



HAL
open science

A Performance Review of Number Entry Interfaces

Patrick Oladimeji, Harold Thimbleby, Anna L. Cox

► **To cite this version:**

Patrick Oladimeji, Harold Thimbleby, Anna L. Cox. A Performance Review of Number Entry Interfaces. 14th International Conference on Human-Computer Interaction (INTERACT), Sep 2013, Cape Town, South Africa. pp.365-382, 10.1007/978-3-642-40483-2_26 . hal-01497448

HAL Id: hal-01497448

<https://inria.hal.science/hal-01497448>

Submitted on 28 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A performance review of number entry interfaces

Patrick Oladimeji¹, Harold Thimbleby¹ and Anna Cox²

¹ Future Interaction Technology Lab, Swansea University, SA2 8PP, UK
{p.oladimeji,h.thimbleby}@swansea.ac.uk

² UCL Interaction Center, University College London, WC1E 6BT, UK
anna.cox@ucl.ac.uk

Abstract. Number entry is ubiquitous and there are several ways a number entry interface can be designed. Until recently, research has been focused mainly on one type of number entry interface: the numeric keypad. Various factors such as the range of values, and the space available for the design allows for several alternative interfaces to be used for number entry. In the design of medical devices such as those used for controlled drug delivery, accurate and timely entry of numbers are required in order to reduce any risk of harm to patients. This paper reviews five number entry interface styles and reports the result of an experiment conducted to evaluate the performance differences of the interfaces based on numbers used in infusion therapy in a hospital. The result shows a significant effect of interface style on speed and accuracy

Keywords. Number entry interfaces, number entry error, user interface performance, safety critical devices.

1 Introduction

Number entry has been a part of human culture since humans learned to count and the use of numbers is evident in spoken languages around the world. Number entry interface design dates back as far as 2nd century B.C. as seen in the Salamis Tablet, an ancient Greek counting board [12, p.300]. Much later, starting in the mid 17th century, the advent of a series of mechanical calculators such as Pascal's Calculator and later, the Arithmometer, brought about a series of different design options for interacting with numbers and designing modern number entry user interfaces. Most of these interfaces are still in use today in interactive devices—although implemented in a variety of ways that account for technological advances both in software and hardware.

Tasks involving entering numbers are extremely common. For instance we enter or select numbers at the cash machine, we enter, select or modify numeric values on our microwave ovens to specify time, and we often change the volume on our music player. While performing any of these tasks, the user might be oblivious to the number entry aspect of the task, after all, you only wish to withdraw some money from the bank, warm up your food or increase the volume of music. Number entry is usually such a subtask in achieving a more primary goal that it is hardly noticed and very

often perceived as trivial.

The reader could probably think of more than one type of interface for performing these tasks. For instance, a cash machine might use a 12-key numeric keypad, a microwave might use a dial and the music player might use a slider. There are several ways a number entry interface might be designed. Despite dating back many years, until recently, research on number entry has failed to identify a classification of number entry interfaces, and a review of the performance of the different styles of interfaces that might be beneficial to designers of interactive systems.

The most common number entry interface is the numeric keypad. It is found on telephones, calculators and keyboards. Its popularity is not surprising since it provides a direct mapping between interaction input and output and allows sequential entry of the digits that constitute the intended number much like spoken western language. Several constraints, factors and requirements in certain contexts may however restrict the adoption of a type of number entry interface. Examples of such constraints might be space requirements or footprint of the device in question or the range and precision of values intended to be addressed in the host application.

In the design of medical devices, these constraints become more compelling and designers ought to be able to make number entry interface design decisions with a clear understanding of the strengths and weaknesses of an interface. The use of many medical devices involves entry of numeric data that represent drug doses, duration of therapy or frequency of therapy. According to a report in 2007, about 7000 medicine doses are administered each day in each hospital in England and Wales [1]. Some drugs have to be administered intravenously due to the treatment requirements of patients. This sometimes involves multiple intravenous drugs to be administered simultaneously [9]. Devices such as infusion pumps, used for controlled delivery of drugs in hospitals, require timely and accurate programming in order to avoid patient harm [14]. Setting up an infusion pump requires entering numbers that correspond to the rate of infusion, the volume to infuse and duration of the infusion. Many adverse incidents in hospitals have been as a result of number entry errors in programming infusion pumps [8, 21, 22].

This paper presents the results of an empirical evaluation of five different number entry user interfaces using a customisable high-fidelity prototype. Our aim was to explore the performance difference across these interfaces with the intent of providing a summary of tradeoffs involved in choosing to implement one of the styles of interface over another.

2 Related work

The majority of research in number entry interface performance has mainly obtained performance metric on a variety of configurations of the numeric keypad. Early research by Deininger [17] in the design of telephone keypads explored the performance differences of 16 layouts and the effects of keying behaviour of users on the keying entry speed. Deininger's experiment found that the entry speed was dependent on the participant's strategy for reading the numbers. Participants who memorized the

numbers before starting the keying sequence, performed significantly better than those who referred back to the number during entry.

Further experiments on the effect of keypad layout by Conrad and Hull [5] initially suggested that the 3 x 3 grid of telephone keypad layout with 1, 2, 3 at the top was more accurate than the calculator layout with 7, 8, 9 at the top. Marteniuk et al. [11] later found that performance differences between different keypad layouts based on the two popular telephone and calculator layouts were as a result of the placement of the zero key, suggesting that the zero key be placed below the other keys.

Other studies have explored the effects of button size, button spacing and auditory feedback on number entry speed and accuracy on touch screen devices. Schedlbauer found better performance with larger button sizes [18] and Bender et al. [2] found that auditory feedback lasting between 50ms and 400ms only had a significant effect on accuracy for small targets of 10mm x 10mm.

Recently, research in number entry interface design has focused on trying to understand number entry error and improving design to reduce the risk for error. Wiseman et al. [23] built a taxonomy of number entry error based on an experiment designed to induce errors while people entered numbers on the numeric keypad. They identified 21 types of number entry errors and organised them into a framework based on the position of their causes in Norman's Action Cycle [13].

Oladimeji [15] proposes that number entry interfaces can be classified into the following groups: (1.) *Serial digit entry* describes interfaces that enforce sequential specification of the digits that make up a number typically using a numeric keypad. (2.) *Independent digit entry* describes interfaces that allow specification of the digits that make up a number in any order. (3.) *Incremental number entry* describes interfaces that allow number entry through widgets (such as dials, knobs or buttons) that are used to increase or decrease the number.

Thimbleby and Cairns [20] have shown that the probability of ten-fold errors, which are a significant risk to patient safety [10, 6], can be significantly reduced. They propose a method for parsing the input stream from a serial entry interface so that syntax errors such as multiple decimal points are correctly detected and alerted to the user.

Oladimeji et al. [16] compared a serial interface to an incremental interface to explore their effects on error detection. They found that the interface style had a significant effect on the parts of the interface on which users placed visual attention. While the incremental interface encouraged visual attention on the display, the serial interface encouraged visual attention on the input keys. Consequently, the serial interface had more undetected errors than the incremental interface.

Number entry interfaces can often be implemented in a variety of ways. For instance an independent digit entry interface such as that found in Figure 1(d) can be implemented in as many as 28 different ways. Variations in implementation might include whether or not changes to the digits wrap around. This means incrementing the digit '9' turns it to a '0' or decrementing the digit '0' turns it to a '9'. Another variation determines whether the action of increasing or decreasing a digit affects the neighbouring digit to the left. In other words, the implementation performs some arithmetic on the entire number. By running simulated trials of users making keying

slips while entering numbers, Cauchi et al. [4] discovered that the differences in the implementation can have effects on the severity of error i.e., by how much an undetected error deviates from the intended number.

With a few exceptions, research in number entry has so far been based on the numeric keypad, usually testing the performance of different layouts. The serial interface offers very quick number entry and its performance scales well as the size of the number to be entered increases. However numbers used for tasks such as infusion therapy in hospitals are from a well-defined range with rules governing the allowed precision of numbers above certain thresholds. For instance, precision of numbers used for rate settings in a critical care unit might be two decimal places for numbers below 10 and only one decimal place for numbers that are between 10 and 100. In addition, from the analysis of interaction logs from infusion pumps used in 4 departments in a hospital, Wiseman et al. found that digits 0, 1, 2 and 5 were the most common digits used when programming infusion pumps [24]. These properties have made it feasible to use other number entry interfaces other than the serial interface in the design of medical devices.

For the rest of the paper, we present a detailed description of five interfaces, followed by an analytical evaluation of the speed of the interfaces using the Keystroke Level Model (KLM) [3] in section 4. We then present details of our user study in section 5, followed by the results and a discussion on the implications of the results. We conclude with some recommendations for number entry interface design.

3 Number entry interfaces

Based on Oladimeji's classification and implementations found on medical devices, we have implemented five exemplar interfaces: one instance of serial digit entry (*number pad*), two instances of independent digit entry (*up-down* and *five key*) and two instances of incremental entry (*chevrons* and *dial*). To evaluate the user interfaces, we built a prototype device with easily customisable keys.

Since previous researchers have explored the performance effects of different layout configurations of the serial interface, we evaluate only one instance of the serial interface in our study.

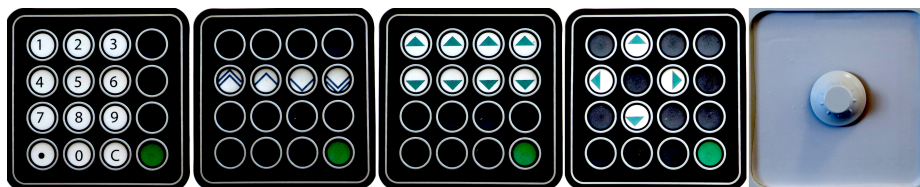


Fig. 1. The different configurations of interfaces used in our setup. From left to right (a) Number pad (b) Chevrons (c) Up-down (d) Five key (e) Dial

3.1 Number pad

This interface allowed number entry using a 12-key numeric keypad in the telephone style layout (see Figure 1(a)). It had a decimal point and a cancel key. The decimal point key appends at most one decimal point to the number on the display. The cancel key deletes the rightmost character on the display.

3.2 Chevrons

This interface utilised four buttons in a single row. The two buttons on the left (i.e., the upward facing chevron buttons) increased the number displayed, while the buttons on the right (i.e., the downward facing chevron buttons) decreased the number. Within each pair of buttons, the double chevron buttons caused a change ten times more than the single chevron buttons. This interface allowed two modes of interaction. The user could press the buttons or they could press and hold the buttons. Pressing the buttons changes the displayed number as specified above. Pressing and holding the buttons changes the displayed number at a rate dependent on the duration of hold. Users were expected to press and hold for faster changes to the number.

3.3 Up-down

This interface had eight buttons arranged in two rows and four columns. The top row buttons were used to increase the number and the bottom row buttons were used to reduce the number. Each column corresponded to a place value in the resulting number. For our set up, the rightmost column matched the hundredth place value and was used to increase or decrease the value by 0.01. This interface worked using the arithmetic configuration described by Cauchi et al. [4]. This means the effect of decreasing a digit from 0 or increasing a digit from 9 is carried over to the digit to the left.

3.4 Five key

This interface had four¹ buttons arranged in a navigation style: up, down, left and right. The left and right buttons moved a cursor on the screen which selected a place value in the number and the up and down buttons increased or decreased the selected digit. Like the *up-down* interface, it worked using the *arithmetic* configuration.

3.5 Dial

This was a 24-step dial interface with unrestricted continuous rotations in both clockwise and anti-clockwise directions. Users entered numbers on this interface by turning the dial left or right to decrease or increase the number. Quicker turns on the dials caused bigger changes to the number.

¹ Although the reader might see this as a four key interface, we refer to this interface as five key to be consistent with naming conventions in literature, for example see Cauchi et al. [4]

4 Analytical Evaluation

Prior to running our experiment, we analysed the performance of the key based interfaces and estimated task completions times using the Keystroke-Level Model (KLM) for user performance [3]. This is a model for predicting error free expert performance and as such, we use this prediction as the best-case performance achievable by users of these interfaces. Moreover, we expected that the relative ranking produced by the KLM analyses should be maintained in the results of the experiment.

4.1 Numbers used

We obtained log files of 60 syringe pumps from our affiliate University hospital in Swansea. The log files were completely anonymous and contained no personal information. We randomly selected 30 numbers used as rate and volume settings from the logs for our analysis. All the numbers had a decimal part and ranged from 0.26 to 83.3. A third of the numbers used had a precision of 2 decimal places.

4.2 Method

Based on simulations of the interfaces used in our experiment, we exhaustively explored the user interface model of each interface using the model discovery technique presented by Thimbleby and Gimblett [7, 19]. The user interface model discovery process produces a graph whose nodes represent the states in an interactive system and edges represent the user actions necessary to transition between the states.

To limit the number of states produced by the model discovery process, we limited the numbers addressable by the interfaces to a range covered by those used in the experiment. For each number entered in the experiment, we derived the optimal keying sequence for entering that number on the interface by searching for a shortest path from 0 to N, where N was the intended number. We ran a JavaScript implementation of the A* path finding algorithm, with cost functions that prioritised estimated time of execution over number of button clicks required to enter a number. We estimated the task completion time using standard KLM estimates for pointing and clicking [3]. As estimated by Card et al. [3], we used a value of 1100ms for the time (P) taken to point to a button and a value of 200ms for the time (K) taken to click a button. In our prediction of task time, we did not include the time (M) taken for mental preparation because we were interested in the execution time of each task. In our prediction, we do not include the initiation time, i.e., the time elapsed before the task is started or the commit time, i.e., the time taken to click the enter button to confirm the task. Using the keying sequence produced by this process, we derived an approximation for the performance time for executing the sequence.

Given that there are 24 steps in the rotary encoder used in the dial interface, to estimate the time T required to enter a given number N on this interface, we used the following expression:

$$T = \begin{cases} \frac{10t \times 100}{24} + \frac{(N-10) \times 10t}{24} & \text{if } N \geq 10 \\ \frac{N \times 100t}{24} & \text{otherwise} \end{cases}$$

Note that t is the time to perform one-step rotation on the rotary encoder. The value for t was set as 200ms. This is the value (K) taken to click a button.

4.3 Result

The Keystroke Level Model analyses produced the estimates displayed in Table 1. The predictions show that the *up-down* interface should be fastest with a slight performance edge over the *number pad* interface and the *chevrons* interface should be slowest. To validate these predictions, we designed and ran a user study.

Table 1. Approximation of the task times for the different interfaces KLM

Interfaces	Number pad	Chevrons	Up-down	Five key	Dial
Time (ms)	4875	9545	4600	6954	7855

5 Experiment

5.1 Design

The experiment was a two-way, mixed design. The within subjects independent variable was the type of number entry interface, and it had five levels: the five interfaces tested. The between subject independent variable was the instruction given to the participant: one group was instructed to enter the numbers as quickly as possible (the speed group) and the second group was instructed to enter the numbers as accurately as possible (the accurate group). We expected the speed group to exhibit more error due to time pressure. The order in which the interfaces were presented to the participants was randomized. The primary dependent variable was the speed of entry of correct numbers. Other dependent variables were the number of incorrect entries, the number of corrected errors.

5.2 Participants

There were 33 participants, 17 in the speed condition and 16 in the accurate condition. There were 22 females with 11 in the speed condition. Three participants were left-handed. The participants ranged in age from 18 - 43 with a mean age of 23.5 years (SD=4.86). The participants were undergraduate and postgraduate students in our University. Participants were randomly allocated to conditions.

Prior experience with interfaces. All participants were familiar with the *number pad* and reported using it on interfaces such as calculators, cash machines and telephones. Five participants (15%) were familiar with variations of the *chevrons* interface with experience using it in digital stop watches and alarm clocks, eight (24%) had prior

experience with the *up-down* interface on medical devices and games, nine (27%) had prior experience with the *five key* interface on remote controls and game controllers and 19 (58%) had prior experience with the *dial* interface on microwave ovens and temperature controls.

5.3 Apparatus

To run our experiments we built a high fidelity prototype unit consisting of a colour display with a resolution of 800 x 480 pixels encased in a box with a configurable front panel and a pole clamp for mounting the device. The front panel served as the input user interface with two variations. The first variation was connected to a 4 x 4 membrane keypad with insert pockets for configuring what is displayed on the keys. The second variation was a dial powered by a 24 step rotary encoder with tactile feedback on rotation and a selection switch that is activated by pushing the dial. Both front panels were controlled by Arduino boards but with different components attached to them. With these two front panels we were able to configure five types of number entry interfaces; we configured four using the membrane keypad and one using the dial. The different configurations are shown in Figure 1.

We used a pole to mount the prototype unit as shown in Figure 2 and connected the unit to a laptop computer (a 15 inch MacBook Pro). The laptop displayed the instruction for the next trial. Instructions were displayed as numbers in the middle of the laptop screen using a white font color on a black background and a font size of 20px. We used a total of 30 different numbers in the experiment. We used 10 numbers in a practice session and 20 for the experiment. Numbers used were those described in section 4.1. We implemented the software for the experiment in JavaScript and HTML and we logged keystrokes with corresponding time values and the numeric value after each keystroke for all the trials in the experiment.

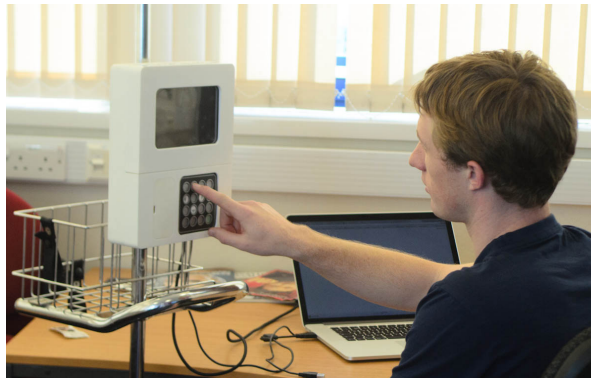


Fig. 2. The setup for the experiment showing the prototype mounted on a pole

5.4 Procedure

Each study session lasted about 45 minutes and we tested all participants individually. We informed each participant that the experiment involved entering numbers using five different number entry interfaces. Before the experiment started, the participants filled a short pre-experiment questionnaire containing demographic information about their age, gender, handedness and whether or not the participant was dyslexic.

The study itself was in five parts: one for each interface. Each part had a practice session followed by an experiment session. We randomly assigned participants to a speed or accuracy group. We instructed the speed group to enter the numbers in the instruction as quickly as possible and the accurate group to enter the numbers as accurately as possible. We randomised the order in which the users encountered the interface. A laptop computer displayed all study instructions. The instruction was a number displayed in the center of the computer screen. The next instruction was automatically displayed once the participant confirmed entry of the current trial. For the key-driven interfaces using the membrane keypad, participants confirmed entry by pressing a green button on the bottom right corner of the keypad. For the dial interface, participants pushed in the knob to confirm entry. A message signified the end of a session after a participant entered all the numbers required for that session.

Before starting each part of the experiment, the participants watched a video showing them how to use the interface they were about to test. They then had a training session where they tried out using the interface by entering 10 numbers. When they were confident with how the interface worked, they proceeded to the experiment.

The experiment session involved entering 20 numbers using the same interface they used in the training session. These numbers were different from those used in the training session. For each interface tested, each participant entered the same set of 20 numbers, although the order in which the numbers were encountered was randomised. Participants successively entered the numbers displayed in the instruction.

After the experiment, we conducted a short post-experiment semi-structured interview to find out prior experience with the interfaces and the participants relative preference for the interface styles. In return for their time, we gave the participants a gift voucher.

6 Results

6.1 Effect of instruction

Table 2 shows means and standard deviations for each group across the five interfaces. Although we expected more errors in the speed group, an ANOVA showed that group had no statistically significant effect on the participants' speed of entry $F(1.67, 51.69) = 0.56, p = 0.55$ or accuracy of entry $F(2.96, 91.84) = 0.62, p = 0.60$. As a result, we combined both groups for the rest of the analysis.

We next summarise the results of the speed and accuracy of the interfaces.

Table 2. Mean and standard deviation for the speed and accuracy of entry between the groups

	Entry Accuracy				Entry Speed			
	Speed Group		Accurate Group		Speed Group		Accurate Group	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Number pad	0.29	0.77	0.06	0.25	1906	423	2266	466
Chevrons	0.65	1.17	0.19	0.54	13355	3122	14471	2691
Up-down	0.65	1.69	0.38	0.62	3990	745	4783	1210
Five key	0	0	0	0	5231	908	5911	1213
Dial	0.94	1.92	0.44	0.81	9072	1211	10276	2024

For the rest of the results below, except the user interface preference statistic, we conducted post-hoc tests using multiple t-tests in order to find out which interfaces differed significantly from the others. For the user interface preference, we conducted post-hoc tests using multiple Wilcoxon Signed-Rank tests.

6.2 Speed of number entry

We separated the speed of entry of the interfaces into three constituent parts. The *initiation time* is the time elapsed between the display of the instruction and the participant’s first key press. The *execution time* is the time elapsed between the participant’s first key press and the last key press involved in setting the required number. The *commit time* is the time elapsed between the last key press in setting the required number and the key press for confirming the task.

Table 3. Mean and standard deviation for the initiation, execution and commit times

	Initiation		Execution		Commit		Total	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Number pad	1286	339	2080	474	730	223	4096	821
Chevrons	1535	433	13896	2931	1005	291	16436	3231
Up-down	1469	392	4374	1061	970	323	6813	1531
Five key	1463	354	5561	1105	1017	389	8040	1571
Dial	167	51	9655	1740	910	282	10732	1861

Initiation time. A one-way repeated measures ANOVA with Greenhouse-Geisser correction found a statistically significant effect of interface style on initiation time $F(3.31, 105.85) = 200.08, p < 0.0001$. Post-hoc analysis showed that the *dial* interface had significantly less initiation time than all other interfaces, and the *number pad* had significantly less initiation time than the *chevrons*, *five key* and *up-down* interfaces. The *dial* interface had the shortest initiation time and the *chevron* interface had the longest. Table 3 shows the mean initiation time for all interfaces.

Execution time. A one-way repeated measures ANOVA with Greenhouse-Geisser correction found a statistically significant effect of interface style on speed of entry

$F(1.69, 54.02) = 425.5$, $p < 0.001$. Post-hoc test showed that the speed of entry of all the interfaces tested were significantly different for all pairs at the 0.001 level. The *number pad* had the shortest execution time while the *chevrons* interface had the longest. Table 3 shows the mean execution time for all the interfaces.

Commit time. A one-way repeated measures ANOVA with Greenhouse-Geisser correction found a statistically significant effect of interface style on commit time $F(2.84, 90.83) = 24.35$, $p < 0.0001$. Post-hoc analysis showed that the *number pad* had significantly shorter commit time than all other interfaces and the *dial* had significantly shorter commit time than the chevrons interface. *five key* interface had the longest commit time. Table 3 shows the mean commit time for all interfaces.

6.3 Errors

We analysed both uncorrected errors and corrected errors. Uncorrected errors were trials for which the user transcribed and confirmed a wrong number whilst corrected errors were keying slips that the user recovered from before confirming the transcribed number.

Uncorrected errors. A one-way repeated measures ANOVA with Greenhouse-Geisser correction found a statistically significant effect of interface style on uncorrected error $F(2.98, 95.25) = 4.68$, $p = 0.004$. Post-hoc test showed that the *five key* interface had significantly less errors than the *dial* interface $t(32) = -3.24$, $p = 0.03$. The experiment elicited a total of 57 uncorrected errors, committed by 20 different participants. Only the *five key* interface was free of uncorrected errors. Table 4 shows the mean uncorrected errors on each interface.

Corrected errors. A one-way repeated measures ANOVA with Greenhouse-Geisser correction found a statistically significant effect of interface style on number of corrections $F(2.55, 81.61) = 63.17$, $p < 0.001$. Post-hoc test showed a significant difference for all pairs of interfaces at a 0.01 level, with the exception of the pair *up-down/five key*, which did not differ significantly. Table 4 shows the mean corrected errors on each interface. The experiment elicited a total of 833 corrected errors. The *dial* interface had the highest number of corrected errors while the *number pad* had the least.

6.4 User interface preference

At the end of the experiment, each user ranked the interfaces in order of preference. We assigned a score of 1 to the lowest preference and a score of 5 for the highest preference. There was a statistically significant difference in the preference rating for the user interfaces $\chi^2(4) = 73.8$, $p < 0.0001$. The *number pad* was most preferred interface while the *chevrons* interface was the least preferred. Table 5 shows the mean ranks for all interfaces. Post-hoc test using multiple Wilcoxon Signed-Ranks test

showed that the numeric keypad was preferred to all other interfaces, the up-down was preferred to chevrons, the up-down was preferred to five key, and the dial was preferred to chevrons.

Table 4. Mean and standard deviation for the corrected and uncorrected errors.

	Corrected Errors		Uncorrected errors	
	Mean	SD	Mean	SD
Number pad	0.88	1.08	0.18	0.58
Chevrons	6.48	3.86	0.42	0.94
Up down	2.73	2.74	0.52	1.28
Five key	3.09	2.38	0	0
Dial	12.06	5.53	0.70	1.24

Table 5. Mean ranks for interface preference.

	Number pad	Chevrons	Up-down	Five key	Dial
Mean Rank	4.81	1.69	3.50	2.44	2.56

7 Discussion

7.1 Relative preference of interfaces

Since all participants had prior experience using the *number pad*, it was not surprising that it was rated highest. This preference rating is also reflected in the speed exhibited by the interface. We were however surprised that the *dial* was not rated significantly worse than *up-down* and *five key* interfaces due to the number of corrected errors that occurred on the *dial*. One possible reason for this could be the significantly shorter initiation time for the *dial*. In addition, the simplicity of the interface which is based on increasing and decreasing the displayed number means the user has to do little thinking while executing the task. This was articulated by one participant, who said:

Dial was easier to turn the numbers. No need to move your hands from button to button.

7.2 Types of errors

The types of errors made during the study spanned across seven classes of errors previously reported in separate studies by Wiseman et al. [23] and Oladimeji et al. [16]. A summary of all errors is provided in Table 6.

The most common type of error was the Digit Added error. Thirteen different participants made this error on three different interfaces. Oladimeji et al. [16] reported this error in their experiment investigating the effect of interface style on error detection. While we classify this error as a member of the Digit Added error type in this

Table 6. Frequency of errors made during the experiment.

Error type	Total	Interfaces	Example
Digit added	31	chevrons, up-down, dial	4.05 for 4.5
Wrong digit	8	chevrons, up-down, dial	60.5 for 62.5
Missing decimal	3	number pad	249 for 24.9
Out by ten	3	number pad, up-down	1.11 for 11.1
Missing digit	1	number pad	6.5 for 62.5
Skipped	4	number pad, chevrons, up-down,	' for 62.5
No clear reason	9	chevrons, up-down, dial	56.7 for 3

paper, we believe the nature of the error makes it different from what the error type suggests. Syntactically, from the numerals that compose the intended number and the transcribed number, the error type suggests that an extra digit has been added to the number. This extra digit, in the case of errors in our experiment, is always zero. Semantically, however, this error appears to involve the inability to correctly understand the difference between the tenths and hundredths part of a number. It is possible that certain people mix up numbers matching the pattern. Indeed one participant transcribed 4.05 for 4.5 and in another trial transcribed 2.5 for 2.05. Over 50% of all unnoticed errors were of this form.

Despite featuring on the *chevrons*, *up-down* and *dial* interfaces, this error did not occur on the *number pad*. This could be because number entry on the *number pad* is a more direct transcription process of keying a sequence of digits that make up the intended number. Analysis of keystroke logs showed that an instance of this error occurred on the *five key* interface although it was noticed and corrected.

7.3 Difference in speed prediction and study results

We expected the absolute differences in the prediction of results and the actual study results since the participants that took part in the study were not expert users of all the interfaces. As a result, they could not match the optimal performance predicted by KLM. For the numbers used in the experiment, our prediction expected the *up-down* interface to be marginally faster than the *number pad*. Participants' familiarity with the *number pad* however meant that their performance was superior on this interface in comparison to the other interfaces. For the *number pad*, participants actually outperformed the expert model prediction and performed the task in less than half the predicted time. This could be due to the reduced target selection time exhibited in the experiment (554ms) for the *number pad* in contrast to the standard estimate used in the prediction model (1100ms).

The relative ranking in performance for the interfaces were preserved in the actual experimental data. For the *number pad*, *up-down* and *five key* interface, the observed task completion time for the experiment was less than the predicted time. On the other hand, the observed times for the *chevrons* and *dial* interfaces were higher than the predicted time. This difference in prediction could be due to the corrected error rates on the *chevrons* and *dial* interfaces which were higher than the corrected error rates

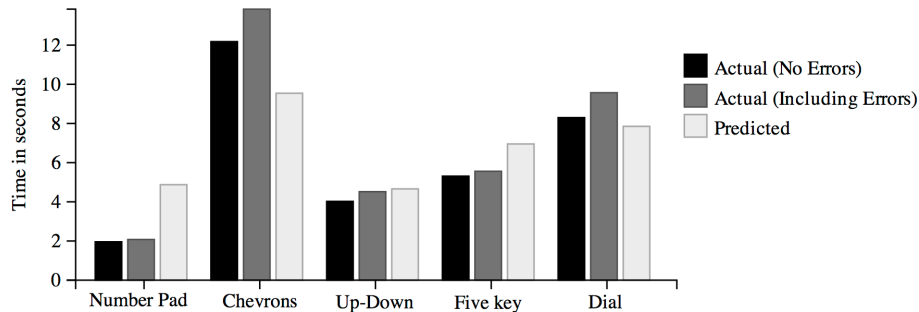


Fig. 3. A comparison of the actual and predicted performance for each interface

on the other interfaces. As a result, users spent a good portion of time correcting those errors which increased the task completion time and in some cases the frustration of users as evident in remarks during the post study interview:

With the dial and the chevrons, you don't really know when it switches to higher changes. I tend to stop just before I reach the value I want so that I can increase in one-step changes.

For chevrons and dial, really had to time it right and let it go at the right time otherwise could be annoying.

The corrected errors alone do not account for the deviation between the prediction and the recorded performance of the *chevrons* and *dial* interface. Another contributing factor is the strategy users develop to reduce the error rates. In the case of the *chevrons* interface, users employed discrete click interactions rather than the faster press and hold interaction, and for the *dial*, users made slower but more careful turns. Figure 3 shows that the cost of correcting an error is larger on both these interfaces in comparison to the others.

7.4 Effects of interface style on number perception

The different styles of interfaces had an effect on how numbers were perceived by the users. When users entered numbers on the serial interface (the *number pad*) or the independent digit interface (*up-down* or *five key*), they were more likely to think about the numbers as a sequence of digits without thinking much about the numeric quantity of the number as a whole. Whereas, using an incremental interface such as the *chevrons* or *dial*, participants were more likely to concentrate more on the number as a whole. One participant commented that:

For the number pad, up-down and the five key, I did not think of the number as a whole, just entered them digit by digit but for the chevron and dial, I had to understand the number.

7.5 Severity of errors committed

The types of errors committed were closely related to the interface used to enter the number and consequently the severity of error, i.e., the deviation of the intended number from the transcribed number or the ratio between the intended and the transcribed number. Theoretically, the *number pad* and the *up-down* interfaces have the potential for producing the largest deviations from the intended number based on keying slips. This is due to the possibility of missing decimal points and missing digits on the *number pad* and the possibility of wrong place value on the *up-down* interface. We defined three levels of error severity based on the errors committed in our experiment. Low error severity referred to those errors where the ratio between the intended number and the transcribed number is at most 2, medium error severity refers to when the ratio is at most 10 and high error severity refers to when the ratio is greater than 10. Table 7 shows a summary of all errors committed and their severity.

Table 7. The severity of undetected errors committed on each interface.

	Total Errors	Error Severity		
		Low	Medium	High
Number pad	4	0	3	1
Chevrons	11	10	1	0
Up-down	16	13	3	0
Five key	0	0	0	0
Dial	21	16	5	0

7.6 Incremental interfaces and varying number precision

As is typical of setting up some infusion devices used in hospital critical care, the set of numbers used for the study required that numbers below 10 were precise to two decimal places while numbers from 10 and above were precise to one decimal place. This factor meant that the display of incremental interfaces would only render numbers to the appropriate precision. As a result of this, button functions changed modes when the precision of numbers change on the display. For instance on the *chevrons* interface, when users change the value 9.99 to 10.0, the double chevron button changes meaning from ‘increase by a tenth’ to ‘increase by a unit’. Similarly on the *dial* interface, one turn on the dial changes meaning from ‘increase by a hundredth’ to ‘increase by a tenth’. We based our implementation of the *chevrons* interface on a medical device. It was also evident that some participant found the hold-down mode of the *chevrons* very difficult and challenging to predict. In this mode, the longer the buttons were held down, the larger the increments made to the number. This mode change was confusing for some users. Two participants expressed that:

For chevrons, the increments were very confusing. The same button did two jobs and the mode changes are confusing. Sudden changes were very confusing . . . for example you could go from 30 - 60 in a very short time span and then going back restarts the counter and climbs up rapidly. . .

Chevrons, seem to jump quite a lot, took too long to get to intended number. Same problem with dial. It goes in sequential order rather than control individual digits.

The feature of varying precision described in this section is a requirement in infusion pumps used in critical care and intensive therapy units where low dose settings are common. It remains a design challenge to create an incremental interface that supports this form of varying precision in a way that is not confusing to the user.

8 Conclusions

Number entry is ubiquitous and number entry interfaces are very common in interactive devices. Unlike text entry, it is difficult to build a predictive model that suggests corrections for number entry error because a number entry error is usually more ambiguous than text entry errors.

In order to approach more dependable design for number entry interfaces, particularly those in use in safety critical scenarios like in medical devices, we have explored the performance differences between five different interfaces. Our results show that the *number pad* is the fastest and *five key* is the most accurate. With the results of our study, we make suggestions to designers concerning the trade offs to expect when choosing between different styles of number entry interfaces as well as the likely errors on an interface style.

The *number pad* offers the fastest mode for number entry but it comes with the risk of high severity errors such as tenfold errors. These high severity errors are mostly caused by unintentional repeated digits or unintentional missing digits. Designers should guard against repeated digits caused by overly sensitive keys, e.g., those caused by key-bounce errors [8]. Repeated decimal points should be properly parsed and alerted to the user as an error, as suggested by Thimbleby and Cairns [20]. Keypad hardware should be rigorously tested to guard against missing digits which might be caused by keys that provide tactile feedback even when they have not been completely activated electronically.

Incremental interfaces, like the *dial* or *chevrons*, have the advantage that users place their visual attention on the display of the interface so they are more likely to notice and correct any errors. They are however much slower for entering numbers and can be frustrating to users due to high likelihood of overshooting and undershooting the target value. Designers should explicitly indicate to the user, what place value in the number is being edited by using a cursor. Designers should consider offering a more direct control of the rate of change of values in order to reduce the error rate and consequently, task time for these interfaces.

For the *up-down* designers should be aware that numbers might be shifted to a wrong place value e.g., 24.5 might be transcribed as 2.45 since the decimal point is not explicitly set and the number entry keys might not be perfectly aligned to the digits they affect. Designers should consider labeling the keys on this interface or using this interface style on touch screens where input and output are on the same media and the cost of changing key layout is minimal.

Prior to this study, the majority of studies on number entry interfaces have been

limited to the numeric keypad, mainly studying the effects of key layout on performance, both speed and accuracy. Our study covers a wider design space for number entry interfaces. The results provide information to designers about the tradeoffs involved in choosing one interface style over the other.

Acknowledgements. Funded as part of CHI+MED: Multidisciplinary Computer-Human Interaction research for design and safe use of interactive medical devices project EPSRC Grant Number EP/G059063/1. Carlos Monroy Aceves played a key role in the design of the prototype device.

References

1. National Patient Safety Agency: Safety in doses: medication safety incidents in the NHS. The fourth report from the Patient Safety Observatory, (2007)
2. Bender, G.T., Dean, D.G., and Dean, V. Touch Screen Performance as a Function of the Duration of Auditory Feedback and Target Size. *LIBERAL ARTS AND SCIENCES, WICHITA STATE UNIVERSITY*: 7, (1999).
3. Card, S.K., Moran, T.P., Newell, A.: The keystroke-level model for user performance time with interactive systems. *Commun. ACM* 23(7), 396–410, (1980).
4. Cauchi, A., Gimblett, A., Thimbleby, H., Curzon, P., and Masci, P. Safer “5-key” number entry user interfaces using Differential Formal Analysis. *Proceedings of HCI 2012 The 26th BCS Conference on Human Computer Interaction*, 29–38, (2012).
5. Conrad, R., Hull, A.J.: The preferred layout for numeral data-entry keysets. *Ergonomics* 11(2), 165–173, (1968).
6. Doherty, C. and Donnell, C.M. Tenfold Medication Errors: 5 Years’ Experience at a University-Affiliated Pediatric Hospital. *Pediatrics* 129(5), (2012).
7. Gimblett, A. and Thimbleby, H. User Interface Model Discovery: Towards a Generic Approach. *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems - EICS ’10*, 145 – 154, (2010).
8. ISMP: Double key bounce and double keying errors. *Medication Safety Alert! Acute Care Edition*, (2006).
9. Keohane, C.A., Hayes, J., Saniuk, C., Rothschild, J.M., Bates, D.W.: Intravenous medication safety and smart infusion systems: lessons learned and future opportunities. *Journal of infusion nursing*, 28(5), 321–328, (2005).
10. Lesar, T.S.: Tenfold medication dose prescribing errors. *The Annals of Pharmacotherapy* 36(12), 1833–1839, (2002).
11. Marteniuk, R.G., Ivens, C.J., Brown, B.E.: Are there task specific performance effects for differently configured numeric keypads? *Applied Ergonomics* 27(5), 321–325, (1996).
12. Menninger, K.: *Number Words and Number Symbols: A Cultural History of Numbers*. Dover Publications, (1992).
13. Norman, D.A.: *The design of everyday things*. BasicBooks, (2002).
14. NPSA: Reducing harm from omitted and delayed medicines in hospital. *Rapid Response Report NPSA/2010/RRR009*, (2010).
15. Oladimeji, P.: Towards safer number entry in interactive medical systems. In: *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. 329–332, (2012).

16. Oladimeji, P., Thimbleby, H., Cox, A.: Number entry interfaces and their effects on error detection. Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction. 178–185, (2011).
17. R.L. Deinger: Human factors engineering studies of the design and use of push button telephone sets. *Bell System Technical Journal* 39(4), 995–1012, (1960).
18. Schedlbauer, M.: Effects of key size and spacing on the completion time and accuracy of input tasks on soft keypads using trackball and touch input. Proceedings of the Human Factors and Ergonomics Society Annual Meeting 51(5), 429–433, (2007).
19. Thimbleby, H.: *PressOn: Principles of Interaction Programming*. The MIT Press, (2007).
20. Thimbleby, H., Cairns, P.: Reducing number entry errors: solving a widespread, serious problem. *Journal of the Royal Society Interface* 7(51), 1429–1439, (2010).
21. Vicente, K.J., Kada-Bekhaled, K., Hillel, G., Cassano, A., Orser, B.A.: Programming errors contribute to death from patient-controlled analgesia: case report and estimate of probability. *Canadian Journal of Anesthesia* 50(4), 328–332, (2003).
22. Westbrook, J.I., Rob, M.I., Woods, A., Parry, D.: Errors in the administration of intravenous medications in hospital and the role of correct procedures and nurse experience. *BMJ Quality & Safety*, (2011).
23. Wiseman, S., Cairns, P., Cox, A.: A taxonomy of number entry error. Proceedings of the 25th BCS Conference on Human-Computer Interaction, British Computer Society, 187–196, (2011).
24. Wiseman, S., Cox, A., Brumby, D.: Designing for the task: what numbers are really used in hospitals? In: Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems Extended Abstracts, 1733–1738, (2012).