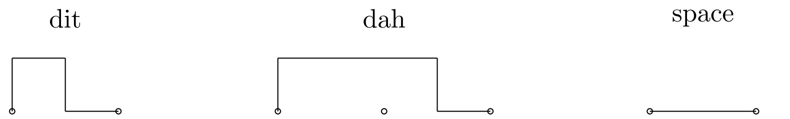
CW Keyers

Anders Helmersson
SM5KAE
email: anders@helmersson.one

September 5, 2025

1 The Morse code

The Morse code is built up from three primitive elements: dit, dah and space. The (character) space and dit have equal unit length and dah is twice as long. Note that I include the trailing inter-element space in the dit and dah symbols.



The spaces can be either a character space (1 unit) or a word space (3 units). The Morse code can be built up from six sequences, which can be combined freely: dit, dah, dit-space, dah-space, dit-space-space-space and dah-space-space-space.

1.1 Information

Now let us look at how to maximize the information that can be transferred in a Morse message. Choose the probabilities for each element according to its length:

- a dit is p
- a dah is p^2
- a dit-space is p^2
- a dah-space is p^3
- a dit-space-space-space is p^4
- a dat-space-space-space is p^5

This should sum up to 1. Thus, $p + 2p^2 + p^3 + p^4 + p^5 = 1$, which results in $p = 0.44859$. We also conclude that dits are more than twice ($1/p$) as common as dahs, and the probability for spaces is somewhere in between.

2 Paddles and keyers

When sending CW code we often use a CW key or manipulator, with one or two paddles. A single paddle has three positions for space (0), dit (1) and dah (2). The critical timing applies to the short spaces and dits, the timing for dahs is more relaxed. The Iambic key has two paddles: one for dits and the other for dahs. If both paddles are pressed or squeezed it becomes position 3.

Whatever statistical model you use, dits are most common. It appears obvious that in order to reduce the effort needed to produce a Morse message, we should follow the rule: *avoid wasting movements on dits.*

In the diagrams below different keyers are illustrated. The Single keyer produces spaces, dits and dahs only according to the position of the paddle. All other keyers add some element when the paddle is *moved.*

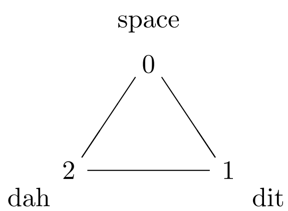
The Iambic keyer produces dit-dahs when both keys are pressed (squeezed), but a single dah is inserted when moving from 1 to 3

The Dit-priority keyer produces dits when both keys are pressed, but a single dah is inserted when moving from 1 to 3. The Ultimatic and Dah-priority keyers are similar, but have a different behavior in the squeezed position (3).

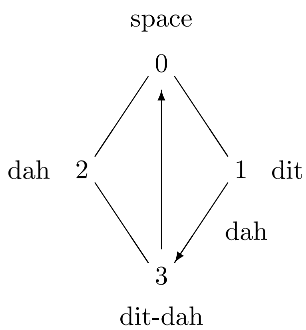
None of these keyers comply to the rule above, since they all produce a sequence of dahs while in position 2; you need to move the paddle to produce dits.

The *Dactylic* keyer (`https://github.com/ahelmersson/dactylic`) always produces dits when the position is kept, dahs when moving the paddle.

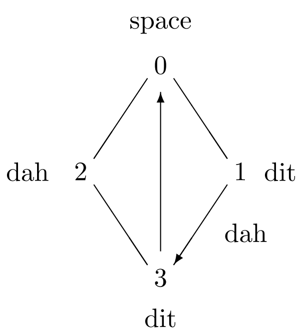
Single



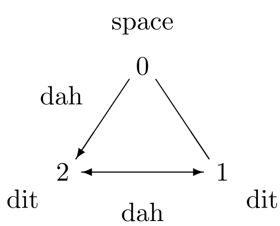
Iambic



Dit prio



Dactylic



3 Evaluation

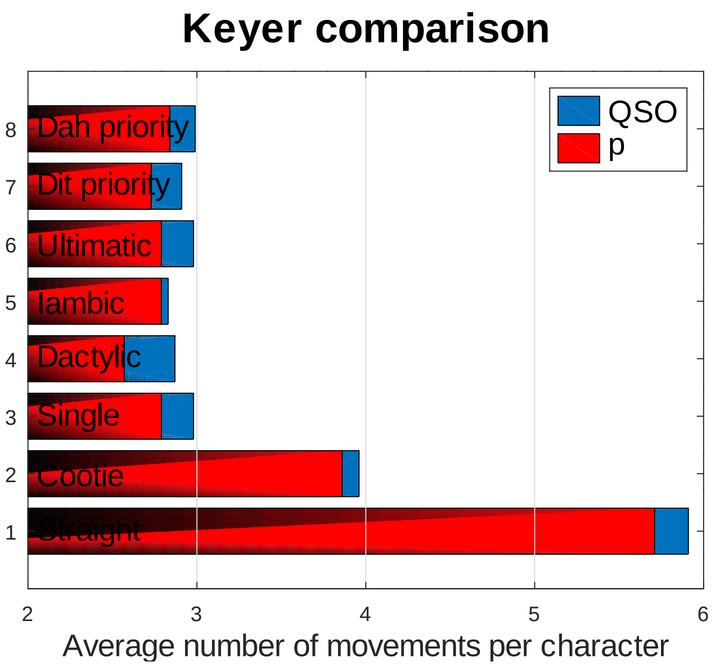
Below, I evaluate the keyers using movements. I count one movement for pressing the paddle and one movement for its release. Moving a single paddle from left to right or opposite, also counts as one movement. The table shows the average number of movements per character or symbol. The second and third columns give the average number of movements with the element probabilities that maximizes the transferred information. The fourth column is based on statistics using QSO messages from the Morse code trainer and QSO generator by Joe Dellinger, Eric S. Raymond, Thomas Horsten and others.

keyer/element	probability/average	$p = 0.45$	QSO
dit	p	0.45	0.43
dah	p^2	0.20	0.32
space	$1 - p - p^2$	0.35	0.25
symbol length	$\frac{1+2p}{(1-p-p^2)(1+p)}$	3.74	4.22
Straight	$\frac{2}{1-p-p^2}$	5.71	5.91
Cootie	$1 + \frac{1}{1-p-p^2}$	3.86	3.96
Single	$m(p) = 2 + \frac{2p^2}{(1-p-p^2)(1+p)}$	2.79	2.98
Dactylic	$2 + \frac{p^2}{1-p-p^2}$	2.57	2.87
Iambic	$m(p)$	2.79	2.83
Ultimatic	$m(p)$	2.79	2.98
Dit priority	$m(p) - \frac{p^3}{1+p}$	2.73	2.91
Dah priority	$m(p) + \frac{p^3}{(1+p)^2}$	2.84	2.99

The symbol length does not include the trailing space.

3.1 The letter O

The main reason for the difference in number of movements for the two statistical models (last two columns in the table), is the coding of the letter O (Oscar). The code (dah-dah-dah) is six units long, not including the trailing space. The letter O is very frequent in English and should have a shorter code. Historically, the Morse code was probably based on reducing the number of elements rather than the length of the code. In addition, zero (0) and one (1), the most common digits, also have long codes (10 and 9 units, respectively).



4 Conclusions

The difference in performance is quite small. For any keyer, you need less than three movements per character on average. In cootie mode you need four movements and with a straight key, slightly less than six movements.

However, there are three keyers that appear as potential winners:

- Dactylic
- Iambic
- Dit priority

Their relative rank depends on the statistics of the code you send.

Choose keyer according to your personal preferencesr!