

Shutter Sync, when failure provides enlightenment

19 July 2017 · automation, critical thinking, illusion, investigation

Shutter sync is an interesting artefact generated when we video moving objects. Take a look at this video of a Helicopter taking off:

I

Notice how the boats are moving as normal, but the rotors appear to be barely moving at all. This isn't a 'Photoshop'. It's an effect of video camera's frame rate matching the speed/position of the rotors. Each time the camera takes a picture or 'frame' the rotors happen to be in approximately the same relative position.

The regular and deterministic behaviour of both machines enables the helicopter to appear to be both broken and flying. The rotors don't appear to be working, while other evidence suggests its rotors are providing all the lift required.

What's so exciting is that this tells us something useful, as well as apparently being a flaw or fail. We could both assume the rotors move with a constant rotation, and estimate a series of possible values for the speed of the rotors, given this video.

Your automated checks/tests can be exhibit this too. Take for example a check that often/always 'fails'. But when you examine the software with other tools the problem disappears.

This might be a probe effect (https://en.wikipedia.org/wiki/Probe_effect) - that is, the 'bug' may only happen because of the testing tool. This is actually quite common. It was a bugbear of mine in the days of pre-webdriver browser automation e.g. Selenium RC, as RC inserted a lot of JavaScript into the page - often resulting in erroneous behaviour.

The 'failure' could also be a race condition. The regular systematic behaviour of the the testing framework, interacts with near perfect timing with the software being tested. The checking code, sees the problem frequently & repeatedly - as it always checks in the narrow window of time, when there is a problem.

Automated test/check 'failures' like the above are often dismissed immediately as things to work-around or fix. While it might make sense to 'clean' this from our results, we could miss potentially valuable avenues for testing. The 'failing' test is presenting us with information. That information might be more valuable than a clean pass/fail result - especially if the apparent failures have an inconsistency of this kind.

Just as with shutter sync, where we determine the behaviour of the rotors from the video. We can glean useful information from the 'failing' test/check. Investigation of these test 'failures' might show that the GUI is not quite in sync with the database or other users screens etc.

Maybe when the UI suggests an action is done - the user/system could actually still write some data for a short while. Or two events that as far as an API shown happen sequentially - in reality happen at the same time.