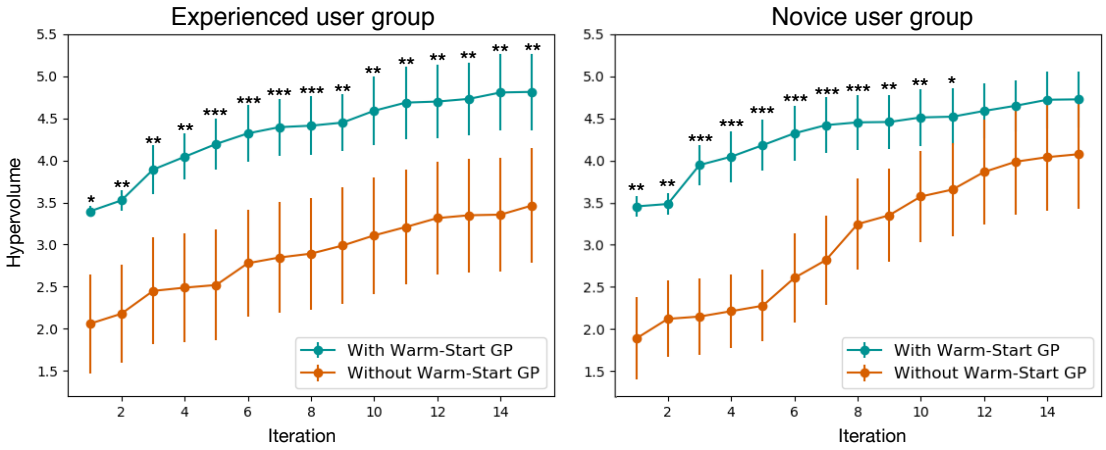


Figure 10. Impact of a Warm-Start GP initialization on experienced and novice users. The hypervolume increases throughout the 15 iterations for both experienced users and novice users during the evaluation. Error bars denote one standard error. We also indicate the significance level for the difference in hypervolume for the conditions with and without initialization at each iteration. The one-star $(*)$ indicates $p < 0.05$, two-star $(**)$ indicates $p < 0.01$, and three-star $(***)$ indicates $p < 0.001$.

a subset of representative points as an initialization for a new GP model. We hypothesized that the number of 'warm-start' points included in the initialization would influence the effectiveness of the adaptation. The reasoning behind this was as follows. If too few points are included, they will not provide a meaningful prior. However, selecting too many 'warm-start' points may give rise to a high initial hypervolume with the consequence that these selected data points may consistently dominate new observations captured during the personalization process. This would lead to a lack of design adaptation for the individual user. To select a reasonable number of points, we created three warm-start models with 5, 10, and 15 initial representative observations. We then used the data from the 10 participants in Phase 1 to construct 10 surrogate GP models to simulate each individual's behavior. We let the new warm-start model sample from each individual's surrogate model using Bayesian optimization to simulate the user-specific optimization process and recorded the simulated hypervolume increase. The results of the simulation indicate that 5 initial warm-start points is the best setting of the three alternatives evaluated and results in the highest hypervolume increase within 15 iterations.

Participants: Two groups of users were recruited for Phase 2. The *experienced users* are the same 10 participants who attended Phase 1. We further recruited 12 additional *novice users* to validate the effectiveness of the Warm-Start GP on new users (4 female, average age $=$ 22.9, sd $=$ 2.4). Both groups went through an identical procedure as described below.

Evaluation Setup The evaluation was conducted two days after Phase 1 and employed a repeated-measures design with one factor and two conditions: MOBO performed with the Warm-Start GP initialization and MOBO performed without including any prior observations. The condition order was counter-balanced. The tasks given to the participants were the same as in Phase 1. For the MOBO condition without initialization, the design configurations in the first five task iterations were randomly selected. A further 10 task iterations were performed with designs proposed by the MOBO procedure. In the MOBO condition with the warm-start initialization, no initial random sampling was performed such that only 15 task iterations were performed with all designs proposed by the MOBO procedure.

Results of Evaluating the Warm-start GP The hypervolume increase for each condition over the experiment iterations is plotted in Figure 10. We performed two separate 2-way repeated measures ANOVAs to analyze the effect of *initialization* (with or without the Warm-Start GP initialization) and *iterations* on the *hypervolume increase* for each of the experienced user group and the novice user group. For the experienced user group, we found no statistically significant interaction between the effect of initialization and iterations ($F(14, 126) = 0.38$, $p > 0.05$). Simple main effects analysis showed that the hypervolume was significantly higher when the experienced users start with the Warm-Start GP initialization than without ($F(1, 9) = 23.43$, $p < 0.001$). Simple main effects analysis also showed that there were significant differences between the iterations for the experienced user group ($F(14, 126) = 31.88$, $p < 0.001$). We further performed paired samples t-tests to compare the hypervolume between the with and without Warm-Start GP initialization conditions at each iteration. There were significant differences throughout all the iterations (all $p < 0.05$, see significance level notation in Figure 10).

For the novice user group, there was a statistically significant interaction between the effect of initialization and iterations ($F(14, 154) = 8.25$, $p < 0.001$). Simple main effects analysis showed that the hypervolume was significantly higher when the novice users started with the Warm-Start GP initialization than without ($F(1, 11) = 19.24$, $p < 0.001$) and there was also a significant effect for iterations ($F(14, 154) = 32.17$, $p < 0.001$). We then ran paired samples t-tests to compare the hypervolume between

the with and without Warm-Start GP initialization conditions at each iteration. The results showed that the Warm-Start GP produced significantly higher hypervolume in iterations 1 to 11 (all $p < 0.05$, see significance level notation in Figure 10). There were no significant differences in the hypervolumes at iterations 12 to 15. This result shows that Warm-Start GP effectively supported faster exploration for novice users.

Overall, this result suggests that our selected warm-start points provided an appropriate prior and thus a useful initialization for delivering rapid adaptation to the individual users. Incorporating the Warm-Start GP initialization enabled the MOBO procedure to present more designs offering improvements in the design objectives for both novice and experienced users, manifesting as a larger final hypervolume. Another way to frame this result is that for novice users, just five iterations using the Warm-Start GP initialization yielded a set of higher performing designs than 15 iterations without any initialization.

Summary

This study demonstrates that the Warm-Start GP method can provide an initialization delivering a faster hypervolume increase than that obtained by MOBO starting with a standard initialization, which allows faster adaptation to the preferences and abilities of an individual user. The Warm-Start GP is effective not just for the same group of users but also for new users, indicating that the Warm-Start GP is an effective method for transferring a prior understanding of the design space to different user groups. Despite the generally positive findings, different groups of users may have various preferences and optimal designs. Therefore, collecting data points from a larger pool of users may produce a more general Warm-Start GP when targeting new groups. Additionally, clustering the users and generating various Warm-Start GPs may also result in faster adaptation.

Discussion and Future Work

The novelty of this work chiefly lies in the demonstration of two practical approaches facilitating the application of MOBO in interaction technique design. To this end, we have: (i) introduced the Global GP concept for extracting Pareto-optimal designs representative of a user group; (ii) introduced the Warm-Start GP method for initializing the MOBO procedure to enable more rapid adaptation at the individual level; and (iii) demonstrated the efficacy of both methods in two representative HCI design problems. Our methods effectively identified the group-optimized designs and reduced the time required for running individual optimizations.

While the studies presented exhibit promising results, our work also highlights several open research questions. The presented approach employs user-specific observations to construct a global model in a post-hoc step. In the future, it is worth investigating an online global model that iteratively updates as more users' data is aggregated. Furthermore, the current method unifies the observations from all users into a GP model. This approach may lead to unwanted outcomes if there are distinct user groups that favor drastically different designs. Future research should also explore the use of more advanced algorithms such as hierarchical Gaussian process regression (62), which can cluster individual user groups according to their design preferences.

Our Warm-Start GP method selects a subset of prior observations to construct an adaptive model; however, the number of selected observations can affect the adaptation behavior. We determine the appropriate number of 'warm-start' points through simulation, but future work should explore more efficient techniques. Also related to the Warm-Start GP, the fact that different user groups may exhibit distinct behaviors with regard to an interaction technique suggests that there may be benefits to deriving group-specific warm-start initialization.

An assumption made throughout this study is that user performance or experience does not drastically vary due to time or order effects. In reality, there are several well-known user-related factors other than the design itself which are likely to affect the user's performance including fatigue, learning effects, and attention. With subjective ratings, a user's preference may drift or be influenced by new exposures. For instance, the rating of a system tends to be biased by recent trials. These effects were not incorporated into our performance models. How to resolve the issues raised by the uncertainty of human users remains a compelling open research question.

Conclusions

Interaction design often involves a large number of parameters which need to be decided while also considering multiple design objectives. Although multi-objective Bayesian optimization (MOBO) offers a principled method for guiding design exploration, prior work has largely ignored the practical need in interaction technique design to meet the requirements of a population or group of users. To bridge this gap, we present: (i) the *Global GP* to identify group-level optimal designs; and (ii) the *Warm-Start GP* for rapid adaptation based upon a suitable prior extracted from the group-level data. We demonstrate the effectiveness of the *Global GP* and *Warm-Start GP* methods in two challenging and representative design problems. We show that the *Global GP* facilitates the identification of group-level Pareto-optimal designs, and that these designs are indeed competitive with a design arrived at by conventional means. We also show that the *Warm-Start GP* improves the efficiency of individual optimization by incorporating group-level data in the initialization for MOBO. Both methods are readily applied to other design problems involving multiple objectives and we hope that the guidance provided in this paper will promote wider uptake of MOBO in interaction technique design.

Open Science

The program and materials in this paper are released on our project page: *to be released*. The Python library is open-sourced for designers and developers to use.

Acknowledgements

References

[1] Hornbæk K et al. Some whys and hows of experiments in human–computer interaction. *Foundations and Trends® in Human–Computer Interaction* 2013; 5(4): 299–373.

[2] Gergle D and Tan DS. *Experimental Research in HCI*. New York, NY: Springer New York. ISBN 978-1-4939-0378-8, 2014. pp. 191–227. DOI:10.1007/978-1-4939-0378-8_9. URL https://doi.org/10.1007/978-1-4939-0378-8_9.

[3] Chen H, Park J, Dai S et al. Design and evaluation of identifiable key-click signals for mobile devices. *IEEE Transactions on Haptics* 2011; 4(4): 229–241.

[4] Shahriari B, Swersky K, Wang Z et al. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE* 2016; 104(1): 148–175. DOI:10.1109/JPROC.2015.2494218. Conference Name: Proceedings of the IEEE.

[5] Chan L, Liao YC, Mo GB et al. Investigating positive and negative qualities of human-in-the-loop optimization for designing interaction techniques. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA. DOI:10.1145/3491102.3501850. URL https://doi.org/10.1145/3491102.3501850.

[6] Liao YC, Dudley JJ, Mo GB et al. Interaction design with multi-objective bayesian optimization. *IEEE Pervasive Computing* 2023; : 1–10DOI:10.1109/MPRV.2022.3230597.

[7] Khajah MM, Roads BD, Lindsey RV et al. Designing Engaging Games Using Bayesian Optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16, New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-3362-7, pp. 5571–5582. DOI:10.1145/2858036.2858253. URL https://doi.org/10.1145/2858036.2858253.

[8] Kadner F, Keller Y and Rothkopf C. Adaptifont: Increasing individuals' reading speed with a generative font model and bayesian optimization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21, New York, NY, USA: Association for Computing Machinery. ISBN 9781450380966. DOI:10.1145/3411764.3445140. URL https://doi.org/10.1145/3411764.3445140.

[9] Dudley JJ, Jacques JT and Kristensson PO. Crowdsourcing Interface Feature Design with Bayesian Optimization. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19, New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5970-2, pp. 1–12. DOI:10.1145/3290605.3300482. URL https://doi.org/10.1145/3290605.3300482.

[10] Koyama Y, Sato I, Sakamoto D et al. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics* 2017; 36(4): 48:1–48:11. DOI:10.1145/3072959.3073598. URL https://doi.org/10.1145/3072959.3073598.

[11] Koyama Y, Sato I and Goto M. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics* 2020; 39(4): 88:88:1–88:88:12. DOI:10.1145/3386569.3392444. URL https://doi.org/10.1145/3386569.3392444.

[12] Brochu E, Brochu T and de Freitas N. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '10, Goslar, DEU: Eurographics Association, pp. 103–112.

[13] Oulasvirta A, Dayama NR, Shiripour M et al. Combinatorial optimization of graphical user interface designs. *Proceedings of the IEEE* 2020; 108(3): 434–464.

[14] Gajos K and Weld DS. SUPPLE: Automatically Generating User Interfaces. *IUI '04* 2004; : 8.

[15] Karrenbauer A and Oulasvirta A. Improvements to keyboard optimization with integer programming. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. UIST '14, New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-3069-5, pp. 621–626. DOI:10.1145/2642918.2647382. URL https://doi.org/10.1145/2642918.2647382.

[16] Lindlbauer D, Feit AM and Hilliges O. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. New Orleans LA USA: ACM. ISBN 978-1-4503-6816-2, pp. 147–160. DOI:10.1145/3332165.3347945. URL https://dl.acm.org/doi/10.1145/3332165.3347945.

[17] Borji A and Itti L. Bayesian optimization explains human active search. In Burges C, Bottou L, Welling M et al. (eds.) *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2013/file/a3f390d88e4c41f2747bfa2f1b5f87db-Paper.pdf.

[18] Nielsen JBB, Nielsen J and Larsen J. Perception-Based Personalization of Hearing Aids Using Gaussian Processes and Active Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 2015; 23(1): 162–173. DOI:10.1109/TASLP.2014.2377581. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[19] Snoek J. *Bayesian Optimization and Semiparametric Models with Applications to Assistive Technology*. PhD Thesis, University of Toronto, Toronto, Ontario, Canada, 2013.

[20] Piovarči M, Kaufman DM, Levin DIW et al. Fabrication-in-the-loop co-optimization of surfaces and styli for drawing haptics. *ACM Trans Graph* 2020; 39(4). DOI:10.1145/3386569.3392467. URL https://doi.org/10.1145/3386569.3392467.

[21] Knowles J. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 2006; 10(1): 50–66. DOI:10.1109/TEVC.2005.851274. Conference Name: IEEE Transactions on Evolutionary Computation.

[22] Zuluaga M, Krause A and Püschel M. e-PAL: An Active Learning Approach to the Multi-Objective Optimization Problem. *Journal of Machine Learning Research* 2016; 17(104): 1–32. URL http://jmlr.org/papers/v17/15-047.html.

[23] Hernandez-Lobato D, Hernandez-Lobato J, Shah A et al. Predictive Entropy Search for Multi-objective Bayesian Optimization. In *International Conference on Machine Learning*. PMLR, pp. 1492–1501. URL http://proceedings.mlr.press/v48/

hernandez-lobatoa16.html. ISSN: 1938-7228.

[24] Picheny V. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing* 2015; 25(6): 1265–1280. DOI: 10.1007/s11222-014-9477-x. URL https://doi.org/10.1007/s11222-014-9477-x.

[25] Hayward V, Choksi J, Lanvin G et al. Design and Multi-Objective Optimization of a Linkage for a Haptic Interface. In Lenarčič J and Ravani B (eds.) *Advances in Robot Kinematics and Computational Geometry*. Dordrecht: Springer Netherlands. ISBN 978-94-015-8348-0, 1994. pp. 359–368. DOI:10.1007/978-94-015-8348-0_36. URL https://doi.org/10.1007/978-94-015-8348-0_36.

[26] Sridhar S, Feit AM, Theobalt C et al. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. Seoul, Republic of Korea: ACM Press. ISBN 978-1-4503-3145-6, pp. 3643–3652. DOI:10.1145/2702123.2702136. URL http://dl.acm.org/citation.cfm?doid=2702123.2702136.

[27] Dunlop M and Levine J. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12, New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-1015-4, pp. 2669–2678. DOI: 10.1145/2207676.2208659. URL https://doi.org/10.1145/2207676.2208659.

[28] Feit AM, Sridhar S and Bachynskyi M. Towards Multi-Objective Optimization for UI Design. *CHI '15 Workshop on Principles, Techniques and Perspectives on Optimization and HCI* 2015; : 4.

[29] Shah A and Ghahramani Z. Pareto Frontier Learning with Expensive Correlated Objectives. In *International Conference on Machine Learning*. PMLR, pp. 1919–1927. URL http://proceedings.mlr.press/v48/shahc16.html. ISSN: 1938-7228.

[30] Zitzler E and Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 1999; 3(4): 257–271. DOI:10.1109/4235.797969. Conference Name: IEEE Transactions on Evolutionary Computation.

[31] Titsias M. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Artificial Intelligence and Statistics*. PMLR, pp. 567–574. URL http://proceedings.mlr.press/v5/titsias09a.html. ISSN: 1938-7228.

[32] Seeger MW, Williams CK and Lawrence N. Fast Forward Selection to Speed Up Sparse Gaussian Process Regression. In *AISTATS*.

[33] Bauer M, van der Wilk M and Rasmussen CE. Understanding probabilistic sparse gaussian process approximations. NIPS'16. URL http://arxiv.org/abs/1606.04820. 1606.04820.

[34] Cao Y, Brubaker MA, Fleet DJ et al. Efficient optimization for sparse gaussian process regression. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'13, Curran Associates Inc., pp. 1097–1105.

[35] Burt DR, Rasmussen CE and Wilk Mvd. Convergence of sparse variational inference in gaussian processes regression. In *Journal of Machine Learning Research*, volume 21. pp. 1–63. URL http://jmlr.org/papers/v21/19-1015.html.

[36] Poupyrev I, Billinghurst M, Weghorst S et al. The go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*. UIST '96, New York, NY, USA: Association for Computing Machinery. ISBN 0897917987, p. 79–80. DOI:10.1145/237091.237102. URL https://doi.org/10.1145/237091.237102.

[37] Argelaguet Sanz F and Andujar C. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers and Graphics* 2013; 37(3): 121–136. DOI:10.1016/j.cag.2012.12.003. URL https://hal.archives-ouvertes.fr/hal-00907787.

[38] Bowman DA, Johnson DB and Hodges LF. Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '99, New York, NY, USA: Association for Computing Machinery. ISBN 1581131410, p. 26–33. DOI: 10.1145/323663.323667. URL https://doi.org/10.1145/323663.323667.

[39] POUPYREV I and ICHIKAWA T. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages and Computing* 1999; 10(1): 19 – 35. DOI:https://doi.org/10.1006/jvlc.1998.0112. URL http://www.sciencedirect.com/science/article/pii/S1045926X98901124.

[40] Frees S, Kessler GD and Kay E. Prism interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)* 2007; 14(1): 2–es.

[41] Meyer DE, Abrams RA, Kornblum S et al. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 1988; 95(3): 340.

[42] König WA, Gerken J, Dierdorf S et al. Adaptive pointing–design and evaluation of a precision enhancing technique for absolute pointing devices. In *IFIP Conference on Human-Computer Interaction*. Springer, pp. 658–671.

[43] Casiez G and Roussel N. No more bricolage! methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11, New York, NY, USA: Association for Computing Machinery. ISBN 9781450307161, p. 603–614. DOI:10.1145/2047196.2047276. URL https://doi.org/10.1145/2047196.2047276.

[44] Nancel M, Pietriga E, Chapuis O et al. Mid-air pointing on ultra-walls. *ACM Trans Comput-Hum Interact* 2015; 22(5). DOI:10.1145/2766448. URL https://doi.org/10.1145/2766448.

[45] Yun J, Lim Yk, Kim KE et al. Interactivity crafter: An interactive input-output transfer function design tool for interaction designers. *Archives of Design Research* 2015; 28: 21–37. DOI:10.15187/adr.2015.08.28.3.21.

[46] Pfeiffer M and Stuerzlinger W. 3d virtual hand pointing with ems and vibration feedback. In *2015 IEEE Symposium on 3D*

User Interfaces (3DUI). pp. 117–120. DOI:10.1109/3DUI.2015.7131735.

[47] Sallnäs E and Zhai S. Collaboration meets fitts' law: Passing virtual objects with and without haptic force feedback. In *INTERACT*. IOS Press.

[48] Cha Y and Myung R. Extended fitts' law for 3d pointing tasks using 3d target arrangements. *International Journal of Industrial Ergonomics* 2013; 43(4): 350 – 355. DOI:https://doi.org/10.1016/j.ergon.2013.05.005. URL http://www.sciencedirect.com/science/article/pii/S0169814113000723.

[49] Wang D, Ohnishi K and Xu W. Multimodal haptic display for virtual reality: A survey. *IEEE Transactions on Industrial Electronics* 2019; 67(1): 610–623.

[50] Zhaoyuan Ma, Edge D, Findlater L et al. Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard. In *2015 IEEE World Haptics Conference (WHC)*. pp. 220–227.

[51] Pitts MJ, Williams MA, Wellings T et al. Assessing subjective response to haptic feedback in automotive touchscreens. AutomotiveUI '09, New York, NY, USA: Association for Computing Machinery. ISBN 9781605585710, p. 11–18. DOI:10.1145/1620509.1620512. URL https://doi.org/10.1145/1620509.1620512.

[52] Liao YC, Kim S, Lee B et al. Button simulation and design via fdvv models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20, New York, NY, USA: Association for Computing Machinery. ISBN 9781450367080, p. 1–14. DOI:10.1145/3313831.3376262. URL https://doi.org/10.1145/3313831.3376262.

[53] Kim S, Son J, Lee G et al. Tapboard: Making a touch screen keyboard more touchable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13, New York, NY, USA: Association for Computing Machinery. ISBN 9781450318990, p. 553–562. DOI:10.1145/2470654.2470733. URL https://doi.org/10.1145/2470654.2470733.

[54] Kim S and Lee G. Haptic feedback design for a virtual button along force-displacement curves. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13, New York, NY, USA: ACM. ISBN 978-1-4503-2268-3, pp. 91–96. DOI:10.1145/2501988.2502041. URL http://doi.acm.org/10.1145/2501988.2502041.

[55] Richter H, Ecker R, Deisler C et al. Haptouch and the 2+1 state model: Potentials of haptic feedback on touch based in-vehicle information systems. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '10, New York, NY, USA: Association for Computing Machinery. ISBN 9781450304375, p. 72–79. DOI:10.1145/1969773.1969787. URL https://doi.org/10.1145/1969773.1969787.

[56] Liao YC, Chen YC, Chan L et al. Dwell+: Multi-level mode selection using vibrotactile cues. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17, New York, NY, USA: Association for Computing Machinery. ISBN 9781450349819, p. 5–16. DOI:10.1145/3126594.3126627. URL https://doi.org/10.1145/3126594.3126627.

[57] Park C, Yoon J, Oh S et al. *Augmenting Physical Buttons with Vibrotactile Feedback for Programmable Feels*. New York, NY, USA: Association for Computing Machinery. ISBN 9781450375146, 2020. p. 924–937. URL https://doi.org/10.1145/3379337.3415837.

[58] Chang Z, Ta TD, Narumi K et al. Kirigami haptic swatches: Design methods for cut-and-fold haptic feedback mechanisms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20, New York, NY, USA: Association for Computing Machinery. ISBN 9781450367080, p. 1–12. DOI:10.1145/3313831.3376655. URL https://doi.org/10.1145/3313831.3376655.

[59] Lee B and Oulasvirta A. Modelling error rates in temporal pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16, New York, NY, USA: Association for Computing Machinery. ISBN 9781450333627, p. 1857–1868. DOI:10.1145/2858036.2858143. URL https://doi.org/10.1145/2858036.2858143.

[60] Wing A and Kristofferson A. Timing of interresponse intervals. *Attention Perception and Psychophysics* 1973; 13: 455–460. DOI:10.3758/BF03205802.

[61] Kim S, Lee B and Oulasvirta A. Impact activation improves rapid button pressing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18, New York, NY, USA: ACM. ISBN 978-1-4503-5620-6, pp. 571:1–571:8. DOI:10.1145/3173574.3174145. URL http://doi.acm.org/10.1145/3173574.3174145.

[62] Park S and Choi S. Hierarchical Gaussian Process Regression. In *Proceedings of 2nd Asian Conference on Machine Learning*. JMLR Workshop and Conference Proceedings, pp. 95–110. URL http://proceedings.mlr.press/v13/park10a.html. ISSN: 1938-7228.

Publication IV

Yi-Chi Liao, Sunjun Kim, Byungjoo Lee, Antti Oulasvirta. Button Simulation and Design via FDVV Models. In *2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, April 2020.

Button Simulation and Design via FDVV Models

Yi-Chi Liao[1] **Sunjun Kim**[1,2,3] **Byungjoo Lee**[2] **Antti Oulasvirta**[1]
[1]Aalto University, Finland [2] KAIST, Republic of Korea [3]DGIST, Republic of Korea
yi-chi.liao@aalto.fi, sunjun.kim@aalto.fi, byungjoo.lee@kaist.ac.kr, antti.oulasvirta@aalto.fi

ABSTRACT
Designing a push-button with desired sensation and performance is challenging because the mechanical construction must have the right response characteristics. Physical simulation of a button's force–displacement (FD) response has been studied to facilitate prototyping; however, the simulations' scope and realism have been limited. In this paper, we extend FD modeling to include vibration (V) and velocity-dependence characteristics (V). The resulting FDVV models better capture tactility characteristics of buttons, including snap. They increase the range of simulated buttons and the perceived realism relative to FD models. The paper also demonstrates methods for obtaining these models, editing them, and simulating accordingly. This end-to-end approach enables the analysis, prototyping, and optimization of buttons, and supports exploring designs that would be hard to implement mechanically.

Author Keywords
Button; haptic; modeling; simulation; tactility; force feedback; vibration; input device; haptic rendering; FD model; FDVV model.

CCS Concepts
•**Human-centered computing** → **Haptic devices; Interaction devices; Keyboards; Interface design prototyping;**

INTRODUCTION
This paper investigates the simulation and interactive design of push-buttons. Many push-button designs use a spring-loaded slider: when the slider is pushed to the activation point, a binary input is registered. Upon release, it returns to the initial state. More generally, buttons are transducers that register a discrete event from physical motion [28, 33, 49]. Numerous types exist, using spring-loading but also other mechanisms, such as rubber and metal domes. Interestingly, each button is unique in its *tactility* or tactile response characteristics [26,40]. Gamers, programmers, typists, and hobbyist groups alike have a keen interest in tactility, which is associated with sensory experience and performance [2, 12, 49]. However, despite the popularity and importance of buttons, researchers have paid relatively little attention to their design.

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376262

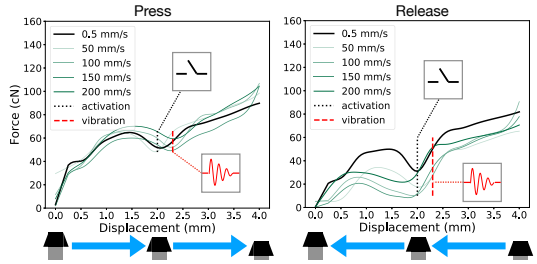


Figure 1. A force–displacement–vibration–velocity (FDVV) model represents speed-dependent physical responses of a button when pressed. We show methods for capturing button presses as FDVV models, rendering them in a physical simulator, and editing and optimizing these in software. The press and release models shown are for a 4 mm tactile button. Blue curves represent the corresponding (velocity-agnostic) force–displacement model typically measured by a probing machine with static and slow velocity.

Simulators have taken on a major role in most branches of engineering and design, where they are used to study reality, predict consequences of design decisions, and derive and optimize solutions. Dedicated simulators can do the same for button design. We believe that one obstacle has been the lack of an accurate yet practical simulation device. It should be able to realistically reproduce different tactilities so that designers and researchers could explore and test them at little cost.

Accurate simulation of button-pressing is challenging, though. Although a press and release occurs in about 100 ms [30], a wealth of sensory feedback is generated [9, 49]. Slow and fast mechanoreceptors [7, 57, 65] deliver information on spatial patterns, change in the contact area at the fingertip, the roughness of the contact support, stretching of the skin, subtle and rapid changes in force, and vibrations during a press. Meanwhile, proprioception provides feedback on displacement, as detected via the joints and muscles of the finger [59]. This information is transmitted through the spine at high rates, up to 1 kHz [13, 20], and contributes to constructing the felt sensation of a press [1, 49]. Hence, to reproduce a realistic button sensation, the simulator needs to capture those physical characteristics of a press that dominate that sensation.

While pre-existing simulators can render tactile and linear buttons [14, 37], we found that force–displacement (FD) approaches cannot accurately render other than a simple linear button. In addition, the perceived realism of the rendered buttons has not been properly evaluated, and no methods have been offered to help designers and engineers exploit such

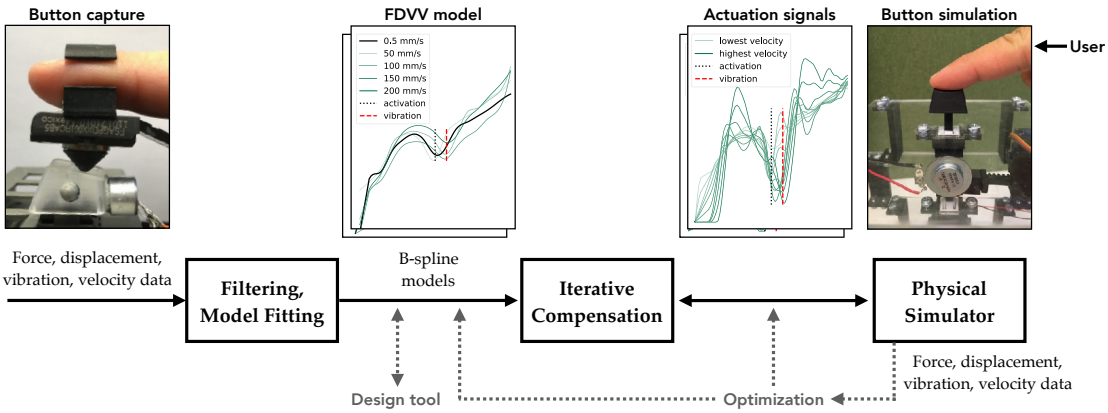


Figure 2. Overview: An end-to-end approach to button simulation. To capture an FDVV model of a button, sensors are placed on the finger, and the button is pressed multiple times. The resulting force, displacement, vibration, and velocity data are filtered, and lower-parametric B-splines are fitted for use of Bayesian Information Criteria (BIC) as the fitness metric. A designer can edit the model produced. To render the model with a given physical plant, an iterative compensation process computes how to cancel the plant's own transfer function. The resulting actuation signals drive the simulator.

simulation. We believe this to be due to three engineering challenges: (1) modeling, (2) simulator construction, and (3) model–simulator separation. Firstly, prior work has modeled buttons' response as the displacement-dependent change in force [14, 37, 46]. However, as we show in this paper, an FD model is adequate only for linear buttons pressed at extremely low speed. It is known that the physical response of a button depends *also* on velocity and acceleration [5, 6, 40], and that buttons elicit a vibration response. Vibration affects the perception of button-pressing [47], and it can be so prominent that it produces an illusion of force change [17, 25, 63]. Secondly, the simulators created thus far have fallen short of the operation rate needed for a button press. General haptic devices such as Phantom [16, 41, 58] and inForce [46] operate at 60 Hz, while the skin's mechanoreceptors fire at about 1,000 Hz and respond to tiny changes in force [13, 20]. Simplistic controller approaches may have further curbed attempts to render anything other than a linear button. To our knowledge, previous work has, at best, applied a linear PID controller to render the response at the force actuator of a simulator, although this is generally recognized as insufficient for nonlinear plants [8, 69]. We should emphasize, thirdly and finally, that the button models must be editable if they are to be of practical use. This calls for model–simulator separation. By avoiding the device-dependency of models, we can support the meaningful editing of buttons – by designers and by software.

To address these challenges, we propose an extended model and an end-to-end simulation pipeline around it. Our approach allows simulating more button types than previously, among them tactile-type buttons and buttons with different click reactions and travel ranges. Furthermore, it permits the analysis and editing of buttons. Our work centers on the *Force–Displacement–Vibration–Velocity* (FDVV) model, illustrated in Figure 1. It adds vibration response and velocity-dependence on top of the FD model. In our implementation, vibration is sampled thorough a microphone during a switch press, and multiple FD curves are sampled, at several speeds. We solve several engineering challenges connected with ambitions to capture and simulate buttons via FDVV models.

In particular, we present methods developed for (1) capturing the FDVV response of a button, (2) computing a lower-parametric FDVV model from the measurement data obtained, and (3) actuating an FDVV model for a given physical plant. Figure 2 gives an overview of the end-to-end approach. The data (force, displacement, vibration, and velocities) obtained during capture are filtered and modeled via B-splines [66], collapsing the data into a lower-dimensional and more manipulable model. For rendering it, we present a novel simulator construction for FDVV models. This is capable of detecting displacement to μm precision at a high sampling rate (1 kHz) and can produce a wide range of force (up to 4.4 N) and vibration (50 Hz – 20 kHz) feedback. In contrast to previous, 60 Hz simulators, our simulator can render high-fidelity vibration arising from the rapid force change near the snap and bottom-out points during the press [29]. It also has a mechanical limiter for rendering a button with various travel distances (0–6.2 mm). Thanks to the iterative compensation method, which translates an FDVV model into actuation signals that cancel out the simulator plant's own transfer function, one can assign button designs from software without hardware tweaks.

In summary, this paper makes three contributions. We present advances in modeling and simulator design that extend the range of supported button types. Secondly, we report the results of a controlled study showing that the FDVV model yields higher perceived realism than FD modeling. Finally, the approach opens new possibilities in design and prototyping; especially, by reducing the effort of exploring designs. We demonstrate applications in interactive button-editing, software-side optimization, and prototyping of innovative button designs the mechanical structure of which would be hard to fabricate. The general principles we applied in the simulation pipeline can also benefit future research toward accurate haptic rendering. For example, simulating elastic materials.

BACKGROUND

Physical buttons are electromechanical devices that make or break a signal upon pressing, then return to the initial (repushable) state upon release. There is incredible variety in the constructions that fit this definition. Our discussion here focuses on commonplace push-buttons of keyboards and button panels. Mechanical keyswitches, rubber domes, and metal domes are the most typical structures. Numerous other design parameters exist, such as physical properties of the keycap (width, slant, and key depth), the materials used (e.g., plastics), and system-level feedback (modalities and latencies) [35]. The response upon pressing can be characterized via the *the force–displacement function* or force curve [35, 54]. Actuation (press-down) and release curves often differ. The FD curve is known to affect not only sensation but also joint kinematics [22], muscle activity [27, 54, 55], and the user's aiming performance [49]. *Linear buttons* have the feel of pressing a spring; there is no tactile landmark or "bump" during press-down. A *tactile-type* button has a so-called snap ratio, which determines the strength of its tactile bump. Rubber-dome buttons utilize a snap ratio greater than 40%. Some tactile buttons emit an audible "click" sound near the snap point. *Travel distance* is the total distance before the keycap hits the bottom, and the distance at which the button is activated is called its *activation point* [28]. While these features can be modeled with FD curves, we stress again that FD neglects velocity and vibration characteristics.

Capturing and modeling physical buttons

There are two main approaches in haptics research applicable to the modeling of buttons. The first is an analytical one aimed at formulating equations that capture the mechanics of haptic interaction, which in the case of buttons would involve the forces, vibrations, etc. during a press. Since these interactions are complex, analytical models almost inevitably need to make simplifying assumptions. For example, the spring–mass-damper system in buttons could be described as a lumped mass [43]. Analytical models applicable to buttons include modeling of spring–mass-damper systems [5] and friction [19]. While prior applications to buttons and knobs do exist [3], the ones presented thus far are too low-parametric to capture the rich design space of push-buttons.

The second approach is a reality-based, or data-driven, one that starts with physical measurements and constructs models based on data. In the case of buttons, one starts by physically probing a button to measure the interaction forces between the button and the probe as displacement and even higher-order variables. Displacement has been approached in such a manner outside the button domain. One could cite as example applications the automotive gearshift [4], non-rigid materials [46, 60], and human tissue [50]. Displacement-only models are relatively simple. For buttons, this approach is insufficient. There have been studies examining higher-order variables, such as velocity [67] and acceleration [11, 38, 39]. This, however, complicates everything from measurement to simulation. Nonetheless, we followed the reality-based approach. We captured the forces involved, displacement values, vibration, and pressing velocities, and we found a way to collapse the data to a more understandable, lower-parametric model.

Haptics rendering

Our work is aligned with haptics research pursuing the creation of rich and realistic sensations [24]. While this area of research is too broad to review here, some relevant findings are worth mentioning. Firstly, research has looked at advanced factors affecting haptic perception, such as friction, temperature, or texture [15]. However, the focus has been on exploration or manipulation of objects, which is quite different from a keypress, which occurs in around 100 ms. The rapid compression and mechanical vibration of tissue in the fingertip are core elements of a button press. Secondly, general-purpose haptic simulators have been produced that could be used for buttons. The Phantom device [16, 41, 58] is a 6-DOF pen-type general force-rendering device capable of emulating the softness of deformable objects. However, a low operating rate (60 Hz), excessive degrees of freedom (six instead of one), and lack of vibrotactile simulation limit its use for buttons. Softness displays [44, 46, 62] too might aid in simulating the stiffness of a button, but these are restricted to so-called simple stiffness, which is inadequate for buttons. Finally, pseudo-force-feedback has been explored. By changing the contact area of the finger [15, 21] or using electro-tactile displays, one can create a softness-like sensation [23, 64]. However, this is not central to commodity push-buttons' design.

Button simulators

Prior work on button simulation has been limited to static (speed-agnostic) FD simulators, which have the limitations described above. Doerrer and Werthschuetzky [14] enabled users to edit FD curves in software, and Liao et al. [37] have presented an FD simulator. The Phantom haptic interface [41] also can render a virtual button. Yet, as other simulators do, these too need an exaggerated FD curve to render the "snap" feeling of a tactile button. These papers omitted velocity and vibration dynamics from their model. Moreover, to our knowledge, the perceived realism achievable by FD simulators has not been empirically validated thus far.

BUTTON CAPTURE

Most previous work has captured buttons by a single-FD curve model. This approach has become conventional also in the way manufacturers present buttons on datasheets. However, button-pressing involves complex phenomena affected by the varying stiffness and damping effects produced by mechanical design. The *stiffness effect* means that the resisting force changes as a function of the button's displacement. The *damping effect* entails the resisting force changing with the button's velocity. An FD-only model captures neither the damping effect nor the high-frequency structural vibrations of a button press.

We propose an extension to the physical measurement of the tactility characteristics of push-buttons. Our capture method features three novel elements: (1) We measure presses under different velocities. (2) A human finger is used for pressing, as opposed to a rigid, static-velocity probing object as in earlier work. This allows us to better cover the response envelope people encounter in everyday button-pressing, via a procedure that requires no more than a few minutes to complete per button. (3) We record vibrations, which are important for

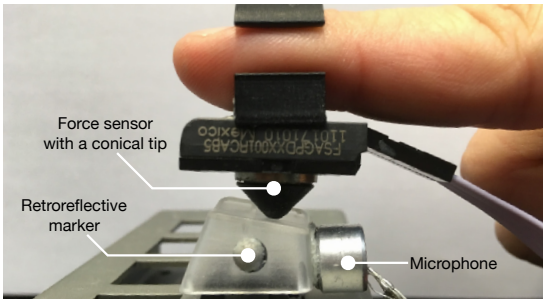


Figure 3. Button capture of a real button (4 mm tactile button). A force sensor is worn on the fingertip. Reflective markers (for motion tracking) and microphone are attached on keycap.

covering more advanced button types. This necessitates a more complex measurement setup than before.

Setup: Our measurement setup is shown in Figure 3. Two retroreflective markers are attached to sides of the keycap. A motion-tracking system (OptiTrack Prime 13, 256 FPS) records the displacement of the button during a press. Also, a microphone (KY-038) is attached to the keycap, to detect vibration during button presses. On the user's index finger is a force sensor (Honeywell, FSAGPDXX001RCAB5), with a conical tip (ABS, 3D printed, Bottom and top: 12 and 5 mm-diameter circles. Height: 5 mm) attached. A microprocessor (Adafruit Metro M0 Express) samples both sensors with its internal 10-bit ADC at 1 kHz frequency, which is the highest sampling rate allowed for the force sensor. Synchronization between the sensor data and the motion data is handled via an optical clapperboard (three 850 nm IR LEDs). The microprocessor is attached to a switch that turns on the LEDs. When the switch is triggered, the microprocessor and the motion tracking system record a reference point for synchronization.

Procedure: A participant is asked to wear the sensors and press the given button, following the instructions on the display. A velocity indicator is presented on this display, which shows animation for various velocities. The velocity indicator also creates a beep sound indicating the moment of contact and that of hitting the bottom. Note that we stated that the animation refers to the rate of pressing and the average velocity, not the moment-by-moment velocity. The participant is asked to press the button at the specified pace. We go through velocities of 50, 100, 150, and 200 mm/s, collecting 15 presses per velocity.

Example buttons: The paper reports on studies of six physical buttons: Cherry MX Clear and Brown (4 mm, tactile), Cherry MX Black and Red (4 mm, linear), HP PR1101U (3.6 mm, tactile), and MacBook Pro 2011 (2.2 mm, tactile). They have distinct haptic profiles (linear/tactile) and travel ranges.

FDVV MODELING

The raw measurement data must be transformed into a lower-dimensional FDVV model to allow efficient design and optimization. A series of preprocessing steps are followed to produce a synchronized and filtered dataset from multiple data sources. We then fit a B-spline model, using Bayesian Information Criteria (BIC) [31].

Preprocessing

The outputs of button capture are *(timestamp, force, sound)* data from the microprocessor and *(timestamp, the 3D position of marker1, the 3D position of marker2)* data from the motion tracker and vibration data. Our goals are to (1) filter out noise and outliers and (2) synchronize the two data sources.

Step 1, filtering: We pass force and vibration data through a low-pass filter for antialiasing, before the analog–digital conversion of the microprocessor. The filter consists of a resistor–capacitor circuit (R 333 ohm, C 1 uF) with a cutoff frequency of 500 Hz, using a Nyquist frequency of 1,000 Hz. Gaussian filters ($\sigma = 1.2$ *mm*) are then applied to both force and displacement data, separately, for further denoising.

Step 2, synchronization: We then synchronize the data from the microprocessor and motion tracker, using the IR LED light data to find keyframes. Because the microprocessor runs at 1,000 Hz and the motion tracker at 256 Hz, resampling is required before synchronizing of the two sides. We chose to upsample displacement data by matching the of the motion tracker to the timestamp of the microprocessor, via linear interpolation. The displacement data get resampled up to 1,000 Hz frequency, after which we have synchronized the signals. We need resampling also to register force data against displacement (here, sampled every 50 μm).

Step 3, outlier removal, averaging and smoothing: We use the resulting dataset to filter out incomplete button presses (ones where the button did not hit the bottom). Finally, we averaged the force data of the representative presses at each displacement point (every 50 μm). A Gaussian filter ($\sigma = 0.8$ *mm*) is then applied to smooth the averaged curve.

Step 4, synchronizing vibration data: Note that vibration data did not pass through *steps 3–4*. In most buttons, vibration is associated with the snap. This can be programmatically exploited to synchronize the vibration signal. On the other hand, one can ignore measurements detected in the beginning and the end portion, because these are caused mostly by the finger hitting the keycap or the keycap hitting the bottom. Hence, we consider only the middle part of the press, for which we use threshold-based event detection to find the onset of the vibration. For some buttons, this sound wave can be very subtle and rapid (typically <25 ms), making it hard to detect programmatically. To compensate for this, we can resort to a human observer (see "Iterative Compensation," below).

B-Spline Fitting

The preprocessed dataset is still too high-dimensional for editing by designers. For example, a typical 4 mm button requires approximately 800 parameters in our procedure. Hence, we use B-splines to achieve a lower-dimensional parametric model. While B-splines offer a suitable model for continuous multimodal data, there is still the question of how many control points are needed. We studied this by fitting B-spline models to our button dataset. To control against overfitting, we used Bayesian Information Criterion (BIC) [31, 53] for

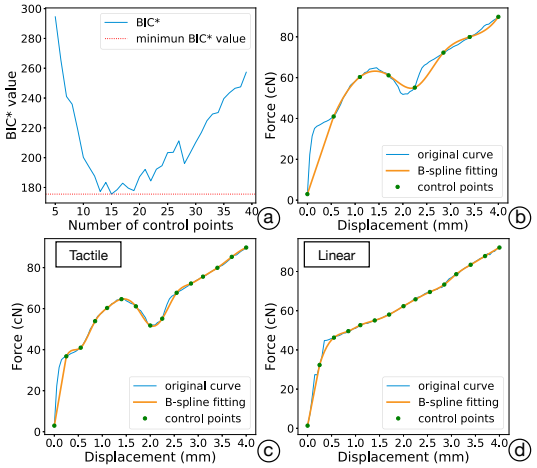


Figure 4. We use B-splines to obtain a lower-dimensional, editable FDVV model from the capture data. (a) We found that 15 control points is the ideal number of parameters to model six commodity buttons. (b) With fewer control points, the model underfits essential features of a button response. Panes c and d show example results for tactile and linear buttons (15 control points).

the fitness criteria. To reduce the number of parameters even further for feasibility for human editing, we added a custom penalty term: *Complexity Penalty, P*. This results in a modified BIC* function:

$$BIC^* = \ln(n)kP - 2\ln(\hat{L}) \quad (1)$$

Where n stands for the number of observations, k for the number of parameters (control points), and $\hat{L}$ for the maximized value of the likelihood function of the model. P is the added Complexity Penalty value, which is set to 2.5. Figure 4 (b, c, d) gives examples of fitting for a pressing segment of a certain velocity for Cherry MX Clear button.

Following this procedure, we found 15 B-spline control points was an ideal tradeoff for the six-button dataset with root mean square error of 0.14 cN; see Figure 4 (a). We also registered *travel range*, the *activation point*, and the *vibration point* in the resulting models, as examples shown in Figure 4 (c, d).

BUTTON SIMULATOR

Ours is the first physical simulator capable of the high-fidelity rendering of FDVV models. An overview is given in Figure 5. Our first design goal was to provide the high-frequency response and high-resolution rendering of forces and vibrations typical of buttons. The second was to enable full control from the software side.

Sensors and actuators: Figure 5 presents the four main components: (1) a linear force actuator (Moticont HVCM-025-022-003-01), (2) a linear position sensor (LVDT MHR 250, resolution: 0.05 mm), (3) a voice coil acting as a vibrotactile motor (Tectonic Teax13C02-8), and (4) a servo motor (Tower Pro Micro Servo, torque: 1.8 kg/cm). The force actuator, the

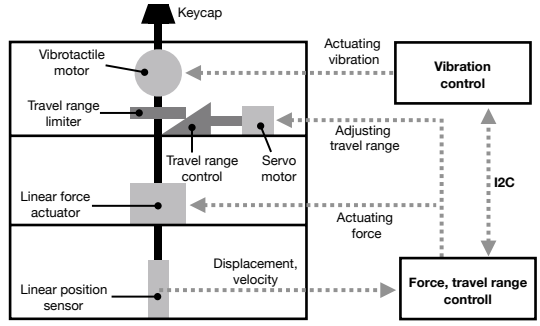


Figure 5. Physical simulator construction for haptic rendering of FDVV models. Our simulator includes a 1D sensor that tracks displacement, a 1D force actuator delivering various levels of forces, and a servo motor drives the travel-range-adjustment component. The components are controlled by a microprocessor. The other microprocessor controls a vibrotactile motor mounted near the keycap.

sensor, and the servo motor are controlled by an Adafruit ItsyBitsy M0 board, which serves as the main processor of the prototype. The vibrotactile voice coil is driven by an Arduino Uno board and wave shield (Adafruit Wave Shield for Arduino Kit). These two boards are connected via the I2C protocol. When adjustments to the *overall travel range* are required, the ItsyBitsy board sends a command to the servo motor to adjust the location of the *Travel Range Control*, which further alters the lowest reachable displacement of the *Travel Range Limiter* and produces varying travel. When vibrotactile feedback is required, this board communicates with the Arduino Uno via I2C and asks it to drive the vibrotactile motor (voice coil) to present pre-recorded wave files.

Microprocessor design: Before the simulation, the actuation signals (see "Iterative Compensation") are uploaded to the main microprocessor (Adafruit ItsyBitsy M0) and it automatically sets the button travel range. During a simulation, the linear sensor constantly sends the reading value to the microprocessor. A moving-average filter (window size = 25) is applied here for denoising the reading from the position sensor. After the microprocessor has processed the values sent, it calculates the current displacement of the button and estimates the user's pressing velocity. Then, it determines the corresponding Pulse-width modulation (PWM) signal and sends it to the linear force actuator. At the displacement where vibration starts, the microprocessor sends a command to the Arduino Uno for emitting the vibration. A high operating frequency is used (1 kHz) for the ItsyBitsy M0 board.

Spatial and temporal accuracy

We measured the spatial accuracy of the simulator via a probing device consisting of a linear actuator and a probe attached to a force gauge [37]. The probing velocity was 0.5 mm/s. We used this device to profile a 4 mm Cherry MX Clear button, checking the intended outputs of the simulator against the measurement results. Over multiple repetitions, we learned that the simulator can reproduce the force responses very accurately, with only 1.44 cN mean error (SD 1.68 cN). Time

spent on each displacement sensing and computing actuation command is about 0.3 ms. From the command of rendering force to force generated takes less than 1 ms. Latency from sending the vibration command to its actuation is about 7ms. We compensated this latency by emitting vibration 300 μm prior to the target starting point.

Simulation procedure

Prior to simulation, the actuation signals obtained from iterative compensation (see later) are uploaded to the microprocessor. Near the beginning of the press – i.e., at 0.5–1.0 mm of distance traveled – the microprocessor calculates the pressing velocity for the press. At least three timestamped samples are needed. From those samples, the simulator computes velocity and switches to the corresponding actuation specification. That specification is used to determine the resisting force and the vibration for the sensed level of displacement. To simulate the vibration, we followed an event-based approach wherein vibration recording is initiated at the right displacement [32], creating a snap-like sensation.

ITERATIVE COMPENSATION

A key objective in our work is to separate the model from the simulator. Any force actuator has its own transfer function in the play that must be canceled out if an FDVV curve is to be simulated correctly. To our knowledge, no prior work on button simulation has considered this issue, which may explain the lack of empirical evaluations of these simulators. To address the issue, we introduce an iterative compensation method shown in Figure 6:

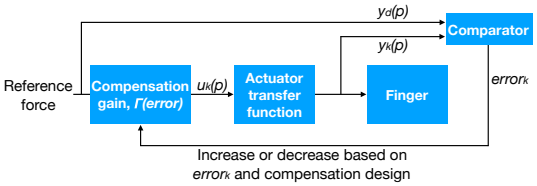


Figure 6. Iterative compensation finds a way to render an FDVV model on a given simulator plant.

The idea is to raise or lower the force-actuation signal amplitude of each displacement point until the desired resisting force is measured by the sensor against the keycap. Per-displacement and per-speed repetition can be applied until the desired FDVV signal is measured from the sensor. This iterative compensation process can be expressed as

$$u_{k+1}(p) = u_k(p) + \Gamma(error_k)(y_d(p) - y_k(p)), \quad p \in [1,n] \tag{2}$$

Here, $u_k(p)$ is the actuation signal of a given displacement point p in the current iteration, and $u_{k+1}(p)$ is the actuation in the next iteration signal of the same displacement point. $y_k(p)$ is the force detected from the sensor worn on the fingertip, and $y_d(p)$ is the desired target force at that given displacement point. $\Gamma(error)$ represents the proportion of adjustment of the actuation signal that must be applied, based on the error value in the current iteration ($error_k$). The error from the current

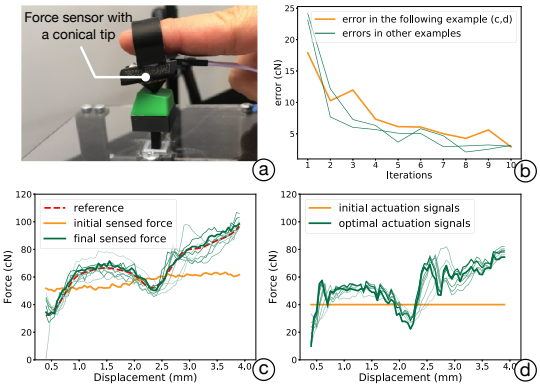


Figure 7. (a) During the iterative compensation, a force sensor is worn on the participant's fingertip which gathered force data and sent it to the controller. (b) Some instances of the evolution of error values during the process. The blue curve represents the errors of (c,d) which fell from 17.89 cN in the first iteration to 2.93 cN in the 10th. (c) An example keypress from which we can see that the sensor worn on the fingertip shows convergence with the reference after the compensation process is complete. (d) The actuation signals starting at a random force level and being gradually tuned. Note that we transform the actuation signals linearly into force level (cN) that can be measured in a steady machine-probing situation.

iteration is defined on a per-curve basis as follows:

$$error_k = \alpha \cdot \frac{\Sigma_{p=1}^{n} |y_d(p) - y_k(p)|}{n} + (1-\alpha) \cdot \max_{p \in [1,n]} |y_d(p) - y_k(p)| \tag{3}$$

In this definition, two terms make up the error. The first is the overall difference between the target FD curve and the measured curve. The second is the displacement error at which the largest error is observed. α is the weight applied between thees two, which we set to 0.7 based on experiences.

Procedure: We first upload the reference data (FDVV model) to the simulator. A participant is asked to press the button repetitively in line with the speed indicator presented by a GUI. Figure 7 illustrates the whole process. The force sensor is connected to the same microprocessor that runs the simulator. The sensor (response rate: 1 kHz) captures the force responses during a press by the fingertip. The value sensed is passed through a resistor–capacitor filter with a 498 Hz cutoff rate, to reduce within sensor noise. All the resisting force samples within a 50 μm interval are aggregated and averaged for this displacement point. In our experience, the procedure converges after only 8–12 presses; however, each button needs to be modeled at multiple velocities. With four velocities, there are 240 presses in total (4 velocities $\times$ 4 rounds $\times$ 15 presses) to compute its actuation signals. A typical example is presented in Figure 7 (b): $error_k$ decreases rapidly to below 3 cN within 10 presses.

Outputs: After the process is complete, the microprocessor records the actuation signals that resulted in the minimal error. For a given reference force curve, we ran the iterative compensation process four times, obtaining four series of ac-

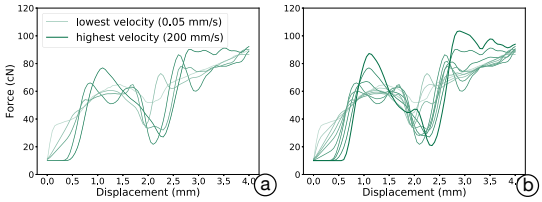


Figure 8. Final, linearly interpolated actuation signals: (a) Example actuation signals before linear interpolation, here shown for a Cherry MX Clear with various velocities, which are the same as in Button Capture. (b) The same data after linear interpolation.

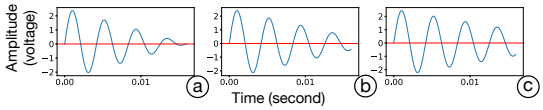


Figure 9. Example decaying sinusoidal-wave templates. They share the same frequency (239 Hz) and duration (16 ms) which are captured from Cherry MX Clear, and the amplitudes and shapes vary for tuning: (a) amplitude decreases from ± 2.43 to 0 Vol, (b) amplitude decreases from ± 2.43 to ± 0.3 Vol, and (c) amplitude decreases from ± 2.43 to ± 0.6 Vol.

tuation signals. We then averaged these at each displacement and finally applied a Gaussian filter ($\sigma = 1.2$ *mm*) to smooth the signals. After all the force-actuation signal curves were obtained (see Figure 8 (a)), we linearly interpolated the signal curves to form denser, more continuous curve sets that responded to more velocity changes (see Figure 8 (b)).

Optional human-in-the-loop vibration tuning

As described earlier, sometimes the vibration emitted at the snap point is weak and our sensor does not reliably pick it up accurately. Also, sometimes when vibration is measured as a soundwave, it may fail to reproduce the same sensation when reproduced using the vibration motor. In order to produce the desired snap sensation, the vibration needs to be accentuated. To this end, we devised a human-in-the-loop method for tuning the vibration response at the simulator side. To further render more realistic vibrotactile feedback, future work should consider more sophisticated modeling and rendering techniques [18, 48, 51, 68].

Procedure: We obtained 3 features from vibration measurements: (1) vibration onset, (2) duration and (3) frequency. Afterward, an algorithm generated several vibration templates that match the recorded vibration and duration. The generated templates are decaying sinusoidal waves with various frequencies and accentuated amplitudes. The generative method we follow is by Park *et al.* [52]. These sound-wave templates were uploaded to the Arduino Uno, which simulates them, using the vibrotactile motor (see Figure 5). Some templates are shown in Figure 9. Finally, as part of our human-in-the-loop tuning process, we asked a human observer to press the simulated button at the pace shown in the animation (see above). The user rated each button-design–vibration combination. We repeated this process for all velocities. The best-rated vibration sets were selected as the final actuation signals for that button.

A USER STUDY: PERCEIVED REALISM

We assessed the perceived realism of the rendered buttons in a controlled study. We adopted the idea of ABX test as used in psychophysics for comparing two sensory stimulus options for identifying a target [10, 29, 42, 45]. A participant tries a real reference button (X) and is then asked to press two simulated buttons (A, B) and decide which offers a more realistic rendering of it. The A, B buttons are rendered via the same physical simulator, and the user can try out the three buttons as many times as desired. We compared our FDVV models against speed-agnostic FD models because it represents the prevailing understanding of button tactility represented in academic literature, hobbyist groups, and manufacturer datasheets.

Method

Participants: We recruited 12 participants (6 females) from a local university, of ages 21–41 (mean 29.75). All of them reported typing and other button-pressing experience. They were rewarded with a movie ticket (valued at 14 euros) for the 60-minute study.

Task and apparatus: The study compared six physical buttons listed previously in "Button Capture". They differ in characteristics, but all are realistic. All the buttons were captured and transformed into (1) a single-FD model and (2) FDVV models. Actuation signals were computed as described in the previous section.

To prevent users' haptic judgments from being biased by their vision, the simulator was placed inside a black box with a hole, into which the user reached to press. The target buttons and the simulator were at the same height. The participants were asked to wear a headset playing white noise and earmuffs, to isolate hearing during the study. A graphical interface showed which of the two buttons (labeled A and B) was currently active. The information displayed was the name of the target button (one of the six), the simulated button's label (A or B), and the current trial number. Double-blind administration was employed: neither the experimenter nor the participant knew which button (A vs. B) used FDVV and which used FD.

Procedure and experiment design: The participants were told about the simulator and the purposes of the study. They were asked to explore the real buttons once, with different pressing velocities. We did not repeat this instruction during the study proper, though; we let them decide what was natural for them.

Again, each round featured a *reference button* and *two simulated buttons*. The interface identified the reference button and the label of the currently active button (see Figure 10). The participants were told that there are two simulated buttons in each round, denoted as *button A* and *button B*. In each round, participants were instructed to press the reference button and to feel it. After that, they were asked to try the alternative simulations, labeled A and B. They could switch among the three buttons as many times as they wished. When ready to make their judgment, they were asked to indicate which button had more realism (A, B, or equal) and to rate the perceived realism of A and B separately, on a seven-point Likert scale. After the study, an interview was conducted. We gave a three-minute

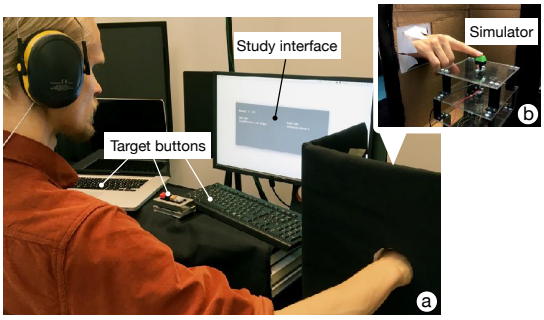


Figure 10. We assessed the perceived realism of the buttons in an ABX task. Firstly, (a) a participant was shown a (real) reference button (X). Then (b) the participant could try one simulated button (either FD- or FDVV-based) on the simulator at a time. The display showed the label for the active button (A or B). Afterward, the participant indicated which of the two (A or B) was more realistically rendering X.

break after every 20 minutes of button presses, to minimize fatigue. There were six rounds for each of the six buttons, making 36 trials in total. The trial order of the six button designs was counter-balanced by Latin square. The assignment of the FDVV and FD models to the labels, A and B, was randomized at each trial.

Results

The results support the FDVV approach. It was associated with higher perceived realism for all the simulated buttons. The participants chose the FDVV model as more realistic 77.31% of the time. An overview is shown in Figure 11. We examined the ratings further by using Wilcoxon Signed Ranks Tests. The analysis showed that there are statistically significant differences for each target button between the FDVV and single-FD models:

1. For the *Clear* button, the median FDVV model ranks (mdn 5.16, mean 4.9, STD 0.72) were significantly higher than the median single-FD ones (mdn 3.83, mean 3.72, STD 0.92), with $Z = -3.06$, $P = 0.002$.
2. For the *Brown* button, the median FDVV model ranks (mdn 4.92, mean 4.64, STD 1.02) were significantly higher than the median single-FD ones (mdn 3.83, mean 3.86, STD 0.87), with $Z = -2.748$, $P = 0.006$.
3. For the *Black* button, the median FDVV model ranks (mdn 5.33, mean 5.4, STD 0.53) were significantly higher than the median single-FD ones (mdn 3.83, mean 4.25, STD 0.72), with $Z = -2.94$, $P = 0.003$.
4. For the *Red* button, the median FDVV model ranks (mdn 5.33, mean 5.14, STD 0.9) were significantly higher than the median single-FD ones (mdn 4.67, mean 4.58, STD 0.54), with $Z = -2.158$, $P = 0.031$.
5. For the *HP keyboard* (spacebar), the median FDVV model ranks (mdn 5.33, mean 5.03, STD 0.58) were significantly higher than the median single-FD ranks (mdn 3.83, mean 4.0, STD 0.85), with $Z = -2.987$, $P = 0.003$.
6. For the *MacBook Pro keyboard* (spacebar), the median FDVV model ranks (mdn 5.42, mean 5.57, STD 0.57) were significantly higher than the median single-FD ranks (mdn 4.67, mean 3.67, STD 0.9), with $Z = -3.06$, $P = 0.002$.

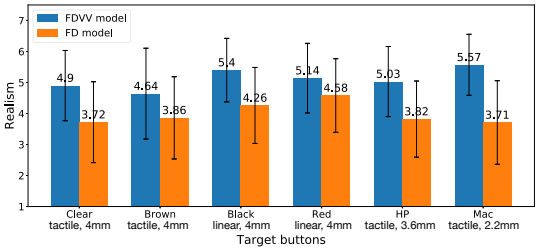


Figure 11. Users in the ABX identity-matching study rated FDVV-based simulations as more realistic than FD-based simulations. Statistically significant differences were found for all the target buttons. The error bar in the figure is 1 STD.

From Figure 11, we see that the smallest difference between the FDVV and FD simulation was for the Red button. This is a linear button, and the FD model offers a reasonable simulation. One participant stated, *"The red one has a smooth (linear) feeling, and it's lighter than the other buttons. I feel two models with a similarly smooth and light feeling, so it's hard to tell the differences."*

APPLICATIONS

We show three applications exploiting the approach.

Human-in-the-loop button optimization

Firstly, we demonstrate the optimization of a button for an interactive task. We look at a temporal pointing task with no visual feedback [34]. It resembles games where a response must be elicited at just the right time (for instance, to catch an enemy). Our goal is to optimize the button's FDV design and also its activation point. Velocity-dependent properties (the last V in FDVV) were excluded due to assuming a person presses a button at a similar speed all the time. The optimization we used was Bayesian optimization (BO), which is favorable for conditions wherein evaluations are noisy and expensive [61]. The objective in BO is to minimize a user's mean asynchrony [34, 56], or the mean difference in time between the target and the user's elicited response. Figure 14 depicts some example FDV models generated by the optimizer. To keep the study below one hour in total length, we limited the BO's task to three control points and two other button parameters. We mapped this to the force-actuation signals, which the BO then manipulated so that once a new design is sent to the simulator, users can instantly try it out without iterative learning.

Method: We recruited 10 participants (4 females) from a local university, of ages 20–40 (mean 26). The study had two phases, *training* and *validation*. In the training phase, they were asked to press the button when the LED strip showed a bullet to have reached the center of the target zone. Two levels of task difficulty (*easy* = 100 pixels/second; *difficult* = 150 pixels/second) were used. For each level, 27 trials were collected and used to compute the mean asynchrony score.

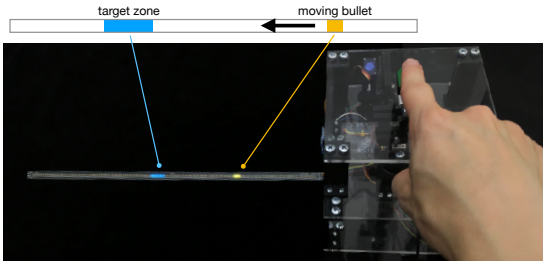


Figure 12. Button optimization was evaluated in a temporal pointing task. Users were asked to hit a temporal target by pressing a button at the right time, as a moving bullet reaches the target zone. After the optimal design was learned, the actuation signals were translated back into an FDV model via the capturing process.

The whole process took about 60 minutes. Short breaks were given after every 15 minutes of presses.

Our BO implementation was based on Python's GPyOpt library[1]. In each iteration, it changed the parameter values of the button model. We had three control points in the FDV and three other parameters: $x1, x2, x3$ (displacements of three control points) and $y1, y2, y3$ (actuation signals of those control points). The ranges of these six parameters were $x1 \in [0,1), x2 \in [1,3), x3 \in [3,6.2)$ and $y1, y2, y3 \in [20,180]$. The other two additional parameters were activation point, p_a, and vibration point, p_v. The ranges of these two parameters were $p_a, p_v \in [0.5,5.5]$. An additional microprocessor, an Arduino Uno, was set to drive an LED strip (Adafruit DotStar) that displayed an array of LED lights with the bullet animation. This microprocessor was connected to the button simulator by software serial port. When the button passed the activation point, the microprocessor of the simulator sent a triggering signal to the LED strip, which would then calculate the temporal error of this press. After 20 presses, we calculated the mean asynchrony and sent the information back to the simulator. Then, the optimizer created a new design of button for the next iteration, based on the data collected, and triggered the simulator to reconfigure itself accordingly.

In the testing phase, a week after the training phase, the optimized button was compared to other non-optimized buttons: four 4 mm mechanical ones (Cherry MX Clear, Brown, Black, and Red) and a random button design (all parameters are randomly given within the defined range). Two difficulty levels are assigned in a counter-balanced order. Each button appeared twice per difficulty level, in counter-balanced order, and the participant needed to press the button 20 times in every trial. In total, we collected 240 observations of mean asynchrony (10 participants $\times$ 2 difficulty levels $\times$ 2 rounds $\times$ 6 buttons).

Results: In Figure 13, we present some instances of optimal button designs. For the difficulty level **Easy**, the resulting mean asynchronies were 65.8 ms (STD 6.15), 83.6 (STD 7.65), 88.04 (STD 7.82), 81.65 (STD 6.19), 84.28 (STD 6.75), and 100.78 (STD 6.89), for the optimal, clear, brown, black, red, and random button, respectively. For the **Hard** difficulty

[1] See http://sheffieldml.github.io/GPyOpt/

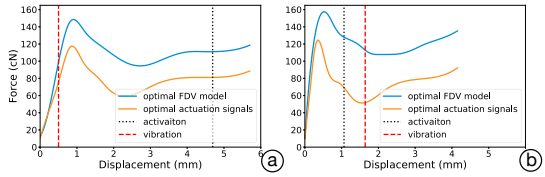


Figure 13. An example outcome of Bayesian optimization for a user under the (a) "Easy" and (b) "Hard" task condition.

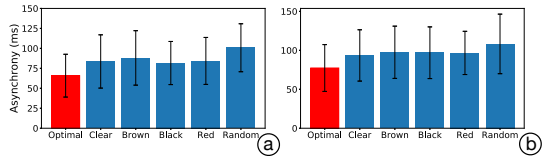


Figure 14. Button optimization significantly decreased mean asynchrony in temporal pointing for both (a) easy and (b) hard tasks. The error bar in the figure is 1 STD.

level, the resulting mean asynchrony values are 77.3 ms (STD 6.89), 93.43 (STD 7.56), 97.48 (STD 7.69), 96.9 (STD 7.61), 96.65 (STD 6.36), and 108.22 (STD 8.77), for the optimal, clear, brown, black, red, and random button, respectively. A two-way repeated measures ANOVA was conducted. The main effect of buttons on mean asynchronies is significant, $F(5,95) = 10.724$, $p < 0.001$. The *post-hoc* tests with Bonferroni correction confirmed the optimal button design as indeed outperforming the rest ($p < 0.05$).

Interactive button design

We implement an *interactive button-editing tool* that lets engineers freely create and edit button designs (Figure 15). Using the tool (implemented on macOS using Swift), designers can manipulate force levels through 15 draggable control points to create a single-FD curve. Travel range, activation point, and vibration point can be edited textually. Next, the model is converted to high-dimensional force actuation signals (with B-spline fitting and iterative compensation). Finally, these signals are used to simulate the button. Several button designs implemented by the interactive button-editing tools are presented below, under "Innovative button designs." While the tool allows for editing of one FDV model at a time, our backend (Python) also allows for more advanced editing, such as assigning a specific FD curve segment to be pressed or released, and merging multiple FD curves into an FDVV model.

Prototyping Innovative Buttons

Our setup enables prototyping innovative and extraordinary buttons that go beyond commodity designs. We give five examples here. The first two are instances of FDVV models, while the remaining are buttons that can be run on the simulator but can not be expressed as FDVV models.

1. A fast tapping button: While humans can only reach about four presses per second in tapping tasks [56], we can increase this capacity via a novel button design. The principle is this:

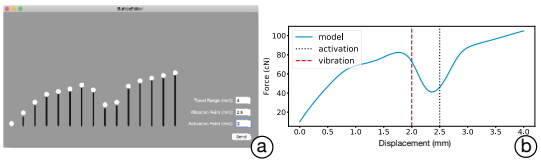


Figure 15. The button-editing tool: (a) The design tool allows the designer to freely create and edit low-parametric FDV models. (B) An edited model can be processed and then simulated via the simulator.

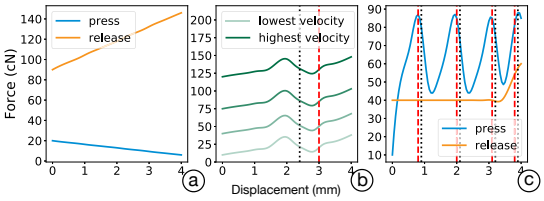


Figure 16. Three innovative button designs realized with our setup: (a) a button for very fast tapping, (b) a non-Newtonian-fluid button, and (c) a multi-level button.

once a press is detected, the button drops to the bottom and returns automatically. This can be especially useful for content requiring high-frequency rhythmic tapping, such as music games (see Figure 16 (a)). One of the authors reached 8 presses per second with this button.

2. A non-Newtonian-fluid button: In non-Newtonian fluids, viscosity changes under force: the medium becomes either more liquid or more solid. We can design a button that adjusts its stiffness following pressing velocity; see Figure 16 (b). As a result, the button is softer when being pressed gently, but the resisting force increases during fast pressing. This design can be used to prevent accidental touches.

3. A multi-level button: We can extend the input modality of a button by giving multi-level haptic feedback. For example, a typical 4 mm button can be divided into several depth ranges. Through the provision of distinguishable haptic signals, a user can effectively activate different functions of the button by pressing down to different depths; see Figure 16 (c). This can be useful for easier use of single-button devices (e.g., in tuning the luminance of a lamp).

4. Vibration cues: We can deliver rich temporal information through continuous vibrotactile cues while the press is at the bottom. This interaction can enhance the effectiveness and efficiency of dwell-press applications [36]. For instance, when the shutter button of a camera is pressed and the camera is continuously shooting, vibration ticks can help the user easily count the number of shots via humans' haptic sense.

5. A dynamically returning button: In certain situations, it might be desirable to avoid fast repetition with a given button. In the example of fighting games or shooting games, many attacks involve a cooldown time – i.e., a minimum duration before the next use of the relevant ability. Our simulator can easily render buttons with just such dynamic returning time, as demanded by the game content.

CONCLUSION

We have shown that the FDVV approach proposed as an extension to the dominant FD model increases the scope and perceived realism of button simulations. While this extension was motivated by prior literature, several engineering problems were solved to capture and simulate FDVV models. The added complexity notwithstanding, the core model is understandable and offers practitioners a workable starting point. We have demonstrated the benefit of model–simulator separation via three applications: human-in-the-loop button optimization, interactive button design, and examples of innovative buttons that would be tricky to realize without this approach.

While the FDVV approach permits greater realism, and opens many new practical possibilities, our results point also to clear opportunities for further improving realism. Firstly, the iterative compensation method can be enhanced, to better capture the *dynamic* effects of damping on the actuation signals. Our approach was to try to cancel the effect of the transfer function, but future research could consider learning a data-driven model of the black box through machine learning. This should be coupled with a clever controller design. Secondly, structural vibration can be modeled better. We opted to measure the snap-like vibration as a sound wave, but sometimes this was inadequate to reproduce the felt sensation of the tactile bump. To enhance the realism of the snap point, researchers could consider applying better measurement, modeling and controlling methods for handling the vibration [48,51]. This would eliminate the human-dependent part of the vibration modeling. While these changes would increase realism, perfect button simulation entails simulating the texture and shape of the keycap, the sound emitted, etc. Finally, once a button has been designed and tested, a suitable electromechanical design should be fabricated. This presents an interesting challenge for future work aimed at bridging the gap between buttons *in vitro* and buttons *in vivo*.

OPEN SCIENCE

The materials and data in this paper are released on our project page at `http://userinterfaces.aalto.fi/button-design`. The materials include 3D models, circuit design, component specifications, construction details of the simulator, and the programs for controllers. The cost for the simulator construction is about $550. From capturing a button to simulation via our pipeline takes about an hour (5 minutes for capturing, 30 minutes for data preprocessing and FDVV modeling, and 30 minutes for iterative compensation and simulation). All the data for button measurements and experiments are released too. The materials also complement Figure 1 and Figure 8 with the graphs of other buttons.

ACKNOWLEDGEMENT

This work has been funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 637991) and by Korea Creative Content Agency (grant agreement No R2019020010).

REFERENCES

[1] Victoria. E. Abraira and David. D. Ginty. 2013. The Sensory Neurons of Touch. *Neuron* 79, 4 (2013), 618 – 639. DOI: http://dx.doi.org/10.1016/j.neuron.2013.07.051

[2] Kenichi Akagi. 1992. A Computer Keyboard Key Feel Study in Performance and Preference. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 36, 5 (1992), 523–527. DOI: http://dx.doi.org/10.1177/154193129203600511

[3] B. Allotta, V. Colla, and G. Bioli. 1999. A mechatronic device for simulating push-buttons and knobs. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, Vol. 1. 636–642 vol.1. DOI:http://dx.doi.org/10.1109/MMCS.1999.779274

[4] M. Angerilli, A. Frisoli, F. Salsedo, S. Marcheschi, and M. Bergamasco. 2001. Haptic simulation of an automotive manual gearshift. In *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)*. 170–175. DOI: http://dx.doi.org/10.1109/ROMAN.2001.981897

[5] Jonathan Becedas, Gabriela Mamani, Vicente Feliu, and Hebertt Sira-Ramírez. 2009. *Estimation of Mass-Spring-Dumper Systems*. Springer Netherlands, Dordrecht, 411–422. DOI: http://dx.doi.org/10.1007/978-1-4020-8919-0_28

[6] Jonathan Becedas, Gabriela Mamani, Vicente Feliu-Batlle, and Hebertt Sira-Ramírez. 2007. Algebraic Identification Method for Mass-Spring-Damper System.

[7] Ingvars Birznieks, Per Jenmalm, Antony W. Goodwin, and Roland S. Johansson. 2001. Encoding of Direction of Fingertip Forces by Human Tactile Afferents. *Journal of Neuroscience* 21, 20 (2001), 8222–8237. DOI: http://dx.doi.org/10.1523/JNEUROSCI.21-20-08222.2001

[8] Stephen P. Boyd and Craig H. Barratt. 1991. *Linear Controller Design: Limits of Performance*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[9] Andy Clark. 2013. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences* 36, 3 (2013), 181–204. DOI:http://dx.doi.org/10.1017/S0140525X12000477

[10] David Clark. 1982. High-Resolution Subjective Testing Using a Double-Blind Comparator. *J. Audio Eng. Soc* 30, 5 (1982), 330–338. http://www.aes.org/e-lib/browse.cfm?elib=3839

[11] M. B. Colton and J. M. Hollerbach. 2005. Identification of nonlinear passive devices for haptic simulations. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. 363–368. DOI:http://dx.doi.org/10.1109/WHC.2005.77

[12] Matthew J. C. Crump and Gordon D. Logan. 2010. Warning: This keyboard will deconstruct— The role of the keyboard in skilled typewriting. *Psychonomic Bulletin & Review* 17, 3 (01 Jun 2010), 394–399. DOI: http://dx.doi.org/10.3758/PBR.17.3.394

[13] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. 2010. Tactile Sensing—From Humans to Humanoids. *IEEE Transactions on Robotics* 26, 1 (Feb 2010), 1–20. DOI: http://dx.doi.org/10.1109/TRO.2009.2033627

[14] C. Doerrer and R. Werthschuetzky. 2002. Simulating Push-Buttons Using a Haptic Display: Requirements on Force Resolution and Force-Displacement Curve. (2002).

[15] K. FUJITA. 2001. A New Softness Display Interface by Dynamic Fingertip Contact Area Control. *5th World Multiconference on Systemics, Cybernetics and Informatics, 2001* (2001), 78–82. https://ci.nii.ac.jp/naid/10031028472/en/

[16] Nico Galoppo, Serhat Tekin, Miguel A. Otaduy, Markus Gross, and Ming C. Lin. 2007. Interactive Haptic Rendering of High-Resolution Deformable Objects. In *Virtual Reality*, Randall Shumaker (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 215–223.

[17] T. Hachisu, G. Cirio, M. Marchal, A. Lécuyer, and H. Kajimoto. 2011. Pseudo-haptic feedback augmented with visual and tactile vibrations. In *2011 IEEE International Symposium on VR Innovation*. 327–328. DOI:http://dx.doi.org/10.1109/ISVRI.2011.5759662

[18] T. Hachisu and H. Kajimoto. 2017. Vibration Feedback Latency Affects Material Perception During Rod Tapping Interactions. *IEEE Transactions on Haptics* 10, 2 (April 2017), 288–295. DOI: http://dx.doi.org/10.1109/TOH.2016.2628900

[19] Vincent Hayward and Brian Armstrong. 2000. A new computational model of friction applied to haptic rendering. In *Experimental Robotics VI*. Springer London, London, 403–412.

[20] Robert D. Howe and Mark R. Cutkosky. 1989. Sensing skin acceleration for slip and texture perception. *Proceedings, 1989 International Conference on Robotics and Automation* (1989), 145–150 vol.1.

[21] Yoshiaki Ikeda and Kinya Fujita. 2004. Display of Soft Elastic Object by Simultaneous Control of Fingertip Contact Area and Reaction Force. *Transactions of the Virtual Reality Society of Japan* 9, 2 (2004), 187–194. DOI:http://dx.doi.org/10.18974/tvrsj.9.2_187

[22] Devin L Jindrich, Aruna D Balakrishnan, and Jack T Dennerlein. 2004. Effects of keyswitch design and finger posture on finger joint kinematics and dynamics during tapping on computer keyswitches. *Clinical Biomechanics* 19, 6 (2004), 600 – 608. DOI: http://dx.doi.org/https://doi.org/10.1016/j.clinbiomech.2004.03.003

[23] Hiroyuki Kajimoto, Naoki Kawakami, Taro Maeda, and Susumu Tachi. 2001. Electro-Tactile Display with Force Feedback.

[24] Krueger L. (Ed.) Krueger L. (Ed.) Katz, D. 1989. *The World of Touch*. New York: Psychology Press. DOI:http://dx.doi.org/https://doi.org/10.4324/9780203771976

[25] Johan Kildal. 2010. 3D-press: Haptic Illusion of Compliance when Pressing on a Rigid Surface. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI '10)*. ACM, New York, NY, USA, Article 21, 8 pages. DOI: http://dx.doi.org/10.1145/1891903.1891931

[26] Jeong Ho Kim, Lovenoor Aulck, Michael C. Bartha, Christy A. Harper, and Peter W. Johnson. 2014a. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied Ergonomics* 45, 6 (2014), 1406 – 1413. DOI:http://dx.doi.org/https://doi.org/10.1016/j.apergo.2014.04.001

[27] Jeong Ho Kim, Lovenoor S. Aulck, Michael C. Bartha, Christy A. Harper, and Peter W. Johnson. 2014b. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied ergonomics* 45 6 (2014), 1406–13.

[28] Sunjun Kim, Byungjoo Lee, and Antti Oulasvirta. 2018. Impact Activation Improves Rapid Button Pressing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 571, 8 pages. DOI: http://dx.doi.org/10.1145/3173574.3174145

[29] Sunjun Kim and Geehyuk Lee. 2013. Haptic Feedback Design for a Virtual Button Along Force-displacement Curves. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 91–96. DOI: http://dx.doi.org/10.1145/2501988.2502041

[30] Sunjun Kim, Jeongmin Son, Geehyuk Lee, Hwan Kim, and Woohun Lee. 2013. TapBoard: Making a Touch Screen Keyboard More Touchable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 553–562. DOI: http://dx.doi.org/10.1145/2470654.2470733

[31] Sadanori Konishi and Genshiro Kitagawa. 2007. *Information Criteria and Statistical Modeling* (1st ed.). Springer Publishing Company, Incorporated.

[32] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer. 2006. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics* 12, 2 (March 2006), 219–230. DOI: http://dx.doi.org/10.1109/TVCG.2006.32

[33] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. Moving Target Selection: A Cue Integration Model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 230, 12 pages. DOI:http://dx.doi.org/10.1145/3173574.3173804

[34] Byungjoo Lee and Antti Oulasvirta. 2016. Modelling Error Rates in Temporal Pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1857–1868. DOI: http://dx.doi.org/10.1145/2858036.2858143

[35] James R Lewis, Kathleen M Potosnak, and Regis L Magyar. 1997. Keys and keyboards. In *Handbook of human-computer interaction*. Elsevier, 1285–1315.

[36] Yi-Chi Liao, Yen-Chiu Chen, Liwei Chan, and Bing-Yu Chen. 2017. Dwell+: Multi-Level Mode Selection Using Vibrotactile Cues. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 5–16. DOI:http://dx.doi.org/10.1145/3126594.3126627

[37] Yi-Chi Liao, Sunjun Kim, and Antti Oulasvirta. 2018. One Button to Rule Them All: Rendering Arbitrary Force-Displacement Curves. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings (UIST '18 Adjunct)*. ACM, New York, NY, USA, 111–113. DOI: http://dx.doi.org/10.1145/3266037.3266118

[38] Karon E. MacLean and William K. Durfee. 1995. Apparatus to study the emulation of haptic feedback. In *ASME Dynamic Systems and Control Division*, Vol. 57-2. ASME, 615–621.

[39] Karon E. MacLean. 1996. The"Haptic Camera": a technique for characterizing and playing back haptic properties of real envi.

[40] Richard W. Marklin and Mark L. Nagurka. 2000. Measurement of Stiffness and Damping Characateristics of Computer Keyboard Keys. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 44, 6 (2000), 678–681. DOI: http://dx.doi.org/10.1177/154193120004400637

[41] Thomas H. Massie and J. K. Salisbury. 1994. The PHANToM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Dynamic Systems and Control Division*. 295–301.

[42] Carr B. Carr B. Meilgaard, M. 2007. *Sensory Evaluation Techniques* (4th edition ed.). Boca Raton: CRC Press. DOI:http://dx.doi.org/https://doi.org/10.1201/b16452

[43] Leonard Meirovitch. 1997. *Principles and techniques of vibrations*. Vol. 1. Prentice Hall New Jersey.

[44] G. Moy, C. Wagner, and R. S. Fearing. 2000. A compliant tactile display for teletaction. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Vol. 4. 3409–3415 vol.4. DOI: http://dx.doi.org/10.1109/ROBOT.2000.845247

[45] W. A. Munson and Mark B. Gardner. 1950. Standardizing Auditory Tests. *The Journal of the Acoustical Society of America* 22, 5 (1950), 675–675. DOI:http://dx.doi.org/10.1121/1.1917190

[46] Ken Nakagaki, Daniel Fitzgerald, Zhiyao (John) Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. 2019. inFORCE: Bi-directional 'Force' Shape Display for Haptic Interaction. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. ACM, New York, NY, USA, 615–623. DOI: http://dx.doi.org/10.1145/3294109.3295621

[47] Daichi Ogawa, Vibol Yem, Taku Hachisu, and Hiroyuki Kajimoto. 2015. Multiple Texture Button by Adding Haptic Vibration and Displacement Sensing to the Physical Button. In *SIGGRAPH Asia 2015 Haptic Media And Contents Design (SA '15)*. ACM, New York, NY, USA, Article 12, 2 pages. DOI: http://dx.doi.org/10.1145/2818384.2818394

[48] A. M. Okamura, M. R. Cutkosky, and J. T. Dennerlein. 2001. Reality-based models for vibration feedback in virtual environments. *IEEE/ASME Transactions on Mechatronics* 6, 3 (Sep. 2001), 245–252. DOI: http://dx.doi.org/10.1109/3516.951362

[49] Antti Oulasvirta, Sunjun Kim, and Byungjoo Lee. 2018. Neuromechanics of a Button Press. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 508, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3174082

[50] Dinesh K. Pai, Austin Rothwell, Pearson Wyder-Hodge, Alistair Wick, Ye Fan, Egor Larionov, Darcy Harrison, Debanga Raj Neog, and Cole Shing. 2018. The Human Touch: Measuring Contact with Real Human Soft Tissues. *ACM Trans. Graph.* 37, 4, Article 58 (July 2018), 12 pages. DOI: http://dx.doi.org/10.1145/3197517.3201296

[51] Gunhyuk Park and Seungmoon Choi. 2018. PhysVib: Physically Plausible Vibrotactile Feedback Library to Collisions on a Mobile Device. In *Haptic Interaction*, Shoichi Hasegawa, Masashi Konyo, Ki-Uk Kyung, Takuya Nojima, and Hiroyuki Kajimoto (Eds.). Springer Singapore, Singapore, 409–413.

[52] Gunhyuk Park, Seungmoon Choi, Kyunghun Hwang, Sunwook Kim, Jaecheon Sa, and Moonchae Joung. 2011. Tactile Effect Design and Evaluation for Virtual Buttons on a Mobile Device Touchscreen. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. Association for Computing Machinery, New York, NY, USA, 11–20. DOI: http://dx.doi.org/10.1145/2037373.2037376

[53] David Posada and Thomas R. Buckley. 2004. Model Selection and Model Averaging in Phylogenetics: Advantages of Akaike Information Criterion and Bayesian Approaches Over Likelihood Ratio Tests. *Systematic Biology* 53, 5 (10 2004), 793–808. DOI: http://dx.doi.org/10.1080/10635150490522304

[54] Robert G. Radwin and One-Jang Jeng. 1997. Activation Force and Travel effects on Overexertion in Repetitive Key Tapping. *Human Factors* 39, 1 (1997), 130–140. DOI:http://dx.doi.org/10.1518/001872097778940605 PMID: 9302885.

[55] David Rempel, Elaine Serina, Edward Klinenberg, Bernard J. Martin, Thomas J. Armstrong, James A. Foulke, and Sivakumaran Natarajan. 1997. The effect of keyboard keyswitch make force on applied force and finger flexor muscle activity. *Ergonomics* 40, 8 (1997), 800–808. DOI: http://dx.doi.org/10.1080/001401397187793 PMID: 9336104.

[56] Bruno H. Repp. 2005. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review* 12, 6 (01 Dec 2005), 969–992. DOI: http://dx.doi.org/10.3758/BF03206433

[57] Yann Roudaut, Aurélie Lonigro, Bertrand Coste, Jizhe Hao, Patrick Delmas, and Marcel Crest. 2012. Touch sense. *Channels* 6, 4 (2012), 234–245. DOI: http://dx.doi.org/10.4161/chan.22213 PMID: 23146937.

[58] J. K. Salisbury and M. A. Srinivasan. 1997. Phantom-based haptic interaction with virtual objects. *IEEE Computer Graphics and Applications* 17, 5 (Sep. 1997), 6–10. DOI: http://dx.doi.org/10.1109/MCG.1997.1626171

[59] Robert A. Scheidt, Michael A. Conditt, Emanuele L. Secco, and Ferdinando A. Mussa-Ivaldi. 2005. Interaction of Visual and Proprioceptive Feedback During Adaptation of Human Reaching Movements. *Journal of Neurophysiology* 93, 6 (2005), 3200–3213. DOI:http://dx.doi.org/10.1152/jn.00947.2004 PMID: 15659526.

[60] E. P. Scilingo, M. Bianchi, G. Grioli, and A. Bicchi. 2010. Rendering Softness: Integration of Kinesthetic and Cutaneous Information in a Haptic Device. *IEEE Transactions on Haptics* 3, 2 (April 2010), 109–118. DOI:http://dx.doi.org/10.1109/TOH.2010.2

[61] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 1 (2015), 148–175.

[62] Aiguo Song, Jia Liu, and Juan Wu. 2008. Softness Haptic Display Device for Human- Computer Interaction. In *Human Computer Interaction*, Ioannis Pavlidis (Ed.). IntechOpen, Rijeka, Chapter 16. DOI: http://dx.doi.org/10.5772/6299

[63] Paul Strohmeier and Kasper Hornbæk. 2017. Generating Haptic Textures with a Vibrotactile Actuator. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4994–5005. DOI: http://dx.doi.org/10.1145/3025453.3025812

[64] Seiya Takei, Ryo Watanabe, Ryuta Okazaki, Taku Hachisu, and Hiroyuki Kajimoto. 2015. *Presentation of Softness Using Film-Type Electro-Tactile Display and Pressure Distribution Measurement*. Springer Japan, Tokyo, 91–96. DOI: http://dx.doi.org/10.1007/978-4-431-55690-9_17

[65] A. B. Vallbo and Roland Johansson. 1999. Properties of cutaneous mechanoreceptors in the human hand-related to touch sensation.

[66] Wenping Wang, Helmut Pottmann, and Yang Liu. 2006. Fitting B-spline Curves to Point Clouds by Curvature-based Squared Distance Minimization. *ACM Trans. Graph.* 25, 2 (April 2006), 214–238. DOI: http://dx.doi.org/10.1145/1138450.1138453

[67] D. W. Weir, M. Peshkin, J. E. Colgate, P. Buttolo, J. Rankin, and M. Johnston. 2004. The haptic profile: capturing the feel of switches. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.* 186–193. DOI: http://dx.doi.org/10.1109/HAPTIC.2004.1287195

[68] Parris S. Wellman and Robert D. Howe. 1995. Towards Realistic Vibrotactile Display In Virtual Environments.

[69] K.J. Åström and T. Hägglund. 2001. The future of PID control. *Control Engineering Practice* 9, 11 (2001), 1163 – 1175. DOI:http://dx.doi.org/https://doi.org/10.1016/S0967-0661(01)00062-4

Publication V

Yi-Chi Liao, Kashyap Todi, Aditya Acharya, Antti Keurulainen, Andrew Howes, Antti Oulasvirta. Rediscovering Affordance: A Reinforcement Learning Perspective. In *2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, USA, April 2022.

Rediscovering Affordance: A Reinforcement Learning Perspective

Yi-Chi Liao
yi-chi.liao@aalto.fi
Aalto University
Finland

Kashyap Todi
kashyap.todi@gmail.com
Aalto University
Finland

Aditya Acharya
a.acharya.1@bham.ac.uk
Aalto University
Finland
University of Birmingham
United Kingdom

Antti Keurulainen
antti.keurulainen@aalto.fi
Aalto University
Finland

Andrew Howes
a.howes@bham.ac.uk
Aalto University
Finland
University of Birmingham
United Kingdom

Antti Oulasvirta
antti.oulasvirta@aalto.fi
Aalto University
Finland

ABSTRACT

Affordance refers to the perception of possible actions allowed by an object. Despite its relevance to human–computer interaction, no existing theory explains the mechanisms that underpin affordance-formation; that is, *how* affordances are discovered and adapted via interaction. We propose an integrative theory of affordance-formation based on the theory of reinforcement learning in cognitive sciences. The key assumption is that users learn to associate promising motor actions to percepts via experience when reinforcement signals (success/failure) are present. They also learn to categorize actions (e.g., "rotating" a dial), giving them the ability to name and reason about affordance. Upon encountering novel widgets, their ability to generalize these actions determines their ability to perceive affordances. We implement this theory in a virtual robot model, which demonstrates human-like adaptation of affordance in interactive widgets tasks. While its predictions align with trends in human data, humans are able to adapt affordances faster, suggesting the existence of additional mechanisms.

CCS CONCEPTS

• **Human-centered computing → HCI theory, concepts and models**.

KEYWORDS

Affordance; Reinforcement Learning; Perception; Action; Modeling; Theory; Robotics; Motion Planning; Adaptation; Interaction; Machine Learning; Design

CHI 2022, April 30–May 06, New Orleans, Louisiana, USA.

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501992

ACM Reference Format:
Yi-Chi Liao, Kashyap Todi, Aditya Acharya, Antti Keurulainen, Andrew Howes, and Antti Oulasvirta. 2022. Rediscovering Affordance: A Reinforcement Learning Perspective. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3491102.3501992

1 INTRODUCTION

Imagine seeing a widget for the first time. How do you know how to interact with it? We often "just know" how to do it; but this ability sometimes breaks down, and we must figure out what to do. Answers to this foundational problem in human–computer interaction (HCI) have built on James Gibson's concept of *affordance* [26]: "*The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill*" [25]. A defining aspect of affordance is its body-relativity: the tight connection between perception and one's body. In HCI, the concept has become part of textbooks and design guidelines [47, 49], and has seen multiple extensions such as technological [18, 61], interactive [73], and social and cultural affordances [51, 65].

But how do people learn affordances of the objects they interact with? According to the *ecological perspective*, body-relative feature comparisons determine possible actions [67]. For example, both the height of the steps and the length of one's legs influence the perception of whether a flight of stairs is "climbable" or not [75]. But body-relativity is not sufficient for HCI as it does not explain how such features are related to experience. For example, what if one fails to climb a particular flight of stairs; should that affordance not be changed accordingly? We believe so. In contrast, the *recognition* perspective, which has been studied in computer vision and AI, considers affordance to be an act of categorization (or classification) learned via supervised learning [14]. For example, a deep neural net can be trained with images of stairs annotated as "climbable" or "not climbable". Upon failing to climb the stairs, images of those stairs should feed into retraining of the network, thus enabling it to see it as "not climbable". Neither view explains how we discover affordances in the first place.

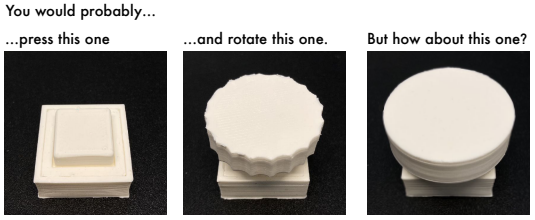


Figure 1: What action would you take with the widget on the right? This paper claims that affordances are associative percepts that map the widget's perceived features to the possible motor actions and their expected utility. The shown widgets are from our empirical study 1.

This paper contributes to the understanding of affordance in HCI, in particular by studying an alternative cognitive mechanism that can explain affordance formation and perception. Furthermore, we show the application of the theory of *reinforcement learning* from cognitive sciences as a tool to explain how we form and adapt affordances through experience. The theory proposed in this paper offers an integrative approach that explains how body-relative perceptions can be obtained and updated via experience.

Our focus is on an HCI-relevant setting where interactive widgets (e.g., buttons and knobs) are presented and a user must decide how to interact with them (Figure 1). We aim to answer two fundamental questions about affordance-formation:

(1) **What cognitive mechanisms underpin affordances?** We argue for two cognitive processes: First, users learn associations between the expected utility of motor actions (e.g., that it is possible to turn an object with a hand) and percepts (e.g., seeing a knob). Second, users learn to assign categories to these motor actions (e.g., turning with hand *means* "rotating"). These two together allow us to perceive an action possibility *with* a label. Both processes have some generalization capacity: they can "leap" beyond previous observations, allowing us to perceive affordances with widgets we have not seen before.
(2) **How are affordances learned?** We propose that the above two types of knowledge are learned via interaction in the presence of reinforcement signals. For example, when you try to turn a widget but it does not turn, negative reinforcement helps you update your beliefs and pick a different motor action next time. The theory of reinforcement learning explains how these two processes – updating and exploring – take place.

In the rest of the paper, we first discuss existing theories of affordance-formation in HCI and other fields. We then argue that reinforcement learning offers a biologically plausible and powerful explanation to the questions of affordance discovery and adaptation. In particular, it can explain how affordance perceptions are updated in a world where we encounter novel designs all the time. The theory achieves this without resorting to any "special mechanism". Rather, affordances are the result of everyday learning. To understand affordance formation and perception, we present findings from two empirical studies with human participants. They were asked to describe or demonstrate what actions were allowed by widgets that they had not interacted with previously. We found evidence for different mechanisms that complement each other, enabling us to make a more accurate judgment of what actions a widget affords. Under uncertainty (i.e., when unsure about the correct action), or when learning and discovering affordances (i.e., figuring out what is the correct ways of using a widget), participants increased the use of motion planning, simulating possible motions in their mind, which is one mechanism in reinforcement learning to predict the utility of possible actions. Finally, we developed a computational model of our theory that enables a virtual robot model to similarly perform affordance elicitation tasks. We tested our model in a simulation environment, where an agent with a virtual arm and eyes interacted with different widgets. We show that when a reinforcement signal is present, affordances could be learned interactively, as predicted.

2 RELATED WORK

We review the present understanding of affordance in psychology, its relevance to design and HCI, and how it is modeled for applications in machine learning and AI.

2.1 Ecological Perspective in Psychology

James Gibson spent years developing the concept of affordance. He offered a definition in his seminal book [26], and concretized it later as follows [24]: *"Subject to revision, I suggest that the affordance of anything is a specific combination of the properties of its substance and its surfaces taken with reference to an animal."* He later characterized it [25]: *"If a terrestrial surface is nearly horizontal (instead of slanted), nearly flat (instead of convex or concave), and sufficiently extended (relative to the size of the animal) and if its substance is rigid (relative to the weight of the animal), then the surface affords support."* To Gibson, affordance refers to opportunities to act based on features of the environment as they are presented to the animal.

Ecological psychologists elaborated on Gibson's theory. The most agreed-on definition, endorsed by Heft [28], Michaels [41], Reed [53], Stoffregen [59], and Turvey [66], is that affordances are body-relative properties of the environment that have some significance to animal's behavior. According to Turvey, affordances are *dispositional properties* of the environment [67]. Dispositional properties are tendencies to manifest some other property in certain circumstances. Something is "fragile" if it would break in certain common but possible circumstances, particularly in circumstances in which it is struck with force. Later, Chermo proposed *relational affordance* [8]. In contrast to the dispositional account, relational affordances are not properties of the environment, nor of the organism, but rather the organism–environment system.

Empirical research on humans has provided evidence for the body-relative perspective. In particular, people are able to judge affordances reliably in various tasks [16, 17, 75]. There is also evidence for body-relativity, such as in the stair climbing experiment [75] and in its extension to doorways [17]. People can accurately

predict if a stair of dynamic height can be stepped on or not. Although Gibson and others mostly agreed that affordance perception relies on body-relative features, it is not clear how they are acquired in the first place. Gibson simply stated that affordances are directly perceived, and we simply *pick up* the *information* [25]. Interestingly, evidence shows that affordances do change with practice [11, 13, 17], and some theorists have proposed that affordances are learned and developed, however without specifying how [22, 23]. To the best of our knowledge, our work is the first attempt to explain how this occurs through interaction.

2.2 Affordance in HCI and Design

Affordance is a fundamental concept in HCI and design. Theories related to it have been developed for years, but the process of formation and adaptation of affordances has not been explained or explored. William Gaver first introduced affordance to HCI by using it to describe actions on technological devices [19]. He defined affordances as *"properties of the world that are compatible with and relevant for people's interaction. When affordances are perceptible, they offer a link between perception and action."* He separated affordances (the ways things can be used) from perceptible information. Hence, he noted that "hidden affordances" (affordances that are not perceptible) and "false affordances" (wrongly perceived affordances) can exist.

Donald Norman later introduced affordance to the design field in his book*The Psychology of Everyday Things* [48]. In his view, affordances suggest how artifacts should be used [47, 49]: *"Affordances provide strong clues to the operations of things. [...] When affordances are taken advantage of, the user knows what to do just by looking: no picture, label or instruction is required."* He implied that affordances should be appropriately incorporated to guide users. He also elaborated that *"affordance refers to the perceived and actual properties of the thing ([48])"*, and later rephrased affordance entirely to "perceived affordance". This suggests that there are *actual* affordances and *perceived* affordances, and they may be different. In contrast, according to Gibson's original definition, affordances are exclusively "perceived action possibilities". McGrenere and Ho [40] later pointed out that due to the lack of a single unified understanding, HCI researchers tend to either follow Gibson's original definition [1, 4, 72, 77], adopt Norman's view [12, 32, 46], or even create their own variations [42, 55, 71]. In an attempt to address the issue, McGrenere and Ho [40] proposed a framework that separated affordances from the information that specifies them. Yet, there are problems related to the meanings of affordance and how to apply it to design [5, 50].

More recently, researchers introduced new types of affordances to expand the concept to more complicated interactions. For instance, several works [2, 3, 33] have proposed an activity-based theoretical perspective to affordances, which is concerned with the social-historical dimension of an actor's interaction with the environment. Turner [65] further classified affordances into simple affordances and complex affordances: *"Simple affordance corresponds to Gibson's original formulation, while complex affordances embody such things as history and practice."* Similarly, Ramstead et al. [52] introduced cultural affordance, which refers to action possibility that depends on *"explicit or implicit expectations, norms, conventions, and cooperative social practices."* While these new classifications expand the application scope of affordance, they also imply that different affordance formation processes may exist in different affordance types. Moreover, the implications and applications of affordances in design practice remain vague.

We argue that there is a need to unveil the discovery, adaptation, and perception mechanisms of affordances. A theory that explains affordance formation through interaction could serve as a common ground and help unify existing theories and definitions. By offering a better understanding of the concept, it can also provide actionable means to understand and improve interfaces.

2.3 Affordance in Machine Learning

Computational models for affordance have been presented in computer vision and robotics. A large number of works consider affordance detection as a combined task of recognizing both the object and the actions it allows. Nguyen et al. [45] suggest a two-phase method where a deep Convolutional Neural Network (CNN) first detects objects; based on this detection, affordances are observed by a second network as a pixel-wise labeling task. Do et al. [14] introduce a combination of object and affordance detection by using one single deep CNN, which is trained in an end-to-end manner, instead of sequential object and affordance detection. Chuang et al. [9] use Graph Neural Networks (GNN) to conduct affordance reasoning from egocentric scene view and a language model based on Recurrent Neural Network (RNN) to produce explanations and consequences of actions. The capabilities for better generalization can be improved by training low-dimensional representations of high-dimensional state representations, such as autoencoders [15, 70]. In Hämäläinen et al. [30], one type of variational autoencoder is used to produce low-dimensional affordance representation from RGB images. In this line of work [35, 43, 57, 58], affordance detection is purely based on extracting features from images by using deep convolutional networks and supervised learning, and there is no exploration or world models involved. Furthermore, the agents are not able to adapt according to the interactions with the environment. Therefore, these models are not suitable for explaining real-world affordance perception.

A few recent works have also looked into Reinforcement Learning (RL) approaches to identify affordances via interaction with the world. Nagarajan and Grauman [44] proposed an RL agent autonomously discovers objects and their affordances by trying out a set of pre-defined high-level actions in a 3D task environment. Similarly, Grabner et al. [27] detects affordance by posing a human 3D model into certain actions on the targets and calculating the probability of success. We are inspired by computational demonstrations like these and develop the argument that the ability to explore via trial-and-errors should be the cognitive mechanism that underpins human affordance learning. However, these methods are limited by pre-defined actions; the agent cannot learn new actions outside the training set nor fine-tune the actions. The assumption of having pre-trained actions is also detached from real-world experience.

Lastly, research in RL and robotic fields brought affordances to the micro-movements, that is, the small actions that an agent can take at each timestep. For instance, Khetarpal et al. [34] defined affordance as the possibility of transition from a state to another

desired state. Manoury et al. [39] trains the agent to learn the feasible primitive actions and by an intrinsically motivated exploration algorithm. This approach allows the agents to learn the possible actions in different states, which effectively boosts the training efficiency. However, the affordance in these works is focused on the possibility of micro, primitive actions, which are distinct from the general notion of affordances of "larger" actions, such as press, grasp, sit. Our work does not extend from this view.

Our work presents a novel framing of affordance formation based on reinforcement learning, and applies this to enable a virtual robot to learn and adapt affordances via interaction.

3 THEORY: AFFORDANCE AS REINFORCEMENT LEARNING

Reinforcement learning is a grand theory that is presently uniting cognitive neurosciences and machine learning in an effort to understand general principles of adaptive behavior [20, 56, 60, 63]. Reinforcement learning can be defined as "the process by which organisms learn through trial and error to predict and acquire reward" [20]. Prior to this paper, reinforcement learning had not been developed as a psychological theory of affordance for HCI. For a review of other applications of this theory in HCI, see [29]. In what follows, we provide a synthesis of the assumptions of the theory as they are relevant for affordance-formation, especially in HCI.

Affordances are learned when reinforcement signals are provided in response to motions. To learn which motor command leads to the highest reinforcement signal (reward), an organism must try out several possible motions. This experience results in concomitant updates in predicted rewards. For example, assume you have never encountered a rotary dial before. If your initial motion (e.g., push) does not lead to the expected or desired feedback, you receive a negative reinforcement signal. The rational response then is to avoid that motion in the future [7]. Even when repeating a correct motion, the exact strategy or action may be further optimized to acquire the positive reward faster and with less effort.

Affordance perception is guided by predicted rewards. The theory of reinforcement learning suggests that prediction is the key to the problem posed by delayed feedback [60]. In order to pick an action right now, the brain must learn to *predict* how good eventual outcomes that choice may lead to [20]. The better these predictions are, the better the action that is based on them. In cognitive neuroscience, dopamine is identified as the transmitter of phasic signals that convey what are called reward prediction errors [21]. We hypothesize that the emergent role of rewards is to report the salience of perceptual cues that leads to a sequence of actions. In this sense, rewards mediate the affordance of cues that elicit motor behavior [10]; in much the same way that attention mediates the salience of cues in the perceptual domain [31].

Affordances are learned by exploring and exploiting. Learning affordances purely via trial and error would be highly inefficient, as there are too many possible motions to try out at random. According to the theory of reinforcement learning, a rational agent, after having identified a satisfactory way to perform, will keep doing that as long as that strategy works (exploitation). In the absence of such a strategy, or when it fails, it needs to explore other options. But when to try out which motion? This is the so-called *exploration/exploitation dilemma* [60]. The theory purports that an organism should pick the action that it expects to accumulate most rewards in some window of time.

Affordance perception generalizes to unseen instances of a category. Affordance perceptions need to generalize. Consider, for example, seeing a beige cup with a triangle-shaped yellow handle. You may have not seen this particular cup before, yet you know how to grasp it. Somehow your experience with thousands of cups in your life transfers to this particular cup. Reinforcement learning proposes two cognitive mechanisms for generalization: 1) generalization of policies via feature-similarity and 2) generalization via motion planning in mind. In the former, two objects that share similar perceptual features are associated with the same policy. The shapes that indicate a cup, and possibly its context, are associated with grasping more than anything else. Such generalization can be modeled, for example, via deep neural nets [36], as in our computational model. In the latter, we simulate motions with our bodies and predict their associated consequences. In reinforcement learning theory, this is called model-based reinforcement learning [20]: a mode of reinforcement learning that is associated with better generalizability but higher effort.

We learn to associate categories (labels) with affordances. Everything that we have stated above could be applied to any animal, not just humans. However, affordance, as it has been treated in previous research, has almost always assumed linguistic categorization of actions [6, 68]. We concur and argue that categorization is not just for sensemaking but an essential mechanism that accelerates the learning of affordances. It enables reasoning, social communication, and acculturation – processes that boost the development of cognitive representations. The most straightforward way to categorize affordances considers the movement itself. For example, motions, where the finger comes in contact with a surface and pushes it downwards, can be classified as "presses". In machine learning terms, this is a classification problem where one needs to go from perceptual input vector to a distribution over possible labels. But we may also categorize actions based on similarities in *consequences*. For example, when a finger pushes down on a keycap and a positive feedback signal ("click" sound) is observed, the motion could also be classified as a "press".

4 STUDY 1: PERCEIVING AFFORDANCES IN INTERACTIVE WIDGETS

To understand the contribution of different cognitive mechanisms in affordance perception and to test the hypothesis that they are adapted based on experience, we conducted two studies. Study 1 aims to shed light on which mechanisms are present in affordance perceptions. More specifically, it seeks to answer two research questions:

> RQ1: Which internal mechanisms are employed during affordance perception?
> RQ2: How does prior experience with particular types of widgets influence affordance perception?

The main idea of the study is to manipulate which widgets are experienced prior to seeing a novel widget (see Figure 2). Affordances have been previously studied in HCI and psychology via a method where participants are presented with some objects, and then asked to self-report perceived actions and feedback [38, 62, 75]. Our method follows this approach where affordance perceptions are elicited using a rating scheme. We hypothesize that affordance perception is a complex process where multiple mechanisms can be flexibly employed. In particular, building on previous work and our theory, we focused on three mechanisms:

- **Feature Comparison**: Here, it is assumed that body-relative features drive our affordance perception, as claimed by the ecological perspective. That is, users compare features of an object (shape, size, texture, structure, etc.) to their own features (finger length, hand size, etc.) to decide what actions are allowed and what affordances are provided.
- **Recognition**: We use visual characteristics and details to identify affordances of objects. Here, attributes of the actor or agent are not considered that essential.
- **Motion Planning**: We simulate possible motor actions to find out which have the highest probability of succeeding when interacting with a widget.

4.1 Participants

Twenty-six (26) participants (15 masculine, 10 feminine, 1 chose not to disclose), aged 21 to 33 ($mean = 27.41$, $s.d. = 4.85$), with varying educational backgrounds, were opportunistically recruited. In the context of the COVID-19 pandemic, the experiment was carried out in accordance with local health and safety protocols. Participants reported normal or corrected vision and no motor impairments. The same set of participants also took part in the second study, presented in the next section. Note that two participants failed to follow the instructions of Study 2, and their data was removed (see subsection 5.2 for more details). In the following, we consider only the remaining twenty-four (24) participants. Each study took under 30 minutes, and the total duration was less than an hour per person. Participation was voluntary and under informed consent; participants were compensated with a movie voucher (approx. 12 EUR).

4.2 Material

Five objects were 3D-printed for the study (Figure 2). These were grouped into two sets of two interactive objects (widgets) and one additional *target* object. Set A consisted of a rectangular pressable widget (similar to a button) and a pseudo-circular rotary widget (similar to a dial). Set B consisted of a rectangular rotary widget and a pseudo-circular pressable widget. The final object ("target") was circular and non-interactive. Each widget consisted of a base and a handle. The dimensions of the base was consistent across widgets ($2.3cm \times 2.3cm \times 0.8cm$). The circular handles had a diameter of 3 cm but varied slightly in their exact shape. The square handle had dimensions of $1.5cm \times 1.5cm$. The objects were presented to the participants on a desk in front of them.

4.3 Procedure

The study consisted of two rounds. Each round had a *training phase* followed by a *testing* phase.

4.3.1 Training phase: One set of two widgets (A or B) was presented; participants were invited to interact with them to discover the right ways of operating them. We then asked them to provide the action(s) each object enabled to confirm that they had discovered the correct one. Note that we avoided mentioning the term "affordances" to prevent varying interpretations and bias between participants.

4.3.2 Testing phase: After interacting with a set of two widgets, participants were presented with the round-shaped target object (Figure 2-Target), but asked to not interact with it. Instead, they were asked to self-report the action(s) this object would allow, demonstrate these action(s) through mid-air gestures, and explain them in their own words. A list of five possible actions[1] (press, rotate, pull, slide, tilt) were presented, and participants were asked to rate their affordance perception on a 7-point Likert scale for each; a rating of 1 meant that they strongly disagreed that the action was supported and 7 meant that they strongly agreed.

In the second round of the study, these two phases were repeated but with a different set of two widgets (B or A). The presentation order of widget sets was counter-balanced between participants.

4.3.3 Semi-structured interview: After the two rounds, participants were asked to describe, in as much detail as possible, their process for making judgments about the perceived action(s) for the target object they encountered in both rounds.

4.3.4 Questionnaire: Next, we presented them with the three candidate mechanisms (feature comparison, recognition, motion planning), along with simple explanations for each, and asked them to rate their relevance during perception tasks on a 7-point Likert scale; a rating of 1 signified that they strongly disagreed that the mechanism played a role during their tasks, while 7 meant they strongly agreed that the mechanism played a vital role. They were also requested to provide a rationale for their ratings.

4.4 Result: Relevance of mechanisms for affordance perception

We analyzed participants' ratings for each mechanism and their responses in the open-ended interviews. The median values for the feature comparison, recognition, and motion planning mechanisms were 4 ($IQR = 5.25$, $mean = 4.17$, $s.d. = 2.25$), 7 ($IQR = 0.75$, $mean = 6.60$, $s.d. = 0.77$), and 7 ($IQR = 1.75$, $mean = 6.21$, $s.d. = 1.03$), respectively. A Friedman Test showed statistically significant difference between these mechanisms ($\chi^2(2) = 22.694$, $p < 0.001$). A post hoc analysis with Wilcoxon Signed-Rank Test was conducted with a Bonferroni correction, and the result showed that the feature comparison mechanism was reported to be the least applicable (statistically significant) compared to the recognition mechanism ($Z = -3.644$, $p < 0.001$) and the motion planning mechanism ($Z = -3.218$, $p = 0.001$). The difference between recognition and motion planning was not statistically significant ($Z = -1.768$, $p > 0.05$).

[1]These five actions were identified during a pilot study with 5 participants. During similar tasks, these were the most frequently reported actions.

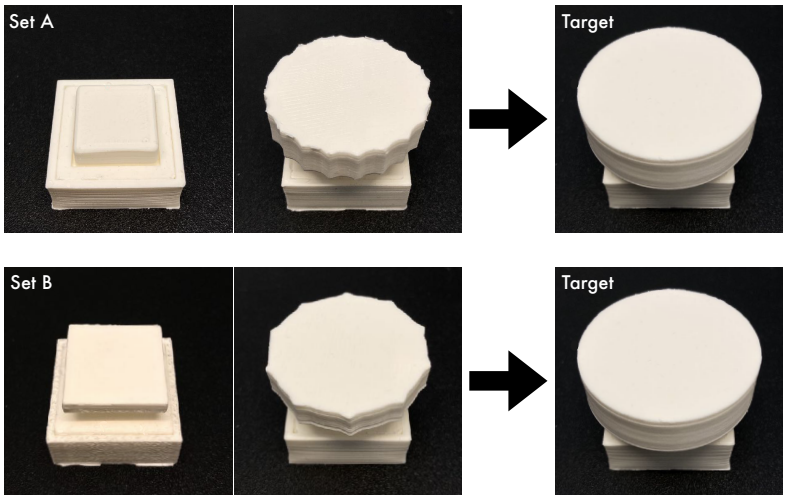


Figure 2: In Study 1, participants interacted with a set of widgets (training) and were then asked how they would interact with a novel widget (testing). We manipulated the widget sets (A or B) with which they interacted in the training phase. SET A: the round-shape (right-hand side) object affords rotation, and the rectangular (left-hand side) object affords pressing. SET B: the round-shaped (right) object affords pressing while the square object (left) affords rotation. The target object: the one which participants were asked not to interact with but just report the affordance in the "testing phase".

Some participants stated that the physical features of the target object were very similar to everyday objects they had used before, so they do not need to consider the feature comparison mechanism as much: *"I have considered the size and features, but it's very quick and automatic because the shape and dimension are very common. Most of my thoughts are on considering what it is and what is the right action to do."* (P17). Meanwhile, the other two mechanisms received high scores, indicating that most agreed with their relevance for affordance perception.

Feature Comparison: Despite receiving a lower overall score, 12 participants mentioned their mental process involved making body-relative feature comparisons, indicating it was useful, but just not as much as the other mechanisms. P14 mentioned: *"The size of the top part (handle) is more or less matched with my thumb's size, so it probably is designed for pressing or tilting"*, and P20: *"I think it can't be pulled because the gap between the object (the handle) and the table is too narrow for my fingers to squeeze in"*.

Recognition: 20 participants reported using the recognition-based approach. Many (15) mentioned the structure and the round shape of the target object directly reminded them of the previous widget sets. Many participants (20) mentioned that the target object reminded them of other round widgets they had encountered in daily life: *"It is very similar to a widget I have seen on a radio or a microwave."* (P15). *"It is a button I see a lot on different machines, so the most likely action to me is pressing-down."* (P16).

Motion Planning: 12 participants reported using motion planning: *"When I see the object, I instantly imagine pushing it and rotating it (meanwhile making the gesture in mid-air), so I conclude that it has the possibilities of these motions"* (P1). Some commented imagining motions to eliminate infeasible actions: *"I tried to press it in imagination, but it seems not pressable. I can sort of imagine pushing and getting stuck, so I gave a lower score"*.

4.5 Result: Changes in affordance perception during the study

We looked at how perceived affordances changed as the study progressed and participants encountered different widgets. The plots in Figure 3 summarize the results. A more detailed analysis for each action follows.

Press action: For the perceived press action on the target object, Wilcoxon Signed-Rank Test showed statistically significant difference between SET A and SET B ($Z = -2.273$, $p < 0.05$). The trend was consistent regardless of the presentation order of the two sets. This shows that recent interactions with other objects (training phase) influence the affordance perception with a new object.

During the open-ended interviews, we probed to identify possible reasons. 10 users reported changing their rating of the press perception guided by the *recognition* mechanism. That is, upon seeing an object similar to the ones they interacted with, they recalled the nearest reference image (the round-shape object in SET A or SET B). P1 mentioned: *"The object (target) is round, so I link it to the previous round widgets that I have tried with. If the round thing offers rotation this time, I will think this one (target) is rotatable. If the previous round thing offers press, I also tend to think maybe it (target) also can be pressed."* 7 users attributed their change in press

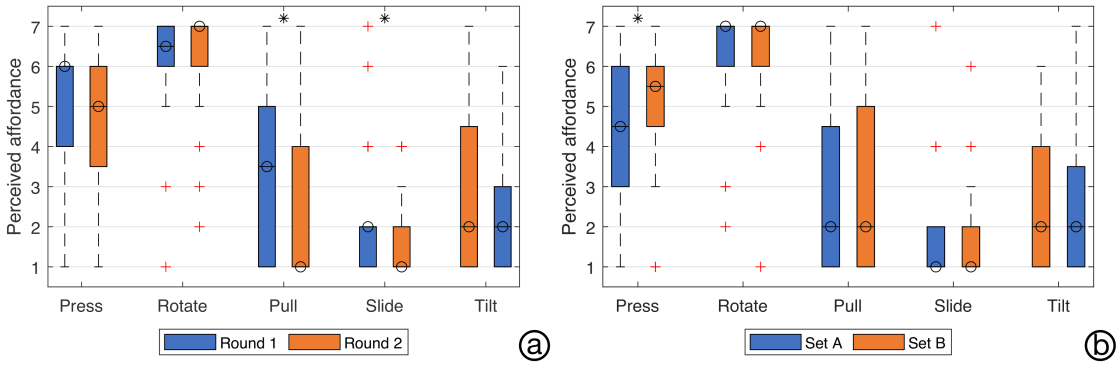


Figure 3: Study 1 shows that the reported affordances of a target widget change when the training widgets change. The red crosses mark outliers, which are defined as beyond 1.5 * IQR +/- Q3/Q1. The one-star (∗) symbol indicates $p < 0.05$ and significant difference. (a) Affordances reported for the target widget on Round 1 and Round 2. (b) Affordances depending on the preceding widget Set.

ratings to *motion planning*. That is, upon seeing a similar object, their planned motions followed the previous interactions in SET A and SET B. P3: *"(In the second round,) I learned that the round shape can be pressable. So when I saw the last thing (target), I directly imagined and wanted to press it. That imagination (pressing it) happens only now. I did not think of it in the previous round (interacting with set A)."*

Pull and slide actions: These actions had consistent ratings between SET A and SET B. We further examined the ratings for these two actions based on the overall time progress (i.e., between the two rounds) regardless of the widget sets. Wilcoxon Signed-Rank Test showed a statistically significant difference between the perceived level of "pull" in the first round and the second round ($Z = -3.355$, $p < 0.05$), indicating the pull perception decreased over time. The perceived level of "slide" exhibited a similar trend ($Z = -2.14$, $p < 0.05$). This shows as participants gained experience, upon not observing these actions with any of the widgets during interactions, their perception or belief about the presence of these affordances reduced over time. In P5's words: *"After interacting 2 objects, I still consider pulling a little, and imagine if it's possible to do. But after experiencing four widgets and learning there are not pulling there, I just stopped to believe that is an option. I don't think of this action or try to simulate it anymore."*

Rotate and tilt actions: There was neither a statistically significant difference between the two widget sets nor between the round for these actions. This can likely be attributed to the target object showing a clear hint of rotation as its shape is very aligned with other rotating objects, which participants have encountered during their everyday interactions, and not showing any indication of being tiltable due to its shallow depth.

4.6 Summary

To answer RQ1, we observed that users employ a combination of all three mechanisms, although motion planning and recognition tend to play more important roles. Addressing RQ2, our statistical analysis revealed the importance of interactions for changing or adapting the perception of affordances. From our qualitative analysis, we learned that this change could be attributed to *motion planning* and *recognition*. Findings from this study offer promising evidence for the existence of our theory during affordance perception tasks. To summarize, study results favor the existence of our theory, and illustrate the process of how motion-planning (RL-based affordance) leads to affordance formation and adaptation through interactions.

5 STUDY 2: ADAPTING AFFORDANCE PERCEPTION WHEN WIDGETS CHANGE

The second study seeks to understand how people adapt their perception of affordances when widgets change dramatically or behave unexpectedly. We asked participants to first identify allowed actions (affordances) on a "deceptive widget", which appeared to be a slider but, in fact, only allowed button-like press actions. Following this, they could perform a single interaction to verify if their perception was accurate (Figure 4). This identification and interaction could be repeated until they correctly identified the widget. We hypothesize that users can adapt and update their affordance perception under such circumstances through motion planning and reinforcement signals (success/failure) during an interaction. If an action leads to a failure signal (e.g., attempting to slide the handle but the handle does not move), the user updates the motion plan in the next iteration, and the detected affordance is also updated. Conversely, if an action leads to a success signal (attempting to press the handle and successfully translating it downwards), the user determines the affordance accordingly.

5.1 Material

A deceptive widget that visually resembled a slider but operated like a button was 3D-printed. The only action allowed by it was a press down on the handle. The base of the widget was 8.5 cm × 2.5

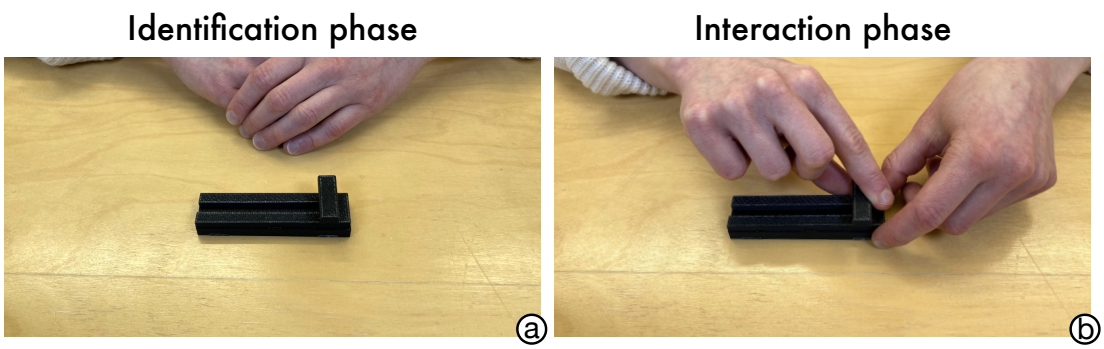


Figure 4: In Study 2, participants were shown a "deceptive widget" and asked to (a) identify what actions they perceived it to allow, following which they could (b) perform a single interaction with it to verify their perception.

cm × 1 cm, and the handle was 1 cm × 3 cm × 0.4 cm. The widget was placed in horizontal orientation, on a desk, at a distance of 15 to 20 cm from the participant. The widget and setup is illustrated in Figure 4.

5.2 Method

During the study, a round consisted of two phases: *identification* and *interaction* (Figure 4).

5.2.1 Identification Phase: At the start of each round, participants were asked to identify and report the most likely action allowed by the widget without making any physical contact. If they perceived multiple actions, they could mention all of them. In addition, they were asked to first openly describe their mental process of making the judgment. Finally, we asked them to rate the relevance of each of the three affordance mechanisms for action identification on a 7-point Likert scale, along with follow-up questions to better understand the rationale behind their ratings.

5.2.2 Interaction Phase: Here, participants were asked to interact with the widget by taking the most likely action they had identified in the previous phase without making other movements. The participants were requested to immediately inform the experimenter if they performed more than one action. The movements were fully recorded by cameras which were set in a short distance, and all the videos were examined afterward for verification. Two participants failed to follow the instruction and performed more than one movement in one round. Their data was consequently removed.

5.2.3 Termination: Participants could complete as many rounds as they felt necessary to confidently identify the correct action allowed by the widget. They could also stop without identifying any action if they determined there were none allowed.

5.3 Result: Change in Affordance Perception

We first analyzed the perceived affordances and their change across different rounds. On average, participants completed 3.17 rounds ($s.d. = 0.64$, $median = 3$) before stopping exploration of further possible actions (7 participants took 4 rounds, 14 took 3 rounds, and 3 took 2 rounds). 23 out of 24 participants successfully perceived the "press" action within 2 to 4 rounds, while one participant did not identify any possible action.

In the first round, among the 24 participants, 22 perceived "slide" to be the most probable action, 1 perceived "rotate", and 1 "pull". Because every participant's first action led to failure, all of them decided to continue to the second round. Here, 7 participants considered "press" as the most possible action. 11 participants selected "rotate", 4 selected "tilt", and 2 chose other actions. 3 participants that correctly identified "press" concluded after two rounds. 21 participants attempted a third round, including 4 that had already discovered "press" in the second round. 9 additional participants identified "press" here. One participant failed to identify any possible actions and concluded that there was no action allowed by this object. All 7 participants that attempted the fourth round discovered the "press".

5.4 Result: Self-reported Mechanisms

In addition to identifying perceived actions, participants also self-reported the mechanisms they employed for identifying and adapting their affordance perception. Figure 5 provides an overview of the results. It is evident that the three mechanisms were similarly used in the first round. However, once the participants interacted with the widget and noticed that their perception was inaccurate, the relevance of recognition and feature comparison consistently dissipated and motion planning gained prominence. Eventually, motion planning was the most relevant mechanism at play when identifying and adapting perception under uncertainty.

Since only seven (7) participants reached the last round, we analyzed the data of only the first three rounds with Friedman Tests. If there was a statistically significant difference, we used Wilcoxon Signed-Rank Tests with a Bonferroni correction for post hoc analysis. In the first round, there was no statistically significant difference between the reported mechanisms ($\chi^2(2) = 10.204$, $p > 0.05$). In the second round, there was a statistically significant difference between the mechanisms ($\chi^2(2) = 14.075$, $p < 0.05$). Motion planning had the highest rating ($median = 7$, $IQR = 1$),

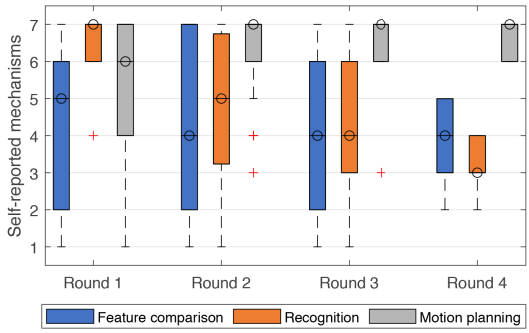


Figure 5: The relevance of each mechanism in the User Study 2. The red crosses mark outliers, which are defined as beyond 1.5 * IQR +/- Q3/Q1. Results suggest that users were initially relying on all three mechanisms but shifted to motion-planning as they gained more experience with the widget.

followed by feature comparison ($median = 4$, $IQR = 5$) and recognition ($median = 5$, $IQR = 3.5$). Post hoc analysis showed a statistically significant difference between motion planning and feature comparison ($Z = -2.623$, $p = 0.009$) and between motion planning and recognition ($Z = -2.534$, $p = 0.011$); the difference between feature comparison and recognition was not statistically significant. In the third round, there was again a statistically significant difference between the mechanisms ($\chi^2(2) = 26.471$, $p < 0.001$). Motion planning had the highest rating ($median = 7$, $IQR = 1$), followed by feature comparison ($median = 4$, $IQR = 4$) and recognition ($median = 4$, $IQR = 3$). Post hoc analysis showed a statistically significant difference between motion planning and feature comparison ($Z = -3.208$, $p = 0.001$) and between motion planning and recognition ($Z = -3.131$, $p = 0.002$); the difference between feature comparison and recognition was not statistically significant. While we did not run statistical analysis for the data in the last round due to too few data points, we can observe a similar trend that motion planning was generally more used.

Open-ended comments shed further light on how participants applied the mechanisms during the task. In the first round, most participants (22) reported perceiving the "sliding" action based on the recognition mechanism. As participants noted: *"It is a slider"* (P5), *"Reminds me of the slider on a panel to control volume"* (P20). Similarly, many participants (15) recalled using motion-simulation as a strategy. As P17 said: *"The motion of holding it and pushing (sliding) it along the direction just came to my mind naturally."* However, after failure in the first round, participants found the recognition mechanism to be less useful because the visual details of the deceptive widget did not strongly resemble objects other than a slider. As P1 said in the second round: *"After the previous fail, if I only look at the handle part, it starts looking like a widget for pulling. But the whole object still does not give me that clue what it is or what should I do."* P19 gave a blunt response after decreasing the recognition score from 7 to 2: *"Because it doesn't work. I thought it's a slider by its look, but it isn't."* Users responded that the motion planning mechanism guided them to discover and adapt actions. P2 mentioned in round 2: *"Even though it doesn't look like a button or a lever to me, I can still imagine pressing it or pulling it. It's coming not from knowing what it is, but more like I can see that motion possible."*

5.5 Summary

This study assessed how users adapt their affordance perception when they are presented with novel objects that depart from their expectations and the role of different mechanisms during this adaptation process. We observed that initial perception was guided by all three mechanisms. However, upon failure, participants adapted quickly, enabling them to successfully identify the appropriate affordance. This adaptation was primarily guided by motion planning and the feedback (reinforcement signals) they received during interactions.

6 A VIRTUAL ROBOT MODEL

Our studies provide evidence for the theory that human perception of affordance involves reinforcement learning where the motion planning mechanism plays a key role. A distinct benefit of our approach over previous theories of affordance is that it can be implemented as a generative computational model that can be subjected to tasks where its affordance-related abilities are tested. The theory and the model are linked via the theory of reinforcement learning in machine learning [60]. Following this methodology, we built a computational model and demonstrated it on a robotic agent, which faced a similar task as in our study 2. In this section, lower case *reinforcement learning* refers to the cognitive mechanism while upper case Reinforcement Learning (or RL) refers to the machine learning method. For an introduction to RL, we refer readers to Sutton and Barto [60].

6.1 The Interactive Widgets Task

Before introducing our affordance model, we briefly describe the task that was used to develop the theory and the model. Similar to our empirical studies, it is motivated by a real-world scenario where people come across unknown objects, widgets, or interfaces in their daily lives and learn or adapt how to interact with them.

In this task, we present a virtual robot with a set of randomly selected interactive widgets. As shown in Figure 6, the agent is presented with a widget at a random location and with a randomly sampled shape and size. Upon successfully reaching the goal state of the widget, it receives a reward signal. Here, the goal state refers to the correct manipulation of the widget. For example, a button widget has a goal state of being pressed down, a slider with a goal state of the handle being moved from left to right. The goal for the agent is to learn the right affordance (action allowed) through interactions with the widget. As a novel widget might be unconventional (similar to the widget in study 2), the agent should be able to adapt its perception appropriately.

6.2 The Virtual Robot

We implemented a virtual robot with an arm whose characteristics are similar to the human arm. As shown in Figure 6, the robot has a shoulder mounted in a static 3D position above a table, and the

upper arm (24 cm long) is connected to the shoulder. An elbow joint connects the upper arm to the forearm (27 cm). The other end of the forearm is the wrist and two finger-like contact points (6 cm each). The entire robot is controlled by 7 virtual motors. A motor can exert a force between 0 to 200 units in two possible directions. Each motor controls one degree of freedom: two motors control the shoulder movement, one motor controls the elbow, one motor controls the rotational movement of the forearm, two motors control the wrist, and the final motor controls the grip of the two fingers.

6.3 Interactive Widgets

We implemented two types of common widgets (Figure 7) – *buttons* and *sliders* – each associated with a particular action. A button affords the "press" action while a slider affords "slide". In addition, we implemented a "deceptive widget" similar to that in our study (section 5), which appears as a slider but operates as a button.

Each widget has two parts: the *handle* and the *base*. The handle is the part the agent interacts with; we designed handles as cuboids with varying dimensions. The base is an immovable part fixed on a table.

(1) Buttons (Figure 7-a): Every button has a handle with a square footprint (equal width and length). The exact size of the width and length is randomly selected during task trials ($\in [3cm, 5cm]$). The height of the handle is set to $3cm$. The base is also square, and has a dimension 1 cm larger than the selected handle width and length to provide padding.
(2) Sliders (Figure 7-b): Slider handles have a rectangular footprint, with their length being larger than their width, similar to sliders found in the real world. During trials, the length is $\in [4cm, 6cm]$ while the width is $\in [1cm, 2cm]$. The height of the handle is set to $4cm$. The base of the slider is rectangular, and has a length 1 cm larger than the handle length and width 10 cm larger than the handle width.
(3) Deceptive Widget (Figure 7-c): The deceptive widget appears similar to a slider; its handle is $2.5cm(width) \times 5cm(length) \times 4cm(height)$, and the base is $12.5cm(width) \times 6cm(length)$. The height of the handle is set to $4cm$.

At the start of each trial, widget dimensions are sampled, and the origin of the widget is randomly positioned within a $5cm \times 5cm$ area on the table to prevent the agent from learning absolute positions.

6.4 Modelling Assumptions

We here describe the main ideas in how we implemented the theoretical assumptions in Section 3. The description requires some familiarity with RL.

First, we assume that the robot's adaptive behavior results from a solution to the Markov Decision Process (MDP) (see below). MDP is a mathematical framework to formulate RL problems; specifically, multi-step, sequential decision-making problems where rewards are deferred. A more formal definition of our MDP will be provided below. Within this framework, we view affordances as a control problem, which maps the percepts (of the observed environment) to possible motor actions (a possible motion) through a value estimate (rewards). This links affordance to policy models.

How to learn a policy model is a central question for Reinforcement Learning. If an action leads to a bad outcome (lower rewards), the policy model will decrease the probability of doing the same action in the future. If an action leads to a good outcome (higher rewards), the policy model will increase the probability of taking similar actions. Via this process, affordances can be updated based on experience with different types of widgets.

We further assume that people can simulate possible motions in their minds and categorize them, essentially giving them labels. Our implementation assumes that the agent recognizes types of actions based on similarities in movements. For example, all motions that end with wrist-rotation movements will be classified as one type of action, and those that end with a finger poking downward will be seen as the same type of action. In our model, we assume that these labels are given by an outside source. This allows implementing recognition with supervised learning. For instance, all successful motions to activate a button are labeled as "press".

The proposed computational model is distinct from all the approaches that have been reviewed in subsection 2.3. The agent is not detecting the affordance purely based on visual features (recognition) [14, 45]. Instead of trying out the pre-trained actions on target objects [44] or learning affordances at the level of primitive actions [39], our agent searches for the optimal policy via exploration-and-exploitation enabled reinforcement learning. Thus, the agent is able to develop novel action plans and fine-tune the learned actions for each object.

6.5 Model Details

We model the interaction of the agent with an object as a Markov Decision Process (MDP). Here, the agent takes an action $a \in A$ to interact with its environment, causing the environment's state to change from $s \in S$ to new state $s' \in S$ with a probability $T(s, a, s') = p(s' \mid s, a)$. The reward function R specifies the probability $R(s, a) = p(r \mid s, a)$ of receiving a scalar reward $r \in \mathbb{R}$ after the agent has performed an action causing a transition in the environment. The agent acts optimally and attempts to maximize its long-term rewards. It chooses its actions by following a policy π, which yields a probability $\pi(s, a) = p(a \mid s)$ of taking a particular action a from the given state s. An optimal policy π^* maximizes the total cumulative rewards $R = E[\sum \gamma^{t-1} r^t]$ where $\gamma \in (0, 1)$ is the discount factor. In essence, a model following an optimal policy selects the action that, in the current state of the environment, maximizes the sum of the immediate and discounted future rewards. The MDP formulation for our model and task is as follows:

State s. A state encapsulates properties of the agent and the environment, which can be observed by the agent:

(1) *Proprioception*: The joint angles and angular velocity of all the joints of the arm.
(2) *Widget dimension*: The three dimensions (width, length, and height) of the widget handle, and the two dimensions (width, length) of the base (the base is assumed to be no height).
(3) *Widget position*: The 3D positions of the center of both the handle and the base.
(4) *Widget velocity*: While the base is fixed, the handle is mounted on a virtual spring (for buttons) or rail (for sliders), allowing

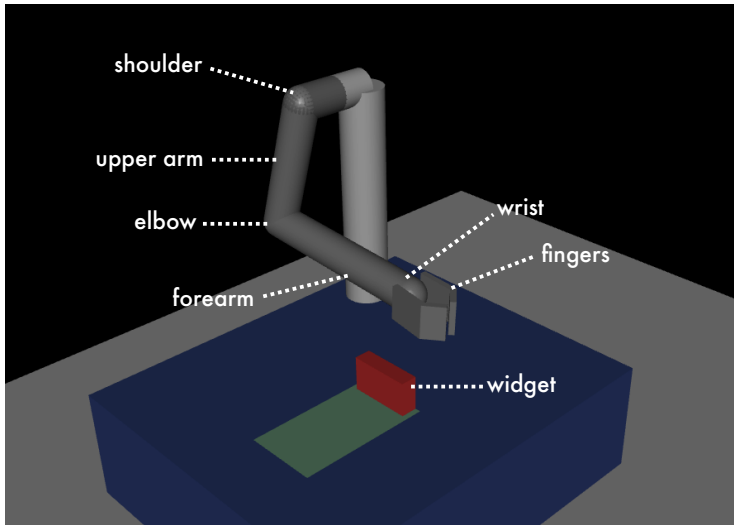


Figure 6: We develop a computational model of our affordance theory. It is implemented in the MuJoCo physical engine, and enables a virtual robot to interact with widgets.

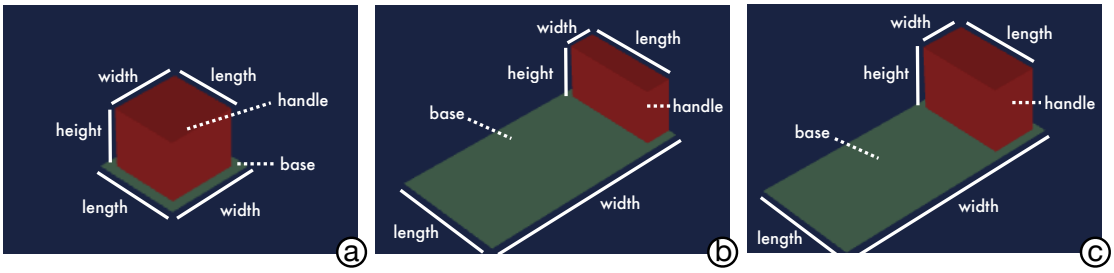


Figure 7: The virtual robot interacted with different types of widgets: (a) Button widgets; (b) Slider widgets; and (c) A deceptive widget that resembles a slider but only allows push actions similar to buttons.

it to move. The velocity of the handle is encoded in the state to allow the agent to perceive the effects of its actions.

Action a. An action is a motion performed by the robot at a single time step. As the agent has 7 joints (degrees of freedom), an action is represented as a vector with 7 values, each representing the force applied by a motor on one joint. An interaction with a widget is composed of a sequence of such actions.

Reward R. When an action a is taken at a time step, the environment (in state s) generates a reward $r(s, a)$. The reward is composed of three parts: distance penalty, movement penalty, and task completion reward. The *distance penalty* is set to be the distance from the center of the agent's fingers to the center of the widget handle, multiplied by a linear factor and constrained to be a value $\in [-0.01, 0]$ (the closer to the target, the less penalty received). The value is set to be relatively small to allow stable learning and avoid over-guiding. The *movement penalty* is the average joint angular velocity, multiplied by a linear factor and constrained to be $\in [-0.01, 0]$ (the faster or more you move, the more penalty received). This is analogous to effort or strain exerted when we make motor movements. Finally, if the widget is successfully triggered (i.e., the button is pressed 2 cm downward or the slider is moved 4 cm toward the target direction), the agent receives a *task completion reward* of value 1, otherwise 0.

Transition $T(s, a, s')$. Taking actions results in state transitions. In our environment, transitions are deterministic; that is, given initial state s, taking an action a always results in the next state s' with probability 1.0.

Discount λ. The discount rate is set to be 0.99 throughout the whole experiment, as it is a generally recommended value for similar robotic applications.

6.6 Implementation

The task environment was implemented within the Mujoco physical simulation engine[2] [64], which is commonly used for robotic simulation and reinforcement learning applications. Our MDP agent was trained using Proximal Policy Optimization (PPO) [54], a state-of-the-art Reinforcement Learning algorithm. For technical details and hyperparameters, refer to the supplementary material.

6.7 Training with Common Widgets

We trained our agent to interact with widgets that had distinct affordances. To evaluate whether it could then adapt its affordance perception through reward signals, we tested it with a novel widget. This widget is analogous to the widget used in our second empirical study (section 5), which investigated how humans adapt their affordance perception under uncertainty.

We trained our model to interact with the two widget types described previously – buttons and sliders. The training process is shown in Figure 8-a. After training, in 1000 trials with each widget, the agent interacted with the button (by pressing it down) with a success rate of 91.3%, and with the slider (by sliding it in the target direction) with a success rate of 94.5%. This offers promising evidence for the agent's ability to interact with objects through reward signals.

Next, we labeled 1000 successful trials with each widget with their corresponding affordance labels: "press" for the button and "slide" for the slider. This dataset was used to train an affordance classifier that can assign labels to motions. We randomly sampled 80% of the data for the training set, 10% for validation, and 10% for testing. The motion classifier achieved 85.8% validation accuracy and 88.1% testing accuracy, indicating high-quality recognition. It is challenging to achieve higher motion recognition due to some small overlap in motions that can occur during interactions with different types of widgets. While pressing a button and sliding a slider are generally different actions, there are some motions that could successfully trigger both buttons and sliders.

6.8 Testing with the Deceptive Widget

After training our model to interact with common buttons and sliders, we introduced it to the deceptive widget. Despite its resemblance to a slider, this test widget has the affordance of a button – press actions provide positive rewards. The agent interacted with this widget and continued learning through these interactions. During these interactions, each action was labeled with a corresponding affordance label using the pre-trained classifier.

6.9 Results

The results are presented in the Figure 8-b and c, which shows the progress in the agent's affordance perception.

We can see that the agent initially perceived higher sliding affordance, resulting in corresponding unsuccessful actions and low success rates. However, through interactions, the agent was able to adapt and learned to perceive the press affordance. As seen in the figure, after 20 simulated time units, the press affordance is dominant. However, the agent's success rate for performing press actions does not immediately improve; this can be attributed to the difference in physical properties of this widget compared to previously learned button widgets. Recall from our empirical study (section 5), human participants also exhibited similar behavior when perceiving affordances. They initially indicated "sliding" being the primary affordance for the widget; after some interactions, they adapted and could perceive the press affordance instead.

6.10 Summary

The results show that driven by reinforcement learning, our robot agent successfully interacts with different widgets, discovers correct affordances, provides labels for them, and adapts its perception when it encounters unfamiliar circumstances as humans do. While the results largely aligned with the trend in human data, the efficiency of adaptation is lower. The potential solutions to mitigate the differences are discussed below.

7 DISCUSSION

How do we learn and perceive affordances? Though the concept of affordance has been previously recognized and modeled using ecological and recognition approaches, the question of how affordances are formed was left unanswered. The paper tackles this grand question by proposing a novel theory based on reinforcement learning. We argued that affordances are learned via trial-and-error when reinforcement signals are present, and continue to adapt in everyday interaction as we encounter different types of interfaces. Our results are in line with the current theory in reinforcement learning, and suggest how we can generalize affordances to previously unseen interfaces. Through *exploitation*, we can intuitively "just see" or perceive the right affordances and act accordingly. When this fails, through exploration, we can resort to motion planning in mind: that is, we can simulate alternate motor actions and assess their success. This exploration process, which is more effortful, works similar to model-based RL in machine learning. This result offers a neat synthesis of previous theories. In particular, the body-relative features that ecological psychology underlines are essentially our affordance percepts: they map observed features to estimated utility in operating the widget *with one's own body*. Our view is also compatible with the recognition-based approach. We believe that categorization of affordances is the key to users' ability to talk and reason about affordances, which facilitates their learning.

An interesting comparison is between our model's performance and human performance. Human participants adapted both motion and affordance perception much faster than our robotic model. The model took nearly 40 policy updates to achieve a 90% success rate and perceive press affordance with a probability of 80%. In contrast, human participants achieved the same results within 2 to 4 trials. Based on the analysis done, we identify three reasons that lead to this difference. First, participants in our studies have had extensive experience of interacting with everyday objects and widgets similar to those presented in the studies; in contrast, the virtual robot had only limited experience via training. Second, participants leveraged

[2] http://www.mujoco.org/

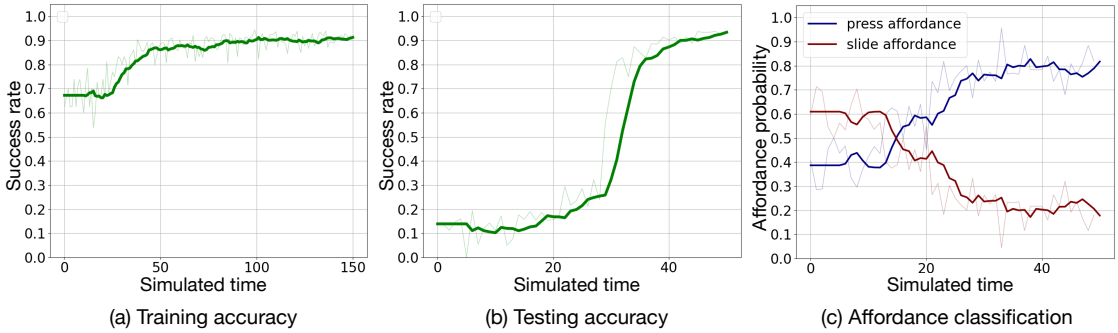


Figure 8: The virtual robot model successfully learned correct affordances by interacting with widgets. Through interactions and adaptation, it achieved a high success rate and labeled the affordance correctly. (a) The accuracy over time during the training phase with basic widgets. (b) The accuracy in the testing phase with a "deceptive widget". At the time 0, the testing deceptive widget was introduced. (c) Adaptation in affordance perception over time during testing.

all three mechanisms we listed in the user study to help identify the correct affordance. However, our model relied exclusively on motion planning, and the policy adapted to the reinforcement signal provided in each round. Third, humans have a striking ability to learn and adapt to new environments or concepts from a limited number of examples, an ability termed *meta-learning* or *learning to learn* [74]. This ability contrasts strongly with the machine learning methods used in our model, which typically require many interactions to reach a similar success rate. Nonetheless, our model shows a trend similar to human affordance formation and adaption. As a future extension to our model, we will consider an integrated model that can utilize multiple affordance mechanisms and will apply more advanced REINFORCEMENT LEARNING techniques, such as META-REINFORCEMENT-LEARNING for more efficient learning.

Based on its ability to discover and learn affordances, our computational model has practical applications for design tasks. Such a model can enable designers to evaluate the usability of novel design instances. It could answer questions such as *"what affordances would a user perceive when interacting with this new design?"* and *"how would a user adapt their perception to a new design candidate?"* For example, an agent could be trained to interact with a wide variety of door handles that require "turning" motions. Upon encountering a novel doorknob design, the agent could reveal the level of perceived "turn" affordance by creating a set of motions and then classifying them, thus enabling the designer to identify whether the new design would be intuitive or not. As the model also adapts with experience, it could reveal how much time users might require to adapt to a novel design instance. Furthermore, by varying the physical properties of the robot agent, such as its dimensions, degrees of freedom, or other motor capabilities, it could facilitate usability and accessibility testing for a range of user groups.

Finally, our work aims to provide a common ground for a better understanding of affordance in HCI. For instance, false affordances and hidden affordances [19, 76] are well-known concepts that lead to poor design, but why do they exist and how to avoid them are unclear. According to our theory, users learn to associate certain action plans to certain visual cues from past experience. Poor designs (with false or hidden affordances) are the ones where planned (assumed) motions do not lead to predicted reinforcement signals. With our theory, one can further reason the quality of a design based on the planned motions and reinforcement learning. Our theory further provides a shared formation process compatible to all the classes of affordances. No matter whether an affordance is simple [65] (a ball affords grasping), complex [19, 65] (a scrollbar on a monitor affords scrolling with a mouse), or social or cultural [52], they are all acquired via the same mechanism – reinforcement learning. Lastly, our theory sparks an interesting discussion for the future: the disputed notion of natural interactions [69] or natural user interfaces [37]. In our view, even perceiving the most *natural*, *intuitive* interaction possibility, such as a button pressing, is one that is *discovered*, *learned*, and constantly *adapted* with experience. To conclude, this paper studies the concept of affordance: a key term that was introduced to HCI decades ago. To date, due to the lack of a theory that explains underlying mechanisms and the formation process, the term has assumed varying interpretations and led to vague applications and implications. We anticipate that this paper will lead readers to rediscover affordance by shedding light on how we, as humans, rediscover and adapt our affordance perception. Our theory could establish a more actionable understanding of affordance in design and HCI, and our model could bring affordance from a conceptual term to a usable computational tool.

8 OPEN SCIENCE

Anonymized data from the two user studies, the virtual robot model (MuJoCo), and the RL model (Python) will be released on our project page at http://userinterfaces.aalto.fi/affordance. The supplementary material provides further technical details about the model implementation in subsection 6.6.

ACKNOWLEDGMENTS

This project is funded by the Department of Communications and Networking (Aalto University), Finnish Center for Artificial Intelligence (FCAI), Academy of Finland projects Human Automata (Project ID: 328813) and BAD (Project ID: 318559), and HumaneAI. We thank John Dudley for his support with data visualization and all study participants for their time commitment and valuable insights.

REFERENCES

[1] Mark S. Ackerman and Leysia Palen. 1996. The Zephyr Help Instance: Promoting Ongoing Activity in a CSCW System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, British Columbia, Canada) *(CHI '96)*. Association for Computing Machinery, New York, NY, USA, 268–275. https://doi.org/10.1145/238386.238528

[2] H Albrechtsen, Hans HK Andersen, S Bødker, and Annelise M Pejtersen. 2001. Affordances in activity theory and cognitive systems engineering. *Rapport Technique Riso* (2001).

[3] Klaus B. Bærentsen and Johan Trettvik. 2002. An Activity Theory Approach to Affordance. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction* (Aarhus, Denmark) *(NordiCHI '02)*. Association for Computing Machinery, New York, NY, USA, 51–60. https://doi.org/10.1145/572020.572028

[4] Marina Umaschi Bers, Edith Ackermann, Justine Cassell, Beth Donegan, Joseph Gonzalez-Heydrich, David Ray DeMaso, Carol Strohecker, Sarah Lualdi, Dennis Bromley, and Judith Karlin. 1998. Interactive Storytelling Environments: Coping with Cardiac Illness at Boston's Children's Hospital. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Los Angeles, California, USA) *(CHI '98)*. ACM Press/Addison-Wesley Publishing Co., USA, 603–610. https://doi.org/10.1145/274644.274725

[5] Leonardo Burlamaqui and Andy Dong. 2015. The Use and Misuse of the Concept of Affordance. In *Design Computing and Cognition '14*, John S. Gero and Sean Hanna (Eds.). Springer International Publishing, Cham, 295–311. https://doi.org/10.1007/978-3-319-14956-1_17

[6] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo. 2011. Using Object Affordances to Improve Object Recognition. *IEEE Transactions on Autonomous Mental Development* 3, 3 (2011), 207–215. https://doi.org/10.1109/TAMD.2011.2106782

[7] Nick Chater. 2009. Rational and mechanistic perspectives on reinforcement learning. *Cognition* 113, 3 (2009), 350–364. https://doi.org/10.1016/j.cognition.2008.06.014 Reinforcement learning and higher cognition.

[8] Anthony Chemero. 2003. An Outline of a Theory of Affordances. *Ecological Psychology* 15, 2 (2003), 181–195. https://doi.org/10.1207/S15326969ECO1502_5

[9] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. 2018. Learning to Act Properly: Predicting and Explaining Affordances from Images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 975–983. https://doi.org/10.1109/CVPR.2018.00108

[10] Paul Cisek. 2007. Cortical mechanisms of action selection: the affordance competition hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences* 362, 1485 (2007), 1585–1599. https://doi.org/10.1098/rstb.2007.2054

[11] Whitney G Cole, Gladys LY Chan, Beatrix Vereijken, and Karen E Adolph. 2013. Perceiving affordances for different motor skills. *Experimental brain research* 225, 3 (2013), 309–319. https://doi.org/10.1007/s00221-012-3328-9

[12] Alex Paul Conn. 1995. Time Affordances: The Time Factor in Diagnostic Usability Heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 186–193. https://doi.org/10.1145/223904.223928

[13] Brian Day, Elham Ebrahimi, Leah S Hartman, Christopher C Pagano, and Sabarish V Babu. 2017. Calibration to tool use during visually-guided reaching. *Acta psychologica* 181 (2017), 27–39. https://doi.org/10.1016/j.actpsy.2017.09.014

[14] Thanh-Toan Do, Anh Nguyen, and Ian Reid. 2018. AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 5882–5889. https://doi.org/10.1109/ICRA.2018.8460902

[15] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. 2016. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 512–519. https://doi.org/10.1109/ICRA.2016.7487173

[16] John M Franchak and Karen E Adolph. 2014. Gut estimates: Pregnant women adapt to changing possibilities for squeezing through doorways. *Attention, Perception, & Psychophysics* 76, 2 (2014), 460–472. https://doi.org/10.3758/s13414-013-0578-y

[17] John M. Franchak, Dina J. van der Zalm, and Karen E. Adolph. 2010. Learning by doing: Action performance facilitates affordance perception. *Vision Research* 50, 24 (2010), 2758–2765. https://doi.org/10.1016/j.visres.2010.09.019 Perception and Action: Part I.

[18] William W. Gaver. 1991. Technology Affordances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New Orleans, Louisiana, USA) *(CHI '91)*. Association for Computing Machinery, New York, NY, USA, 79–84. https://doi.org/10.1145/108844.108856

[19] William W. Gaver. 1991. Technology Affordances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New Orleans, Louisiana, USA) *(CHI '91)*. Association for Computing Machinery, New York, NY, USA, 79–84. https://doi.org/10.1145/108844.108856

[20] Samuel J. Gershman and Nathaniel D. Daw. 2017. Reinforcement Learning and Episodic Memory in Humans and Animals: An Integrative Framework. *Annual Review of Psychology* 68, 1 (2017), 101–128. https://doi.org/10.1146/annurev-psych-122414-033625

[21] Samuel J Gershman and Naoshige Uchida. 2019. Believing in dopamine. *Nature Reviews Neuroscience* 20, 11 (2019), 703–714. https://doi.org/10.1038/s41583-019-0220-7

[22] Eleanor J Gibson. 2000. Perceptual learning in development: Some basic concepts. *Ecological Psychology* 12, 4 (2000), 295–302.

[23] Eleanor Jack Gibson, Anne D Pick, et al. 2000. *An ecological approach to perceptual learning and development*. Oxford University Press, USA.

[24] James J Gibson. 1977. The theory of affordances. *Hilldale, USA* 1, 2 (1977), 67–82.

[25] James J Gibson. 2014. *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press. https://doi.org/10.4324/9781315740218

[26] James Jerome Gibson and Leonard Carmichael. 1966. *The senses considered as perceptual systems*. Vol. 2. Houghton Mifflin Boston.

[27] Helmut Grabner, Juergen Gall, and Luc Van Gool. 2011. What makes a chair a chair?. In *CVPR 2011*. 1529–1536. https://doi.org/10.1109/CVPR.2011.5995327

[28] Harry Heft. 1989. Affordances and the body: An intentional analysis of Gibson's ecological approach to visual perception. *Journal for the theory of social behaviour* 19, 1 (1989), 1–30. https://doi.org/10.1111/j.1468-5914.1989.tb00133.x

[29] Andrew Howes, Xiuli Chen, Aditya Acharya, and Richard L Lewis. 2018. Interaction as an emergent property of a Partially Observable Markov Decision Process. *Computational interaction* (2018), 287–310.

[30] Aleksi Hämäläinen, Karol Arndt, Ali Ghadirzadeh, and Ville Kyrki. 2019. Affordance Learning for End-to-End Visuomotor Robot Control. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1781–1788. https://doi.org/10.1109/IROS40897.2019.8968596

[31] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 11 (1998), 1254–1259. https://doi.org/10.1109/34.730558

[32] Jeff A. Johnson. 1995. A Comparison of User Interfaces for Panning on a Touch-Controlled Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 218–225. https://doi.org/10.1145/223904.223932

[33] Victor Kaptelinin. 2014. Affordances and design. *The interaction design foundation* (2014).

[34] Khimya Khetarpal, Zafarali Ahmed, Gheorghe Comanici, David Abel, and Doina Precup. 2020. What can I do here? A Theory of Affordances in Reinforcement Learning. In *International Conference on Machine Learning*. PMLR, 5243–5253.

[35] Hedvig Kjellström, Javier Romero, and Danica Kragić. 2011. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding* 115, 1 (2011), 81–90. https://doi.org/10.1016/j.cviu.2010.08.002

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. https://doi.org/10.1038/nature14539

[37] Weiyuan Liu. 2010. Natural user interface- next mainstream product user interface. In *2010 IEEE 11th International Conference on Computer-Aided Industrial Design Conceptual Design 1*, Vol. 1. 203–205. https://doi.org/10.1109/CAIDCD.2010.5681374

[38] Pedro Lopes, Patrik Jonell, and Patrick Baudisch. 2015. Affordance++: Allowing Objects to Communicate Dynamic Use. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2515–2524. https://doi.org/10.1145/2702123.2702128

[39] Alexandre Manoury, Sao Mai Nguyen, and Cédric Buche. 2019. Hierarchical Affordance Discovery Using Intrinsic Motivation. In *Proceedings of the 7th International Conference on Human-Agent Interaction* (Kyoto, Japan) *(HAI '19)*. Association for Computing Machinery, New York, NY, USA, 186–193. https://doi.org/10.1145/3349537.3351898

[40] Joanna McGrenere and Wayne Ho. 2000. Affordances: Clarifying and Evolving a Concept. In *Proceedings of the Graphics Interface 2000 Conference, May 15-17, 2000, Montr'eal, Qu'ebec, Canada*. 179–186. http://graphicsinterface.org/wp-content/uploads/gi2000-24.pdf

[41] Claire F. Michaels. 2000. Information, Perception, and Action: What Should Ecological Psychologists Learn From Milner and Goodale (1995)? *Ecological Psychology* 12, 3 (2000), 241–258. https://doi.org/10.1207/S15326969ECO1203_4

[42] Mike Mohageg, Rob Myers, Chris Marrin, Jim Kent, David Mott, and Paul Isaacs. 1996. A User Interface for Accessing 3D Content on the World Wide Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*

(Vancouver, British Columbia, Canada) *(CHI '96)*. Association for Computing Machinery, New York, NY, USA, 466–ff. https://doi.org/10.1145/238386.238608

[43] Austin Myers, Ching L. Teo, Cornelia Fermüller, and Yiannis Aloimonos. 2015. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 1374–1381. https://doi.org/10.1109/ICRA.2015.7139369

[44] Tushar Nagarajan and Kristen Grauman. 2020. Learning Affordance Landscapes for Interaction Exploration in 3D Environments. In *NeurIPS*.

[45] Anh Nguyen, Dimitrios Kanoulas, Darwin G. Caldwell, and Nikos G. Tsagarakis. 2017. Object-based affordances detection with Convolutional Neural Networks and dense Conditional Random Fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5908–5915. https://doi.org/10.1109/IROS.2017.8206484

[46] Jakob Nielsen. 1997. User Interface Design for the WWW. In *CHI '97 Extended Abstracts on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 140–141. https://doi.org/10.1145/1120212.1120312

[47] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.

[48] Donald A Norman. 1988. *The Psychology of Everyday Things*. New York : Basic Books.

[49] Donald A. Norman. 1999. Affordance, Conventions, and Design. *Interactions* 6, 3 (May 1999), 38–43. https://doi.org/10.1145/301153.301168

[50] Martin Oliver. 2005. The Problem with Affordance. *E-Learning and Digital Media* 2, 4 (2005), 402–413. https://doi.org/10.2304/elea.2005.2.4.402

[51] Lidia Oshlyansky, Harold Thimbleby, and Paul Cairns. 2004. Breaking Affordance: Culture as Context. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction* (Tampere, Finland) *(NordiCHI '04)*. Association for Computing Machinery, New York, NY, USA, 81–84. https://doi.org/10.1145/1028014.1028025

[52] Maxwell J. D. Ramstead, Samuel P. L. Veissière, and Laurence J. Kirmayer. 2016. Cultural Affordances: Scaffolding Local Worlds Through Shared Intentionality and Regimes of Attention. *Frontiers in Psychology* 7 (2016), 1090. https://doi.org/10.3389/fpsyg.2016.01090

[53] Edward S Reed. 1996. *Encountering the world: Toward an ecological psychology*. Oxford University Press.

[54] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[55] Eviatar Shafrir and Jafar Nabkel. 1994. Visual Access to Hyper-Information: Using Multiple Metaphors with Graphic Affordances. In *Conference Companion on Human Factors in Computing Systems* (Boston, Massachusetts, USA) *(CHI '94)*. Association for Computing Machinery, New York, NY, USA, 142. https://doi.org/10.1145/259963.260165

[56] David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. 2021. Reward is enough. *Artificial Intelligence* 299 (2021), 103535. https://doi.org/10.1016/j.artint.2021.103535

[57] Hyun Oh Song, Mario Fritz, Daniel Goehring, and Trevor Darrell. 2016. Learning to Detect Visual Grasp Affordance. *IEEE Transactions on Automation Science and Engineering* 13, 2 (2016), 798–809. https://doi.org/10.1109/TASE.2015.2396014

[58] Michael Stark, Philipp Lies, Michael Zillich, Jeremy Wyatt, and Bernt Schiele. 2008. Functional Object Class Detection Based on Learned Affordance Cues. In *Computer Vision Systems*, Antonios Gasteratos, Markus Vincze, and John K. Tsotsos (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 435–444.

[59] Thomas A. Stoffregen. 2000. Affordances and Events. *Ecological Psychology* 12, 1 (2000), 1–28. https://doi.org/10.1207/S15326969ECO1201_1

[60] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[61] Lucia Terrenghi, David Kirk, Abigail Sellen, and Shahram Izadi. 2007. Affordances for Manipulation of Physical versus Digital Media on Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1157–1166. https://doi.org/10.1145/1240624.1240799

[62] John Tiab and Kasper Hornbæk. 2016. Understanding Affordance, System State, and Feedback in Shape-Changing Buttons. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2752–2763. https://doi.org/10.1145/2858036.2858350

[63] Kashyap Todi, Gilles Bailly, Luis Leiva, and Antti Oulasvirta. 2021. Adapting User Interfaces with Model-Based Reinforcement Learning. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 573, 13 pages. https://doi.org/10.1145/3411764.3445497

[64] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5026–5033. https://doi.org/10.1109/IROS.2012.6386109

[65] Phil Turner. 2005. Affordance as context. *Interacting with computers* 17, 6 (2005), 787–800.

[66] Michael T Turvey. 1992. Affordances and Prospective Control: An Outline of the Ontology. *Ecological Psychology* 4, 3 (1992), 173–187. https://doi.org/10.1207/s15326969eco0403_3

[67] Michael T Turvey, Robert E Shaw, Edward S Reed, and William M Mace. 1981. Ecological laws of perceiving and acting: in reply to Fodor and Pylyshyn (1981). *Cognition* 9, 3 (June 1981), 237—304. https://doi.org/10.1016/0010-0277(81)90002-0

[68] Emre Ugur, Erol Sahin, and Erhan Öztop. 2009. Affordance learning from range data for multi-step planning.. In *EpiRob*. Citeseer.

[69] Alessandro Valli. 2008. The design of natural interaction. *Multimedia Tools and Applications* 38, 3 (2008), 295–305. https://doi.org/10.1007/s11042-007-0190-z

[70] Herke van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. 2016. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3928–3934. https://doi.org/10.1109/IROS.2016.7759578

[71] Leslie Carlson Vaughan. 1997. Understanding Movement. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '97)*. Association for Computing Machinery, New York, NY, USA, 548–549. https://doi.org/10.1145/258549.259028

[72] K.J. Vicente and J. Rasmussen. 1992. Ecological interface design: theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics* 22, 4 (1992), 589–606. https://doi.org/10.1109/21.156574

[73] Dhaval Vyas, Cristina M. Chisalita, and Gerrit C. van der Veer. 2006. Affordance in Interaction. In *Proceedings of the 13th Eurpoean Conference on Cognitive Ergonomics: Trust and Control in Complex Socio-Technical Systems* (Zurich, Switzerland) *(ECCE '06)*. Association for Computing Machinery, New York, NY, USA, 92–99. https://doi.org/10.1145/1274892.1274907

[74] Jane X Wang. 2021. Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences* 38 (2021), 90–95. https://doi.org/10.1016/j.cobeha.2021.01.002 Computational cognitive neuroscience.

[75] William H Warren. 1984. Perceiving affordances: visual guidance of stair climbing. *Journal of experimental psychology: Human perception and performance* 10, 5 (1984), 683. https://doi.org/10.1037//0096-1523.10.5.683

[76] Dylan E. Wittkower. 2016. Principles of anti-discriminatory design. In *2016 IEEE International Symposium on Ethics in Engineering, Science and Technology (ETHICS)*. 1–7. https://doi.org/10.1109/ETHICS.2016.7560055

[77] Shumin Zhai, Paul Milgram, and William Buxton. 1996. The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, British Columbia, Canada) *(CHI '96)*. Association for Computing Machinery, New York, NY, USA, 308–315. https://doi.org/10.1145/238386.238534