Modeling

Chaklam Silpasuwanchai

Asian Institute of Technology

chaklam@ait.asia

Overview

1. Fitts' Law

2. Choice reaction time

3. Keystroke level model

Sources

- Mackenzie, Chapter 7, **Modeling Interaction**, Human Computer Interaction: An Empirical Research Perspective, 1st ed. (2013)

Model

- Model is a **simplification of reality**, allowing us to explore the phenomena, **without actually doing it**.
- Here we shall focus on three classic predictive models
  - **Fitts' Law**: predict selection time
  - **Choice reaction time**: predict reaction time given choices
  - **Keystroke level model**: predict task completion time

Fitts' Law

One of the most widely used models in HCI is Fitts' law (1954). Three primary usage are

- To see if the interaction technique follows Fitts' law
- To analyze design alternatives
- To use Fitts' index of performance (now throughput) as a dependent variable in a comparative evaluation

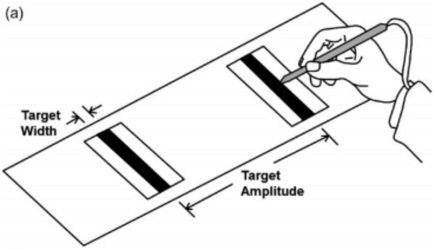
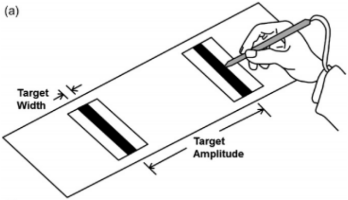


Figure: Source: Figure 7.14 (Mackenzie)

Fitts' Law



- Fitts proposed a variable quantifying movement task's difficulty - *ID, the index of difficulty*.

$$ID = log_2\left(\frac{A}{W} + 1\right)$$

- To use Fitts law to predict MT, it is a linear function of ID where a and b are obtained from experiments

$$MT = a + b * ID$$

- Fitts' index of performance, now called throughput (TP, in bits/s), is calculated as

$$TP = \left(\frac{ID}{MT}\right)$$

Improved Fitts' Law

- Crossman (1956) proposed new **effective target width** (W_e). W_e is computed from the **standard deviation** in the selection coordinates gathered over a sequence of trials for a particular A-W condition. If the selections are logged as x coordinates along the axis of approach to the target, then

$$W_e = 4.133 * SD_x$$

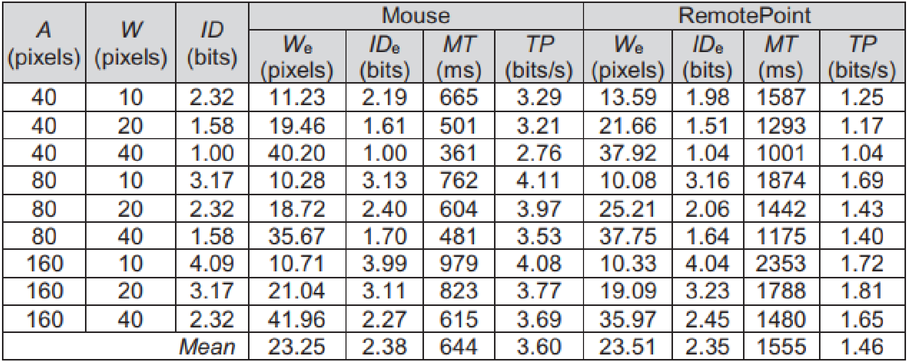
- hence

$$ID_e = log_2\left(\frac{A}{W_e} + 1\right)$$

- hence

$$TP = \left(\frac{ID_e}{MT}\right)$$

Example



A (pixels)	*W* (pixels)	*ID* (bits)	Mouse				RemotePoint			
			W_e (pixels)	ID_e (bits)	*MT* (ms)	*TP* (bits/s)	W_e (pixels)	ID_e (bits)	*MT* (ms)	*TP* (bits/s)
40	10	2.32	11.23	2.19	665	3.29	13.59	1.98	1587	1.25
40	20	1.58	19.46	1.61	501	3.21	21.66	1.51	1293	1.17
40	40	1.00	40.20	1.00	361	2.76	37.92	1.04	1001	1.04
80	10	3.17	10.28	3.13	762	4.11	10.08	3.16	1874	1.69
80	20	2.32	18.72	2.40	604	3.97	25.21	2.06	1442	1.43
80	40	1.58	35.67	1.70	481	3.53	37.75	1.64	1175	1.40
160	10	4.09	10.71	3.99	979	4.08	10.33	4.04	2353	1.72
160	20	3.17	21.04	3.11	823	3.77	19.09	3.23	1788	1.81
160	40	2.32	41.96	2.27	615	3.69	35.97	2.45	1480	1.65
		Mean	23.25	2.38	644	3.60	23.51	2.35	1555	1.46

Figure: Figure 7.16 (Mackenzie)

Example

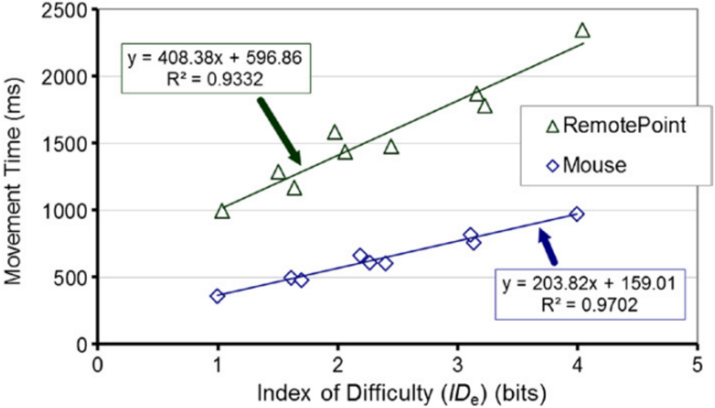


Figure: Figure 7.17 (Mackenzie)

Fitts' Law

- By using Fitts' law as objective/cost function, we can find optimal design alternatives - an area called **design optimization**

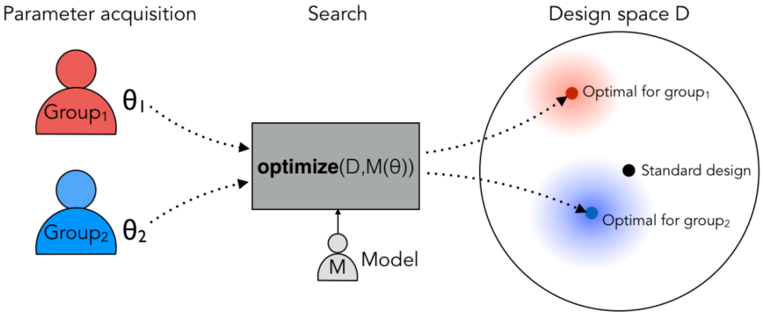


Figure: Source: Sarcar et al., **Ability-Based Optimization of Touchscreen Interactions**, IEEE Pervasive

Activities

Classwork

Recall that we did a 1D Fitts task in the previous class. Use the data and (1) calculate ID, (2) plot the linear model with the equation and (3) calculate the R^2. You can use Excel or any spreadsheet diagram.

Hick-Hyman Law

- Given n stimuli, associated one-for-one with n responses, the time to react (RT) to the onset of a stimulus is given by,

$$RT = a + b * log_2(n + 1)$$

- where a and b are empirically determined constants. Typical values for a is 200ms and b is 150ms/bit

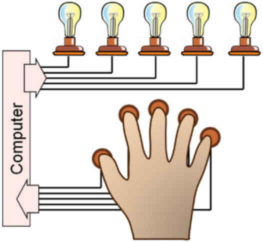


Figure: Source: Figure 7.18 (Mackenzie)

Hick-Hyman Law

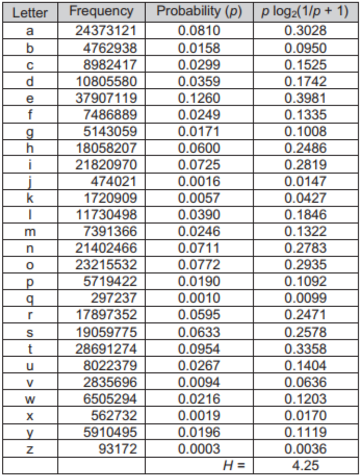
- If **some choice is more probable than others**, this reduces the information content, thus in turn reduces the choice reaction time
- For a set of alternatives with different probabilities, the information load H is

$$H = \sum p_i log_2 \left(\frac{1}{p_i} + 1 \right)$$

- Consider a choice selection task where the choice is among 26 alternatives and all appear with equal probability, the information content of the task is simply

$$H = log_2 (27) = 4.75 bits$$

Hick-Hyman Law



Letter	Frequency	Probability (p)	$p \log_2(1/p + 1)$
a	24373121	0.0810	0.3028
b	4762938	0.0158	0.0950
c	8982417	0.0299	0.1525
d	10805580	0.0359	0.1742
e	37907119	0.1260	0.3981
f	7486889	0.0249	0.1335
g	5143059	0.0171	0.1008
h	18058207	0.0600	0.2486
i	21820970	0.0725	0.2819
j	474021	0.0016	0.0147
k	1720909	0.0057	0.0427
l	11730498	0.0390	0.1846
m	7391366	0.0246	0.1322
n	21402466	0.0711	0.2783
o	23215532	0.0772	0.2935
p	5719422	0.0190	0.1092
q	297237	0.0010	0.0099
r	17897352	0.0595	0.2471
s	19059775	0.0633	0.2578
t	28691274	0.0954	0.3358
u	8022379	0.0267	0.1404
v	2835696	0.0094	0.0636
w	6505294	0.0216	0.1203
x	562732	0.0019	0.0170
y	5910495	0.0196	0.1119
z	93172	0.0003	0.0036
		H =	4.25

Figure: Source: Figure 7.19 (Mackenzie)

Hick-Hyman Law

- Card et al. (1983, 74) describe an example of a **telephone operator** selecting among ten buttons
- Landauer and Nachbar (1985) applied the Hick-Hyman law in measuring and predicting the time to select items in **hierarchical menus**
- Ruiz et al. (2008) used the Hick-Hyman law to model the perception, planning, and activation time for users to **switch modes with their non-dominant hands** in a tablet interface

Activities

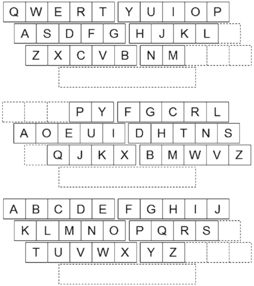
Example

A human operator attends to eight stimulus lights and presses one of eight keys when the corresponding light turns on. Two of the lights turn on more frequently than the others, accounting for 40 percent and 30 percent of all activations, respectively. The other lights activate with the same frequency. What is the information content of the task?

Activities

Example

Below are the layouts for the Qwerty and Dvorak keyboards, as well as an alphabetic layout proposed by Card et al. (1983, 63). Assuming the layouts are implemented as standard physical keyboards, which design provides the most even split between lefthand and righthand keying?



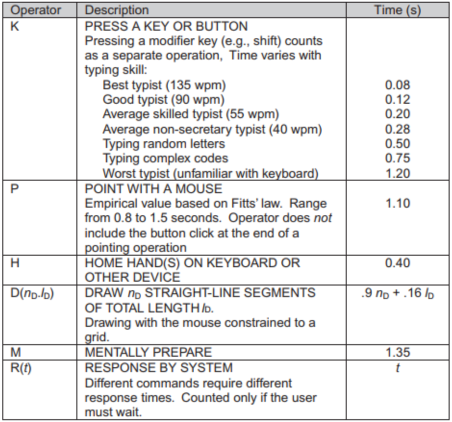
Keystroke level model

- Card et al. (1980; 1983, ch. 8) developed KLM that predict **error-free task completion time**
- The models works with four motor-control operators (K = keystroking, P = pointing, H = homing, D = drawing), one mental operator (M), and one system response operator (R):)

$$t_{EXECUTE} = t_k + t_p + t_H + t_D + t_M + t_R$$

Keystroke level model

Note that **empirical parameters can be updated**

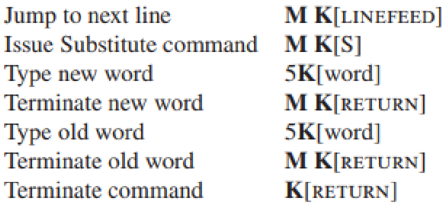


Operator	Description	Time (s)
K	PRESS A KEY OR BUTTON Pressing a modifier key (e.g., shift) counts as a separate operation, Time varies with typing skill:	
	Best typist (135 wpm)	0.08
	Good typist (90 wpm)	0.12
	Average skilled typist (55 wpm)	0.20
	Average non-secretary typist (40 wpm)	0.28
	Typing random letters	0.50
	Typing complex codes	0.75
	Worst typist (unfamiliar with keyboard)	1.20
P	POINT WITH A MOUSE Empirical value based on Fitts' law. Range from 0.8 to 1.5 seconds. Operator does *not* include the button click at the end of a pointing operation	1.10
H	HOME HAND(S) ON KEYBOARD OR OTHER DEVICE	0.40
$D(n_D.l_D)$	DRAW n_D STRAIGHT-LINE SEGMENTS OF TOTAL LENGTH l_D. Drawing with the mouse constrained to a grid.	$.9\ n_D + .16\ l_D$
M	MENTALLY PREPARE	1.35
R(*t*)	RESPONSE BY SYSTEM Different commands require different response times. Counted only if the user must wait.	*t*

Figure: Source: Figure 7.20 (Mackenzie)

Keystroke level model

- On one system, POET, the required sequence of subtasks was:



Jump to next line	**M K**[LINEFEED]
Issue Substitute command	**M K**[S]
Type new word	5**K**[word]
Terminate new word	**M K**[RETURN]
Type old word	5**K**[word]
Terminate old word	**M K**[RETURN]
Terminate command	**K**[RETURN]

- The task required four mental operations (M) and 15 keystroking operations (K):

$$t_{EXECUTE} = 4 * t_M + 15 * t_k = 4 * 1.35 + 15 * 0.23 = 8.85s$$

Keystroke level model

Predicted and observed values are very close

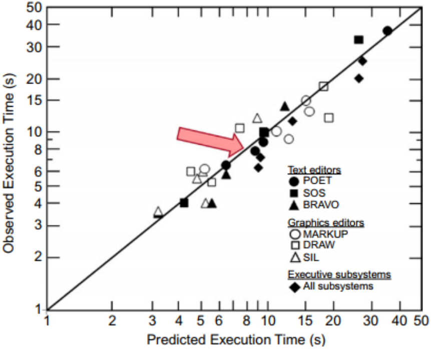
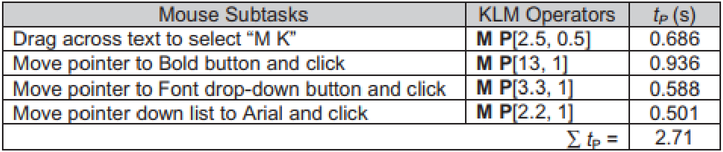


Figure: Source: Figure 7.21 (Mackenzie)

Keystroke level model

- Consider the editing operations to change the font style and font family for text, e.g., the word "M K"
- For mouse, four pointing operations are required: select the text, select the Bold, select the drop-down arrow in the Font list, and select Arial
- Table belows show KLM operators, where P is written in the format P[A, W]. The total time is

$$t_{EXECUTE} = 4 * t_M + \sum t_p = 4 * 1.35 + 2.71 = 8.11s$$



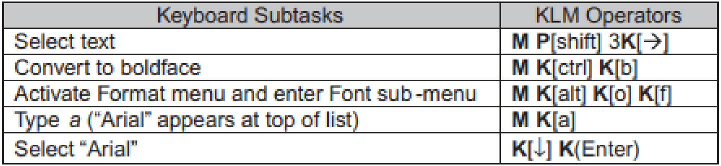
Mouse Subtasks	KLM Operators	t_P (s)
Drag across text to select "M K"	**M** P[2.5, 0.5]	0.686
Move pointer to Bold button and click	**M** P[13, 1]	0.936
Move pointer to Font drop-down button and click	**M** P[3.3, 1]	0.588
Move pointer down list to Arial and click	**M** P[2.2, 1]	0.501
	$\sum t_P$ =	2.71

Figure: Source: Figure 7.24 (Mackenzie). Note that P of Fitts' Law contains target selection as well, i.e., the click.

Keystroke level model

- The total time when using a keyboard is

$$t_{EXECUTE} = 4 * t_M + 12 * t_k = 4x1.35 + 12x0.75 = 14.40s$$



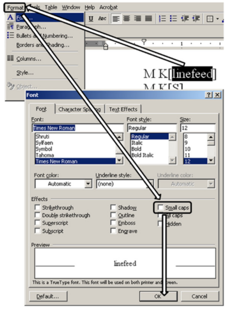
Keyboard Subtasks	KLM Operators
Select text	**M** **P**[shift] 3**K**[→]
Convert to boldface	**M** **K**[ctrl] **K**[b]
Activate Format menu and enter Font sub-menu	**M** **K**[alt] **K**[o] **K**[f]
Type *a* ("Arial" appears at top of list)	**M** **K**[a]
Select "Arial"	**K**[↓] **K**(Enter)

Figure: Source: Figure 7.25 (Mackenzie). Typo note: The first task should be M K[shift] 2K[→]. Btw, why the last subtask doe not have a M operator?

Activities

Classwork

This task requires you to predict the time of mouse vs. keyboard using KLM. For this question, assume $t_K = 0.4$ seconds. For t_P, use Figure 7.17 in previous slide where x is *ID*. Provide a KLM breakdown of all operations for both mouse and keyboard. Note that you don't have to worry about being exact, rough reasonable estimates are ok.



M K[linefeed]

M K[S]

Font: Times New Roman

Font style: Regular

Size: 12

Font color: Automatic

Underline style: (none)

Underline color: Automatic

Preview: linefeed

This is a TrueType font. This font will be used on both printer and screen.

Questions