Interaction Interferences: Implications of Last-Instant System State Changes

Philippe Schmid, Sylvain Malacria, Andy Cockburn, Mathieu Nancel

Interaction Interferences: Implications of Last-Instant System State Changes

Philippe Schmid[1], **Sylvain Malacria**[1], **Andy Cockburn**[2], **Mathieu Nancel**[1]
[1] Inria, Univ. Lille, UMR 9189 - CRIStAL, Lille, France
[2] University of Canterbury, Christchurch, New Zealand
{philippe.schmid, sylvain.malacria, mathieu.nancel}@inria.fr
andrew.cockburn@canterbury.ac.nz

ABSTRACT
We study *interaction interferences*, situations where an unexpected change occurs in an interface immediately before the user performs an action, causing the corresponding input to be misinterpreted by the system. For example, a user tries to select an item in a list, but the list is automatically updated immediately before the click, causing the wrong item to be selected. First, we formally define *interaction interferences* and discuss their causes from behavioral and system-design perspectives. Then, we report the results of a survey examining users' perceptions of the frequency, frustration, and severity of interaction interferences. We also report a controlled experiment, based on state-of-the-art experimental protocols from neuroscience, that explores the minimum time interval, before clicking, below which participants could not refrain from completing their action. Finally, we discuss our findings and their implications for system design, paving the way for future work.

Author Keywords
interferences; reaction time; movement inhibition.

CCS Concepts
•**Human-centered computing** → **Human computer interaction (HCI);** *User interface management systems; Interaction design theory, concepts and paradigms;*

INTRODUCTION
Graphical User Interfaces (GUIs) are typically built with software toolkits that interpret user inputs and update the interface in response. These toolkits work as discrete state machines: if an event is detected within a state that has a matching transition, then the transition necessarily occurs. This paradigm works well in most cases, but it implicitly assumes that the human cognitive and motor processes are intentional and instantaneous, i.e. that users are aware of the system state, regardless of how recently it changed, and that users can react instantly to system state changes. However, just like computers, the human visuo-motor system takes time to perceive, interpret, and adapt a response to a change in the user interface.

This implicit assumption of instantaneous response results in a family of frustrating interface behaviors that we term *interaction interferences*: specific situations in which the user interface has changed, either visually or in the state of the application, immediately before the user carries out an atomic action such as clicking or typing a key, and too late for them to discontinue their action, causing the resulting input to misrepresent the user's intention or invoke the wrong process. An example of an interaction interference is when the user is about to select an email from a list, but the list is updated by incoming email right before the click, causing the wrong email to be selected.

In this paper, we investigate the phenomenon of *interaction interferences*. To our knowledge this phenomenon has not been described, studied, or considered in the HCI literature, perhaps because the resulting issues appear rare, inconsequential, or irresolvable — we intend to show that they are common, consequential, and we discuss designs that could mitigate their occurrence. First, we propose a formal definition of *interaction interferences*, along with a discussion of their causes from both human and system perspectives. Second, we report on the results of an online survey aimed at gathering examples of interaction interferences, along with the overall frustration they generate and the consequences they can have. Following the results of that survey, we report on a first exploration of "human refrain-ability" in selection tasks (i.e., how long before a click can the user no longer refrain from performing it), using a method inspired by state-of-the-art experiment protocols from neuroscience. We conclude by discussing how current interactive systems could handle this family of issues and better consider human latency in the user-event handling loop.

This paper makes the following contributions:

1. Proposes the first formal definition of *interaction interferences* as an HCI phenomenon.
2. Presents empirical results on the occurrence, frustration, and consequences of interaction interferences.
3. Adapts a state-of-the-art experimental protocol from neuroscience [30] to an ubiquitous HCI task – mouse pointing.
4. Empirically quantifies temporal aspects of the human ability to inhibit stimulus-response selection tasks in an HCI setting.

Authors' version
UIST '20, October 20–23, 2020, Virtual Event, USA
https://doi.org/10.1145/3379337.3415883

FRAMEWORK AND RELATED WORK

We formally define *interaction interferences* as situations that satisfy the following three conditions:

- the user is performing or about to complete an action,
- the state of the interface changes in a way that will affect the system's interpretation of the corresponding input,
- the change occurs in the moments prior to completion of the action, leaving insufficient time for the user to inhibit it.

The most critical aspect is that the system state change occurs in the instants before the user action, leaving insufficient time for the user to notice and/or prevent it. We are therefore interested in short-term phenomena, both on the user and on the system sides. The actions involved in interaction interferences are typically atomic inputs like taps or keypresses, or short-term chunks of movements such as finishing to type a character n-gram or performing a short gesture.

The change in the interface may be visibly obvious (e.g., a popup-window appearing under the cursor) or it may be more subtle, such as a change in window focus between two preexisiting windows. The consequences of misinterpretation range from the relatively benign outcome of ignoring the user input (e.g. clicking on a now-empty area), to the potentially serious outcomes of executing an undesired and inappropriate action (e.g. typing text in the wrong window after an automated focus change). Serious consequences can be imagined and do occur: for instance, an unanticipated focus change resulting in typing a password into a clear-text field that is projected to an audience; and as another example, a time-consuming and unwanted reboot will occur if a return key press is misdirected to a system-update notification window. In any case, this (in-)ability to inhibit an input occurs before any choice can be made about how to deal with the semantics of the interfering event itself (e.g. whether and how to make an unwanted popup-window disappear after it was "successfully not clicked").

Interaction *interferences* are to be distinguished from interaction *errors*, which occur when users misunderstand the system state or produce an incorrect action. For example, if the system state changes, but the user fails to notice that change (perhaps due to inattention or due to a lack of salience in the display), then this is not a problem of interaction interference. The essential difference is that, with interaction interferences, the user could not have suppressed their action *even if they had noticed the change*.

Differences between interferences and interruptions

The focus on last-instant changes affecting the system's interpretation of a user input also distinguishes interferences from *interruptions*, which are defined around the impact on the user's mental workload and other higher-level aspects. Research on interruptions has typically investigated the consequences of events that require the user's attention to be redirected from a primary task towards a secondary task, forcing a task-switch. A typical example is receiving a phone call or notification while working on a computer [8].

Work on interruptions often focuses on issues relating to the cognitive demands of context switching, and with the challenges of losing and reattaining a thread in the user's workflow, as in this definition: *"we define IT interruptions as perceived, IT-based external events with a range of content that captures cognitive attention and breaks the continuity of an individual's primary task activities"* [2]—note however that there is little consensus on the definition of interruptions, see [2] for discussion and taxonomy. In the phone-call example, the interruption is the phone call itself, which prompts a transition of attention from the user's primary task towards the phone call, even though the user's primary task might have been conducted on a different device. An *interference*, on the other hand, would occur from that same event only if (1) the call happens when the user is about to complete a physical action with the phone, *e.g.* picking it up from a table, (2) too close to that action's completion for the user to prevent it, and (3) that action causes an unwanted response, such as hanging up instead of picking up the call.

The impact of interruptions on task performance has been studied in various ways [3, 5, 15, 33], generally by asking participants to complete a task as fast as possible while various interruptions are sent to the interface. As an example, in [1] participants were asked to edit a text document back to its original state as a primary task. They were occasionally instructed to answer an interrupting question, immediately and accurately, before being allowed to continue that task. Studies also investigated the optimal moment at which the user should be interrupted in order to minimize the cognitive cost of the interruption [1]. Others took into account neuro-scientific aspects of interruptions to better understand their impact on the user's mental state, including their focus and emotional states [4]

While distinct phenomena, interferences and interruptions cohabit in interactive systems. For example, an interference can lead to an interruption: if a user clicks an e-mail notification window that popped-up right above a weblink they were about to activate, it both triggers the wrong response (opening the corresponding e-mail) and forces the user to switch focus to correct it (possibly marking the e-mail as unread, switching back to the web browser, locating the weblink, and clicking it again). In this situation, the interference corresponds to clicking on the pop-up window, whereas the interruption corresponds to the temporary yet substantial break that impacts the user's activity. Of course interferences can occur without triggering interruptions (*e.g.* if the erroneous response is that nothing happens, and the user remains focused on what they initially wanted to achieve), and vice versa.

Interferences from the system's perspective

Interferences stem from simplistic transitions between interface states that interpret the same action differently. To our knowledge, all GUI widgets work the same way: if a widget is displayed and active, it is listening for a specific set of events to which it will react. If the system state can change other than as an immediate consequence of a user action, then situations can occur in which the state change temporally overlaps with the user's processing and execution of an action (Fig. 1-a).

Even if the interface designer is conscious of this potential problem, today's tools for designing and implementing GUIs provide little support for identifying and remedying the occurrence of these events. At best, the system supports undo

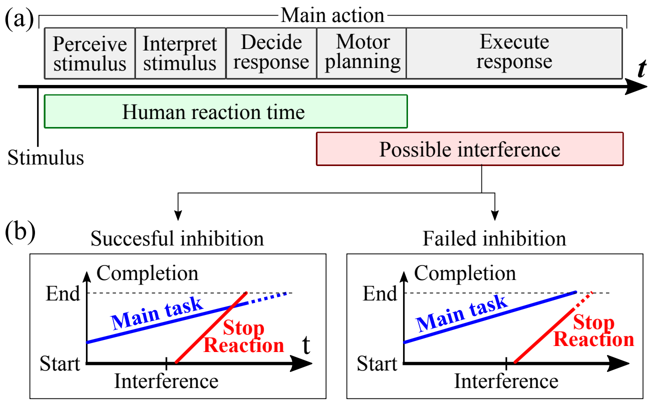

Figure 1. Top: simplified diagram of human reaction to a stimulus. Interferences can start before movement onset. Bottom: the "horse race" model of movement inhibition. The user successfully suppresses his action if the stop-process ends before the main action process.

functionalities, but the success of undo depends on the user noticing the misinterpretation, that the type of action is one that can be undone within the system, and that the action did not cause a switch between applications (undo seldom supports return to previous states across applications). Furthermore, for the system to identify the likely occurrence of an interaction interference it would need timing data regarding both the previous system state change and the latest user action, and this data would need to be accessible within and between applications – yet while accurate timing data is almost always available for the most recent user event, few systems maintain a history of state changes within and across applications.

Interferences from the user's perspective

Fig. 1 provides a conceptual summary of the low level human activities associated with interaction interferences. During normal interaction (top of the figure), users perceive a stimulus or feedback from the system, interpret that stimulus, decide on a response, plan their motor action, and execute their action. These low level components and their associated approximate timings have been examined since Card, Moran and Newell's seminal work on the 'human information processor' [9]. At some point in this chain of human activities (shown indicatively as the boundary between 'Decide response' and 'Motor planning' in the figure), it is too late for the user to modify their action, and the human is committed to completing their action even if new stimuli are received that would override the action given sufficient time for the human to process them.

Within the motor control literature the human capability to inhibit a movement in response to a 'stop signal' has been referred to as '*a "horse race" between two sets of processes, one that generates a response for the primary task and one that responds to the stop signal: If the primary-task process finishes before the stop-signal process, the response is executed; if the stop-signal process finishes before the primary-task process, the response is inhibited*' [19] (see Fig. 1-b).

Results from Henry and Harrison's [14] early experiments demonstrated that people were unable to inhibit their movements in response to a 'stop' signal, even when the stop signal was presented to the subjects *before* their movement had begun. In their experiments, participants began with their finger at their hip, and they were instructed to move their finger to the tip of a string in front of their shoulder as quickly as possible when a 'go' signal was given. The average response time (depicted as 'Human reaction time' in Figure 1 was 214 ms, and the average movement time ('Execute response' in the figure) was 199 ms. During some trials a 'stop' signal was played, and subjects were instructed to avoid moving their finger to the string in their event. In different experimental conditions, the 'stop' signal was played at four timing points: 110, 190, 270 and 350 ms after the 'go' signal. Results showed that subjects were only able to *begin* to slow their finger when the stop signal was presented at the earliest timing interval of 110 ms, and that subjects were unable to even initiate slowing their movement when the stop signal occurred 190 ms after the go signal, despite this signal being present *before* any movement had begun. Several studies have produced related findings (see [29] for a review, including studies of proficient typists who appear to be incapable of interrupting well-trained typing patterns such as 'the_' in response to a stop signal [19]).

The various components of reaction time (RT) to a stimulus (Fig. 1) have been studied thoroughly in the fields of psychology and neuroscience (see *e.g.* [10, 32]), but also early on in HCI [23] wherein RT has become a frequent measure of interaction technique performance (see *e.g.* [13, 17, 22, 25] for recent examples). Interaction interferences involve a specific type of reaction time, sometimes called "Stop-Signal Reaction Time" (SSRT), in which the stimulus calls for an *absence* of reaction, or for the inhibition of an already planned reaction. SSRTs have been studied in neuroscience and psychology but remain mostly nonexistent in HCI, besides early work on typing by Logan et al. [18] and Salthouse et al. [26, 27].

Research Agenda

We propose that interface designers would be better able to limit or mitigate interaction interferences if they had access to two forms of information: (a) from a system engineering perspective, the availability to the system of accurate information about the timing of user *and* system events; and (b) from a user perspective, knowledge about users' ability to react to last-instant stimuli.

In the remainder of this paper, we focus on the latter. We first report the results of a survey into user perceptions regarding the occurrence of interaction interferences in everyday use of interactive systems. We then describe a study examining the human capability to suppress selection actions, providing a quantitative foundation for future work on mitigating interaction interferences, as discussed at the end of the paper.

A SURVEY ANALYSIS OF INTERFERENCES

We first conducted an online survey to better assess the frequency, range, impact, and perception of interaction interferences on users.

Survey

We set up an online questionnaire,which gathered various demographics (such as age, gender, occupation, etc.), as well as questions relating to the respondent's use of interactive devices such as computers, tablets, smartphones, etc.

The questionnaire then defined the notion of "interaction interferences." In pilot surveys, we found that participants understood the phenomenon better and were more inspired by concrete examples than by a technical definition alone (discussed further below), therefore we also provided the following:

– *You are about to click a hyperlink on a webpage, or a menu button on a familiar software. Right before you click, a pop-up window appears under your cursor.*

– *You are about to select an element in a list, for instance an email or a WiFi network. Right before you click, the list is updated and new elements appear, displacing the one you were aiming for.*

– *You grab your phone from your desk. At the same time you receive an incoming call, but the device registers the movement as a "reject call" gesture and hangs up.*

– *You are typing text on a word processor. Another application suddenly takes the focus, e.g. to notify you about new updates, causing parts of the typed input to be sent to the wrong application.*

We expect that, in combination with a theoretical description, these examples facilitated the participants' understanding of interferences, and illustrated the variety of problems we were interested in – but not limited to. Participants were then asked if they thought they understood the topic, and if so to rate, in general, how *frequently* interferences happen to them, how *annoying* they are, and the *severity* of their consequences.

Participants were then invited to describe individual examples in detail. For each, we asked the context in which it happened, the possible consequences it had at the time, and the steps they took to alleviate them, if any. They were also asked, for each example, to rate its specific *frequency*, *annoyance*, and the *severity* of its consequences. Participants could provide more than one example if they wished, and could come back to the survey at a later time if more examples came to mind.

For the numeric responses, 5-point scales were used to assess *frequency* (from "Once a month or less" to "More than once a day"), *annoyance* (from "Not at all" to "Extremely"), and *severity* of consequences (from "None at all" to "Very severe"). The labels for all scales are shown in Fig. 2, and the entire questionnaire is provided as appendix.

The survey was distributed via university and science organization mailing lists, mostly computer science. It was deployed for a month, and we sent one reminder email 15 days after its start to the participants who had already answered, to remind them that they could also offer new examples. We used email addresses to identify returning participants, but kept the addresses separately from their answers and used a hash as identifier in the response database, in accordance with local ethics board recommendations. The email addresses of participants who said they did not want to be re-contacted were erased from our database after the 1-month collection period.

Participants

We received 41 survey responses, of which 15 indicated that the participants had misjudged the nature of the interaction issues we were interested in (e.g., citing examples of making the wrong gesture), and 1 that did not provide any example that would confirm a correct understanding of what interaction interferences are.

We gathered 30 relevant examples of interaction interferences from the 25 remaining respondents (8 females, 16 males, 1 non-binary, ages 20 to 55), mostly from Europe (17) and Canada (6). A majority were students (13, either bachelor, master, or PhD) or researchers (7). Ten participants related their occupation to computer science, the rest to other fields or did not specify a domain. All participants were daily computer users, all but one were daily smartphone users, five were daily tablet users, and only one reported using a gaming console daily. All responded that they understood the type of issue in which we are interested.

Overall impressions

When asked about interferences overall, before entering specific examples, participants reported frequencies ranging from daily to less than once a month (Fig. 2-a) with a majority (9) answering "Once every 2-3 days or less". A majority of participants (16) reported that it felt "Very-" or "Extremely annoying" overall (Fig. 2-b), and no-one answered "Not annoying at all." Finally, the general consequences of interferences were mostly felt to be "Minor" (Fig. 2-c), but six participants ranked them as "Moderate", one "Severe", and one "Very severe" overall.

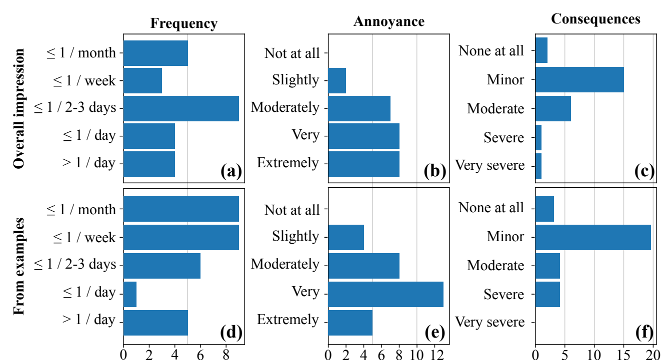

Figure 2. Participants assessments of frequency (left), annoyance (middle), and severity of consequences (right), for interferences overall (top) and in the provided examples (bottom).

Examples of interferences

Two coders analysed the full set of answers and separately (a) rated whether each example fit the topic, and (b) grouped the examples into meaningful categories, focusing on the type of interference and context of use. They then compared their categories, which revealed only minor differences in terminology that were adjusted to obtain a definitive classification.

Overall, most of the examples occurred on mobile phones (17) and on computers (9), the remaining four being spread between Tablets, TVs, and examples involving multiple devices. There was no evidence that assessments of Frequency, Annoyance, and Severity differed across Devices. Frequencies of individual examples ranged from daily to less than once a month (Fig. 2-d), with a drop at "Once a day or less" that we

Category	Typical examples (paraphrased from answers)	Phenomenon	Cause
Update (19)	*"I wanted to click a link on a loading webpage. When I clicked, the notification about cookies & personal data appeared."*	Target covered	Delay
	"When I try to interact [with an image list on Instagram] the app suddenly gets refreshed and loads new images. So, I end up viewing or liking a different image [than intended]."	Target displaced	External input
	"Word suggestion above a [gesture] keyboard that changes just when I tap it."	Target changed	User input
Focus switch (6)	*"Trying to unblock my phone, but in that exact moment a call [...] came in. The movement to unblock the phone is very similar to the one [...] to reject calls. So the call was rejected."*	Input misinterpreted	External input
	"As I was coding on Sublime Text, the popup (to buy the software) appears and the text [input] is not registered."	Input ignored	User input
	"typing longer text [...] and something was installing in the background. The installation window popped up [...] while I was about to hit enter - which [...] issued some sort of button on the installation window and had no way to go back and see what I had even done."	Input misinterpreted	As designed
Pop-up (6)	*"chatting [on Facebook Messenger], suddenly I got a notification from the facebook app. However, when I tried to dismiss the notification, it disappeared suddenly and ended up tapping on the call button - which was not my intention and [brought me to] an awkward situation"*	Target disappeared	Delay
	"Sometimes i prefer not to check a [...] message immediately [...] to avoid the 'seen' status and view it at a later time. [...] it has happened to me that I wanted to go 'back' but a messenger notification popped up at the same time and I ended up opening that message instead."	Target covered	External input

Table 1. Examples of intereferences provided by respondents, depending on their category, with corresponding phenomenon and cause.

ascribe to an overly fine granularity in the possible answers. Levels of annoyance (Fig. 2-e) ranged from "Slightly-" (4) to "Extremely annoying" (5) with a majority of "Very annoying" (13). Finally, consequences were mostly "Minor" (Fig. 2-f), with four examples in each of "Moderate" and "Severe", and three with "No consequence".

Participants' descriptions indicated that most interferences occurred when selecting a target (17 examples) or typing text (12), but there was no indication that the frequency, annoyance, and severity of the interferences were affected by these two tasks. The only remaining example involved performing a gesture with a smartphone.

Based on these reports, we defined three general categories of interferences (see Table 1 for examples):

UPDATE (19 examples): Happens when the currently-used – or targeted – interface element is updated, e.g. when a webpage is loading or a list is refreshed. Reported examples happen up to several times per day, mostly on phones (13).

FOCUS SWITCH (6): Happens when another process seizes the focus and therefore receives input events in priority, e.g. when a different application signifies that new updates are available. Reported examples happen up to once every 2-3 days, mostly on computers (5).

POP-UP (6): Happens when an element external to the current application appears in the foreground, e.g. a pop-up window or a notification. Reported examples happen up to once every 2-3 days. This is in essence a subset of FOCUS SWITCH, in which the change is visible and co-located with the user's finger or cursor.

Interferences can manifest in different ways. In the FOCUS SWITCH category, interferences caused inputs to be *ignored* or *misinterpreted* because they were redirected to a non-responsive area (i.e., dead space) of an unexpected interface. Among the UPDATE and POP-UP categories, the target of the user's ongoing action can be either *displaced* (e.g. a list is updated and the target item is moved around), *hidden* (typical with pop-ups), *changed* (e.g. word suggestions change while

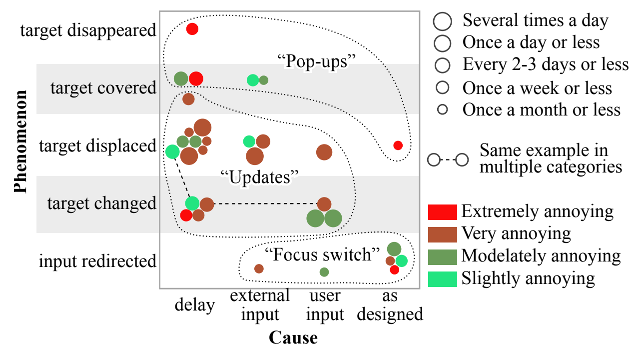

Figure 3. Main classifications of the participants' examples of interferences: by phenomenon (Y), trigger (X), and category (dotted areas). One dot is one example, the bigger the more frequent, and the redder the more annoying.

typing), or it can simply *disappear* (e.g. a notification with a fixed duration that disappears right before clicking it).

Finally, we identified the actual triggers of each of these interferences on a system level. In most of the reported examples (15), the change was triggered by *lag* in one form or another. For instance, a hyperlink suddenly moves down in a loading webpage, because images further up just finished loading and rendering, "pushing" everything else down; or a notification disappears right before being clicked because it had a predefined duration on screen. One fifth (6) were triggered by *external input*, e.g. when a new email is received and shifts the whole list down, or a call is received when the user is about to unblock her phone. In five examples the interferences were triggered by *user input*, e.g. typing one last character in a search field after the correct target was suggested, making the current suggestion change again for something unwanted. Finally, in five examples the interference came as simple outcomes of the interactive system's *design*. For instance, an application grabs focus to remind to the user that an update is available; or malicious attempts to coerce the user into clicking an ad, by making a pop-up window appear as soon as the cursor hovers over a known item of interest.

Discussion

Overall, respondents understood the definition of interference, and provided a majority (63%) of relevant examples. The "irrelevant" ones were mostly user errors such as using the wrong command in a normal situation, bad interface design, or annoying behaviors such as frequently asking for updates with no adverse consequence. These confusions likely pertain to an overly broad understanding of the studied phenomenon. Further studies need to take this into consideration, e.g. by narrowing the scope of the general definition of interferences.

Subjectively, interferences are characterized by *medium-to-high annoyance* and mostly *minor consequences* (Fig. 2) – even though a few examples were deemed severe – and independently of the device or task at hand. Interestingly, no participant ever answered "not annoying at all", either in general or for specific examples, suggesting that a main characteristic of interferences is the frustration they generate. Perceived frequencies, on the other hand, are uniformly distributed from monthly to several times per day.

We acknowledge that possible biases remain in our survey, as a first subjective exploration of the issue of interferences. Using examples, in addition to a more general description of the phenomenon, might have affected participants' recollection as well as the nature of the reported cases. When piloting the survey, we observed that examples were much more efficient at conveying the nature and breadth of the phenomenon than a neutral (and necessarily technical) definition, especially with non-computer scientists or professionals. People given only a technical description were more likely to find no example to report, or stay focused on one use-case, while examples gave them a clearer general idea of the type of issues we were interested in. We combined a general definition with a small set of examples to suggest to respondents the diversity of the issue without overly steering their answers. In fact, some or our examples were rarely reproduced by respondents (*e.g.* only one involved device gestures), and some user-provided examples didn't fit our own examples (*e.g.* with virtual keyboards and real-time suggestion updates).

Some participants misunderstood the nature of interferences, offering examples that involved *e.g.* their own errors (4 participants), faulty layout design (3), or unrelated technical issues like interface freezing (2). These misunderstandings did not predominantly appear in our pilot surveys, and will require specific reformulations in future studies.

Finally, our sample size is arguably small to fully qualify phenomena that could affect any and all interactive systems and users. Even taking these limitations into account, we believe that our results give a useful first picture of the type, frequency, and disturbance of interaction interferences in a representative set of use-cases. As for all exploratory work, our results are not meant to provide a final characterization of this newly described phenomenon, and will be strengthened by replication and generalization in future work.

Moving forward, the most frequently 'interfered' tasks in the gathered examples are target acquisitions (57%) and text input (40%). This is not surprising, as they represent the main tasks requiring user input. It is however informative regarding which interactions to focus on when dealing with interface interferences, as the psychomotor mechanisms involved likely differ. Previous work [18, 26, 27] on interrupted typing tasks consistently found that users are able to stop typing about 250-300 ms (1-3 characters) after being signaled to do so, and that the chances to suppress an incoming key stroke decrease as the delay between the first stroke of a word and the signal to stop increases. Less is known about pointing in the context of movement inhibition, so our next study will focus on pointing tasks and users' ability to refrain from completing them, *i.e.* to suppress clicking at the end of a pointing movement.

AN EXPERIMENT ON TARGET SELECTION INHIBITION

We conducted a controlled experiment to characterize users' ability to inhibit interaction input on a desktop computer. We focused on target acquisition and more precisely on the atomic confirmation component of an indirect pointing action, i.e. a mouse click. Inspired by state of the art research in neuroscience and experimental psychology, we designed a controlled study to assess the maximum delay between a change in the interface and a user's click, under which the user cannot interrupt the physical action of clicking. This phenomenon is sometimes referred to as *inhibiting* a planned motor action, here applied specifically to clicking at the end of a typical pointing movement.

We chose to run the study using mouse input on a desktop computer rather than a smartphone for two main reasons:

1. *motion coverage and precision* – all mouse movements and actions are easily logged by software, facilitating analysis of motion characteristics such as velocity, acceleration, clicks, and their timing, and of which might be influenced by the onset of a stop signal. In contrast, touch interactions are likely to be initiated off the surface of the device, and monitoring details of such actions would be complex.

2. *platform simplicity* – while monitoring off-surface movements is feasible, ambiguities arise in discriminating intentional movements from unintentional ones; these problems are much less acute when input is provided through a self-stabilising device, such as a mouse.

Task and Analysis Overview

The experiment consisted of series of 1-dimensional mouse pointing tasks in which participants were required to move the cursor horizontally to click on a green target. However, in a controlled percentage of randomly selected tasks ("*stop trials*") the target turned red (the "*stop signal*") at some controlled point in time, and when that happened participants were to refrain from clicking it. Specific instructions and various control mechanisms were used to ensure that their behavior remained similar to normal clicking actions, detailed below.

Whether participants managed to *not* click when the target turned red, and how it affected performance and subjective experience, was analyzed to assess their capacity to *inhibit* selection as a function of the time they had before clicking ("SSCD"). Since we cannot know when participants *would* have clicked in trials where they successfully inhibited their click, we estimated effective SSCD from the gathered data.

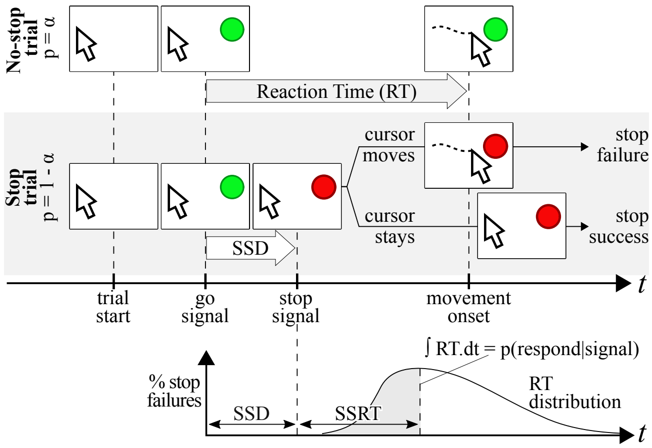

Figure 4. The stop-signal paradigm applied to movement *onset* [20]. A majority (prob=α) of normal pointing tasks are performed following a 'go-signal' (green). In the reminder (p=$1-\alpha$), a 'stop-signal' (red) is displayed shortly after the go-signal to instruct the participant *not* to move. Statistical analysis of reaction time and successful movement inhibition will reveal the go-stop delay for which participants fail 50% of the time.

The following subsections detail our experiment protocol, including methodology, specific setup, and the assumptions and steps taken to analyze the data.

Adapting existing methodology

The inhibition of pointing movement onset has been studied for decades, see e.g. [18, 19, 30, 31], although not in the context of interaction with computing systems. The "stop-signal" or "countermanding" paradigm [19, 30] (see Fig. 4) is the current standard of experimental protocols to characterize the maximum delay under which a user cannot refrain from initiating a movement after being instructed to perform it. Participants first practice a task aimed at measuring their reaction time (RT) to a visual stimulus ("*go signal*"). Once this baseline RT is measured, the experimenter informs participants that in the remainder of the experiment, a "*stop signal*" might randomly occur shortly after the go signal, and that they should refrain from completing the action if the stop signal occurs. Trials with a stop-signal are called "*stop trials*", and "*no-stop trials*" otherwise. The main independent variable is the Stop-Signal Delay (SSD), i.e., the time between the go and the stop signals. The probability that a trial will contain a stop-signal is usually kept constant, e.g., around 25% [30]. *Stop success* and *stop failure* respectively describe whether the participant managed to inhibit their response within a stop-trial. The outcome of these studies is traditionally a Stop Signal Reaction Time (SSRT) that corresponds to the SSD under which the probability to fail to inhibit the movement onset, in the presence of a stop signal, exceeds a predefined threshold, typically 50% (see Fig. 4). Note that the stop-signal paradigm is close but distinct from the "go/no-go" paradigm [12], in which the stimulus is either 'go' (react) or 'no-go' (do not react) but does not change afterwards, in that they recruit different neural dynamics [24].

In studying interaction interferences, we are most interested in the user's ability to suppress a *confirmation action* that selects an item once an aiming movement has been initiated. In contrast, prior psychology work has primarily focused on subjects' ability to suppress *movement onset*. To account for the different foci of interest between our study and that of prior work (i.e., movement termination versus movement onset) we made certain adaptations to the existing methodology, described below.

Hypothetically, a variety of events in the pointing action might serve as the stimulus to the user to begin their terminating selection action (the click/tap). For example, crossing the edge boundary of the target might be considered to serve this purpose for the sake of experimental analysis. However, doing so could misrepresent performance for a variety of reasons:

1. it would likely produce very small values for SSD before the user clicks, which is discouraged in practice [30],
2. it would fail to capture or describe stop-failures in which the decision to click, or its motor planning, occur before the cursor enters the target,
3. and overshooting could cause false starts of the SSD.

To avoid these and related limitations, we devised a modified countermanding protocol to control the delay between the stop signal and the click planned by the user, later referred to as Stop-Signal-to-Click Delay (SSCD, see Fig. 5). The stop signal must precede the click, so we need to be able to estimate the duration of the selection action $\widehat{\text{MT}}$ at the start of the trial. The time at which the stop signal will be displayed is then $T_{stop} = T_0 + \text{SSD} = T_0 + \widehat{\text{MT}} - \text{SSCD}$, where T_0 is the start of the movement.

The value of the predicted $\widehat{\text{MT}}$ will first be estimated using a preliminary set of trials without stop signals, then updated throughout the study to account for learning and fatigue effects, as well as effects from the task itself. We expect that several factors will affect the estimation of target acquisition time on a trial-by-trial basis. First, distributions of MT for pointing actions are notoriously skewed right, so we will follow previous recommendations [28] and use the geometric mean to estimate a central tendency. Second, target acquisition time is affected by typical Fitts study parameters such as distance (D) and target size (W), but also direction (DIR) [16], possibly differently for each participant (P). We therefore need to estimate $\widehat{\text{MT}}$ separately for different conditions: $\widehat{\text{MT}} = \text{E}[\text{MT}|\text{P},\text{D},\text{W},\text{DIR}]$. Third, it has been observed in prior work on countermanding tasks that participants alter their behavior once the stop trials start occurring. This phenomenon has been well documented and studied, and recommendations exist to minimize it [30]. It should be expected that participants' behavior will change nonetheless, and therefore we will constantly update $\widehat{\text{MT}}$ throughout the experiment. We found in pilot tests that geometric means of the last 3 selections of a given condition (not counting stop-successes, after which participants were instructed to wait before clicking) provides usable estimations.

A documented adaptation behavior to stop-signal tasks is to pause before starting the movement in order to trigger a possible stop signal earlier from the click. In addition to instructing participants to refrain from doing so, we consider the start of a pointing movement T_0 as the first `mousemove` event in the submovement that brought the cursor to 5% of the distance D to the next target. This allows us to keep reliable $\widehat{\text{MT}}$ estimates even when participants alter their behavior in that fashion.

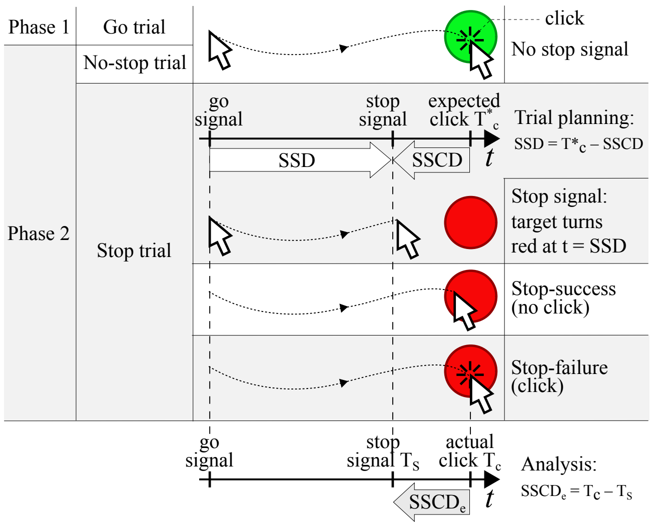

Figure 5. The updated countermanding protocol used in this experiment. The delay parameter SSCD is defined backward from the estimated click time. Effective SSCD_e is calculated post hoc using the actual click time.

To summarize:

- T_0 is the timestamp of the first event in the submovement that led the cursor to 5% of the task distance D,
- $\widehat{\text{MT}}$ is the geometric mean of the durations of the 3 most recent trials of the same condition, excluding stop-successes,
- the stop signal is displayed at $T_{stop} = T_0 + \widehat{\text{MT}} - \text{SSCD}$.

Experiment protocol

The participants first signed a consent form and filled a demographics questionnaire. They were then explained that the experiment concerned normal target acquisition using a normal computer mouse. They sat at a desk in front of the experiment's setup computer, and were allowed to adjust the height and inclination of the seat.

The measurements happened in two phases (Figure 5). First, we calibrated an initial estimation of the mean value $\widehat{\text{MT}}$ through a Fitts' 1D reciprocal pointing task (Fig. 5, top row). Participants were instructed to perform a sequence of target acquisition operations as quickly and accurately as possible.

For each trial, participants had to select a target colored in green, of a width W, and located at a distance D and in the opposite direction from the previously selected target. To select the target, participants had to position the cursor over the target and click on it. The experiment software moved to the next target only when the target was correctly selected. Targets were squares displayed left and right of the center of the screen.

We varied the distance (D: {10, 20} cm), target width (W: {1.5, 3} cm), and movement direction (DIR: {left, right}) of the tasks. Pointing difficulty was kept relatively low in hopes to lower the variability of selection time for any given condition. For each $2\text{D} \times 2\text{W} \times 2\text{DIR}$ combination, participants performed 16 selections in order to obtain a stable assessment of the participant's baseline pointing performance in each target condition.

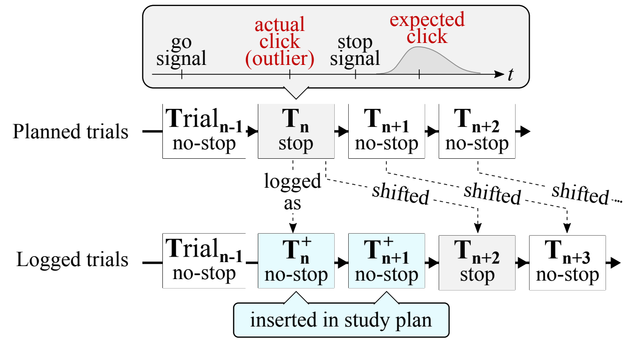

Figure 6. The trial replacement and shifting scheme that we apply when a user successfully acquires a target with a movement time $\text{MT} < \widehat{\text{MT}} - \text{SSCD}$ within a stop-trial. Two trials are added to maintain left-right alternation.

In the second phase of the experiment, *i.e.* after $16 \times 2 \times 2 \times 2$ successful selections, we introduced the participants to the concept of the stop-task, i.e. that during the rest of the study the target might turn red at times (stop signal). We instructed them to refrain from clicking the target when that happened. If they succeeded to do so, the target would turn back green after a delay of 1 second, after which they could click it to start the next trial. They were explained that the stop-signal would appear at random. In order to limit post-error slowing effects, participants were informed that a stop-trial would always be followed by a no-stop trial, regardless of errors. We set the probability of a stop-trial $p(stop)$ at 15%, but participants were not informed of that number.

We varied the Stop-Signal-to-Click Delay (SSCD: [100, 200, 300, 400, 500] ms, see Figure 5) with large increments to account for the display's refresh rate and the uncertainty of the $\widehat{\text{MT}}$ estimation. To account for the likely evolution of the participant's pointing behavior when exposed to stop tasks [30], we first used the $\widehat{\text{MT}}$ value obtained in Phase 1. As the participant progressed through Phase 2, we updated these estimates with the MTs measured in no-stop-trials and in failed stop-trials, always using the geometric mean of the last 3.

Since target acquisition time is variable even for trained users, it could happen that a participant be particularly fast in a given trial, to the point of clicking the target even before the stop signal is displayed ($\text{MT} < \widehat{\text{MT}} - \text{SSCD}$, see Figure 6-top). Such a trial can no longer be considered a stop-trial since the target never actually turned red. In these situations, we logged the trial as if it was a no-stop-trial (T_n^+ in Fig. 6), updated the value of $\widehat{\text{MT}}$, and "inserted" two additional trials to the experiment plan: one no-stop trial immediately afterwards in the opposite direction (T_{n+1}^+), then a clone of the intended stop trial (T_{n+2}^+). The intermediate no-stop trial is added in order to maintain the left-right alternation before repeating the intended stop-trial, with its intended direction. This method ensured that we obtained the intended number of stop-trials in our data, while having minimal effect on $p(stop)$ and on the total duration of the study. We report the frequency of these events in the Results.

In both phases we used the same values for D, W, and DIR. In Phase 2, we defined the number of stop-trials to be 6 for any $\text{D} \times \text{W} \times \text{DIR} \times \text{SSCD}$ condition, meaning that it had at least

$6 \times 100/15 = 40$ trials per condition (or more, depending on whether trials had to be inserted due to clicking the target before the stop signal was displayed). Each participant thus performed 2 D $\times$ 2 W $\times$ 16 = 64 trials in Phase 1, and at least (2 D $\times$ 2 W $\times$ 2 DIR) $\times$ 40 $\times$ 5 SSCD= 1600 trials in Phase 2. Combinations of D and W were counter-balanced across participants using a Latin square, DIR was always alternated, and SSCD values were randomized. As a result, participants performed 4 sessions of at least 400 trials (1 for each D $\times$ W combination) in Phase 2. These sessions were split in blocks of 16 trials (at least, depending on whether trials had to be inserted). Participants were invited to pause between blocks if needed. Finally, in order to prevent users from manipulating task difficulty by adjusting their pointing speed on purpose, the screen between blocks displayed a delay warning message in red if $\widehat{\text{MT}}$ in the last block was at least 300 ms longer than the $\widehat{\text{MT}}$ measured for this D $\times$ W $\times$ DIR condition in Phase 1. The experiment lasted between 30 and 40 minutes per participant.

Apparatus

The experiment was run on an 2017 13 inches Apple MacBook Pro running under macOS Mojave 10.14.6. Pointing was performed via a Logitech G502 Proteus Core mouse controller plugged via usb, set to 800 dpi and with the system pointing acceleration setting set to the maximum value. The experiment was displayed on the native 13 inches monitor with a resolution of 1440 $\times$ 900 pixels (130 ppi). The experimental software was implemented as a web application using JavaScript and the PixiJS graphics library[1] chosen for its relatively low latency.

Participants

16 participants took part in the study (4 F, 12 M, ages 23 to 47, mean 29). All participants had normal or corrected-to-normal vision, and were right-hand mouse users. Eight are primarily mouse users, two primarily touchpad users, and six use both on a regular basis. All participants reported using a computer five hours or more per day (up to 12).

Results

We used repeated-measures analyses of variance (ANOVA) to identify main statistical effects and their interactions. Errors were aggregated using arithmetic means and time measures using geometric means, as recommended in [28]. Participant was a random factor using the REML procedure of the SAS JMP package. Post-hoc Tukey tests were used for factors with more than two levels, t-tests otherwise.

We gathered 500 additional trials that were inserted according to the mechanism described in Fig. 6, following 250 stop-trials for which the participant successfully selected the target before the stop-signal. In what follows we removed all no-stop trials that took longer than 1.5 s (192 trials, 0.7%), and all trials for which the estimation of $\widehat{\text{MT}}$ was shorter than the intended SSCD (214 trials, 0.8% of all trials, but 5.6% of all stop-trials, mostly with SSCD= 500 ms).

Fitts Analysis of Go-Trials

We found no significant effect of Block on MT or selection errors, suggesting no specific training or fatigue effect. As

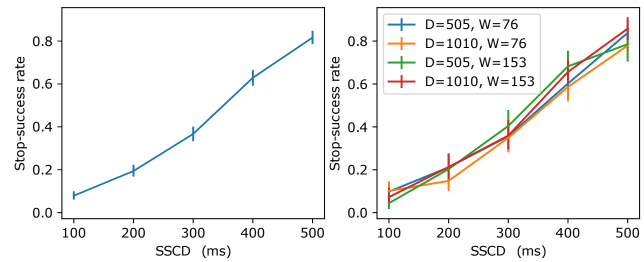

Figure 7. Effect of SSCD on stop-success rate, overall (left) and by D and W. Error bars are 95% CI.

expected, Fitts Index of Difficulty (ID$= \log_2(1 + \text{D}/\text{W})$) had a significant effect on MT ($F_{2,1155} = 1745$, $p < 0.0001$) and errors ($F_{2,1155} = 11.2$, $p < 0.0001$). There was no interaction effect. Fitts law showed a near-perfect fit on aggregated data ($R^2 = 0.99$).

Click Inhibition

In our pilot studies, participants commented that intentionally missing the target after it turned red might be easier or faster than suppressing the click altogether: even though a missed click is recorded, it does not constitute a stop-failure. This form of intentional missing may misrepresent users' ability to properly inhibit their action, or even be used as an intentional "winning" strategy by participants (see below). However, we detected only 112 occurrences of missed clicks occurring after a stop-trial (3.1% of all stop-trials). While we cannot know how many of those were intentional, it suggests that this phenomenon had at most a minor impact on the results.

We found no significant effect of target D, W, or Fitts' ID on stop-success rate. More interestingly, SSCD had a significant effect on stop-success rate ($F_{4,60} = 149.5$, $p < 0.0001$). Fig. 7 illustrates a sigmoid pattern similar to patterns found in previous work on movement inhibition, and locates the SSCD for which $p(\text{stop}|\text{signal}) = 50\%$ at about 350 ms before the click, and somewhat consistently across Fitts' IDs.

However, before drawing any conclusion we must inspect how well the values of the independent variable SSCD depict the delay that effectively happened between a stop signal and a click. Ideally, we want an *effective* SSCD (SSCD_e) for each trial, corresponding to the exact delay of the stop signal before the click. Retrieving SSCD_e for stop-failed trials is easy as it corresponds to MT$-$SSD (Fig. 5). However, the MT measured in stop-success trials includes the additional time for the target to turn back green and be clicked. In effect, in stop-success trials, there is no direct way to know when the user *would* have clicked, so SSCD_e can only be known for sure in stop-failure trials, i.e. when the participant could not refrain from clicking. We established earlier (Figure 7) that stop-success rate increases with SSCD, so estimating SSCD_e only from stop-failure trials is potentially much less accurate at low values of SSCD.

As can be expected, the $\widehat{\text{MT}}$ estimation mechanism did not perfectly predict pointing time. It had a remarkably central average error (0.04 ms), but a wide spread (SD=150.88 ms), as shown in Fig. 8-a. The calculation of SSCD_e is directly dependent on $\widehat{\text{MT}}$, and therefore their accuracies are equal for any given stop-trial:

[1] https://www.pixijs.com/

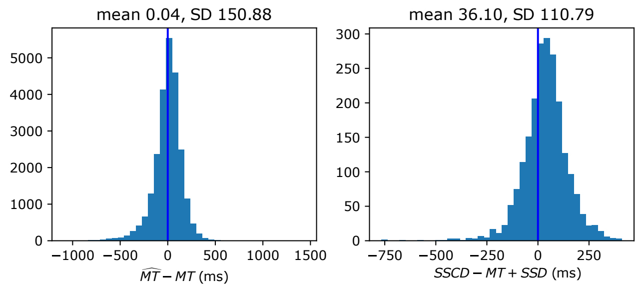

Figure 8. (a) Estimation errors for $\widehat{\text{MT}}$ for all trials. (b) Estimation errors for SSCD in stop-failure trials.

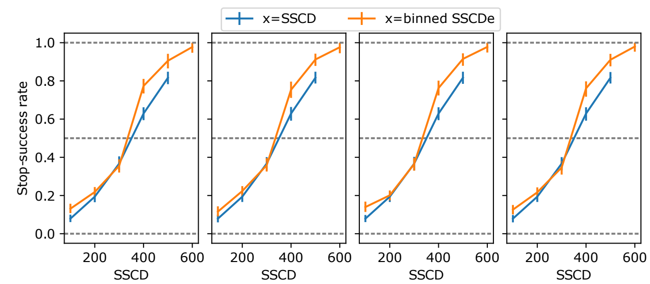

Figure 9. Stop-success rate as a function of SSCD (blue), and of four instances of re-sampled, re-binned SSCD_e(orange) to illustrate consistency. Error bars are 95% CI.

$$\begin{aligned} \text{SSCD}_\text{e} &= \text{MT} - \text{SSD} \\ &= \text{MT} - (\widehat{\text{MT}} - \text{SSCD}) \\ \text{SSCD} - \text{SSCD}_\text{e} &= \widehat{\text{MT}} - \text{MT} \end{aligned}$$

Fig. 8-b shows the distribution of SSCD errors = $\text{SSCD} - \text{SSCD}_\text{e}$, which varies from Fig. 8-a because it only depicts stop-failure trials. This criterion introduces more asymmetric errors (avg 36.1, SD 110.79). This is possibly because trials in which the participant moved more slowly, and was therefore further away from clicking the target when the stop-signal flared ($\text{SSCD}_\text{e} > \text{SSCD}$), are more likely to be stop-successes and de facto be excluded from the represented set.

In summary, to obtain a clear picture of the effects of (effective) SSCD_e on stop-success rate, we need to overcome two issues. First, we only know SSCD_e for stop-failures, but stop-success rate increases with SSCD, so the calculations of SSCD_e use fewer and fewer data points as SSCD increases (down to 19% of stop-failures for SSCD=500 ms). Second, SSCD_e is poorly estimated by SSCD, with standard deviations of errors wider than 100 ms, so we should expect that a portion of stop-successes labeled with a given SSCD belong in fact to a neighboring level. We address those issues in two steps.

1) Re-sampling missing data
First, we note that the distribution of $\widehat{\text{MT}}$ errors is remarkably consistent across D, W, DIR, ($p < 0.001$ for two one-sided equivalence t-tests with 20 ms tolerance between every value pair), and even across Participants ($p < .05$ for all but one pair), for all trials that are not stop-successes. Working under the assumption that participants could not predict stop-trials since they were presented in random order, we can hypothesize that the error estimation of $\widehat{\text{MT}}$ in successful stop-trials followed the same distribution.

We therefore impute the missing data, i.e. the time at which participants *would possibly have clicked in otherwise successful stop-trials*, by sampling from that distribution. More precisely, for each stop-success trial T_i with a given SSCD_i, we randomly select (with replacement) a trial T_j in which a click happened "normally", i.e. either a go-trial (excluding the initial Fitts calibration) or a stop-failure trial. We assign to T_i the SSCD_e that T_j *would have had*, had it been an unsuccessful stop-trial with the same target SSCD_i, as follows:

$$\text{SSCD}_{\text{e}i} = \text{MT}_j - \widehat{\text{MT}}_j + \text{SSCD}_i$$

Since we only impute for the stop-success trials, the impact of this process will increase with stop-success rate and therefore with SSCD.

2) Re-binning with SSCD_e
It remains that some SSCD_e values associated to a given SSCD might be off by 50 ms or more, *i.e.* by more than half the increment between SSCD levels, which casts doubt on the trends shown in Fig. 7. Having exact SSCD_e values for stop-failures, and imputed values for stop-successes, we can re-allocate each trial into the closest 'bin', which we will keep 100 ms wide and centered around {100, 200, ... 500}.

Fig. 9 shows four instances of this process, represented against our initial result using SSCD (the blue curves are the same as in Fig. 7). First, we can see very limited differences between the four instances, suggesting that this process yields relatively stable results. Second, we also observe that Stop-success rate after this process is slightly higher for the extreme SSCD values (100 ms, 400 ms, 500 ms), emphasizing the sigmoid pattern identified in previous studies on movement inhibition. Finally, this process has minimal impact on the SSCD at which a Stop-success rate of 50% is found, which remains stable around 350 ms in both cases.

Naturally, data imputation as we do in step (1) is not without risks, and the results in Fig. 9 should be taken with a pinch of salt. Our process relies on participants being unable to predict stop-trials, which seems confirmed by subjective feedback (see below). Since the distribution of $\widehat{\text{MT}}$ errors is essentially independent from task parameters (Fig. 7), this hypothesis allows us to reuse that distribution regardless of the condition.

Subjective Feedback & Implications for Future Studies
Our survey, pilot studies, and previous work raised concerns that the repetitive and attention-intensive experimental tasks could be frustrating or tiring. We therefore made the study as short as possible. At the end of the study, we asked participants to rate Phase 2 (the stop-signal part) for frustration on a scale ranging from 1 ("Not at all") to 7 ("Unbearable"). Most found the study somewhat frustrating (average 4.4, median 5, SD 1.4). No-one rated it Unbearable (7), but more than half the answers were 5 or above. We did not see evidence that fatigue or frustration increased through the experiment, but anyone conducting related future studies should carefully consider study duration and its implications for fatigue.

At the start of the experiment participants were asked to select items as they would do normally, and to not try to anticipate the stop-signal. Having completed the experiment, we asked participants whether they adopted certain strategies throughout the study to attempt to improve their stop-success rate. 12 out of 16 reported that they did at some point. Most of them reported having changed behavior starting at least 20% and up to halfway into the study, which did not appear in our analyses. Preferred strategies included:

- slowing down (6 participants), which was usually detected by our application and resulted in a specific delay message,
- decoupling the moving and clicking actions (5), which could also trigger delay warnings,
- accelerating (4) to click the target before the stop-signal (they were not informed of the adjusted $\widehat{\mathrm{MT}}$ estimation), which most participants noticed caused more stop-failures.
- "betting" (2) on the occurrences of stop-trials (e.g. "*every 5-8 squares will turn red*") or on the timing of the stop-signal, both of which were random.
- actively lowering the accuracy of the pointing movement (2), i.e. seeking "lucky misses" or systematically overshooting, which also triggered delay warnings.

All but three participants thought that their strategies had no impact on stop-success rate (1 'Yes', 2 unsure). Three also reported, post-experiment, that a few of their stop-successes were due to involuntary lucky misses and overshoots, which can happen in real use of an interactive system.

Overall, these observations are consistent with those of previous studies using the stop-signal paradigm [30]. In particular, it is difficult to avoid at least some frustration and experimentally-induced behaviors when operationalizing movement or reaction inhibition. While rare (15% probability), the stop-signals happened much more frequently in the study than might happen due to interaction interferences during real system use. However, decreasing the frequency of the stop-signal would have led to longer and possibly more frustrating study sessions. Despite that, the data that we gathered appears reasonably stable across participants and conditions, possibly because the stop-signals were truly unpredictable. This suggests that these behavior artifacts, while adding noise to the measures, might not strongly affect the observed phenomena, or at least to a manageable extent. Future work will attempt to improve this experiment protocol.

DISCUSSION AND DESIGN IMPLICATIONS

This work identifies interaction interferences as a frustrating usability issue with potentially serious consequences, and reveals various examples as well as important characteristics to address them. Our participants' ability to inhibit an action follows previous findings in neuro-psychology (*e.g.* an inhibition threshold above 200 ms) and will help design and tune mechanisms to alleviate or prevent interferences. While further work with larger populations will help confirm and generalize this threshold in different setups and tasks, or with less tech-savvy participants, we believe our findings can serve as a useful baseline.

Our results also hint that our participants' "refrain-ability" is not affected by target distance, size, or direction in typical pointing tasks. This is encouraging, because detecting interferences, or even predicting them, might not require detailed knowledge of the user's ongoing goals or target characteristics, which can be notoriously difficult to assess in rich interactive systems [11].

Two main approaches can be considered to alleviate the problem of interaction interferences: recognizing occurrences immediately after they happened, or identifying situations wherein they are likely to happen in anticipation. Both approaches rely on fine monitoring of input *and* display changes in real time. To be applicable, mitigation mechanisms will also need to happen fast enough: 350 ms is longer than most interactive systems' end-to-end latencies [21], but that period also needs to include the potential detection correction processes.

Occurrence identification

Given our findings, and previous work on human ability to inhibit an action, input events occurring after a state or visual change on their target can safely be considered interferences *a posteriori*, say 400 ms or less (~75% of stop-successes in our case) after the change. This requires the system to continually monitor the timing of interface state changes and user input. The system might also assign a confidence level to each candidate interference event, inversely proportional to the delay between the state change and the user input.

The response to candidate interference events could vary depending on that inferred confidence. A high-confidence event might trigger an automatic 'rewind' of the system's state, followed by replaying the original user event in its intended context, with adequate feedback indicating the correction. For low-confidence interferences, the system could explicitly prompt the user to confirm their intention before acting upon it, *e.g.* in the form of a dialog with choices 'Do what I meant' vs. 'Do nothing'. Such remediation strategies might create new problems, including false positives and tedious additional dialog boxes, and it may be that a simple 'global undo' option is preferable (even though implementing such a global, cross-applications undo is a significant technical challenge today). Careful evaluation should be conducted in order to assess the acceptability of this type of solution.

Anticipatory avoidance

The complementary approach is to design systems that can anticipate and avoid interferences, using knowledge of realistic human reaction thresholds and psycho-motor dynamics, and the ability to pause scheduled interface updates. Typically, a scheduled interface change that could affect the user – e.g. updating a widget on which the user appears currently active, or an area that seems likely to be hovered soon – could be delayed until the cursor leaves the area or stops moving. To avoid delaying updates indefinitely, visual feedback like the ones explored in Mnemonic Rendering [7] or the afterglow effects used in Phosphor [6] could be adapted to provide subtle indications that interface updates are being withheld.

Technical feasibility

From a software engineering perspective, constant input monitoring is already feasible to a high level of detail: for example, individual key-press patterns can inform on motor "chunking" dynamics, and cursor location can serve as a proxy for user interest [11]. Monitoring system state changes at such fine granularity can be arduous, however. In practice, a change in the interface is always preceded by a trigger signal of some sort, be it external (*e.g.* a new email is received) or internal (*e.g.* a timeout is reached), user-originated (*e.g.* a key is pressed), or system-originated (*e.g.* a download is completed). These signals are emitted and received at different levels (internal timer, OS events, socket to a specific application, etc.) and mostly not centralized beyond individual applications. Short of accessing *all* information exchanged within an interactive system, or requesting that every pending interface update be declared to a centralized validation entity, a practical workaround could be to observe the interface directly, and map visual changes to just-detected input events.

SUMMARY AND FUTURE WORK

We described the phenomenon of *interaction interferences*, which are changes of state in an interface right before a user action, and too late for the user to inhibit their action. We report the results of two studies investigating the nature, impact, and characteristics of interferences. A first survey revealed that while often resulting in mild consequences, interface interferences are relatively frequent and can be very annoying. From participants' examples, we were able to classify interface interferences in three main categories (update, focus switch, and pop-up) with distinct causes and varying behavior. We also learned that interferences seem to mostly occur during pointing and typing tasks. The latter being already well documented, we then conducted a controlled experiment aimed at quantifying users' ability to suppress an initiated pointing action, aiming to determine how long before a click can a user refrain from clicking on a target in response to a visual stop-signal. To that end, we adapted the *countermanding* experimental protocol [19] used in neuroscience and experimental psychology. Our results revealed a sigmoid-shaped relationship between pre-click stop-signal delay and the probability to refrain from clicking, that is consistent with previous findings for other types of motor actions. The delay for which participants had a 50% chance to successfully refrain from selection was found to be around 350 ms. Finally, we discuss the feasibility of detecting and mitigating interface interferences.

The primary aim of this work is to provide new information on the nature and influence of interferences in interactive systems. We hope that these findings will stimulate further work on system interventions, to help reduce the occurrence of interferences and mitigate their impact when they arise.

ACKNOWLEDGEMENTS

This work was partially supported by ANR (Causality, ANR-18-CE33-0010-01).

REFERENCES

[1] Piotr Adamczyk and Brian Bailey. 2004. If not now, when?: The effects of interruption at different moments within task execution. *Proc. Conf. Hum. Factors Comput. Syst* 6 (01 2004), 271–278. DOI: http://dx.doi.org/10.1145/985692.985727

[2] Shamel Addas and Alain Pinsonneault. 2015. The Many Faces of IT Interruptions: A Taxonomy and Preliminary Investigation of their Performance Effects. *Information Systems Journal* 25 (04 2015), 231–273. DOI: http://dx.doi.org/10.1111/isj.12064

[3] Brian Bailey, Joseph Konstan, and John Carlis. 2000. Measuring the effects of interruptions on task performance in the user interface, Vol. 2. 757 – 762 vol.2. DOI:http://dx.doi.org/10.1109/ICSMC.2000.885940

[4] Brian Bailey, Joseph Konstan, and John Carlis. 2004. The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface. *INTERACT* (04 2004).

[5] Brian P. Bailey and Joseph A. Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior* 22, 4 (2006), 685 – 708. DOI:http://dx.doi.org/https://doi.org/10.1016/j.chb.2005.12.009

[6] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. 2006. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 169–178. DOI: http://dx.doi.org/10.1145/1166253.1166280

[7] Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. 2006. Mnemonic Rendering: An Image-based Approach for Exposing Hidden Changes in Dynamic Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 159–168. DOI: http://dx.doi.org/10.1145/1166253.1166279

[8] Matthias Böhmer, Christian Lander, Sven Gehring, Duncan P. Brumby, and Antonio Krüger. 2014. Interrupted by a Phone Call: Exploring Designs for Lowering the Impact of Call Notifications for Smartphone Users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3045–3054. DOI: http://dx.doi.org/10.1145/2556288.2557066

[9] SK Card, TP Moran, and A Newell. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates.

[10] Ian Deary, David (Dave) Liewald, and Jack Nissan. 2010. A free, easy-to-use, computer-based simple and four-choice reaction time programme: The Deary-Liewald reaction time task. *Behavior research methods* 43 (11 2010), 258–68. DOI: http://dx.doi.org/10.3758/s13428-010-0024-1

[11] Abigail Evans and Jacob Wobbrock. 2012. Taming Wild Behavior: The Input Observer for Obtaining Text Entry and Mouse Pointing Measures from Everyday Computer Use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1947–1956. DOI: http://dx.doi.org/10.1145/2207676.2208338

[12] Pablo Gomez, Roger Ratcliff, and Manuel Perea. 2007. A model of the go/no-go task. *Journal of Experimental Psychology: General* 136, 3 (2007), 389.

[13] Jay Henderson, Sylvain Malacria, Mathieu Nancel, and Edward Lank. 2020. Investigating the Necessity of Delay in Marking Menu Invocation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Honolulu, United States, 13. DOI: http://dx.doi.org/10.1145/3313831.3376296

[14] Franklin M. Henry and John S. Harrison. 1961. Refractoriness of a Fast Movement. *Perceptual and Motor Skills* 13, 3 (1961), 351–354.

[15] J. G. Kreifeldt and M. E. Mccarthy. 1981. (1981). http://hdl.handle.net/2060/19820005848 oai:casi.ntrs.nasa.gov:19820005848.

[16] Byungjoo Lee and Hyunwoo Bang. 2013. A kinematic analysis of directional effects on mouse control. *Ergonomics* 56 (09 2013). DOI: http://dx.doi.org/10.1080/00140139.2013.835074

[17] Wanyu Liu, Julien Gori, Olivier Rioul, Michel Beaudouin-Lafon, and Yves Guiard. 2020. How Relevant is Hick's Law for HCI?. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. DOI: http://dx.doi.org/10.1145/3313831.3376878

[18] Gordon D. Logan. 1982. On the ability to inhibit complex movements: A stop-signal study of typewriting. *Journal of Experimental Psychology: Human Perception and Performance* 8, 6 (1982), 778–792. DOI: http://dx.doi.org/10.1037/0096-1523.8.6.778

[19] Gordon D. Logan and William B. Cowan. 1984. On the ability to inhibit thought and action: A theory of an act of control. *Psychological Review* 91, 3 (1984), 295–327. DOI:http://dx.doi.org/10.1037/0033-295X.91.3.295

[20] Giovanni Mirabella, Pierpaolo Pani, Martin Paré, and Stefano Ferraina. 2006. Inhibitory control of reaching movements in humans. *Experimental Brain Research* 174, 2 (Sep 2006), 240–255. DOI: http://dx.doi.org/10.1007/s00221-006-0456-0

[21] Mathieu Nancel, Daniel Vogel, Bruno De Araujo, Ricardo Jota, and Géry Casiez. 2016. Next-Point Prediction Metrics for Perceived Spatial Errors. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 271–285. DOI: http://dx.doi.org/10.1145/2984511.2984590

[22] Mathieu Nancel, Daniel Vogel, and Edward Lank. 2015. Clutching Is Not (Necessarily) the Enemy. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 4199–4202. DOI: http://dx.doi.org/10.1145/2702123.2702134

[23] Cindy Orosy-Fildes and Robert W. Allan. 1989. Psychology of Computer Use: XII. Videogame Play: Human Reaction Time to Visual Stimuli. *Perceptual and Motor Skills* 69, 1 (1989), 243–247. DOI: http://dx.doi.org/10.2466/pms.1989.69.1.243

[24] Liisa Raud, René Westerhausen, Niamh Dooley, and René J. Huster. 2020. Differences in unity: The go/no-go and stop signal tasks rely on different mechanisms. *NeuroImage* 210 (2020), 116582. DOI: http://dx.doi.org/https://doi.org/10.1016/j.neuroimage.2020.116582

[25] Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. 2013. Augmented Letters: Mnemonic Gesture-Based Shortcuts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2325–2328. DOI: http://dx.doi.org/10.1145/2470654.2481321

[26] Timothy Salthouse and John Saults. 1987. Multiple Spans in Transcription Typing. *The Journal of applied psychology* 72 (06 1987), 187–96. DOI: http://dx.doi.org/10.1037//0021-9010.72.2.187

[27] Timothy A Salthouse. 1986. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological bulletin* 99, 3 (1986), 303.

[28] Jeff Sauro and James Lewis. 2010. Average task times in usability tests: What to report? *Conference on Human Factors in Computing Systems* (01 2010), 2347–2350. DOI:http://dx.doi.org/10.1145/1753326.1753679

[29] Richard A. Schmidt and Timothy D. Lee. 2005. *Motor control and learning: a behavioral emphasis* (4th ed.). Human Kinetics, Champaign, IL.

[30] Frederick Verbruggen, Adam R Aron, Guido Ph Band, Christian Beste, Patrick G Bissett, Adam T Brockett, Joshua W Brown, Samuel R Chamberlain, Christopher D Chambers, Hans Colonius, and et al. 2019. A consensus guide to capturing the ability to inhibit actions and impulsive behaviors in the stop-signal task. *eLife* 8 (Apr 2019). DOI:http://dx.doi.org/10.7554/eLife.46323

[31] Frederick Verbruggen and Gordon D Logan. 2008. Response inhibition in the stop-signal paradigm. *Trends in Cognitive Sciences* 12, 11 (Nov 2008), 418–424. DOI: http://dx.doi.org/10.1016/j.tics.2008.07.005

[32] David L. Woods, John M. Wyma, E. William Yund, Timothy J. Herron, and Bruce Reed. 2015. Factors influencing the latency of simple reaction time. *Frontiers in Human Neuroscience* 9 (2015), 131. DOI: http://dx.doi.org/10.3389/fnhum.2015.00131

[33] Fred R. H. Zijlstra, Robert A. Roe, Anna B. Leonora, and Irene Krediet. 1999. Temporal factors in mental work: Effects of interrupted activities. *Journal of Occupational and Organizational Psychology* 72, 2 (1999), 163–185. DOI:http://dx.doi.org/10.1348/096317999166581