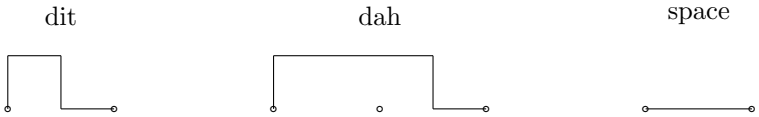
CW Keyers

Anders Helmersson
SM5KAE
email: anders@helmersson.one

September 25, 2025

1 The Morse code

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



The spaces can be either a character space (one unit) or a word space (three 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.

2 Paddles and keyers

When sending CW code we often use a 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*.

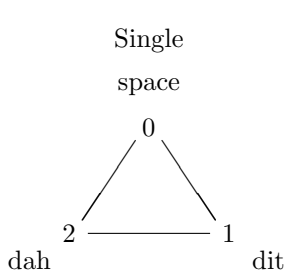
In the diagrams below different keyers are illustrated. The *Single* keyer produces spaces, dits and dahs only according to the position of the paddle. All other keyers add some element when the paddle is *moved*.

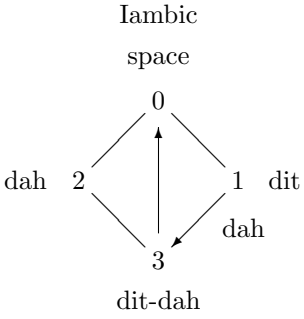
The Iambic keyer produces dit-dahs when both keys are pressed (squeezed), but a single dah is inserted when moving from 1 to 3

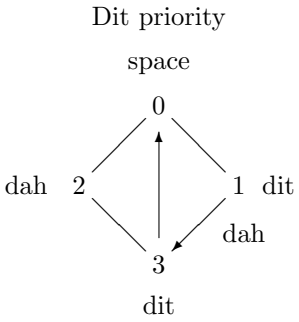
The *Dit-priority* keyer produces dits when both keys are pressed, but a single dah is inserted when moving from 1 to 3. The *Ultimatic* and *Dah-priority* keyers are similar, but have a different behavior in the squeezed position (3).

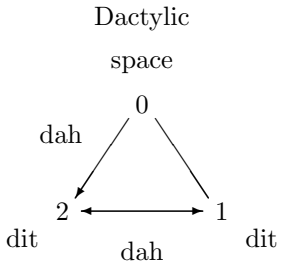
None of these keyers comply to the rule above, since they all produce a sequence of dahs while in position 2; you need to move the paddle to produce dits.

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









3 Comparing keyers

In order to evaluate the keyers we need some statistical model of the Morse code that we send.

3.1 Information

Let us first 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:

dit:	p	dit-space:	p^2	dit-space-space-space:	p^4
dah:	p^2	dah-space:	p^3	dah-space-space-space:	p^5

These 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.

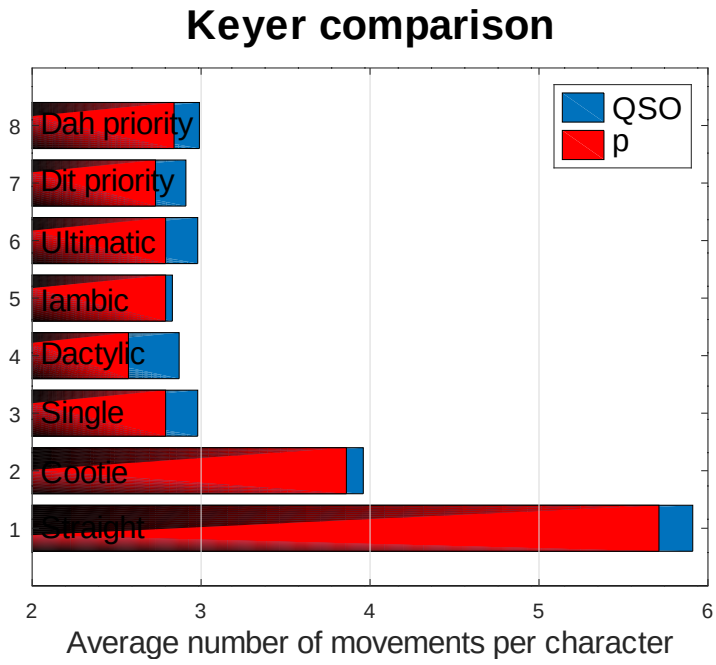
The maximum information transferred is $1.1565 \approx 133/115$ bits per time unit.

3.2 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	$1 + \frac{1+2p}{(1-p-p^2)(1+p)}$	4.74	5.22
Number of elements	$1 + \frac{1}{1-p-p^2}$	3.86	3.96
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 includes the trailing character space.



3.3 The letter O

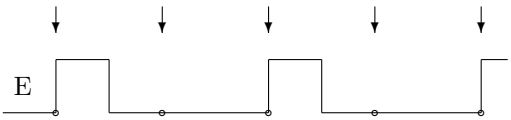
One reason for the difference in number of movements for the two statistical models (last two columns in the table), is the presence of symbols including consecutive dahs. The most prominent is the letter O (Oscar). The code (dah-dah-dah-space) is seven units long, including the trailing space. The letter O is very frequent in English and should have had 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, common digits like zero (0), one (1) and nine (9), also have long codes (11 and 10 units).

4 Keyer timing

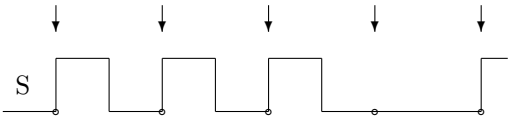
The timing logic tries to spread out the decision intervals as evenly as possible. Note that this logic can be used for any keyer. In the pictures below the decisions are taken at the arrows depending on the position of the key/paddle.

During each interval the keyer looks for changes in the position of the key/paddle relative to the element that it is sending. This is performed using a simple state machine, which is sampled at each decision point.

The letter that requires the fastest movements is E. For sending a sequence of Es, we need to make a movement each time unit (a press during the first interval and a release in the second). This applies to all common keyers.



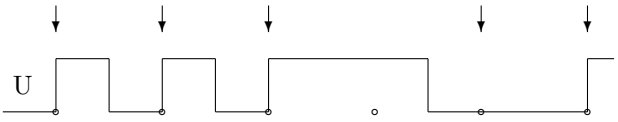
For the letter S, the decisions can be taken as late as possible for each element. We press the dit paddle to start sending. Then, we hold the dit paddle until the beginning of the last dit and release. If we release too early, we send the letter I, and if we release too late, we get an H. We must be accurate within one time unit ($\pm\frac{1}{2}$ units).



The correctly timed character space is inserted automatically by the keyer, but the word space need typically manual timing.

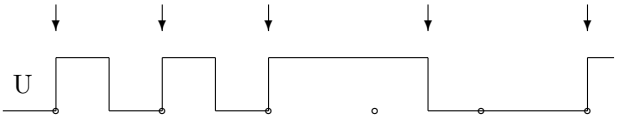
4.1 Late decisions

If the decisions are taken as late as possible, just before the next element begins, we get the following timing for the letter U. The interval during sending a dah is twice as long as the remaining intervals.

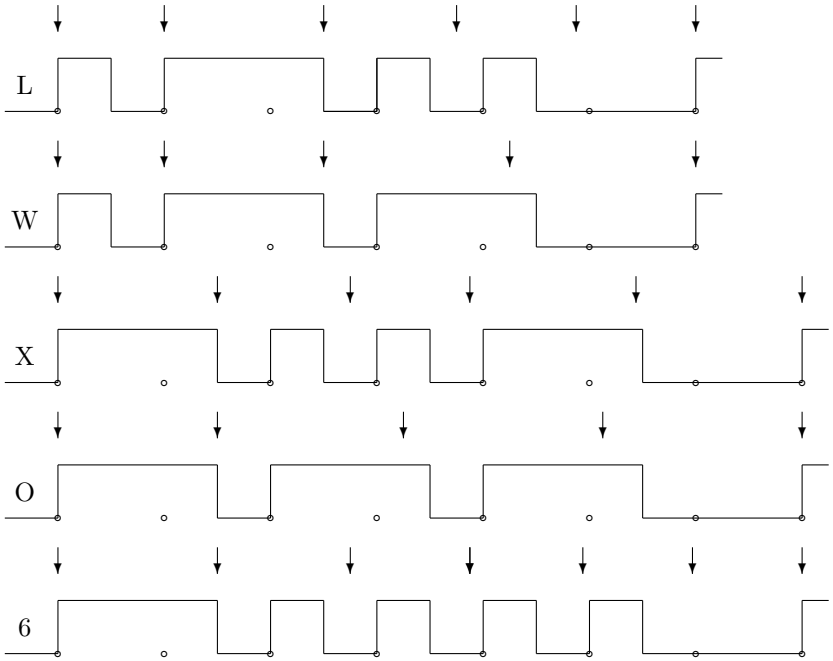


4.2 Relaxed timing

We will now spread out the intervals more evenly. For the letter U, we can relax the timing during the dah by moving the decision half a time unit earlier compared to the late decision.



The general logic is as follows. The prelatching is set to zero at the beginning of each symbol. The prelatching interval (compared to late decisions) is half the prelatching interval of the previous element; if the present element is a dah, we add half a time unit.



5 Conclusions

The difference in performance between the keyers 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 preferences!